UNIVERSITÄT
KOBLENZ · LANDAU

unikorn
Computer Networks Group

Proactive Content Placement in Information-Centric Connected Vehicle Environments

by

Dennis GREWE

Approved Dissertation thesis for the partial fulfilment of the requirements for a
Doctor of Natural Sciences (Dr. rer. nat.)
Fachbereich 4: Informatik
Universität Koblenz-Landau

Chair of PhD Board:	Prof. Dr. Ralf LÄMMEL, University of Koblenz and Landau
Chair of PhD Commission:	Prof. Dr. Susan P. WILLIAMS, University of Koblenz and Landau
Examiner and Supervisor:	Prof. Dr. Hannes FREY, University of Koblenz and Landau
Further Examiners:	Prof. Dr. Christian F. TSCHUDIN, University of Basel

Date of the doctoral viva: 3rd September 2021

Abstract

Connected vehicles will have a tremendous impact on tomorrow's mobility solutions. Such systems will heavily rely on information delivery in time to ensure the functional reliability, security and safety. However, the *host-centric* communication model of today's networks questions efficient data dissemination in a scale, especially in networks characterized by a high degree of mobility.

The Information-Centric Networking (ICN) paradigm has evolved as a promising candidate for the next generation of network architectures. Based on a loosely coupled communication model, the in-network processing and caching capabilities of ICNs are promising to solve the challenges set by connected vehicular systems. In such networks, a special class of caching strategies which take action by placing a consumer's anticipated content actively at the right network nodes in time are promising to reduce the data delivery time.

This thesis contributes to the research in active placement strategies in information-centric and computation-centric vehicle networks for providing dynamic access to content and computation results. By analyzing different vehicular applications and their requirements, novel caching strategies are developed in order to reduce the time of content retrieval. The caching strategies are compared and evaluated against the state-of-the-art in both extensive simulations as well as real world deployments. The results are showing performance improvements by increasing the content retrieval (availability of specific data increased up to 35% compared to state-of-the-art caching strategies), and reducing the delivery times (roughly double the number of data retrieval from neighboring nodes).

However, storing content actively in connected vehicle networks raises questions regarding security and privacy. In the second part of the thesis, an access control framework for information-centric connected vehicles is presented. Finally, open security issues and research directions in executing computations at the edge of connected vehicle networks are presented.

Kurzfassung

Die voranschreitende Vernetzung von Fahrzeugen wird einen erheblichen Einfluss auf die Mobilitätslösungen von Morgen haben. Solche Systeme werden stark auf den zeitnahen Austausch von Informationen angewiesen sein, um die funktionale Zuverlässigkeit, Sicherheit von Fahrfunktionen und somit den Schutz von Insassen zu gewährleisten. Allerdings zeigt sich bei näherer Betrachtung der verwendeten Kommunikationsmodelle heutiger Netzwerke, wie beispielsweise dem Internet, dass diese Modelle einem *host-zentrierten* Prinzip folgen. Dieses Prinzip stellt das Management von Netzwerken mit einem hohen Grad an mobilen Teilnehmern vor große Herausforderungen hinsichtlich der effizienten Verteilung von Informationen.

In den vergangen Jahren hat sich das Information-Centric Networking (ICN) Paradigma als vielversprechender Kandidat für zukünftige datenorientierte mobile Netzwerke empfohlen. Basierend auf einem lose gekoppelten Kommunikationsmodell unterstzützt ICN Funktionen wie das Speichern und Verarbeiten von Daten direkt auf der Netzwerkschicht. Insbesondere das aktive, gezielte Platzieren von Daten nahe der Benutzer stellt einen vielversprechenden Ansatz zur Erhöhung der Datenbereitstellung in mobilen Netzen dar.

Die vorliegende Arbeit legt den Fokus auf die Erforschung von Strategien zum orchestrieren und aktiven Platzieren von Daten für Fahrzeuganwendungen im Netzwerk für mobile Teilnehmer. Im Rahmen einer Analyse unterschiedlicher Fahrzeugapplikationen und deren Anforderungen, werden neue Strategien für das aktive Platzieren vorgestellt. Unter Verwendung von Netzwerksimulationen werden diese Strategien umfangreich untersucht und in im Rahmen eines prototypischen Aufbaus unter realen Bedingungen ausgewertet. Die Ergebnisse zeigen Verbesserungen in der zeitnahen Zustellung von Inhalten (die Verfügbarkeit spezifischer Daten wurde im Vergleich zu existierenden Strategien um bis zu 35% erhöht), während die Auslieferungszeiten verkürzt wurden.

Allerdings bedingt das aktive Platzieren und Speichern von Daten auch Risiken der Datensicherheit und Privatsphäre. Auf der Basis einer Sicherheitsanalyse stellt der zweite Teil der Arbeit ein Konzept zur Zugriffskontrolle von gespeicherten Daten in verteilten Fahrzeugnetzwerken vor. Abschließend werden offene Problemstellungen und Forschungsrichtungen im Kontext Sicherheit von verteilten Berechnungsarchitekturen für vernetze Fahrzeugnetzwerke diskutiert.

Acknowledgements

During my doctorate, I am indebted to several people without this thesis would not have been successful and to whom I would like to express my thanks.

Firstly, I would like to thank my supervisor Hannes Frey for giving me the opportunity for this work. Without his openness to supervise a student in collaboration with a company, this journey would have never happened. Furthermore, I would like to thank him for all the support, the frequent discussions and hints, his experience, the guidance throughout all the years and the trust he put in me. Furthermore, I would like to thank Christian F. Tschudin for reviewing this thesis.

Throughout the journey, I was fortunate enough to have the support of two institutions. I would like to thank the present and past associates of the Computer Networks Group at the University of Koblenz (Unikorn), especially Florentin, Jovan, Sabine, Daniel and Ahmad for all the support, feedback and the discussions. The same applies to the associates of the Communication and Network Technologies group of Robert Bosch GmbH for the support throughout the years. Special thanks to my mentor Marco Wagner, who played a significant role in my research. I thank him for convincing me to start the journey of pursuing a PhD, for all the guidance, countless discussions, and the confidence in my abilities. Special thanks go to Sebastian Schildt for acting as a sparring partner and challenging my research ideas and Naresh Nayak for helping me to finalize this thesis. I was glad to supervise excellent masters students who contributed to this thesis. I want to thank Pavithra Rao, Andong Tan, Matthias Paul and Johannes Kloeckner.

Furthermore, I would like to thank all the people outside Unikorn and Robert Bosch GmbH. Special thanks to Christopher Scherb and Claudio Marxer from the University of Basel for the cooperation in research and development of named functions for connected vehicles. The same applies to Dirk Kutscher, Mayutan Arumaithurai, Ioannis Psaras, with whom I had the pleasure of discussing ideas and research directions.

Last but not least, this journey would also not be possible with all the backing and support of my family and friends. Especially, I would like to thank my parents for all the encouragement and support throughout my whole life.

Finally, my special thanks to my wife Jessica for all the support and the patience that I experienced during the periods of frustration and self-doubts. This thesis is dedicated to her and our little star.

Renningen, October 2021

Contents

1 Introduction

> "If we can dream it, we can do it"
>
> Walt Disney

Connected vehicles will have a tremendous impact on tomorrow's mobility solutions. Billions of vehicles will be processing and provisioning information in the network, forming a *heterogeneous vehicular network*. In order to process the vast amount of the data collected and used by future automotive applications, vehicular systems need to complement information analysis performed by their built-in components, and aggregated and fusioned by cloud backends or computing resources at the edge of the network. For example, such systems are able to provide map and road condition updates in time. The communication infrastructure is expected to play a key role in supporting and enhancing automotive services, using both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication.

However, when looking into today's communication networks, it can be seen that they work in terms of end-to-end communication. The interconnection of billions of devices poses new opportunities but also introduces new challenges to the network. For example, the efficient dissemination of data, especially when participants are characterized by a high degree of mobility (e.g., vehicles, trains, etc.). Furthermore, it also challenges the network on different levels such as network management by maintaining address changes while switching multiple times between network Access Points (APs). This results in many handovers and frequent changes of the data delivery routes.

Such development demands for new investigations working towards novel network technologies and paradigms to provide efficient data communication, especially in high dynamic mobile networks[1].

1.1 From Simple Vehicles to Vehicular IoT

The search for efficient mobility solutions to travel faster from A to B is driving humankind since time immemorial. In 1885, the invention of the automobile, build in "production" has started a success story through the last 100 years and had a revolutionary impact on subsequent mobility solutions.

While the development of the automobile was driven by efficient engines and increasing the range in the beginning, it has changed over the years to invent systems to make the automobile more safe and driving more comfortable. For example, noteworthy safety developments are the seat belt (cf. [3]) or the air bag (cf. [4]) which decreased the number of fatal accidents and passengers seriously injured. Examples of comfort-driven developments include the in-vehicle air conditioning or the car radio.

In the last decades, the rapid development in information technologies has enabled mobile devices and machines to be equipped with micro-controllers and transceivers to form cyber-physical systems. As part of the the so-called Internet of Things (IoT), this also includes vehicular systems. Equipped with different sensors and transceivers, connected vehicles are able to offer and participate in information sharing among each other, infrastructure components and cloud environments, by choosing from a variety of wireless networking technologies [5].

[1] The content of this section is based on the published work in [1, 2]. Parts of it are extracted from these sources.

An example for a vehicular system which evolved in the past decades are navigation systems. First introduced as stand-alone solutions which were added to the dashboard, the systems evolved as built-in components up to connected systems receiving information from ITS stations, e.g., about traffic jams in near real-time.

The connectivity to the Internet offers cars and its passengers the possibility to participate in many other services and applications. It enables in-vehicle and ex-vehicle services and applications to exchange information between other vehicles, pedestrians, or any other computer system - also known as Vehicle-to-Everything (V2X) [5]. In general, upcoming vehicular applications can be clustered into different classes such as (i) connected infotainment - consuming information for passengers (e.g. audio/video streaming or point of interests nearby), (ii) connected driver assistance - consuming information about the vicinity such as road conditions or the structure of the road network, as well as (iii) connected automated driving - consuming information about surrounding vehicles, obstacles, hazards, or high detailed maps.

Networking vehicular systems to be able to exchange information with its surroundings is expected to play an important role for upcoming automotive applications such as automated driving and to further increase the safety and the comfort of passengers following the ideas 100 years ago.

1.2 Exemplary Use Cases

Use cases within the vehicular IoT can be separated into (i) *safety-driven* and (ii) *comfort-driven* applications [6]. Examples for safety-driven applications are hazard and collision warnings - notifying the driver about any hazardous or dangerous situations on the road ahead. Examples for comfort-driven applications are traffic management and cooperative navigation – optimizing traffic flows or avoiding traffic jams, as well as infotainment applications – entertaining passengers during the journey. While in the past decades, applications have been developed constantly, the advent of automated driving has rapidly increased the development cycles today. Looking into automated driving applications, it can be seen that they will heavily rely on timely information sharing and receiving of computation results. For example, the retrieval of information about traffic flows to optimize the traffic on the road and to avoid accidents. It is expected that each car will generate approximately 4,000 Gigabytes of data per day[2], a figure that will undoubtfully challenge future networks.

Such upcoming vehicular systems will be equipped with sensors (e.g., front, rear, blind spot cameras, radar systems, ultrasonic or brightness sensors) to monitor the environment. Autonomous cars will communicate with each other as well as with infrastructure components to share sensed information. Processing such big amounts of data by the vehicles themselves may be impossible, while pushing everything up to the cloud requires excessive amounts of bandwidth, but most importantly induces prohibitive round-trip latency [2].

The following subsections introduce two fictitious but realistic use case scenarios from the automotive IoT and illustrate the limitations of today's communication systems.

1.2.1 The Electronic Horizon

One early example of a connected vehicle application is the *electronic horizon* which is already being developed today. It describes a cloud based virtual sensor that computes an

[2] https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving/

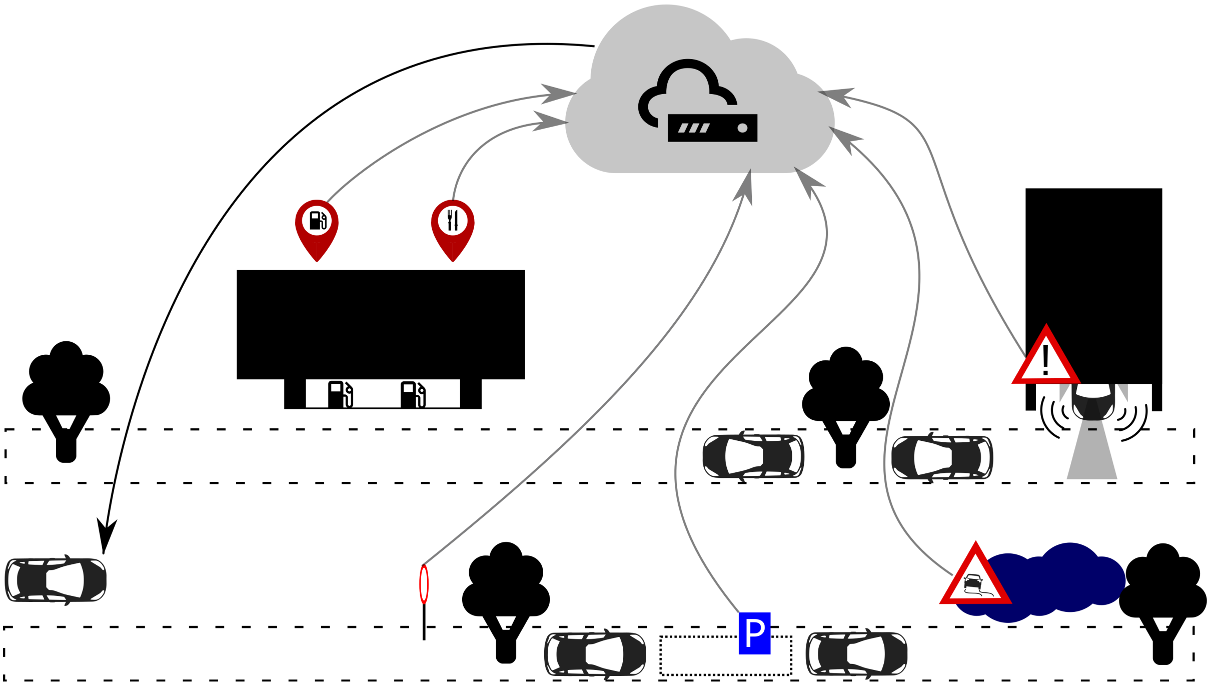


Figure 1.1: Exemplary illustration of the Electronic Horizon use case. The environmental model computed by a function in the cloud contains a combination of personalized, popular and local information. Such model can be downloaded by the car during the journey which allows for new features and functions.

environmental model of the vicinity including map data, the vehicle's mobility model as well as additional data regarding the road ahead. The result is downloaded by the car during the journey. Figure 1.1 illustrates the idea of the electronic horizon. For example, the vehicle utilizes local data from in-vehicle systems such as the Adaptive Cruise Control (ACC) – a system to regulate the vehicle's speed according to vehicles ahead or speed limits, the Electronic Stability Program (ESP) – a system to detect and reduce loss of traction, or other equipped radar and video systems as well as from external systems. Such data may include *common (popular)* information such as topographical information, traffic infrastructure, traffic or hazard information, *geo-specific* information such as available parking spots nearby as well as *personalized* information such as driver's preferences in food or the energy consumption level of the own smart home [7].

Based on the fusion of all data in the cloud, an environmental model is computed by the electronic horizon function providing a detailed preview of the road ahead. A vehicle can download the model from the cloud during the journey and make use of the information, for example for adaptive cruise (e.g. reduce velocity to catch next green light) and predictive power-train control (e.g. gear up and down to reduce fuel or battery consumption), adaptive navigation (e.g. based on the traffic ahead) or adaptive headlight adjustment (e.g. to spot to a hazard). The model itself contains a combination of personalized, popular and local information and is displayed to the driver individually such as the infotainment system or a head-up display, depending on the driver's needs and the visualization strategy of the car manufacturer.

Limitations of Today's Systems

Currently developed as a solution with cloud backends, there are several challenges for realizing the electronic horizon functionality in a large scale deployment:

- **Traditional Cloud Processing**: The fusion of data and computation of such a model requires a high amount of computing power, however downloading the relatively large model to the vehicle periodically during the journey dictates strict response deadlines. Acceptable response delays can be as low as a few milliseconds. For those cases, sending everything up to the cloud requires excessive amounts of bandwidth, but most importantly induces prohibitive round-trip latency [2].

- **In-Vehicle Processing**: The exchange of raw data between cars and an infrastructure as well as the fusion of data within the car can be another option. However, in-vehicle processing capabilities are restrained due to limited on-board compute capabilities[3]. Furthermore, each vehicle has to gather and process the required information while generating its own environmental model. This leads to highly inefficient resource usage.

Key Challenges

The major requirement of the electronic horizon use case is the high degree of mobility. Participants freely join and leave the network, while demanding for up tp date information of the environmental model which needs to be downloaded periodically during the journey. Furthermore, additional challenges are describe by other network aspects. For example, the access to computational resources is required to process expensive operations such as the fusion of large amounts of data produced, collected, received and processed in order to get the vital information from several sources nearby. Furthermore, dealing with bandwidth and latency limitations describe additional challenges to be addressed by the network. Finally, safety and security aspects need to be also taken into account to ensure the functional safety of the application (e.g., display a notification of a hazardous situation ahead) as well as to protect the access of personalized data.

1.2.2 Community-based Sensing

Community-based sensing describes an interconnected use case scenario in which mobile as well as stationary network participants use their built-in sensors, cameras and radar systems to stream sensed data to the network, in order to achieve a common goal. In this case, a community can consist of one or more mobile nodes such as cars, buses or pedestrians as well es stationary sensors for example placed at signals, or detection loops installed. Figure 1.2 illustrates the use case. For example, such information can be hazardous situations (e.g., red car in Figure 1.2) or information about available parking spaces (e.g., sensed by ultrasonic sensors of the green vehicle in Figure 1.2) in a certain geo-location.

Different in-vehicle and ex-vehicle automotive applications are able to consume and process such information in order to create added value, for example a parking service can create an environmental model of available spots nearby, which can be consulted by a vehicle (e.g., blue car in Figure 1.2).

[3] `http://www.nordsys.de/en/car2x-produkte-2.html`

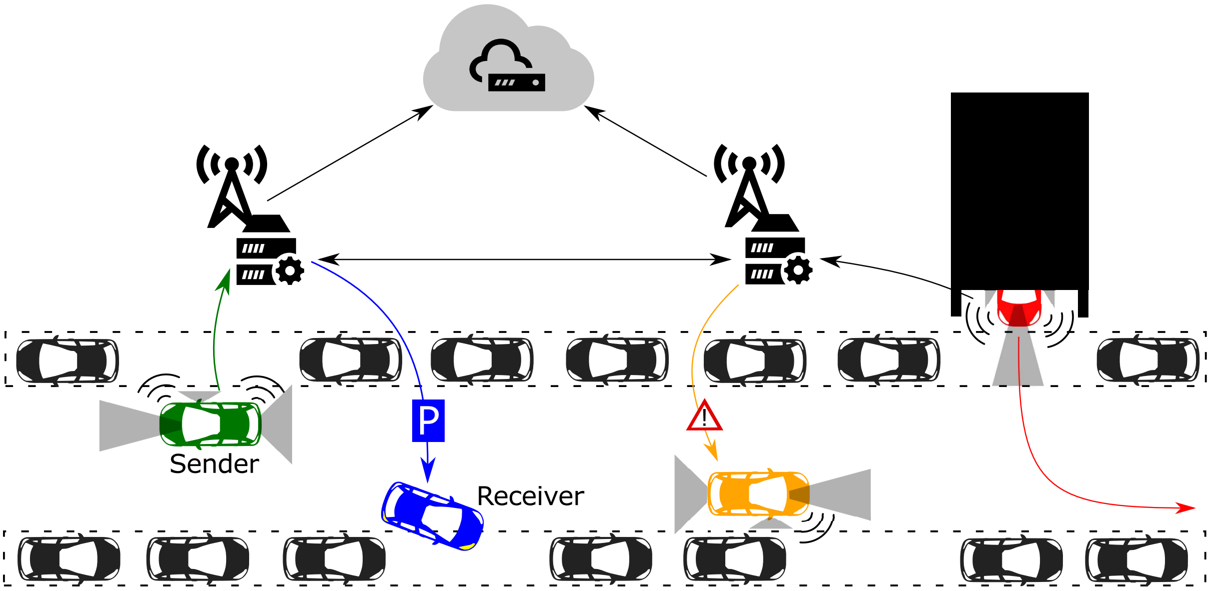


Figure 1.2: Exemplary illustration of the community-based sensing use case. Mobile nodes such as cars, buses or pedestrians as well es stationary sensors offer their data to be shared with others in order to create added value.

Limitations of Today's Systems

Currently, there are several challenges and limitations for realizing community-based sensing functionality in a large scale deployment:

- **Traditional Cloud Processing**: All the sensed information needs to be streamed towards centralized cloud backends in order to process the large amount of data. However, sending all data from a large amount of devices requires excessive amounts of bandwidth, and induces round-trip latency [2]. Furthermore, centralized cloud environments hinders a fair access to data and services in the market, which will be crucial to create a thriving economy of data [1].

- **Direct Vehicle-to-Vehicle Communication**: Another option is the introduction of direct information exchange between nodes in the vicinity. While it is expected that small sensor data can be transmitted in urban/rural areas, this is challenging for larger volumes of data or in non-residential roads where vehicle are likely to move with speeds that are prohibitive for direct communication (e.g., at motorways). Furthermore, finding the right participant providing the information is challenging in today's host-centric communication networks.

- **In-Vehicle Processing**: Consuming and processing all the information from sensors in the vicinity describes another option. However, in-vehicle processing capabilities are restrained due to limited on-board compute capabilities[4] which are different between the manufacturers and models.

[4] http://www.nordsys.de/en/car2x-produkte-2.html

Key Challenges

The major requirements of the community-based sensing use case is described by seamless interoperability. In the use case, the variety of the communication and processing capabilities of the participants – ranging from powerful cloud backends to small sensors – challenges interoperability between manufacturers and service providers. While interoperability is a challenge for accessing information across each other, discovering the right node providing the desired information is another challenging task, especially in host-centric driven networks. Moreover, mobility and network aspects such as bandwidth and latency are also challenging in such a decentralized use case environment. Finally, safety and security aspects need to be also addressed in the use case. Especially the question of who owns the data and is able to grant or restrict access to data [8]. This includes the issue of who is allowed to give away, and hence potentially, make profit from the information. This is especially challenging since such connected applications may stretch across national borders.

1.3 Enabling Information-Centric Networking for Connected Vehicles

In recent years, research activities in academia and industry have been working towards *data-oriented* networking approaches to overcome the challenges set by the classic host-oriented model. As one of the the potential paradigms, ICN has been identified to solve the issues set by mobile and heterogeneous networks such as connected vehicles (e.g., [9, 10, 11]). Based on a loosely coupled communication approach, the ICN paradigm separates data from its hosts. Applications always address *data* and *not hosts*. ICNs replace the host-centric view of classic networks with a *content-centric* paradigm. It is directly concerned about the data itself as the principal entity for information dissemination. Instead of node names or node identifiers, ICN works with naming schemes using *content identifiers*. Such identifiers provide access to data directly achieving a loosely coupled communication model [12, 13]. Therefore, the paradigm naturally facilitates in-network caching and processing as well as dealing with mobility.

Especially in mobile scenarios, the benefits of ICNs compared to host-centric networks are significant. Figure 1.3 illustrates the results of the request to response ratio for a mobile scenario (cf. Section 5.6 for the simulation setup) and describes the requesting effort at the consumer side to receive a certain content object from the network. In the first scenario, vehicular nodes send requests to a data server by establishing a dedicated end-to-end connection to the server (in this case using the UDP/IP protocol). This results in delivering the same data multiple times in the core network and a ratio value of 9.3% (low traffic) to 1.2% in higher traffic volumes. As the number of communication participants sharing the same network resources increases from low to high traffic volumes, the number of connections in the network also increases and therefore, the load on network resources. This is reflected by the decreasing values of the request to response ratio, while the number of participants increases. The results in this setup illustrate the inefficiency of the host-centric paradigm regarding the network resources in mobile scenarios. By changing the addressing scheme towards data, vehicles in the second scenario query the network for data which are provided by a data server. If multiple consumers are interested in the same data, ICN aggregates requests and supports multicast delivery intrinsically. As shown by the simulation results, ICN improves the request to response ratio by using the available network resources more efficiently (12.2% in low to 10.8% in mid traffic).

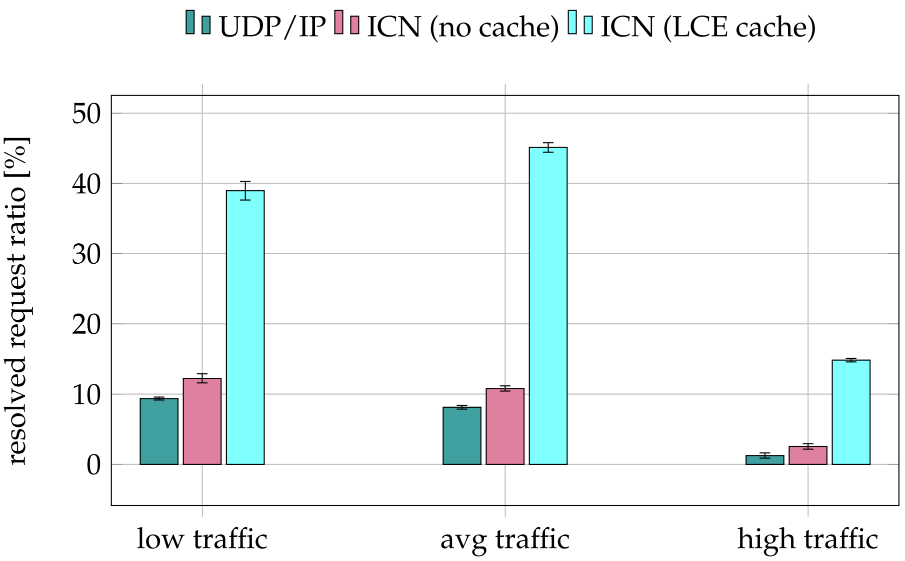


Figure 1.3: Comparison of IP and ICN communication in a mobile scenario: As part of a motorway deployment, vehicles request for data. As part of the UDP/IP scenario, each vehicle establishes a dedicated end-to-end connection to the data server. Due to the loosely coupled communication model of ICNs, request aggregation and intrinsic multicast support increases the request to response ratio. When caching is enabled at the network edge, the increased availability of data also increased the ratio and is promising to deal with the characteristics of mobile scenarios.

In the third scenario, in-network caching is enabled at edge nodes (e.g., Road-Side Unit (RSU) or cellular base stations along the road). Due to the loosely coupled communication model, data can be replicated and stored multiple times in the network to increase its availability. The results of the third scenario shows the potential of ICNs in mobile scenarios. Storing data closer to consumer reduces the delivery time and increases the request to response ratio up to 45% in mid traffic volumes.

In order to solve all the technical and socio-economic challenges described as part of the exemplary use cases, this thesis is built on the vision of an open and distributed data market place for future connected vehicle applications – a harmonized architectural design providing dynamic access to data and services. The vision introduces a common space for information exchange between different kinds of communication participants such as connected vehicle, cloud infrastructures, third party service providers, and other devices from the IoT (cf. Figure 1.4). The goal of the market place is to provide an open data space in which every entity is able to collect, provide and share data across the network, while every eligible entity is able to participate and consume available data [1].

The glue layer of the market place is created using the Information-Centric Networking paradigm. The capabilities such as the natural support of mobility and additional in-network features are the main reasons for using ICN as the underlying network architecture in the market place vision.

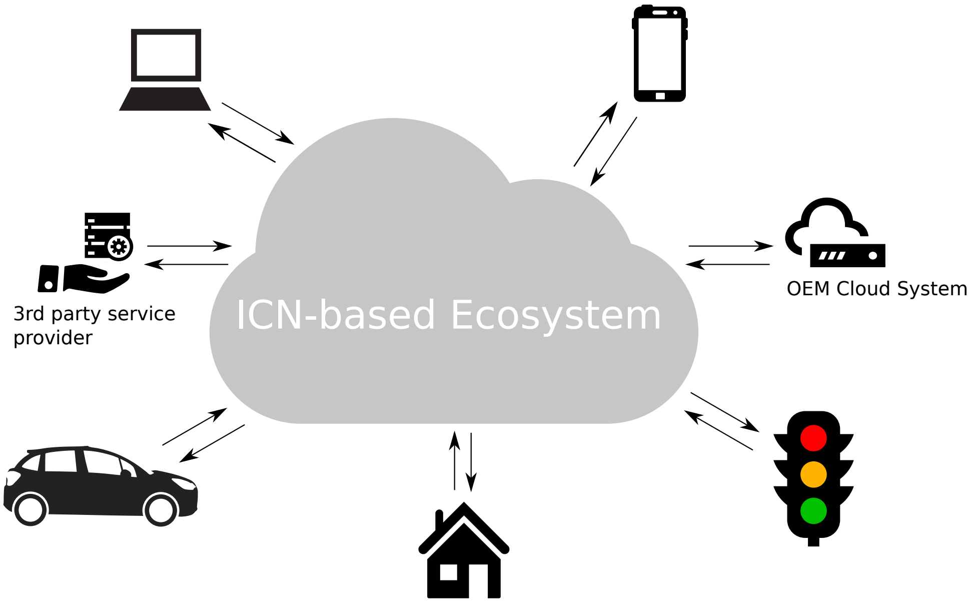


Figure 1.4: An ICN-based platform serving heterogeneous IoT environments such as connected vehicles.

As part of the vision, five different building blocks have been identified, in order to realize such a system:

- **Data Discovery**: This building block realizes the identification of data within the distributed market place. It allows a participant to search for a specific information or data sample and offers mechanisms to identify the meaning of data.
- **Data Dissemination**: This building block is in charge of transporting data packets between the participants of a distributed application. This includes forwarding strategies especially for mobile consumers and producers as well as network management and Quality of Service (QoS) demands.
- **Data Caching**: This building block provides mechanisms to deploy different caching strategies. The main aspect of such mechanisms is to ensure a high degree of availability of data and short retrieval times for consumers, while keeping the number of duplicates at a manageable level.
- **Data Security**: This building block implements mechanisms in order to ensure data integrity as well as privacy of the participants.
- **Access Control Management**: This building block realizes a system that allows data producers to define access policies and assign or withdraw access rights to specific participants or groups. Furthermore, it implements mechanisms that ensure access to consumers that are entitled.

As one of the building blocks, the in-network caching capabilities of ICNs have attracted the attention of researchers in academia and industry to improve the performance of the network. In-network caching describes a feature of a network architecture to store data closer to the consumers. The advantages of supporting such feature are twofold:

- *decrease the traffic in the core*: By caching data at multiple elements (e.g., at the edge of the network), the traffic within the core network is decreased, instead of transferring all requested data through the entire network.
- *reduce response times*: By caching data closer to consumers (e.g., at nodes in a certain geo-area such as North America or Europe), the response times between requesting and receiving a packet is reduced, while the quality of service for the consumer is improved.

In ICNs, there are two kinds of caching mechanisms: (i) *reactive caching* and (ii) *proactive caching*. *Reactive caching mechanisms* – a class of caching strategies which stores data at intermediate nodes during delivery – have shown performance improvements by increasing the availability of data closer to consumers (e.g., [14, 15, 16]). However, such strategies are not efficient for highly dynamic networks due to the fact that forwarding routes between the mobile participants change constantly.

A promising solution for highly dynamic networks is *proactive caching*. It describes a class of strategies taking action by placing a consumer's anticipated content at the right network nodes in time, before a request is sent by a consumer. As a result, proactive caching mechanisms are able to reduce the latency of content retrieval, and thus, provide a certain degree of quality for data delivery by reducing handover delays in WiFi and cellular networks (cf. [17]). However, placing the right content at the right node in-time is a non-trivial task, since network topology changes steadily in vehicular networks due to the high degree of mobility of the communication participants.

1.4 Contributions of this thesis

This thesis will present the research done in the field of proactive content placement in data-oriented connected vehicle environments, based on the exemplary use cases introduced in Section 1.2 and the environment presented in Section 1.3.

All contributions in this thesis have been investigated, validated and evaluated in sophisticated virtual environments as well as part of real world tests in small scale. In the following, the main research questions are presented and the contributions of the thesis are outlined.

1.4.1 Main Research Questions

There are three main research questions addressed in this thesis. Each of the main questions is separated into sub-questions which are used in this document to fulfill the overall research topic.

Q1 What are the benefits of placing automotive data proactively in the network and closer to consumers?

Q1.1 What kind of automotive data is beneficial to be cached proactively in the network?

Q1.2 Where do current networks (e.g. IP-based, ICN-based) fall short?

Q1.3 Can the availability of automotive data be increased when vehicles carry data passively through the network?

Q2 How to place automotive data proactively in data-oriented connected vehicle networks?

Q2.1 What kind of network architectures are useful for proactive data placement?

Q2.2 How to identify where automotive data is needed?

Q2.3 How to actually place automotive data within network component caches?

Q3 What are *security* related implications when caching data proactively in data-oriented connected vehicle networks?

Q3.1 What kind of security related challenges exist when introducing named data/functions in connected vehicle environments?

Q3.2 How to restrict the access to automotive data which is placed proactively in the network only for eligible users?

Q3.3 How can a (new) consumer access already cached automotive data in a non-trusted distributed environment while being highly mobile?

The first block of research questions target the benefits of placing automotive data proactively in the network. As a first step, the data classes worth to be stored close to the consumer need to be identified by analyzing different automotive applications and their data traffic. As a next step, the challenges of placing content in existing network architectures need to be analyzed in both today's host-centric as well as in data-oriented networks. The contributions of this block of research questions have also been published in [18, 1, 2].

The second block of research questions target the mechanisms to place data at the edge of the network. This includes the identification of where data is required in the network as well as the consideration of different caching architectures for efficient placement in the network. Based on the analysis of data-oriented network architectures, novel caching strategies are introduced and evaluated against the state-of-the-art in both extensive simulations as well as part of real world deployments. The contributions of this block of research questions have also been published in [18, 19, 20].

The third block of research questions target security implications of placing automotive data actively at the edge of the network. This includes the analysis of security related challenges in the context of data-oriented networks regarding connected vehicles. Especially, the decentralized fashion of data-oriented networking architectures describe a challenge to deal with access control of cached copies of data in the network. The contributions of this block of research questions have also been published in [21, 22].

1.4.2 Contributions to fulfill the Research Questions

The contributions in this thesis are listed in the following paragraphs and in their order of appearance in the manuscript. A detailed discussion of the fulfillment of the research questions is presented in Chapter 9.

ICN-based Open, Distributed Data Market Place for Connected Vehicles: Challenges and Research Directions

The paper introduces the vision of a open distributed market place running the ICN paradigm as the underlying network layer. Based on the introduction of different automotive use cases, the current challenges and research directions have been identified and discussed compared to the state-of-the-art literature.

The vision of the unified platform for connected vehicles has been published at the workshop of Convergent Internet of Things in the proceedings of the IEEE International Conference on Communications, May 2017. The paper was authored by D. Grewe and co-authored by M. Wagner and H. Frey. The content of the paper is used in Section 1.2 and Section 1.3 to introduce automotive use cases and to highlight open challenges in the research area of information-centric networks.

Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions

The paper introduces the mobile edge computing paradigm in conjunction with the information-centric networking paradigm. Based on a detailed futuristic vehicular scenario – Electronic Horizon – open challenges and research direction towards information-centric networking for connected vehicles are presented and discussed in detail.

The paper has been published at the workshop on Mobile Edge Communications in the proceedings of the ACM SIGCOMM conference, August 2017. The paper was authored by D. Grewe and co-authored by M. Wagner, M. Arumaithurai (University of Goettingen), I. Psaras (University College London) and D. Kutscher (Huawei Germany). The open challenges and research directions discussed in the paper are used in Section 1.2.

A domain-specific Comparison of Information-Centric Networking Architectures for Connected Vehicles

The article provides a detailed examination of the different available ICN approaches regarding the specific requirements of connected vehicles. While some preliminary publications showed the principal applicability of particular ICN architectures in vehicular ad-hoc networks, a detailed comparison has not been conducted yet. The article closes the gap by discussing and comparing the available ICN architectures in an automotive context and identifies open research questions in ICN.

The results of the survey are published in the IEEE Communications Survey and Tutorials Journal, May 2018. The article was authored by D. Grewe and co-authored by M. Wagner and H. Frey. The content of the paper is used in Chapter 4.

PeRCeIVE: Proactive Caching for ICN-based VANETs

The work in this paper introduces a proactive caching approach for information-centric vehicular networks. The approach shows that a directed placement of personalized, transient, large data will improve the performance of the network by distributing the content with a minimal number of replicas one-hop away from the consumer.

The results of the paper are published in the proceedings of the IEEE Vehicle Networking Conference, December 2016. The paper was authored by D. Grewe and co-authored by M. Wagner and H. Frey. The content of the paper is used in Section 5.3.

ADePt: Adaptive Distributed Content Prefetching for Information-Centric Connected Vehicles

In this paper, an adaptive decentralized prefetching mechanism for ICNs in vehicular scenarios is proposed and evaluated. By placing relevant data in caches near to consumers proactively, the overall service quality and delivery rate in the network is improved. The algorithm is evaluated using simulation based on a real V2X motorway testbed in Austria.

The results of the paper are published in the proceedings of the IEEE Vehicular Technology Conference Spring, June 2018. The paper was authored by D. Grewe and co-authored by M. Wagner, S. Schildt and H. Frey. The content of the paper is used in Section 5.4.

A Real World Information-Centric Connected Vehicle Testbed supporting ETSI ITS-G5

This paper proposes an architectural concept in which ICN and the inter-vehicle communication system ETSI ITS-G5 (based on IEEE 802.11p) coexist and complement each other. Based on the *OpenC2X* open source platform, a prototype implementation is introduced and verified within a real world deployment. The concept and its implementation were jointly derived by A. Tan, D. Grewe, M. Wagner and O. Parzhuber in the course of A. Tan's Bachelor's thesis:

> A. Tan: "Prototype Implementation of a Wireless Vehicular Information-Centric Testbed," Bachelor's thesis, University of Applied Science Munich, Germany, Feb. 2018.

The thesis was supervised by D. Grewe and M. Wagner. The results of the thesis are published in the proceedings of the European Conference of Networks and Communication, June 2018. The paper was authored by D. Grewe and co-authored by A. Tan, M. Wagner, S. Schildt and H. Frey. The concept and its implementation is used in Section 5.8.

Caching-as-a-Service in Virtualized Caches for Information-Centric Connected Vehicle Environments

In today's networks, intermittent connectivity caused by sparse network deployments and the movement of vehicles as well as the end-to-end (host-centric) communication model challenge efficient data dissemination in connected vehicle environments. The loosely coupled communication model as well as the in-network caching capabilities of ICN are promising to overcome the challenges of future connected vehicle environments. In ICNs, mobile nodes are able to store and carry data items into areas not covered by the communication network. This paper proposes the concept of *virtual cache areas* in which nodes can carry and exchange cached data items on demand.

The results of the paper are published in the proceedings of the IEEE Vehicular Networking Conference, December 2018. The paper was authored by D. Grewe and co-authored by M. Wagner, H. Frey, S. Schildt, and M. Arumaithurai (University of Goettingen). The content of the paper is used in Section 6.

Resolutions Strategies for Networking the IoT at the Edge via Named Functions

Named Function Networking is an extension for ICN to support the execution of in-network computations. While the networking paradigm was originally designed for computer centers for big data processing, it evolves to be deployed toward edge computing scenarios including connected vehicle environments. In this paper, resolution strategies to execute named functions are investigated and novel approaches for the constrained IoT such as small sensors and connected vehicles are proposed.

The results of the paper are published in the workshop on Edge Computing in the Proceedings of the IEEE Consumer Communications & Networking Conference, January 2018. The paper was authored by C. Scherb, D. Grewe and co-authored by M. Wagner and C. Tschudin (University of Basel). The concept and its results are used in Section 7.

A Network Stack for Computation-Centric Vehicular Networking

Based on the Named Function Networking principles, a network stack for the data exchange in the automotive IoT is presented as part of a prototype implementation. While the networking paradigm was originally designed for computer centers for big data processing, it evolves to be deployed toward edge computing scenarios including connected vehicle environments. In this paper, a prototype implementation is presented using real world experiments on a test course.

The results of have been presented as part of a demonstration at the ACM Conference of Information-Centric Networking in Boston, U.S.A., September 2018. The implementation was done in conjunction with D. Grewe, C. Marxer, C. Scherb (University of Basel). The demonstration abstract was authored by D. Grewe and co-authored by C. Marxer, C. Scherb, M. Wagner and C. Tschudin (University of Basel). The concept an its results are used in Section 7.

EnCIRCLE: Encryption-based Access Control for Information-Centric Connected Vehicles

This work introduces an encryption-based access control mechanism for information-centric connected vehicles. It provides access control and other security features in ICN-based IoT systems and is applied to a vehicular system, which is especially challenging due to mobile and intermittently connected network participants. The concept and its results were jointly derived by P. Rao, D. Grewe and M. Wagner in the course of P. Rao's Master's thesis:

> P. Rao: "Security Evaluation and Concept Definition for an ICN-based Data Market Place for Connected Vehicles," Master's thesis, University of Applied Science Esslingen, Germany, Feb. 2017.

The thesis was supervised by D. Grewe and M. Wagner. The results of the thesis are published in the proceedings of the International Conference Network of the Future, November 2017. The paper was authored by D. Grewe and co-authored by P. Rao, M. Wagner, S. Schildt, D. Schoop, and H. Frey. The concept and its results are used in Section 8.2.

Open Security Issues for Edge Named Function Environments

This work introduces open security issues for in-network function execution at the edge of an information-centric network. Based on a use case from automotive IoT, security challenges are identified and different options to tackle these challenges are discussed in detail. The challenges and open issues were jointly derived by M. Krol, C. Marxer and D. Grewe as part of a GI Dagstuhl seminar and in conjunction with I. Psaras and C. Tschudin. The use case and its challenges regarding security issues in named function environments are authored by D. Grewe.

The work is published in the special issue of “Information-Centric Network Security” of the IEEE Communciations Magazine, November 2018. The open issues and derived challenges are used in Section 8.3.

List of Publications of the Author

11. D. Grewe, M. Wagner, S. Schildt, M. Arumaithurai, H. Frey, "Caching-as-a-Service in Virtualized Caches for Information-Centric Connected Vehicle Environments", in Proceedings of the IEEE Vehicular Networking Conference (VNC), 2018, Taipei, Taiwan

10. D. Grewe, C. Marxer, C. Scherb, M. Wagner, C. Tschudin, “A Network Stack for Computation-Centric Vehicular Networking“, in Proceedings of the ACM Conference on Information-Centric Networking, 2018, Boston, MA, U.S.A.

9. M. Król, C. Marxer, D. Grewe, I. Psaras, C. F. Tschudin, “Open Security Issues for Edge Named Function Environments“, in IEEE Communications Magazine – SI: Information-Centric Networking Security, 2018

8. D. Grewe, A. Tan, M. Wagner, S. Schildt, H. Frey, “A Real World Information-Centric Connected Vehicle Testbed supporting ETSI ITS-G5”, in Proceedings of the European Conference on Networks and Communications (EuCNC), 2018, Ljubljana, Slovenia

7. D. Grewe, M. Wagner, S. Schildt, H. Frey, “ADePt: Adaptive Distributed Content Prefetching for Information-Centric Connected Vehicles“, in Proceedings of the IEEE Vehicular Networking Conference (VTC-Spring), 2018, Porto, Portugal

6. D. Grewe, M. Wagner, H. Frey, “A Domain-specific Comparison of Information-Centric Networking Architectures for Connected Vehicles”, in IEEE Communications Survey and Tutorials, 2018

5. C. Scherb, D. Grewe, M. Wagner, C. F. Tschudin, “Resolution Strategies for Networking the IoT at the Edge via Named Functions“, in Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC) Workshop: Edge Computing, 2018, Las Vegas, U.S.A

4. D. Grewe, P.K.P. Rao, S. Schildt, M. Wagner, D. Schoop, H. Frey, “EnCIRCLE: Encryption-based Access Control for Information-Centric Connected Vehicles”, in Proceedings of the IEEE Conference Network of the Future (NOF), 2017, London, UK

3. D. Grewe, M. Wagner, A. Arumaithurai, I. Psaras, D. Kutscher, “Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions”, in Proceedings of the ACM SIGCOMM Workshop on Mobile Edge Communications (MECOMM’2017), 2017, Los Angeles, CA, U.S.A

2. D. Grewe, M. Wagner, H. Frey, “ICN-based Open, Distributed Data Market Place for Connected Vehicles: Challenges and Research Directions”, in Proceeding of the IEEE International Conference on Communications (ICC): WS06-Convergent Internet of Things. . . On the synergy of IoT systems, 2017, Paris, France

1. D. Grewe, M. Wagner, H. Frey, “PeRCeIVE: Proactive Caching for ICN-based VANETs”, in Proceeding of the IEEE Vehicular Networking Conference, 2016, Columbus, Ohio, U.S.A.

1.5 Outline

Chapter 2 introduces the fundamental concepts of connected vehicle environments by providing an overview of available architectures and standards around the globe. One of the main objectives of this chapter are the presentation of wireless automotive communication standards as well as an outlook of upcoming technologies and networking paradigms.

A general overview on data-oriented networking and caching principles is presented in Chapter 3. The first part of the chapter provides a high level overview of networking principles such as Content-Delivery, Delay-Tolerant, Information-Centric, and Computation-Centric Networks. The second part of the chapter presents a taxonomy of caching and prefetching techniques.

Chapter 4 provides a systematic and comprehensive survey of information-centric networking architectures based on the requirements derived from automotive use cases introduced in Chapter 1. The scope of the survey is limited to the features directly provided by existing ICN architectures, while modifications and extensions are not considered. Furthermore, a state-of-the-art analysis of caching techniques in ICN in the context of mobile systems is presented.

Chapter 5 proposes ICN as a potential solution to increase the availability of data by using *proactive caching* techniques. Based on the introduction of the mobile node delivery problem, three novel proactive caching strategies are presented and evaluated using simulations based on a real world network deployment. Furthermore, a network stack supporting ICN for information exchange is presented as part of a prototype implementation using real hardware.

By using mobile nodes as data carriers into areas not covered by infrastructure node deployment, models to assess the potential of in-network caching in such environments are presented in Chapter 6. The concept of virtual cache areas in information-centric connected vehicle environment is presented and evaluated by using simulations based on a real world network deployment.

Instead of serving static data, the computation-centric networking paradigm provides access to dynamic computed results. Chapter 7 provides a discussion and elaboration of enhancing ICN towards the effective provisioning of data using principles from the computation-centric networking domain. First, the limitations of the Named-Function Networking approach in vehicular networks are presented, followed by novel resolution strategies to access cached computation results. These theoretical results have been elaborated as part of a prototype implementation using real hardware.

When storing content actively in the network, security implications have to be considered. Chapter 8 provides an analysis of security challenges with respect to cached objects and introduces a novel access control mechanisms for information-centric connected vehicles. The chapter ends with a discussion on security challenges for in-network function execution in computation-centric vehicular networks.

Finally, Chapter 9 concludes the thesis. It provides a discussion to which extent the introduced research questions are answered by the presented contributions. The chapter ends by providing an outlook on future perspectives and research directions.

2 Connected Vehicle Environments

> Today, progress happens so fast that even as one person declares that something is totally impossible, another interrupts them to say that he has already done it.
>
> Albert Einstein

Connected vehicle systems will heavily rely on on-time information exchange. One example is the introduction of connected automated driving demanding information such as real-time maps or street conditions. Communication systems will play an important role to realize connected vehicle environments, to fulfill the challenges and requirements of future applications and use cases. This chapter will provide an overview of networking architectures for connected vehicle environments with a special focus on wireless communication technologies and standards.

2.1 Cooperative Intelligent Transportation Systems

Equipping mobile systems (e.g., vehicles) with communication units enables new opportunities for transportation systems in general. Intelligent Transportation Systems (ITS) aim to harmonize the operation of vehicles, for example, a collision avoidance system by notifying drivers about a hazardous situation ahead. In the past decade, ITS evolved towards Cooperative Intelligent Transportation Systems (C-ITS). It describes a new generation of intelligent transportation systems, using communication units to operate between each other and other C-ITS systems in a distributed fashion. Figure 2.1 illustrates an intelligent transportation system. Vehicles such as buses, cars, or trains exchange information with an infrastructure managing the traffic flow or operating the balance of parking lots at a bus terminal.

Participants in such transportation systems are characterized by mobility. In such environments, wireless communication systems are essential to consume, process and share information, for example to manage traffic flows, avoid hazardous situations, assist drivers with additional information as well as provide comfort-driven applications for passengers during the journey. The main objectives of C-ITS are improving safety, efficiency and comfort for passengers and the surrounding vicinity [23].

Use cases within C-ITS can be separated into (i) *safety-driven* and (ii) *comfort-driven* applications [6]. Examples for safety-driven applications are hazard and collision warnings – notifying the driver about any hazardous or dangerous situations on the road ahead. Examples for comfort-driven applications are traffic management and cooperative navigation – optimizing traffic flows or avoiding traffic jams, as well as infotainment applications – entertaining passengers during the journey.

2.2 Architecture Principles in Connected Vehicle Environments

The operational picture of the automobile has evolved with the rapid development of technological progress as well as the consumption behavior of travelers. The demand of new services is increasing with the technological advancements in the fields of micro-controllers and mobile

Figure 2.1: Illustration of an intelligent transportation system. Vehicles are able to exchange information with each other and an infrastructure in order to harmonize the operation of vehicles (source: Bosch Mediaspace [24]).

communication. An example of the technical evolution in the automotive domain are navigation systems. First available as built-in devices, followed by a decade as additional external devices, they are now available again as built-in services in today's in-vehicle infotainment system while consuming up to date information from cloud services.

While it is to be expected, that automotive services will vary in their use cases, the equipment, such as sensors used or the communication technologies available, there are common denominators:

- **Mobility**: Participants of such networked system are characterized by a high degree of mobility freely join and leave the network during the journey.
- **Data Dissemination**: Dissemination of data is obtained across different network participants (e.g., infrastructure components) using several wireless communication technologies simultaneously.

Academic and industrial activities have shown a trend towards *heterogeneous vehicular networks* in networking connected vehicles. In such network systems, nodes are able to communicate with each other by choosing from a variety of wireless networking technologies such as cellular or Dedicated Short Range Communication (DSRC). As a result, Vehicular ad-hoc Networking (VANET) enables new communication relations between a communication enabled vehicle and other network participants. V2X incorporates numerous types of communication categories. Figure 2.2 illustrates the different vehicular communication categories exemplary. For example, information exchange between a vehicle and other vehicles, cloud infrastructures or other nodes in the vicinity. These relations can be categorized into the following subdomains:

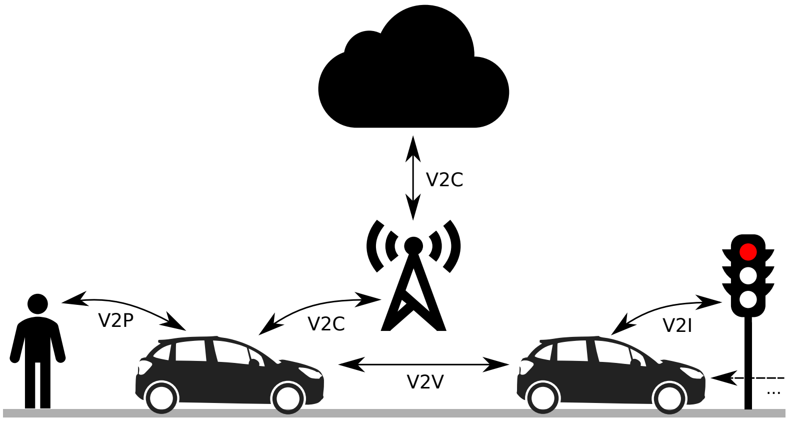


Figure 2.2: Exemplary illustration of the different vehicular communication categories, namely vehicle-to-cloud (V2C), vehicle-to-infrastructure (V2I), and vehicle-to-vehicle (V2V) communication.

- **Vehicle-To-Vehicle (V2V)**: Vehicular applications exchange information between other vehicles in close vicinity. It includes safety-driven use cases such as collision warnings, or optimization-driven use cases such as traffic flow optimization or platooning.

- **Vehicle-To-Infrastructure (V2I)**: Information is exchanged between vehicle and roadway infrastructure components in the vicinity. It includes sensors providing information such as the current traffic situation, accidents or hazardous situations, or the road conditions ahead. Such information is used for safety purposes as well as to optimize the traffic flow (e.g., in urban areas) or to increase fuel economy.

- **Vehicle-To-Cloud (V2C)**: Applications communicate with in-vehicle components and services hosted in a cloud backend infrastructure. Examples for such cloud infrastructures can be public available cloud service providers (e.g., Amazon AWS [25] or Microsoft Azure [26]) or dedicated on-premises solutions at the application providers such as the car manufacturer (e.g., BMW) or third-party service providers (e.g., Google or Bosch).

- **Vehicle-To-Pedestrian (V2P)**: To increase the safety by preventing situations in which pedestrians get seriously injured, information is exchanged between vehicles and a broad set of road users including walking people, cyclists or passengers embarking and disembarking buses.

For each of the introduced communication sub-domains, different network access technologies and standards have evolved in the past decades. Two popular technologies include cellular and IEEE 802.11 (also known as: WLAN) communication standards, which will be introduced and described in detail in the following subsections.

2.3 Wireless Communication Technologies in Connected Vehicle Environments

Over the past decades, different wireless access technologies have been proposed and investigated towards C-ITS providing enhanced connectivity for vehicles. As a result, two wireless communication technologies are considered for interconnecting vehicular nodes with each other: *cellular-based* and *WLAN-based* communication technologies.

2.3.1 Cellular communication

Cellular wireless communication technologies are well established in the market. First introduced to connect mobile user devices and to share voice information, cellular networks have evolved towards efficient data sharing from generation to generation. Since its third generation (3G), cellular networks are also considered for vehicular communication [27].

Universal Mobile Telecommunication System (3G)

In the past decades, standardization of cellular telecommunications technologies were driven by the 3rd Generation Partnership Project (3GPP). In 1998, several telecommunication standards organizations around the globe started to collaborate together as part of the partnership project to work towards a global standard of cellular telecommunication technologies [28]. In 1999, the project released the first document of the third generation called Universal Mobile Telecommunication System (UMTS). The main goals of the 3GPP for the third generation were to optimize mobile networks towards broadband services and to improve the network connectivity. An important development in 3G was the implementation of Wideband Code Division Multiple Access (WCDMA) channel access method to increase the available bandwidth. In WCDMA, both methods Frequency Division Duplex (FDD) and Time Division Duplex (TDD) variants are supported. Table 2.1 illustrates the characteristics of UMTS such as the frequency bands and channel width used.

Since the release 5 and 6 [29] and the introduction of high-speed (uplink/downlink) packet access (HSxPA), cellular technologies are of interest for data communication in vehicular networks. While the development of UMTS and the enhancements such as HSxPA had a large impact on consuming information on mobile user device, 3G was not used for vehicular networks. One of the major requirements such as low latency communication were not possible in 3G networks.

Long Term Evolution (4G)

Long Term Evolution (LTE), and more specific Long Term Evolution Advanced (LTE-A), defines the fourth generation of cellular telecommunications standards. First parts of the new generation were released in 2006 and 2008 in Release 8/9 [29]. Due to the success of the 3rd generation for human mobile users, the main objectives of LTE was to deliver more capacity for faster mobile broadband experience and as an enabler of cellular technologies to new frontiers. The main development in LTE and the enhancement LTE-A were the introduction of a new channel access method (OFDMA) including wider channels (up to 20MHz), the simultaneous support of multiple antennas, a simplified core network, as well as a *reduction of latency times* for both user and control plane. Especially the improvements regarding lower latency values, led to the fact that LTE and the enhancement LTE-A were considered for vehicular networking (e.g., [30]).

Table 2.1: Cellular-based communication technologies and their characteristics [31, 32].

Feature	UMTS	LTE-(A)	LTE-V
Frequency band(s)	1.8-2.02GHz, 2.11-2.2 GHz	700/800MHz, 2.1,2.3,2.6,3.5GHz	5.9GHz (planned)
Channel width	5MHz	1.4, 5, 10, 15, 20MHz	10MHz
Bit rate	384Kbit/s - 42Mbit/s	50Mbit/s - 1Gbit/s	1.15 - 17.71Mbit/s
Radio resources	WCDMA	OFDM	SC-FDMA
Release	1999	2006/2008	2016

Long Term Evolution Vehicle (LTE-V): In release 14 [29], the 3GPP published the first standard of cellular assisted vehicular networking to support V2X communication for upcoming vehicular applications. The standard introduces a new network interface (called PC5) addressing high node velocity and high node density. Additionally, two new communication modes are introduced for V2X (cf. [31]) and V2V (cf. [32]) communication: (i) *mode 3* – infrastructure assisted mode to manage the resources for V2X –, and (ii) *mode 4* – an ad-hoc mode for V2V to operate in cellular uncovered areas. Table 2.1 illustrates the cellular telecommunication standards and their characteristics. LTE-V is in direct competition to the WLAN-based communication technologies such as the IEEE 802.11p access technology (e.g., [33]).

2.3.2 WLAN-based communication technology

Dedicated Short Range Communication (DSRC) describes a class of communication variants based on the IEEE 802.11 (also known as wireless local area network - WLAN) specification in unlicensed spectrum. The overall WLAN standard itself describes a superset of several amendments, established by the IEEE. More specific, IEEE 802.11 is separated into several subsets, defining how wireless devices can interconnect with each other.

Regarding DSRC, it need to be distinguished between several variants available around the globe [6, 34]:

- North America: an ITS communication stack is available, standardized by the IEEE as IEEE 1609 protocol suite - also known as the WAVE protocol suite (cf. 2.4.1).
- Europe: an ITS communication stack is available, standardized by the European Telecommunications Standardization Institute (ETSI) as ETSI ITS-G5 protocol stack of the 5th generation (cf. 2.4.2).
- Japan: an ITS communication stack is available, standardized by the Association of Radio Industries and Businesses (ARIB)

All DSRC variants have one thing in common: they all use the same physical layer – IEEE 802.11p [35]. The basis for this technology is created in the IEEE 802.11a [36] standard - well known from the WLAN used at home. However, the amendment has been modified to meet the requirements of vehicular networks such as the extremely short latency requirements for road safety messaging and control. Table 2.2 provides an overview of available vehicular WLAN-based standards. While the standards in North America and Europe are based on the IEEE 802.11p OCB mode (OCB = Out of the Context of a BSS) access layer operating in

Table 2.2: WLAN-based communication technologies and their characteristics [35, 37].

Feature	North America & Europe	Japan
Physical Layer	IEEE 802.11p	IEEE 802.11p
MAC layer	IEEE 802.11p OCB	ARIB STD-T109
Frequency band(s)	≈ 5.9GHz	≈ 700MHz
Channel width	10MHz	9MHz
Bit rate	6Mb/s	5 - 10Mb/s
Radio resources	CSMA/CA	CSMA/CA ǀ TDMA
Release	2010	2012

the 5.9 GHz frequency band, Japan standardized the ARIB STD-T109 access layer [37] – an adapted layer in the 700 MHz frequency band which uses both carrier-sense (CSMA/CA) and time-slotted (TDMA) access in parallel. For example, vehicles supporting the standard are able to communicate ad-hoc between each other, without the need of an infrastructure component as management unit such as a base station.

In summary, there are different wireless access and physical layer technologies available to enable inter-vehicular information exchange. These access technologies are used by higher layers, e.g., network protocol stacks to realize the actual information exchange.

2.4 Network Protocol Stacks for Connected Vehicles

As a part of a standardized network architecture for C-ITS, communication protocol stacks are required to actually exchange information across different nodes in an C-ITS. The most relevant WLAN-based protocol stacks used in North America – the IEEE 1609 protocol suite, and the European stack European Telecommunications Standards Institute (ETSI) ITS-G5 are presented in the subsequent sections.

2.4.1 IEEE 1609 - Wireless Access for Vehicular Environments (WAVE) Protocol Suite

In North America, IEEE 1609 protocol suite describes the de facto standard for V2X communication. It is also known as the Wireless Access for Vehicular Environments (WAVE) protocol suite [38]. WAVE is based on the IEEE 802.11p OCB (cf. 2.3.2) access layer technology and consists of a set of standards for secure message and service information exchange in vehicular communication systems. A list of the most important parts of the protocol suite includes:

- IEEE 1609.2: standard for secure information exchange between services and applications.
- IEEE 1609.3: standard for networking services including the WAVE Service Advertisement (WSA) and WAVE Short Message Protocol (WSMP) specifically designed for V2X communications.
- IEEE 1609.4: standard for multi-channel operation within the WAVE suite.
- IEEE 1609.11: standard for electronic payment via over-the-air for ITS.

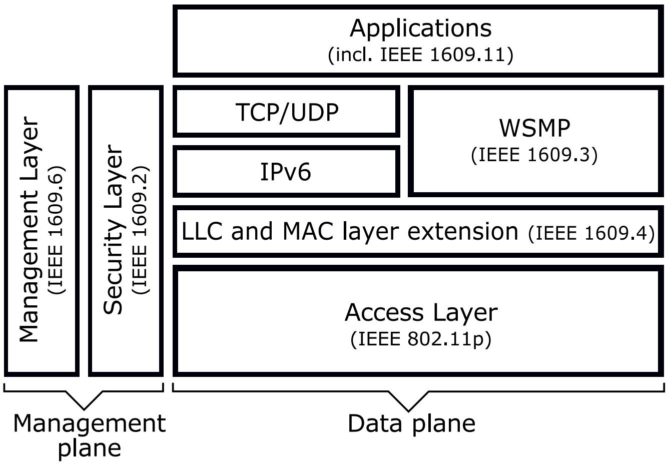


Figure 2.3: Illustration of the IEEE 1609 protocol suite for wireless access for vehicular environments (WAVE) based on [38].

Figure 2.3 illustrates the structure of the IEEE 1609 protocol suite. It is separated into two planes: (i) the management plane - providing mechanisms to manage and secure the information exchange, and (ii) the data plane - providing mechanisms to exchange messages between the vehicular applications. As part of the data plane, information exchange is realized using communication channels. WAVE standardizes two types of channels: a *control channel* (CCH) - dedicated for short, high-priority, data and management messages using broadcast communication, and a *service channel* (SCH) - for application specific information exchange using backchannel communication. While CCH messages only allow transmissions based on the WSMP protocol, SCH messages can be transmitted using both IP-based communication (here: IPv6 standard) as well as WSMP [38].

2.4.2 ETSI Intelligent Transportation System Generation 5

Besides the standardization of an ITS architecture for Europe, the ETSI also standardized a protocol stack for the 5th generation of intelligent transportation systems. The protocol stack defines four horizontal layers – required to support ITS applications and the underlying features such as the access technology or the communication protocol used, and two vertical layers – one for managing the stack instance and the other providing security features. Figure 2.4 illustrates the structure of the ETSI ITS-G5 station protocol stack [39].

- **ITS access layer**: provides various number of medium access technologies for the physical and data link layers of the stack. For example, it includes wireless communication technologies such as WiFi, cellular, Bluetooth, Global Positioning Service (GPS), but is extensible to the needs within an operated ITS.

- **ITS networking layer**: provides network and transport protocols to be able to exchange information across other ITS stations, other network nodes and the core network (e.g., the Internet). The layer is separated in two parts: transport specific protocols such as the Internet Protocol (IP) (version 4 and 6) or the GeoNetworking protocol and related higher layer transport protocols such as the Transmission Control Protocol (TCP) or Cooperative Awareness Messages (CAM) respectively.

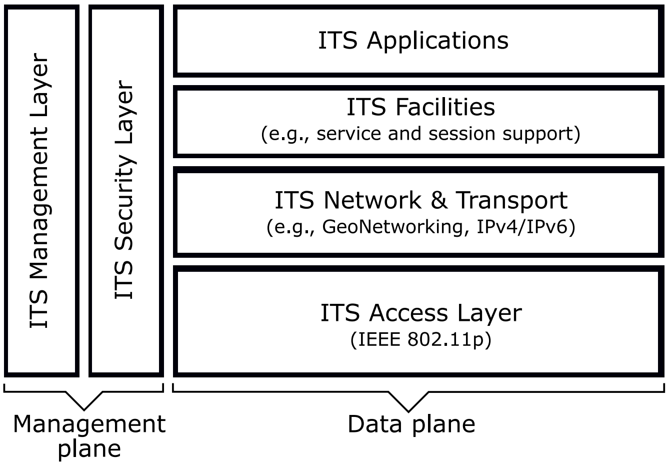


Figure 2.4: Illustration of the station protocol stack and its layers for the 5th generation of intelligent transportation systems of the European Telecommunications Standards Institute based on [39].

- **ITS facilities layer**: provides common functions to support and assist ITS applications. For example, it includes functions for session and message handling, management of services running on the ITS node, as well as data structures to store, aggregate or maintain data.
- **ITS applications layer**: provides functions to execute safety-critical and comfort-driven ITS applications.
- **ITS management layer**: provides functions to configure and manage an instance of an ITS station node. It also includes the functions to exchange information across other layers in the protocol stack.
- **ITS security layer**: provide security features such as identity management, privacy features, as well as security features such as tamper-proof hardware support, firewalls, etc.

Similar to the WAVE protocol suite, the ITS-G5 stack provides control and service channels for the exchange of information. As part of the stack, one CCH is used for management of high-priority message exchange, while seven SCH are offered by the stack to exchange application specific information [40]. The stack specified by ETSI is designed to be deployed on every ITS node including in-vehicle and infrastructure nodes.

2.5 Upcoming Technologies for Connected Vehicle Environments

In the past decade, several new communication technologies and networking paradigm have been announced and are under development. The following subsection briefly introduce the emerging technologies which are of interest for upcoming vehicular networks.

2.5.1 Next Generation Wireless Communication Standards

While LTE-V (cf. Section 2.3.1) has been introduced recently, the development of the next generation of cellular telecommunication standards is already being pursued by the standardization organizations. Release 16, which has been published recently, includes first parts of the upcoming 5th generation (5G) of cellular mobile communication systems [29]. However, it is still under active development. Key aspects of the next generation are (i) maximized bandwidth - allows for higher data rates, (ii) ultra low latency - enables fast exchange of time-sensitive data in wireless networks, as well as (iii) massive machine-to-machine communication - supports the integration of a tremendous number of machines into the cellular picture (e.g., [41]). These key aspects, especially the low latency support, are also crucial for upcoming connected vehicles, e.g., for automated driving. In 2016, several industry partners founded the 5G Automotive Association (5GAA) to bring 5G solutions to future mobility and transportation service [42].

Regarding WLAN-based systems, the European Electronic Communications Committee (ECC) analyzed the options of enhancing the frequency band used for future ITS in 2009. According to the report ECC (09)01 [43], the committee proposes to use the 63-64GHz frequency band as an amendment to the already reserved 5.9GHz band in European countries.

Cloudification of Vehicle Environments

The rapid development in computing resources as well as software frameworks and platforms has created new opportunities in servicing computational resources. Cloud Computing (CC) defines a paradigm in which all available resources of a data center, including hardware and software, are offered as a service to host, operate and manage Internet services (cf. [44]). For example, service providers which use a cloud infrastructure are able to pay for the use of hardware and software resources on demand, instead of deploying and maintaining their own infrastructure.

This development has also an impact for services in vehicle environments. For example, CC is used to receive information from the cloud such as firmware or software updates for in-vehicle components, or to receive on demand traffic update information. Furthermore, the cloud offers the option to offload computation intensive operations to the cloud for which in-vehicle hardware is not capable of computing the result in time (e.g., as stated in the electronic horizon example in Chapter 1.2.1).

The introduction of CC has had a huge impact for Internet services and still contributes to services which speed up the interconnection of devices such as vehicle environments. However, the geographical distance between consumers and the cloud infrastructure describes a challenge with respect to in-time data delivery. Information is transmitted through a public wide area network (e.g., the Internet) which may result in transmission delays or losses of packets during the delivery.

Edge Computing (EC): In the past years, research activities in academia and industry are investigating the EC paradigm to integrate highly mobile and dynamic networks. Instead of connecting to a cloud backend, EC brings system resources such as computational or storage, to the edge of the network (e.g., network access points such as cellular base stations) and therefore closer to the consumers (cf. [44, 45]).

Vehicular Cloud Computing (VCC): Cloud computing is also influencing vehicular networks. With the increasing demand of equipment within vehicles, e.g., adding sensors for driver assistance services, the available in-vehicle resources, including storage, computing

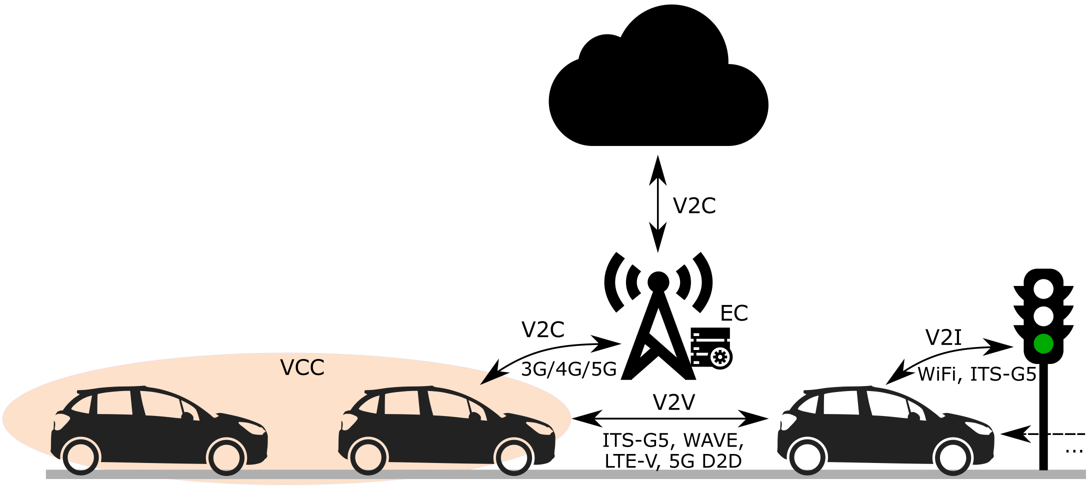


Figure 2.5: Illustration of the different vehicular communication relations, including the presented and upcoming wireless communication technologies.

power or sensing capabilities, increased as well. Vehicular cloud computing (VCC) (cf. [46]) describes a specialization of the cloud computing paradigm in which vehicles have the ability to define a mobile cloud, offering access to their available computing and sensing resources and information in order to fulfill a service or task (cf. Figure 2.5). Instead of down- or uploading information to a Internet cloud infrastructure, vehicular clouds applications participate in the resources and information in the vicinity (e.g., local relevant information such as traffic flows) and therefore experience efficient information delivery in a timely manner [46, 47].

2.6 Summary

This chapter outlined the topic of connected vehicle environments. By introducing Intelligent Transport Systems and its cooperative enhancement, different V2X communication categories in vehicular networks such as V2I or V2V, existing architectures and available communication standards around the globe were presented. Figure 2.5 illustrates the relations and their corresponding as well as proposed communication standards. At present, cellular and WiFi technologies are competing to capture the field of connected vehicles and try to establish themselves as de facto standard. Besides the telecommunication standards, other technologies such as the 5th generation of cellular standards as well as paradigms from the cloud computing domain are pushing into the market of vehicular networks.

While the wireless telecommunication standards have focused on the special mobility requirements of vehicular networks, the networking concepts of the inter-connection technologies, i.e., , receive data from the Internet, are still based on host-centric principles. However, such communication model challenges the network at multiple levels. For example, address updates need to be maintained frequently while mobile nodes switch multiple times between network access points, and therefore, frequent changes of the data delivery routes.

To unleash the potential of inter-connecting vehicular systems and its surroundings (e.g., get the latest software update for a highway pilot assistant efficiently during the journey), principles from the data-oriented networks are promising to overcome the presented challenges of host-oriented networks.

3 Data-Oriented Networking and Caching Principles

> We can't solve problems by using the same kind of thinking we used when we created them.
>
> Albert Einstein

The engineering principles and architectures of today's Internet were created in the 1960s and '70s. Back then, the Internet consisted of just a few machines and was designed to trigger remote services on a specific device. Applications those were built around such host-to-host model (host-centric), e.g., to login or transfer files remotely from one machine to another.

Today, consumers value the Internet for its content. But the communication model is still based on *where* the content is located. The host-centric model results in a conversation between exactly two machines, one wishing to use the resource and one providing access to it. The problem: the same data is transferred multiple times across the same route for each consumer which results in higher latency and bandwidth consumption.

This becomes more problematic when looking into the trend of networking everyday objects such as vehicles and home automation environments in the IoT. Formerly deployed as specific solutions for each domain, the development is continuing to link different domains together to form a large *heterogeneous* IoT ecosystem. This development raises challenges in different fields such as scalability of billions of devices, interoperability across different IoT domains and the need for mobility support.

In the past decades, research activities were concerned about tackling the needs of the consumers by "*re-designing*" today's networks towards a data-oriented fashion. The following chapter introduces data-oriented networking paradigms and illustrates their core concepts. Furthermore, a survey on caching architectures is provided to introduce the subject of storing data closer to consumers.

3.1 Data-Oriented Networking Principles

One of the popular methods regarding data-oriented network architectures is the separation of content from one specific physical node hosting the data by changing or modifying the addressing scheme. In the past years, different network approaches have evolved which can be classified into two categories: (i) evolutionary approaches – building up on top of existing network architectures and technologies as an *overlay*[5] solution, and (ii) revolutionary approaches – replacing the existing network architecture completely. The following sections introduce approaches from both categories and highlights the strengths and weaknesses of each network paradigm.

[5] The term "overlay" is defined as follows based on [48]: "An overlay network is build on top of existing network technologies (e.g., the Internet Protocol), offering a mix of various features such as efficient search of data items or data storage capabilities."

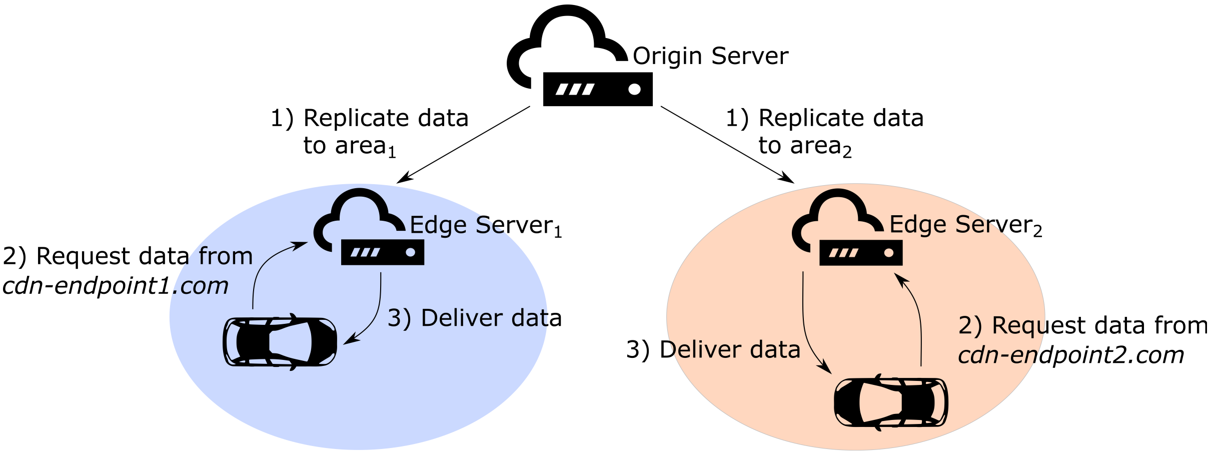

Origin Server
1) Replicate data to area$_1$
1) Replicate data to area$_2$
Edge Server$_1$
Edge Server$_2$
2) Request data from *cdn-endpoint1.com*
3) Deliver data
3) Deliver data
2) Request data from *cdn-endpoint2.com*

Figure 3.1: Content Delivery Networks: Exemplary structure of deployed CDN making use of servers at the network edge (see EC paradigm in Section 2.5) to improve delivery times of data and reducing load at the origin server by replicating data at several surrogate servers.

3.1.1 Content-Delivery Networks

First introduced in the late 1990, Content Delivery Networks (CDN) describe a network paradigm (overlay solution) for today's Internet by providing network resources and service in a distributed fashion to improve the network performance. In CDNs, the network has two major tasks to manage: (i) provide fast access to data/content by reducing the delivery times, and (ii) reduce the load of the original server in the core network [49]. Figure 3.1 illustrates an exemplary structure of a deployed CDN. To achieve these tasks, content offered by an original server is replicated and pushed, or pulled to powerful surrogate servers around the globe, and therefore closer to the consumers.

From a consumer perspective, applications have to be optimized to request information from the surrogate servers. In case data is not available, the server downloads the content from the origin server and stores it for subsequent requests. As a result, CDNs improve the network performance by increasing the availability and accessibility of data at several nodes closer to the consumer, reducing the load at the original server by serving content directly from CDN nodes, as well as maximizing the bandwidth [49, 50]. Examples for existing CDN infrastructures are Akamai Technologies [51], or Amazon CloudFront [52].

Improvements compared to conventional Internet deployment

CDNs distinguish between different types of data which can be separated into *static content* - data which will be delivered to all consumers in the same representation, and *dynamic content* - (personalized) data which will be created during the request. By identifying such kind of data, CDNs create replicas and store the content at valuable surrogate servers in the network [49].

Over the years, CDNs have been improved and optimized based on the behavior of the consumers. Routing and caching algorithms try to anticipate data items which may be requested soon. Based on the results of such algorithms, content is fetched from the original server beforehand and closer to the consumers (Figure 3.1, step 1). In case of a consumer request, data can be directly served by the surrogate server in a certain geo-area (Figure 3.1, step 2, 3). Such mechanisms reduce the delivery times and improve the service quality of Internet applications. One area which benefits from such mechanisms are e-commerce and streaming provider such as Amazon, Alibaba, or Google's Youtube. As an example, Google Global Cache describes a service to interconnect Google's network with internet service providers by installing dedicated Google servers within the provider's network [53].

Challenges of Content Distribution Networks

While the mechanisms of CDNs improve the delivery of information in today's Internet, the paradigm also possesses some weaknesses. Depending on the consuming audience, surrogate servers may be not close enough to the target location. In the worst case, the CDN server is further away than the original server (e.g., applications have not been optimized to use a CDN sufficiently), and therefore losing the benefit of shorter delivery times. Furthermore, most of today's CDNs are commercial-driven and cost additional money for the producer of data. Applications need to be CDN-enabled and optimized which represents a dependency on the CDN provider. Since CDNs are presented as an overlay solution on top of IP networks, all the additional introduced mechanisms of the underlying IP network increase the complexity and susceptibility to errors of the system and also effect the performance of CDNs. Examples for such additional mechanisms are MobileIP, or DynDNS.

3.1.2 Delay-Tolerant Networks

Delay-Tolerant Networks (DTN) define a research area and paradigm for heterogeneous networks which are characterized by a lack of continuous network connectivity. Especially, research activities in Mobile Ad-hoc Networks (MANET) such as wireless sensor networks or vehicular ad-hoc networks have been attracted by DTN while dealing with challenges such as frequent intermittent connectivity, link failures or limited network coverage of infrastructure nodes.

The main design principle of DTN is the change of the common end-to-end communication model towards a loosely coupled hop-by-hop Store-Carry-Forward architecture (SCF) model. It offers the possibility of using two addressing schemes: the traditional addressing of hosts (e.g., the requester) or addressing the content directly (e.g., the packet itself). Furthermore, the SCF model takes the responsibility in transferring the data until it reaches the destination. As a result, the model provides the opportunity to improve data delivery, since corrupted or dropped packets can be restored directly from the previous hop, instead of re-transmitting the data all the way back from the originator [54, 55]. Especially in networks which are characterized by a high degree of mobility, DTNs have shown improvements towards a reliable network (e.g., [56, 57]).

The Bundle Protocol

The most popular realization of a DTN is the Bundle Protocol (BP) [58]. The main objective of the protocol is the reliable transfer of data from one location to another within unstable communication networks. Corresponding data blocks are grouped together into *bundles* which is different from "classic" packet transmissions. This is due to the fact that only segments of data are received by the consumer node. When data is transmitted from one node to another, the protocol ensures that nodes always receive a complete bundle. The bundle itself is constructed in several blocks and requires at least one block containing the source and destination address. Besides, addressing a bundle directly is not excluded by the protocol design.

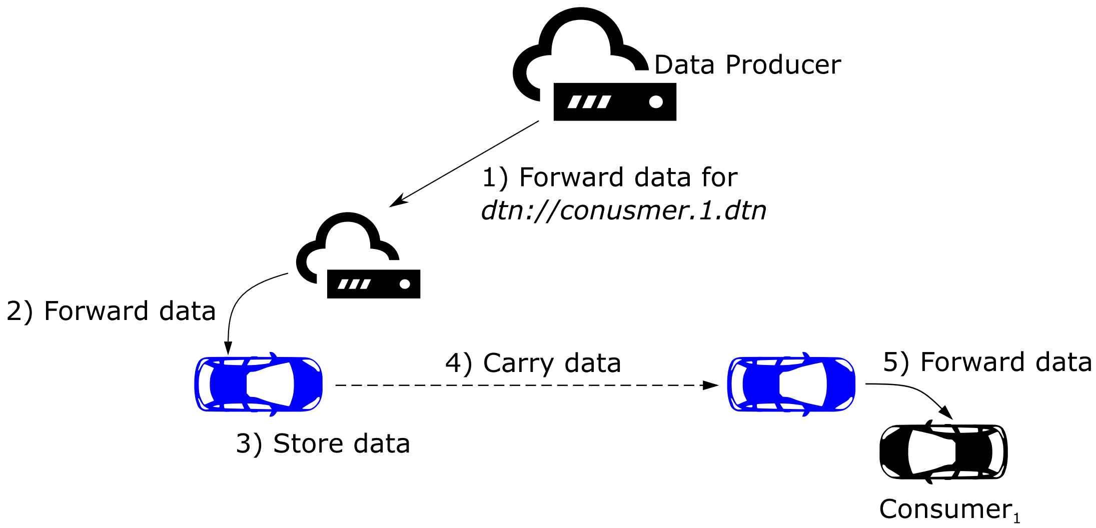


Figure 3.2: Delay Tolerant Networks: Exemplary structure of deployed DTN following the VCC paradigm. A mobile node (blue car) can store and carry data towards the consumer improve the data delivery in sparse network areas.

Moreover, the design of the BP is flexible and agnostic to the underlying transport layer. While addressing, routing and forwarding are defined as part of the BP, lower layer and transport specific requirements are realized by *convergence layers* [59].

The transmission of a bundle follows the SCF strategy. Bundles are stored and carried by a node until they can be forwarded to another node, closer to the destination. During the transmission, intermediate nodes are able to create and store a replica of the received bundle. While it is possible that multiple copies exist in the network, BP also offers a mechanism to ensure that the bundle stays alive in the network and is not dropped until it reaches the destination. Any intermediate node can take “custody” of a bundle, a special responsibility which ensures that at least one copy of data exists in the network.

Improvements compared to conventional Internet deployment

Compared to conventional Internet deployments, DTNs provide mechanisms to overcome the challenges in heterogeneous and sparse network deployment. The flexibility of the BP with respect to addressing nodes as well as bundles directly, represent just two positive aspects of DTNs. Furthermore, the focus on data, especially in avoiding the loss of bundles, by building up on the mechanism of SCF are some of the major differences compared to today’s Internet.

A communication example within a DTN is illustrated in Figure 3.2. Requested information is forwarded towards the consuming node as part of a bundle (cf. Figure 3.2, step 1). Every intermediate node forwards the data according to the SCF strategy. Meanwhile, every node is able to store a replica for subsequent requests. Dependent on the application, a node can take “custody” of the bundle, until it is released by another node. As part of the illustration in Figure 3.2, the blue vehicle stores and carries the data towards the consumer (cf. Figure 3.2, step 3 & 4). The mobility aspect of moving vehicles can be used in DTNs to bring data into areas uncovered by any infrastructure node. Finally, the requested data is delivered to the consumer (cf. Figure 3.2, step 5).

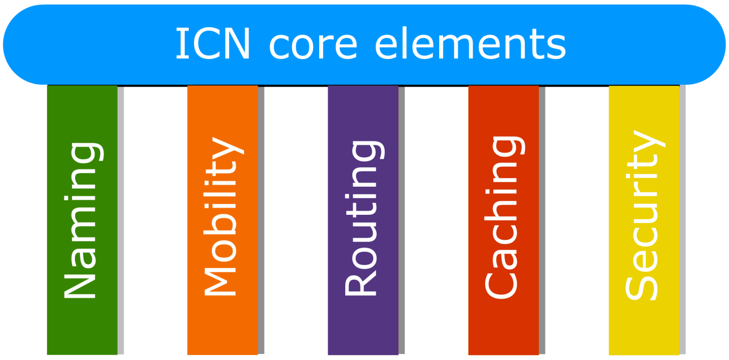


Figure 3.3: Information-Centric Networking core elements which are an integral part of each introduced ICN architecture.

Challenges of Delay Tolerant Networks

Especially in sparse networks, DTNs improve the delivery of data. However, there are also some challenges regarding this networking paradigm. For example, efficient routing and forwarding of data. Due to the fact of frequently occurring route changes or link losses, forwarding of data describes a challenging task. This becomes even more challenging when security features (e.g., establishing a trust model) are considered. As part of a DTN, nodes need to be trustworthy while receiving or forwarding data to other nodes in the network. Malicious nodes are able to stop the forwarding process, drop packets, or flood the network with wrong data.

3.1.3 Information-Centric Networks

Information-Centric Networking describes a paradigm of the Future Internet Architecture (FIA) research which puts data as the first class citizen in the network and separates content from its physical location. In ICNs, consumers are accessing data directly using *content identifiers* instead of *host identifiers*. This contrasts significantly to a host-centric communication model, where consumers need to first connect to the source of the data (e.g., using IP addresses) before accessing it. The structure of these content identifiers can be represented in a hierarchical (e.g., similar to Uniform Resource Identifier (URI) [60]), flat (e.g., hash values of the content), or a combined structure. As a result, ICN achieves a loosely-coupled communication model directly on the network layer, providing direct access to data.

First introduced in 2001 by Gritter et al. [61], naming schemes are used to access data in the network directly. This conceptual idea has been used by Jacobson et al. [12] and proposed as part of the *Networking Named Content* architecture. It describes a revolutionary network approach for the future Internet by replacing the classical host-centric communication model entirely.

In the last decade, several ICN architectures have been proposed using both evolutionary (e.g., as overlay on top of existing designs) or as revolutionary approaches (e.g., replacing existing designs entirely). In general, ICN architectures support five building blocks (cf. Figure 3.3) [62, 63]:

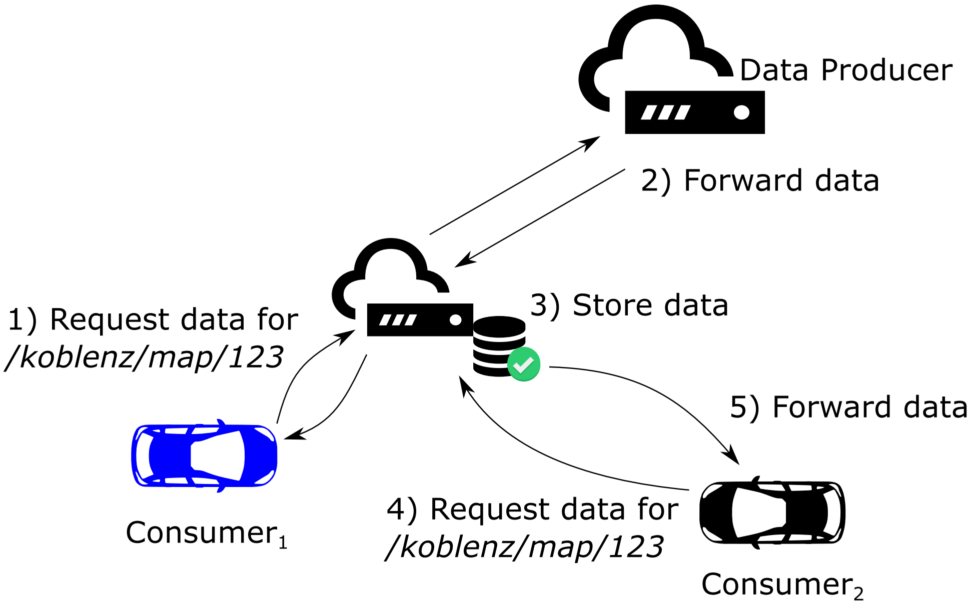


Figure 3.4: Information-Centric Networks: Illustration of an ICN deployment. A mobile node requests for data using a name, while intermediate nodes are able to cache the data for subsequent requests. As a result, the traffic in the core network is reduced.

- **Naming**: The access to data is provided by using content-identifiers such as naming schemes.
- **Mobility**: Data can be accessed directly from any node in the network supporting mobility by nature, while data is no longer affiliated to a single host.
- **Name-based Routing**: Hop-by-hop routing and forwarding decisions based on content identifiers instead of physical locations. However, Internet routing and forwarding algorithms such as link-state routing or geo-forwarding can also be performed by ICN.
- **In-Network Caching**: Network nodes provide in-network storage capabilities to balance and reduce traffic volumes in the core network by delivering valuable data from caches nearby the consumers.
- **Data-Centric Security**: The data-centric security design of ICNs is concerned about securing the payload of a packet instead of the communication channel.

Especially the loosely-coupled communication model which facilitates mobility as well as in-network caching and processing nominate ICN as a potential candidate for heterogeneous and mobile networks. Further details of ICNs are provided in Section 4.

Improvements compared to conventional Internet deployment

The major difference between ICNs and other network paradigms is the focus of providing and accessing data directly using *content-identifiers*. Figure 3.4 illustrates a communication example within an ICN. A consumer uses the name of the data to request it from the network (cf. Figure 3.4, step 1). Based on the name, the request is forwarded towards any node providing the information and delivered the reverse path back to the consumer (cf. Figure 3.4, step 2). Any intermediate node is able to store a replica of the data, for example for subsequent consumers (cf. Figure 3.4, step 3). Due to the loosely coupled communication model, another consumer interested in the same information (here: consumer$_2$), can easily access it directly from any cache in the vicinity which keeps a vital copy of the information, including neighboring nodes, or network entry points such as cellular base stations (cf. Figure 3.4, step 4 & 5).

As a result, ICNs have the ability to represent a network of distributed information, providing a simplified method of accessing the data by using naming schemes. Such mechanism has the potential to increase the delivery times of data as well as reducing the network traffic load in the networks.

Challenges of Information-Centric Networking

One of the major challenges is described by the *discovery* of names. In ICNs, information is accessed using the name, therefore the discovery of names is crucial for a successful information exchange. However, there is no common discovery strategy, due to the variance in existing architectural approaches.

As part of the research towards future Internet architectures, several ICN architectural approaches consider a clean-slate deployment to unleash the full potential of ICNs. Such approach will require the replacement of infrastructure components such as routers, since the hardware is still based on the host-centric communication model.

Another challenge is described by the data-centric security. By changing the addressing scheme in ICN, a secure end-to-end communication channel can not be established anymore. Especially, the combination of in-network caching and data-centric security describes a challenge difficult to achieve. For example, encrypted data is stored within caches, however, a consumer which joins the network recently is not able to access the content, while not already applying for decryption information.

3.1.4 Computation-Centric Networking

In the past years, advances in compute virtualization, activities towards data-oriented networks as well as the push of cloud computing technologies into distributed, heterogeneous networks have resulted in novel research directions of integrating computing technologies into networking. One example for such a development are *computation-centric networks* or *compute-first networks* (e.g., [64]). By making use of the loosely coupled communication model of ICNs, such network architectures provide access to dynamic computed results in the network instead of serving static content. This development is promising, especially, for scenarios demanding for computation-intensive and latency-sensitive information (e.g., as presented as part of the *community-based sensing* use case in Section 1.2.2).

Figure 3.5 illustrates the difference of the querying semantics between information-centric and computation-centric networking. Instead of requesting for data using the content-identifier in ICNs (Figure 3.5 flow 1), expressions are used to request for computation results (Figure 3.5 flow 2). These expressions are forwarded by the network to the most suitable execution node. Nested expressions are decomposed into subroutines and outsourced to other execution nodes if needed. If required, computation nodes request for data using the ICN principles.

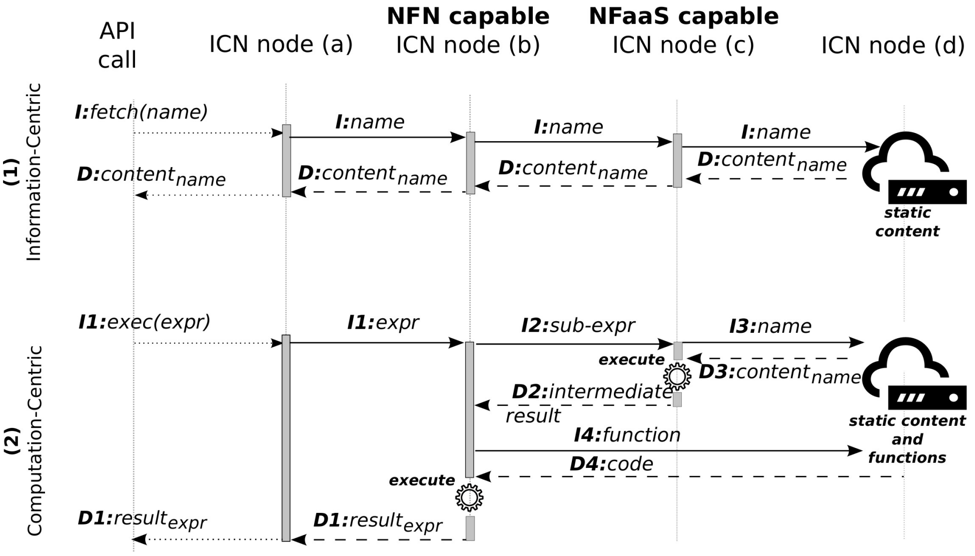


Figure 3.5: Computation-Centric Networks: The differences between information-centric and computation-centric networking from a requestor perspective. While consumers in ICN request for content, consumers in a computation-centric network request for computation results. The network architecture takes care of gathering required data, executing the computation and delivering the computation result to the requestor [22].

Finally, the computed result is encapsulated into an ICN Information Object and forwarded back to the consumer. Since the concepts are based on ICN principles, the in-network caching capabilities of ICN allow the network to store computation results, and therefore, increases their availability.

Two of those architectures are Named Function Networking (NFN) [65] and Named Function as a Service (NFaaS) [66]. Both approaches are extension of ICN.

Named Function Networking Similar to the exchange of information in ICN, NFN function code is stored and carried through the network as part of Information Objects. Such functions are executed at any element in the network providing a NFN runtime such as intermediate nodes, edge nodes or any end user devices able to run the computation. Requests and results in NFN are delivered as part of Information Objects and forwarded through the network following the principles of ICNs. This also implies that function results are cached in the network for later reuse. Additionally, NFN defines *workflow definition* for an optimal delivery of results. Such workflow definitions are created using λ expressions – a formal definition of a mathematical logic describing a Turing complete operation.

An example of a NFN expression is given as:

```
/example/object1 (λ x.call /lib/func x
        (call /lib2/func2 /example/object2))
```

In order to be processed by an ICN, the workflow starts with a name followed by the NFN related parameters. This includes first the expression of the function which has to be invoked (here: func), followed by the input parameters of the function. Such parameters can also be another workflow definition requesting for the result of the operation of "func2" and the parameter "/example/object2/".

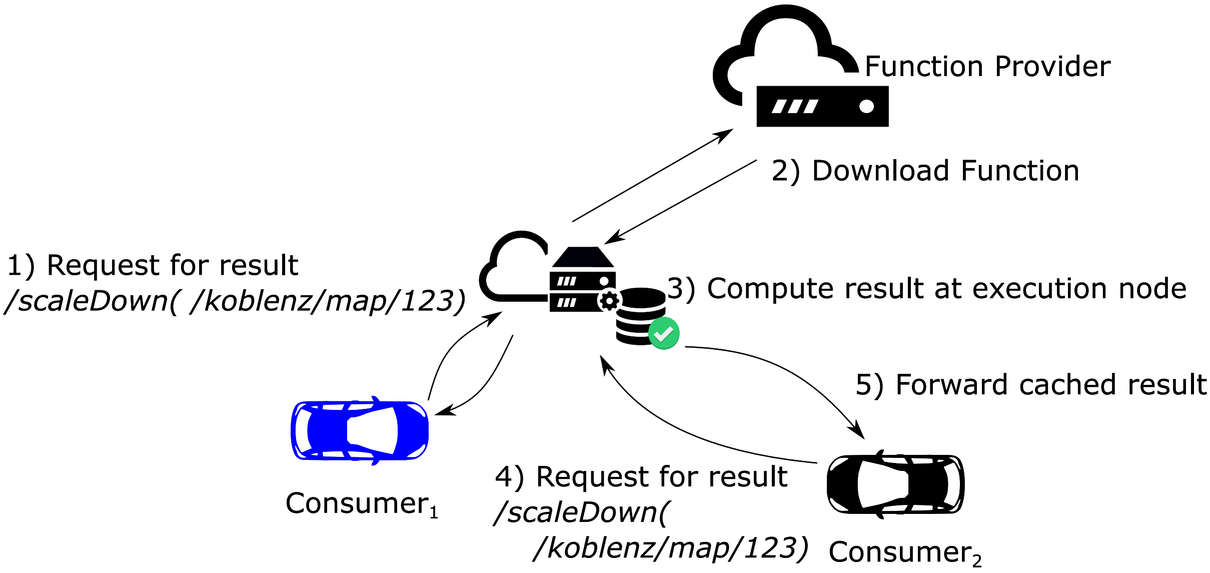


Figure 3.6: Computation-Centric Networks: A mobile node requests for a computation result performed at the edge of the network using a name, while intermediate nodes are able to cache the results for subsequent requests.

Named-Function-as-a-Service NFaaS is based on the concepts of *serverless computing* - an emerging technology which perform functions without requiring a dedicated backend infrastructure [67]. Similar to NFN, function code is stored and transfered as Information Objects in the network. To create and perform in-network functions, NFaaS uses unikernels. Unikernels include all system components required to execute a function [68]. The approach make use of hierarchical naming principles of ICN and pre-defines certain name prefixes to express the execution of functions or the access of results in the network.

An example of an execution expression in NFaaS is given as:

```
/exec/delay/func1/parameter1
```

NFaaS encodes all required execution information as part of a routable name. Additionally, NFaaS supports application-related service metrics which are also expressed as part of the name. In this example, a delay-sensitive application is considered, directly expressed as part of the name. In the provided example, the "exec" prefix indicates an execution request for a delay-sensitive function with the name "func1" followed by its parameters.

Improvements compared to conventional Internet deployment

One of the major improvements of computation-centric networking is bringing computation functionality closer to consumers. However, instead of creating a connection to a cloud environment as in conventional network deployments, the networking paradigm offers a flexible execution network for functions. Based on the loosely coupled communication model of ICNs, the network provides access to the most appropriate, in many cases the closest, execution node. Examples for such nodes are infrastructure nodes such as cellular base stations or other mobile nodes in the vicinity providing execution resources. Figure 3.6 illustrates an example of a computation-centric network deployment. A consumer uses a name to express a request for a computation result, for example scaling a certain maptile down (cf. Figure 3.6, step 1).

Based on the name, the request is forwarded towards any node capable of executing the operation. If required, functions can be downloaded from repositories on-demand (cf. Figure 3.6, step 2). After the computation has finished, the result is delivered back to the consumer. Furthermore, a replica of the result is stored within the local cache for subsequent consumers (cf. Figure 3.6, step 3). Due to the loosely coupled communication model, another consumer interested in the same result (here: consumer$_2$) can access it directly from nodes keeping a copy within their local caches (cf. Figure 3.6, step 4 & 5).

As a result, computation-centric networking decouples computational resources, storage and functions from the cloud backend closer to the consumers. By making use of the loosely coupled communication model of ICNs, the paradigm provides dynamic access to computation results. Therefore, it is a promising candidate for scenarios demanding for computation-intensive and latency-sensitive information.

Challenges of Computation-Centric Networking

Computation-centric networking denotes a relatively recent area of research. At the present time, there are challenges such as :

- *naming schemata*: establishing common naming schemes to ensure interoperability for function request and function execution as well as to provide access to dynamic content in heterogeneous networks.
- *orchestration and management*: supporting efficient strategies to resolve execution nodes and to keep the computation network in a vital state.
- *stateful/stateless computations*: supporting stateful and stateless computation.
- *computation security and privacy*: ensuring that the execution and the computed results are correct, valid, verified and trustworthy.
- *access management*: ensuring that only eligible network participants are able to trigger a computation or access a result.
- *network security*: preventing the network from attacks such as Denial-of-Service (DoS).

More details and discussions about security related challenges of computation-centric networks are presented in Section 8.

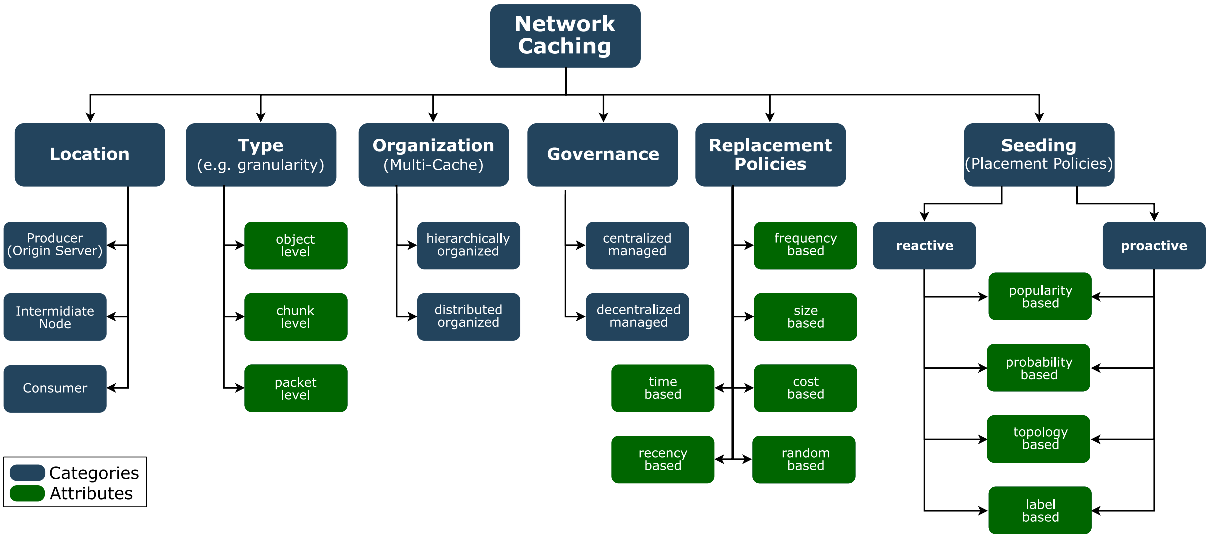


Figure 3.7: A taxonomy of network caching mechanisms. The different categories are illustrated in blue, while the corresponding attributes are given in green.

3.2 A Taxonomy of Network Caching

When looking into the previously introduced data-oriented networking paradigms (cf. Section 3.1), there is one common design element present in all of the paradigms: bringing data closer to consumers via *caching*.

Caching mechanisms have shown performance improvements such as reducing data delivery times. Due to the increasing number of network consumers over the past decades [69], caching mechanisms have played an important role in the success of global deployed networks such as the Internet. More precisely, caching mechanisms have the ability to [70, 71]:

- reduce the network bandwidth consumption, by storing data closer to the consumers, and therefore, decrease the risk of network congestion.
- reduce the perceived data access latency, by increasing the availability of frequently used data within the caches in the network.
- reduce the load at the origin server, by decoupling data from the server towards caching nodes.
- increase the resilience of the network, by providing data from cache nodes in case of an unexpected network partitioning (e.g., failure of the origin server).

The performance of a caching system is dependent on the number of participants in the network. The larger the number higher is the probability that a certain data item is requested again [70]. However, there are additional cache properties which affect the system. An analysis of different caching mechanisms and architectures has shown that network caching strategies consist of several elements. Figure 3.7 illustrates these components as part of a taxonomy of network caching: (i) *cache location* - the physical deployment (location) of the cache component in the network, (ii) *cache types* - the responsibility of caches to store a specific type of content, (iii) *cache organization* - the structure of multi-tier cache components in the network,

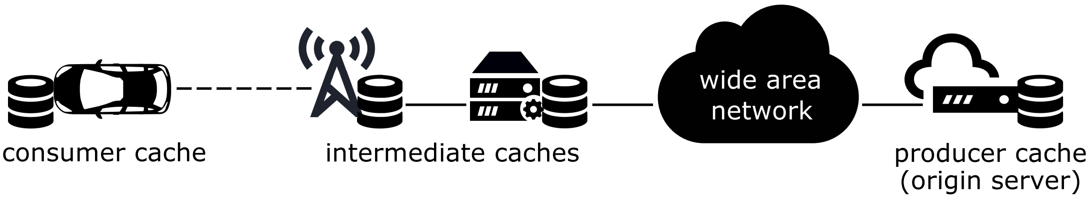


Figure 3.8: Cache memory deployment options including the local consumer cache, proxy caches at the edge or in the core network, as well as the local cache of the producer.

(iv) *cache replacement policies* - the management rules for evicting items from the local cache (e.g., to safe memory), and (v) *data placement* - the strategies of seeding the right data item into the right cache node. The following subsections will introduce the elements more in detail.

3.2.1 Cache Memory Deployment (Location)

The physical deployment (location) of the cache components in a network is a major factor of performance improvements and the consumer's experienced usability (e.g., [72]). A distinction is made between (cf. Figure 3.8):

- *Producer cache*: dedicated memory reserved at the origin server for load balancing purposes and to reduce the workload of the server applications.
- *Intermediate node cache (Proxy)*: any cache node between the producer and the consumer. These nodes can be regional or location specific caches, e.g., as present in CDNs (cf. Section 3.1.1), gateway or border nodes between the connection of network domains, or any other node in the core network including edge nodes, intra-domain nodes, etc.
- *Consumer cache*: the cache present on the end-user device, such as application specific caches (e.g., web browser).

The optimal deployment of caches is highly dependent on the application case. In the present time, a hybrid deployment solution supporting more than one of the introduced types is offered by network operators. Furthermore, the location of the cache component has an influence on the access times of data. Local and edge node caches can be used to bring data closer to consumer applications. Intermediate caches such as location or intra-domain nodes are able to reduce the traffic load in the (sub)-core network, and therefore, *increase the robustness* of the overall system. Finally, the deployment of cache nodes close to the producer balances the number of requests and prevents the producer application from being overloaded.

3.2.2 Cache Specialization Types

Another element of network caching is the specialization of the cache nodes regarding the data items to be cached (cf. Figure 3.7, second element). This applies particularly to the types of data items as well as the level of granularity. In the latter case, the decision of storing a data item can be made on three granularity levels:

- **object level**: storing an entire data object (e.g., a document), which may consist of several fragments. This granularity level requires more processing and memory resources as the other levels.

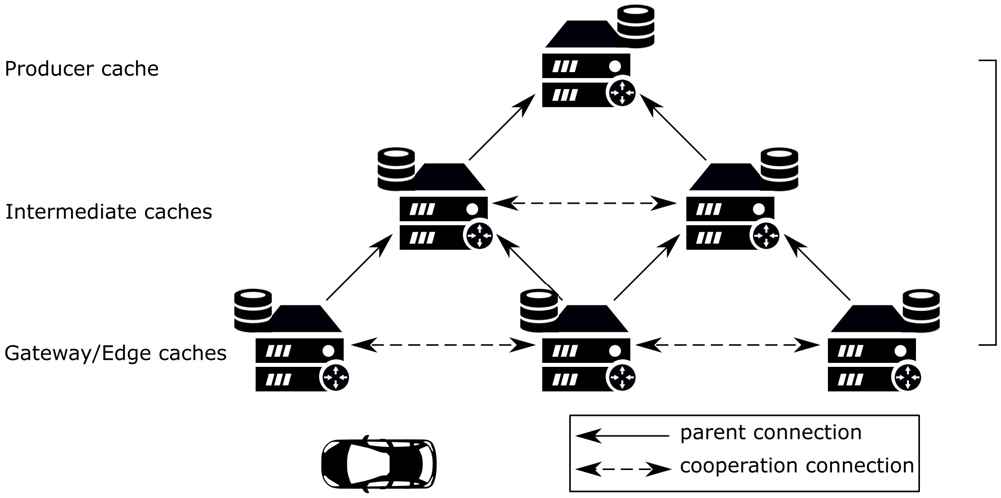


Figure 3.9: Example of a hierarchical (tree) organized cache deployment. Caches within the same level are able to consult their neighboring caches cooperatively. If no data can be served by the siblings, requests are forwarded to parent caches.

- **chunk level**: storing chunks – a unit of a data object which is divided into several parts (cf. [12]).
- **packet level**: storing packets of specific transport level flows. Such granularity is challenging since the transport flows need to be detected by the cache component (e.g., [73]).

Besides the cache granularity, the uniformity of caches describes another aspect of cache types. *Homogeneous* caches are optimized to store certain types of data items, for example, entertainment content or items of a certain application/service provider (e.g., Netflix, Spotify, etc.). However, the optimization goal of *heterogeneous* caches are different compared to the *homogeneous* ones, since they treat types of data items equally (e.g., geo-specific caches of a CDN provider).

3.2.3 Cache Organization Structures

Caching systems need to be flexible to react to communication path and topology changes depending on the topology and the behavior of the network participants. *Cache cooperation* describes a powerful concept to improve the effectiveness of the system, by both serving requests from other cache components as well as executing cache decisions (cf. Section 3.2.4). Dependent on the optimization goal (e.g., reducing bandwidth consumption or delivery times), one option to coordinate caches is setting up cache hierarchies. In general, there are two dimensions of cache organization: (i) *multi-tier* organized (e.g., hierarchically/meshed or on topic basis), and (ii) *distributed* organized cache deployments [74, 70]:

Hierarchically organization: Hierarchical organization of caches describes a topology in which multiple cache nodes are organized together according to a leveled structure. The idea was first proposed by the Harvest project in 1996 [75]. Structured in different levels, cache components are able to cooperate with each other by exchanging information such as statistics or actual data items.

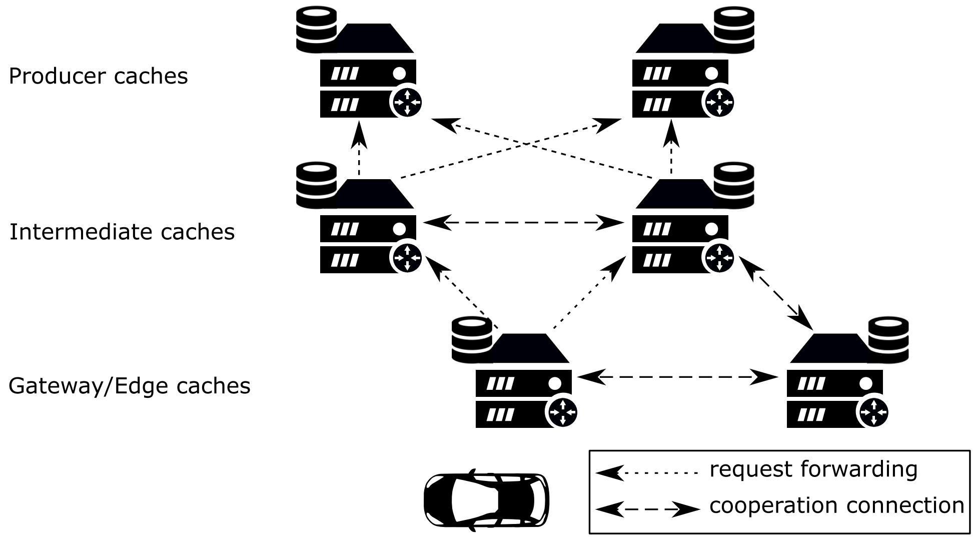


Figure 3.10: Example of a distributed organized cache deployment. Caches act independently according to the optimization goal. Furthermore, they are able to consult their neighboring caches cooperatively (optional). If no data can be served from the local cache, requests are forwarded to neighboring nodes towards the producer.

Each level follows a purpose: represented by geo-area (e.g., such as in CDNs: region → nation → continent → originator), application-specific (e.g., Youtube caches) or driven by other organization targets. The hierarchical structure depends on the optimization goal of the caches. For example, caches can be organized in an access-driven fashion, e.g., a tree structure (cf. Figrue 3.9), to reduce the bandwidth consumption at the higher levels in the core network. Figure 3.9 illustrates a hierarchical cache organization (three level depth) in which caches within the same level are able to cooperate with each other. If a data item cannot be directly served by the cache placed close to the consumer, the component can consult neighboring caches within the same level or forward the request towards the parent caches. These again can serve the requested item directly or forward the request, at least to the originator of the item. During the delivery of the item to the consumer, the intermediate cache nodes can store a replica of the item for subsequent requests.

One of the benefits of hierarchical caching is bandwidth efficiency, while the caches are able to cooperate with each other (e.g., on same level or at least consulting parent caches.) However, they have to be placed at strategical valuable positions in the network (e.g., closer to consumers).

Distributed organization: In distributed organized cache systems, components are entirely independent of each other. Figure 3.10 illustrates a distributed cache deployment. In such systems, there are no cache levels at all. If a cache component is not able to serve the requested item, it forwards the requests to the network until it reaches another cache components. Typically, most of the content is stored close to the consumers in such systems. As a result, distributed cache systems tend to be more fault tolerant by storing several copies of same data items close to consumers (e.g., [76]). However, the overhead of sharing information in a cooperative fashion bears the risk of increased load in the network or additional overhead of management.

While hierarchical and distributed cache organizations are opposing approaches, hybrid structures are using concepts of both extremes, for example by cooperating with other cache components within the same or at higher levels. One possibility to *cooperate* and to share information between cache components is the Internet Cache Protocol (ICP) [77]. It describes a protocol which is used for information exchange between different Web caches. Examples for such exchange is the advertisement of data items to neighboring cache nodes, which are therefore able to prefetch these items and store them into their local cache, in order to decrease delivery times.

3.2.4 Cache Governance

Besides the cache organization (cf. Section 3.2.3), other influencing aspects in cache cooperation includes the *decision* making process to increase effectiveness. Governance in IT systems describe the processes of decision making required to ensure the effective management of the system [78].

Cache *governance* includes who in the network (e.g., institutions, network providers, application providers, etc.) is eligible to perform optimization decisions, for example place certain data items closer to consumers. An important part of such systems are the *management* capabilities to perform the actual action of storing a certain item into a cache component to ensure governance [78]. As part of this thesis, a distinction is made between (i) *centralized* managed, and (ii) *decentralized* managed decision making.

Centralized managed: In centralized managed systems, there are dedicated components in the network which are eligible to make decisions (e.g., [79]). Centrality describes a property which is strongly linked to the topographical structure of the system such as inter-domain nodes (betweenness centrality), geo-specific nodes (closeness centrality), etc. In these systems, the decision nodes have additional knowledge (e.g., structure or statistics of the network such as topology, bandwidth usage, etc.) required to find a perfect solution for the optimization goal. Typically, such managed systems are applied to homogeneous networks such as telecommunication operators or specialized networks such as CDNs.

Decentralized managed: In decentralized managed systems, the decision making is managed by the cache nodes itself, autonomously, and not coordinated by a few nodes (e.g., [14]). It is based on the local available knowledge gathered by the cache component itself such as forwarding frequency, etc. Typically, such cache systems are applied to networks characterized by their heterogeneity, e.g., including ad-hoc networks.

3.2.5 Cache Replacement Policies

Besides the physical deployment and the organization of cache components in the network, there are further important aspects to be concerned about in a caching system: (i) the data items to be delivered, and (ii) the corresponding resource management.

It is to be expected that the available memory resources in the network will further increase, while the costs of storage decreases steadily. However, the available storage at each individual cache component is limited, since the requested amount of data is expected to increase as well [76]. This becomes more problematic, since items in network caching systems vary in their size, unlike as in CPU caches or virtual memory.

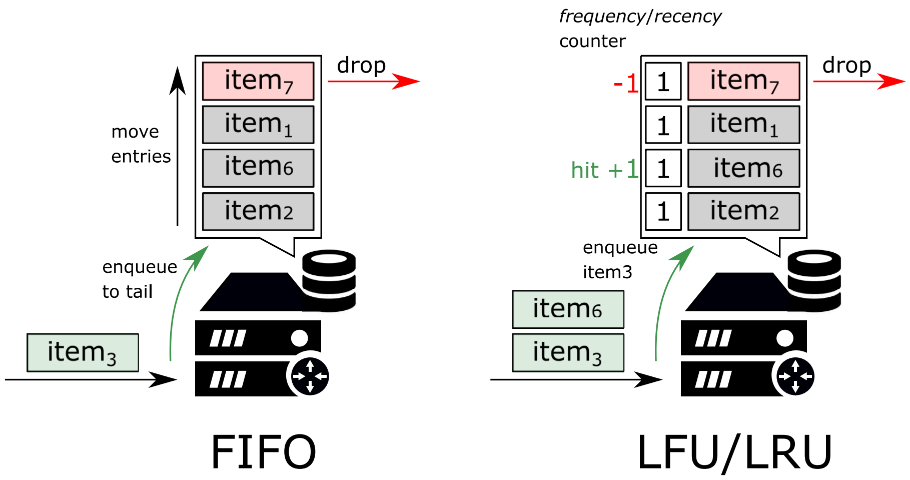


Figure 3.11: Examples of replacement algorithms for cache systems based on [81]. All examples make use of a queue implementation with limited capacity of four entries. Dependent on the strategy, old entries are evicted and replaced with newer ones. In case of the Least Recently Used (LRU) strategy, the semantics of the queue counter is slightly different compared to the Least Frequently Used (LFU) strategy.

While large memory storage results in more content to be cached locally, and hence, find items in the local cache (cache hit), it results in additional lookup overhead, and therefore, lead to longer response times.

Cache replacement policies describe a class of caching algorithms concerned about the limited resources at each cache node. Such strategies take action when reaching the capacity limits of a cache component, requiring data items to be evicted from the local storage for new ones (cf. [80]). Since the early ages of cache systems, replacement strategies have been an essential part of the research. In the past decades, several categories of replacement strategies have been proposed in the literature (e.g., [71, 80]). As part of this thesis, just a few popular ones will be introduced in detail (cf. Table 3.1):

- **frequency/recency-based**: the number of requests or the time since the last request of an item matters – less frequent/recent requested items are evicted first from the cache
- **size-based**: size of an item matters – larger items are evicted first from the cache
- **cost-based**: cost of fetching an item from the origin server matters (e.g., distance, bandwidth, etc.) – less expensive items are evicted first from the cache
- **random-based**: items are evicted randomly from the cache
- **time-based**: time since the last modification, or expiration time – items with expired or old timestamps are evicted first from the cache

Frequency/Recency-based replacement: As long as data is of interest to a larger group of consumer/applications, caching such information make sense. Over the past decades different frequency and recency-based algorithms have found their way from computer designs (e.g., CPU or virtual memory management) into the domain of computer networks.

Table 3.1: Examples of replacement strategies in caching systems based on [71, 80]

Policy	Reference	Criteria	Category
FIFO	[81]	arrival time	recency-based
LFU	[82]	request frequency	frequency-based
LRU	[83]	popularity	recency-based
Pitkow/Recker	[86]	popularity	recency-based
Size	[85]	content size	recency-based/size-based
RAND	[81]	random	randomized
GD-Size	[85]	cost/size	function-based
GDSF	[84]	cost/size/frequency	function-based

One of the first replacement algorithms is First-In-First-Out (FIFO) [81] (cf. Figure 3.11, left-hand side). Using FIFO as a strategy, items are enqueued based on their arrival time. In case the queue has reached the limit, the item on top will be deleted and the latest item is enqueued at the tail. More advanced replacement algorithms use approximations to find the most suitable optimum of cached items. LFU [82] (cf. Figure 3.11, middle) and LRU [83] (cf. Figure 3.11, right-hand side) are two of the most popular approximation algorithms in the caching context. Regardless of the size of the items in the cache, the request frequency or the last use of an item is utilized to decide which items have to be kept vital in the cache. Less frequently requested or less used items are evicted from the local storage. Over the years, optimizations of these algorithms have been proposed in the literature such as LFU-Aging [84], LFU with Dynamic Aging (LFU-DA) [84], or LRU-Min [83].

Cost-based replacement: More complex algorithms combine decision criteria together towards cost-based approaches. For example, the GreedyDual-Size (GD-Size) [85] algorithm replaces the object with the lowest utility, while the utility is defined by *cost/size* values: the effort to fetch the item from the network and its allocated size in the cache. Further improvements of the algorithm such as GreedyDual-Size with Frequency (GDSF) [84] incorporates the request frequency as additional decision criteria.

While the deployment of one replacement algorithm may result in good performance (hit ratios), the combination of these algorithms may even improve these values. For example, LRU and SIZE (cf. Table 3.1): In case items within the cache have the same recency values, the SIZE algorithm is responsible for the replacement decision. In order to decrease the miss ratio, the SIZE algorithm incorporates the allocated size of the items in the cache. It tries to delete one large item rather than many smaller items from the cache.

3.2.6 Cache Placement Strategies

One last group of the introduced taxonomy is described by data item *placement* strategies. Placing the right data item (*what*?) within the local storage of the right node in the network (*where*?) at the right time (*when*?) has the potential to decrease the consumer perceived delay tremendously, as well as to balance the network bandwidth consumption. However, it describes a non trivial task.

Table 3.2: Examples of content placement strategies in caching systems based on [87, 88]

Feature	Reference	Cache Model	Execution	Category
LCE	[12]	decentralized	reactive	probability (fixed)
LCD	[89]	cooperative	reactive	graph/popularity
RND	[90]	decentralized	reactive	probability
Prob	[89]	decentralized	reactive	probability
ProbCache	[14]	cooperative	reactive	probability (dynamic)
Betweeness Centrality	[91]	centralized	reactive	graph
WAVE	[15]	cooperative	reactive	graph/popularity

Cache placement policies describe a class of caching algorithms concerned about the efficient distribution of data items across network caches, for example to increase their availability. A distinction is made between *reactive* and *proactive* caching mechanisms:

- *reactive caching mechanisms* describe a class of caching strategies which store data at intermediate nodes during delivery. While such class of strategies has shown performance improvements by bringing data closer to consumers (e.g., in CDNs), it is not efficient for highly dynamic networks due to the fact that forwarding routes between the mobile participants change constantly.

- *proactive caching mechanisms* describe a class of caching strategies taking action by placing a consumer's anticipated content at the right network nodes in time, before a request is sent by a consumer. As a result, proactive caching mechanisms are able to reduce the latency of content retrieval, and thus, provide a certain degree of quality for data delivery by reducing handover delays in WiFi and cellular networks (cf. [17]). However, placing the right data at the right node in time is a non-trivial task.

Recently, such kind of strategies have attracted the attention of researchers in the field of ICN. The loosely coupled communication model and the intrinsic in-network caching capabilities leverage the distribution of items. In general, data items are placed in caches either on the delivery path (*on-path*) or anywhere else in the network (*off-path*). Several categories of placement algorithms have been proposed in the literature (e.g., [87, 88]). Table 3.2 illustrates the most popular placement strategies in caching systems.

- **popularity-based**: In *popularity-based caching strategies*, the decision is based on distribution metrics such as the request frequency and recency of data items. Examples for popularity-based algorithms are Leave Copy Down (LCD) [89] or WAVE [15][6].

- **probability-based**: In *probabilistic caching strategies*, the decision of creating a replica at a node is based on a probability value p. It is determined between a *fixed* (immutable, defined a priori) and a *dynamic* (mutable) value, based on a distribution function. Examples for such placement algorithms are Leave Copy Everywhere (LCE) [12] (fixed) or ProbCache [14] (dynamic).

[6] Please not to be confused with the vehicular wireless communication stack WAVE presented in Section 2.4.1.

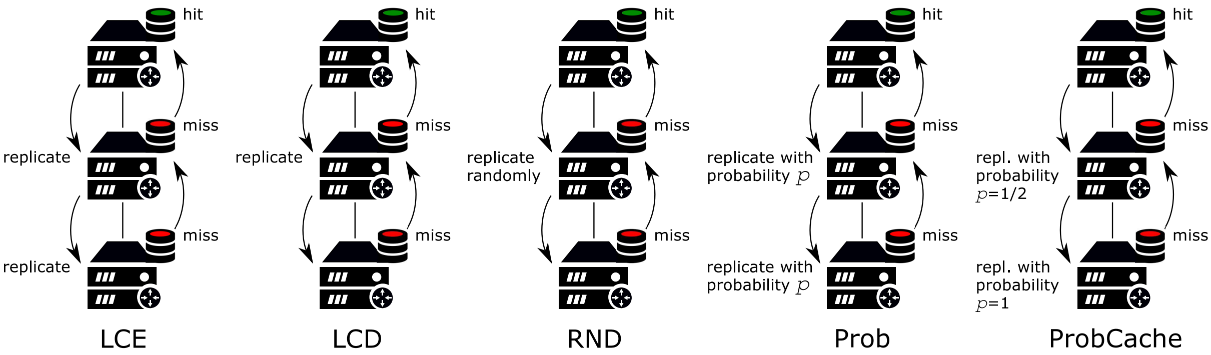


Figure 3.12: Illustration of different probability-based placement algorithms and their operational steps based on [87]. In case of a cache hit, intermediate nodes create replicas of the forwarded response item according to the deployed placement algorithm.

- **topology-based**: In *topology/graph-based caching strategies*, the topological metrics of the network are used for decision making. This includes centralized nodes in the network topology serving a large amount of communication traffic. An example for such a strategy is the *Betweeness Centrality* [91] algorithm.

- **label-based**: In *label-based caching strategies*, cache nodes are strongly advised to create replicas of a certain kind of data types. For example, this includes data items of a specific type of content, identified by their labels [92].

An illustration of the operational steps of some of the presented placement algorithms of Table 3.2 is given in Figure 3.12. Every time a request can not be answered directly from the local storage (cache miss), the request is forwarded until an item is found in a cache (cache hit). During the delivery path, each node which provides cache functionality can decide to create a replica of the information. While the LCE (fixed probability $p = 1$) algorithm induces that every node on the delivery path creates a replica (cf. Figure 3.12, 1st algorithm), the LCD approach optimizes the memory allocation by moving data items one step closer to the requesting node every time a cache hit occurs at a higher level in the network (cf. Figure 3.12, 2nd algorithm). Instead of replicating each data item during the delivery path, algorithms such as *RND* [90] is based on the result of a random calculation (cf. Figure 3.12, 3rd algorithm). As a result, the number of replicas created by *RND* is less than the number of replicas created by the LCE approach, and hence allocates less memory at the cache node. More complex algorithms consider distribution functions for the placement decision. For example, the *Prob* [89] caching strategy (cf. Figure 3.12, 4th algorithm). Based on a priori probability values, cache nodes create a replica according to these values. Another example is given by *ProbCache* [14]. Similar to the *Prob* algorithm, ProbCache creates a replica according to a probability p. However, p varies inversely proportional to the distance from the requester to the cache node (cf. Figure 3.12, 5th algorithm).

The usage of cache placement algorithms has shown performance improvements by bringing data items closer to the consumer, especially in mobile ad-hoc networks. Placement and replacement strategies complement each other, while the former decides *what* data item is stored *where* in the network, and the latter algorithm decides *how* the local memory is optimally used.

In general, the decision on which placement as well as replacement algorithm(s) to be used is highly dependent on the applications executed in the network as well as the optimization goal to achieve.

3.3 Summary

This chapter has introduced the state-of-the-art in data-oriented networking principles. Core concepts of four networking paradigms, namely (i) content-delivery networks, (ii) delay-tolerant networks, (iii) information-centric networks, and (iv) computation-centric networks have been presented, compared to conventional Internet deployment and are outlined the main challenges. While the paradigms are different in their focus area (e.g., support sparse networks, or reducing the load in the core network), in-network caching capabilities are present in all of the introduced approaches.

Based on this finding, a taxonomy of network caching has been presented in the second part of the chapter. In more detail, the elements of the taxonomy, namely (i) memory deployment, (ii) specialization types, (iii) organization structures, (iv) cache governance, as well as (v) replacement and (vi) placement strategies have been derived from the analysis of the literature. Examples are provided for each of the element groups based on existing policies.

Summarized, the principles of ICN networks such as the loosely coupled communication model, the natural support for mobility as well as the intrinsic in-network caching capabilities have shown to be promising to improve data delivery, especially in mobile ad-hoc networks. The following chapter introduces the state-of-the-art in information-centric networking regarding connected vehicle environments.

4 A Survey on Information-Centric Networking Architectures for Connected Vehicles

> Data is a precious thing and will last longer than the systems themselves.
>
> Tim Berners-Lee

In information-centric networks, the loosely coupled communication model and the resulting support of mobility has attracted the attention of researchers in the area of connected vehicles. This includes the Information-Centric Networking Research Group (ICNRG) which defined connected vehicles to be one of the ICN focus topics [93]. In the past decades, several ICN architectures have been proposed, raising the question of the most applicable ICN approach in the area of connected vehicle environments.

In this chapter, a comprehensive survey on information-centric networking architectures and their capabilities regarding connected vehicles is given. Due to the quantity of publications related to the field of ICN in connected vehicle environments, this chapter is divided into two parts. First, a survey of different ICN architectural approaches is presented (cf. Subsection 4.1)[7]. Second, an extensive analysis of the literature in connected vehicle environments with focus on caching in ICN-based connected vehicle environments (cf. Subsection 4.2) is outlined.

4.1 Survey on Information-Centric Networking Architectures for Connected Vehicles

In the past decade, a significant number of ICN architectures have been proposed by research activities in academia and industry around the globe. An illustration of the historical events of modern ICN research is presented as time line in Table 4.1.

First started by the work done as part of the TRIAD project, Gritter et al. [61] proposed an overlay solution supporting content routing design based on naming schemes. It described one of the first approaches of changing the addressing scheme from hosts towards data and formed the basis for all subsequent research activities in the field of data-oriented networks.

Based on the idea of separating content from its physical location, Van Jacobson presented the vision of future trends in networking as part of a Google tech talk such as name-based routing in 2006 [94]. The first architecture considering the ICN core elements was introduced by the Data-Oriented Networking Architecture (DONA) project in 2007 [95]. The architecture proposed an addressing scheme by separating identifiers and locator from each other as well as replacing hierarchical URLs with flat names. This was soon followed by other projects, such as Publish-Subscribe Internet Routing Paradigm (PSIRP) [96], 4WARD [97] and Line Speed Publish/Subscribe Inter-Networking (LIPSIN) [98]. The results of these projects were used in subsequent projects such as Publish-Subscribe Internet Technology (PURSUIT) [99] (successor of PSIRP and parts of LIPSIN), and the Network of Information (NetInf) [100] (as part of the Scalable and Adaptive Internet Solutions (SAIL) [101] project and the successor of 4WARD).

[7] The content of this section is based on the published work in [20]. Parts of it are extracted from these sources.

In 2009, the full-fledged ICN architecture Content-Centric Networking (CCN) was introduced by Jacobson et al. [12]. The architecture describes a revolutionary networking approach by replacing the Internet protocol stack and using hierarchical names. As part of this architecture, the *interest-based* exchange of information was first introduced which prescribes the exchange based only on two packet types: `INTEREST` and `DATA`. A first implementation of the concepts of CCN were introduced as CCNx platform [102]. In subsequent years, architectures such as (i) Content Centric Inter-networking Architecture (CONET) [103] (part of the Convergence project [104]), (ii) Convergence of Fixed and Mobile Broadband Access/Aggregation Networks Architecture (COMBO) [105] and (iii) Content-centric fashion MANET (CHANET) [106] adopted concepts of CCN in their architectures.

In 2010, the US National Science Foundation introduced a program for FIA. As part of the program, two proposed architectures are considering ICN core principles: *MobilityFirst (MF)* [107] and *Named Data Networking* [13]. The main objective of the MF project is to support mobility by providing seamless communication between participants. The architecture also makes use of naming principles and is therefore categorized as an ICN approach. Named Data Networking (NDN) started as an academic spin-off of the CCN approach. Similar to CCN, there are some projects which adopt or extend the concepts such as (i) Green Information Centric Networking (GreenICN) [108] or (ii) NFN [109]. While the information exchange mechanism in NFN are based on the CCN/NDN principles, NFN is implicitly covered by the capabilities of CCN/NDN in this comparison.

As part of the FP7 EU program, the Content Mediator Architecture for Content-aware Networks (COMET) [110] introduced a content-oriented architecture. The main objectives of the project is to simplify the access to content and to support certain quality of service mechanisms by introducing a flexible content mediation plane. The project is also based on essential concepts of CCN/NDN introduced by Jacobson et al [12].

In 2015, the European funded projects Architecture for an Internet for Everybody (RIFE) [111], Universal Mobile-centric and Opportunistic Communications Architecture (UMobile) [112], Bonvoyage [113], and POINT (iP Over IcN - the betTer IP) [114] started their work on the next steps of ICN-based networks. Based on principles of the PURSUIT architecture, RIFE [115] focused on a unified architecture providing access to information by combining principles from ICN and DTN for the future Internet. UMobile complements the major goals of the RIFE project by focusing on an ICN-based mobile-centric architecture to simplify the communication exchange between mobile participants in opportunistic networks such as sensors, home automation environments and mobile handheld devices. For this purpose, UMobile defines an architectural framework supporting different connectivity options from the domains of information-centric (using NDN principles), delay-tolerant and opportunistic networking [116].

Bonvoyage pushes the idea of providing access to information further by proposing a platform used for unified collection, distribution and sharing of information across heterogeneous IoT networks.

On the hypothesis that some current IP-based applications benefit more from an underlying ICN-based network, POINT provides a platform to run IP-based applications such as legacy applications during the interim period of a global deployed ICN network. Similar to RIFE, the POINT project is based on principles of the PURSUIT architecture [117].

In the mid of 2016, the European project ICN2020 started. The main objectives of the project are to enhance existing ICN solutions and architectures in the context of the IoT by including principles of cloud, virtualization and CDN principles into ICNs.

In 2017, Cisco acquired the CCN platform from the Palo Alto Research Center (PARC) [118] and published the sources of the platform under the name Community Information Centric Networking (CICN) as an open source project [119]. In the same year, the US National Science Foundation and the Intel company formed the Information Centric Networking in Wireless Edge Networks (ICNWEN) partnership to investigate ICN principles in wireless edge network environments [120].

Due to the fact that ICN2020, CICN and ICN-Enabled Secure Edge Networking with Augmented Reality (ICE-AR) are based on the principles of CCN/NDN, as well as that the projects are still under development, there are no final versions of the architectures available yet, and therefore, the projects have not taken into account for this comparison.

The following sections are separated into two parts. First, requirements from the automotive IoT are introduced based on the use cases presented in Section 1.2 such as *naming*, *mobility*, and *security*. The second part provides a detailed comparison of the introduced ICN approaches and their capabilities aligned to the previous presented requirements.

4.1.1 Architectural Requirements

The requirements to be discussed are based and derived from the introduced use case scenarios of Section 1.2, namely the *electronic horizon* and the *community-based sensing*. The requirements are categorized into five groups:

- *Naming* - how the different naming strategies affect automotive services with respect to data discovery
- *Mobility* - how the high degree of mobility of participants is supported by the architectural approaches
- *Routing, Caching and Transport* - how the different data handling mechanisms operate in the context of connected vehicles
- *Safety & Security* - how the examined ICN architectural approaches support the construction of safe & secure services in the automotive context
- *Interoperability & Community* - how the interoperability of different ICN approaches as well as legacy architectures are supported and how the architectural approach is supported by a community

Naming

The discovery of network participants providing data defines one of the major requirements in vehicular networks (cf. [6]). This also includes vehicular systems using ICN as the underlying network technology (e.g., [128, 11, 129, 130]). Based on *content identifiers*, ICNs work with naming schemes to identify and access data, represented as Information Objects within the network (e.g., [12, 13]).

CFN [64]	2019	
	2018	Cefore [121]
NFaaS [66]	2017	CICN [119] | ICE-AR [122]
	2016	ICN2020 [123] i3 [124]
	2015	**POINT** [114] **UMobile** [112] **RIFE** [111] **Bonvoyage** [113]
	2013	**COMBO** [105] **GreenICN** [108]
NFN [109] | JUNO [125]	2012	
CHANET [106] | COPSS [126]	2011	ANR Connect [127]
	2010	**SAIL** [101] **PURSUIT** [99] **COMET** [110] **NDN** [13] **Convergence** [104] **MobilityFirst** [107]
CCN [12] | LIPSIN [98]	2009	CCNx [102]
	2008	PSIRP [96] | 4WARD [97]
	2007	DONA [95]
CCN Google Tech Talk [94]	2006	
	2001	TRIAD [61]

Table 4.1: Timeline of the historical events of modern ICN research activities based on [63]. Architectural ICN papers are presented on the left side, while ICN-related projects are presented on the right side.

Regarding the introduced use cases in Section 1.2, both network participants – the mobile consumer and producer of data – need to know how to discover each other and access application specific data within the network. For example, a producer who sensed a hazard needs to provide such information, so that the approaching drivers get it in time.

Unified naming and addressing schemes are crucial for automotive service communication. In order to provide an efficient accessibility of data for automotive services, naming schemes are supposed to be *readable* to be processed by network components automatically, and therefore, facilitate discovery. Furthermore, particular schemes offer the opportunity to define *self-described* names or functions to describe the representation of the data and its semantics. These elements provide mechanisms to augment information objects with syntactic and semantic meta-information describing the context of data or the data itself (e.g., [131]). By enhancing and combining descriptions in a machine processable structure, it creates the basis for automated information selection, aggregation, and processing according to the preferences of the network participants.

Besides providing syntactic and semantic information, naming capabilities to define the scope of the data are useful to simplify data discovery across different IoT domains. Such a feature can be used to define the limitations a data item is offered or used the network participants, e.g., the information or access context. Regarding the *community-based sensing* example, the name ensures that only a group of subscribed vehicles within a certain geo-location receive the information of available parking spots.

Long life cycles of vehicles have an effect on the naming schemes as well. According to IHS Markit [132], the average age of light cars in the U.S. is about 11.6 years in 2016. Schemes need to be *extensible* to react to conceptional or technological changes and *flexible* to cover multiple automotive service domains, in the future. Furthermore, *scoping* concepts supported as part of a naming scheme are used to provide access to data in a given context (e.g. certain vehicle models, geographical region or security group).

Based on the introduced naming aspects, the following naming requirements are considered: (i) *machine/human readable*, (ii) *self-describing*, (iii) support of *scoping* and (iv) the support of *flexible schemes* capabilities.

Mobility

In the network of connected vehicles, participants freely join and leave the network while they are moving. As a result, participants are required to reconnect to network access points frequently. This fact also influences the communication network (cf. [6]), and is also valid for ICNs (e.g., [133, 134, 130]). Regarding the *electronic horizon* example, vehicles passing by an accident communicate the hazard as warning information with an infrastructure and oncoming vehicles periodically. Such a degree of mobility results in high variations and fast changing network topology and challenges network structures to provide *seamless connectivity*. The network architecture needs to account for the movement of vehicles within and between local networks.

As mentioned in Section 2.3, the number of available communication technologies forms the basis for *heterogeneous vehicular networks* – supporting multiple communication interfaces and technologies. Built-in components offer new opportunities to communicate with network participants efficiently. On the one hand, *multi-channel* offers the combination of multiple available communication technologies concurrently, for example to decrease latency [135]. On the other hand, *multi-homing* is a feature to access data from multiple access points and multiple sources simultaneously, for example to increase availability. Such capabilities ensure Quality of Service (QoS) and Quality of Experience (QoE) of automotive services.

With respect to the support of mobility, the following requirements are considered for further investigations: (i) support of *consumer & producer mobility*, (ii) support of *seamless connectivity*, (iii) *multi-channel* and (iv) *multi-homing*.

Routing, Caching and Transport

The high degree of mobility in connected vehicles challenges a networking technology at different levels such as *routing*, *caching* and *transport*.

In general, the network of connected vehicles is characterized by low or limited bandwidth capacity, high-latency and communication failures [5]. A loosely coupled communication model facilitates the exchange of data in dynamic networks and by caching everywhere in the network (e.g., [6]). Through its nature, *reactive* routing and forwarding of data defines a crucial requirement matching network metrics or use case specific requirements such as latency. A desirable ability of the network is defined by *predictive* routing and forwarding mechanisms for the efficient dissemination of data within the network.

This also includes *caching* capabilities. Ideally, the next data chunks are cached *proactively* on multiple nearby radio cells or wireless access point. The term proactive caching describes a class of strategies taking action before a request is sent by a consumer. Placing data at the right network elements in-time increases the network performance by reducing the total number of cached objects and decreasing latency (cf. [18, 136]). However, some data is required to be cached at some network nodes *reactively* (cf. Section 3.2.6).

By looking at long life-cycles of vehicles (cf. Section 4.1.1), transport protocols need to be *extensible*, however, mainly attributable in low overhead and a *flexible* chunk size to cope with bandwidth characteristics of underlying network technologies. The term flexible chunk size describes the capabilities to modify the size of a data chunk to the underlying maximal unit to transfer, and therefore, achieve optimal bandwidth usage. On the one hand, it needs to be capable of adaption to any changes in the future (e.g. technical evolution). While on the other hand, the backward compatibility must be given to communicate with existing versions of automotive services.

With respect to the group of routing, caching and transport requirements, the following metrics are considered: (i) *reactive and predictive routing*, (ii) *proactive and reactive caching*, (iii) *caching nodes* and (iv) the *routing strategies*,as well as the (v) *extensibility of the protocol* and the support of (vi) *flexible data chunk sizes*.

Safety & Security

Especially in the automotive industry, safety and security mechanisms play an important role in both the development and the operational phase of vehicles [137]. Such mechanisms are intended to provide maximum protection for the passenger and its surroundings (cf. [6]). In-vehicle safety-critical applications are expected to share data with other vehicles and cloud systems (e.g., [46, 129]). As part of the use case examples, vehicles which are heading towards a hazardous scene need to be warned about the situation. Such time-sensitive data needs to be communicated in a *reliable, accurate and consistent* manner across the network to ensure *functional safety*.

From another perspective, security concerns arise by opening a previously local system, as the vehicle, to be able to participate in services within networks. Along with unique naming schemes, essential security requirements are defined by identity *authentication*, verified content *integrity*, and a broad range of access controls via *authorization* policies. Within the *electronic horizon* example, only eligible network participants shall be able to access and process individual-related data. An ongoing discussion is given by *self-certifying* names offering new options to verify whether returned data is valid or not by checking its digital signature (cf. [138]).

Regarding safety and security relevant aspects, the following requirements are taken into account: (i) *authentication*, (ii) *authorization*, (iii) *data integrity* and (iv) the support of *other QoS machanisms*.

Interoperability & Community

One last group of requirements is driven by the market situation within the automotive industry: *interoperability* and the backing by a *community*.

The heterogeneous market of the automotive industry does not only consist of a few big car manufacturers but also includes countless suppliers and engineering companies (e.g., Robert Bosch GmbH, Denso or Magna International Inc.) [139]. The combination of high pressure regarding new innovations and cost efficiency, it is not very likely that all of these companies will agree on a single ICN approach. In order to keep chances high for ICN in this market, an architecture needs to be inter-operable towards other information-centric architectures as well. Another point on the way forward for ICN in connected vehicle environments is to convince the rather conservative automotive industry towards such a disruptive technology. On one hand, this makes it necessary to create a migration path for the transition from a classic host-centric communication approach towards ICN. Therefore, the selected ICN approach should provide features that allow a mixture of data-oriented and host-centric communication in a single system. On the other hand, the selected architecture should be backed by a large community of academic and industrial researchers to guarantee a long-term perspective for the manufacturers. Finally, the license model of the selected architecture is of great importance as well. This is because a car manufacturer does not want to rely on a single supplier only, but asks for a license that allows many companies to develop and sell components using this technology.

Summarizing the last group of requirements, the following interoperability and community metrics are: (i) *interoperability with other ICN architectural approaches* and (ii) *legacy host-centric networks*, as well as the backing of (iii) *an active community* and the (iv) *availability of source code and its license model*.

4.1.2 A Comparison of different ICN Architectures towards Connected Vehicles

Based on the introduction of the considerable ICN architectures as well as the previously presented requirements, the architectural capabilities are compared against each other. The labels used in the tables describe the degree of a supported feature and are given as follows: ✓ indicates a full-fledged feature, ● marks a partially supported, while ✗ indicates a not supported feature. Not available information are noted as *n.i.a.*

Naming

Unified naming and addressing schemes are crucial for automotive service communication since the consumer and producer of data need to know how to access data within the network. The introduced ICN approaches are compared to their naming capabilities and against the naming requirements of Section 4.1.1: (i) *machine/human readability*, (ii) *self-describing*, (iii) *scoping* and (iv) the support of *flexible schemes*. Table 4.2 provides an overview of the naming comparison.

PURSUIT: PURSUIT provides access to information by using a statically unique pair of *scope IDs* (SID) and *rendezvous IDs* (RID). Composed by rendezvous nodes (RN), both identifiers are used within a rendezvous network (RENE) system to link consumer and producer together. While theoretically any naming scheme can be used in PURSUIT in the RENE, the architecture does not fulfill the requirement of supporting flexible naming schemes completely. This is due to the fact, that after registration of an information object the statically unique naming pair can not be altered afterwards. After the process of identifying individual information objects by using the RENE, PURSUIT places the information into a context. This context is called *scope* and defines a set of information that is disseminated for the realization of a particular request of information.

NetInf: The architecture uses a predefined naming scheme to access information in the network. The scheme is based but not limited on a hierarchical structure and supports name spaces for human readable and flat names. Furthermore, NetInf provides the option to augment data with additional contextual information such as a scope and an authority. Within the scope of the predefined naming scheme, the names of information objects can be defined on application level.

MobilityFirst: MF defines a separate name-based service layer where data and services are identified and routed on a *flat label* strategy, similar to PURSUIT. First human readable names are used to lookup hash-based *global unique identifier* (GUID) in the system. Afterwards, the GUID is resolved to a physical network address using a global name resolution service (GNRS) (similar to DNS). The advantage of such approach lies in a strict decoupling of naming and addressing. The network layer of MF supports the definition of flexible groups of devices or users, and thus, provides a certain level of *scoping*. Due to the fact that human readable names are mapped to unique GUIDs, there is no self-describing feature in MF.

COMET: If a producer offers novel information in the network, COMET prescribes a *registration* procedure as part of a *content resolution system* (CRS). During the registration process, each information can be specified as a *scoped publication*. The approach identifies information objects by using (i) unique content IDs (which are machine-readable) or (ii) content names (representing human-readable aliases). The architecture does not specify a certain naming scheme. Therefore, flexible names need to be handled by the CRS.

CCN/NDN approaches: Regarding the CCN/NDN based architectures, there is no standardized naming scheme for addressing data objects. All approaches support a human (e.g. hierarchical URL based) and machine (e.g., hash-based) readable naming scheme as well.

The NDN project team published a tech report regarding naming conventions in NDN [140]. In CCN, so-called *active names* – names for Interests that do not exist yet – allow CCN to transparently support a mix of statically cached and dynamically-generated content [12]. The commonalities of the approaches also applies to the capabilities with respect to *scoping* and *self-description*, but not completely to the requirement of *flexible schema*. The architecture of the Convergence project extends the naming scheme and specifies the structure of identifiers as a tuple of <namespace ID, label>. The namespace ID determines the format of the label field. Thus, the label field is a namespace-specific string. Within the namespace, each of them is allowed to use its own rules to generate unique names in its own format. Such mechanism supports the definition of fine-grained naming scopes.

POINT and RIFE: In the POINT project, as well as in RIFE (since RIFE uses the development of POINT for IP-over-ICN communication [141]), the principles of receiving named content is evolving. The main focus of POINT is to provide a drop-in replacement of the existing core network (e.g., of a network provider) to increase the network performance by combining the ICN and IP world together [117]. In order to serve IP requests from end-devices across an ICN core network, POINT defines a gateway approach directly at the APs. Afterwards, content is resolved within the ICN core network according to the PURSUIT principles within the RENE. In addition to the translation of IP and ICN, the RIFE architecture introduces a DTN handler required for applications demanding for delay-tolerant transport of content. Based on the gateway approach of POINT, the DTN handler is a part of the AP components in RIFE [141].

UMobile: Based on the hierarchical naming concepts of CCN/NDN, UMobile extended the concept by providing more flexible schemes, better suited for facing the challenges set by today's heterogeneous networks and their services. The concept consists of two major parts: (i) fixed hierarchical name (based on NDN principles), and (ii) hashtags . While the former part is used to identify static content or application functionality in the network, the latter part provides a mechanism to semantically annotate content (e.g., provide contextual information) or to pass additional data to the application. Such flexible and enhanced naming concept is used to resolve content in the network, in the caches as well as for efficient routing and forwarding [142]. Additional functionality such as scoping is supported by the DTN part of UMobile [143].

Bonvoyage: The main concepts of Bonvoyage are based on the NDN architecture. However, the approach extends the naming capabilities towards more flexible naming schemes to address a wide range of participants in the network including humans, geo-location, group of devices. Names are managed by a object resolution system (ORS). In order to resolve a name for a desired data item, consuming entities consult the ORS to search for names. By introducing the concept of *Internames* [144], Bonvoyage enables name-to-name realms able to address any content, application programming interfaces, nodes, groups, geo-area etc. in the network including IP-based addresses. As a result, names in Bonvoyage can converge to any other naming concepts (e.g., those of MobilityFirst).

Summary: Naming defines one of the major objectives in the ICN paradigm and thus is considered in all presented ICN approaches. The comparison of naming capabilities shows different concepts with respect to naming schemes, the resolution of names to Information Objects

Table 4.2: Comparison of Naming capabilities of the presented ICN architectures.

Naming	machine / human readable	flexible scheme	scoping	self-describing
PURSUIT	✓ [145]	● [145]	✓ [145]	n.i.a.
NetInf	✓ [146]	✗ [146]	✓ [146]	n.i.a.
MF	✓ [147]	● [147, 148]	✓ [147]	✗ [149]
COMET	✓ [150]	✓ [150]	✓ [150]	n.i.a.
CCN	✓ [12]	✓ [12]	✓ [151]	✓ [12]
NDN	✓ [13]	✓ [13]	✓ [152]	✓ [12]
Convergence	✓ [153]	● [153]	✓ [153]	✓ [153]
COMBO	✓ [154]	✓ [154]	✓ [12]	✓ [12]
GreenICN	✓ [155]	✓ [108]	✓ [155]	✓ [12]
RIFE	✓ [145]	● [145]	✓ [145]	n.i.a.
POINT	✓ [145]	● [145]	✓ [145]	n.i.a.
UMobile	✓ [143]	✓ [143]	✓ [116]	✓ [116]
Bonvoyage	✓ [144]	✓ [144]	✓ [144]	✓ [144]

and the flexibility to cope with different naming schemes. Furthermore, it can be seen that only UMobile and Bonvoyage provide built-in features to augment information objects with additional meta-information as part of *self-describing* requirement. It reflects the focus of both projects on mobile network participants.

Mobility

One identified common denominator of connected vehicles is the high degree of mobility of communication participants. One of the reasons to facilitate mobility in ICN is the loosely coupled communication model introduced by the separation of data from physical locations. This section examines the mobility concepts and capabilities of the ICN approaches with respect to (i) *consumer & producer mobility*, (ii) *seamless connectivity* support as well as (iii) *multi-channel* and (iv) *multi-homing* capabilities. An overview of the mobility comparison is provided by Table 4.3.

PURSUIT: The publish/subscribe nature of PURSUIT decouples producer and consumer, and therefore, facilitates mobility. Regarding seamless connectivity at consumer side, the architecture introduces two proxy-based approaches in which additional network components, so-called proxies, act on behalf of a mobile consumer. These two approaches are namely: (i) proactive approach – prefetching subscriptions to neighboring proxies and (ii) reactive approach – notifying the proxy before detaching, both based on subscription proxies at the edge of the network.

Producer mobility is more difficult to achieve in PURSUIT. This is due to the fact that the concept of a Topology Managment System (TM) results in overhead to monitor the overall network topology. To provide suitable delivery information, the new network location of the producer needs to be updated within the TM system periodically. Additionally, the architecture document (cf. [145]) describes the option to extend the functionality of TM to support *mobility prediction* to respond quickly to changes to the delivery path while a producer is moving.

Table 4.3: Comparison of Mobility capabilities in the presented ICN architectures.

Mobility	consumer & producer	seamless connectivity	multi-channel	multi-homing
PURSUIT	✓ [145]	✓ [145]	✓ [156]	✓ [156]
NetInf	✓ [146]	✓ [146]	✓ [146]	✓ [146]
MF	✓ [149]	✓ [149]	✓ [148]	✓ [148]
COMET	✓ [157]	✓ [157]	n.i.a.	✓ [150]
CCN	● [12]	✓ [12]	✓ [12]	✓ [12]
NDN	● [13]	✓ [13]	✓ [13]	✓ [13]
Convergence	● [103]	● [103]	n.i.a.	✓ [158]
COMBO	✓ [154]	✓ [12]	✓ [12]	✓ [12]
GreenICN	✓ [155]	✓ [155]	✓ [155]	✓ [135]
RIFE	✓ [117]	✓ [117]	✓ [159]	✓ [159]
POINT	✓ [117]	✓ [117]	✓ [159]	✓ [159]
UMobile	✓ [143]	✓ [143]	✓ [143]	✓ [160]
Bonvoyage	✓ [161]	✓ [161]	✓ [162]	✓ [161]

Additionally, the TM can also report multiple paths of published information objects (e.g., directly by the producer or any in-network cache) to provide multi-homed capabilities as well as support multi-source/multi-path transfer with respect to multi-channel capabilities.

NetInf: In NetInf, mobile participants benefit from two options to update the bindings between object names and locations dynamically: (i) a *Late Name Binding* (LNB) strategy within the Name Resolution Systems (NRS) and (ii) dynamic updates of the NetInf layer routing information within the routing system. Both options update the current location of the information objects, and thereby, foster soft handover procedures to achieve seamless connectivity. Multi-homed information objects are registered with multiple bindings in the NRSs.

MobilityFirst: Mobility defines one of the core aspects of MF. The architecture separates naming from addressing, and thus, supports *Late Name Binding* as well. Such feature is used to update the bindings at the local/global name resolution system (GNRS) as late as possible or periodically. In this case, the flat *global unique name* (GUID) of an entity can be resolved to a network address at different access points along the route.

Furthermore, the combination of the MF naming strategy and the *Late Name Binding* functionality opens the door for multi-homing and multi-channel capabilities. A producer can specify multiple network attachment points (e.g. in form of <GUID:NA1,NA2>) as well as set communication policies (e.g., 'send to both', 'send to any') during the registration of an information object at the *Global Name Resolution System*.

COMET: The architecture introduces specialized Content Aware Router (CAR) components placed at the edge of the networks to support user mobility. During mobility, based on the connection to previous access points, the COMET network layer predicts future locations of the participants. Its *Content Forwarding Plane* is responsible to provide multi-path selection, if there are nearby multi-homed copies of an information object.

CCN/NDN approaches: Regarding the CCN/NDN based approaches, capabilities such as support of mobility, seamless connectivity as well as multi-homing are inherently supported via as a result of the loosely coupled communication model and fulfill the requirements. However, consumer and producer mobility is supported to different degrees. Consumer mobility is realized on application layer by re-transmitting Interest packets after the user device moved to the next access point.

The same behavior can be used by a consumer if link failures occur by joining and leaving the network while they are moving. Producer mobility defines a challenging task. By default, a data response follows the reverse path of a consumer's request. This challenges producer mobility because dynamic updates of forwarding information are required within the disseminating routers. GreenICN introduces an extension of its architecture called *On-Path Resolver Architecture* in which a producer sends location updates after moving to the next access router. A similar approach is used by COMBO based on [163]. Due to the fact, that the Convergence project defines a stateless communication as one of the main objectives, it is difficult to provide seamless connectivity while not maintaining information on the ongoing communication.

Regarding multi-channel capabilities, CCN and NDN define an abstract communication layer independent of a certain underlying link technology, and therefore, facilitate multi-channel communication. The same requesting packet can be send out by the ICN layer to multiple available link layers (e.g., WiFi and cellular) concurrently to improve reliability of data delivery especially in combination with multi-homing. This architectural design is also valid for the subsequent approaches based on CCN.

POINT and RIFE: Similar to the principles of PURSUIT, consumer and producer mobility in POINT is managed by at least one TM component. Additionally, the AP serves as a *Mobile Access Gateway* (MAG) performing mobility actions on behalf of the end-user device. For example, if a MAG detects the movement of a consumer or producer device, it notifies the TM component which sets up or modifies the required routing state in the ICN core network. Furthermore, POINT supports multi-path and multi-homing capabilities as part of a Mobility-based Proactive Multicast (MPM) model within the APs [159]. Due to the fact that RIFE deploys the results of the POINT project, the mobility management is also handled by the APs and the TM components [141].

UMobile: One of the major focus area in UMobile is the support of information sharing in case of disaster scenarios. In order to provide such functionality, ad-hoc communication between mobile devices is required, e.g., to foster communication between citizens and rescue team, while infrastructure components are occupied or unavailable. Based on such scenario requirements, UMobile introduces a name-based replication system (NREP) system [160], in which data items are replicated using prioritization rules integrated within the name of the item (e.g., lifetime or spreading ranges of a certain geo-area). Such system facilitates mobility support by increasing the availability of data item. Furthermore, the project defines a *connectivity manager* component as part of the architecture. It is responsible for managing multiple network interfaces as well as to select the most appropriate interface to maintain a constant network connection [143].

Bonvoyage: The support of mobility describes on of the major goals in Bonvoyage. The support of mobility is encapsulated by the introduction of proxy components for both consumer and producer. These components provide a common interface (offered by the Internames Service Layer) for data consuming and producing entities in the architecture. Entities communicate with the core network using these proxies which are able to aggregate, store and offer data items on behalf of the entities in case of connectivity losses [161].

Summary: The topic of mobility defines one of the core elements of the ICN paradigm and is thus present in all introduced approaches. The functional scope of each architecture is similar and they only differ slightly with respect to producer mobility. The paper [164] describes an extension for CCN to realize global mobility including mobile producers. Towards NDN, there are some producer mobility solutions available. For example, Meisel et al. [9] describes a *Listen First, Broadcast Later* (LFBL) forwarding protocol to support producer mobility in wireless adhoc networks.

Routing, Caching and Transport of Data

ICN as the underlying network introduces new challenges with respect to (i) *routing*, (ii) *caching* and (iii) *transport* of data. Especially, when it is to be expected that most of the communication participants are highly mobile. The following metrics are used to compare the introduced ICN approaches: (i) *reactive and predictive routing*, (ii) *proactive and reactive caching*, the flexibility of the approach with respect to (iii) *caching nodes* and (iv) the *routing strategies*, as well as the (v) *extensibility of the protocol* and the support of (vi) *flexible data chunk sizes*. The results of this comparison is illustrated in Table 4.4.

PURSUIT: Name resolution, data routing and transport are strictly decoupled from each other in PURSUIT. After the RENE resolves the name of a data object, the optimal delivery path is handled via the TM components taking into account policies and current network conditions. During the delivery of data, special network nodes (so-called *forwarding nodes*) cope with dissemination strategies and transport profiles. While such strategies are pre-defined, transport profiles can be bound to one or multiple communication interface and can be dynamically configured in a host.

Regarding PURSUIT's caching capabilities, forwarding nodes are responsible to decide to cache data reactively on the path. Furthermore, PURSUIT defines additional caching components acting as publishers within the network to provide caching pro-actively. In principle, the dedicated cache components act like pub/sub participants. However, PURSUIT does not specify functionality for mobile devices acting as cache component for other network participants, for example other vehicles.

With respect to a flexible chunk size, PURSUIT does not match the requirement. Each chunk is set to a fixed size, depending on the underlying link technology, for example Ethernet or Ethernet Jumbo frames.

NetInf: The approach introduces a *convergence layer* to support multiple routing capabilities of underlying networks. By default, a *route-by-name* strategy is used to route data on a per hop basis and realized via the information of the global/local NRSs.

Table 4.4: Comparison of the routing, caching and transport capabilities in the presented ICN architectures.

Network Management	reactive routing	predictive routing	proactive caching	reactive caching	flexible cachable nodes	flexible routing strategies	flexible chunk size	extensible protocol format
PURSUIT	✓ [145]	● [145]	✓ [145]	✓ [145]	● [145]	● [145]	✗ [156]	n.i.a
NetInf	✓ [165]	✓ [165]	✓ [146]	✓ [146]	● [165]	✓ [165]	✓ [165]	● [165]
MobilityFirst	✓ [149]	✓ [149]	✓ [149]	✓ [149]	● [149]	✓ [148]	✓ [147]	✓ [149]
COMET	✓ [157]	✓ [157]	✓ [157]	✓ [157]	● [157]	✓ [157]	n.i.a	✗ [157]
CCN	✓ [12]	✓ [12]	n.i.a.	✓ [12]	● [12]	✓ [12]	✓ [166]	✓ [166]
NDN	✓ [13]	✓ [13]	✓ [167]	✓ [13]	● [12]	✓ [168]	✓ [169]	✓ [152]
Convergence	✓ [158]	● [158]	✓ [103]	✓ [103]	● [153]	✓ [158]	✓ [153]	● [170]
COMBO	✓ [12]	✓ [12]	✓ [154]	✓ [154]	● [154]	✓ [154]	✓ [166]	✓ [154]
GreenICN	✓ [171]	✓ [171]	✓ [172]	✓ [172]	● [172]	✓ [171]	✓ [172]	✓ [155]
RIFE	✓ [141]	✓ [141]	● [141]	✓ [141]	● [141]	✓ [141, 173]	n.i.a	● [173]
POINT	✓ [173]	✓ [173]	✓ [173]	✓ [173]	● [145]	✓ [173]	n.i.a	● [173]
UMobile	✓ [116]	✓ [116]	✓ [143]	✓ [116, 143]	✓ [116, 143]	✓ [116]	✓ [116]	✓ [116]
Bonvoyage	✓ [162]	✓ [162]	✓ [162]	✓ [162]	● [12]	✓ [162]	✓ [169]	✓ [152]

On the other hand, routing is done via a routing protocol based on the late binding feature, and thus, without any NRSs support in the network. Special NetInf domains can define their own routing and forwarding strategies, and thus, support some kind of predictive routing ability by introducing intermediate NetInf nodes along the path to the destination.

The general, *NetInf Cache* is defined as an extension which can be present within a forwarding node as well as in NRSs, except original publisher and subscriber. Such decoupling offers the possibility to cache data both reactively and proactively using collaborative caching strategies. However, caching in NetInf is limited to in-network components which prevents consumer or producer of data to act, for example as carrier of data into other locations (e.g. data mules).

Facing different sizes of information objects, NetInf provides a flexible chunking of data which can be either performed on application or NetInf layer. This is different with respect to the protocol extensibility. In principle, protocols can be adopted to NetInf using the so-called convergence layer. However, there is no information available about the effort to create such protocol binding.

MobilityFirst: Within its Content Routers, MF provides a routing layer supporting both name-based (consulting GNRSs to resolve GUIDs hop-by-hop) and address-based (GUIDs are resolved to host addresses) routing proposed on a conditional routing behavior hop-by-hop strategy. The Content Routers route data based on actual network addresses (e.g., IP addresses) and can consult local/global GNRS(s) on the delivery path to resolve the destination of the publishers GUID. This option of re-consulting the resolution components hop-by-hop is described as late-binding and useful for mobile destinations.

Regarding caching capabilities, each CR acts as a cachable network component by storing incoming content (equipped with an expiration time) and forwarding it to the next hop. Furthermore, the content will be cached and replicated for future use or marked for opportunistic delivery, for example in DTN scenarios. Additionally, a producer can specify routing policies (e.g. "send to both", "send to any") to influence the routing at the edge of the network during the registration of data. However, there is no information whether such policies are flexible, or pre-defined within the MF protocol. Based on hop-by-hop routing, MF copes with different packet sizes of the underlying link layer by providing a flexible chunk size.

COMET: The core component of COMET defines a *Content Mediation Plane* which mediates information objects between the network providers and the servers. By offering a *coupled* and *decoupled* architectural design, COMET is able to route information in both a content-centric and a host-centric fashion. The mediation plane is built up on functional blocks providing functions such as *content resolution*, *path management*, *content mediation* and *content-aware forwarding*. Such function blocks are available within its mobility-aware CARs.

To exchange function block information, COMET uses a variety of protocols. Regarding in-network caching capabilities, the approach describes two strategies to cache data: (i) *probabilistically* caching within the CARs and (ii) centrality-based caching where additional cache components are placed within the network infrastructure. Based on responded traffic and resource capacity information of CARs, COMET can decide optimal placement of cached information in the network.

CCN/NDN approaches: Routing is based on three data structures within the CCN/NDN based approaches: (i) lookup cached information objects within a Content Store (CS), (ii) check Pending Interest Table (PIT) for issued Interest packets and (iii) forward Interest packets towards potential sources related to pre-defined rules as part of a Forwarding Information Base (FIB). During the transport of data, all approaches pursue a flexible hop-by-hop fragmentation strategy. Additionally, NDN defines a fourth structure: Forwarding Strategy Module (FSM) which offers capabilities of dynamic routing. Such module offers a router to decide whether to route (also predicted) or drop a packet in certain situations, for example after detecting failures or broken links. Subsequent approaches added similar strategy modules or components within their design. Convergence describes a central Routing Information Base which can be consulted or used to update FIBs by network nodes, if there is no matching entry within their FIBs. GreenICN places additional nodes within the network to resolve the most suitable route described in its *On-Path Resolver Architecture*. This architecture introduces a new packet type (the Binding Update packet) to update bindings within network nodes for new content locations.

In-Network caching in CCN/NDN is done within the CS as part of the network nodes. The architectural design supports reactive caching within all nodes providing a CS. Proactive placement of content at certain nodes is harder to achieve due to the pull-based communication model of CCN/NDN. One possible solution is to issue caching Interest packets to other caching nodes, e.g., based on a certain strategy. However, such approach results in additional round trips. Subsequent approaches try to minimize the issue by providing additional strategies or solutions such as multiple registration of information object to the name resolution system (NRS)s in Convergence or introducing a cache controller component as in COMBO.

POINT: The POINT architecture consists of three core functions, similar to the PURSUIT architecture: the rendezvous function (matching consumer and producer), a topology manager function (to resolve delivery paths) and forwarding nodes. Forwarding nodes are managed by the TM system using principles from the Software-Defined Networking (SDN) paradigm. SDN provides features to monitor, control and route packets in the network on a packet-by-packet basis. As a result, the architecture is able to react to network changes as well as to optimize the traffic flows actively in the network.

Protocol translations between the IP and ICN world are performed by the APs. While the protocol used to (re)-configure network router could be extended in the future, there is no information available about the extensibility of the data exchange protocols.

Regarding POINT's caching capabilities, forwarding nodes are responsible to decide to cache data re-actively on the path, similar to PURSUIT. In principle, the SDN functionality can be used to modify delivery paths and to bring data to cache nodes proactively. However, there is no information available about the options to influence the cache decision strategies remotely.

RIFE: While RIFE deploys the IP and ICN developments of the POINT project, it is different when looking into the forwarding layer. It provides an additional layer of abstraction – the ICN communicator – to be agnostic to the underlying ICN approach [141]. Furthermore, the project also focuses on opportunistic networking by supporting DTN functionality. It features SCF mechanisms by offering storage capabilities at the network devices. Additionally, RIFE introduces a pull-based edge caching architecture used to store valuable data items in appropriate nodes in the network actively [141].

UMobile: Besides the extension of the naming principles of NDN architecture, UMobile also extended the routing and forwarding mechanisms. In addition to the standard NDN and the previously introduced NREP mechanisms, the architecture provides other forwarding strategies and pipelines supporting delay-tolerant and opportunistic information exchange. The forwarding pipelines are accessible through powerful programming interfaces and dependent on the application [116]. Flexible forwarding of data is managed by the underlying connectivity manager, taking responsibility of efficient data dissemination across multiple forwarding pipelines and thus network interfaces. While such forwarding layer offers a degree of flexibility regarding the protocol used, it also poses challenges in extending formats such as maintaining programming interfaces.

With respect to in-network caching, UMobile also extended the functional scope of network storage. Based on the extension to support *push-based* information exchange, the architecture introduces *migration platform* for services and data items, e.g., to be actively placed in the network [143].

Bonvoyage: In Bonvoyage, two logical components are used to manage routing and forwarding information in the network: the NRS and the routing resolution system (RRS). While the NRS is responsible for intra-domain routing (e.g., within a core network running NDN), the RRS is responsible for inter-domain routing (e.g., between an IP-based and NDN-based network). Routing operations are handled by the Internames Service Layer present at each router in the network. The service allows the flexible optimization/modification of bidirectional delivery flows. Since Bonvoyage is based on principles of the NDN architecture, such service provides an alternative to the reverse path forwarding of NDN. Furthermore, the Internames service also provides a standard routing behavior which is backward compatible with NDN.

In Bonvoyage, in-network caching is based on the capabilities of NDN. Since the NRS is responsible for discovering routes, knowledge such as available data as well as the topology of the entire system can be used to plan optimal placement of data within the caches of the routers [162].

Summary: Routing, caching and transport capabilities describe a broad field of comparison. Regarding this group of requirements, all approaches provide capabilities to cope with effective routing and in-network caching. Differences have been shown in the feature sets of the recent generation of ICN architectures namely UMobile, POINT and Bonvoyage providing mechanisms to flexible react to network changes due to mobile participants compared to the other approaches. Furthermore, differences have been seen in features such as flexibility of cache components after deployment and the extensibility of the transfer protocol.

Safety & Security

Minimizing the occurrences and consequences of accidents and traffic collisions are main goals in the automotive safety domain. Progress towards the networking of vehicles can only achieve effective safety features if there are security properties involved as well. In the following, the introduced ICN approaches are compared to each other based on their safety and security related capabilities, such as (i) authentication, (ii) authorization, (iii) data integrity and (iv) the support of QoS mechanisms. Table 4.5 provides an overview of the supported safety and security features.

PURSUIT, POINT and RIFE: The architecture supports flat names (e.g., the object's hash value) which also permit self-certifying names for immutable data objects. From a security perspective, the *scope* mechanism can be used to enforce boundaries for access rights. Since POINT and RIFE are based on the concepts of PURSUIT, this also applies to these project as well.

With respect to data integrity, PURSUIT provides encryption and signing mechanisms at packet level which can be evaluated on both network (e.g. forwarding node) or destination (e.g., the subscriber application) to identify non-authenticated producer of data.

Moreover, the approach extends its topology management by including *traffic engineering* features such as QoS to meet the needs of network operators and users. Within these extensions, QoS dissemination strategies can be applied and operated in the forwarding nodes along the transport path. One example for such an extension is described in the POINT project [174]. To protect the system against DoS attacks, POINT defines encryption and rotation mechanisms for the forwarding information within the AP components [175].

NetInf: Security in NetInf is described as a modular building block system by providing pluggable security services to the NetInf layer. The building block system distinguishes services with respect to authentication, authorization, naming security and content integrity. Furthermore, a *Provenance* service allows entities to trace overall actions upon a set of objects within ICN.

MobilityFirst: Due to the name-based service layer in MF, flat labeled GUIDs can directly be derived from public keys to be a cryptographically verifiable identifier (based on the *Accountable Internet Protocol* (AIP) [176]) and thus to improve trustworthiness and provide traffic accountability.

During the registration of data, a *name certification service* (NCS) is consulted to bind human-readable names to a GUID securely. Within the GNRSs, the GUIDs are securely mapped to network addresses. Since the architecture is based on the AIP protocol [147], it inherits some security features such as traffic accountability. The concept of decentralized name certification services enables end-users to choose which NCS to trust. Furthermore, quorum-based techniques can be used by NCSs to identify and block untrusted NCSs within the network.

Table 4.5: Comparison of Safety & Security capabilities in the presented ICN architectures.

Safety & Security	authentication (self-certification)	authorization (access control)	data integrity	support other QoS mechanisms
PURSUIT	✓ [145]	✓ [145]	✓ [145]	● [145]
NetInf	✓ [165]	✓ [165]	✓ [165]	✗
MobilityFirst	✓ [147]	✓ [147]	✓ [147]	✗
COMET	✓ [150]	n.i.a.	n.i.a.	✓ [157]
CCN	✓ [151]	● [166]	● [166]	✗
NDN	✓ [13]	● [13]	● [13]	✗
Convergence	✓ [153]	✓ [153]	✓ [153]	● [177]
COMBO	● [12]	● [12]	● [12]	● [154]
GreenICN	✓ [155]	✓ [155]	✓ [155]	● [155]
POINT	✓ [145]	✓ [145]	✓ [145]	✓ [145, 175]
RIFE	✓ [145]	✓ [145]	✓ [145]	✓ [145, 175]
UMobile	✓ [13]	● [13]	● [13]	✓ [178, 116]
Bonvoyage	✓ [144]	✓ [144]	✓ [144]	n.i.a

COMET: The COMET architecture describes options to achieve some level of security by supporting self-certification of content names as well as standard third-party security infrastructure for authentication (e.g. within a Public Key Infrastructure). Each node in COMET acting as a producer of data is capable of authenticating their neighboring nodes by exchanging public keys.

Furthermore, the approach describes the ability to adapt security mechanisms from other ICN approaches. However, COMET does not provide further information in detail. Regarding additional QoS mechanisms, COMET follows an end-to-end *Class of Services* (CoS) approach which provides QoS in a multi-domain network. Focus of the CoS is to enforce content delivery paths and routing awareness to provide content delivery better than best-effort.

CCN/NDN approaches: The feature set of the CCN/NDN based ICN approaches are aligned to a *data-centric* security approach describing features to typical security requirements such as authentication, authorization and data integrity on packet level. The parent approaches, CCN and NDN, provide a per packet signature that can be used by applications for authentication or authorization.

The same applies to CCN/NDN applications to check validity and integrity of data. Subsequent approaches provide additional instruments. In Convergence, QoS mechanisms are described as information-based quality of service. Based on the name of a certain resource, network nodes can decide to serve data on certain performance criteria (e.g., some characters within the name indicating a high priority). In UMobile, security mechanisms are provided by the underlying network technologies such as NDN and the DTN implementations [178]. However, additional responsibility regarding authorization is supported by the network access gateways, which are under control of different authorities. By introducing a *Config* packet, the COMBO architecture is able to carry cache replacement policies and content pre-fetching management. Since Bonvoyage uses CCN/NDN principles, the architecture also supports all features of these approaches. Furthermore, the flexible concept of Internames allows to introduce additional name-based security mechansism, for example, from the the MobilityFirst architecture [144].

Summary: All described architectures support typical security related capabilities, such as authentication, authorization and data integrity or offer possibilities to add such support. Some of them take additional requirements into consideration to meet special needs of users or network operators such as PURSUIT, Convergence and COMBO. Only the COMET approach provides an extensive feature list of QoS for content delivery.

Interoperability & Community

The last field of comparison examines mostly non-technical characteristics of the different ICN approaches. The following subsection compares the interoperability capabilities, namely (i) *interoperability to other ICN architectural approaches* and (ii) *legacy host-centric networks*, as well as (iii) the backing of *an active community* and the (iv) *availability of source code and its license model*. The results of the comparison are shown in Table 4.6.

CCN and NDN: Looking at CCN (here CCNx as the implementation of CCN) and the NDN platform, both approaches are based on the same concepts of a *type-length-value* (TLV) protocol format. However, full interoperability is not guaranteed because the protocol structure differs between the approaches, and therefore, requires forwarders to agree on packet handling actions. At the time of this thesis, there is a harmonization effort of the CCNx and NDN protocols driven by the Information Centric Networking Research Group (ICNRG) [179]. However, since Cisco acquired the CCNx platform from PARC, the harmonization efforts were stopped. CCNx in today's implementation is partially interoperable to host-centric networking in form of a forwarding daemon. Further research has shown that an additional control plane could solve this issue [180]. Regarding the research and development community it is to be noted that CCNx is very actively supported and does feature open source software (OSS) as well as commercial implementations.

NetInf: Regarding interoperability efforts in NetInf, the approach introduces a convergence layer which ensures compatibility to other ICN architectures as well as an interoperability with other URI-based protocols (e.g., Hypertext Transfer Protocol (HTTP)). NetInf is still actively supported but future development is questionable due to the end of the NetInf project in 2013. Finally, it offers different open source implementations using varying license models.

PURSUIT, COMET, POINT and RIFE: The COMET architecture lacks information regarding its compatibility to other ICN approaches, while PURSUIT describes options to achieve such compatibility based on its high-level API design. This also applies to the POINT and the RIFE project, since both adopt concepts of the PURSUIT architecture. While all of the architectures offer open source implementations, only COMET and POINT supports implementations for communication within host-centric networks. Since RIFE deploys the implementation of POINT [141], host-centric communication is also supported in this project. While, the the first two projects seem to be inactive since 2013, the latter ones are also inactive since the end of the projects in mid 2018.

Convergence and Bonvoyage: Due to the fact that Convergence and Bonvoyage have many similarities with NDN, it is assumed that both show some interoperability between each other and NDN. However, there is no information available. While both feature at least one open

Table 4.6: Comparison of the Interoperability & Community features in the presented ICN architectures.

Interoperability & Community	other ICN architectures	host-centric networks	active community	licensing
PURSUIT	● [181]	● [181]	✗ [99]	OSS [182]
NetInf	✓ [165]	✓ [165]	● [101]	OSS [100]
MobilityFirst	✗ [147]	✓ [147]	✗ [107]	OSS [107]
COMET	n.i.a	✓ [150]	✗ [110]	OSS [110]
CCNx	✗ [179]	● [12]	✓ [119]	OSS [119]
NDN	✗ [179]	● [183]	✓ [183]	OSS [183]
Convergence	✓ [170]	✓ [103]	✗ [104]	OSS [104]
COMBO	CCN [154]	● [12]	✓ [105]	✗ [105]
GreenICN	n.i.a.	● [155]	✓ [155]	n.i.a.
POINT	● [181]	✓ [173]	● [114]	OSS [184]
RIFE	● [181]	✓ [141, 173]	● [111]	OSS [185, 184]
UMobile	● [116]	✓ [116]	● [112]	✓ [186]
Bonvoyage	● [161]	✓ [162]	n.i.a.	n.i.a.

source implementation, they differ when it comes to compatibility with host-centric networking. Both projects, Convergence and Bonvoyage, directly include this feature via additional service layers [103, 161], NDN does only offer support partly in form of a named-data forwarding daemon, similar to CCN. On the other hand, NDN is very actively driven by a huge community while the future support of Convergence as well as Bonvoyage stays unclear.

MobilityFirst: MF is not compatible to any other ICN architecture but the authors in [147] show how it can interact with host-centric networks. While MF features an open source implementation, the last updates were made in 2014 which questions an active further development.

COMBO and GreenICN: The only two ICN approaches under investigation that do not offer directly accessible open source code are COMBO and GreenICN. While both feature an active community, the compatibility towards host-centric networking is rather questionable. While COMBO seems to offer some partial support due to its relationship with CCN, GreenICN does not give any hint regarding this issue.

UMobile: Based on principles of NDN as well as extending its implementation, UMobile seems to be compatible with this architecture. However, there is no other information available about other approaches. With respect to host-centric communication, UMobile provides a gateway concept in order to translate interest packets to HTTP requests and vice versa [116]. The source code is available on GitHub, however, the project seems to be inactive, since the end of the project in mid 2018.

Summary: Regarding this group of requirements, CCNx, NDN and NetInf seem to be most suitable. While NetInf scores in all relevant fields, CCNx and NDN stand out by their active and durable community.

The comparison of available ICN architectures of the past few years has shown differences with respect to the requirements of connected vehicles. On the one hand, all approaches are well suitable regarding mobility issues, while safety and security, naming and data dissemination reveal major differences. On the other hand, the *interest-based* approaches as first presented by Jacobson et al., namely CCN, NDN, Convergence, GreenICN, COMBO, UMobile, and Bonvoyage do fulfill most of the requirements. The other introduced approaches, namely PURSUIT, NetInf, MF, POINT are restricted by a fixed naming scheme or lack support for self-description. Furthermore, it has been shown that the feature sets of the recent generation of ICN architectures, namely UMobile, POINT and Bonvoyage provide mechanisms to meet requirements of mobile participants compared to the previous approaches.

Bringing all findings together, the *interest-based* family members seem to be the most suitable approaches for connected vehicle environments. From this group of seven, CCN and NDN stand out due to the huge community contributing to them which makes future commercial success much more likely. This group of ICN architectures is followed by MF and NetInf which both have drawbacks regarding naming, routing and transport as well as the level of support within the ICN community.

4.2 Information-Centric Networking for Connected Vehicles

As stated in the previous section, the flexible, decentralized nature of the *interest-based* ICN architectures (e.g., CCN and NDN) is promising for ad-hoc networking such as connected vehicular systems. The related work of ICN in the context of vehicular systems can be separated into three groups: (i) research of ICN core elements in connected vehicular systems, (ii) ICN framework solutions for vehicular systems, and (iii) work *surveying* ICN-based ad-hoc networks. Figure 4.1 provides an illustration of these groups and shows that most of the work is based on the core elements of the interest-based architectural approaches.

4.2.1 First Steps of ICN in Vehicular Systems

First proposals introducing named data principles in the context of vehicular systems were presented by Meisel at al. [9] and Wang et al. [226], in 2010. By introducing the advantages of a loosely coupled communication model, both publications highlight the flexible access to named content and advocate against a host-centric approaches. In the following years, several ICN frameworks have been proposed in the context of vehicular systems by introducing prototype implementations and extensions of certain ICN architectures. Examples are DMND [226], Information-Centric Networking on Wheels (IC-NoW) [225], CarSpeak [227], and Content-Centric Vehicular Networking (CCVN) [10].

Besides the first extensions of vehicular ICNs, research groups explored the suitability of ICN architectures for connected vehicles and described in detail the challenges and the options to bring ICN-based approaches to VANETs (cf. Figure 4.1, survey subtree). For example, the authors of [243, 129, 130] provide surveys of ICN in the automotive context and introduce open research challenges regarding each of the core principles such as efficient *data dissemination*, as well as *naming* extensions to provide access to data in high dynamic ad-hoc networks. The topic of *mobility* support for both, consumer and producer of data, is brought into focus by the authors of Tyson et al. [133] and Zhang et al. [134]. These early research efforts formed the basis for the activities of vehicular ICN in the following years.

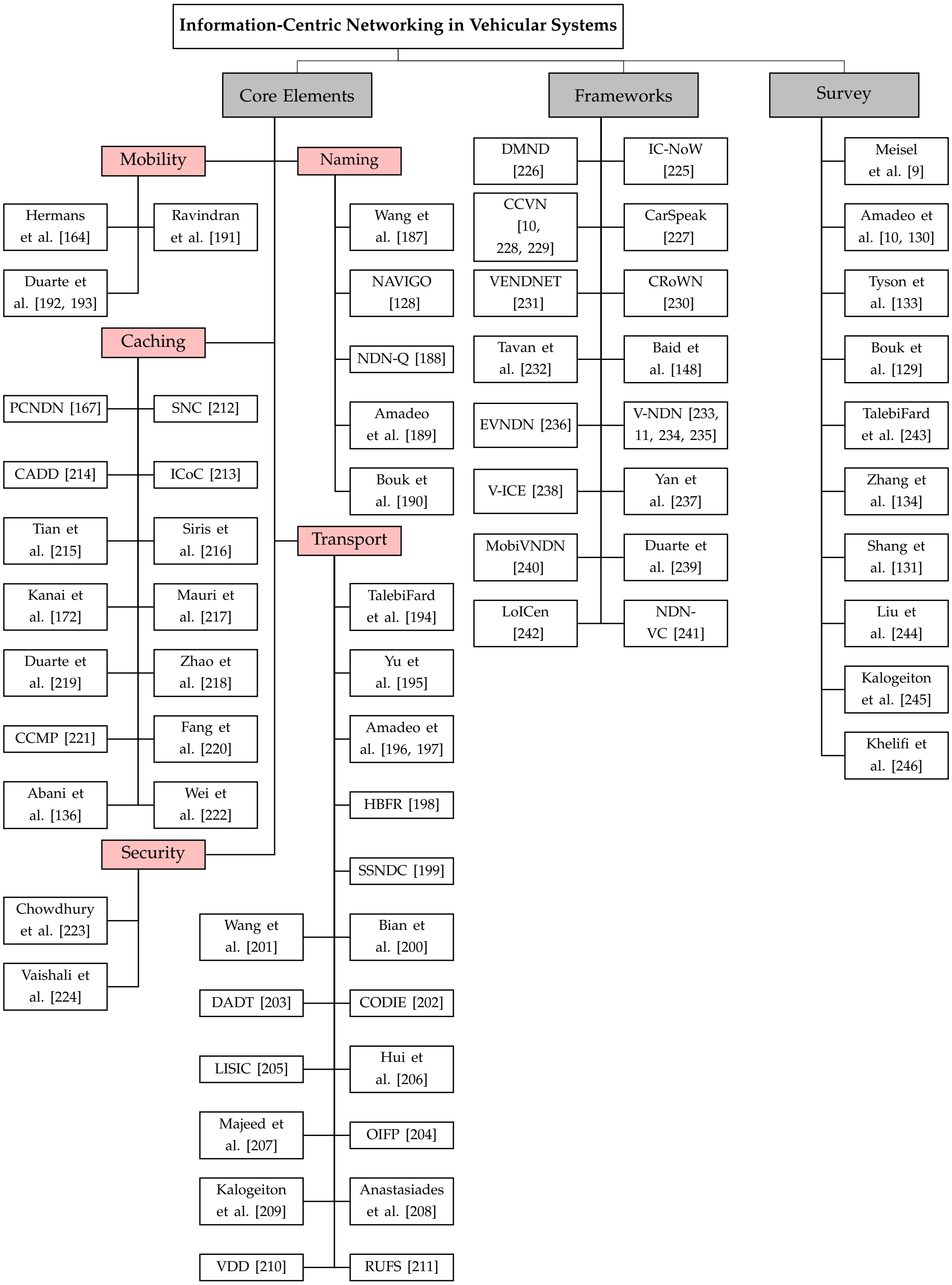


Figure 4.1: Recent research efforts of information-centric networking addressing challenges of connected vehicular systems. Publications with focus on at least two core elements are listed in the *framework* sub-tree, while literature *surveying* the topic are listed in the third sub-tree.

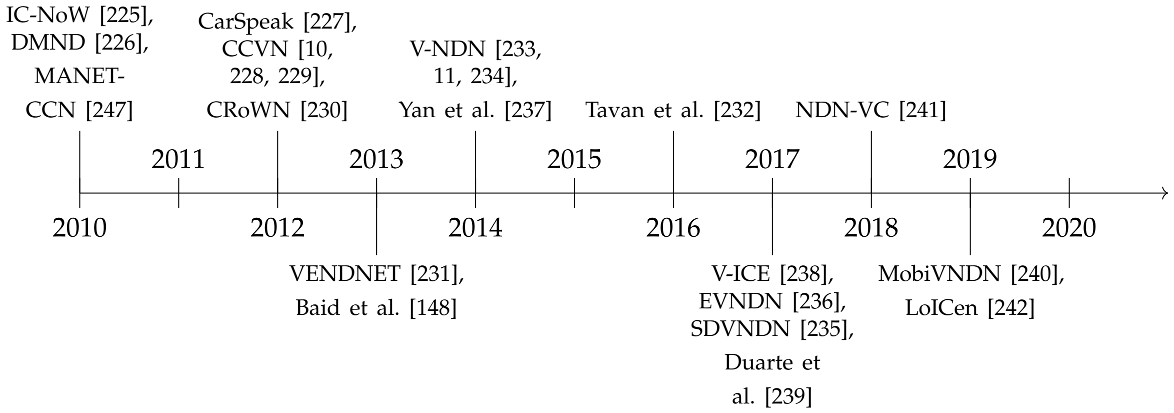

IC-NoW [225], DMND [226], MANET-CCN [247]
CarSpeak [227], CCVN [10, 228, 229], CRoWN [230]
V-NDN [233, 11, 234], Yan et al. [237]
Tavan et al. [232]
NDN-VC [241]
2011
2013
2015
2017
2019
2010
2012
2014
2016
2018
2020
VENDNET [231], Baid et al. [148]
V-ICE [238], EVNDN [236], SDVNDN [235], Duarte et al. [239]
MobiVNDN [240], LoICen [242]

Figure 4.2: Timeline of the proposed ICN-based vehicular frameworks over the past few years.

4.2.2 ICN-based Vehicular Frameworks

An analysis of the related work in ICN-based vehicular shows a large list of framework proposals over the past few years. Figure 4.2 illustrates the framework proposals in chronological order, starting with the first proposals back in 2010.

One major aspect outlined by Bai et al. [225] is the expected variety of medium access technologies (e.g., cellular or DSRC) present in vehicular ad-hoc networks. As a consequence, the authors demand for network technologies which are agnostic to the underlying access technologies by introducing the concept of IC-NoW. The framework uses naming schemes to access information in the network and presents open research directions such as efficient data dissemination via data aggregation and geo-specific replication, as well as data-centric security features.

Wang et al. [226] introduce NDN in vehicular networks which is one of the first work considering a full-fledged ICN architecture. Based on the principles of NDN such as hierarchical names or the intrinsic in-network caching, the authors show in their DMND framework the benefits of decoupling data from physical locations in vehicular ad-hoc networks. Similar to the previous work, the authors of Oh et al. [247] propose a scalable content-oriented framework (MANET-CCN) for mobile ad-hoc networks based on the principles of CCN.

The introduction of full-fledged ICN architectures in the context of vehicular systems has opened the way for further framework developments. Two prominent examples are CCVN and Vehicular Named Data Networking (VNDN), both based on the architectural concepts of CCN and NDN respectively.

CCVN: CCVN extends the concepts of the CCN architecture to cope with impairments in vehicular networks. The authors of Amadeo et al. [10, 228, 229] introduce routing and forwarding extensions for efficient data dissemination in wireless vehicular networks. For example, this includes mechanisms to avoid broadcast storms or to provide reliable content retrieval. The introduction of the *CRoWN* framework [230] has been a another big step towards ICN-based vehicular ad-hoc networks. Amadeo et al. propose an architecture providing content-centric networking capabilities directly on top of the IEEE 802.11p physical and medium access layer. Furthermore, the authors envisioned to deploy the CCN protocol stack in parallel to the WAVE stack as an enhancement for data-oriented communication.

VDND: Similar to the research of avoiding broadcast storms, Wang et al. [233] propose an efficient routing and forwarding scheme for NDN in which participants are characterized by a high degree of mobility. Other work in the context of VNDN is described by Yan et al. [237] which propose optimized naming schemes to provide access as well as to aggregate and distribute data efficiently in mobile networks. Besides the presentation of using in-network caching capabilities of vehicles to carry data from one location to another, Grassi et al. [11] present an implementation used for real world experiments.

The technological advances in cloud computing and data centers also push into the networking domain. This opens the way for enhancing the concepts of CCVN and VNDN, e.g., enhancing the VNDN framework by selecting valuable relay nodes to overcome mobility and data dissemination issues caused by poor signal quality [236] (EVNDN) or bringing cloud computing [241] (NDN-VC) and software-defined networking [235] capabilities into the VNDN framework.

4.2.3 Related Work of In-Network Caching in ICN-based Vehicular Systems

The last group of related work addresses ICN core elements coping with impairments in vehicular networks (cf. Figure 4.1 1st sub-tree). By analyzing the related work of this group, it can be seen that most of the work addresses challenges of routing, forwarding and transport of data items followed by caching and naming strategies. In this manuscript, the major focus lies on in-network caching strategies for ICN-based vehicular networks.

As presented in the previous chapter, in-network placement strategies are separated into groups of *reactive* and *proactive* caching (cf. Section 3.2.6). While plain interest-based ICN architectures such as CCN and NDN only support reactive data placement strategies by opportunistically caching data at nodes on the path to a destination, optimized strategies can increase the efficiency and the service quality of the vehicular network.

Reactive Placement Strategies

Regarding reactive placement strategies, a request made by a consumer triggers the mechanism on all nodes along the delivery path. By default, the forwarding behavior of interest-based ICN architectures such as CCN and NDN follows the reverse path of the request back to the consumer (e.g., [13]). While such behavior facilitates reactive placement strategies such as LCE or LCD, it may result in a lot of replicas within the network. Knowing the probability that certain data is requested by a number of consumers is a useful criterion to decide which data to be stored on-path. Over the last years, many placement strategies such as *ProbCache* [14], *WAVE* [15] or *Progressive* [16] have been proposed for static networks using input values for decision making such as content flow characteristics (e.g. ProbCache) or the popularity of data items (cf. Section 3.2.6).

The presented challenges motivate researchers to enhance reactive caching strategies by taking characteristics of mobile ad-hoc networks into account. Tian et al. [215] increases the availability of data items in vehicular networks by proposing an replication scheme according to the LCE strategy. Complex strategies include neighboring nodes (e.g., [214, 213, 220]), the traffic density (e.g., [219]) or different available communication relations such as V2I and V2V (e.g., [218]) into account by proposing cooperative cache algorithms to provide access to data items.

As part of vehicular ICN-based networks, reactive caching strategies have shown performance improvements such as decreasing delivery times or reducing the load in the core network, especially in terms of popular data items.

Proactive Placement Strategies

All the previous presented work trigger caching actions reactively, after consumers in the network have requested for information. When looking into mobile ad-hoc networks, it can be seen that delivery routes between the mobile network participants change constantly. Storing data reactively on these paths may result in inefficient delivery of these data items. Ideally, the desired information is already placed proactively at network nodes a consumer will be connected soon, e.g., apart from the delivery route. However, such strategy requires some degree of coordination and cooperation which can be separated into (i) *path coordination* - where data is cached at nodes nearby the delivery path such as WAVE [15] and (ii) *neighborhood coordination* - where data is cached at neighboring nodes and caching decisions are made locally (e.g. nearby the consumer or source node) such as *Controller-Based Caching and Forwarding Scheme* (CCFS) [248]. However, the number of available proactive caching strategies is rather low as this topic is addressed just by a few researchers in the ICN community.

In highly dynamic networks with a high degree of mobility, the placement of content in the network, matching the mobility pattern of nodes is a non-trivial task. Rao et al. [167] introduce a proactive caching mechanism to enhance user mobility in NDN called PCNDN. Before disconnecting from the network, the consumer notifies the network AP about the event. Based on a proxy approach, the AP responses and stores further incoming data on behalf of the consumer. Furthermore, the proxy notifies surrounding APs to speed up the reconnecting during the handover, and therefore, increase data delivery after process. However, PCNDN requires the introduction of a new packet type in NDN, a direct link between each APs as well as significant knowledge about the topology and management resources at each network node. Especially in connected vehicle use cases, such an approach faces scalability problems, where a node moves through large parts of the network.

Another strategy is introduced by Vasilakos et al. [212]. The authors present a selective neighbor caching algorithm used for determining the appropriate subset of neighbors by taking the mobility behavior of users into account. Similar to the previous approach, the strategy requires knowledge about neighboring nodes and the current state of their caches.

Kanai et al. [172] introduces a proactive caching solution for transportation systems according to a fixed time schedule. Data chunks are stored at train stations proactively by using the fixed time schedules of trains to easily predict the movement. Such prediction can not be adopted to mobile consumer in VANETs as they have a more complex behavior.

The authors of Siris et al. [249] introduce proactive caching mechanisms to support seamless mobility in mobile ICNs. By exploiting mobility prediction of mobile device users (e.g., smartphone users), the authors describe a collaborative caching mechanism to arrange data in the caches of the APs proactively. However, the challenges of connected vehicles (e.g. fast topology changes) are not taken into account.

In the context of connected vehicles, albeit without considering ICNs, the authors of [250] suggest placing demanded content at the edge of the network. With regard to ICNs, Mauri et al. [217] or Abani et al. [136] introduce centralized managed network components to calculate the optimal content placement at APs by using the knowledge about a subnetwork. The approaches have shown that data items are treated equally in the cache decision process.

Similarly, Khelifi et al. [251] propose a decentralized optimal caching scheme based on mobility prediction techniques. Based on a-prior knowledge about the road infrastructure, the approach resolves the next optimal network component (e.g., a RSU) to store a specific data chunk. However, the authors consider one class of applications (here: video streaming), while there are more traffic classes of applications in the context of connected vehicles.

4.3 Summary

This chapter has introduced ICN architectures proposed over the past few years. Based on the introduced automotive use case scenarios in Chapter 1.2, architectural requirements have been derived such as support of naming, mobility, caching, as well as the interoperability features against other approaches. A detailed examination and discussion of the different available ICN approaches regarding the specific requirements of connected vehicles have been presented. As a result, *interest-based* approaches such as CCN and NDN are matching most of the needs of connected vehicles such as mobility support, in-network caching capabilities, as well as are represented by an active research and development community.

Based on the findings, the second part of this chapter has provided a detailed analysis of the literature of *interest-based* ICN approaches in the context of vehicular ad-hoc networks. The related work has been separated into three blocks: literature addressing ICN core elements such as naming, routing, in-network caching, literature presenting frameworks providing solutions for several core elements and work surveying the recent advances of ICN in connected vehicle environments. While the related work has shown improvements such as network performance, security or increasing the availability of data in ICN-based vehicular ad-hoc networks, one group of in-network caching mechanisms still remains a challenge: *bringing data proactively to mobile consumers*. The following chapters close this gap by investigating and presenting novel mechanisms in ICN-based as well as computation-centric based connected vehicle environments.

5 Proactive Content Placement in Connected Vehicle Environments

> Don't approach life's challenges by being *reactive*. Be *proactive*. Prepare for the possibilities before they arrive.
>
> Stephen Covey

The assistance in disseminating and storing data actively in the network is a promising approach to improve the network performance. As presented in the previous chapter, information objects are stored at least reactively at intermediate nodes during the delivery in interest-based ICN vehicular networks. Especially for popular data (e.g., latest news), reactive caching have shown network performance improvements while reducing the delivery time. However, reactive caching strategies have their limitations, for example in networks characterized by frequent changes of the data delivery routes. While *proactive* caching schemes have been proposed in recent years to overcome the limitations of reactive caching, the strategies focuses on one specific class of applications in connected vehicle environments.

As part of this chapter, novel *proactive* caching approaches are introduced based on the Named Data Networking architecture. First, the problem statement is presented. Second, an analysis of automotive data traffic classes which benefit from being placed proactively in the network are identified (cf. research question Q1.1 in Section 1.4.1). Based on the results of the analysis, different caching strategies are presented as part of a proactive caching framework to improve the delivery of information objects in interest-based vehicular ICNs networks, and thus, overcome its limitations (cf. research question Q2 in Section 1.4.1)[8].

5.1 Problem Statement: Mobile Node Delivery Problem

From a network perspective, the exchange of information in a NDN is achieved by using two packet types – `INTEREST` and `DATA` . Figure 5.1 illustrates the structure of the `INTEREST` and `DATA` packet according the the packet format specification v0.3 [253]. When using the standard NDN forwarding mechanism, a `DATA` packet is forwarded the exact reverse path an `INTEREST` has taken through the network [13]. In fast changing networks such as connected vehicle environments, a reverse path forwarding strategy may result in an undeliverability of a packet. An exemplary presentation of a data exchange describes this in detail.

During the journey, a vehicle is interested in receiving a map tile as part of an online navigation application (cf. Figure 5.2). It sends out an `INTEREST` packet to an AP nearby querying/pulling for such information using a name (e.g., `/koblenz/map/123`). When the `INTEREST` is received by the AP, the packet is processed according to the forwarding implementation of NDN (cf. Figure 5.2, `INTEREST` processing plane). As part of this example, the AP is not able to find a copy of the desired object in its cache. Before the `INTEREST` is forwarded into the

[8] The work in this chapter is published in the proceedings of the 2016 IEEE Vehicular Networking Conference [18], the 2018 IEEE Vehicular Technology Conference VTC-Spring [19], and the 2018 European Conference on Networks and Communications [252]. Parts of it are extracted from these sources.

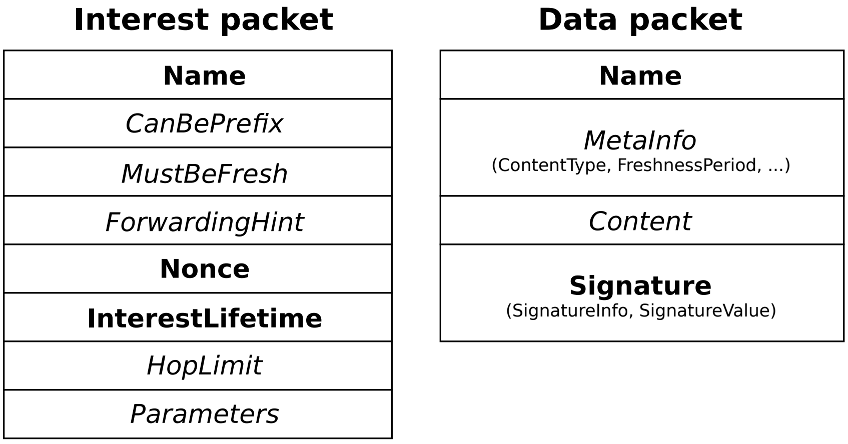


Figure 5.1: Illustration of the `INTEREST` and `DATA` packet formats v0.3 of the NDN architecture based on [13]. Bold fields are **mandatory**, italic fields are *optional*.

upstream network towards any node providing the information, the AP creates an entry in its PIT to keep track of Information Objects. Afterwards, the request is forwarded hop-by-hop through the network until it reaches a node that provides the desired information, according to the FIB entries. That node may be the original source, or any network member that has a vital copy of the data in its cache. If there is a node providing the desired information, it replies with a corresponding `DATA` packet which has the same name as the `INTEREST` packet (cf. Figure 5.3).

When a node responds to the `INTEREST` packet, the corresponding `DATA` is forwarded hop-by-hop the exact reverse path back to the consumer as the `INTEREST` packet has taken through the network (symmetric information exchange). This is achieved by following the PIT entries of each node. Once the packet is forwarded to the AP, which overhears the initial `INTEREST`, it is ready to deliver the `DATA` to the consumer (cf. Figure 5.3, Data processing plane). However, the vehicle may have lost the connection to the AP and hence may not able to receive the `DATA` (cf. Figure 5.3, lightning symbol). As a result, it has to repeat the `INTEREST` at the next AP. In the worst case, a consumer does not receive the `DATA` packet in time, which may result in unreliable function execution of connected vehicle applications. In this thesis, the issue is defined as the *mobile node delivery problem*.

Looking into the *reactive* caching capabilities of ICN, the desired `DATA` packet has been already cached at an intermediate node or at the next access point. This might be the case in some scenarios, for example the desired information is popular and of interest to a larger group of consumer (e.g., road condition or latest news), or passing car has requested the same information recently. But there are also scenarios in which reactive caching is not efficient or has no effect at all. For example, storing personalized information which is of interest for a specific consumer is not efficient when cached reactively in the network. It is expected that the problem statement is valid for mobile scenarios in which participants are moving with high velocity (e.g., motorway), however, it is also expected to be valid for ICN-based vehicular systems in rural regions characterized by a sparse infrastructure deployment.

In order to understand the full impact of the mobile node delivery problem, an analysis of different automotive data traffic classes is required.

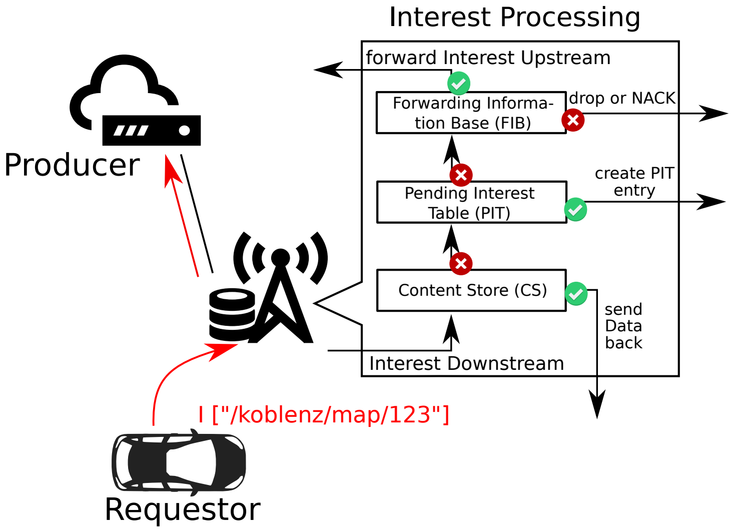


Figure 5.2: During the journey, a mobile node sends out an `INTEREST` packet to an infrastructure node asking for map tile as part of an online navigation application. The forwarding plane of the Named Data Networking architecture is processed at theAP according to [13]. After the `INTEREST` is processed it is forwarded upstream towards any node providing the corresponding `DATA` packet.

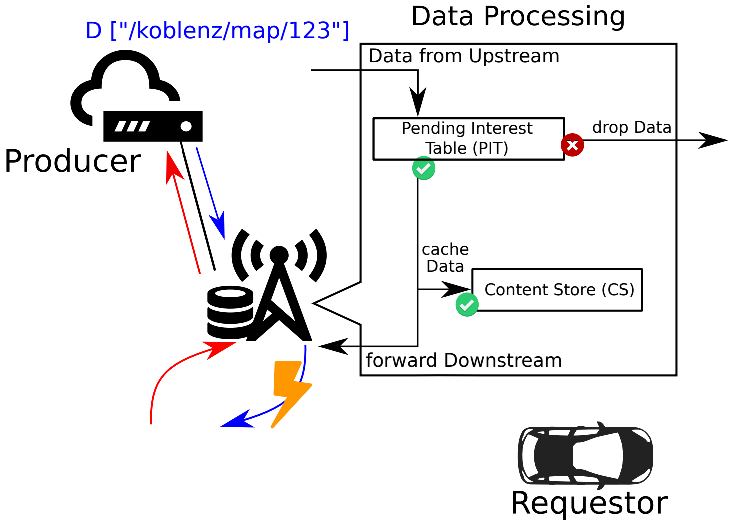


Figure 5.3: A `DATA` packet is forwarded the exact reverse path the `INTEREST` has taken through the network, but may not reach the consumer. Due to the reverse path forwarding, a mobile node may not able to receive the `DATA` packet in time in worst case.

Table 5.1: Types of data transferred in connected vehicle environments.

Data	popular	personalized
transient (small)	road conditions	point-of-interests nearby
transient (large)	traffic updates	smart home events
static (small)	road construction notifications	personal address book
static (large)	high resolution map	firmware updates of the own car

5.1.1 Automotive Data Traffic Classes

From a network perspective, data in the automotive IoT can be classified in terms of its popularity, its size, and the duration of validity. Examples for different classes of data is given by Table 5.1.

Popular data is requested by many consumers. Examples are traffic updates, the latest blockbuster movie or high resolution maps of the vicinity. Placing such popular data at the edge of the network increases the availability to cars without generating traffic in the backbone. In contrast, personalized data is of interest only to a single participant in the network. Examples of personalized data is the personal address book, firmware and software updates for a car, or information exchange with the personal SmartHome environment.

A second important differentiation for data in vehicular use cases is its duration of validity. This thesis distinguishes between static and transient data. A static data item is valid for a longer time period. Examples are the static layers of maps or updates for applications and build-in controllers in the car. Transient data is only valid for a short time period. Examples include updates on environmental conditions, information about parking spaces or a picture from a security camera in your apartment.

A third important differentiation is described by the size of data. Small data (e.g., traffic updates) can be easily cached multiple times at the edge of the network, requiring minimal resources and is transferred in a short time. On the other hand, the number of copies for large data objects (e.g., maps) should be kept as low as possible in order to save memory on the nodes.

As long as data is popular, caching makes sense, no matter the duration of validity (static or transient) or the size of data. However, caching transient data is challenging, because cached data might already be outdated, when it is requested by the next node. Consider the example of a continuously updated state of a traffic light in a green wave application: It may be very popular, but an individual instance of the state is only up-to-date for a short time. In this case network performance can be increased by caching the latest data at the edge of the network actively.

Table 5.2: Contribution: Proactive caching strategies and their deployment scope regarding the presented automotive data traffic classes.

Data	popular	personalized
transient (small)	ADePt (cf. Section 5.4)	PeRCeIVE (cf. Section 5.3)
transient (large)	Predictive Prefetching (cf. Section 5.5)	PeRCeIVE (cf. Section 5.3)
static (small)	*reactive*	-
static (large)	*reactive*	-

5.1.2 Proactive Caching: Potential Solution Space and Challenges

As a potential solution to overcome the mobile node delivery problem in highly dynamic networks, caching strategies which place a consumer's anticipated content at the right network nodes in time are promising to increase the availability, and thus, the chances to deliver it in time. The mobile node delivery problem can be given as part of a formal definition as:

$$t_{all} = \sum_{h=0}^{Nhops} t_h \tag{1}$$

where t_{all} describes the total amount of time required to transfer a packet from the provider to the consumer, which is dependent on the transmission time between all intermediate network components, and thus, the total number of hops t_h. Ideally, the packet has been already stored closer to the consumer, which reduces the number of hops and therefore the total amount of time to receive the packet. However, the proactive placement describes a non-trivial task. The efficiency of such strategy is dependent on the characteristics of the requested data items. Table 5.2 illustrates the contributing proactive caching strategies with respect to the presented automotive data traffic classes (cf. Section 5.1.1).

The idea of placing data closer to the consumers proactively at network nodes has attracted researchers in academia and industry. Active management approaches have shown advantages, especially in networks in which participants are characterized by a high degree of mobility, e.g., reducing the hand-off processing times between a consumer and an access point (e.g., [17]). The use cases introduced in Section 1.2 will benefit from data which has been placed at edge nodes proactively (e.g., road condition information). According to the caching taxonomy elements in Section 3.2, there exist a number of open challenges towards proactive caching strategies in information-centric connected vehicle environments that need to be solved before it can hit the market on a large scale:

C1 **Scalability**: How do caching approaches scale when the number of participants and services within this system rise?

C2 **Deployment Strategies and Organization**: How could (potentially ad-hoc) deployment strategies of proactive caching mechanisms look like as well as the efficient organization of the nodes involved in the caching procedure?

C3 **Availability**: How to ensure the availability of services and data cached proactively in the network with the conflicting demands set by connected vehicles (e.g. latency)?

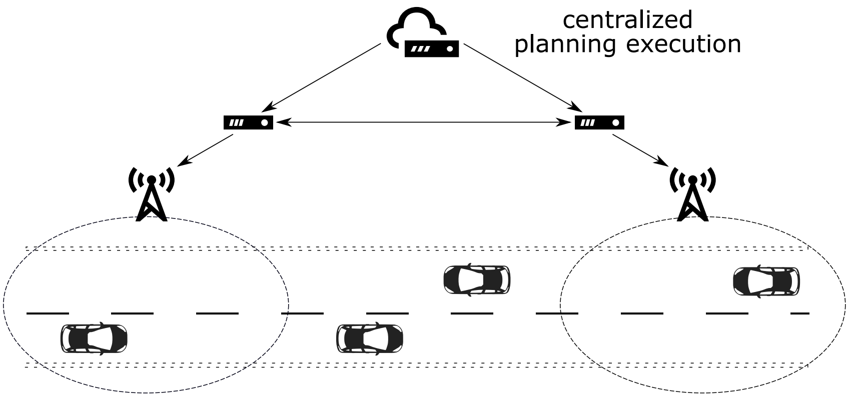


Figure 5.4: Exemplary structure of a centralized planning execution. Just a few high level nodes are involved in the planning process and instruct lower layer nodes to store data within their local storage.

C4 **Dissemination Strategies**: How to distribute and place information as well as caching strategies in a timely and efficient manner within the caches?

C5 **Security & Privacy**: How can security and privacy be ensured, for example to prevent misbehavior of content placement and to protect against security threats?

Application scenarios in which the number of participants vary over time (e.g., more participants in a certain geo-location during rush hours) require a **scalable** infrastructure which also include caching mechanisms. While the design of a caching system is relatively simple if the number of users and applications is rather low, it becomes difficult when both numbers rise. Planning the proactive placement of data can be a very complex calculation, for example by taking into consideration all the vehicles that could be present at an access point at different time periods.

Another challenge is described by the **deployment** (location) of caching strategies at nodes as well as the **organization/governance** of the nodes involved in the caching process. This includes the question of *where* to deploy and execute the placement planning as well as execution mechanisms. On the one hand, centralized/hierarchically managed planning approaches involve just a few centralized components (cf. Figure 5.4), however require knowledge about the subsequent network such as topology, available bandwidth, or the mobility pattern of mobile nodes. This may result in additional management overhead in the core network (e.g., [217, 136]). On the other hand, in decentralized managed approaches, nodes plan the placement by themselves locally or coordinate it with neighboring nodes in a distributed fashion, without the need of a centralized component (cf. Figure 5.5). However, local planning is associated with a degree of uncertainty, while not having access to the global state of the (sub)network (e.g., [212, 167]), and thus may result in a level of inefficiency (e.g., large number of duplicates in the caches).

In the automotive domain, **availability** of data and services defines a crucial requirement to ensure the functional reliability and safety of applications (e.g. automated driving). Data cached proactively to nodes closer to the consumer has to remain in the caches until it is retrieved by the consumer. Data *replacement strategies*, managing the local memory resource of a node, also plays an important role, however, it defines a challenging task.

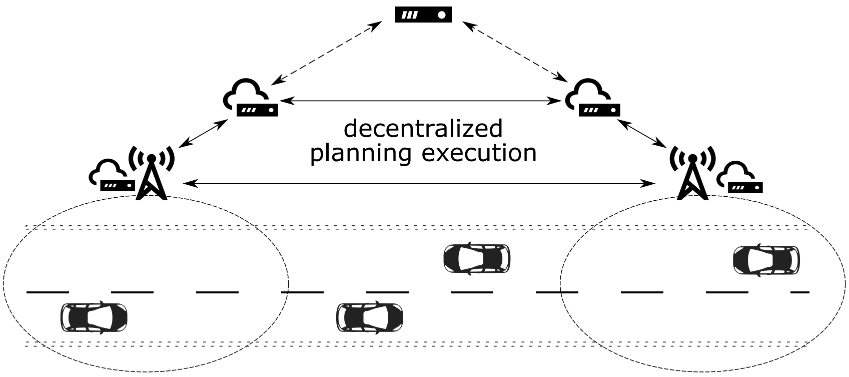


Figure 5.5: Exemplary structure of a decentralized planning execution. Each node operates the planning independently or collaborates with neighboring nodes in a distributed multi-tier cache system.

A potentially large number of participants demanding for various different content while freely join and leave the network.

The placement of data within the caches as well as the dissemination of caching strategies in the network require **efficient dissemination patterns**. Especially, the active placement of data within the cache of a certain node describe a challenging task in ICNs. For example, NDN [13] relies on a pull-based communication and reverse path forwarding pattern. Efficient push mechanisms reduce the time required to place content at network nodes in time. Furthermore, a multicast-push mechanism would allow to transfer data to multiple nodes simultaneously.

Especially in the automotive domain, **security and privacy** are important to protect passengers and people in the vicinity. The functionality of the overall application, the data of the application as well as the consumer's personal data have to be protected. This is especially challenging in distributed systems, when data might be cached anywhere in the network. Furthermore, the infrastructure need to be protected against attacks, for example, to prevent misbehavior of content placement and other security threats (e.g., DoS attacks).

5.2 Towards an Adaptive Framework for Active Content Placement in Information-Centric Connected Vehicles

In order to understand the full extent of the *mobile node delivery problem*, an analysis of the resolved request ratio – ratio of request sent and responses received at a consuming node – considering reactive content placement strategies is simulated. The scenario consist of three different traffic volumes in a motorway scenario (cf. Section 5.6), while caching is enabled at all infrastructure components (e.g., RSUs). The two default reactive placement strategies of the Named Data Networking architecture are evaluated, namely (i) LCE – every node creates a copy of a `DATA` packet, and (ii) ProbCache – every node creates a copy dependent on a probability value. Furthermore, placement and replacement strategies are coupled in NDN, the evaluation considers different replacement policies as well (e.g., LFU, LRU, FIFO, etc.). Figure 5.6 illustrates the results of the comparison of the default placement strategies. The best results are achieved for both placement strategies combined with the LRU replacement policy (cf. Table 5.3).

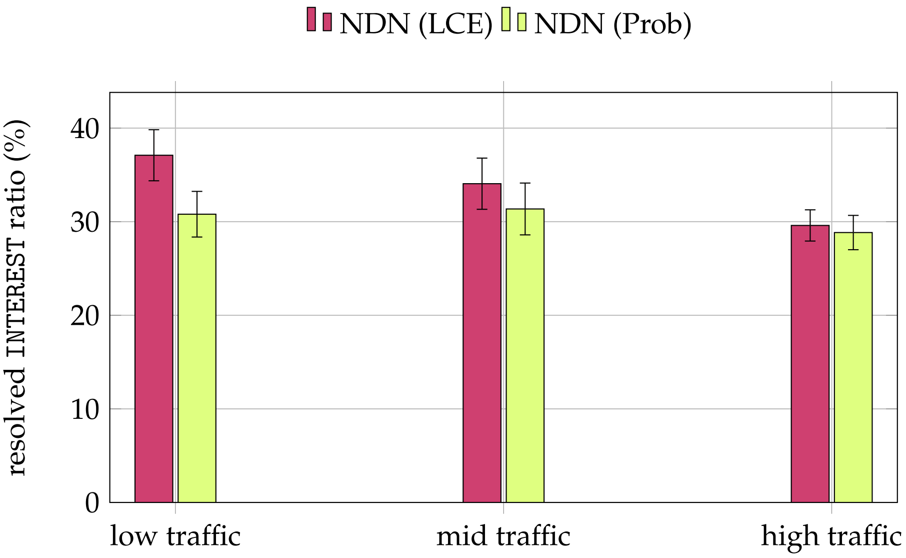


Figure 5.6: Comparison of reactive content placement strategies in NDN using the resolved `INTEREST` ratio metric. The results show that more requests are answered by the infrastructure if content is cached at any node (LCE placement) in combination with the LRU replacement strategy.

When looking into the results of resolved requests, the best values are achieved by replicating and storing the same content at every node on the deliver path (NDN (LCE): *low* ≈ 37%, *mid* ≈ 34%, *high* ≈ 29%), while storing the same content with a certain probability results in moderate values (NDN (Prob): *low* ≈ 30%, *mid* ≈ 31%, *high* ≈ 28%). However, the evaluation results have shown there is still potential to increase the resolved ratio by increasing the availability of Information Objects close to the consumers.

In this thesis, novel caching strategies will be introduced in order to overcome the problem of the *mobile node delivery problem*. These strategies are envisioned as part of a proactive caching framework, and will form the basis for it. The framework describes a vision of a modular and harmonized architectural design in which proactive caching strategies can by applied and executed at any node in the network dynamically. The framework supports to plug-in and execute caching strategies, providing the flexibility to be agnostic to cache any data traffic class in the network and supporting both centralized- and decentralized approaches. Every node supporting such functionality is able to participate in planning and executing caching strategies, or to be part in storing data actively in the network. For example, the placement planning for popular and small transient data is well suited to be managed in a decentralized fashion. A node is able to evaluate such traffic class locally and prefetch the data, nevertheless in case of existing multiple copies in the network. Planning the placement of data chunks of a personalized data stream requires certain knowledge about the mobile node, the network topology, and other additional parameters. Such kind of data is suited to be managed in a centralized fashion.

Figure 5.7 illustrates the vision of a harmonized architectural design of the caching framework. By introducing such framework in information-centric connected vehicular systems, the following actors are involved (cf. Figure 5.7):

Table 5.3: Comparison of the available content placement + replacement strategies in the NDN architecture for mid traffic volume.

Policy	NDN (LCE)	NDN (Prob)
LFU	$\approx 18.3\%$	$\approx 18.8\%$
LRU	$\approx 34.1\%$	$\approx 31.3\%$
Random	$\approx 20.7\%$	$\approx 20.9\%$
FIFO	$\approx 31.9\%$	$\approx 29.2\%$

- **Requestor/Trigger**: the node which triggers the caching strategy. For example, a vehicle requests for some data objects, which trigger the planning functionality at an orchestration node. Furthermore, this can also be a 3rd party information provider or the network provider itself, which is interested in caching its data close to the consumer proactively.

- **Execution Node**: a node involved in executing and distributing a proactive caching strategy and/or actually performing the functionality of placing a data item at any node in the network, including itself.

- **Strategy Provider**: provider of the caching strategy to be performed. In our use case this might be a regulation authority that provides the strategy that runs on the execution nodes.

- **Caching Node**: the actual node storing data items within its local cache.

By sending a request to the network asking for data, the requestor triggers a placement strategy at any intermediate execution node. This also applies to the strategy provider which is able to trigger certain strategies or able to deploy new ones at execution nodes (Figure 5.7, step 1). As a next step and depending on the strategy used, the execution node starts the calculation of the optimal placement of data at nodes in the network providing storage capabilities. The main element in the framework is the *cache strategy management* component, which is part of each node (Figure 5.7, 'Cache Manager'). It is responsible to manage and keep track of caching strategies as well as to provide an interface for managing proactive caching requests for nodes that are organized as part of a multi-tier cache system (cf. Section 3.2).

Strategies will be provided and assigned to the strategy management component by the operator of the network. For example, operators of infrastructure road-side units can assign caching strategies according to their specific needs or trigger caching strategies remotely. While it is expected that geo-specific execution nodes are well suited to execute centralized planning and dissemination approaches, edge components such as road-side units are expected to perform local or cooperative strategies. Each of the introduced roles can be under control of one or more authorities. It has to be noticed, that it is possible, that the requestor and the execution node reside on the same physical location (e.g. vehicle or infrastructure node), while a caching strategy is triggered.

As part of the presented proactive caching framework, the focus of bringing data closer to consumers is strongly associated with the automotive data traffic classes (cf. Section 5.1.1). According to theses classification, data can be grouped into different classes dependent on the characteristics of the data itself. These characteristics are used as a basis to investigate caching

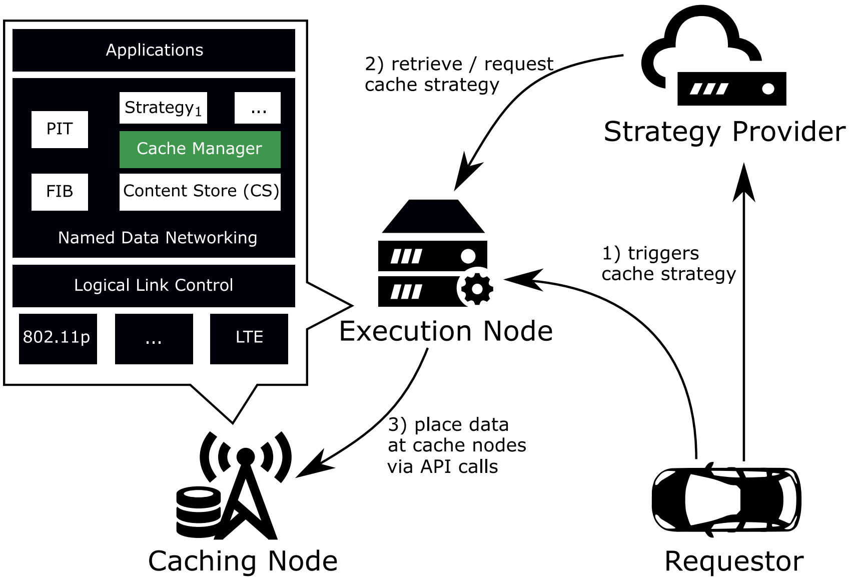


Figure 5.7: Vision of the *Proactive Caching Framework* for Information-Centric Connected Vehicles. The Cache Strategy Manager is responsible to retrieve and execute the different caching strategies. The manager enables the network node to participate in planning and executing caching strategies. PIT, FIB) and the CS are data structures of the Named Data Networking architecture [13].

strategies and develop novel ones to increase the availability of data, decreasing the delivery time as well as to reduce the overall load in the core network.

5.3 PeRCeIVE: Centralized Managed Approach for Personalized Data

The first caching strategy presented in this thesis focuses on bringing Information Objects associated with the class of personalized data to the consuming nodes. Such class of data is of interest to an individual or a small group of consumers. The amount of data to be transferred through the network ranges from small personalized data (e.g., information of the consumer's smart home) up to large files such as personalized map layers of the *electronic horizon* use case in Section 1.2.1 (e.g., information of Point-of-Interests nearby including extensive background information), or software updates for in-vehicle components of a specific car model. Independent of the amount of data to to be transferred through the network, consumer will benefit from such data placed close to the road. In order to place the desired information at the right node in time, a proactive caching strategy focusing on such class of data requires a certain knowledge about the network topology, the requested data items as well as some information about the consuming node(s).

The **PR**oactive **C**aching strategy for **I**cn-based **V**an**E**ts (PeRCeIVE) describes a placement strategy in the category of *hierarchically organized* and *centrally managed* proactive mechanisms. The approach focuses on placing **personalized**, **transient** data closer to consumers. Based on environmental information, PeRCeIVE is based on the NDN architecture and can be executed at geo-specific nodes in the caching framework, e.g., nodes managing several APs including RSUs or cellular base stations.

As it is intended by NDN, the strategy uses a hierarchical namespace. In order to provide an optimal placement within the caches along the road, PeRCeIVE requires some information from the network as well as from the requesting node.

5.3.1 Requirements

The requirements of PeRCeIVE are separated into (i) consumer related information, and (ii) network infrastructure related information. In order to determine the right network nodes to place data items to which a consumer will be connected soon, the execution node running the PeRCeIVE strategy requires the following information from a consumer:

$\vec{P_{car}}$ **Position**: describes the location of the vehicle given as position vector and used to identify the part of the infrastructure network, where the data items need to be cached.

$\vec{V_{car}}$ **Velocity**: describes the velocity vector of the car. This information is used to approximate the movement of the mobile node, and therefore, to identify the right network nodes the vehicle will be connected soon.

f_I **`INTEREST` Frequency**: describes the number of received `INTEREST` packets by the `PeRCeIVE` node and used to predict the points in time of the different requests.

Furthermore, the execution node requires knowledge about the structure of the sub-network to determine the optimal distribution of data within the infrastructure nodes:

o **Information Object**: describes the Information Object requested by the consumer. This information is used to calculate the number of total chunks that build the object, in case it has to be divided into several parts.

$\vec{P_{AP_i}}$ **AP positions**: describes a set of position vectors of each AP within the network (e.g., RSUs). The vector is used to identify APs involved within the data dissemination, close to the vehicle's position.

R_{AP_i} **AP ranges**: describes the transmission ranges of each APs and is used to resolve the position of the vehicle within the range of a AP.

Based on all the required information, PeRCeIVE is able to calculate the optimal distribution of data chunks in the network. The following subsection describes the approach in detail.

5.3.2 The `PeRCeIVE` Approach

The `PeRCeIVE` strategy is separated into two parts: (i) calculation of the optimal distribution of chunks, and (ii) loading the data chunks into the caches of APs, both explained in detail as part of the following paragraphs.

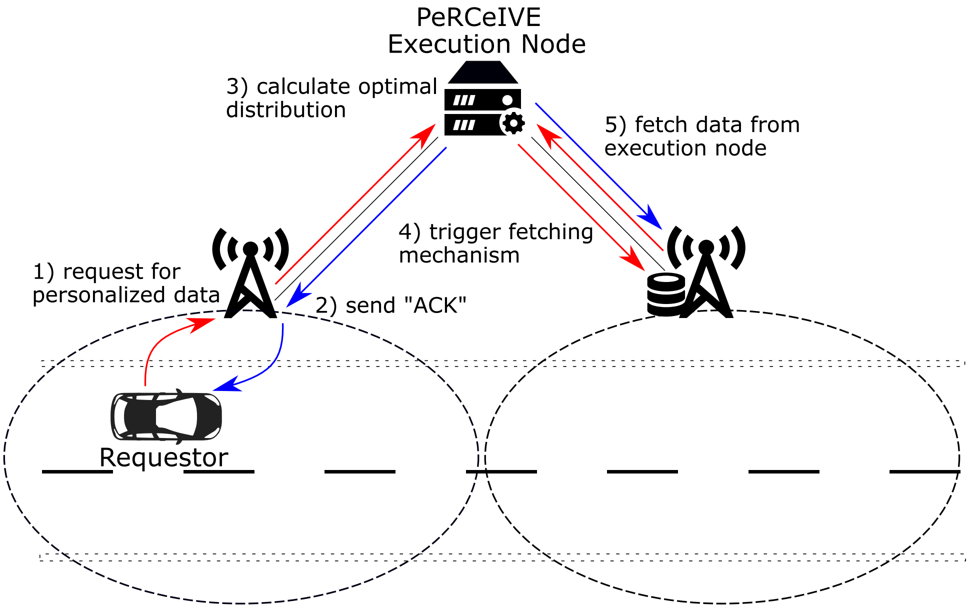

PeRCeIVE
Execution Node
3) calculate optimal distribution
5) fetch data from execution node
4) trigger fetching mechanism
1) request for personalized data
2) send "ACK"
Requestor

Figure 5.8: PeRCeIVE: Step-by-step instructions of the centralized managed caching approach. `INTEREST` packets are given in red, while `DATA` packets are illustrated in blue.

Optimal Content Distribution

PeRCeIVE monitors traffic flows in the network to be aware of `INTEREST` packets requesting large data. A step-by-step illustration is given as follows: A vehicle sends out an initial `INTEREST` (Figure 5.8, Step 1) to the network, requesting for a personalized Information Object by using a hierarchical name. As part of the initial `INTEREST`, the vehicle provides additional information such as the current position, velocity and possible `INTEREST` frequency (cf. Section 5.3.1). This information is observed by the PeRCeIVE strategy which runs on any execution node nearby, to determine the right placement of data chunks at the edge of the network. The initial `INTEREST` is forwarded from the vehicle over the APs to the execution node, following the standard NDN forwarding rules [13].

Based on all input parameters from the vehicle as well as the network, the PeRCeIVE approach calculates an output containing the list of chunks as well as an optimal distribution strategy of the chunks along the available APs. A representation of the distribution algorithm of PeRCeIVE is provided by Algorithm 1. In this manuscript, the algorithm works in a two-dimensional space: (i) calculate and respond with the right number of chunks (if the Information Object has to be divided into several parts) , and (ii) calculate the optimal distribution by identifying the APs for each dedicated chunk.

As a first step, the approach requests the Information Object from the network, if the object is not already part of the local cache. As a next step, the algorithm calculates a list of chunks used to transmit the requested data object. For this, the size of the object and the maximum payload is used to calculate the number of chunks (cf. Algorithm 1, line 1). Included within a Manifest response (e.g., File-like ICN Collection (FLIC) [254]), the number of chunks to be requested by the vehicle is sent back to the consumer (cf. Figure 5.3.1, Step 2 & Algorithm 1, line 2) and therefore be used to request the different `DATA` packets accordingly.

As part of the second dimension, the actual chunk distribution is calculated. By taking into account the vehicle's position $\vec{P_{car}}$, its velocity $\vec{V_{car}}$ and the `INTEREST` frequency $f_{Interest}$, the algorithm iterates over all chunks c to determine the position $\vec{P_{I_c}}$ where the vehicle is expected to send out an `INTEREST` packet for a specific chunk (cf. Algorithm 1, line 4). Afterwards, the algorithm iterates over all APs i to calculate the distance vector $\vec{D}$ between $\vec{P_{I_c}}$ and the position of a particular access point $\vec{P_{AP_i}}$ (Algorithm 1, line 6). The distance vector is used by the execution node to determine if a vehicle's request position $\vec{P_{I_c}}$ is within the range of an access point i by using the length of the distance vector and the communication range $Range_{AP_i}$ (Algorithm 1, lines 7-10). If a vehicle's request position $\vec{P_{I_c}}$ is within the range of the AP_i, the chunk will be added to a list of chunks which will be cached proactively at the AP_i. As a result, an optimal distribution strategy of data chunks is calculated by the algorithm of PeRCeIVE.

Algorithm 1 PeRCeIVE distribution algorithm

```
Require: $\vec{P_{car}}, \vec{V_{car}}, f_I, o, \vec{P_{AP_i}}, \vec{R_{AP_i}}$
 1: function CALCDISTRIBUTION
 2:     noOfChunks = o/maxPayload
 3:     sendManifest($noOfChunks$)
 4:     for all chunks c of noOfChunks do
 5:         $\vec{P_{I_c}} = \vec{P_{car}} + \vec{V_{car}} * c/f_I$
 6:         for all APs i do
 7:             calc $\vec{D} = \vec{P_{I_c}} - \vec{P_{AP_i}}$
 8:             calc $len_D = |\vec{D}|$
 9:             if $len_D <= R_{AP_i}$ then
10:                 add chunk c to list of chunks of $AP_i$
11:             end if
12:         end for
13:     end for
14: end function
```

Load `DATA` into Caches

The need for push-based mechanisms in ICN have been extensively argued in the literature. However, most of the ICN architectures introduce a *pull-based* communication approach which result in additional overhead, for example when loading content into caches proactively (cf. Problem Statement in Section 5.1). This also affects the NDN architecture [13]. As part of the PeRCeIVE caching strategy, an efficient push mechanism is required to send content to one (*simple push*) or a set of network nodes (*group push*). In interest-based ICNs, the available options of "pushing" data items towards nodes are [131, 255, 126]:

- **Interest-Overloading**: Arbitrary data is included in the name of an `INTEREST` packet.
- **Unsolicited-Data & Long-term Pending Interests**: Special types of `INTEREST` packets are used to keep forwarding states in the routers alive, so that `DATA` packets can be forwarded back to the consuming node.
- **Content-Descriptor-based Technique**: By introduction "subscription" mechanisms to NDN, subscription nodes are notified of prefetching `DATA` into their local caches.

- **Interest-Triggered**: Classic INTEREST packets are used to trigger prefetching mechanisms at caching nodes.
- **Interest-Polling**: INTEREST packets are sent by the caching nodes periodically to check for new DATA to be prefetched.

The options are described in detail in the following paragraphs, highlighting the advantages and disadvantages with respect to proactive content placement strategies.

Interest-Overloading: Arbitrary data can be included in the INTEREST packet type as (i) part of the name component, or (ii) by including signed and/or encrypted data as an INTEREST payload. On the one hand, including information as part of the name complicates the naming structure and is only feasible for small data. On the other hand, content included within the INTEREST packet implies some substantive changes of the packet structure of ICN approaches. Forwarding structures (FIB) have to be used to support this effort and accordingly all caches need to have a routable unique name by which the execution node could send data to each of the caching points. The advantage of overloading INTERESTs is that network balancing is not changed, so each INTEREST packet is followed by a DATA (e.g., COPSS architecture [126]). However, APs have to be routable via the FIB which is more complicated if APs are virtual nodes that could be migrated (similar to approaches such as Network Function Visualization (NFV)). In the case of PeRCeIVE, INTEREST overloading is useful for vehicles/APs to send periodic status updates, since the payload is relatively small. However, INTEREST overloading for content/service sent by the execution node to the APs is not efficient (since it is a violation of the design of INTEREST packets in NDN). Moreover, INTEREST overloading does not facilitate distribution of the same DATA to a set of APs using the same name since the routers are enabled to forward the DATA to any one of the matching entries in FIB.

Unsolicited-Data & Long-term Pending Interests: Unsolicited data is difficult to route, since DATA packets flow the reverse path of INTERESTs by default and are dropped by the intermediate nodes if there is no entry in the PIT. Long lived INTERESTs or *stable PIT entries* are introduced to overcome that problem. However, such mechanism may lead to a decrease of performance algorithms since a large number of entries is expected in large-scale networks which occupies valuable resources. Furthermore, DATA packets can be abused to flood the network. In PeRCeIVE, caching points could send long term pending interest to the *execution node*. Since the entries in the PIT will stay for a long time in the routers, the execution node is able to push DATA packets to caching nodes at any time. However, that would imply that each AP needs to have a unique identifier that the execution node could use to address the caching point. For example in Figure 5.8, the left hand side AP sends an INTEREST with name: /exec_node/ap1/, while the right hand side AP sends an INTEREST with name: /exec_node/ap2/. The FIB is configured to forward such requests to the suitable execution node in the network.

Content-Descriptor-based Technique: A Content Descriptor (CD) defines a combination of tags, keywords, location and other properties. In this mechanism, the caching points will have to send a "subscription" that is forwarded to either a rendezvous point in the network or directly to the execution node (e.g., [126]). These entries are stored in a subscription table (or the PIT) and used by the execution node to push DATA to those subscription channels.

The advantage of this mechanism is that multiple APs can be grouped and addressed together to receive one certain DATA .

Interest-Triggered: Additional INTERESTs are used to notify the consumer about the presence of new DATA. Furthermore, it can be used to trigger a "normal" INTEREST at consumer side. Obviously, the disadvantage is that additional round trips are required, resulting in additional delay before the produced DATA arrives at the consumer. This approach is a variant of the "Interest Overloading" approach, wherein only a snippet, manifest or meta-data is sent in the first step. Using this mechanism, the execution node sends out INTERESTs to each AP that should cache a data chunk, including the name used to request for the exact DATA. Similarly, the authors of [255] introduce a new packet type in CCNx - the *Notification* packet.

Interest-Polling: INTEREST polling describes a mechanism at the application level to send out INTEREST packets periodically, asking a producer for any new information. The performance of this approach is very much dependent on the INTEREST frequency. High frequency increases the network load and PIT occupancy, while low frequency may lead to missing information changes.

The introduced options mostly differ in where state is stored or transferred through the network. There are several options such as transferring it as part of the INTEREST packet itself, storing it in existing data structures at forwarding nodes such as FIB and PIT or as part of a new data structure such as the *subscription table* in COPSS [126]. Additional influencing factors are the size/overhead of the first packet that is sent (e.g., large in the case of Interest-Overloading or Interest-Triggered), as well as the frequency of the messages sent (e.g., large number of polling/refresh messages). The placement concept of PeRCeIVE is agnostic to all of the presented options to load data into cache nodes.

To summarize, PeRCeIVE neither replaces the default forwarding behavior of NDN nor requires changes in the set of messages or other NDN related strategies. So if an INTEREST does not directly hit an AP cache, the packet is forwarded towards the originator using the default NDN routing and forwarding strategies. An extension of the concept may include a CD-based technique, since the CDs could represent a combination of properties such as geo-area, type of content/service or priority. Additionally, CDs can be used by the various execution nodes to push the right content/service to a set of APs such as RSUs. However, a modification of the original NDN network stack is required in order to provide CD-based techniques.

5.4 ADePt: Decentralized Managed Approach for Popular Transient Data

The second caching strategy presented in this thesis focuses on bringing popular Information Objects which are associate with the class of transient and small data closer to consuming nodes. Such class of data is of interest to a larger audience. Regarding this class of data, the amount to be transferred through the network is small compared to the focus area of PeRCeIVE (cf. Section 5.3). Examples for such class of data are road condition or traffic update information which are of interest for a larger group of consumers in a certain geo-location, but only valid for a short period. Instead of loading such information from cloud infrastructures or 3rd party service providers, it is beneficial to load this information proactively into cache nodes along the road.

The **A**daptive **D**istributed Cont**e**nt **P**refe**t**ching (ADePt) describes a proactive caching strategy which is characterized by a *decentralized managed* and *distributed organized* structure. The approach focuses on prefetching **popular**, **transient**, **small** data closer to the consumers of the network. Besides being of another class of automotive data traffic, ADePt is different to PeRCeIVE by placing data items at the edge of the network in a fully distributed fashion without relying on a central component on the network. The major goal of ADePt is to increase the availability of the right content, at the right time and at the right node by enabling routers at the edge of the network to analyze and prefetch popular data locally.

In order to place the desired information at the right edge nodes, the caching strategy defines some requirements which are introduced in detail in the next section.

5.4.1 Requirements

Similar to PeRCeIVE, the ADePt strategy also monitors traffic flows in the NDN environment to learn about popular data. Whenever an INTEREST or DATA packet arrives at an node executing the ADePt caching strategy, certain information is extracted from each of the traversed packets independent of the applications and services executed in the network:

I_{name} INTEREST **Name**: describes the name of the received INTEREST packet used to address a certain information. The name is used to keep track of requests and responses of the same data item.

f_I INTEREST **Frequency**: describes the number of received INTEREST packets by any ADePt node in a certain period. The interest frequency tracks how many vehicles ask for the same data. This information is used to identify popularity of that data item.

t_{rec} INTEREST **Receive Time**: describes the time an INTEREST packet is received at an ADePt node.

t_{fresh} **Freshness Time**: indicates a period during which the DATA will remain valid in the cache of the node before staled. In NDN, the freshness value implies how long the originating source considers the content of the DATA packet valid and up-to-date [152]. This information can be used to determine the prefetching (pull) frequency for new content, if a DATA packet is elected as popular.

In addition to the extracted information, the ADePt mechanisms proposes to use an hop counter as part of a monitored DATA packet.

$h_{\texttt{DATA}}$ **Number of Hops**: the number of hops a received DATA packet took from the responding node in the network. A larger number of hops, indicates that the information has been sent from a node further away and additional latency might be expected when requesting it. The prefetch algorithm can incorporate this information to decide when to prefetch new data.

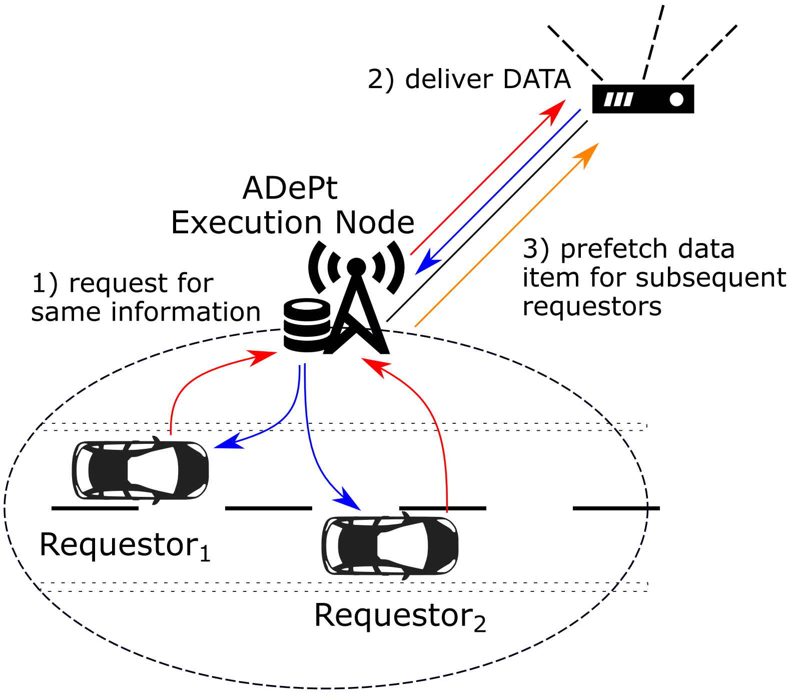


Figure 5.9: ADePt: Step-by-step instructions of the decentralized managed caching approach. Popular data is evaluated by the ADePt execution node (here: requestor1 and requestor2). After the data item expires in the cache of the execution node, ADePt sends out a prefetching requests to update the local cache with a vital copy of the data item for subsequent requestors.

5.4.2 The `ADePt` Approach

The ADePt approach learns about popular, transient data and prefetches it from the network actively for caching at the edge. ADePt is separated into two parts: (i) gathering information of named data to learn about popular data, and (ii) the decision and prefetching execution mechanism.

Gathering Information

Based on the capability to monitor traffic flows at a node running ADePt, the approach learns about popular, transient data transferred in an NDN environment. Whenever a vehicle sends an `INTEREST` packet querying for a specific Information Object (Figure 5.9, Step 1), any node running the ADePt caching strategy extracts information such as the name I_{name} of the requested Information Object as well as the frequency f_I. A processing example of the extraction mechanism is stated in Algorithm 2 (Figure 5.9, Step 2). While processing incoming `INTEREST` packets, the name I_{name} is added to a list of `INTEREST`s for observation. Besides the name, the time when the packet has been received t_{rec} is stored as part of an event. According to the NDN processing rules of incoming `INTEREST` packets, the router consults its local cache for a copy of the desired object. If the router is not able to directly answer the `INTEREST` with a cached `DATA` packet, an entry to its PIT is added and the packet will be forwarded into the core network towards the producer of the data.

Algorithm 2 ADePt: Incoming Interest packet processing algorithm

```
Require: $observationList$[]
function onInterestReceived(interest)
    $I_{name}$ = interest.name
    if $observationList$.contains($I_{name}$) == False then
        $observationList$.add($I_{name}$)
    end if
    increaseEventCounter($I_{name}$, $t_{rec}$)
end function
```

In NDN, all incoming DATA packets are processed by the forwarding plane [13]. In ADePt, an algorithm (cf. Algorithm 3) extracts information directly from the incoming DATA packet which is required for the decision making. As part of the evaluation of incoming DATA packets, the freshness of the data t_{fresh} is extracted. Additionally, a hop counter $h_{\texttt{DATA}}$ is also extracted from the packet. Including the hop counter information is introduced as part of the ADePt concept.

If there is a matching INTEREST in the observation list of ADePt, the entry is set to verified (Algorithm 3). It indicates a valid data flow in the network and ensures that ADePt only prefetches data from the network which has a valid flow. The hop count h_{DATA} as well as the freshness time t_{fresh} is added to the observation. In any case, the DATA packet is forwarded towards the consumer(s) in the network according to the forwarding rules in a NDN system (Figure 5.9, Step 3).

Algorithm 3 ADePt: Incoming Data packet processing algorithm

```
Require: $observationList$[]
function onDataReceived(data)
    $D_{name}$ = data.name
    $h_{Data}$ = data.hopCount
    $t_{fresh}$ = data.freshness
    for all entries $observable$ in $observationList$ do
        if $observable.I_{name}$ == $D_{name}$ then
            $observable.verified$ = True
            $observable.hopCount$ = $h_{Data}$
            $observable.freshness$ = $t_{fresh}$
            break;
        end if
    end for
end function
```

In case an INTEREST cannot be satisfied by the network, the corresponding entry in the *observationList* will never become verified. To prevent the list growing ever larger with stale entries, a fixed length list with an LRU eviction strategy can be used (cf. Section 3.2.5). Alternatively, the *observationList* can be coupled with the management of the PIT entries in the NDN stack. Whenever an INTEREST from the PIT is removed that has no "verified" state in the *observationList* (e.g., it was an INTEREST that has never been answered), the entry can be removed from the *observationList* as well.

Decision Making and Prefetching Mechanism

In parallel to the gathering of the key values described above, two decision making and prefeteching algorithms are regularly scheduled on each node running ADePt. The first algorithm is required to evaluate popular data items based on the entries of the observation list. The second algorithm monitors the vitality of the data items in the local cache and prefetches items from the network accordingly. Dependent on the results of these algorithms, ADePt decides whether to prefetch certain named data from the observation list.

Evaluate Popular Data: In ADePt, the popularity of content is tracked by individual routers. Each time a request for a certain data reaches the router, a counter is increased. ADePt bases its popularity evaluation on the INTEREST frequency f_I, where frequency denotes the number of requests for the same data over a certain timespan. The approach evaluates the counts of the entries on the observation list regularly to determine the INTEREST frequency of each observed data item during the last time window. Thus, the popularity of a named data item can be defined as:

$$popular = \begin{cases} True, & f_I \geq T_{popularity} \\ False, & \text{otherwise} \end{cases} \tag{2}$$

If f_I exceeded a certain popularity threshold $T_{popularity}$, the data is considered as popular. The parameter $T_{popularity}$ can be used to configure the aggressiveness of the prefetching algorithm. If $T_{popularity}$ is smaller, less requested data is cached and the system reacts faster before the first item is prefetched. On the other hand, a low $T_{popularity}$ threshold increases false positives and thus increase load on the core network and consume resource capacity (e.g. cache memory) on NDN routers by unnecessarily prefetching data. It has to be mentioned that there are several mechanisms available to identify popular content in ICNs (e.g., [16, 249]). To evaluate ADePt, a generic approach has been chosen as baseline, however this algorithm can be exchanged within the proactive caching framework of ADePt to use more complex and application-specific algorithms.

Loss of Content Freshness: Another factor for decision making is the loss of freshness of popular DATA within the caches of network components such as RSUs. In a NDN, every DATA packet has a freshness value, that denotes how long the source assumes that DATA item to be valid. As long as the freshness has not expired yet, a NDN router will answer with the cached object, when it receives a corresponding INTEREST packets. Once the freshness of DATA has expired, it will not be delivered to consumers anymore and subsequent INTERESTs will be forwarded to neighbors.

To increase the availability of transient data items, the goal is to prefetch new DATA before cache entries expire. But, such mechanisms have to be triggered not too early as this would generate unnecessary traffic in the network, if no updated information is expected. In this case, the number of hops $h_{\texttt{DATA}}$ defines an important input to determine the optimal time to send out a prefetching request. The larger the number of hops $h_{\texttt{DATA}}$, the higher the latency values and the older the content is at the time of receiving the DATA packet. Especially in a NDN, more hops can imply higher latency even in a fully wired backbone network, as NDN operations can require costly operations reading and writing from a persistent cache. "Outdated" data is defined as data that is about to expire and should be prefetched, as:

$$outdated = \begin{cases} True, & t_{remain} - (h_{DATA} * c) \leq T_{vitality} \\ False, & \text{otherwise} \end{cases} \tag{3}$$

As stated in Equation 3, t_{remain} defines the time before an item looses its freshness (and will no longer be delivered to consumers). If that time is below a certain threshold $T_{vitality}$, ADePt considers that data item as outdated. In addition to t_{remain}, a "penalty" is added to the approach for every hop the data travels, weighted by a constant factor c. Note, that this system can also be adapted for a cascade of ADePt routers: If data becomes popular and an edge router running the ADePt strategy decides to prefetch data, h_{DATA} might be high for the initial DATA packets received. However, once immediate ADePt nodes in the core network begin actively prefetching data, the edge node will see a smaller h_{DATA} and adapt its prefetching strategy accordingly.

Decision Making and Prefetching: The decision making Algorithm 4 regularly evaluates each entry in the observation list. Based on the popularity and the expiration time of the freshness value of a certain entry in the observation list, the item will be prefetched from the network to be loaded into the local cache. If an entry exceeded the popularity threshold $T_{popularity}$ and has fallen below the freshness threshold $T_{vitality}$, then the entry is prefetched by generating and sending out a INTEREST packet using the verified name I_{name} from the observation list.

Algorithm 4 ADePt: Prefetching decision making algorithm

```
Require: $observationList[]$
function ONPREFETCHINGDECISIONMAKING
    for all entries $observable$ in $observationList$ do
        if $observable$.isVerified == True then
            if $observable$.isPopular == True && $observable$.isOutdated == True then
                sendInterest($observable.I_{name}$)
                resetEventCounter($observable.I_{name}$)
            end if
        end if
    end for
end function
```

To summarize, ADePt detects popular, transient data items by monitoring the traffic flows and extracting the request frequency for these items. The strategy does not replace the default forwarding behavior of NDN. However to improve the performance of the decision algorithms, the ADePt approach requires some changes in the packet structure to add a hop counter, and thus, to determine the optimal time to send out a prefetching request. When an item is selected to be loaded from the network, ADePt sends out a classic INTEREST packet to the network and stores/overrides the responding item in the local cache of the node running ADePt. As a result, the item is available for subsequent consumers interested in the same information.

5.5 Advanced Prefetching via Data Prediction

The previous presented PeRCeIVE and ADePt caching strategies reduce the delivery times of Information Objects by storing them closer within infrastructure nodes proactively. When looking into their trigger mechanisms, both strategies are either *delay* triggered (e.g., in case of ADePt), or directly consumer triggered (e.g., in case of PeRCeIVE). This has consequences for the quality of the caching result. For example, the ADePt strategy has no positive effect for the requesting consumers before reaching the prefetching threshold.

Compared to host-centric networks, the paradigm shift in ICNs of addressing data directly using naming schemes as well as pushing higher layer functionality to the networking layer offers new opportunities to utilize such information and to improve the performance of the network. For example, information encoded within the names can be used and combined with other techniques from computer science for proactive cache decision making.

One of these techniques is Machine Learning (ML). It defines a research area concerned about the analysis of large amount of data to detect and recognize patterns, e.g., to improve further processing or to achieve an optimization goal [256]. Especially in the research of Internet traffic classification, ML techniques have shown performance improvements, e.g., learn and predict traffic peaks in order to manage the network resources efficiently (e.g., [257]). In this section, a predictive prefetching approach is presented to forecast the appearance of consumer requests to prefetch the right content beforehand based on techniques from the domain of ML.

5.5.1 Introduction into Predictive Analytics

Predictive analytics describes a technique to make estimations about future events based on historical knowledge. It uses techniques from data mining, data modeling, statistics as well as ML to analyze the historical data and to derive common rules for future events. The following paragraphs start with an introduction to ML techniques, followed by a presentation of regression algorithms required for the subsequent concept sections.

From a abstract perspective, machine learning algorithms can be separated into three categories (cf. Bishop [256]):

- **Supervised Learning**: The provided data samples include input as well as corresponding output values to learn about patterns and to derive common rules to be mapped for future input values. *Regression* algorithms are useful to solve the problem, if the desired output values consist of a set of continuous variables – a linear relation between input and output values exist (e.g., forecasting weather reports). In case the output values correspond to a discrete number of categories, *classification* algorithms are useful to solve the problem (e.g., email filter to detect "spam").

- **Unsupervised Learning**: The provided data samples only include input values without any corresponding outputs. Only input values can be used to learn about patterns, for example by discover groups, events, items that can be clustered (e.g., categorize recently published articles).

- **Reinforcement Learning**: The goal of these learning algorithms is to find an appropriate action for a given event based on a rewarding system. These algorithms require to discover the influence (output) of an action to an environment to learn about the effects of the suggested action and to maximize the reward. An example for such problem is the balancing of request/task to utilize the resources of a data center in order to keep the response times as low as possible.

When looking into algorithms of each of the categories, there are common processing phases which apply to each machine learning algorithm. Figure 5.10 illustrates these phases in chronological order. Any ML approach starts by **collecting the data**. This data forms the basis to learn about patterns and to derive common rules. In the next step, the collected data has to be pre-processed (cleansed) to drop incomplete data sets, re-structure events, filter out errors, etc. Such **pre-processing** improves the quality of the ML result and increases the performance of the algorithms. After the data basis has been prepared, the appropriate machine learning algorithm has to be selected and applied to the data basis. Correlations and causation helps to determine relationships of input values within the data basis, and to **create the appropriate model**. Before the deployment phase, it is required to **evaluate** the performance of the created model. Ideally, the model should neither over-fit (fit the pattern perfectly) nor under-fit (unable to capture the pattern) the detected pattern in the data set. In this case, *cross-validation* is used by dividing the data basis into a *learning* (e.g., 70% of the data basis) and a *test* set (e.g., 30% of the data basis). This procedure is repeated periodically until the parameters of the model fits the pattern. Finally, the model is implemented and **deployed** in a live system, so that the outputs of the model are used (e.g., create a forecast report).

In this thesis, techniques from the *supervised learning* are used to make predictions of Information Objects to be prefetched proactively. More precisely linear *regression* models (cf. Bishop [256]) are created, since the data basis used provides both input and corresponding output values (cf. Section 5.7.3).

Related Work of Predictive Analytics Mechanisms in Information-Centric Networks

In contrast to the default reactive cache strategies in the NDN architecture (cf. [13]), prefetching strategies are loading data items into nodes before a request is overseen in the network. However, such strategies are only as good as the mechanisms used to identify the valuable content.

In the literature, several mechanisms have been proposed to identify valuable data items in ICNs. For example, Cho et al. [15] use a popularity-based approach as part of the WAVE caching strategy. The proposed *selective neighbor caching* by Vasilakos et al. [212] uses consumer related information for predicting the next node a mobile user will be connected soon. While both approaches (cf. Section 4.2.3) target on increasing the availability of certain data items closer to the consumer, they focus on a certain class of data items (e.g., popular data) or require additional information from network participants.

The introduction of ML techniques for optimizing the performance of ICNs is introduced by Singh et al. [258]. In the context of wireless sensor networks, the authors introduce strategies based on heuristics to accelerate the convergence of learning algorithms for improving data delivery paths in sensor networks. Regarding caching at the network edge, the authors of Khelifi et al. [259] propose the combination of different machine learning techniques and the in-network caching capabilities of ICNs to increase the network performance at the edge.

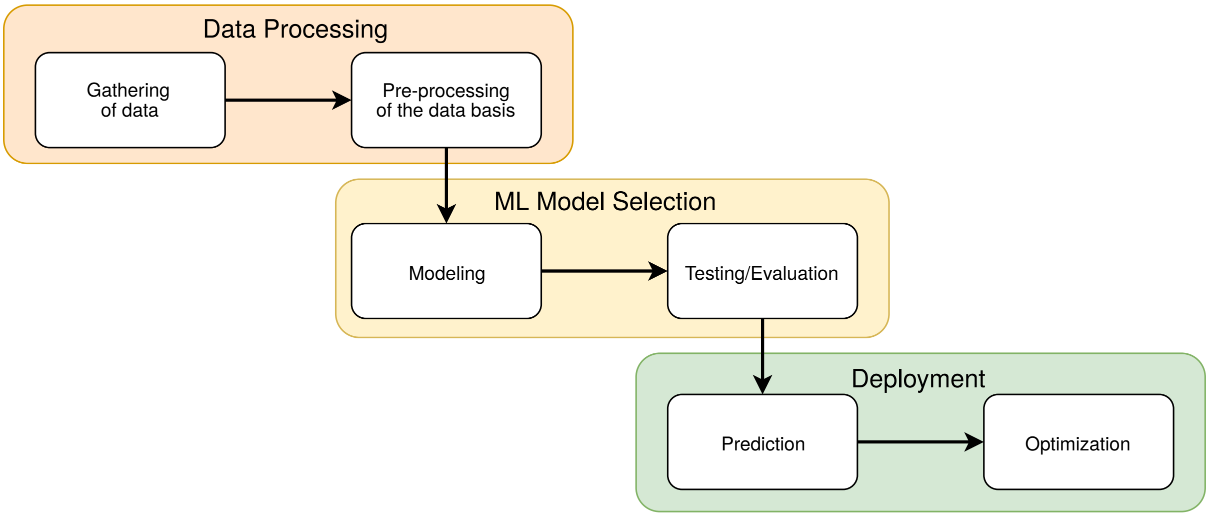


Figure 5.10: Illustration of the common processing steps of Machine Learning approaches. First, a data basis is required in order to create an appropriate model for predicting future events. If the model has been tested and considered suitable, the model is deployed in the live environment, and optimized if necessary.

However, the authors leave information open of how to actually apply ML to ICNs and deploy the approaches in detail for future work.

5.5.2 Requirements

As mentioned in the previous section, a data basis is required to train a model and to be able to make any predictions about future events. Commonly deployed IP traffic classification techniques are based on packet inspection to extract valuable information and to create flow classes for each of the network participants (cf. [257]). Since in ICNs naming schemes are used to query the network for content, these queries are typically sent to the network anonymously. Therefore, the creation of “classic” flows is challenging. While there exist first work towards flow classes in ICN (cf. [260]), the mechanism in this thesis is not built on flow classes. Similar to the ADePt approach, a monitoring and inspection mechanism is required to extract at least the following information to create a significant data basis within an ICN:

ID_{node} **Node ID**: describes the unique identifier of the node overhearing an `INTEREST` and a corresponding `DATA` packet. Such information is helpful in order to identify *where* an information exchange has happened.

I_{name} **`INTEREST` Name**: describes the name of the received `INTEREST` packet used to address a certain information. Since names in ICN carry semantic information, it is used to keep track of requests and responses of data.

t_{rec} **`INTEREST` Receive Time**: describes the time an `INTEREST` packet is received at a specific node.

t_{fresh} **Freshness Time**: indicates a period during which the `DATA` will remain valid in the cache of the node before it expires. In NDN, the freshness value implies how long the originating source considers the content of the `DATA` packet valid and up-to-date [152]. This information can be used to determine the prefetching (pull) frequency for new content.

5.5.3 The Predictive Data Prefetching Approach

The predictive prefetching strategy is based on elements of ADePt (cf. Section 5.4). Similar to `ADePt`, the predictive strategy learns about the request patterns for overheard Information Objects to predict the next requests for a specific Information Object and to prefetch it from the core network proactively. In this thesis, the concept uses linear regression models (category: supervised learning) combined with the *Elastic Net* regularization method and is aligned to the common processing steps of ML approaches (cf. Section 5.5.1). However, the prediction model is continuously improved to perform the actual prefetching steps. The concept is separated into five phases:

P_1 **Traffic monitoring**: A data basis is created by monitoring traffic flows and extracting the information required to perform the prediction approach. This data basis increases constantly, while the monitoring is performed during the runtime of the system.

P_2 **Modeling**: Since the data basis increases constantly during the runtime of the system, the model used for making predictions need to be created, learned and tested periodically. As part of this phase, the existing model is extended by the latest gather traffic information.

P_3 **Prediction and Decision:** By using the model, expected time values of overheard `INTEREST` packets are predicted and suggested to a decision making algorithm. If a request for an `INTEREST` is predicted in the near future, the name is marked to be prefetched from the core network.

P_4 **Prefetching:** The prefetching phase actually loads data from the core network which has been enabled in P_3. It is executed periodically. In order to improve the accuracy of prefetching events, each marked data item is augmented with a *delayed time* penalty value.

P_5 **Prediction Verification:** The verification phase describes a mechansism to verify the accuracy of the predicted time value. If a request for an `INTEREST` is predicted by the model, the time value of the real world request is verified against the predicted time value. The result of the phase is a modification of the values for the prefetching phase P_4, if necessary.

There are several options to deploy different phases at nodes in the network:

- **Centralized deployment:** Similar to the PeRCeIVE strategy (cf. Section 5.3), one central node in the network performs all phases. It coordinates and instruct other nodes to load certain Information Objects into their local caches. The advantages are described by a large data basis which is promising for accurate prediction results. However, gathering all the monitored traffic flows results in additional overhead in the core network. Candidates for such a deployment are geo-specific nodes, or cluster heads in a hierarchical network deployment.

- **Decentralized deployment:** Similar to the ADePt strategy (cf. Section 5.4), each node running the predictive cache strategy performs the phases individually. Prefetching decisions are mad on a local model. There is no additional coordination or traffic flow exchange required. However, the data basis to learn from is expected to be small which may result in inaccurate prediction results. Candidates for such a deployment are infrastructure nodes at the edge such as RSUs or cellular base stations.

- **Hybrid deployment:** It describes a mix of centralized and decentralized deployment by using the best of both approaches. For example, each node which monitors the traffic flows sends it to a central node running the *modeling* phase. As a result, a large data basis and thus appropriate or updated models are created. These models are enrolled to the edge nodes periodically. The remaining phases such as decision making and prefetching are performed at each edge node individually.

Based on the capability to monitor traffic flows at a node running the predictive caching strategy (e.g., a RSU), the approach extracts information from incoming `INTEREST` and `DATA` packets such as the name I_{name}, or the receiving time t_{rec}. This information is stored locally to create a historical data basis of request and response events. When running the monitoring mechanism at several nodes in the network, the history pieces can be gathered by neighboring or at a central node to create a large basis and therefore improving the accuracy of prediction model.

Based on the data basis, models are created using techniques from the linear regression combined with the *Elastic Net* regularization method (cf. [256]). These models are cross-validated by dividing the data basis into *learning* and *testing* sets. If an `INTEREST` packet i is received by a node running the predictive caching strategy, it consults the model to calculate the prediction time $t_{i_{predicted}}$ and stores the time value into a prefetching list. In this case, the prefetching algorithm of the ADePt strategy can be used to load Information Objects from the core network.

If the node overhears the query from a consumer, the `DATA` packet can be directly served from the local cache. Furthermore, the strategy verifies the predicted time value with the real world measured time. The prefetching event can be modified to be executed sooner or later according to the difference between the values.

To summarize, the predictive prefetching strategy learns about consumer requests by monitoring the traffic flows. The strategy uses techniques from the domain of *predictive analytics* which includes data modeling and machine learning algorithms in order to predict `INTEREST` events. Based on the prediction of such events, the strategy prefetches Information Objects into cache nodes at the edge of the network. As a result, the Information Objects are directly delivered to each consumer and overcome the triggering and threshold problem of strategies such as ADePt.

5.6 Simulation Environment

In computer science, there are several methods available to evaluate the performance of a networking concept. The most direct method is based on actual measurements collected from experiments of a real world system. However, at the beginning of a research project, such systems are unavailable, or if available inflexible and expensive to make any extensions or changes. To overcome this limitation, modeling a system, e.g., using probabilistic and statistical methods, describes another option. Especially for evaluating the performance of small systems, such methods are well suited to achieve first generic results rapidly. While recent advances have extended the capabilities of such models towards more complex scenarios, analytical methods often require unrealistic assumptions and approximations to evaluate large and complex systems. For such systems, system-level simulation models describe an option to evaluate the entire system on different levels of abstraction [261].

However, the evaluation of a concept using simulation models involves several risks to be considered in advance to create appropriate results:

- **Model creation and verification:** Simulation models are created based on the abstraction from the real world system such as system properties, characteristics, behavior, or any other assumptions. The details of the model should reflect the details of the available information about the real world system which can make the model complex. Furthermore, the model has to be verified before creating any results.

- **The simulation tools:** The tools used to build up the simulation environment need the ability to create an appropriate model. Sometimes, combination of several simulation tools are required.

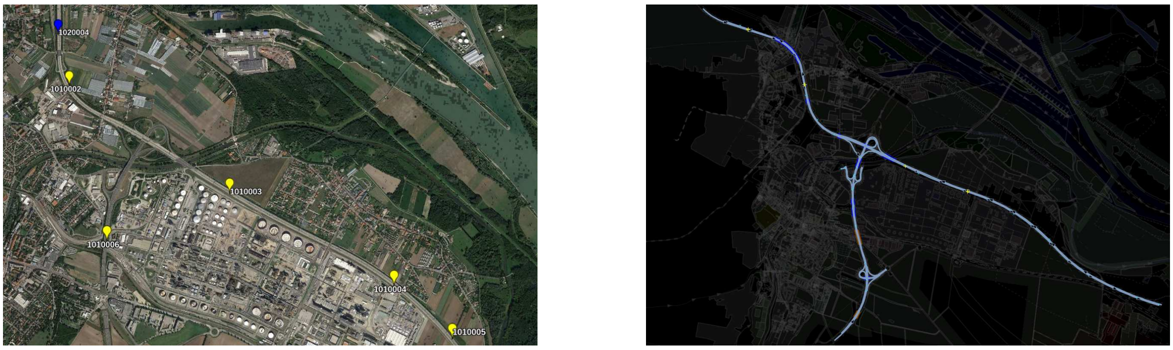
- **Execution duration:** Simulating a large and complex model may result in large execution times. Parallel development as well as distributed execution of parts of the model describe options to tackle long execution times.

As part of this thesis, discrete-event simulations are used to evaluate the performance of the concepts introduced in the previous sections. This kind of simulations are widely in evaluating network concepts. The simulation model created is based on a real world connected vehicle deployment in Austria. The following paragraphs describe the simulation model and tool environment in detail.

5.6.1 The European Corridor Austrian Testbed

The European Corridor-Austrian Testbed for Cooperative Systems (ECo-AT) is an Austrian project to create harmonized and standardized cooperative ITS applications jointly with partners in Germany and the Netherlands [262]. The infrastructure deployment includes several RSUs on the motorway A4 between Vienna and the Hungarian border. Figure 5.11a illustrates the deployment of some units on a map. Each RSU is equipped with an IEEE 802.11p [35] short-range communications interface in the 5.9 GHz band. The communication range is about 150 meter. The RSUs are directly connected to a centralized server via fiber using a simple point-to-point connection.

One of the project partners of the ECo-AT is the Austrian toll company ASFINAG (Autobahnen- und Schnellstraßen-Finanzierungs-Aktiengesellschaft). It operates the motorways and other roads in the road network of Austria and monitors traffic flows, e.g., using cameras, overhead detectors such as radar and ultra-sonic sensors, or induction loops directly installed in the road. Such information are used to provide C-ITS services, for example to increase the safety of passengers offering a traffic jam notification service.



(a) ECo-AT RSU deployment (yellow markers) at the motorway A4 close to Vienna, Austria.

(b) Road structure model of the ECo-AT motorway A4 extracted from OpenStreetMap.

Figure 5.11: Extracts of the ECo-AT RSU deployment at motorway A4 between Vienna and the Hungarian border.

Furthermore, the company also offers the possibility to access traffic flow performance data on a monthly based, collected at over 270 positions including the motorway A4 [263].

5.6.2 Analysis of Real Mobile Applications for Simulation

In order to achieve realistic simulation results, measurements of automotive applications under real conditions are required to simulate the information exchange between consumers and producers. However, there is no vehicular ICN network deployed at the present time which can be used to monitor the communication behavior of automotive applications. While there exist some data traffic traces from mobile IP networks (e.g., [264, 265]), there are no traces available in the context of vehicular applications. Aggravating the situation, most of the traces are given in anonymous/pseudonymous form which complicates the extraction of packet flows and meta information. Furthermore, the presented predictive prefetching strategy (cf. Section 5.5) also requires realistic communication traces to create plausible prediction models.

To overcome this problem, an analysis of mobile application is performed according to the use cases (cf. Section 1.2) and automotive data traffic classes presented in Section 5.1.1. For example, this includes mobile applications such as:

- **Online navigation application:** This class of applications provides information about a predefined route. During the journey the application downloads the required material such as map tiles, traffic updates, points-of-interest, parking information etc. Such data is expected to be coupled to a specific geo-location. Besides the start and destination information, such applications constantly require GPS data to determine the location of the vehicle.

- **Entertainment application:** This class of applications is used for entertainment purposes of the passengers. It includes radio or video broadcasting, or on-demand services and only requires a subscription to the service. Such data is independent of a geo-location.

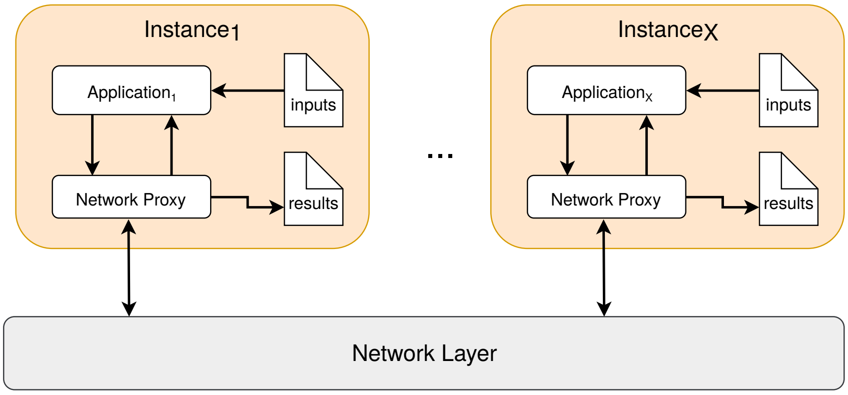


Figure 5.12: Structure of the virtual test drive environment. Emulator instances are used to execute real world applications, while network proxies intercept and record the communication behavior of each of the applications.

There are several possibilities to measure and record the communication behavior of such applications: vehicular and mobile applications can be measured under real conditions during test drives within the ECo-AT road network. While such option is expected to provide the best results, it is coupled with a great deal of effort since vehicle, test track, and measurement equipment have to be provided.

This thesis uses the option of *virtual test drives*. In this option, parts of the test drive are virtualized, for example the road network, the movement of the vehicles, etc. While such virtualization influences the measurement outcomes, it reduces the effort to an acceptable quality. Tools such as emulators can be used to execute real applications, to manipulate inputs such as GPS signals, and therefore virtualize test drives within a virtual machine. Network proxy applications are used to intercept and record the communication traffic, in order to collect application specific information as well as meta-information. According to RFC 7234 [266], recordable information to control cache behavior using HTTP include:

- *URI* of the service to be requested - used to identify the semantics of the application requests (e.g., using available application programming interface definitions of the services)
- *timestamp* of requesting and receiving packets
- *type and size* of the payload received
- *dates* of data generation and expiration
- ...

Multiple instances of the same emulator setup can be executed in parallel, increasing the number of test drives and results, while reducing the required time effort tremendously. Figure 5.12 illustrates the structure of the mobile application recording environment.

The selection of mobile applications follow the presented application categories and automotive data traffic classes (cf. Section 5.1.1). Further criteria in the selection are the assessment and the number of downloads of the mobile applications within their application stores. As a result of the virtual test drives, a large data collection of the communication behavior of mobile applications within the IP world has been created.

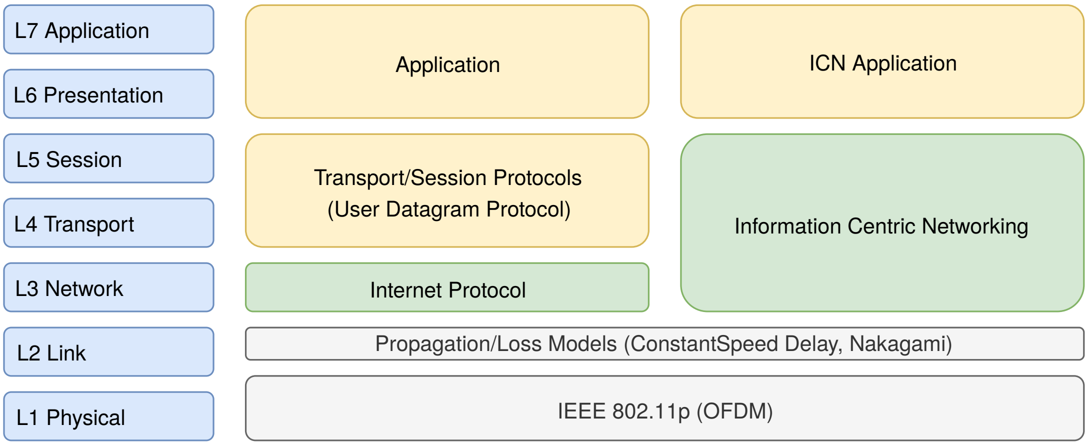


Figure 5.13: Simulation model of the communication stack used within the OSI model. The access layer forms a common basis, created using the IEEE 802.11p standard. The higher layers are different from each other supporting information exchange using the traditional IP protocol stack as well as an ICN stack based on the NDN platform.

5.6.3 The Simulation Model

In this thesis, the simulation model is based on a realistic scenario rather than an artificial academic set-up using the information about the network deployment of the ECo-AT as well as the performance data of traffic flows. It is separated into three different level of abstractions, namely *the network deployment* based on the ECo-AT, *mobility traces* based on the performance data of traffic flows, and a *communication stack* supporting traditional IP as well as NDN information exchange.

The Network Deployment Model

According to the network deployment of the ECo-AT, the model contains several RSU nodes along the motorway A4 in Vienna. The nodes are distributed in a corridor of 10km x 5km and are connected to a centralized server using a simple point-to-point connection characterized by a bandwidth of 100 Mbps and a delay of 5 ms. In total, 2.4km of the corridor are covered by RSUs. Each RSU as well as the vehicle nodes are equipped with an IEEE 802.11p [35] short-range communications interface.

The Mobility Model

While the ECo-AT centralized server and the RSUs are fixed to a constant position, the vehicle nodes have to move through the scenario. In order to model the movement of vehicles as realistically as possible, real world traffic traces from the ASFINAG toll company are used illustrating the distribution of vehicles within the network deployment corridor. The traces include approximately 98.000 vehicular nodes per day, collected during a week in June 2017 [9].

[9] According to a non-disclosure agreement between the author of this thesis and the ASFINAG company, it is not possible to provide the exact numbers of the traffic traces.

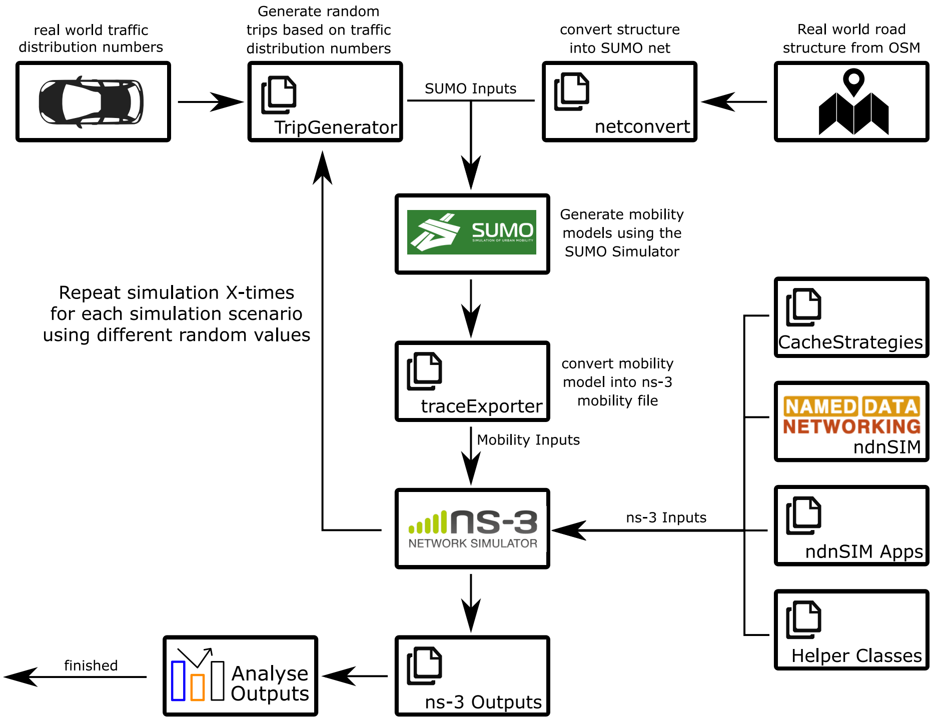

real world traffic distribution numbers
Generate random trips based on traffic distribution numbers
TripGenerator
SUMO Inputs
convert structure into SUMO net
netconvert
Real world road structure from OSM
SUMO
Generate mobility models using the SUMO Simulator
Repeat simulation X-times for each simulation scenario using different random values
traceExporter
convert mobility model into ns-3 mobility file
Mobility Inputs
CacheStrategies
NAMED DATA NETWORKING ndnSIM
ns-3 NETWORK SIMULATOR
ns-3 Inputs
ndnSIM Apps
Helper Classes
finished
Analyse Outputs
ns-3 Outputs

Figure 5.14: The design of the simulation environment: Based on real world deployment and real traffic information, the setup is used to evaluate proactive caching strategies and their effects on the delivery times and cache utilization.

The Communication Model

The basis of the communication model is described by the broadcast IEEE 802.11p short-range communications interface configured to run in the Outside the Context of a Base station (OCB) mode in the 5.9 GHz band with a gross data rate of 6 Mbps. The mode describes a mode which simplifies the exchange of messages between devices without the need of a base station by disabling authentication/association procedures and security mechanisms [35]. In order to model the wireless communication as realistically as possible, propagation loss and delay models are included into the communication such as the Nakagami fading and constant speed propagation delay models. The higher layers in the communication model are different from each other. In order to be able to evaluate the performance, the model supports information exchange using the traditional IP protocol stack as well as an ICN stack based on the NDN platform. Figure 5.13 illustrates the communication model used including the different layers of the protocol stack.

5.6.4 The Design of the Simulation Environment

The simulation model described in the previous section is implemented using the popular discrete-event network simulator ns-3 [267]. ns-3 provides a large range of communication modules including different medium access technologies, protocol stacks and example applications (e.g., client-server applications). In order to be able to evaluate ICN information exchange, the Named Data Networking Simulator (ndnSIM) [268] module is used as a basis.

It describes an additional module developed for the ns-3 network simulator. The entire simulation environment including all tools used is illustrated in Figure 5.14.

In this thesis, the ns-3 version 3.27 [267] and the NDN specific protocol stack bundle ndnSIM 2.4 [268] are used to implement the simulation model (cf. Section 5.6.3). The NDN stack included in the ndnSIM 2.4 bundle is the 0.5 version of the NDN library with experimental extensions (ndn-cxx) [269] as well as a slightly modified version of the Named Data Networking Forwarding Daemon (NFD) [270] for ns-3. In order to be able to use the real world traffic mobility traces in the ns-3 environment, the mobility model of the vehicle nodes is created using the 0.28.0 version of the Simulation of Urban MObility (SUMO). It defines a simulation tool which allows the modeling of traffic systems including vehicles, public transport as well as pedestrians [271].

Based on the network deployment model introduced in the previous section, a detailed map is created by using map information from the OpenStreetMap provider [272]. The created deployment map includes the road network of the motorway A4 in Austria, including parameters such as the permitted maximum speed on the road sections, as well as the position of all nodes of the ECo-AT environment. Figure 5.11b shows the road model of the A4 motorway in Austria. Based on the road model as well as the traffic flows information, a simulation scenario is created in SUMO to generate mobility traces for each vehicle driving through the ECo-AT corridor. Such mobility traces are used later on in the ns-3 simulation environment to move mobile nodes in the simulation according to the real world traffic flows.

In order to be able to exchange information between the modeled nodes, the communication stacks are implemented according to the model presented in Section 5.6.3. In case of ICN, the ndnSIM module is used to deploy an NDN software stack including the NFD on each of the nodes within the simulation scenario. For simulating and measuring the existing caching mechanism, ndnSIM reference applications are used on all network nodes including the vehicles consuming data, the forwarding RSUs and the producer providing access to data in the network. For being able to measure the behavior of the different caching approaches to be evaluated, extended variants of NDN reference application and tracers have been deployed at all nodes in the environment.

5.7 Evaluation of the Proactive Placement Strategies

In order to compare the existing cache strategies and the proactive placement approaches presented in the previous sections, a performance model consisting of four metrics is defined in this manuscript, namely the (i) *cache utilization*, (ii) the *producer load*, (iii) the *resolved `INTEREST` ratio*, and (iv) the *one-hop ratio*.

Cache Utilization (C_A): The cache utilization C_A describes the average number of cached copies of a certain Information Object within the network. The metric is defined as

$$C_A = \frac{\sum_{j=0}^{J} C_j}{J} \tag{4}$$

where J is the number of different Information Objects in the network and $\mathrm{C_j}$ is the number of copies of the Information Object j in the network. This metric is used to investigate the influence of proactive caching strategies with respect to limited caching resources. An efficient proactive cache strategy should keep the number of copies as low as possible in order to save memory on the nodes within the network. Focusing on limited cache resources in information-centric IoT scenarios has been identified as one of the main research issues in the ICN research community [93].

Producer Request Load (L_P): The producer request load L_P describes the total number of received `INTEREST` requests by a producer in the network which are directly answered by the entity. The metric is defined as

$$L_P = \frac{\sum I}{t} \tag{5}$$

where $\sum I$ is the sum of all received `INTEREST` packets sent by vehicles in time t. This metric shows the load on the producer. An efficient caching approach should exhibit few false positives and increase the load at the producer only slightly.

Resolved `INTEREST` Ratio (RR_I): The resolved `INTEREST` ratio RR_I describes a metric to examine the requesting effort for certain data items from the network. The metric is defined as

$$RR_I = \frac{I_{resolved}}{I_{total}} \tag{6}$$

where $I_{resolved}$ is the number of resolved `INTEREST` packets sent by the vehicle and which have been verified via received corresponding `DATA` packet, divided by the overall number of `INTEREST` packets I_{total} sent by the vehicle. This metric evaluates the efficiency of the proactive caching strategies as it takes into account how many of the requests initiated by the mobile consumer are actually fulfilled.

One-hop Ratio (R_1): The one-hop ratio R_1 describes the number of resolved `INTEREST` packets which have been directly answered by network nodes one hop away of the consumer. The metric is defined as:

$$R_1 = \frac{I_1}{I_{total}} \tag{7}$$

where I_1 is the number of `INTEREST` packets sent by the vehicle and which are directly answered by the first hop, and I_{total} is the overall number of `INTEREST` packets sent by the vehicle. The one-hop ratio is a specialization of the resolved `INTEREST` ratio RR_I, howerver, it only considers the resolved `INTEREST` packets which have been verified via `DATA` packets directly received from surrounding nodes. Regarding the *mobile node delivery problem*, a request that needs a high number of hops in order to be satisfied may cause significant problems when the mobility of a vehicle and hence the intermittent connectivity is taken into account. For this reason it is beneficial when the requested Information Object is available on the RSU the car is directly connected to.

5.7.1 `PeRCeIVE`

As presented in Section 5.3, the PeRCeIVE cache strategy is characterized by a *centrally managed* and *hierarchically organized* structure, and focuses on storing **personalized** data at the edge of the network. The implementation for loading data items from the execution node running

Table 5.4: Simulation parameters used to evaluate the PeRCeIVE cache strategy.

Parameter	Values
Comm. technology	IEEE 802.11p OCB
Comm. range RSU ($Range_{RSU}$)	150 meters
no. of RSUs involved	8
simulation time t_{sim}	30 minutes
traffic density (low, average, high)	(15, 30, 60) vehicles/min
Request rate	2–60 seconds
no. of vehicles	450–1800 units
no. of total runs	9000

PeRCeIVE into the caches of APs at the edge network is realized as part of an *Interest-Triggered* mechanisms. For this purpose, each AP provides a unique identifier (e.g., the geo-location within a specific naming scheme) which is well known to the execution node. Each triggering `INTEREST` packet contains all related names of data chunks which need to be cached at the particular AP. These names are used by the APs to fetch the data chunks from the execution node.

To evaluate the strategy against the standard, reactive cache behavior of NDN, simulation runs have been executed for different caching strategies and traffic loads according to the traffic performance data of the ECo-AT environment. The reactive caching strategy in the simulation setup includes LRU, LFU and FIFO replacement on all nodes. Table 5.4 illustrates the simulation parameters for evaluating PeRCeIVE.

Assumptions

Regarding the evaluation of PeRCeIVE, the following assumptions are made:

- only V2I communication is taken into account. Direct communication between other vehicles is not considered.
- the focus is on requesting data from the class of large personalized files, such as the continuous download of an environment model created by the *electronic horizon* (cf. Section 1.2.1) which needs to be separated into smaller chunks for transmission
- the requested data items are personalized, hence the content popularity is low.
- if caching is enabled, only the storage of Information Objects at the RSUs is considered

Results of the `PeRCeIVE` Strategy

As part of the simulation environment, the results of PeRCeIVE strategy have been evaluated against the standard NDN with and without caching capabilities. The main focus of the metrics to be analyzed are: (i) the cache utilization, (ii) the resolved `INTEREST` ratio, and (iii) the one-hop ratio. Since the concept of the PeRCeIVE strategy is *centrally managed*, the number of requests at the originator is expected to be low. Therefore, the *producer request load* metric is not considered for evaluating PeRCeIVE.

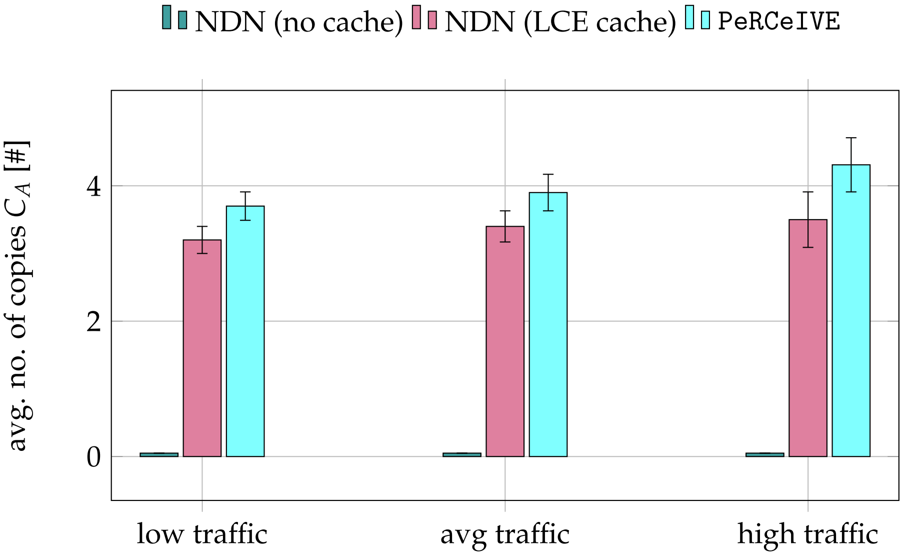


Figure 5.15: PeRCeIVE: Total number of cache utilization. If caching is enabled at the RSU nodes, PeRCeIVE occupies more cache resources than the standard reactive caching strategy LCE.

Results of the PeRCeIVE Cache Utilization C_A: The cache utilization metric C_A describes the average number of cached copies of a content within the network. In this case, the number of copies for a certain chunk is expected to be low due to the fact that the requested data object is personalized, and thus, the content popularity is low. Figure 5.15 illustrates the results of the cache utilization metric.

Obviously, the absolute minimum value of 0 is reached when all network nodes do not cache any content at all. This is independent of the vehicle's velocity. Besides this finding, the results show that C_A mainly depends on the cache management used, rather than on the caching strategy. This can be seen by comparing the respective numbers: Reactive caching using either LRU, LFU or FIFO as a caching management (C_A : $low = 3.32$, $medium = 3.47, high = 3.52$) occupies fewer cache resources compared to the proactive strategy (C_A : $low = 3.71$, $medium = 3.93, high = 4.31$) using the same cache management mechanism. The increased number of used cache resources by PeRCeIVE is a result of storing the same data chunk at multiple RSU, if the strategy calculates the position of a vehicle between multiple RSUs. Abani et al. [136] describes a possible option to handle the problem of increased cache resource allocation in edge networks caused by proactive caching strategies. By introducing a hierarchical cache decision strategy, data chunks are be placed at neighboring nodes one level deeper in the network topology, instead of stored multiple times at edge nodes.

Summarizing the results for the cache utilization C_A it can be seen that the main influencing factor is the cache management approach used which should be set to a RSU only mechanism when C_A is to be reduced. At the same time, this means that the caching strategy does not have significant effects on this metric.

Results of the PeRCeIVE Resolved `INTEREST` Ratio RR_I: The resolved `INTEREST` ratio RR_I describes the number of resolved `INTEREST`s compared to the overall number of `INTEREST`s sent by the vehicle. The mobility of the vehicle affects this rate due to handovers from one RSU to another. As result of the *mobile node delivery problem*, the response does not reach the

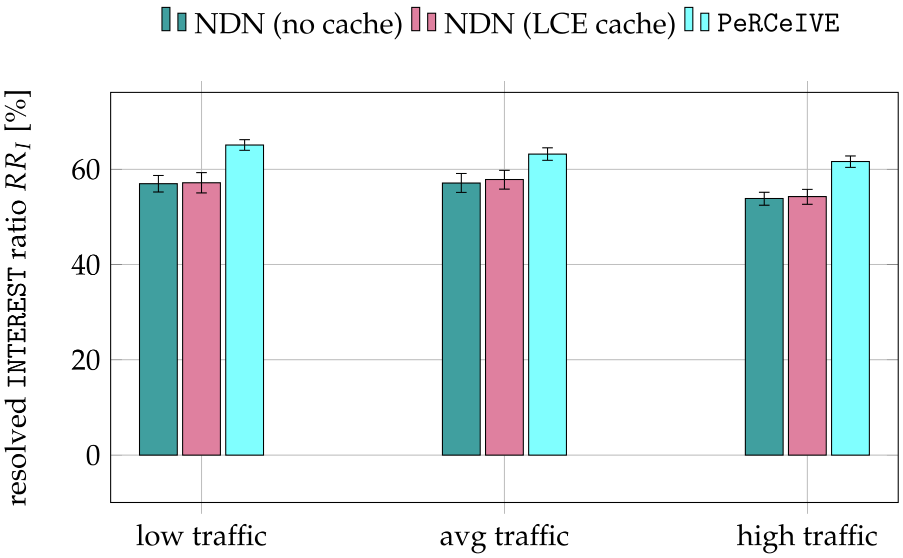


Figure 5.16: PeRCeIVE: Results of the resolved `INTEREST` ratio. PeRCeIVE resolves more `INTEREST` packets compared to the 'no cache' and the standard reactive caching strategy.

mobile node and hence is being repeated. It decreases the resolve rate and thus increases the overall bandwidth used. Since the scenario focuses on personalized data, there is fairly no difference between the values of the 'no caching' and reactive caching strategy.

By placing data chunks at the right nodes, the resolve ratio for the proactive approach is expected to be higher than the reactive ones. The results illustrated in Figure 5.16 confirm this assumption. Using the reactive caching strategy, between 42% (low velocity) and 47% (high velocity) of the `INTEREST`s are not responded by the network which strongly depends on the vehicles velocity and the network topology. This has two reasons: a significant number of `INTEREST` packets get lost when vehicles are not connected to the infrastructure nodes. And, some of the `INTEREST`s are not fulfilled due to the mobile node delivery problem. Comparing this number to the results of PeRCeIVE, the ratio ranges between $RR_I = 61$ and $RR_I = 65$. This values indicate that the proactive placement reduces the impact caused by the mobile node delivery problem and increases the results of this metric.

Results of the PeRCeIVE One-hop Ratio R_1: The one-hop ratio R_1 describes the number of `INTEREST`s sent by the vehicle that were answered directly by the first hop compared to the number of all `INTEREST`s sent by the vehicle. By placing the data at the RSUs before it is requested, the one-hop ratio of PeRCeIVE is expected to be rather high compared to the reactive ones independent of the vehicle's velocity. Figure 5.17 illustrates the results of the one-hop ratio. Due to the fact that the reactive caching mechanisms take action after an `INTEREST` reaches the data provider, the ratio is expected to be 0. However, due to the nature of WiFi connectivity as well as the fact that within the ECo-AT environment two RSU overlap in their communication range, it happens that an `INTEREST` is received by an RSU more than once. As the request is processed by the RSU, the `DATA` becomes available in a one-hop delivery. For the reactive strategy using LRU, LFU or FIFO cache management respectively, these WiFi characteristics result in one-hop ratio values ranging between $R_1 \approx 12\%$ (low velocity) and $R_1 \approx 8\%$ (high velocity). This is different when looking at the results of PeRCeIVE.

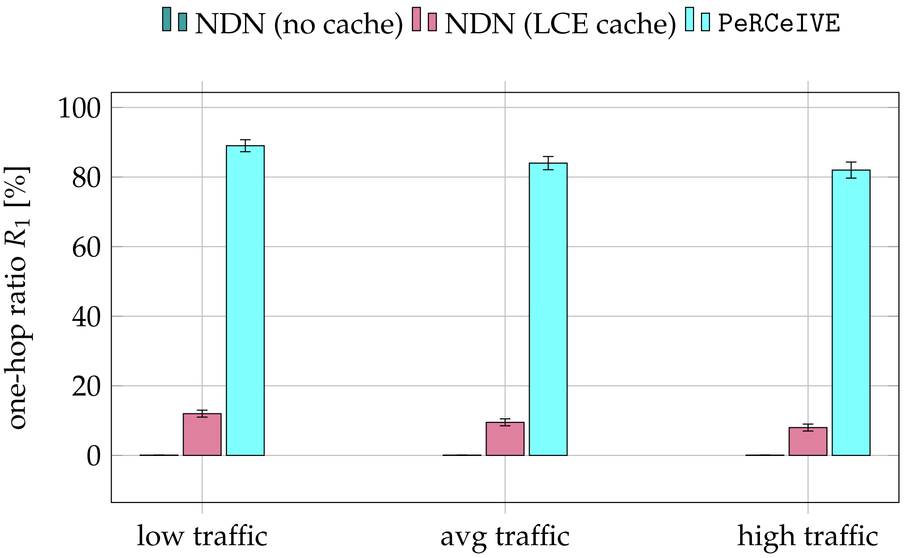


Figure 5.17: PeRCeIVE: Results of the One-hop Ratio evaluation. PeRCeIVE resolves more data chunks directly with one hop by storing them proactively at the edge, so a vehicle can directly access the information.

In this case, almost every DATA is available at the RSU caches before the vehicle sends an INTEREST and thus the one-hop ratio is close to 90% (low velocity).

The results of the one-hop ratio clearly show the benefits of PeRCeIVE compared to reactive caching. The novel proactive caching approach PeRCeIVE increases the efficiency of the data delivery for the class of *personalized* data by reducing latency while providing an improved one-hop DATA delivery.

5.7.2 ADePt

As presented in Section 5.4, the ADePt cache strategy is characterized by a *decentrally managed* and *distributed organization* and focuses on storing **popular**, **transient**, **small** data at the edge of the network.

To evaluate the strategy against the standard cache behavior of NDN (NDN no-cache, standard NDN with LCE placement strategy), simulation runs have been executed for different traffic loads according to the ECo-AT environment. Table 5.5 lists the simulation parameters used to evaluate the ADePt cache strategy.

The DATA packets have a set of initial t_{fresh} values ranging from "500" to "60000" milliseconds cross-tested with the same $T_{popularity}$ values. $T_{vitality}$ has been set to values ranging from "10" to "100". For each simulation run, new mobility patterns are generated (cf. Section 5.6.4), and the combination of each parameter and scenario have been performed (9000 runs in total). When caching is enabled in the simulation scenarios, the cache memory has been chosen sufficiently large so that the cache pressure is not an issue.

Table 5.5: Simulation parameters used to evaluate the ADePt cache strategy.

Parameter	Values
Comm. technology	IEEE 802.11p OCB
Comm. range RSU ($Range_{RSU}$)	150 meters
no. of RSUs involved	8
simulation time t_{sim}	30 minutes
traffic density (low, average, high)	(15, 30, 60) vehicles/min
T_{fresh}	500–60000 milliseconds
$T_{popularity}$	500–60000 milliseconds
$T_{vitality}$	10–100 seconds
Request rate	10–30 seconds
no. of vehicles	450–1800 units
no. of total runs	9000

Assumptions

To focus on the effects of ADePt, the following assumptions are made:

- only V2I communication is taken into account. Direct communication between other vehicles is not considered.
- caches are enabled at all RSU components. All other components are not able to cache any item including the producer, intermediate and mobile nodes.
- the focus is on requesting data from the class of popular and transient, such as *traffic updates* or querying a *parking service*.
- the requested data is commonly visible, hence the content popularity is high.

Within the scenario, vehicles regularly request for popular transient data of a parking service (e.g., "`/vienna/parking/center`") using `INTEREST` packets. A comparison of the ADePt strategy is made using a standard NDN implementation with and without caches deployed at the RSUs.

Results of the `ADePt` Strategy

The results of the ADePt strategy have been evaluated against the standard NDN with and without caching capabilities at the RSU components. All results are measured using the introduced implementation description as well as the simulation environment presented in Section 5.6. The main focus of the metrics to be analyzed are: (i) the Producer Request Load, (ii) the Resolved `INTEREST` Ratio, and (iii) the One-hop Ratio. Since the ADePt strategy is *decentrally managed*, collaboration between neighboring RSU components towards an intelligent replacement of items to keep the number of duplicate copies low is not considered in this thesis. Therefore, the *cache utilization* metric is not examined for ADePt.

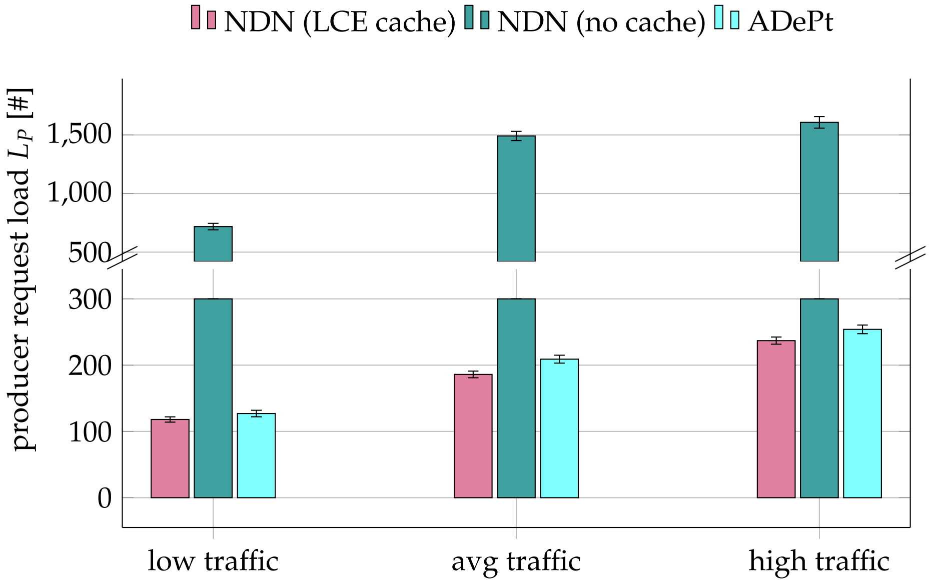


Figure 5.18: ADePt: Total number of requests at the producing application. The ADePt caching strategy slightly increases the load at the data origin, while still being far better than no caching at all.

Results of the ADePt Producer Request Load L_P: The producer request load L_P is defined as the total number of received `INTEREST` packets at a producer application which are directly answered via a corresponding `DATA` packet. As part of the evaluation of ADePt, the request load is used to evaluate the effects of prefetching content from applications which also affects the load of the core network. Figure 5.18 illustrates the results of the producer request load metric over the entire simulation time t_{sim}.

Obviously, the highest values of L_P are reached when caching is disabled at all nodes (e.g., L_P: low traffic $\approx$ 715 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$, avg traffic $\approx$ 1490 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$, high traffic $\approx$ 1600 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$). In this case, all requests are forwarded towards the applications. When enabling the caching capabilities by using the default LCE strategy of NDN, the load at the producer, including the load in the core network, is reduced dramatically (e.g., L_P: low traffic $\approx$ 118 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$, avg traffic $\approx$ 185 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$, high traffic$\approx$ 235 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$).

When looking at the results of the producer request load caused by ADePt, it shows that the strategy increases the load L_P slightly compared to the results of NDN LCE (e.g., L_P: low traffic $\approx$ 120 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$, avg traffic $\approx$ 210 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$, high traffic $\approx$ 250 $^{\text{requests}}/_{\text{t}_{\text{sim}}}$). This result is as expected, because in ADePt it is possible that data is prefetched from the producer which will later not be requested by any vehicle (false positive). Overall ADePt increases L_P by 7.3 to 12.7 % in the simulation scenarios.

As an option to adjust the prefetching aggressiveness, the effect can be minimized by changing the $T_{Popularity}$ threshold of ADePt. The lower the threshold the better the potential service quality for vehicles and higher load is produced in the core network.

Summarizing, it can be seen that the ADePt caching strategy slightly increases the load at the producer, while still being far better than no caching at all, and thus still offers the main benefit of NDN.

Results of the ADePt Resolved `INTEREST` Ratio RR_I: The Resolved `INTEREST` Ratio RR_I is defined by number of resolved `INTEREST` packets compared to the overall number of `INTEREST`s sent by the vehicles. Since a corridor of 2.4km is only covered by RSUs (cf. Section 5.6.3), the high degree of mobility of the vehicles have a significant influence on the RR_I ratio. The situation is even more complicated, due to the fact that the physical layer of the IEEE 802.11p standard is tuned up the whole time [35]. There is no option for a vehicle to determine if there is any connection to an RSU. In this case, the vehicles send out `INTEREST` packets to the network, even if there is no connection to an infrastructure node. Therefore, a significant number of `INTEREST` packets remain unanswered. By prefetching data items at the right nodes, the resolved `INTEREST` ratio for the ADePt strategy expected to be higher than the reactive one of NDN, however, not lower than the reactive strategy.

The results are illustrated in Figure 5.19. The number of `INTEREST`s resolved by the reactive NDN LCE strategy is slightly higher than the values when caching is disabled at all nodes. This is due to the fact that some `INTEREST` packets are fulfilled by cached `DATA` at the RSUs as well as the items have a short lifetime within the caches. The results of the NDN LCE caching strategy ($RR_I \approx$ 54% (avg traffic), and $RR_I \approx$ 51% (high traffic)) compared to the results of the ADePt strategy ($RR_I \approx=$ 55% (avg traffic), and $RR_I \approx$ 53% (high traffic)), it can be seen that the ADePt increases the number of delivered `DATA` slightly.

Regarding the effects of the varying number of network participants, it can be seen that the number of the RR_I decreases while increasing the number of participants in the network. In this case, the limitation is the congested physical and medium access layer. Requests for data items get lost on the wireless channel and thus `DATA` is not delivered by the network. Therefore, in such an overload scenario `ADePt` performs almost similar to standard NDN with LCE cache strategy, as the large number of constantly generated `INTEREST`s by consumers will request for new data.

Summarizing the results of the RR_I for ADePt, it shows that the number of delivered `DATA` packets have increased slightly. The improvement in this evaluation are rather incremental, because the used simulation framework is optimistic regarding latency and mainly takes transmission and propagation latency values into account, but does not simulate processing latency values that would increase the multi-hop penalty in a real NDN.

Results of the ADePt One-hop Ratio R_1: The One-hop ratio R_1 is defined by the number of resolved `INTEREST` packets which have been directly answered by network nodes one hop away divided by the number of all `INTEREST` packets sent by the consumer.

By actively loading popular, transient data from the core network into the caches of RSUs using the ADePt strategy, the R_1 ratio is expected to be higher than the values of the default reactive caching of NDN, independent of the number of participants in the network. Figure 5.20 illustrates the results of the one-hop ratio.

Regarding the One-hop ratio results of the NDN LCE cache strategy, the ratio ranges between $R_1 \approx$ 41% (low traffic) and $R_1 \approx$ 44% (high traffic). When looking into the results of ADePt, the ratio ranges between $R_1 \approx$ 49% (low traffic) and $R_1 \approx$ 53% (high traffic volume). The relative change in the One-hop ratio between standard NDN and ADePt shows that the active placement of data items increased the R_1 values between 7 – 9 % for transient, popular data within the simulation environment.

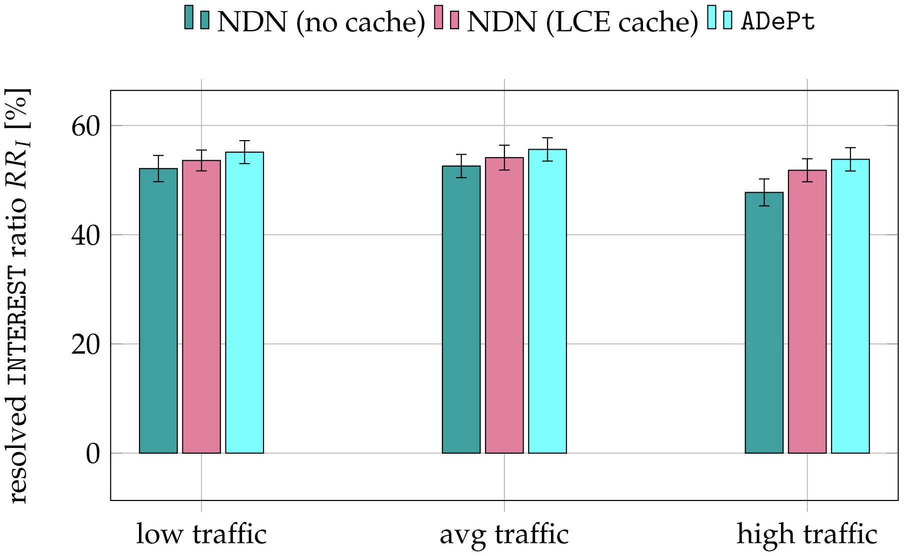


Figure 5.19: ADePt: Results of the resolved INTEREST ratio. The ADePt caching strategy increases the number of delivered DATA slightly compared to standard NDN.

Summarized, the results of the one-hop ratio shows clearly the benefits of ADePt compared to the reactive caching of NDN. By actively fetching popular, transient data items at infrastructure nodes close to consumers, the delivery times of data items is decreased, while the number of requests directly answered by the first hop is increased.

Discussion of the ADePt parameters: As part of the evaluation of ADePt using simulations, different values for the strategy parameters have been taken into account to explore the sensitivity of these parameters. As one of the results, it can be seen that the INTEREST frequency f_I is dependent on the number of current vehicles, the application and the type of the requested data. The value of $T_{popularity}$ is highly dependent on f_I. The evaluation has shown that small values for $T_{popularity}$ increases the number of prefetching INTEREST packets, while a large value never triggers the ADePt functionality. In case of larger values, ADePt achieves the same results as the standard NDN LCE caching strategy. Similar effects can be explored by adjusting the value for $T_{vitality}$. Small values lead to a belated fetching of new data items which reduces the number of satisfied one-hop ratio INTERESTs to the same values of standard NDN. While large values lead to an increasing number of INTEREST packets overloading the core network and therefore unnecessary fetched DATA packets. The best results are achieved by ADePt when the vitality threshold equals the freshness time ($T_{vitality} = t_f$) for each individual data item in the network. The introduction of an individual vitality threshold for each data item can be adopted by ADePt easily in future versions of the strategy.

The results within the ADePt section have shown that the adaptive, distributed content prefetching strategy increases the efficiency of the data delivery by reducing the latency, while responding directly at the next hop.

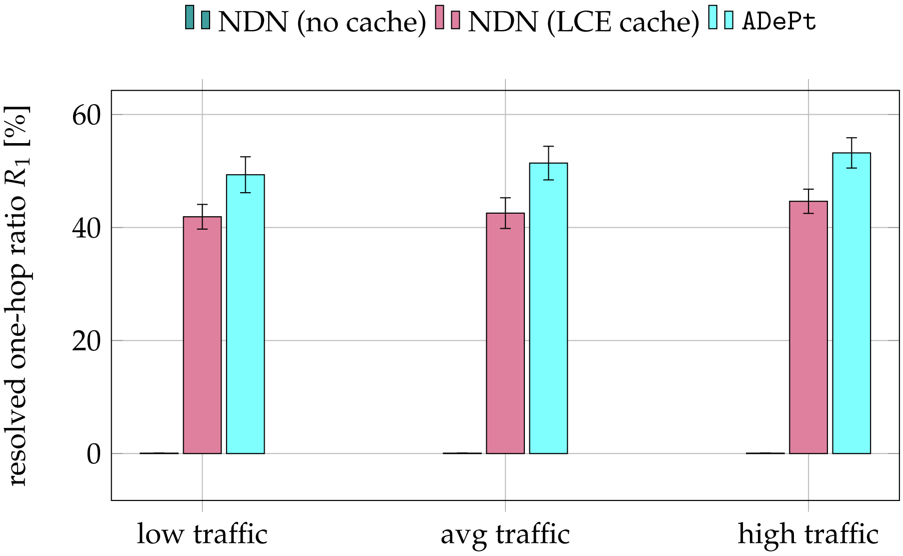


Figure 5.20: ADePt: Results of the One-hop Ratio evaluation. By actively fetching popular, transient data items using the ADePt caching strategy at infrastructure nodes close to consumers, the delivery times of data items is decreased, while the number of requests directly answered by the first hop is increased.

5.7.3 Predictive Data Prefetching

The following paragraphs present the generation process of a synthetic communication data basis required to evaluate the concept of the ML-based predictive prefetching approach, as well as the results of the approach within the ECo-AT simulation environment.

Generation of a synthetic communication basis: As presented in Section 5.5, the predictive prefetching approach combines techniques from the ML domain using principles of ADePt such as the monitoring of local information at the network node. However, as there is no vehicular ICN network deployed at the present time which can be used to monitor the communication behavior of real world automotive applications, a synthetic data basis has to be created to form the basis for the evaluation of the prefetching approach using simulations.

Based on the recorded communication behavior of different real world applications (cf. Section 5.6.2), valuable information from request and response messages have been extracted and transferred from IP world into the ICN world. For example, this includes:

- the *URI* of the IP packet forms the basis for a name component in ICN.
- *timestamps* of requesting and receiving packets to resolve the communication behavior (e.g., periodicity of `INTEREST`s to be sent by an application).
- *content length and type* for size and type of the payload of a `DATA` packet.
- *data expiration* values are used for declaring the freshness of a `DATA` packet
- ...

Table 5.6: Simulation parameters used to evaluate the ML-based predictive prefetching strategy.

Parameter	Values
Comm. technology	IEEE 802.11p OCB
Comm. range RSU ($Range_{RSU}$)	150 meters
no. of RSUs involved	8
t_{fresh}	6–1800 seconds
payload sizes	1000–8000 bytes
Request rate of applications	1–60 seconds
no. of vehicles	1600–4200 units
no. of total runs	5000

Based on the recorded communication behavior and the constructed ICN packets, synthetic traffic traces are created using the ECo-AT simulation environment. ndnSIM reference applications have been modified to emulate the recorded communication behavior extracted from the IP world. As a result, the following consumer and producer applications have been created:

- *Navigation Application*: emulates the communication behavior of an online navigation application by using three different types of request messages: *route data* – providing information of route options, *map tiles* – used to present the map, and *miscellaneous data* – providing additional information of the route ahead such as points of interest, etc. Each type of a message is requested based on changes in the position of the mobile node. It represents a class of applications providing popular, parts of transient/static data of varying size.

- *Update Application*: emulates the communication behavior of a software update application and follows the structure introduced by Tschudin et al. [254]. The application uses two types of request messages: *manifest* – containing metadata for a group of accompanying stream segments, and *stream segments* – containing the actual content. The requesting order is given by a requests for streaming manifests, followed by requests for stream segments. It represents a class of applications providing individual, static data of varying size.

- *Entertainment Application*: emulates the communication behavior of a broadcasting application following the behavior of the update application using *manifest*, and *stream segments* packets. The requesting order is given by a requests for streaming manifests, followed by requests for stream segments. It represents a class of applications providing popular, transient, small data.

During the generation of the synthetic traffic traces, monitoring components are installed in the simulation environment on all nodes – incl. consuming/producing nodes, and infrastructure nodes such as RSUs – to record all means of communication. Simulation runs for a synthetic traffic volume within 24 hours have been executed, resulting in a large data basis used to train the ML-based prefetching models.

Assumptions and Simulation Parameters

To focus on the effects of the predictive prefetching approach, the following assumptions are made:

- only V2I communication is taken into account. Direct communication between other vehicles is not considered.
- caches are enabled at all RSU components. All other components are not able to cache any item including the producer, intermediate and mobile nodes.
- the requested data items are following the introduced class of applications, hence the content popularity is high.
- each roadside unit is tagged by a unique node identifier ID_{node}. The identifier is used to mark an `INTEREST` packet overheard by the node.
- each roadside unit is able to record the time an `INTEREST` or `DATA` is overheard by the node itself.
- each roadside unit provides a trained ML model *containing historical information of overheard ICN packets*. The roadside unit is able to consult the model to make prefetching decisions by the node itself.

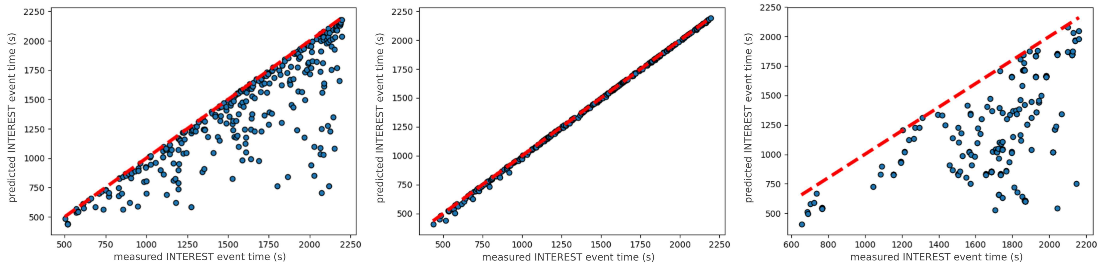
Table 5.6 provides an overview of the simulation parameters. The results of the predictive prefetching strategy are created using the simulation environment presented in Section 5.6 and are evaluated against the standard NDN with and without caching capabilities at the RSU components. Depending on the application, the `INTEREST` names are chosen according to the communication behavior. For example, there is a limited set of names for requesting for a specific software update, distributed randomly across the mobile nodes. Each roadside unit holds a trained ML model based on historical data and is independent of the actual simulation runs. The `DATA` packets have a set of initial t_{fresh} values ranging from "6" to "1800" seconds. Simulation runs have been made for caching scenario including standard NDN with LCE placement strategy, and the predictive prefetching strategy. For each simulation run, new mobility patterns are generated (cf. Section 5.6.4).

The main focus of the metrics to be analyzed are: (i) the Producer Request Load, (ii) the Resolved `INTEREST` Ratio, and (iii) the One-hop Ratio. Since the predictive prefetching strategy is deployed in a *decentrally managed* fashion, collaboration between neighboring RSU components towards an intelligent replacement of items to keep the number of duplicate copies low is not considered. Therefore, the *cache utilization* metric is not examined.

Results of the Predictive Prefetching Strategy

The goal of the predictive prefetching approach is to learn and to forecast the appearance of consumer requests to prefetch the right content into caches at the edge of the network, and therefore, to reduce data delivery time.

Before evaluating the predictive prefetching approach according to the presented metrics, an investigation of the accuracy of the forecast of `INTEREST` events using linear regression model is made. Such investigation is used as an indicator for the simulation results.



(a) Navigation application **(b)** Entertainment application **(c)** Update application

Figure 5.21: Extracts of `INTEREST` event time predictions using linear regression models for representatives of the different traffic classes: (a) navigation application, (b) entertainment application, and (c) update application. The red line indicates a perfect event prediction.

Figure 5.21 illustrates the results of the accuracy investigations. The results ranges from being accurate – for the class of entertainment applications, the model almost predicted every `INTEREST`s occurrence (Figure 5.21b) correctly – to being inadequate with respect to the results of the update application (cf. Figure 5.21c).

As presented in Figure 5.21, event requesting for geo-specific content as well as following a consecutive message flow is easier to predict by the linear regression model. While requesting events for decoupled data events (e.g., personalized data such as application/hardware updates) are hard to predict by the model, it is still worth to be prefetched to be available for a wider range of consumer, and thus, reduce the load in the core network.

Results of the Producer Request Load L_P for Predictive Prefetching: The producer request load L_P describes the total number of received requests which are directly answered with a `DATA` packet. Similar to the evaluation of ADePt, the request load shows the effects of prefetching content from the applications within the core network, also affecting the load in the core network. Figure 5.22 shows an extract of the results of the producer request load within a time box of five hours using the mobility traces of the ECo-AT environment.

When looking into the results, the highest values of L_P are reached for the application classes online navigation and software update (e.g., L_P: navigation (3h) $\approx$ 760 requests/h, update (3h) $\approx$ 720 requests/h) representing an increasing volume between 6% – 13% compared to the default LCE strategy of NDN.

One reason for the increased volume of requests is that prefetching events triggered more frequently at the RSU nodes than required. The reason for early prefetching triggers relies in the trained model which is responsible for decision making. Prefetching events are triggered based on an `INTEREST` forecast, however, the request is not overheard afterwards.

Another interesting observation is given when looking to the L_P results of the update applications. As presented in Figure 5.21c, individual-related data is hard to be predicted by the model, and thus, the results are expected to be worse than the results of, e.g., the entertainment application. However, the results of the update application are not below the producer request load values of the the standard NDN.

Summarizing, it can be seen that the predictive prefetching strategy increases the load at the producer independent of the class of application considered. More historical data is required to improve the quality of decision making.

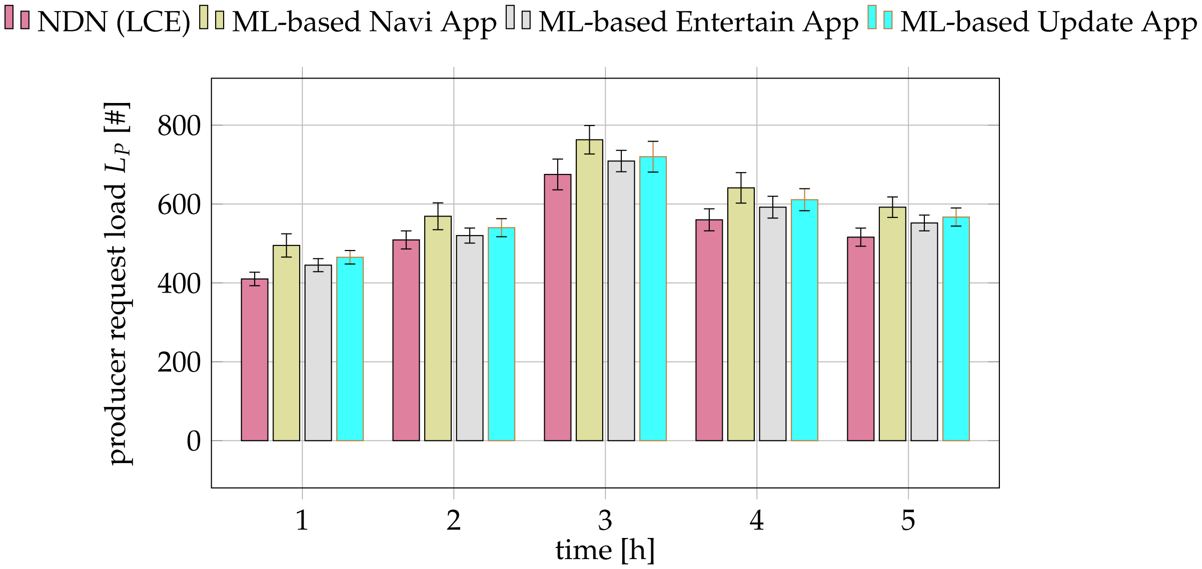


Figure 5.22: Exemplary results of producer request load L_P of the predictive prefetching approach for each class of applications compared to the standard NDN LCE cache behavior. Irrespective of the class of applications, the predictive prefetching approach results in a higher producer load.

Results of the Resolved `INTEREST` Ratio RR_I for Predictive Prefetching: The resolved `INTEREST` Ratio RR_I describes the number of `INTEREST` packets for which a corresponding `DATA` packet has been received, compared to the overall number of `INTEREST` packets sent by the mobile node. By using principles from the ML world, the resolved `INTEREST` ratio for the predictive prefetching approach is expected to be higher by fetching data items into the caches, compared to the reactive one of NDN.

The results are illustrated in Figure 5.23. Looking into the results of resolved `INTEREST`s of the predicted prefetching approach it can be seen that the values are slightly higher (RR_I $\approx$ 48% (2h entertainment application), and $RR_I \approx$ 46% (2h navigation application)) than the results of the NDN LCE caching strategy ($RR_I \approx$ 45% (2h)). This is due to the fact that more `INTEREST` packets are fulfilled by prefetched `DATA` at the RSUs.

Regarding the effects of the increasing number of mobile participants, it can be seen that the number of resolved `INTEREST` slightly increases over time (e.g., total number of nodes 1h $\approx$ 1800$units$ and 3h $\approx$ 3800$units$). However, similar to ADePt, an increasing number of participants result in more data losses on the wireless channel due to limitation of the congested physical medium access layer. While it is expected that the ML-based approach performs almost similar to standard NDN with LCE cache strategy, it is not the case for the results of the predicted `INTEREST`s of the update application. It shows that prefetching wrong `DATA` items at the wrong time slightly decreases the resolved `INTEREST` metric compared to standard NDN.

Summarizing the results of the RR_I shows that the number of delivered `DATA` packets have increased slightly (except of the class of update applications). The evaluation has shown minor improvement regarding the resolved `INTEREST` packets.

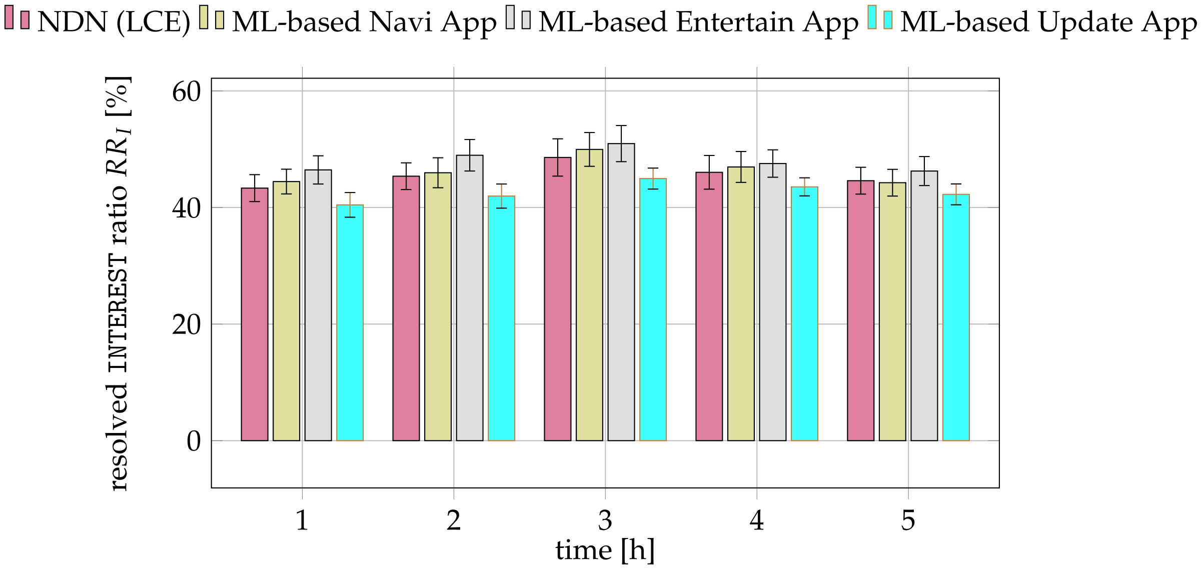


Figure 5.23: Exemplary results of the resolved INTEREST ratio RR_I of the predictive prefetching approach for each class of applications compared to the standard NDN LCE cache behavior. The values of the predictive prefetching approach are slightly higher than compared to the standard NDN, except for the class of update applications.

Results of the One-hop Ratio R_1 for for Predictive Prefetching The one-hop ratio R_1 describes the number of resolved INTEREST packets which are directly answered by neighboring nodes within the first communication hop divided by the number of all INTEREST packets sent by the consumer.

By loading DATA items into caches at the edge based on the prediction of occurrence of INTEREST packets, the R_1 ratio is expected to be higher than the values of the standard, reactive caching strategy of NDN. Figure 5.24 illustrates the results of the one-hop ratio.

Regarding the one-hop ratio results of the NDN LCE cache strategy, the ratio ranges between $R_1 \approx 38\%$ (1h) and $R_1 \approx 42\%$. When looking into the results of the predicted prefetching approach, the ratio ranges between $R_1 \approx 39\%$ (1h) and $R_1 \approx 42\%$ (3h) of the class of online navigation and $R_1 \approx 41\%$ (1h) and $R_1 \approx 45\%$ (3h) of the class of entertainment applications. The results show a minor improvement in the data delivery ranging between 2 – 3 %, except for the results of the class of update applications which are close to the values of the standard NDN cache strategy.

Summarized, the results of the one-hop ratio shows minor improvements of predicted prefetching approach compared to the reactive caching of NDN.

Discussion of the results of the ML-based prefetching approach: The results in the previous sections have shown minor improvement of the data delivery when using techniques from the domain of ML. One reason for that is the relatively small data basis used for training. The predicted occurrence of an INTEREST packet was often out of the actual monitored time window. Based on the real world but limited mobility traces of the ECo-AT environment (only available for a week), the available traffic traces are not sufficient large enough to generate synthetic data basis for multiple weeks in order to train the machine learning models.

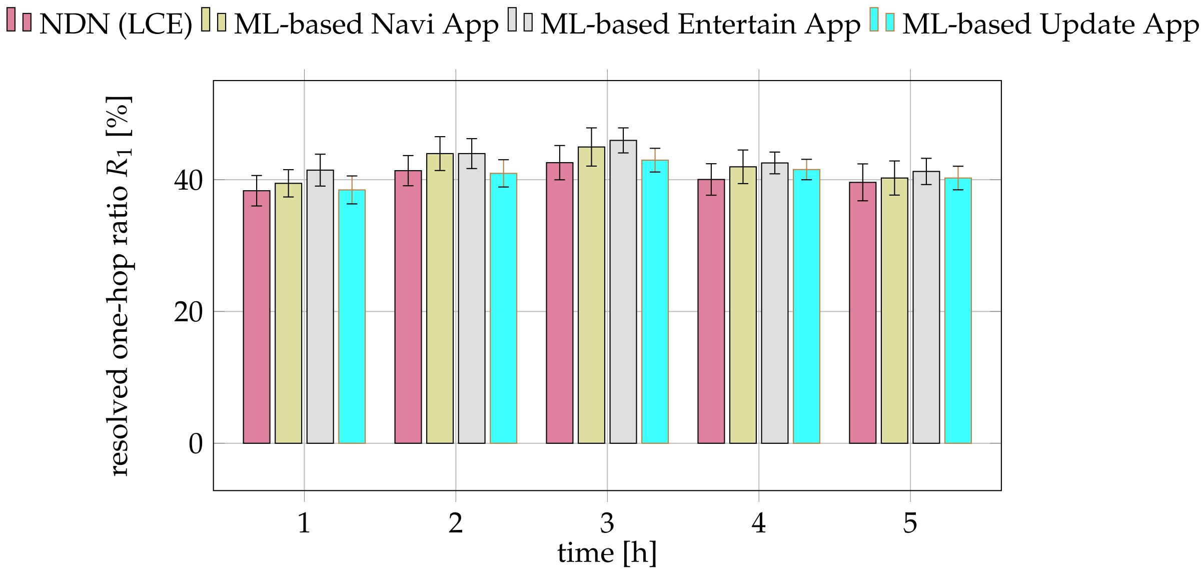
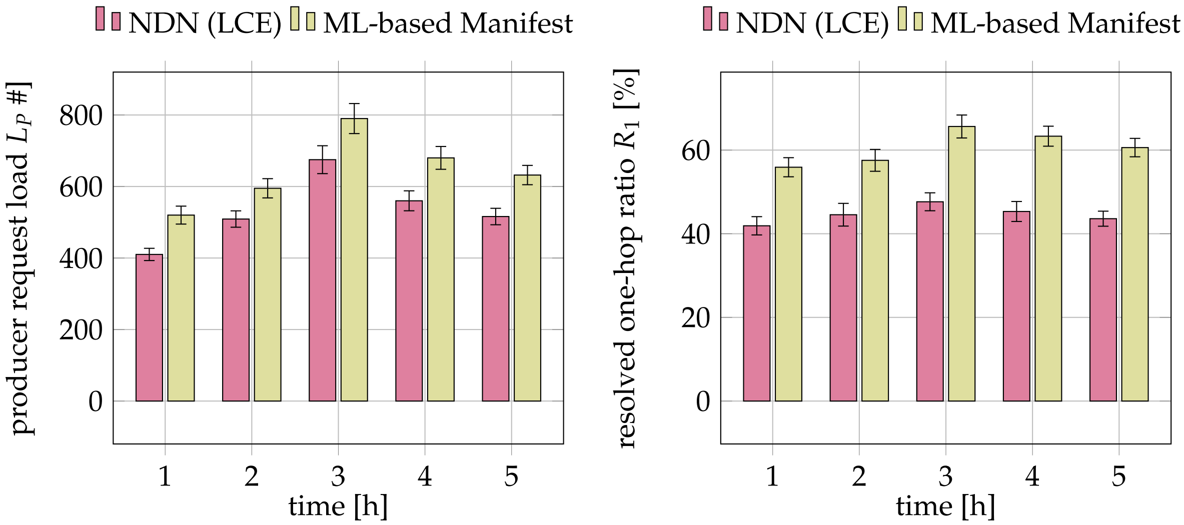


Figure 5.24: Exemplary results of the one-hop ratio R_1 of the predictive prefetching approach for each class of applications compared to the standard NDN LCE cache behavior. The results of the predictive prefetching approach are slightly higher or closely even to the values of the standard NDN LCE cache behavior.



(a) Producer request load for predictive manifest prefetching of an update app.

(b) One-hop ratio for predictive manifest prefetching of an update app.

Figure 5.25: Extract of the producer request load and the one-hop ratio when predicting and prefetching manifest packets including their content for the class of update applications. While the number of requests at the producer increases, the number of packets directly answer with one hop is increased significantly.

Potential option to face the challenge of wrong `INTEREST` forecasts are enhancements of the prediction procedure using *classification* methods to group types of requests in order to make better predictions, for example, using support vector machines (cf. Burges [273] for comprehensive tutorial on support vector machines).

An investigation of the predicted prefetching results of the class of update applications shows that the prediction of `INTEREST` packets requesting for manifest content worked accurate, while the prediction of requests for the corresponding content of the manifests – e.g., the actual stream segments – resulted in an inadequate prediction (e.g., prediction of `INTEREST` names which were never overheard by the RSU node). As a result, the mobile nodes have to request the stream segments from the core network. Such behavior increases the delivery time, and thus, increases the probability that mobile nodes never receive the `DATA` packet in time.

An enhancement of the decision making and prefetching procedure for manifest based requests is proposed. Based on the assumption that the content of a manifest is not encrypted (maybe not the case for payed services such as Netflix), the decision making is enhanced to extract the content of the manifest by using the information as input for the prefetching procedure. As a result of the manifest prediction, the content is made available for downloading at the RSUs by prefetching it from the core network proactively. Figure 5.25 illustrates the results of the enhanced decision making and prefetching procedure for the metrics producer request load L_P and the one-hop ratio R_1.

When looking to the results of the producer request load (cf. Figure 5.25a), it can be seen that the modifications of the prefetching procedure resulted in a higher load at the producer, and thus, in the core network. This is due to the fact that the prefetching triggered more frequently by loading `DATA` packets into the caches of the RSUs, maybe not collected by the mobile nodes in the same frequency. However, when looking to the results of the one-hop ratio R_1, it can be seen that the modifications have improved the direct delivery of `DATA` packets significantly. While the traffic in the core network increased between 17 – 24% in the simulation environment, the number of `INTEREST`s directly answered by the RSU in the vicinity has been improved by 12 – 15 %.

Summarized, the results of the predictive prefetching strategy have shown minor improvements in the data delivery. Modifications of the strategy have shown the potential of ML-based prefetching approaches, however, require a sufficiently large data basis as well as some fine tuning of the parameters. To improve the result of the forecast of `INTEREST` packets, other techniques from the domain of ML such as support vector machines can be used.

Summary

This section presented the evaluation results of the novel caching strategies to reduce the effects of the *mobile node delivery problem* (cf. Section 5.1). The presented proactive caching strategies address at least one class of automotive data traffic (see Table 5.2 for deployment scope of the strategies). The results ranges from incremental performance improvements (e.g., predictive prefetching approach) up to significant improvements of the data delivery (e.g., PeRCeIVE w.r.t. the class of personalized, large data), by showing strengths regarding one of the automotive traffic classes presented in Section 5.1.1.

5.8 Towards a Real World Vehicular ICN Testbed Environment

As mentioned in Section 5.6 "Simulation Environment", there are several methods available to evaluate the performance of a network concept in computer science. While simulations offer the possibility to investigate and compare concepts in scale and against the state-of-the-art, it also has its limitation – the lack of real world hardware and realistic condition investigations. This section presents the work of building a real world ICN demonstrator supporting the 5th generation of Intelligent Transportation System Standard of the European Telecommunications Standards Institute (ETSI ITS-G5) based on the IEEE 802.11p standard for wireless access in vehicular environments.

5.8.1 Integrating ICN into the ETSI ITS-G5 Station Architecture

In the last decades, different standards and architectures of ITS have been developed and published around the globe. This includes communication technologies such as cellular (in licensed spectrum) as well as WiFi based solutions (in unlicensed spectrum) as presented in Section 2. DSRC describes a class of technologies for vehicular ad-hoc networking. Today, there exists three variants: (i) IEEE 1609 alias WAVE [38] in North America, (ii) ETSI ITS-G5 [39] in Europe, and (iii) variants in Japan [6]. However, all standards have in common that they share the same access layer technology - the IEEE 802.11p [35] standard.

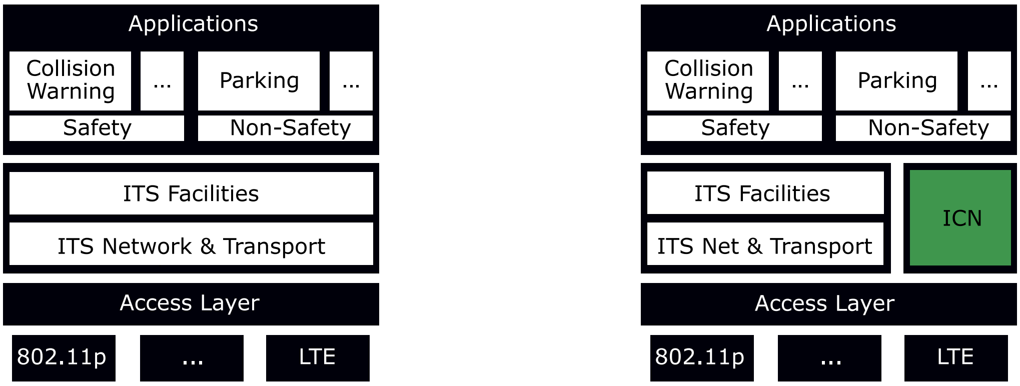
Setting up a real world vehicular prototype supporting an ICN protocol stack requires the IEEE 802.11p as the underlying access medium. Furthermore, the concept presented in the following sections deals with the integration of ICN into the ETSI ITS-G5 inter-vehicle communication stack. Figure 5.26a illustrates a legacy inter-vehicle communication stack supporting the ETSI ITS-G5 station architecture.

The integration concept considers interest-based ICNs as a complement to the existing supported components of the ETSI ITS-G5 station protocol stack. Figure 5.26b illustrates the process of integrating ICN into the stack. Existing safety-critical applications are still supported using the ITS-G5 stack, while ICN can be used by non-safety applications for V2V and V2I communication. The data-oriented fashion of ICN and the resulting in-network caching and processing capabilities leverage the access to information for other participants in the network. Another benefical feature of integrating ICN into an inter-vehicle communication stack is described by the data-centric security of interest-based ICN architectures (e.g., [12]). It provides new opportunities to manage the access to information.

However, such an approach also describes some challenges: For example, the management of the network. A separated integration of ICN as a complement may result in message collisions when using the plain access technology, while mechanisms to maintain the network stability still exists (e.g., the Decentralized Congestion Control (DCC) mechanism as part of the ITS-G5 access layer). As part of this concept, it is proposed that ICN has the option to join the ETSI ITS-G5 stack (e.g., add ICN support in the DCC mechanism to disseminate information across the network in a managed fashion) and complement existing functions if required.

5.8.2 A Prototype Implementation

Based on the introduced integration concept, a prototype implementation is created, separated into three blocks: (i) support of the IEEE 802.11p access layer, (ii) support of the ETSI ITS-G5 facilities to be able to exchange message according to the standard, and (iii) the support of



(a) The legacy ETSI ITS-G5 protocol stack and its architectural layers.

(b) The ETSI ITS-G5 protocol stack supporting ICN.

Figure 5.26: Concept illustration of integrating an ICN protocol stack into an inter-vehicle communication system supporting ETSI ITS-G5 based on [39]. The ICN stack complements the network and transport layers of the ITS-G5 stack to increase the efficient data dissemination for non-safety applications.

ICN message exchange using the NDN protocol stack. The following subsections describe the building blocks of the prototype in detail. Figure 5.27 illustrates the feature set of the prototype implementation supporting the exchange of data items using both ETSI ITS-G5 and ICN protocols in parallel.

IEEE 802.11p Access Layer Support

The entry point to the wireless communication access layer is supported using the IEEE 802.11p standard. In this prototype implementation, the access layer is based on a Linux kernel modification for the Atheros 9k WLAN chip series from the Czech Technical University of Prague [274]. The patch enables the chip to run in the IEEE 802.11p OCB mode as part of the 5.9 GHz band. While the IEEE 802.11p standard defines several 10MHz service channels for exchange information, the ITS-G5D band (5905 MHz - 5925 MHz frequency with 10MHz channel spacing) and the service channel G5-SCH5 are used in this prototype implementation. These service bands are reserved by the standard for non-safety and future applications [275]. Based on these modifications, the prototype is able to send and receive messages physically from and to the access layer. Afterwards, these packets are forwarded to the ITS-G5 platform for further processing.

The `OpenC2X` ICN Module

In order to build a low-cost prototype, the communication stack is based on the *OpenC2X* platform [276] – an open source experimental and prototyping platform supporting ETSI ITS-G5. The architectural design of the platform aims to be flexible by providing each of the supported features as modules. For example, the implementation of the DCC algorithm as well as protocols such as CAM are provided as individual service modules. In order to investigate different parts of a prototype, it is possible to modify or replace modules of the platform. While each service is executed in a dedicated thread, information exchange between these modules are realized using the asynchronous messaging library ZeroMQ [276].

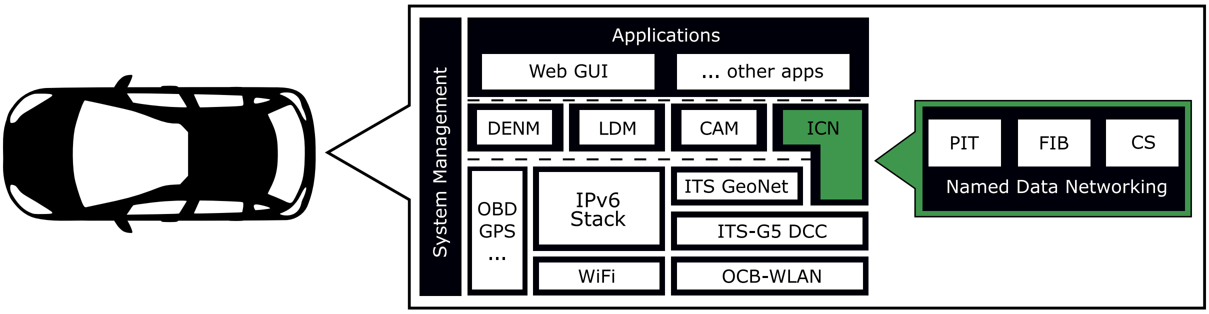


Figure 5.27: Integration of an ICN module (green boxes) in the OpenC2X platform. As part of the prototype, NDN packets are directly transmitted via the ITS-G5 DCC layer by default. Forthermore, packets can also be transmitted over IP and classic WiFi.

In the prototype implementation, the support of ICN message exchange is created using the NDN project platform [183] including the ndn-cxx C++ library (in version v0.6) as well as the NFD. By using the access layer provided by the Atheros 9k WLAN chip series, the implementation of the communication stack is created by a new module of the OpenC2X platform supporting the exchange of ICN messages including a binding to the NDN platform.

Following the modular design of OpenC2X, multiple options are available to create an ICN service module:

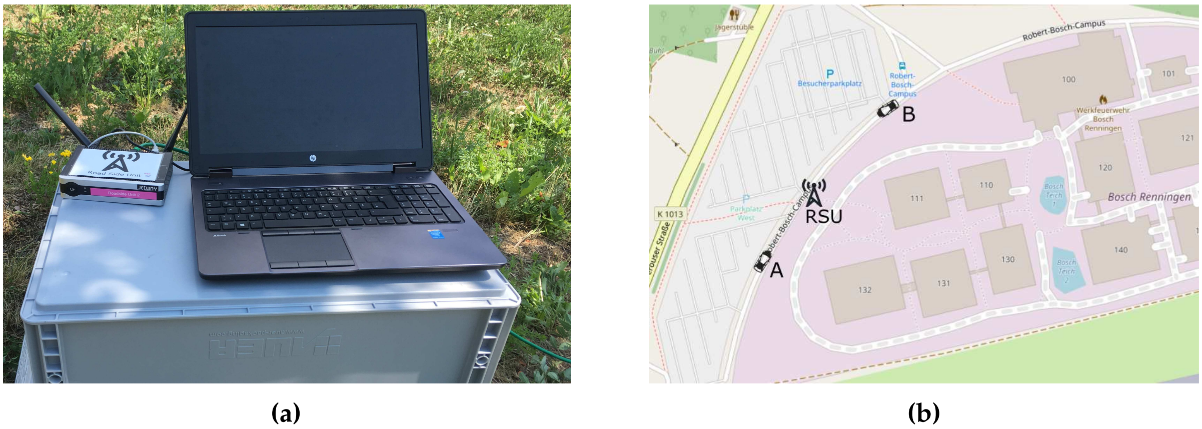
O1 A binding between the NFD engine and the asynchronous message queue of the OpenC2X platform, used to directly send and receive packets to the underlying OpenC2X platform layers.

O2 An application-based gateway leveraging the programming interfaces of the NDN platform.

O3 connect both platforms using network sockets.

While the first to options (O1 and O2) introduce drawbacks such as incompatibility of later versions of the messaging library or inconsistency with future version of NDN, the most suitable option is described by using network sockets (O3). This option has two major benefits: First of all, it is aligned to the forwarding procedures of the vanilla NDN implementations. Second, the option can easily adapt changes to upcoming versions of NDN or OpenC2X . Figure 5.27 shows the final structure of the prototype implementation.

Regarding the packet processing in the prototype, incoming and outgoing packets are processed as follows: Packets are received via the IEEE 802.11p physical and medium access layer and forwarded towards the ETSI ITS-G5 DCC module of OpenC2X . It forwards the packet via the message queue towards the next processing module according to the message type extracted from the packet. In case the packet is marked to carry NDN messages, the ICN module forwards the packet to the local NDN forwarding engine. Packets sent from local NDN applications are received by the ICN module and forwarded towards the OpenC2X DCC module using the message queue system. Finally, the DCC module transfers the packet to the access layer and therefore through the network.



(a) **(b)**

Figure 5.28: Functional tests of the prototype: (a) Illustration of a RSU deployment at the road side. For measurement purposes, a labtop PC is used to monitor the traffic flows. (b) The testbed deployed next to the Bosch research campus in Renningen, Germany.

5.8.3 Functional Tests of the Prototype on the Road

Based on the prototype implementation, functional tests have been made on the road. In order to demonstrate the in-network caching benefits of ICN in connected vehicle environments, two scenarios have been defined as part of the functional tests:

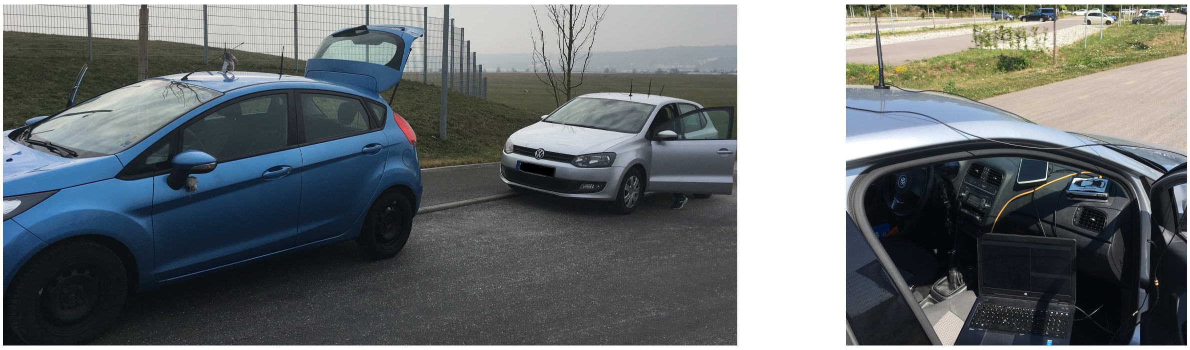
S1 **Infrastructure assisted in-network caching**: The availability of popular data items is increased by caching such items at infrastructure nodes such as RSUs. In this scenario, V2I communication relation is considered.

S2 **SCF assisted in-network caching**: Passing vehicles store and carry data items towards consumers which are present in areas uncovered by any infrastructure node. In this scenario, V2I and V2V communication is considered.

Scenarios

The first scenario (S1) is based on the *electronic horizon* use case which has been introduced in Section 1.2.1. As part of this use case, a consumer is interested in receiving information about available parking sports nearby. The prototype implementation is used to increase the availability of parking information at the edge of the network. In this case, a consumer (cf. Figure 5.28b Car A) sends out an `INTEREST` packet requesting for parking information nearby. Such information is offered by a parking provider accessible via the installed RSUs (e.g., Figure 5.28a). After sending the information towards the consumer, the local RSU cache is used to store the information for upcoming vehicles. In order to prefetch the content for subsequent vehicles (cf. Figure 5.28b Car B), the RSU runs the ADePt approach to provide directly access to the information from the RSU cache.

The second scenario (S2) is based on the *community-based sensing* use case, introduced in Section 1.2.2. In this scenario, a passing vehicle (e.g., part of sensing community) helps to increase the availability of parking information by storing and ferrying data into infrastructure uncovered areas. By deploying the prototype implementation on vehicles, they are able to act as data mules, providing access to information for other participants. First, a vehicle (cf. Figure 5.28b Car A) receives `DATA` packets while passing a RSU (e.g., Figure 5.28a).



(a) **(b)**

Figure 5.29: In-vehicle deployment of the prototype: (a) In the scenarios (S1 and S2), two vehicles are equipped with IPC boards running the prototype implementation. (b) Illustration of the in-vehicle deployment of the prototype. Two antennas are installed on the roof of the vehicle, while the IPC board as well as the power battery are installed inside the vehicle.

As a next step, the vehicle leaves the communication range of the RSU and carries the `DATA` packets within its local cache. Some seconds later, approaching vehicles (cf. Figure 5.28b Car B) wants to receive the same `DATA` packets from the network. In this case, car B asks the passing car A for the desired information. Instead of waiting to receive the packets from the RSU, car A directly provides the `DATA` packets from the local cache.

Bring the Prototype on the Road

The prototype is set up nearby the Bosch research campus in Renningen, Germany (cf. Figure 5.28b). The setup consists three components (IPC Board NF36-2600 - 1GHz CPU, 1GB RAM), two in-vehicle deployments (cf. Figure 5.29a) and one deployed as an infrastructure node at the road (e.g., Figure 5.28a). The in-vehicle deployment consists of one IPC board running the prototype implementation, one battery to power the IPC board and two antennas (cf. Figure 5.29b). The RSU component is directly connected to a parking provider service (Raspberry Pi Model 3), acting as a producer of parking data. For measurement and controlling purposes, laptop PCs are used at each of the prototype components to monitor the traffic flows. The communication range of the RSU is about 50 meters, while the communication ranges of the cars are limited to approx. 30 meters. During the tests, the position of the RSU is fixed, while the cars are traveling around the campus with a velocity approx. 20-30 km/h.

In order to exchange NDN messages, reference applications are used on the vehicle nodes frequently requesting for information with a period of 5 seconds. On the parking provider component, a NDN application offering parking information is deployed. On each of the components, a logging system has been deployed to measure the requests and responses. During the tests, the NDN applications as well as ETSI ITS-G5 CAM messages have been exchanged in parallel using the prototype platform. In order to trigger the ADePt functionality and to prefetch content from the parking provider, the popularity threshold value $T_{popularity}$ is set to a minimum.

Results of the Functional Tests

As part of the first scenario (S1), in-network cache functionality is disabled at the mobile nodes as well as the producer. Only at the RSU, cache functionality is enabled and configured to be sufficiently large so that cache pressure is not an issue. While passing the RSU, vehicle nodes send out `INTEREST` packets requesting for parking information periodically, according to the scenario description of S1. An analysis of the measurement traces at the producer has shown that the number of received `INTEREST` packets is low compared to the received `DATA` packets at the cars. This is due to the fact, that the RSU component caches forwarded `DATA` packets. In case the `DATA` is still valid in the cache, the RSU component delivers the `DATA` packets directly from the local cache, otherwise a vital copy is prefetched from the data producer. It can be seen that in-network caching capabilities at nodes deployed closer to consumer have the ability to increase the availability of data closer to the consumer. As a result, it also reduces delivery times of `DATA` packets. However, the results show that many `INTEREST`s are not answered at all by the infrastructure network. This is caused by the limited number of deployed infrastructure nodes and thus results in a low communication coverage. As this is a infrastructure related problem, it is also not solved by a deployment of the PeRCeIVE approach.

As proposed by the second scenario (S2), a promising solution is the introduction of vehicles as data mules to overcome the limitations of infrastructure uncovered areas. In this case, the vehicle caches have been enabled in S2 to ferry information into uncovered areas, and therefore, increase the availability of information in the vicinity. The first vehicle which passed the RSU and receives a `DATA` packet stores the item in its local cache and ferries it into an uncovered area. As part of the functional test, the second vehicle (Figure 5.28b, Car B) received the `DATA` packet directly from the cache of the ferrying, while it is driving through an infrastructure uncovered area. The test scenario has shown that the availability of information can be increased by introducing cars as mobile cache nodes (data mules).

Summarized, both use cases were successfully completed in a real world test deployment. The prototype implementation has been tested by monitoring the number of requests and responses at producing and consuming nodes. While it was not possible to extract quantitative results from the small testbed setup (due to costs and scaling challenges), the results have shown the potential of performance improvements on different parts of the network. By caching information at the edge of the network, the number of requests at the producer has decreased. Furthermore, the number of `DATA` packets answered directly with the first hop has increased. Additionally, the availability of information in uncovered areas has increased using vehicles as data ferries (data mule) and ICN for V2V communication.

5.9 Summary

In mobile scenarios, the native support of in-network caching in ICNs have shown performance improvements by storing data closer to consumers. However, the reverse path forwarding as well as the variety of automotive data have shown limitation of *reactive* caching strategies.

Based on the analysis of automotive applications, data traffic classes have been identified and mapped to caching criteria. On the one hand, reactive caching approaches are reasonable as long as data is popular and valid over a long period. On the other hand, the results of the analysis have identified data classes which are beneficial to be cached proactively in the network, namely *popular transient*, as well as *personalized transient* data both independent of its size. This analysis contributes to the research question *Q*1.1 (cf. Section 1.4.1) of this thesis.

According to the data classes, three novel proactive caching strategies have been introduced and evaluated against the state-of-the-art in reactive content placement. While each of the presented strategies have shown their strengths addressing one of these traffic classes, the presented strategies, `PeRCeIVE`, `ADePt`, and the predictive data prefetching approach complement each other within an active content placement framework for automotive applications. Different options of dissemination strategies, to actually load data into caches, have been identified and discussed. This includes mechanisms triggered by the consumer itself (e.g., as in PeRCeIVE), by the infrastructure nodes using the popularity of data items (e.g., as in ADePt), or using historical information to load items into cache components (e.g., as in the predictive prefetching approach). The presentation of the evaluation results have shown that the novel proactive caching strategies improve the performance of the data delivery by increasing the availability of data closer to consumers and thus reduces the delivery times tremendously. These results contribute to the main research question *Q*2 and its siblings (cf. Section 1.4.1) of this thesis.

While the results of the strategies have been performed in a simulation, a real world vehicular prototype has been created in order to investigate the principles of the Named Data Networking architecture (incl. state-of-the-art reactive caching) in the real world. Initially created to evaluate the presented proactive caching strategies on a small scale, functional tests have shown further potential of proactive caching by introducing vehicles as data mules and thus bringing data actively into areas uncovered by any infrastructure nodes. These results form the basis for caching strategies presented in the next chapter.

6 Virtual Cache Areas for Information-Centric Connected Vehicles

> You can have data without information,
> but you can not have information
> without data.
>
> Daniel Keys Moran

While the results presented in Section 5 have shown network performance improvements such as the reduction of delivery times by placing Information Objects proactively at infrastructure nodes, the mechanisms are not ideal when looking into sparse network deployments. For example, the functional tests of the vehicular NDN prototype (cf. Section 5.8) as well as the sparse network deployment of the ECo-AT have shown that the intermittent connectivity between vehicles and the infrastructure network challenges in-time delivery of data items for connected vehicle applications.

In this chapter, the concept of *virtual cache areas*, which extends the scope of physical cache nodes to a virtual area, is introduced based on the Named Data Networking architecture. In the first part of the chapter, the problem statement is presented. Next, an analysis is provided in which vehicles are considered to carry Information Objects into areas uncovered by infrastructure nodes (cf. research question Q1.3 in Section 1.4.1) using principles from stochastic geometry. Based on the results of the analysis, the concept of virtual areas is introduced, followed by a discussion of the effects of loading Information Objects into moving cache nodes (cf. research question Q2.3 in Section 1.4.1). Finally, the results of the virtual cache areas are presented using simulations[10].

6.1 Problem Statement: Intermittent Infrastructure Connectivity

Future vehicular systems, e.g., fully automated driving systems, will require information retrieval in time. First versions of such systems are presented as part of the introduced use cases (cf. Section 1.2). However, retrieving always the latest and vital version of information in mobile networks describes a non trivial task. Due to the sparse network deployments in vehicular networks, for example the ECo-AT deployment in which only 2.4km out of a 10km road corridor is covered (cf. Section 5.6.1), intermittent connectivity challenges in-time retrieval of Information Objects for connected vehicle applications. Figure 6.1 illustrates the challenge of sparse network deployments in connected vehicle environments. Vehicles within the "uncovered area" are not able to receive any information from network services.

While the previously presented infrastructure assisted proactive caching strategies have shown performance improvements, the efficient delivery of Information Objects highly depends on the RSU deployment. For example, an infrastructure assisted placement strategy such as PeRCeIVE is helpful, if the "next" RSU is just a few meters away. One promising option to overcome this challenge is the introduction of mobile nodes carrying Information Objects from one location to another.

[10] The work in this chapter is published in the conference Proceedings of the 2018 IEEE Vehicular Networking Conference [277]. Parts of it are extracted from these sources.

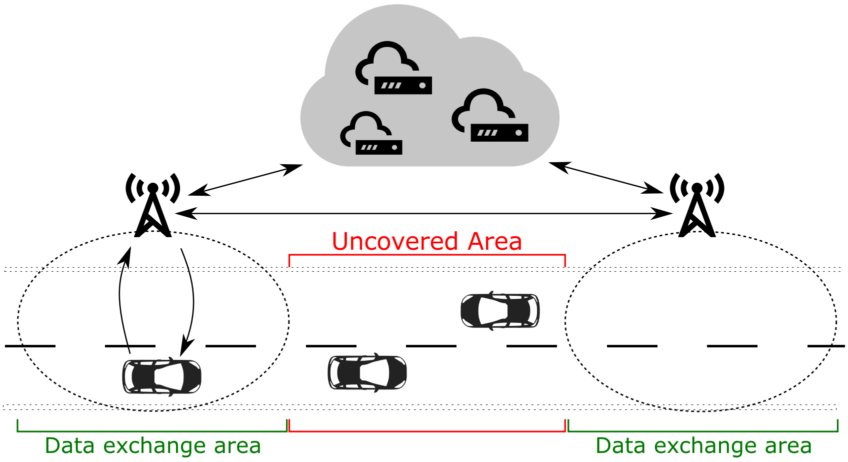


Figure 6.1: Example of a sparse infrastructure network deployment. Vehicles within the communication area of a infrastructure node are able to exchange information, while vehicles present in the uncovered area are not able to exchange any information. Depending on the deployment structure and the geo-location, the amount and distance of uncovered communication areas varies.

By introducing V2V communication, such mobile nodes are able to provide access to cached data. For example in vehicular DTNs, research activities have shown the benefits of the SCF paradigm in the context of vehicular networks (e.g., [56, 57]). Regarding vehicular ICN, the functional tests of the prototype introduced in Section 5.8 have shown that vehicles running an ICN protocol stack can use their in-network caching capabilities to carry data items in areas which are uncovered by any infrastructure nodes, similar to SCF in DTNs.

By deploying an ICN-enabled protocol stack on vehicular nodes, each node becomes a potential mobile cache, able to provide or carry data from one location to another. Each time a node meets other nodes during the journey, they are able to exchange information objects from their caches. The loosely coupled communication model of ICNs simplifies the access to data. In this thesis, this scenario is introduced as a potential virtual cache area in which data can be placed/carried by mobile nodes proactively. Figure 6.2 illustrates the creation of several virtual cache areas.

However, in-vehicle resources such as storage capabilities as well as computational power are limited and may include restricted access to these resources. While the infrastructure may assist in providing data items which have to be loaded onto the vehicles' caches, selecting and carrying the right data at the right node in time is still a challenge.

According to the research question *Q1.3* of Section 1.4.1, the following sections provide an analysis of the benefits of introducing ICN-enabled vehicles to carry Information Objects into uncovered network areas. Based on the principles of stochastic geometry, models are created to assess the potential of in-network caching capabilities in order to answer three related questions:

- What is the overall available cache capacity of an instance of a virtual cache area?
- How much cache capacity is accessible for a certain vehicle in a vCache instance?
- What is the probability that vehicle x receives a desired information item i from a node within a cache area?

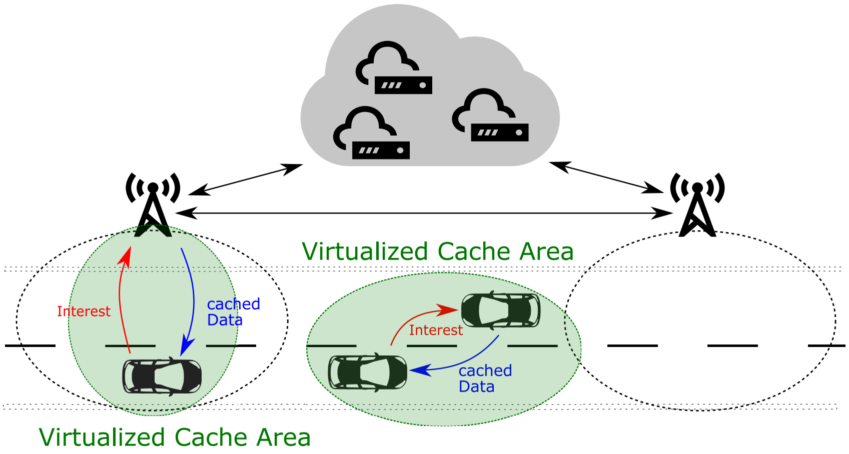


Figure 6.2: Example scenarios of the creation of virtual vehicular cache areas. By introducing ICN as the underlying network, the in-network caching capabilities combined with the direct access to Information Objects using names enable vehicles to load and ferry items into areas uncovered by any infrastructure node. As a result, the availability of items is increased facilitating information retrieval in time.

6.2 Introduction to Stochastic Geometry in Vehicular Networks

From an abstract point of view, mobile communication systems are represented by a collection of nodes in a given geographic area. Such nodes may be mobile consumers or producers, relaying nodes, or fixed APs such as cellular base stations or RSUs, each of it able to transmit and receive data packets. A decisive factor in mobile communication systems is the geometry of the locations of the nodes, and thus, the distance between the transmitting and receiving nodes. It affects the probability of a successful transmission of data packets [278].

In computer science, the modeling of such a system can be done using *stochastic geometry*. Similar to *queuing theory*, in which all potential traffic patterns influencing the performance of a system are used, e.g., to estimate response times or packet losses, the properties of a system are averaged over all geometrical patterns in *stochastic geometry*.

In the past decade, the interest in modeling wireless networks using principles from stochastic geometry increased. A detailed overview of the principles are provided by Baccelli et al. [278] or Haenggi [279], while a detailed tutorial about the modeling of a wireless system is provided by Coeurjolly et al. [280].

If the entire mobile communication system is considered as a series of *snapshots*, each of it can be analyzed in a probabilistic way. Each location of the network components within a snapshot can be seen as a instance ϕ of a spatial Point Process (PP) Φ. It models the random distribution of points within the whole Euclidean space $\mathbb{R}^d$, where d represents the dimensional space (e.g., $d = 1,2,3$). The probabilistic analysis allows modeling of system properties as functions such as connectivity, packet losses, or capacity. Each of these functions can be applied to each of the points within the process when a snapshot is taken of the network.

Modeling mobile communication systems as point processes allows statements about the entire class of such systems, instead of one certain configuration. The model used in this thesis is based on the model introduced by Tong et al. [281]. The positions of vehicles on a specific road segment P with distance d in meters form a *homogeneous* PP Φ with intensity λ. Figure 6.3 illustrates an example of such a process in the context of vehicular systems.

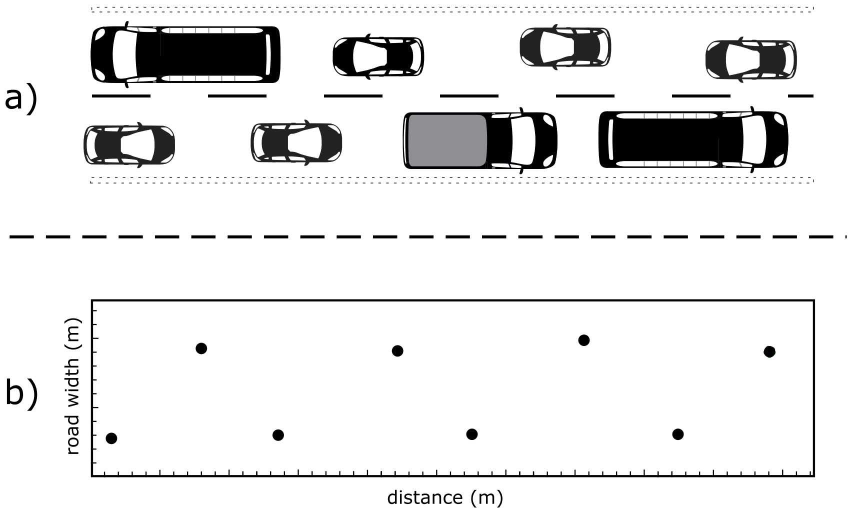


Figure 6.3: Example of a snapshot of a mobile communication system creating a point process from a stochastic geometry perspective. Figure a) illustrates a road network for top view perspective. Figure b) illustrates the positions of the vehicles on the road in the two-dimensional euclidean space as part of a scatter plot (the axes are scaled equally).

If the distribution of the nodes in Φ follows a Poisson distribution, the process is called Poisson Point Process (PPP). If nodes in the point process have characteristics which are independent of each other, it allows to model such characteristics as an independent mark/function R_x associated with each node x, forming a marked point process (cf. [278, 281]).

In order to resolve the expected values of a marked point process, the *theorem of Campbell* can be used to sum over all points (cf. [278, 279]):

$$\mathbb{E}[\sum_{(x,R)\in\hat{\Phi}} f(x,R)] = \lambda \cdot \mathbb{E}[\int f(x,R)\,dx] \tag{8}$$

A more specific example of using principles from the stochastic geometry in vehicular networks is provided by Tong et al. [281]. The authors use geometric models to capture the behavior of network protocols for future V2V safety applications.

6.3 Related Work in Collaborative Caching in Vehicular ICNs

The usage of vehicles to carry content through sparse networks has been investigated in the past. Especially, the SCF mechanism of DTNs has attracted the attention of researchers by providing a solution space regarding intermittent network connectivity. While publications such as Gao et al. [282] or Hyytiae et al. [283] present collaborative caching approaches in DTNs to provide access to information in sparse networks, the benefits of introducing ICNs in vehicular systems are not considered.

By introducing ICNs and their intrinsic in-network caching capabilities, Anastasiades et al. [284] present an agent-based SCF content retrieval approach in vehicular networks. Mobile nodes are used to carry Information Objects on behalf of a consumer in the system. Duarte et al. [239] investigated the effects of low vehicle density in vehicular named data networks and introduces two agent-based SCF approaches, one supported by the infrastructure and another directly considering vehicle-to-vehicle communication. However, the topic of proactive content placement is not considered by the presented work.

While first publications have shown the benefit of introducing vehicles as content carriers in sparse networks, the benefits of using ICN enabled vehicles regarding *proactive* caching, as well as a formal model of the cache capabilities have not been elaborated yet.

6.4 Analysis of Intermittent Infrastructure Connectivity in Vehicular ICNs

In order to answer the presented research questions raised in Section 6.1, formal models are introduced using principles from *stochastic geometry* (cf. Section 6.2). In order to verify the applicability of the models, simulations have been performed based on the ECo-AT simulation environment introduced in Section 5.6.

In order to compare the models against the performance of the different network configurations, the performance model presented in Section 5.7 is used as a basis. First, the *cache utilization* metric C_U is considered. It describes the number of allocated memory at each cache node by keeping n items out of N in the local storage. Besides the metric of C_U, an additional metric is introduced:

Information Object Availability Ratio (A_R): The overall Information Object availability ratio A_R describes a specialization of the *Resolved `INTEREST` Ratio* presented in Section 5.7. It is an indicator of the availability of a certain Information Object in the network and is defined as:

$$A_R(i) = \frac{\sum \texttt{DATA}\ (i)}{\sum \texttt{INTEREST}\ (i)} \tag{9}$$

where $A_R(i)$ is the number of successful received `DATA` packets for requesting a certain object i from the network, divided by all `INTEREST` packets sent by the consumer for object i.

6.4.1 Overall Cache Utilization

A model of the overall cache utilization C_U (cf. first question in Section 6.1) is created as stationary point process. It is used to indicate the overall potential cache utilization of a virtual cache area.

As introduced in Section 6.2, a vehicular communication system is modeled as point process Φ in which each road network is separated into segments providing two heading directions. In this thesis, a road segment of 1500 meters on the highway A4 in Vienna close to the junction "Schwechat" is considered, based on the road network of the ECo-AT (cf. Section 5.6.1).

A realization of the vehicle environment is given as a part of a one-dimensional stationary point process $\phi = \{x_1, x_2, ...\}$, while x represents the different vehicles in the segment. The dimension of the area of interest is represented by a road segment P given as $[d_S, d_E] \in \mathbb{R}$, while d_S defines the start and d_E the end of the area. The following assumptions are made for each vehicle x, the road segments P and ϕ:

1. U_x describes a random variable indicating the currently *used* cache capacity
2. M_x describes a random variable indicating the *maximum* cache capacity
3. U_x and M_x are independently and identically distributed (i.i.d)
4. P describes the road segment of interest
5. λ describes the average intensity of vehicles in P, where $0 < \lambda < \infty$
6. ϕ is stationary with intensity λ

When modeling the overall cache utilization, a function which indicates the resource ratio at each vehicle is required and defined as:

$$R_x \hat{=} \frac{U_x}{M_x} \tag{10}$$

Each vehicle in ϕ is marked with r_x. As a result, the process ϕ is given as a stationary marked process $\hat{\phi} = \{(x_1, R_1), (x_2, R_2), ...\}$, while (x_n, R_n) is given:

$$f(x, R) = R \cdot \mathbb{1}_{[d_S, d_E]}(x) \tag{11}$$

for all vehicles x within the road segment boundaries of P:

$$\mathbb{1}_{[d_S, d_E]}(x) = \begin{cases} 1, & x \in [d_S, d_E] \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

To determine the overall cache utilization, C_U, the expected value can be expressed via the theorem of Campbell (cf. Section 6.2) for stationary i.i.d marked processes as follows (cf. [279]):

$$C_U = \sum_{x \in \hat{\phi}} f(x, R) \tag{13}$$

for which the expected value is given by:

$$\begin{aligned} E[C_U] &= E\left[\sum_{(x,R)\in\hat{\phi}} f(x, R)\right] \\ &= \lambda \cdot E\left[\int_{\mathbb{R}} f\left(x, \frac{U}{M}\right) dx\right] \\ &= \lambda \cdot E\left[\int_{\mathbb{R}} \frac{U}{M} \cdot \mathbb{1}_{[d_S, d_E]}(x)\, dx\right] \\ &= \lambda \cdot E\left[\int_{d_S}^{d_E} \frac{U}{M} dx\right] \\ &= \lambda \cdot E\left[(d_E - d_S) \cdot \frac{U}{M}\right] \\ &= \lambda \cdot (d_E - d_S) \cdot E[U] \cdot E\left[\frac{1}{M}\right] \end{aligned} \tag{14}$$

Table 6.1: Simulation parameters used to evaluate the cache utilization and data availability models with respect to virtual cache areas.

Parameter	Values
Comm. technology	IEEE 802.11p OCB
no. objects	5000 unique objects
cache size	100 $^{\text{object}}/_{\text{node}}$
I	1500 meters
λ	1.500–7.000 $^{\text{vehicles}}/_{\text{hour}}$
Request rate	2–60 seconds
Sim Duration	6-24 hours

As illustrated in equation 14, the overall capacity is linear proportional to the traffic density λ as well as of the dimension of the road segment I. Due to the fact that the number of vehicles on the road varies depending on the time of day, the overall cache capacity of a certain area is expected to vary with traffic volume.

Simulations of the Overall Cache Utilization

In order to validate the model of the cache utilization C_U, a simulation environment is created to get values for the random variables U and M (cf. Section 6.4.1). The simulation environment is based on a road segment of 1500 meters on the highway A4 in Vienna as part of the ECo-AT deployment (cf. Section 5.6). This road segment has been chosen for simulation due to the fact that RSUs are deployed at the beginning and the end of the segment. This deployment structure is of importance for the subsequent sections. The intensity λ of C_U is based on the real world traffic performance data traces. varying on an hourly basis. In total, 4800 runs have been made to analyze the overall cache utilization C_U of the corridor. Additional simulation parameters are given in Table 6.1. By deploying ndnSIM reference applications, all network nodes including the vehicles are able to exchange NDN packets. Furthermore, tracing applications of the cache components as well as packet flow tracers have been implemented and installed on each node in the simulation environment.

Two scenarios are considered in the simulation. The first scenario takes only V2I communication between vehicles and infrastructure nodes into account (no direct communication between vehicles). In this scenario, caching is only enabled at the RSU nodes and disabled at all the other nodes in the environment. The second simulation scenario allows for V2X communication including V2I and V2V packet exchange. In the second scenario, caching is enabled at all nodes in the network except the producer of data. Additionally, the following assumptions are made for the evaluation of the overall cache utilization:

- the distribution of Information Objects in the communication system follows a Zipf-like distribution with exponent 0.8 and according to [285]. Each item is of same size and allocates the same number of memory.
- the velocity of the vehicles varies between 80–130 $^{\text{km}}/_{\text{hour}}$, according to the speed limit of the road segment.
- multi-hop communication is not considered in any simulation runs.

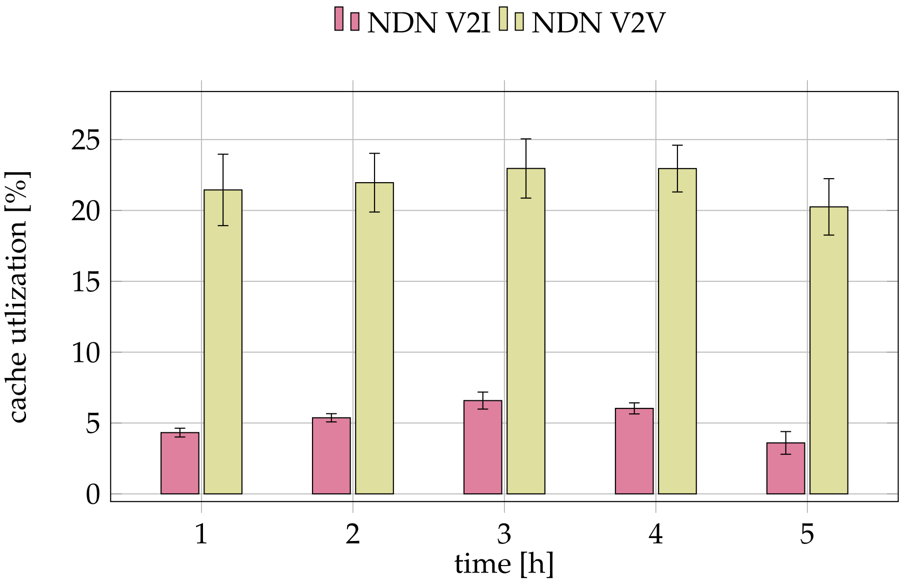


Figure 6.4: Sample results of the overall cache utilization over time. By enabling caching functionality at all nodes including vehicles, the overall cache utilization is higher compared to caching at infrastructure components only.

Results of the Cache Utilization

The analysis of the simulation results of C_U is shown in Figure 6.4 and the corresponding availability ratio in Figure 6.5. The traffic density within the simulation is averaged, and used as input for λ in Equation 14. Averaged values for the random variable indicating the currently *used* cache capacity U_x as well as for the random variable indicating the *maximum* cache capacity M_x are extracted from the mobile nodes. As part of the first simulation scenario in which communication and data caching is only allowed by infrastructure components, the averaged values of the cache utilization C_U is rather low (lowest: 1.2%, highest: 3.5%). Dependent on the traffic density, λ, the number of items stored in RSU caches increases during peak hours. One important aspect to be mentioned is the vitality period of the Information Objects with the RSU caches. If the vitality value of a DATA packet expires, the packet is not delivered to the requesting node. In this case, a natural balancing of the cache utilization can be observed. However, not considering vehicles as cache nodes shows the under-utilization of the cache potential.

By introducing vehicles as cache nodes, the utilization increases (lowest: 20.2%, highest: 22.9%). However, it can be seen, that the overall available cache capacity is not reached. Again, this is explained by the natural balancing of vital data items in the caches of the nodes.

Summarized, using the averaged values for the traffic density as well as the cache information at each node from the simulation runs, the model for C_U as stated in Equation 14 is useful to calculate the cache utilization. The simulation runs have shown that C_U is highly dependent on the varying traffic density as well as the vitality values of packets affecting cache utilization balancing. Finally, there is still memory left in the observed cache area, to be used to increase the availability of Information Objects for consumers.

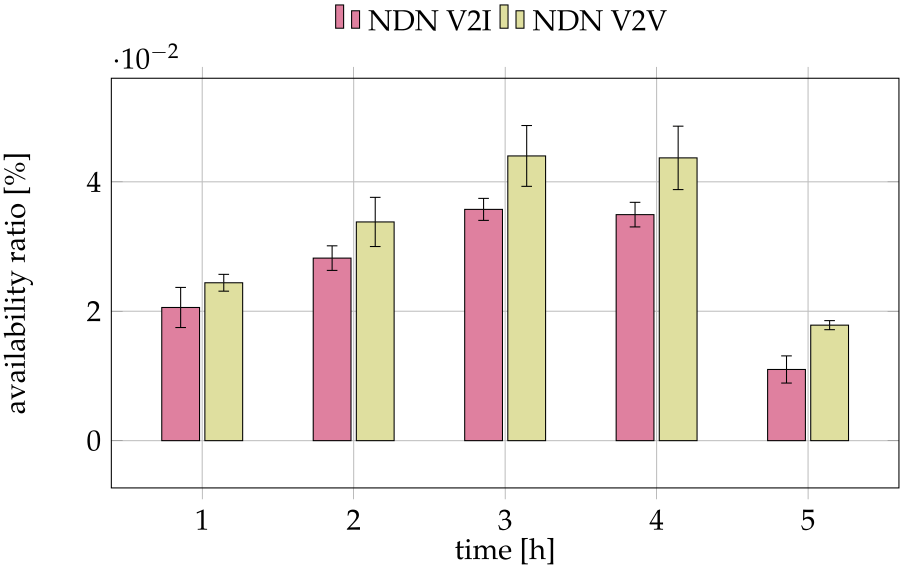


Figure 6.5: Sample results of the overall availability ratio over the simulation time. By taking moving vehicles into account for delivering cached objects, the overall availability of those items is higher compared to infrastructure caching only.

6.4.2 Potential Available Cache for a Single Vehicle

As part of the problem statement (cf. Section 6.1), the second research question addresses the potential available cache capacity for a single vehicle. It presents an expectation value whether a certain vehicle is able to receive a specific Information Object while traveling through the road segment. In order to determine the availability of a certain item, the available cache partner nodes for a specific vehicular node in the system defines an important indicator.

For this purpose, a model as part of a *homogeneous* PPP is considered (cf. [281]). The model is similar to the one presented for calculating the spatial averages of the cache utilization C_U. However, the distance of the area to observe is limited by the communication range of vehicle x given as $[(d_x - d_R), (d_x + d_R)] \in \mathbb{R}$, where d_X defines the position of vehicle x and d_R the range. Since the location of the vehicular nodes are placed according to conditions of a PPP, $\mathbb{1}_{[d_S, d_E]}$ of equation 14 is modified to the unified limited communication range value for each node.

The results show that the potential available cache for a single vehicle is linearly dependent on the intensity λ (here the traffic density) as well as the communication range d_R. While the value of the communication range d_R varies less than the traffic density λ, (see next subsection "Results of the Cache Availability for a Single Vehicle"), the latter one is the dominating factor.

Simulations of the Available Cache for a Single Vehicle

Based on the same simulation setup, 200 runs for every hour of the day were simulated (in total 4800). In addition to the simulation parameters given in Table 6.1, the communication range for each vehicle is set to 250 meters, according to [286]. Based on the simulation results, the number of contacts is calculated to analyze the existence of at least one cache partner.

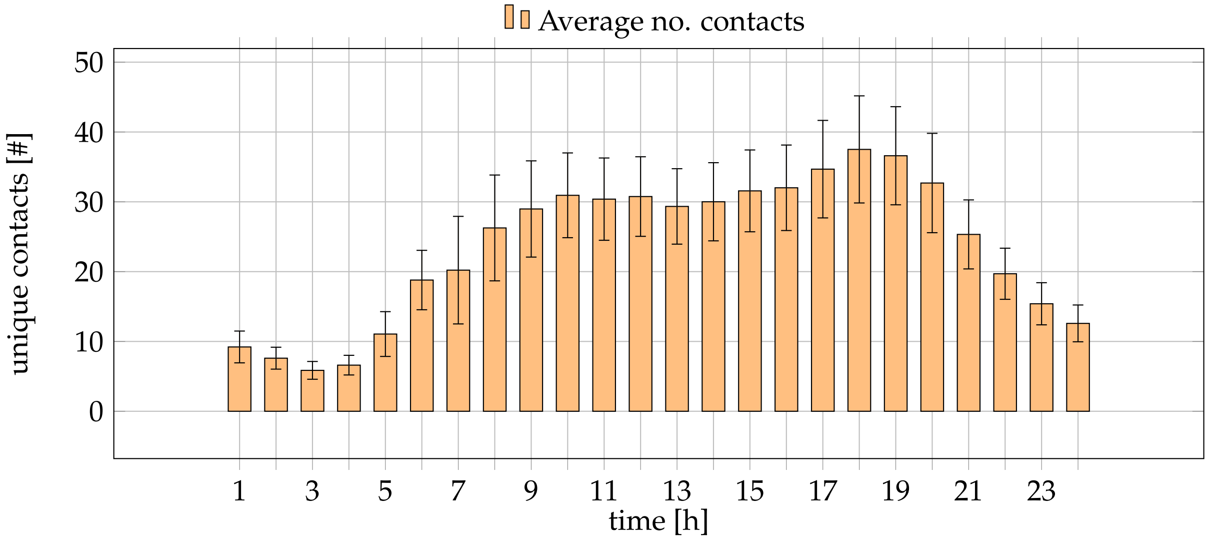


Figure 6.6: The average number of contacts for each vehicle driving through the simulation corridor at the hour of the day, based on the real world traffic performance data of a working day in June 2017.

Results of the Cache Availability for a Single Vehicle

Figure 6.6 illustrates the average number of contacts. While the number of nodes which have no contact at all varies between 3% (6pm) and 9% (3am), a large number have at least one contact to another vehicle, while driving through the simulation corridor. The lowest average contact value of 5 contacts is reached during off-peak hours (3am), while the maximum values are reached during the rush hours (9am and 6pm). It shows that the number of contact times is dependent on the traffic density λ, while the communication range has only a minor effect on the contact times. The variation in the number of contacts is due to the fact that the vehicles are passing the corridor with random velocity values ranging between 80–130 km/hour.

6.4.3 Data Retrieval Probability

The last research question to be addressed is described by the overall data retrieval probability (cf. Section 6.1). Since each vehicle is treated as a mobile cache, each individual mobile node is able to store a finite number of Information Objects. An investigation of the data retrieval probability is made based on the findings of the previous sections, including the number of node contacts in the road segment of interest.

As part of the retrieval probability model, the maximum number of Information Objects in the overall system is given by $N \in \mathbb{N}$. The conditional probability to get a specific Information Object i from a cache in the system is given as $P(i)$.

As IEEE 802.11p OCB describes a broadcast communication technology and is used in this thesis as media access technology, the number of nodes within the communication range of vehicle x represents the number of cache nodes M to retrieve a potential item. The distribution of the Information Objects within the caches of the vehicle follows a zipf-like distribution. This assumption is made, based on research of Breslau et al. [285], which states that the distribution of items within Web caches follows a Zipf-like distribution, sorted by their popularity. Therefore, the probability of accessing item i is given by $P(i)$ (cf. [285]):

$$P(i) = \frac{\Omega}{i^{\alpha}} \tag{15}$$

for which

$$\Omega = \left(\sum_{i=1}^{N} \frac{1}{i^{\alpha}} \right)^{-1} \tag{16}$$

A decisive factor is given by the α value, describing the class of the distribution function. According to Breslau et al. [285], the value ranges between $0 < \alpha < 1$.

Finally, the model of the data retrieval probability for a certain Information Object i out of N in M cache nodes is given by equation 17:

$$P_{Hit} = \sum_{i=1}^{N} P(i) \cdot \left[1 - [1 - P(i)]^{M} \right] \tag{17}$$

The probability P_{Hit} of retrieving an item i depends first on the probability $P(i)$ that item i is requested by the vehicle nodes M in the vicinity of x, followed by the probability to find i in M cache nodes. As stated previously, the distribution function plays an important role for the overall data retrieval probability P_{Hit} in the system.

Results of the Data Retrieval Probability

By using the average number of contacts for M (cf. Section 6.4.2) in Equation 17, P_{Hit} is calculated for an increasing number of available Information Objects N. The results show that the probability of receiving a specific Information Object in the system follows the Zipf distribution (cf. [285]). However, by including the contact times M, the probability of retrieving a specific Information Object increases with the number of contacts. Figure 6.7 illustrates a plot of the retrieval probability. It can be seen that the retrieval probability follows a logarithmic curve when the number of contacts increase. As a result, the probability of receiving the most popular items is more or less independent on the time of the day, while the probability of receiving less popular items is more likely during peak hours than in off-peak hours.

Summarized, the results of the analysis have shown the under-utilization of cache nodes for a given area based on the real world performance data of the ECo-AT deployment. By using principles from stochastic geometry, the provided models have proven useful to examine the potential of virtual cache areas towards improving the availability of data items in regions not covered by any infrastructure deployment. Each of the analysis related research question raised as part of the problem statement has been addressed in this section. The fact that the retrieval probability follows a logarithmic curve when the number of contacts increase is used in the next section to increase the availability of Information Objects in such virtual cache areas.

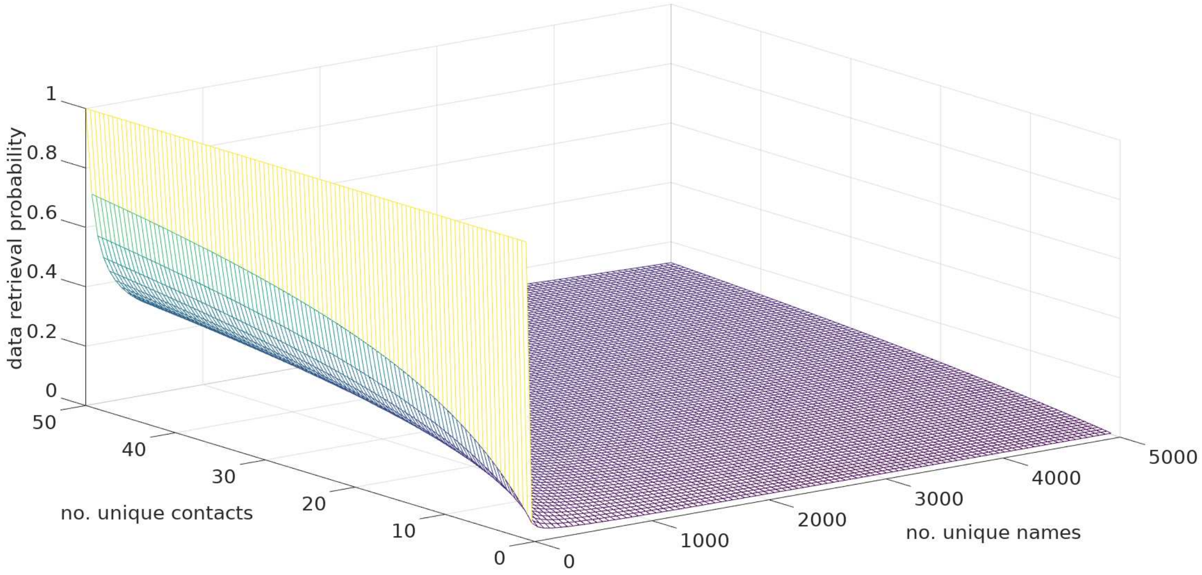


Figure 6.7: Illustration of the data retrieval probability. It follows a logarithmic curve when the number of contacts increases.

6.5 Caching-as-a-Service for Connected Vehicle

There is a lot of potential to increase the availability of an Information Object by loading it proactively into the CS of passing vehicles. Bringing the idea of virtual cache areas (presented in Section 6.1) and the results of the analysis of such cache areas together, the concept of vCaches is promising to overcome the challenges of intermittent connectivity in ICN-based vehicular networks.

In this work, *Caching-as-a-Service* (CaaS) describes a network service in an ICN to store specific Information Objects in vCache areas by using the caches of network nodes. Theoretically, *network nodes* include any types of network components such as fixed infrastructure components (e.g., cellular base station) as well as mobile components (e.g., vehicles) or a group and a mix of each of them. While the concept of CaaS is flexible regarding the types, number of components and the underlying network structure constituting such service, the concept of vCache areas is enhanced towards an infrastructure assisted CaaS in this section. Infrastructural nodes are able to load Information Objects into mobile vehicle caches to increase their availability. These objects are carried by mobile caches into areas towards consumers.

6.5.1 Requirements

In order to increase the availability of valuable Information Objects, CaaS requires information about objects worth to be loaded into the local caches of passing vehicles. Such information can be collected by the node itself running or governed by other nodes in the network.

In addition to the information of valuable objects, CaaS requires strategies to load Information Objects into passing vehicles efficiently. Due to the decoupling of data from physical locations in ICNs, protocol mechanisms to fulfill the loading strategies have to be aligned to the underlying ICNs implementation.

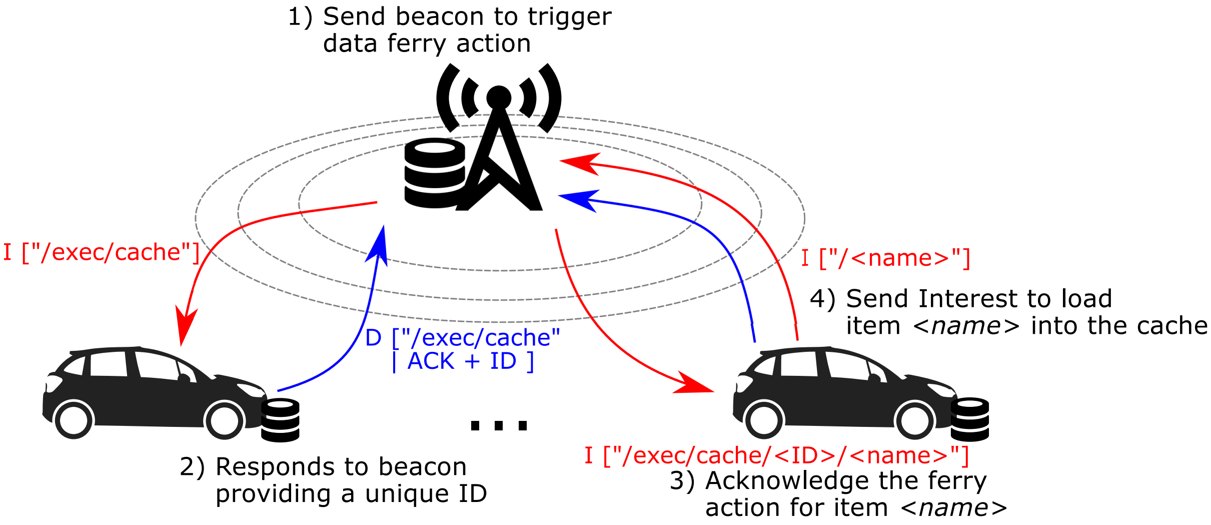

1) Send beacon to trigger data ferry action
I ["/exec/cache"]
D ["/exec/cache" | ACK + ID]
I ["/<name>"]
4) Send Interest to load item <*name*> into the cache
...
2) Responds to beacon providing a unique ID
I ["/exec/cache/<ID>/<name>"]
3) Acknowledge the ferry action for item <*name*>

Figure 6.8: Beacon-based loading mechanism: An infrastructure node sends out beacon message periodically. If a passing vehicle is within the communication range, both communication participants agree on loading a specific Information Object into the content store of the passing vehicle.

6.5.2 The Caching-as-a-Service Approach

In CaaS, service nodes (e.g., RSUs) require information about Information Objects to be loaded into caches. One option is monitoring of traffic flows in the network and learning about requested Information Objects. Here, the concept of ADePt (cf. Section 5.4) is used to monitor the traffic in an NDN. However, it is also possible to use other strategies to evaluate valuable Information Objects in the network (e.g., PeRCeIVE [18] or WAVE [15]). Whenever a vehicle sends out an `INTEREST` packet for a specific data item, the node forwards the request and keeps track of its name. In this case, an infrastructure node can identify the characteristics of data items (e.g., popularity or if an item is related to another one). Such items are of interest to be loaded into caches of vehicles to increase the availability, especially in areas which are not covered by any infrastructure nodes.

In case a service node is aware of Information Objects to be loaded into a vCache area, there are the following options of loading items into caches proactively in an NDN:

Beacon-based approach (hard): The *beacon-based* approach describes a mechanism which is triggered by the infrastructure component (e.g., a RSU) for keeping a copy of the object to be loaded into another cache. Figure 6.8 illustrates the beacon-based information exchange procedure. The infrastructure component sends out `INTEREST`s periodically requesting for mobile nodes (so-called *ferry* nodes). For this purpose, a special name prefix is added to the `INTEREST` packet: "`/exec/cache`" (e.g., [66]). Ferry nodes in the vicinity are strongly advised to commit to the request using the provided information. Furthermore, ferry nodes have to provide additional individual information (e.g., an ID to ensure to reach same mobile node) as part of a `DATA` packet. To avoid the injection of the same item at multiple nodes, a 2-phase commit mechanism is introduced. The infrastructure node sends out an acknowledging `INTEREST` using the individual information as part of an `INTEREST`, answered by the mobile node. Afterwards, the ferry node requests the item and load it into its local cache.

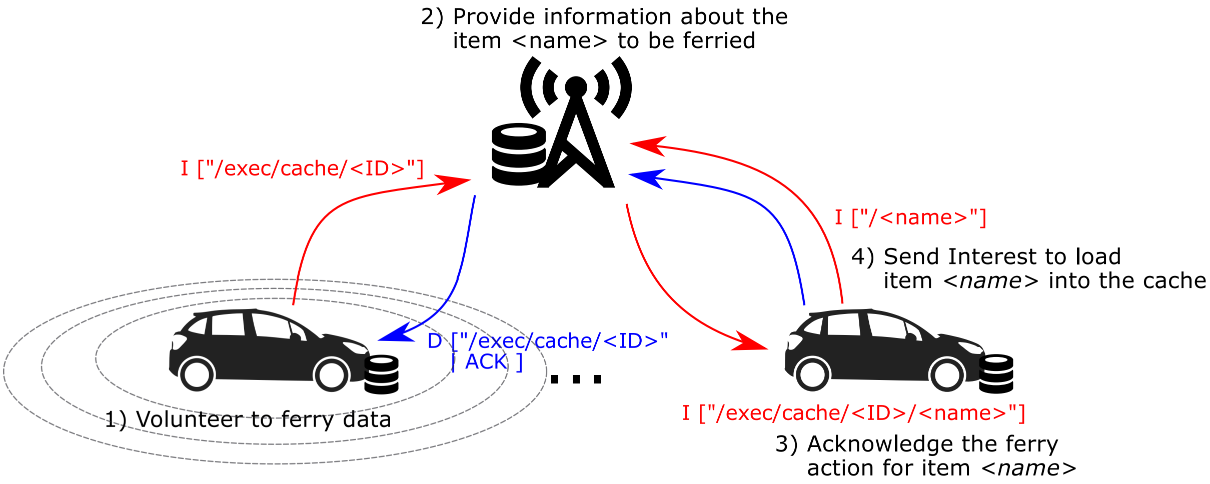

2) Provide information about the
item <name> to be ferried
I ["/exec/cache/<ID>"]
I ["/<name>"]
4) Send Interest to load
item <name> into the cache
D ["/exec/cache/<ID>"
[ACK]
...
1) Volunteer to ferry data
I ["/exec/cache/<ID>/<name>"]
3) Acknowledge the ferry
action for item <name>

Figure 6.9: Volunteer-based loading mechanism: A vehicle can volunteer to ferry specific Information Objects by sending out beacon message periodically. If an infrastructure node providing an object worth to be ferried into a vCache area is in the communication range, both communication participants agree on loading a specific Information Object into the content store of the passing vehicle.

Volunteer-based approach (soft): From the initiator perspective, the *volunteer-based* approach is different compared to the beacon-based mechanism. It is triggered by potential *ferry* nodes. Nodes can apply for loading Information Objects into their local cache from service nodes by sending out `INTEREST` packets. Figure 6.9 illustrates the volunteer-based mechanism initiated by a potential ferry node. Similarly, a special name prefix: "`/exec/cache`" is used. An infrastructure service node in the vicinity can answer to such volunteering request by providing information about the object to be loaded into the ferry node's cache. As a next step, both participants commit to the load request. Compared to the beacon-based approach, all the required information is transferred as part of the payload of a `DATA` packet, and therefore, not directly visible to other nodes.

Regarding the CaaS requirements (cf. Section 6.5.1) the presented concept is compliant with the loosely coupled model of ICNs. Both loading options incorporate the naming principles of NDN. From a data availability perspective, it makes no difference which node ferries the object into a virtual area, as long as there is at least one node.

6.5.3 Caching-as-a-Service Loading Strategies

In order to know *What* Information Object has to be loaded using *Which* mechanism, there is another important element of the CaaS concept: *strategies* which target the *How* to load objects efficiently into the caches of passing vehicles. Every time an object i is loaded into the cache of a node, the distribution function, and hence, the probability $P(i)$ changes. As a consequence, the data retrieval probability of all items in the overall system changes as well. In this case, the equation 17 of the data retrieval probability of Section 6.4.3 has to be modified accordingly:

$$P_{Hit} = \sum_{i=1}^{N} P\left(i\right) \cdot \left[1 - \left[1 - \widetilde{P(i)}\right]^{M}\right] \tag{18}$$

while $P\left(i\right)$ describes the initial probability, and $\widetilde{P(i)}$ describes the influenced probability of retrieving item i out of M cache nodes.

In this thesis, the concept of CaaS loading strategies is introduced. These strategies target to load object into mobile caches dependent of certain performance characteristics (e.g., popularity, priority, etc.), and therefore, influence $\widetilde{P(i)}$. In this work, the following strategies are introduced, investigated and analyzed using simulations:

S_1 **Treat all equally**: The first optimization option describes a caching strategy which loads items independent of their popularity: $\widetilde{P(i)} = 1/N$. As a expected result, the availability of less popular data will be increased, while popular data is still present in the area.

S_2 **Load k-popular items**: The second optimization option describes a fairness-based caching strategy, loading items which are below a certain popularity threshold, k. This decision strategy is illustrated by equation 19. As a expected result, the availability of less popular data is increased in the area, while initial popularity of the k-popular data items stays the same.

$$\widetilde{P(i)} = \begin{cases} 0, & i \geq \mathrm{k} \\ \frac{P(i)}{1-\sum_{j=k}^{N} P(j)}, & i < \mathrm{k} \end{cases} \tag{19}$$

6.5.4 Simulation Setup for Caching-as-a-Service Strategies Evaluation

The introduced loading strategies are implemented in the same ECo-AT simulation setup presented in Section 5.6. The implementation of the strategies is done using the built-in functionality of the NDN software stack. Applications deployed on RSUs within the simulation environment allow to load items into mobile caches using the beacon-based approach. Each vehicle application is able to respond to such beacon according to their request frequency (ranges 2–60 seconds). Every time an Information Object i has been loaded successfully to a mobile cache, the RSU application erases the object from its beacon list. Note, the novel caching strategies influence the distribution of objects within mobile caches, however, the mobility pattern is neither modified nor influenced at all.

6.5.5 Results of the Caching-as-a-Service Strategies

Compared to the standard caching scenarios (cf. Fig. 6.11), the results of the manipulation of the data retrieval probability have shown improvements in the overall measured data availability ratio A_R by loading data items proactively into caches of passing vehicles. The overall availability ratio has increased (e.g., roughly doubled the ratio at hour 1 compared to the reactive LRU caching of standard NDN) by loading less popular content into caches of vehicle passing areas out of coverage, replacing some of the most popular items in the caches.

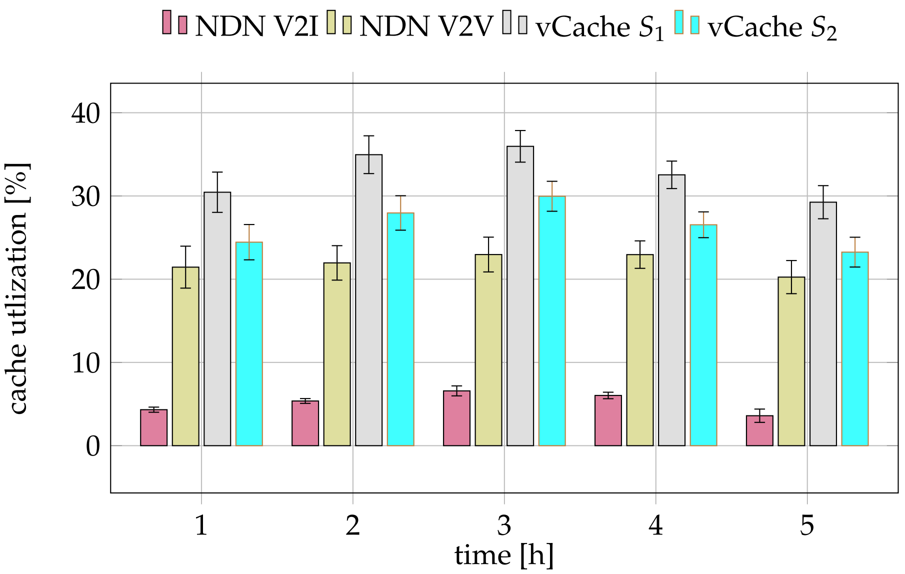


Figure 6.10: Overall cache utilization C_U of the vCache loading strategies compared to standard NDN behavior.

However, since any vehicle is able to participate in loading items into its cache (in average 69% of the vehicles respond to the beacon messages), the S_1 approach allocates more cache memory than the V2V reactive one (cf. Fig. 6.10): ranging between 29% and 36% in the sample hours. This is due to the fact that vehicles load data into the caches even if they are not requesting for their own local applications.

Regarding the second loading strategy (S_2), the average number of participating cache nodes (47%) is lower than the number of participants of S_1. This is due to the fact that the overall number of beacon events is lower, since only less popular data is advertised by the infrastructure nodes. The S_2 strategy also increases the availability ratio compared to the reactive LRU caching of standard NDN. However, A_R is below the numbers of the S_1 strategy. This is due to the fact, that the S_2 strategy loads semi-popular items in the cache nodes, while the probability of retrieving the most popular items is not changed. Regarding the values of the cache utilization, strategy S_2 allocates more memory than the standard NDN strategy, however, is below the utilized memory of S_1.

6.6 Discussion

The concept of CaaS in conjunction with vCaches contributes to the vision of a proactive caching framework (cf. Section 5.2). Important elements of the framework are described by the *strategies* of bringing information proactively toward consumers. CaaS in connected vehicle environments evolves the framework from placing objects at the edge of an infrastructure deployment, towards ferrying and delivering objects actively to mobile consumers.

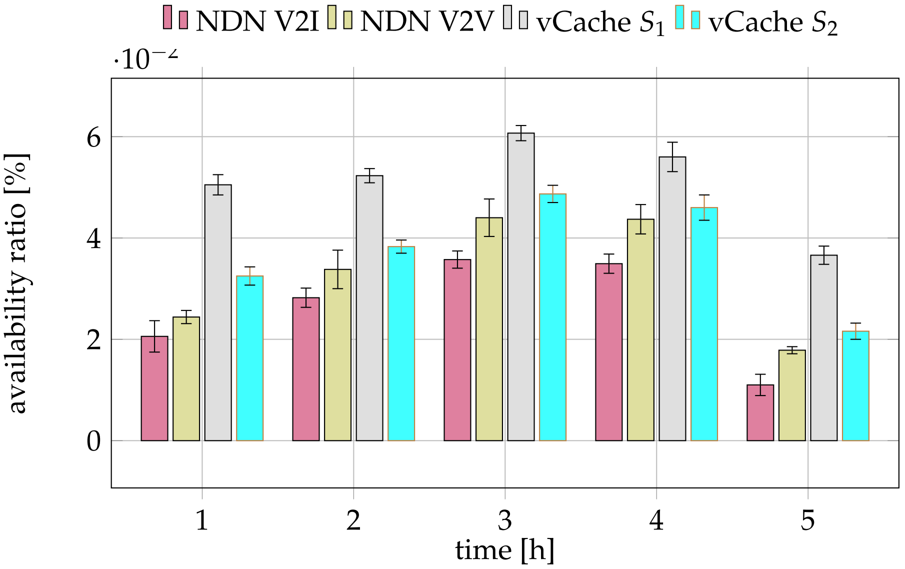


Figure 6.11: Overall availability ratio A_R of the presented vCache loading strategies compared to the standard mechanisms in NDN.

The measurements of the retrieval probability have shown that the probability follows a logarithmic curve when the number of contacts increases. As a result, the probability of receiving the most popular items is more or less independent on the day of time, while the probability of receiving less popular items is more likely during peak hours than in off-peak hours. Focusing on these results, the presented loading strategies have shown a performance improvement by using passing vehicles as data mules. Popular data is still present and is served by the vehicular nodes, while the availability of semi-popular data is increased by reducing the experienced data delivery time. However, the evaluation results have shown that there is still potential in the concept illustrated by the under-utilization of the overall cache capacity using the presented strategies.

In order to use this potential, CaaS and the presented placement strategies such as PeRCeIVE and ADePt have to complement each other within a common framework. For example, ADePt can be used to monitor traffic flows in order to identify Information Objects worth to be loaded or PeRCeIVE can be used to place Information Objects at edge components in order to load them into vCache areas. In summary, a common framework providing the presented strategies will further increase the availability of information in future connected vehicle environments.

6.7 Summary

The work in this chapter contributes vCaches – a concept of cache areas which are created each time network nodes meet one another, able to exchange information objects from their caches. It provides a solution for the *intermittent connectivity* problem which challenges the retrieval of information in vehicular networks. It can be seen that the CaaS concept complements the results of proactive placement strategies of Section 5 by introducing vehicles as data mules. The results of the chapter have shown potential as well as improvements in increasing the availability of data items in vCache areas, especially for sparse networks, thus answering research question Q1.3 of Section 1.4.1.

Based on the introduction of three formal models, an analysis has been made to determine influencing factors for virtual cache areas using principles from stochastic geometry. The results of the analysis have shown an under-utilization of the cache resources in ICN-based vehicular networks using standard NDN. One realization of virtual cache areas is introduced by the concept of Caching-as-a-Service in vehicular information-centric networks. Two novel caching strategies have been presented which have shown the applicability of CaaS in mobile networks. Based on a real world IEEE 802.11p collaborative ITS network deployment combined with real world traffic traces, the concept of CaaS in virtual cache areas have shown performance improvements by bringing specific data items *proactively* towards consumers in infrastructure un-covered areas.

7 Towards Computation-Centric Connected Vehicle Environments

> We are entering a new world in which data might be more important than software.
>
> Tim O'Reilly – CEO O'Reilly Media

When looking into the networking principles of ICNs, it can be seen that such networks offer the service to deliver information based on the content's name (e.g., CCN [12], NDN [13], etc.). Despite the conceptual fit to provide access to data in mobile networks, the vast majority of ICN architectures focus on the delivery of static content. As the network technologies move towards distributed, edge-computing environments, there is an urgent need to define a *"computation-centric"* approach to support decentralized and distributed computation. This is also aligned to network services in C-ITS systems, providing additional information for in-vehicle automotive applications. One promising candidate for computation-centric networks is Named Function Networking (cf. Section 3.1.4). NFN is an extension of the interest-based ICN approaches (e.g., CCN or NDN) to execute computation inside the network, and therefore, provide access to *dynamic* content on demand. Instead of reducing the delivery times of Information Objects by storing them proactively at the edge, NFN advocates to bring computations from the cloud to the edge, and therefore, closer to the consumer. From a use case perspective, parts of the electronic horizon (cf. Section 1.2.1) functionality may be distributed and executed in such networks along the road. This allows for new features (e.g., support of real-time computations along the road) which were not possible before. Originally designed for data centers, there are some challenges of introducing NFN in mobile networks characterized by a high degree of mobility.

By presenting an enhanced version of the *mobile node delivery problem*, this chapter introduces the limitations of NFN in IoT with respect to cached information in a vehicular computation-centric network architecture. Based on a discussion of the capabilities of NFN, novel strategies are proposed to deploy the architecture in connected vehicle environments. These strategies have been evaluated in simulation against the default mechanisms of NFN. Finally, an implementation is made to prove the strategies based on the real world prototype implementation of Section 5.8[11].

7.1 Problem Statement: Mobile Node Delivery Problem – Reissued

While automotive applications are not exclusively demanding static data, they are also requesting for data dynamically created and processed from other applications. Examples are requesting for point-of-interest information based on the current geo-location, or the closest patrol station. While the computation capabilities of in-vehicle components are limited (e.g., due to costs, size of the hardware, etc.), cloud computing and data centers to execute applications

[11] The work in this chapter is published in the Proceedings of the Edge Computing Workshop of the 2018 IEEE Consumer Communications & Networking Conference [287], and the 2018 ACM Conference on Information Centric Networking [288]. Parts of it are extracted from these sources.

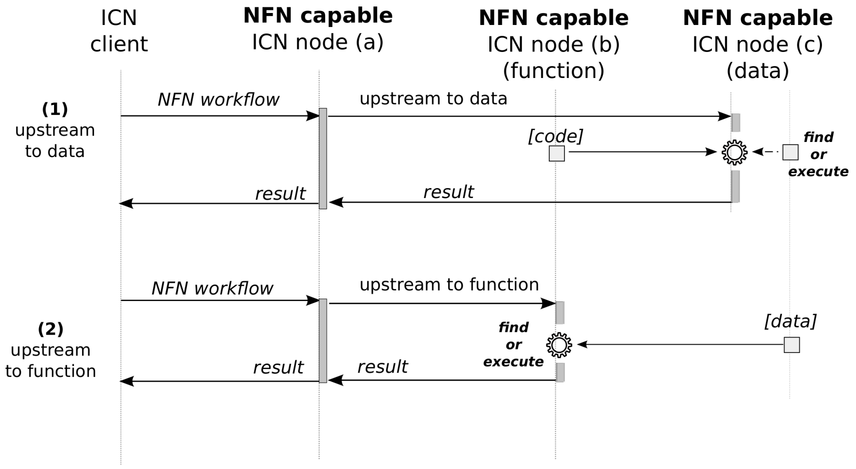


Figure 7.1: Find-or-Execute resolution strategy in NFN based on [65]: First, the network tries to find cached results by dividing the workflow definition into its components. If a cached result has been found in the network, it is delivered to the requestor. Otherwise, an execution node executes the computation and fetches related data from the network.

are well established in the domain of consumer electronics. However, participants in vehicular networks are characterized by a high degree of mobility, hindering timely data retrieval. In worst case scenarios, vehicles may have lost the connection to infrastructure components and therefore are not able to receive any data. The same problem occurs if computation intensive operations cannot be executed within the vehicle (e.g., resources are occupied by higher prior applications required for highly automated driving), and therefore, need to be offloaded from the vehicle.

Recent advances towards the introduction of computation resources at the edge has shown performance improvements in retrieving data (e.g., [45]), however, a vehicle application may not be able to receive a result of a computation intensive operation in time, during the journey. This challenge is also valid for a NFN that uses an ICN as the underlying transport.

Initially, NFN was designed for cloud computing and data centers to execute computation next to the data itself (e.g., for big data processing), instead of transferring large amount of data towards execution nodes. In NFN, the execution of computation is optimized to be efficient, if the computation time along with the transmission time is less than the deadline of the application requiring the result:

$$Deadline_{Application} \geq Time_{Computation} + Time_{Transmission} \tag{20}$$

To realize such functionality, NFN consists of two core elements on top of ICN principles: A *workflow definition* – used to express a computation, and a *resolution strategy* – used to resolve elements required to perform the computation such as the optimal execution node or required data. The underlying ICN communication model enables NFN to reuse already computed results by utilizing the network's caches.

In order to resolve a computation, NFN first tries to find a cached result and only executes a computation if a result is not found in the network. In NFN, this *resolution strategy* is called Find-or-Execute (FoX). Figure 7.1 illustrates the processing steps of the FoX strategy. The strategy considers two major scenarios: upstream the function towards an execution node

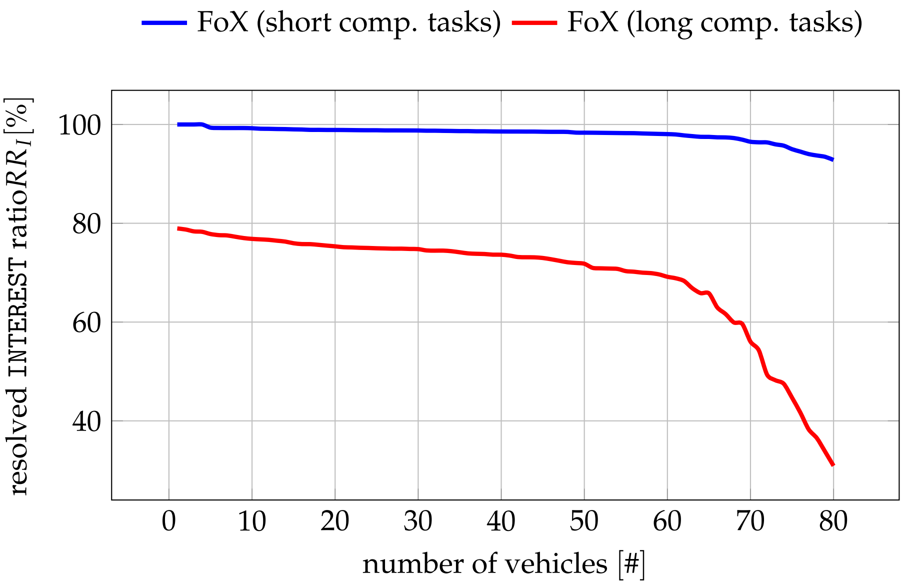


Figure 7.2: Results of the resolved INTEREST ratio for the NFN FoX resolution strategy in a mobile scenario. The setup includes 20 RSUs and 30 computations. While the connection time to the NFN infrastructure is sufficient to deliver short running computation results, the resolved INTEREST ratio for long running computations decreases tremendously with the increasing number of mobile nodes.

providing the data (Figure 7.1, case 1), or upstream the data towards an execution node providing the function (Figure 7.1, case 2).

Regarding data center deployments, the ratio between the $Time_{Computation}$ (time required to calculate a result) and the *Deadline* (time interval available to deliver data) is expected to be below a certain threshold, because resources are typically over-provisioned and deployed close to each other – resulting in shorter transmission times. However, regarding networks in the IoT, the execution strategy in NFN has to follow the characteristics of the IoT scenarios, for example in vehicular networks, the delivery of dynamic computation results on time (timeliness affect the *Deadline* tremendously) is challenged due to the mobility aspect. As part of a simulation setup including 20 RSUs, the performance of NFN FoX strategy applied to mobile scenarios is analyzed and illustrated in Figure 7.2. While the connection time to the NFN infrastructure is sufficient to deliver short running computation results (cf. Figure 7.2, blue line), the connection time is insufficient for long running computations using NFN FoX, resulting in a significant degradation of the resolved INTEREST ratio with the increasing number of mobile nodes.

This issue is related to the *mobile node delivery problem* introduced in Section 5.1. As presented in the previous chapter, the introduction of proactive data placement strategies helps to overcome the problem in plain ICNs. While it is expected that proactive caching may reduce the impact of the problem in an IoT deployed NFN, for example by placing computation results closer to consumers or in-network functions, it will not solve the problem related to the resolution procedures in NFN. Such strategies have to follow the characteristics of IoT scenarios, for example, by executing computations close to consumers or by considering node mobility to deliver dynamic computation results in time.

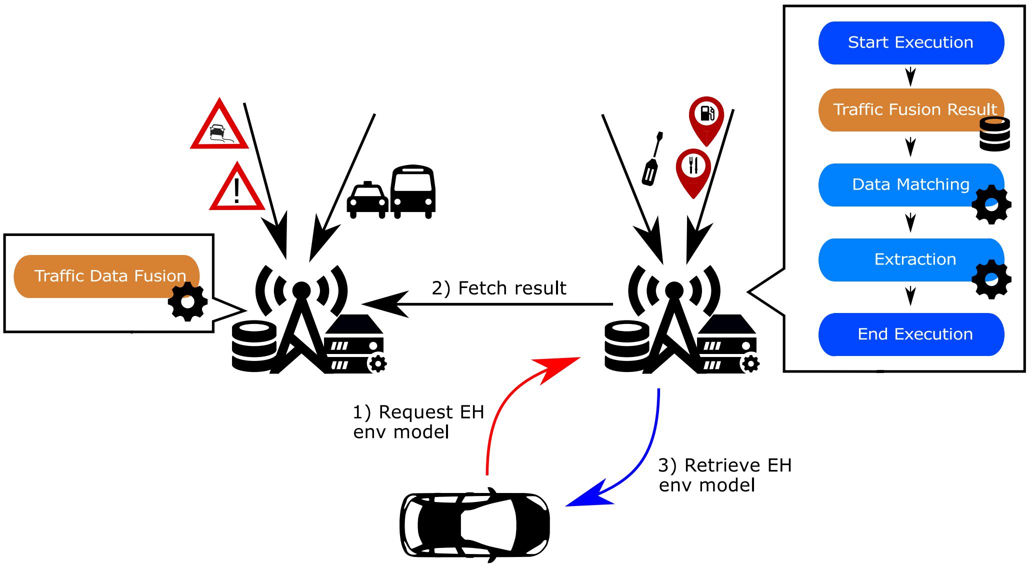


Figure 7.3: Electronic horizon functionality in a Named Function Network: The functionality is divided into independent partial computations and distributed across the execution nodes in a Named Function Network. An electronic horizon function is able to fetch results of sub-computations from other nodes in the system.

7.2 Resolution Strategies for Connected Computation-Centric Vehicles

The deployment of NFN in IoT environments enables the network to execute any function which has been previously performed in the cloud. As part of the automotive IoT, use cases such as the electronic horizon as well as the community-based sensing will benefit from a NFN. Instead of transferring, processing and downloading data from/to cloud backends, data can be stored, processed and delivered from NFN-enabled execution nodes in the vicinity.

An NFN-enabled version of the electronic horizon function is illustrated in Figure 7.3. The application is divided into multiple functions. In a NFN, these functions are flexible regarding their deployment, for example they are available at the edge of the network (e.g., at RSUs or cellular base stations). As a result, the execution of such distributed application at the edge has the potential to reduce latency in delivering dynamic computed results.

7.2.1 Limitations of the Named Function FoX strategy

Due to the *mobile node delivery problem*, the default NFN FoX strategy shows weaknesses in such networks (see Figure 7.2). Vehicles may have lost the connection to edge components executing the electronic horizon functions (cf. Figure 7.3), and hence, are not able to receive any data. As a result, a vehicle has to repeat the request, while starting the function execution at the next point again. The focus of the NFN strategies have to shift towards *time-sensitive* data delivery in mobile networks, instead of focusing on *efficient* resources utilization - as ensured by the FoX strategy. If the ratio between the computation time and the time interval to deliver data (cf. Equation 20) exceed the given deadline threshold, the default FoX strategy fails to deliver a result in time.

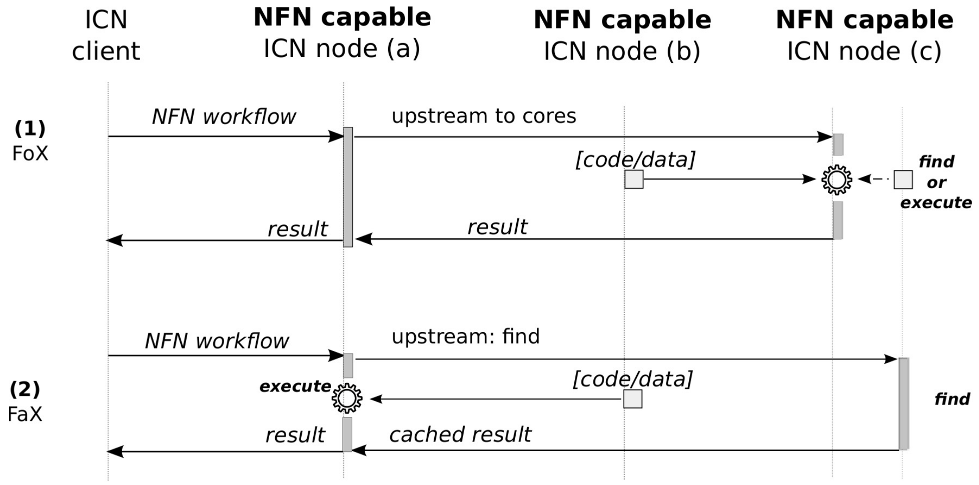


Figure 7.4: Illustration of the differences between the default FoX and the novel FaX resolution strategies. Instead of waiting and searching the network for cached computed results as stated by the FoX strategy, the computation will be started immediately by the FaX strategy.

7.2.2 Find-and-Execute (FaX) Strategy for Mobile NFN Scenarios

In order to reduce the data delivery time interval of the NFN FoX strategy, the Find-and-Execute (FaX) strategy is proposed. It describes an enhanced version of the default resolution strategy of NFN. Figure 7.4 illustrates the differences between the FoX and FaX strategy. Instead of waiting and searching the network for cached computed results, the computation will be started immediately at the first execution node (cd. Figure 7.4, FaX). If a cached result is delivered by the network before the computation has finished, the result will be forwarded to the requester and the execution will be stopped. Otherwise, the newly computed result will be delivered and the searched data will be dropped.

Regarding the problem statement presented in Section 7.1, the FaX strategy reduces the decision time of resolving a cached Information Object or starting a computation. As a consequence, the FaX strategy may start the same computation in the network multiple times. This strategy decreases efficiency, however, increases chances of data delivery by not waiting for any result of the network.

Another influencing factor of the delivery time is the computation time of named functions. While it is to be expected that FaX will perform well regarding small computations, it is expected to perform poorly, if the computation of intermediate results take time. Regarding the electronic horizon example, if partial computations (e.g., the fusion of data from multiple sources, cf. Figure 7.3) require long computation time, the strategy to execute the function immediately will result in unsatisfied requests too. This is due to the fact, that a vehicle may has lost the connection to the current edge node, or joined another network entry point, before the computation has finished. In this case, the result will not be delivered by the network, due to the reverse path forwarding of the underlying ICN.

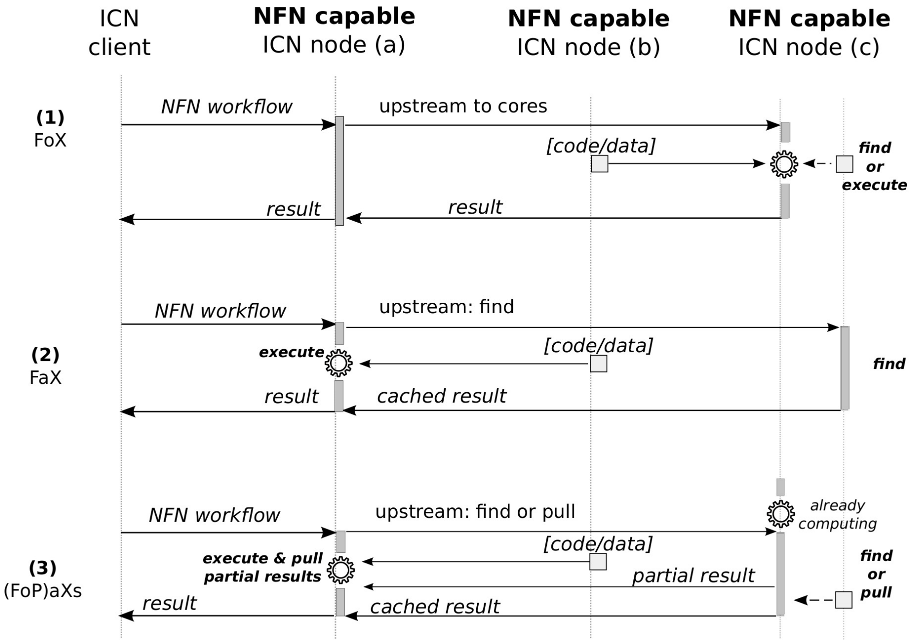


Figure 7.5: Illustration of the differences between the default FoX and the novel FaX and (FoP)aX resolution strategies. The (FoP)aX strategy is able to pull intermediate results from other execution nodes in the network using R2C messages.

7.2.3 Find-or-Pull-and-Execute ((FoP)aX) Strategy

In order to overcome the issue of delivering results of long running computations in mobile scenarios, an integration of Request-to-Computation (R2C) messages [289] is proposed as part of a Find-or-Pull-and-Execute ((FoP)aX) resolution strategy.

R2C messages are a protocol extension to NFN to steer a running computation in the network. It is based on `INTEREST` and `DATA` messages, and thus, is compatible with the ICN forwarding principles. Such a message contains name components to trigger remote functionality at an execution node. These name components are added to the original name of a NFN `INTEREST` packet requesting for a result. In order to reach the same node that executes the corresponding computation, the NFN layer is able to keep track of forwarding states within the PITs in an ICN. The introduction of R2C messages allows named functions to prevent timeouts for long running computation, fetch intermediate results of complex computations or to stop a running computation [289].

R2C messages allow the FoPaX strategy to pull intermediate results from other execution nodes in the network. Figure 7.5 illustrates the differences between the introduced resolution strategies in NFN. Similar to the FaX, FoPaX also start a computation immediately, if possible. In parallel, the execution node tries to fetch intermediate results from neighboring nodes to reduce the overall computation time using the R2C principle. In case a sub-computation has already been finished, the FoPaX strategy fetches the result from neighboring caches. Regarding the electronic horizon example, if partial computations are already calculated, but not available as cached objects within the NFN network (e.g., results of the data fusion or data mapping functions, cf. Figure 7.3), the execution node fetches the intermediate results from its neighboring nodes and continues the computation of the electronic horizon function.

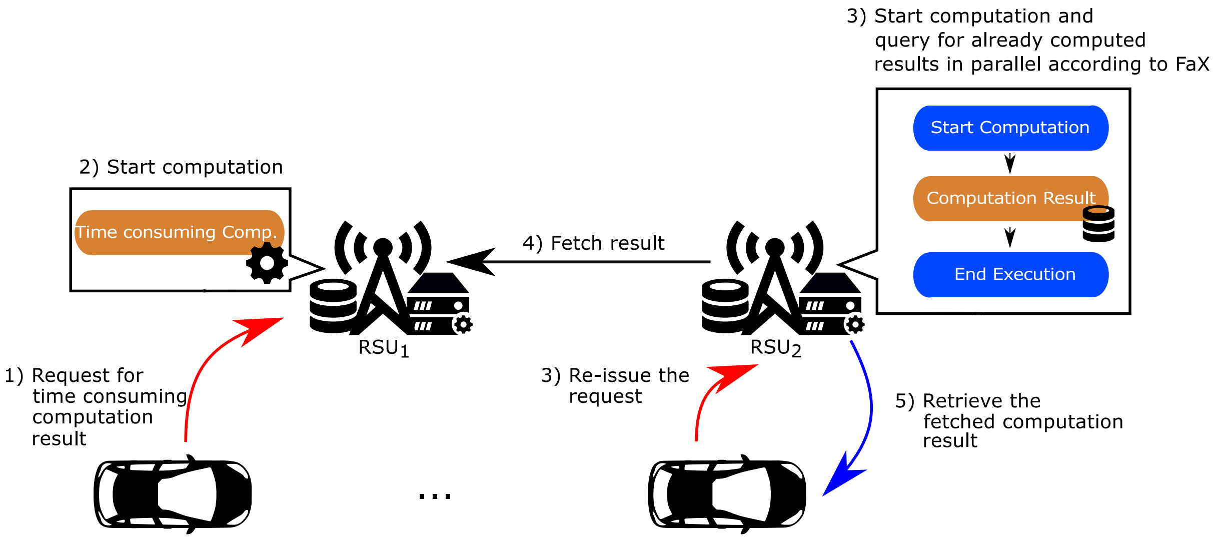

3) Start computation and query for already computed results in parallel according to FaX
Start Computation
Computation Result
End Execution
2) Start computation
Time consuming Comp.
4) Fetch result
RSU$_1$
RSU$_2$
1) Request for time consuming computation result
3) Re-issue the request
5) Retrieve the fetched computation result
...

Figure 7.6: Experimental set up of the vehicular NFN prototype on the road. The setup consists of two infrastructure nodes directly connected with each other and one vehicle. The prototype implementation is deployed on all nodes including the support of the FaX strategy.

Summarized, both presented novel NFN resolution strategies, namely FaX and FoPaX, focus on the timely delivery of computation results in mobile networks. The deployment of these strategies is dependent on the applications requesting for more or less complex computations in the network. Therefore, the NFN network layer needs to be flexible to support the presented as well as future resolution strategies.

7.3 Evaluation of the Resolution Strategies

A simulation setup is created, in order to evaluate the performance improvements of the presented resolution strategies. While there is no implementation module of NFN available for ndnSIM, the simulation environment is created using the open source NFN implementation Python ICN (PiCN) [290] and its simulation layer.

Scenario and Setup

The scenario to be considered involves both, short running and long running (time-consuming) Named Functions. An example for the first type of computation includes the conversion of a value from Celsius to Fahrenheit from a temperature sensor, while examples for a long running computation may include the fusion of massive sensor data or from the domain of augmented/virtual reality. Within the simulation setup, the execution of long running computations takes more time than the vehicle is connected to the communication infrastructure. Within the simulation setup, it is assumed that in-vehicle components are not able to compute the results, e.g., due to constrained built-in resources or the lack of required information from a cloud infrastructure. In this case, a vehicle can offload the computation to NFN infrastructure nodes, able to process and compute the requested function (cf. Section 1.2.1 Electronic Horizon application). Figure 7.6 shows the processing steps of the use case scenario. During the journey, a vehicle sends out a request to offload a computation as part of the Electronic Horizon

Table 7.1: Simulation parameters used to evaluate the performance of FoX and FaX strategy.

Parameter	Values
no. named functions	30 unique functions
no. RSUs	20
communication range RSU	50 meters
no. vehicles	1–80
vehicle speeds	50–120 km/hour
Request rate	2–10 seconds
result vitality	2–10 seconds
no. runs each	500

application (Figure 7.6, Step 1). A NFN-enabled RSU resolves the request and starts the computation immediately (Figure 7.6, Step 2). However, the vehicle has left the communication range of the RSU$_1$, and hence, is not able to receive the computation results. As a consequence, the vehicle sends the request again when reaching the next RSU (Figure 7.6, RSU$_2$). Following the FaX resolution strategy, RSU$_2$ is able to fetch the computation result and serve it to the vehicle (Figure 7.6, Step 5).

The simulation setup includes a increasing number of mobile nodes traveling with varying velocity ranging between 50–120km/hour and joining the simulated road segment randomly from different directions. The mobile nodes request for computation results of a set of 30 named functions, assigned to the mobile nodes by following a Zipf-like distribution with coefficient 0.8 [285]. The infrastructure consists of 20 RSUs processing and executing the function requests. Short running and long running named functions have been defined in order to evaluated the performance of the novel FaX resolution strategy compared to the standard FoX strategy in NFN. Caching is enabled only at the infrastructure nodes, while time a results stays in the cache before staled is ranging between 2–10 seconds. Simulation runs are performed for each type of computation (short vs. long) for both resolution strategies by increasing number of vehicles. Table 7.1 provides an overview of the simulation parameters used.

7.3.1 Results of the Simulation of Resolution Strategies

As the FaX strategy is created to increase the delivery rate of computed results, the main focus of the metric to be analyzed is the resolved `INTEREST` ratio RR_I (cf.Section 5.7). Figure 7.7 illustrates the performance evaluation of the RR_I for both resolution strategies and the aspect of computation time.

The results of FoX for short running computations are within the connection time of a mobile node to the executing RSU. They show that the number of delivered results are close to 100%, decreasing slightly with the number of mobile nodes ($RR_I \approx 93\%$ for 80 vehicles). The performance of FaX for short running computations shows similar values as FoX. The results show slightly improved performance with an higher number of mobile nodes ($RR_I \approx 95\%$ for 80 vehicles). This is due to the fact, that the RSUs immediately start the computation, while FoX tries to find a cached copy in the network first. It can be observed that the number of enqueued computations at each RSU increases with the number of mobile nodes.

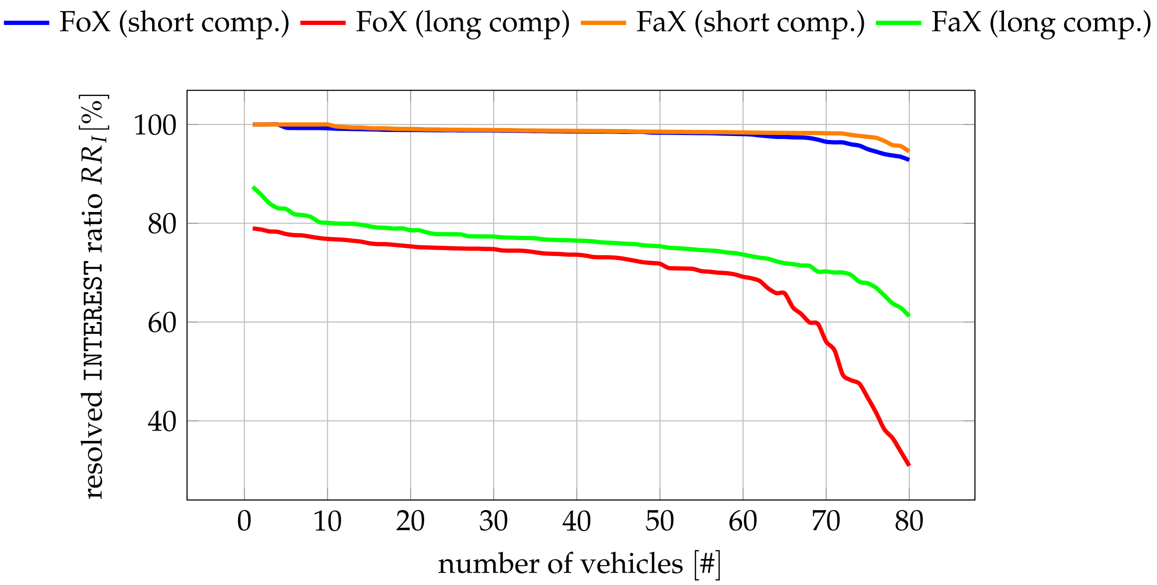


Figure 7.7: Results of the resolved `INTEREST` ratio for the FoX and FaX resolution strategy in NFN. By executing computation and pulling computation results in parallel, the resolved ratio is increased using the FaX strategy.

When looking at the results of long running computations, it can be seen that the values for the resolved `INTEREST` ratio is different for FoX and FaX. Since the mobile nodes join and leave the communication range of the infrastructure nodes, the computation requests trigger the *find* operation of FoX at each RSU multiple times, before executing the computation locally. Such behavior results in additional expenditure of time. Dependent on the number of mobile nodes, the computation requests resolved by the default FoX strategy ranges between $RR_I \approx 80\%$ (low no. mobiles) and $RR_I \approx 35\%$ (high no. mobiles).

When applying FaX to the simulation, the ratio values for long running computations are increased between $RR_I \approx 9\%$ (low no. mobiles) and $RR_I \approx 28\%$ (high no. mobiles). This is due to the fact, that computations are immediately executed while an infrastructure node receives the request. In parallel, the *find* operation is used to consult neighboring nodes for cached replica of an already computed results. Such strategy speeds up the data delivery. however, causes additional protocol overhead (ranging between 5% (low no. mobiles) up to 34% (high no. mobiles)) in the network. It also increases the utilization of computation resources at the RSUs up to 20%. While in ICN `INTEREST` packets are relatively small, the additional overhead should not be underestimated.

As it can be seen from the comparison, both strategies are performing similar in the simulation setup with respect to an average number of mobile nodes ranging from 20–60. There are two influencing factors: First, the number of requests for computations and the available resources at the RSUs are keeping each other in balance. Second, the advantages of caching influences the delivery ratio, as computation results can be shipped directly from the caches. However, the performance is highly dependent on the computation time of a function to be executed. If the expected computation time is known a-prior (e.g., a named function is annotated with some meta-information) and the resource utilization at the edge is low, a deployment of FaX may be beneficial. Otherwise FoX can be still used as a fallback strategy.

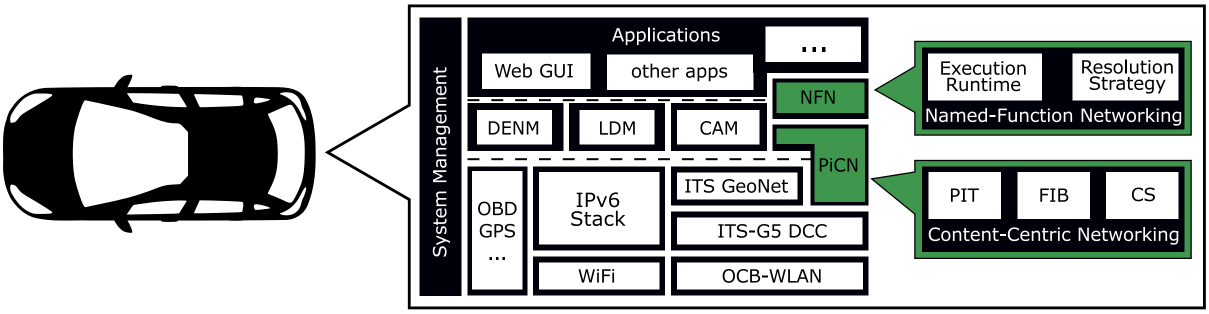


Figure 7.8: Computation-centric vehicular prototype: Presentation of the elements of the prototype including NFN as well as ETSI ITS-G5. The prototype allows for parallel information exchange using NFN and ITS-G5 protocols.

7.4 A Network Stack for Computation-Centric Vehicular Networks

In order to evaluate the applicability of the presented resolution strategies, a real world prototype is created. It is based on the prototype implementation presented in Section 5.8. Instead of using the Named Data Networking platform to exchange information, the prototype uses the PICN [290] implementation for Named Function Networking. Besides the feature to support function execution, the implementation also supports several resolution strategies including FaX.

7.4.1 Integration of Named Function Networking into ETSI ITS-G5

According to the structure of the vehicular ICN prototype implementation supporting ETSI ITS-G5 (cf. Section 5.8), the computation-centric prototype also consists of three building blocks:

- **IEEE 802.11p**: A wireless communication access layer supporting the vehicular IEEE 802.11p standard, based on a Linux kernel modification for the Atheros 9k WLAN chip series from the Czech Technical University of Prague [274]. As part of the prototype, the ITS-G5D band (5905MHz - 5925MHz frequency with 10MHz channel spacing) and the service channel G5-SCH5 are used which are reserved for non-safety future applications [35].

- **Vehicle Platform**: A communication prototyping platform supporting ETSI ITS-G5 vehicular standard, based on the `OpenC2X` platform from the University of Paderborn [276].

- **NFN**: The PICN [290] platform implementation from the University of Basel supporting CCN and NDN on the transport layer as well as an additional NFN layer.

Figure 7.8 illustrates the integration of a computation-centric network stack into the prototype. The required data structures to provide ICN information exchange is supported via the PICN layer. On top, the implementation provides the NFN related elements such as a function execution runtime as well as resolution module including the FoX and FaX strategies.

7.4.2 Functional Tests of the FaX Resolution Strategy

The advantages of the FaX strategy, as presented in Section 7.3, have also been evaluated as part of a real world deployment on the road. According to the vehicle scenario presented in Section 7.3 and illustrated in Figure 7.6, the setup consists of three components (IPC Board NF36-2600 - 1GHz CPU, 1GB RAM):

- *Two infrastructure nodes*: used to deploy a computation-centric infrastructure network, both installed as execution nodes and able to execute named functions. Both nodes are deployed along a road with a distance of 50 meters to each other. The access medium is based on the IEEE 802.11p standard in the G5-SCH5 service channel. The communication radius for each RSU has been adjusted to be 20 meters.
- *One in-vehicle node*: used to request for a computation-intensive function result. The in-vehicle component also uses the IEEE 802.11p standard in the G5-SCH5 service channel. The communication range has been adjusted to be 15 meters. The vehicle travels with a velocity of 20km/h.

Results

The vehicle travels with a velocity of 20km/h through the deployment set up. As part of the first tests, the vehicle is not able to receive a result directly from the RSU_1 which received the request first, since the average processing time of the named function is configured to be greater than 5 seconds. As a result, the vehicle sends out a second request for the function execution at the next RSU (here: RSU_2) which restarts the computation and thus occupy resources in the network. Following the FaX resolution strategy, RSU_2 consults the network for a previous result, fetches it from the neighboring node and delivers it to the vehicle. The results of the tests have shown the advantages of offloading computation-intensive tasks from a vehicle to a NFN-enabled infrastructure network. The introduction of the FaX strategy has increased to accessibility of function results for mobile participants and therefore increase the network performance in vehicular NFNs.

7.5 Towards Management Strategies in Vehicular Named Function Networks

As presented in the previous section, the support of decentralized and distributed computations applied to a vehicular NFN network is beneficial. The support of several resolution strategies extends the application area of NFN in IoT networks such as connected vehicles. The prototype implementation including the FaX strategy has shown performance improvements by increasing accessibility of computation results in the network.

The introduction of requesting for computations using an ICN substrate as NFN supporting different resolution strategies as well as the topic of active content placement complement each other in two dimensions:

Named Function Networks as Enabling Platform: NFN defines a networking paradigm to support the execution of functionality directly in the network. As a part of the vision of an adaptive framework for active content placement in ICN-based connected vehicle environments (cf. Section 5.2), NFN defines an enabling platform to realize the strategies in such a framework. Proactive placement strategies such as PeRCeIVE, ADePt, and the enhancement of placing Information Objects in virtual cache areas can be realized as part of in-network functions executed at geo-specific nodes (e.g., in case of PeRCeIVE), directly at RSUs along the road (in case of ADePt), or as a part of a set of distributed functions (e.g., in case of vCaches) to increase the availability of data and computation results in connected vehicle environments.

The introduction of the novel resolution strategies presented for NFN into such a framework, combined with the proactive caching strategies, are promising to increase the delivery of computation results in mobile scenarios. However, a flexible selection of the right resolution strategy during runtime is dependent on further metrics such as current utilization of network and compute resources of the execution nodes involved. Furthermore, specific application requirements such as weak/hard real-time constraints, or hardware specific requirements are influencing factors for the decision making of the most suitable execution node (e.g., certified hardware) incl. caching decisions. In such a framework, those information have to be accessible to ensure efficient decision making.

As of today, the results of the evaluation have shown that the FaX strategy is useful if the load in the network is low as well as compute resources are available. However, as the FaX strategy increases the utilization of computation resources at the RSUs (cf. Section 7.3.1) by immediately starting to execute a function, it might lead to an infrastructural DoS attack. A individual network entity can occupy the computation resources at the edge by requesting for a result of a long and computation intensive Named Function multiple times. In case both the number of requests from individual network entities as well as the load factors increase, the framework can decide to switch to the FoX strategy.

Active Placement Strategies to pre-load Computation Inputs: The second dimension introduces active placement strategies as enabler to increase the performance of a Named Function Network. Such strategies can be used to pre-load required computation input such as data items or function results closer to execution nodes in the network. Depending on the function as well as the amount of required input data, the presented proactive placement strategies can improve the computation performance by reducing the retrieval time of required computation input.

Summarized, proactive placement strategies and computation-centric networks such as NFN complement each other and are beneficial to be deployed in data-oriented connected vehicle environments.

7.6 Summary

In the future, vehicle applications will require access to external information, e.g., from network services, or offload computations from the vehicle to the network. This is due to the fact, that the vehicle is not able to collect data, or to process information by itself due to limited resources. Computation-centric networks support the access to decentralized and distributed computation. One example of such a network is NFN, however, it was originally designed for data center deployments lacking efficient support of mobile networks. While the introduction of proactive placement strategies helps to bring data closer to in-network computations, the resolution strategies of NFN have to be adopted to the characteristics and the needs of mobile networks.

The presented resolution strategies, namely FaX and FoPaX, have shown performance improvements by increasing the accessibility to function results in a NFN. Based on a real world prototype implementation, one of the strategies have been tested as part of functional tests on the road.

The deployment of proactive placement strategies within a NFN supporting the delivery of function results in mobile networks is beneficial to be deployed in future connected vehicle environments.

8 Dealing with Cached Objects in Data-oriented Vehicular Systems: A Security & Privacy Perspective

> Security is better when it's built-in – not bolted out.
>
> Stephen Yu - Infoblox, Inc.

Connected vehicular communication networks posses several challenges due to the high mobility of their network participants. The presented features such as simplified access to content as well as the in-network caching capabilities of ICNs, including its computation-centric enhancements, have shown improvements regarding the availability of content, especially in connected vehicle environments (cf. Section 5, Section 7). The active placement of content or named functions multiple times in the network implies security and privacy threats. The decentralized and distributed nature of ICNs challenges today's end-to-end security and privacy concepts, becoming more challenging in high dynamic scenarios such as vehicular networks where participants frequently join and leave the network.

The following section introduces security and privacy challenges concerning cached objects and named functions in ICN-based vehicular networks and discusses opportunities to overcome these challenges. As one important security aspect, access control management of cached objects is discussed more in detail. Based on the findings, an access control concept for cached objects is presented and evaluated against common security threats[12].

8.1 Problem Statement: Cached Objects in Connected Vehicle Environments

Instead of adding security features to communication technologies afterwards (e.g., as often presented in host-centric networks such as IP), security features are an essential part of ICN research (e.g., [138]).

As stated in the previous chapter, storing content proactively in the network (e.g., at the edge) is beneficial to increase the availability, while decreasing delivery times (cf. Section 5). The loosely coupled communication model in ICNs simplifies the access to data from a consumer perspective, however, producer mobility is still considered to be difficult to solve [138]. In this manuscript, the consuming entity is of interest in the following sections.

Figure 8.1 illustrates the system architecture to be considered for the introduction of the security and privacy problem statement. Consuming nodes query the network for information or computation results using naming schemes. For example, a function is invoked by a vehicle using location-independent names (e.g., in form of a lambda expression as in NFN) dependent on the underlying architecture. One option to use is a proposed encoding scheme by Pesavento et al. [128] to represent geographic area as part of hierarchical names.

[12] The work in this chapter is published in the Proceedings of the 2017 Network of the Future Conference [21], and the Special Issue of Information-Centric Networking Security of the IEEE Communications Magazine [22]. Parts of it are extracted from these sources.

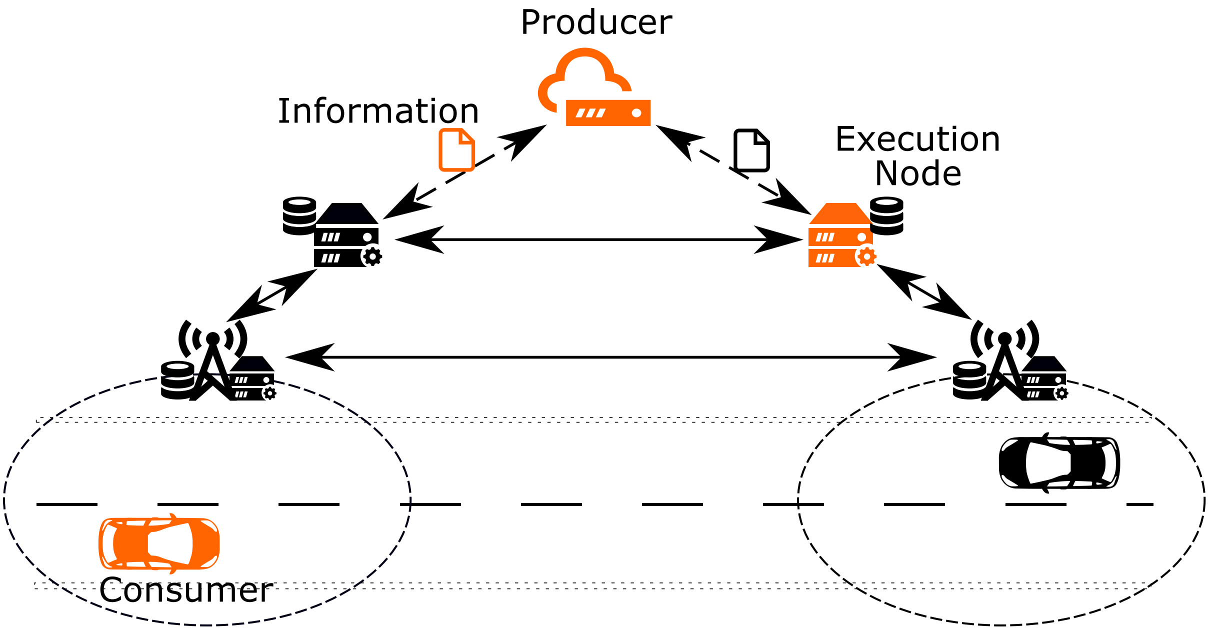


Figure 8.1: System architecture of an information/computation-centric vehicular system. The assets to be protected include the producer of information/function, the execution node computing a results, the consumer requesting for information, as well as the information itself. The assets are highlighted in orange in this example.

A query is processed by the network in order to find the desired information/result (e.g., within the local cache of a node or at least at the producer), splitting a computation request into sub-tasks, or find the right execution node to perform and compose computations. Following the loosely coupled communication model of ICNs, delivery of Information Objects as well as the execution of a computation is not bound to a specific physical node. Consecutive queries can be satisfied by multiple nodes simultaneously, or already created objects are distributed for reuse from caches in the network. While such mechanisms enable a high degree of flexibility and scalability of a networked system, storing content multiple times in a decentralized and distributed network poses several threats and attack vectors from a security and privacy perspective. In this regard, the following actors and assets need to be protected include (cf. Figure 8.1):

- **Consumer**: The entity in the network requesting for information. In this thesis, consumers are represented by vehicle nodes.
- **Producer**: The entity in the network creating and providing information[13]. In a *computation-centric network*, this also includes the provision of executable function code.
- **Execution Node**: In a *computation-centric network*, the entity performing the execution of a function provided by a producer entity. The result of such function execution is a Information Object which is transferred through the ICN network.

[13] As part of this section, the term ”producer“ and ”originator“ are used interchangeably.

- **Information**: The actual information which is transferred through the network as part of an Information Object . In a *computation-centric network*, function code as well as the function result are disseminated through the network as part of Information Objects.

In this thesis, the consuming entity is of interest in the following sections. According to this perspective: it does not matter from which node an Information Object or function result is delivered, as long as the received content is correct, valid, verified and trustworthy. However, this is challenging due to the intermittent connectivity of participants in vehicular networks.

8.1.1 Security Challenges and Requirements

Considering the different actors and assets, the following challenges are related to an ICN-based connected vehicle environment (including the computation-centric paradigm):

C1 **Authenticity and Integrity**: How can a consumer as well as an execution node verify that the data received or the request to invoke a computation is created from a trustworthy network participant, while it has not been tampered on the delivery path across a non-trusted distributed environment?

C2 **Confidentiality of information items**: How can eligible consumers, execution nodes and the producer ensure that submitted input parameters as well as received data is only readable by eligible entities?

C3 **Revocation of access rights**: How can a producer revoke the access to data or prevent the execution of functions for a specific user or group without hampering access of other eligible consumers across a distributed environment?

C4 **Automated Interoperability**: How to provide interoperability between various security mechanisms implemented by different vendors required to interact in a decentralized environment?

Based on the introduction of the security challenges, the requirements can be derived using the use case descriptions of Section 1.2:

In the electronic horizon use case (cf. Section 1.2.1), a consuming vehicle need to be able evaluate whether a collision warning message is **valid** and **trustworthy**, independent of the physical node that has served the Information Object. In order to ensure the protection of business models, a producer of information must allow the **access** to valuable data only for eligible consumers (e.g., for a payed concierge service to offer recommendations). Furthermore, a producer must be able to **revoke the access** to information, e.g., the permission to access information of the concierge service has expired for a specific consumer.

Regarding networks supporting in-network computations, the result of an offloaded computation (e.g., requesting for available parking spots nearby as part of the community-based sensing use case in Section 1.2.2) needs to be at least **verifiable** and **trustworthy**, while **preventing required input parameters** (e.g., current geographical coordinates) to be leaked. An execution node receiving a request must be able to **verify the identity** of the consumer, while the input parameters provided by the consumer must be protected due to privacy reasons.

8.2 An Encryption-based Access Control Mechanism for Information-Centric Connected Vehicles

One option to tackle the challenges C_1 and C_2 is to use cryptography mechanisms – in the form of encryption schemes. Actually, encryption is a subset of the field of cryptography and describes a mechanism in which information (plain-text or clear-text) is transformed into seemingly random incomprehensible data (cipher-text). Algorithms in this domain use key material, in order to encrypt or decrypt information from plaintext to cipher-text and vice versa. By combining identity management and key management together in a system, access control methods can be implemented in order to realize authorization and deny or allow access to information in a system [291].

However, the loosely coupled communication model in ICNs in combination with the high degree of mobile participants in vehicular systems challenge the distribution of key material. This is due to the fact that the model decouples data from physical locations. As a result, there might not be a direct end-to-end communication between a consumer/producer of data, or an execution node. Instead, the data and required input parameters need to be secured in such a way, that it can provide such security features in a loosely-coupled way without continuous end-to-end connectivity. This requires novel strategies for access control mechanisms and security in ICN.

8.2.1 Introduction to Encryption-Based Access Control Mechanisms

The objective to transfer private information across a public non-trusted network, e.g., to hide it from unauthorized entities, still describes a research field over several decades. When referring to cryptosystems, it is distinguished between two types of techniques in general [292]:

- **Symmetric Cryptosystem**: The same key material or at least one key pair (to derive one key from another) is hold by both communication participants. These keys are used to introduce uncertainty during the encryption and decryption process, in order to prevent the access to information for unauthorized network entities.
- **Asymmetric Cryptosystem**: Both communication participants hold different, independent key materials, which makes it impossible to derive one key from the other.

The key material of symmetric techniques require less computational resources than asymmetric techniques, hence, it seems to be useful in the first place. However, they have disadvantages, especially when transferring data securely through a public, non-trusted network such as the Internet. As a result, asymmetric techniques have gained acceptance in the past decades.

Public-Key Cryptography

Public-Key Cryptography (PKC) describes a cryptosystem following the principles of a asymmetric system. In PKC, a pair of keys are used to secure the access to information: a public key (PK) – shared by the owner in the network to be used by communication partners, and a Secret Key (SK) – only known by the owner of the key pair and never shared in the network [292]. In order to provide access to PKs for potential communication partners, a Public-Key Infrastructure (PKI) is required to manage and distribute PK material. Such infrastructure consists of one or multiple authorities which are responsible to bind certain PK material to the identity of a network entity. Every time a network entity creates a public/secret key pair, the PK is registered to an PKI. As a result, PK material can also be used in order to verify the identity of a network entity. Examples of popular PKC algorithms include RSA [293], Elliptic-Curve Cryptography (ECC) [294], or Identity-based Cryptography (IBC) [295].

Over the past decades, several techniques for PKC cryptosystems have been proposed. Regarding the distributed and decentralized fashion of ICNs, mechanisms such as from the domain of IBC are promising. One of such techniques is Attribute-Based Encryption (ABE) which is promising for the IoT including connected vehicle systems.

Attribute-Based Cryptography

Based on principles of IBC, ABE introduces a public-key encryption mechanism in which the secret key of a user depends on attributes including access policies. In such a system, key material and cipher-texts are labeled with sets of descriptive attributes. Such cipher-text can be decrypted if there is a match between the attributes and the key material of a consumer. Depending on the cryptosystem, a successful decryption of a cipher-text is possible when at least k attributes match the key material of a specific consumer [296].

There are mainly two types of ABE: (i) Key-Policy Attribute-Based Encryption (KP-ABE) [296] where an access tree is described by a set of the consumer's secret keys defining the privilege scope, and (ii) Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [297] where the access tree is described by the attributes included in the cipher-text. The major strengths of ABE is a flexible encryption mechanism as well as the possibility to enforce fine-grained access control to data for certain consumers and groups. As part of this manuscript, the CP-ABE scheme is introduced in detail.

In CP-ABE the access policies are created using the attributes of all eligible users and are directly incorporated into the cipher-text. The combination of multiple access policies with each other is represented as a data structure. The structure is defined by a collection $A \subseteq \{P_1, P_2, ...P_n\}$ where the elements $P_1, P_2, ..., P_n$ form a collection of all possible access policy values. All values that are part of A are indicating authorized access, while all other elements not part of A are called unauthorized [297]. The policy is expressed as an access structure which allows for AND and OR gates as well as numerical comparison (in a limited way). The following example describes an access structure AS providing fine-grained access for consumers of a stream of parking information as part of the electronic horizon use case and illustrated by Figure 8.2 and Listing 1. The structure AS is divided into two sub-trees T_1 - T_2. The first subtree defines a specific geo-location in which every vehicle within the area is allowed to access the data (cf. Listing 1, line 1). The second sub-tree T_2 is defined by a group related attribute in which every subscribed consumer of such service is eligible to access the data until the specific date of "06/2019" (cf. Listing 1, line 2). In this example, a consumer is able to access the information if his key material matches at least one tree of attributes.

Listing 1: Example of a hierarchical access policy structure given as cipher-text.

```
('GEO': "lat:48.8" AND 'GEO': "long:8.92") OR
('parking': "until_06/2019")
```

The example illustrates a simple hierarchical access control structure to provide fine-grained access for both one-to-one and one-to-many communication relations within the same encrypted data.

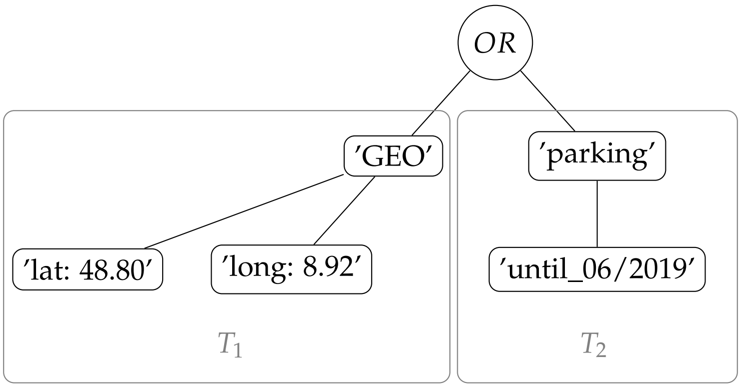


Figure 8.2: Simple illustration of a CP-ABE policy structure consisting of two sub-trees and expressed as an access tree which allows for AND and OR gates as well as numerical comparison (in a limited way). A consumer is able to decrypt the cipher-text if there is a match between the attributes of at least one sub-tree and the key material of the consumer.

However, the original ABE mechanism is limited to a single central authority to generate and manage attributes. Within a highly mobile environment with sometimes unsteady connections such as the case with vehicular networks, relying on a single centralized authority to manage security mechanisms is not feasible. To overcome this challenge, Mueller et al. [298] introduce a distributed version of ABE based on multiple authorities which are responsible for a certain set of attributes. However, the introduction of such authorities still introduce a dependency for producer and consumer on such central authorities. Within highly mobile environments such as connected vehicles, such dependency might be a problem regarding availability. Furthermore, additional algorithms in the ABE scheme are required by [298], which would cause significant drawbacks.

Related Work of Attribute-based Encryption in Information-Centric Networks

In the past years, several access control schemes have been proposed for information-centric networks. Misra at al. [299] and da Silva et al. [300] use broadcast encryption and attribute-based encryption to enable instantaneous access control and revocation in a NDN. However, the mechanisms rely on a continuously available centralized PKI authority. In a connected vehicle environment, such dependability can not be ensured due to sometimes unsteady connection to a infrastructure network.

The challenge of a centralized authority is addressed by Ion et al. [301]. The authors propose the usage of distributed access control policies based on attribute-based encryption to support data confidentiality. The proposed scheme supports large scale environments with no need to share keys. However, it relies on a trusted authority in order to perform the encryption and decryption activities.

In [302], the authors propose an encryption based access control solution using access control lists based on the principles of HIBE. This is suitable only in cases where the namespace is centrally managed and prior knowledge of all the authorized entities is available. This approach can not be applied to a dynamically evolving open data market place. It also lacks the trust-bootstrapping required to establish trust on the namespace manager.

The authors of [303] motivate the introduction of both a centralized and decentralized approach for trust management and access control. The authors introduce four different approaches for key distribution using centralized authorities. One approach describes a key dissemination by using the caches of routers to store keys if the authority is not available. However, such a decentralized approach is questionable, since keys are stored within the caches of multiple routers.

So far, none of the existing work have taken the specific requirements of automotive use cases into consideration.

8.2.2 The Concept of EnCIRCLE

In order to provide an access control mechanism for information-centric connected vehicles, the **En**cryption-based access **c**ontrol for **i**nfo**r**mation-centric **c**onnected Vehic**le**s (EnCIRCLE) framework is proposed. It focuses on supporting mechanisms to exchange Information Objects securely between consuming nodes (e.g., vehicles, functions) as well as providers of data or computation results through a non-trusted, open and distributed network.

The main concepts of EnCIRCLE are built as part of the NDN architecture [13]. The framework describes a mechanism to exchange Information Objects securely through a non-trusted, open and distributed network. These objects may include static data, computation results as well as the transfer of in-network function code. In the case of vehicular networks, it is safe to assume that there are mechanisms in place to authenticate users that are either required by law for security critical applications, or there most likely is already a trust relationship between a car and the cloud services of the appropriate manufacturer. Such first register procedure of a vehicular device or component is required to bootstrap trust in a system. Since this is a well investigated topic in ICN, mechanisms such as NDN DeLorean [304] can be used in the first place.

In order to cope with most of the challenges introduced in Section 8.1.1, EnCIRCLE consists of four building blocks:

B_1 **Authentication and Integrity** - This building block describes the verification of the authenticity of the original sender of a received data packet. The verification includes the validity of data, in order to reject outdated ones, e.g., rejection of data coming from a message replay. Depending on the use case, the authentication of consumers, producers, or execution nodes is desirable. For example, when the business case rests on the consumer requests sent. The authentication of a producer is required, while the need for a consumer to authenticate herself can be optional for privacy reasons (e.g., in order to prevent consumer tracking). However, consumer authentication is required in case of a in-network function request.

B_2 **Confidentiality** - This building block describes an encryption mechanism to protect the access to data and function results. The Attribute-Base Encryption [297] mechanism is used which provides an access control structure as part of the encrypted Information Object.

B_3 **Onboarding / Joining** - This building block describes the exchange of encryption related information between the producer/execution node and the (new) consumer, enabling the consumer to access encrypted information.

B_4 **Revocation of access rights** - This building block describes the revocation procedure, when a producer/execution node decided to withdraw the access rights for a certain consumer or a group of consumers.

Authentication and Integrity

Being able to verify the authenticity of a received `DATA` packet serves as a proof to the consumer, as well as an execution node, that the received information is sent by a trustworthy producer. Since in NDN every `DATA` packet must be signed by the creator (cf. Figure 5.1 in Section 5.1), signatures are used to authenticate the producer as well as to verify that a packet has not been modified along the path [13]. This also includes a function execution node, which has to sign a `DATA` packet created after a successful computation. The signature is calculated over all fields, including *Name*, *MetaInfo* and *Content* [253]. As part of the concept, information related to a specific producer (e.g., unique key material) is added to a packet and used in a signature over the packet's contents. EnCIRCLE allows to define a use case specific algorithm f to calculate the signature, using the producer specific key material, like for example specified in ECC [294].

A consumer needs to extract the signature value and related information provided by the *SignatureInfo* field of the received packet (cf. Figure 5.1 in Section 5.1). The consumer verifies this `DATA` specific signature on its own if $I_{Producer}$ and the algorithm f is already known to the consumer or via a trusted service present in the network. This procedure is similar to the structure of a IBC cryptosystem. Furthermore, the freshness period stored in the signed *MetaInfo* enables a consumer to reject outdated data.

From a electronic horizon scenario perspective, it is necessary to know that the received data is authentic, not modified during delivery and fresh. To validate the incoming `DATA`, the introduction of a trusted service provider might be a challenge due to intermittent connectivity of vehicle nodes. In a mobile system, it is desirable that participants can validate received packets by themselves. As EnCIRCLE uses principles from the decentralized ABE, encrypted messages can be verified and accessed by communication participants offline in a decentralized fashion [305]. Nevertheless, the concept of EnCIRCLE recommends the introduction of a hybrid solution: A trusted backend service enables network participants to authenticate data in case of an existing connectivity, otherwise the authentication procedure is performed by the vehicles themselves for a certain time.

Confidentiality via Attribute-Based Encryption

Confidentiality ensures that only eligible participants can read content from a received DATA packet. As part of the EnCIRCLE concept, the flexible encryption mechanism ABE is used, providing the possibility to enforce fine-grained access control to information on a user basis. However, the original ABE mechanism is limited to a single central authority to generate and manage attributes. Within a highly mobile environment with sometimes unsteady connections such as is the case with vehicular networks, relying on a single centralized authority to manage security mechanisms is not feasible. [298] introduces a distributed version of ABE based on additional authorities which are responsible for a certain set of attributes. However, the introduction of such authorities still introduces a dependency for producer and consumer on such central authorities.

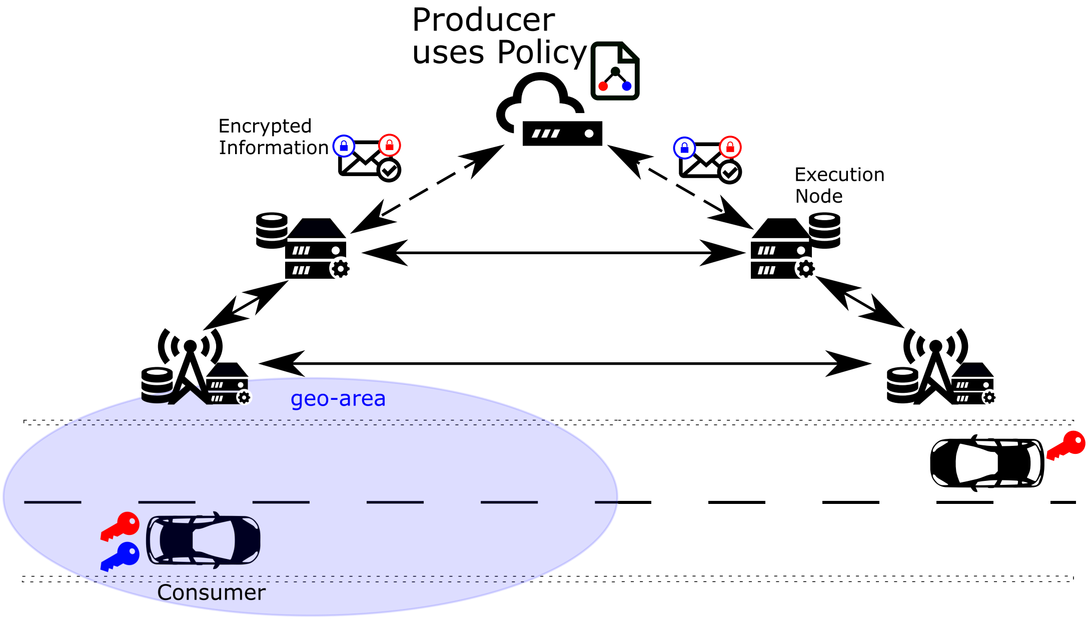


Figure 8.3: Architectural view of EnCIRCLE. A producer encrypted data using a policy structure consisting of attributes. A consumer can access the data if its key material matches the attributes within the structure. In this figure, the car on the left hand is able to access the data, the car on the right hand side is not due to a missing key.

The encryption concept in EnCIRCLE is separated into two components: (i) an access management component responsible for managing policies within the ABE access structure, and (ii) a cryptography component responsible to encrypt content, based on the policies given by the access structure. EnCIRCLE allows but does not require that both components can be deployed separately on different devices, which provides flexibility in a distributed environment. In this manuscript, both components are located at the `DATA` creator side (including the producer as well as a function execution node), while only the encryption-based access component is located at consumer side for decryption.

Access Management Component: In CP-ABE, the access policies are created using the attributes matching the key material of all eligible users. The combination of multiple access policies with each other is represented as a data structure. As stated in Section 8.2.1, the structure is defined by a collection $A \subseteq \{P_1, P_2, ...P_n\}$ where the elements $P_1, P_2, ..., P_n$ form a collection of all possible access policy values. Only values within A are eligible to access the information [297]. In EnCIRCLE, the elements of collection A are proposed to be handled in two ways: (i) by an external instance to manage authorized values, such as the vehicle manufacturer cloud or a third party provider or (ii) by the producer itself to be executed standalone (see Section 8.2.2).

Similar to the access structure example in Section 8.2.1, Figure 8.4 and Listing 2 illustrate an extract of an access structure in EnCIRCLE in cipher-text. Based on the structure introduced in Section 8.2.1, the concept of EnCIRCLE defines an additional sub-tree T_3. It defines a set of predefined attributes added by the creator of a `DATA` packet proactively (e.g., the producer or an execution node) to cater for consumers with access to already disseminated data. For example, this is the case, if a producer has encrypted and delivered an Information Object which is

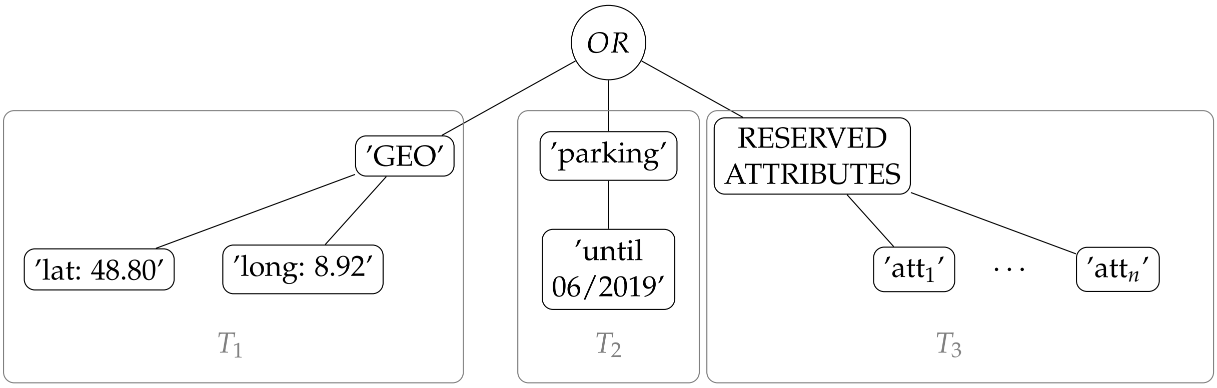


Figure 8.4: Illustration of an access policy structure used within the EnCIRCLE concept. The structure consits of three sub-trees expressing an access tree which allows for AND and OR gates. In the EnCIRCLE concept, reserved attributes are introduced as part of the access structure to simplify the access of new consumers for already encrypted content.

theoretical cached multiple times in an ICN. When a new consumer querying the network for such Information Object may retrieve a already encrypted item, however, not including the key material of the new consumer. Using predefined attributes, a producer is able to assign these attributes to the new consumer. This one-time access is necessary to overcome the problem of cached data which is already encrypted by the creator of the packet. Such mechanisms requires a key exchange procedure which is introduced in Section 8.2.2.

Listing 2: Example of a hierarchical access policy structure as used in the EnCIRCLE framework.

```
('GEO': "lat:48.8" AND 'GEO': "long:8.92") OR
('parking': "until_06/2019") OR
('RESERVED ATTRIBUTES': "att1" AND 'RESERVED ATTRIBUTES': "att2", ...)
```

Cryptographic Component: The cryptographic access component consists of five algorithms. They are based to the CP-ABE encryption scheme presented by Bethencourt et al. [297]:

- **Setup** – Require A: The algorithm is used to generated public parameters PP as well as a master key MK. The public parameters PP corresponds the element $\{P_1, P_2, ...P_n\}$ of an access structure $A \subseteq \{P_1, P_2, ...P_n\}$. In order to be aligned to the decoupled design of ICNs, the master key MK can also be generated by the producer or execution nodes themselves to be able to introduce new parameters or to adjust the access structure if required, according to Lewko et al. [305].

- **Key Generation** – Require MK and A: By using the access policy structure A and the generated master key MK, the algorithm is used to generate private key material required to decrypt a cipher-text. In EnCIRCLE, this algorithm is also deployed at data creator side (e.g., producer or execution node). In case an entity provides access to several Information Objects, it can use different master key material for each item independently. In order to provide more flexibility, the algorithm can also be deployed as part of a manufacturer cloud infrastructure.

- **Encrypt** – Require PP, A, and $Plaintext$: By using the public parameters PP and the access policy structure A, the algorithm encrypts a given plain-text into a cipher-text CT. It is deployed on either of the creators including a producer or as part of an in-network function execution node.

- **Re-Encrypt** – Require CT, PP: The re-encryption algorithm uses a set of public parameters PP and the cipher-text CT including the access policy structure A. The algorithm returns a re-encrypted cipher-text in which only consumers holding key materials for the corresponding PP are able to decrypt the ciper-text. For example, such procedure takes action, if some attributes are removed from the set of public parameters. It is deployed on either of the creators including a producer or as part of an in-network function execution node.

- **Decrypt** – Require PP, CT, and SK: This algorithm is deployed at consuming nodes. It uses the public parameters PP and the secret key material SK to decrypt a received cipher-text CT. In case PP and SK matches the access policy structure, the algorithm outputs the plain-text.

Depending on the configuration of the cryptosystem, the deployment of the algorithms may vary. For example, the *Setup* and *Key Generation* algorithms can be under control of an external instance, such as the manufacturer cloud infrastructure or the government.

On-boarding / Joining

In order to allow the access of information to eligible network entities only, it is requires to share key material between the encrypting and decrypting entities in the network. In EnCIRCLE, such information sharing is handled in the *on-boarding/joining* building block.

When looking into the community-based sensing use case (cf. Section 1.2.2), it is conceivable that either, individual vehicle or a dynamic ad-hoc group (e.g., for all vehicles within a specific geo-location), are granted access to encrypted data. This also includes function execution nodes which are pre-processing and filtering data to provide a computation result.

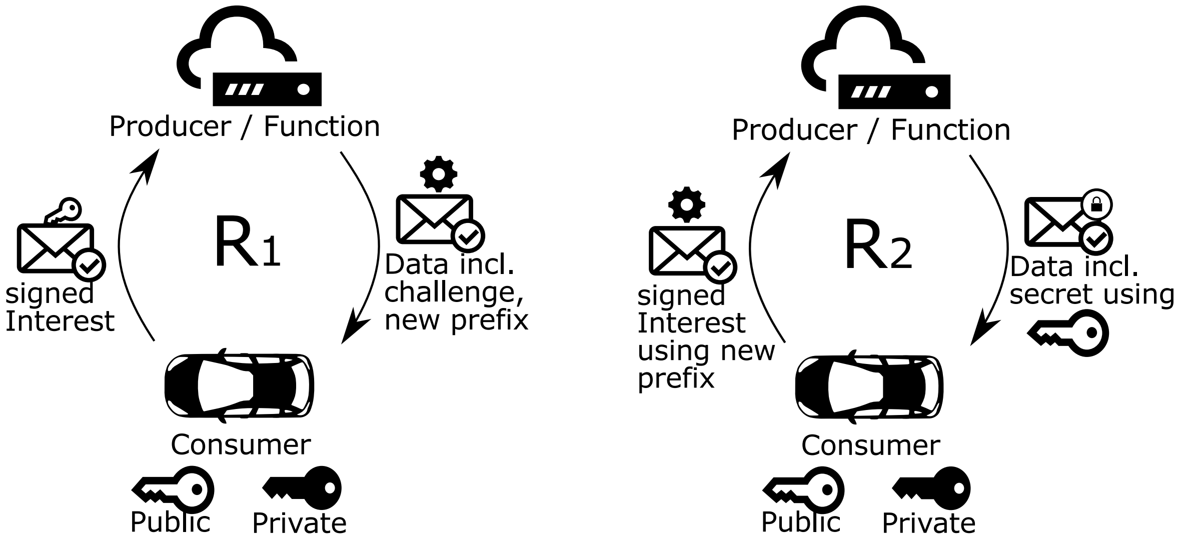
When applying for access to Information Objects, the decryption information SK has to be shared between the consumer and the creator. In EnCIRCLE, information sharing is based on the proposed CCNx-KE [306] scheme and separated into at least two rounds:

R_1 establishing an authentic, confidential session.

R_2 sharing of key material to decrypt a cipher-text.

R_3 (optional) sharing additional key material of a consumer for future cipher-texts.

Establishing a Session (R_1): Before sharing the actual key material, a session need to be established between a consumer and a producer. Figure 8.5a illustrates the session initiation procedure. A consumer creates a temporary public/private session key pair (e.g., using ECC). When applying for the key material to access an Information Objects, the consumer sends out a signed `INTEREST` packet to the producer providing the temporary PK. In order to reach the producer, the *MustBeFresh* flag can be used to bypass cached copies [253] in the first place.



(a) Round1 of the on-boarding procedure: A consumer sends out a signed INTEREST packet and receives a DATA packet including a source challenge and a temporary local prefix.

(b) Round2 of the on-boarding procedure: The consumer uses the prefix and the source challenge to query for the key material provided by the producer or in-network function encrypting the desired Information Object.

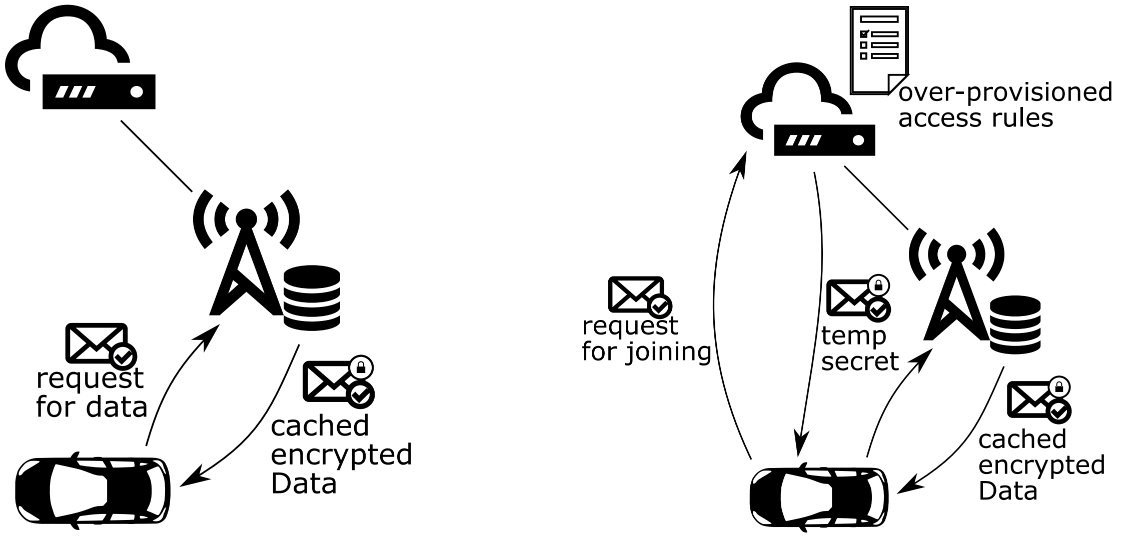
Figure 8.5: Illustration of the *on-boarding* procedure in EnCIRCLE. Two rounds are at least required to on-board a consumer in order to provide access to an encrypted Information Object.

A producer directly answers back with a corresponding DATA packet including a source challenge and a temporary local prefix. The source challenge is used to identify a consumer for further incoming INTEREST packets. The temporary local prefix ensures that subsequent INTEREST packets are routed towards the same producer, and not answered by other instances or cache nodes. This completes the first round.

Sharing of Key Material (R_2): After a session has been established, the consumer requests for the decryption key material. It sends out a new signed INTEREST packet including the result of the source challenge and the temporary local prefix. In the meanwhile, the producer prepares the SK to be shared with the consumer using the key generation algorithm introduced in Section 8.2.2, and send it back to the consumer. Finally, the received information are used by the consumer to decrypt the cipher-text, and the temporary keys are destroyed by both the consumer and producer.

Sharing Additional Key Material (R_3): If required, the on-boarding procedure can be extended by an additional round to grant access for future Information Objects of the producer or a in-network function. As part of this round, consumers' are able to provide unique attributes in order to extend the amount of public parameters $A \subseteq \{P_1, P_2, ...P_n\}$. This can be used to generate an enhanced version of the access policy structure AS, for example to provide access for a group of consumers.

Depending on the desired level of security to be achieved during the exchange of key material, additional rounds can be introduced to make it harder for malicious entities in the network.



(a) A consumer queries the network for data, however received an encrypted Information Object which is not usable by the consumer.

(b) The consumer can consult the creator of the packet to request for one-time access temporarily granted by the creator.

Figure 8.6: Illustration of the *joining* phase of EnCIRCLE: A consumer received an already encrypted Information Object from a cache nearby, however is not able to decrypt the information due to missing key material. He can request for one-time access to speed up the data processing.

Exchange of Key Materials for Already Encrypted Information Objects: In case a consumer received an already encrypted Information Object from a nearby local cache, it can apply for one-time access as part of the *joining* phase in EnCIRCLE (cf. Figure 8.6a). Similar to the exchange procedure presented as part of the *on-boarding* phase, a producer over-provisioned the number of public parameters PP in the access structure AS (cf. Figure 8.6b). These *reserved attributes* are used to provide one-time access for consumers to overcome the problem of cached and already encrypted Information Objects. Finally, the consumer has to be officially on-boarded by following the procedure of the on-boarding phase.

Revocation of access rights

Some of the most challenging tasks in information security is the enforcement of authorization policies, especially, performing policy updates in a network (e.g., user access revocation) [307]. However, revocation of access privileges is required to protect information which is still valid.

Regarding the electronic horizon use case, some data is valid only for a short period (e.g., available parking spots nearby) and requires a re-encryption of data periodically using the current access policies at the time the data is generated. In such use case, the access policy has to be updated by removing attributes from unprivileged consumers', so that only the eligible consumer attributes remain in the access policies. In such update scenarios, security degradation regarding backward and forward secrecy is a challenge. In the context of ABE, backward secrecy describes a mechanism to prevent the access to already encrypted information by a new consumer, since the consumer's key material has not been considered during the encryption procedure. Forward secrecy describes a mechanism to prevent the access to information by consumers' whose authorization has been withdrawn [308].

In the past decade, time restricted policies have been introduced in the context of ABE to be included to the policy structure, so continuous access is only possible if consumers renew their key material from time to time (cf. [307]). However, the intrinsic in-network caching capabilities of ICNs complicates the revocation of access rights. This is due to the fact that `DATA` packets can be cached multiple times at any node in the network, without the knowledge of the creator.

Besides providing time restricted information as part of the policy, `DATA` packets need to be encrypted again by the producer and replaced within the caches. However, pushing new `DATA` to the network is a challenge in ICNs due to the fact that most of the ICNs architectures follow a *pull-based* communication approach (e.g., CCN and NDN).

The concept within this manuscript proposes the following option to deal with cached content: Before a `DATA` packet is sent out, a creator (e.g., producer, in-network function, etc.) modifies the *FreshnessPeriod* (cf. NDN packet specification [253]) value of the packet aligned with the policy present in the access structure. After the freshness value expires, the `DATA` packet will not be delivered by the caches within the network.

8.2.3 Evaluation & Results

In order to evaluate EnCIRCLE, an analysis of the security properties is performed to evaluate how and under what conditions can an adversary break such a protected producer/in-network function. The section is separated into two parts. The first part introduces the attacker model and security threats used for the evaluation, while the second part describes the results of the concept evaluation of EnCIRCLE.

Attacker Model

According to the challenges and requirements presented in Section 8.1.1, the security objectives to be considered in this manuscript include:

- **Authentication**: consuming entities should be able to authenticate the sender of data.
- **Verification and Integrity**: consuming entities should be able to verify if the received data is valid and not modified during delivery.
- **Confidentiality**: only eligible entities should be able to access and process data.
- **Privacy**: the privacy of entities should be ensured against unauthorized observers.

The attacker model used in this manuscript follows the well-established Dolev-Yao attacker model in an extended version presented by Raya et al. [309]. Instead of considering only an active attacker as presented by Dolev-Yao [310], the model is classified in three dimensions [309]:

- **Insider vs. Outsider**: An successful authenticated attacker is eligible to access the system, while the outside attacker gets access to the system as an intruder.
- **Malicious vs. Rational**: An attack which is interested in causing maximum damage, while the other attacker pursues personal goals (e.g., steal information).
- **Active vs. Passive**: An attacker actively engages with network traffic (e.g., creating packets), or eavesdrops the traffic passively.

Based on the use cases descriptions (cf. Section 1.2), the assets to be protected include the *producer of data* (including in-network functions), the *consumer of data*, the *funtion execution node* and the *data* itself. The aim of EnCIRCLE is to secure the access to data and prevent data to be compromised against the following potential attacks [311]:

- **Impersonating**: An attacker pretends to be an authenticated entity.
- **Masquerading**: An attacker uses the identity information of another entity (reflection attack).
- **Eavesdropping**: An attacker eavesdrops the communication of a network entity or the communication medium in order to inject, manipulate or drop messages.
- **Jamming & Spoofing**: An attacker blocks the access communication medium or manipulates messages by injecting fake information.
- **Man-in-the-Middle**: An attacker is positioned between two communication participants, able to monitor the information exchange between both.
- **(Distributed) Denial-of-Service**: An attacker floods the network with generated messages in order to block eligible entities to exchange information.

Threats

Using the presented attack model as well as the potential attacks, the resulting threats are derived using the list of assets to be protected by the EnCIRCLE framework. The following threats are considered for evaluation:

- **Access Information**: An attacker is able to access a specific Information Object to public.
- **Block Information**: An attacker is able to block the dissemination of specific Information Objects in the network.
- **Escalate Privileges**: In case an attacker has escalated privileges, he can act as any entity in the network including consumer, producer, and execution node.
- **Block Producer/Execution Node**: An attacker is able to block any information exchange between a consumer and the corresponding communication participants.
- **Manipulate / Spoof Information**: An attacker is able to manipulate packets as well as its content. This also includes information required by execution nodes in order to perform a computation.

Evaluation of the EnCIRCLE Framework:

Based on all the introductions about the attack model, assets, and threats, attack trees are created in order to evaluate EnCIRCLE against each threat.

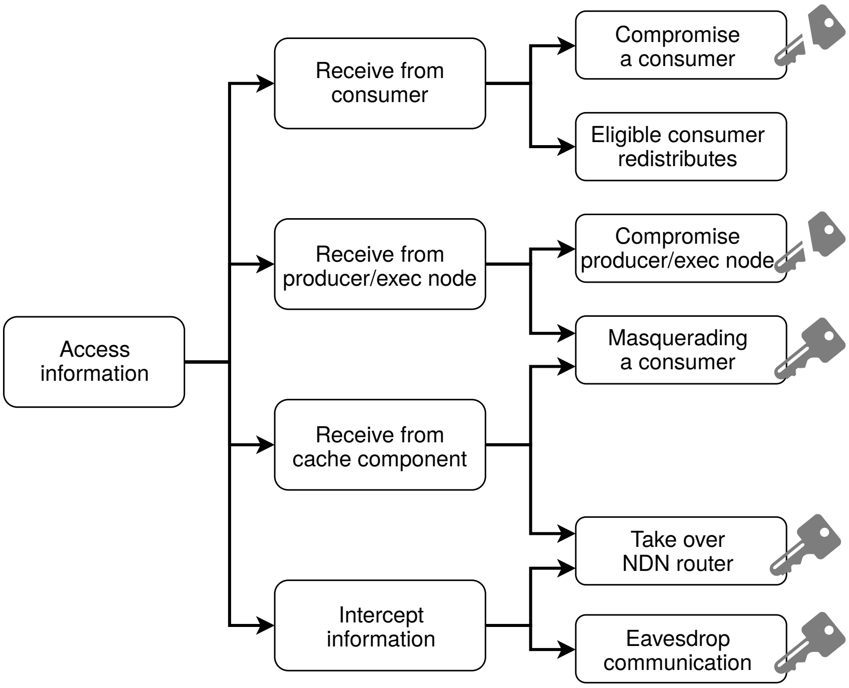


Figure 8.7: Illustration of the attack tree to get access to a specific information object.

Access Information Threat: Figure 8.7 shows an attack tree for getting access to an Information Object in an ICN which is running the EnCIRCLE framework. An outside attacker can just passively listen to the channel used by the underlying NDN transport protocol. Additionally, in a public deployed NDN, it is usually quite easy to get access to data as you can just act as a legitimate NDN router to forward any queries to the network. However, in all cases where the adversary does not have access to key material, the gathered encrypted data is useless to the attacker. This is also the case for an inside attacker. Masquerading as a legitimate consumer does not work, without having the right key material. Moreover, the join procedure of EnCIRCLE for acquiring keys requires that a consumer needs to authenticate against the producer or an execution node.

All the mechanisms discussed above *will* lead to an ineligible access to data, if the attacker could gain access by compromising the consumer, the producer, or an execution node directly and thus leaking their key material. However, if such a breach is detected, future data can be encrypted again in such a way that the compromised party can not decrypt it anymore (cf. Section 8.2.2). The same is true, if a malicious consumer redistributes decrypted data to third parties.

Block Information Threat: Figure 8.8 shows an attack tree for blocking a specific Information Object to be distributed across the network. A simple attack is described by blocking the communication medium. An outside attacker can jam the wireless channel, so both consumer and creator are not able to exchange any information. Comprehensive attacks can be performed by an insider, for example, by selectively block an entity. Producer or execution nodes can be blocked by starting a DoS attack (e.g., flood the network with INTEREST packets), or simply drop queries and responses if an attacker is acting as a legitimate NDN forwarding node.

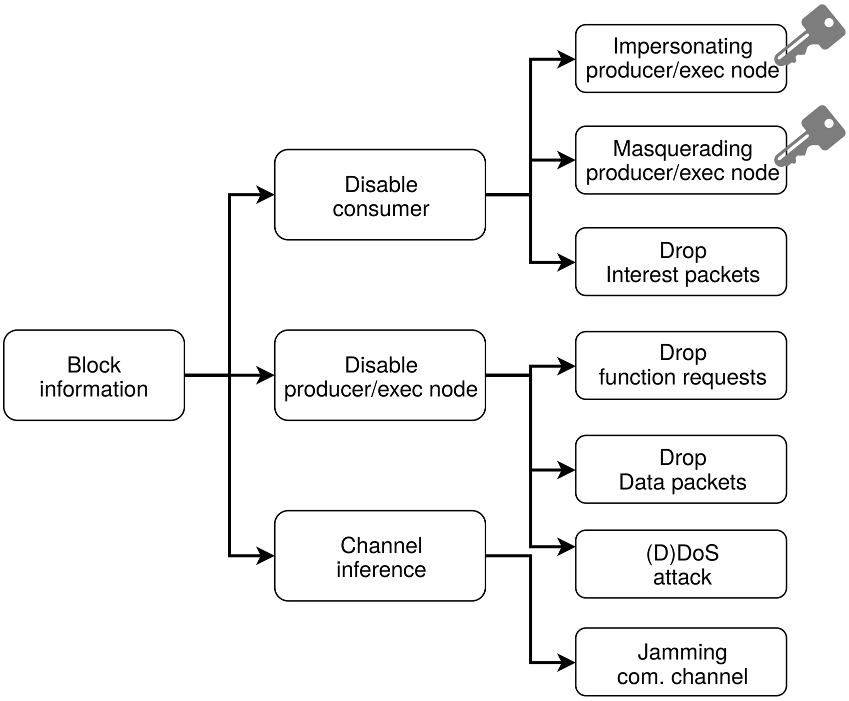


Figure 8.8: Illustration of the attack tree to block dissemination of specific information object.

However, masquerading or impersonating a producer/execution node will not work, due to the fact that the adversary does not have access to key material, to sign a packet correctly, as well as to have the right temporary key material during the on-boarding and joining phase.

Escalate Privileges Threat: Figure 8.9 illustrates an attack tree for getting access to any key material in the network. An outside attack can try to escalate privileges by masquerading either a consumer or a creator of information, for example replay monitored packets to the communication partner. Since in NDN at least every creator has to sign a packet, a consuming node can verify the signature (cf. Section 8.2.2). As part of the NDN architecture, an `INTEREST` packet can also be signed by the consumer [312], which makes hard to perform any masquerading attack. However, if an insider is able to compromise a creator or consumer, the attacker is able to impersonate this entity.

Block Producer/Execution Node Threat: The attack tree to block a producer or execution node is shown in Figure 8.10. Similar to the block a specific Information Object, an outside attacker can try to jam the wireless communication channel. As a result, the creator is not able to receive or to answer to any queries. A similar result is achieved, if an inside attacker acts as a legitimate NDN forwarding node. In this case, the attacker is able to drop any packet from and to the creator. Masquerading a creator is hindered by the standard mechanisms of NDN, since `DATA` packets have to be signed by a authenticated creator and misbehavior can be easily detected. However, all the security mechanisms become ineffective, if an insider is able to gain access by compromising the creator (e.g., stop any information exchange immediately). Henceforth, a consumer or an intermediate node is not able to detect the impersonated attacker.

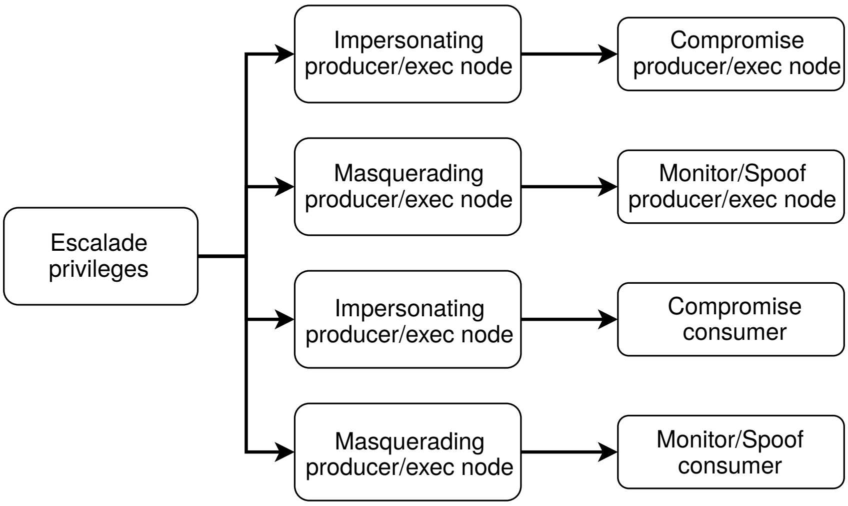

Escalade privileges
Impersonating producer/exec node
Compromise producer/exec node
Masquerading producer/exec node
Monitor/Spoof producer/exec node
Impersonating producer/exec node
Compromise consumer
Masquerading producer/exec node
Monitor/Spoof consumer

Figure 8.9: Illustration of the attack tree to escalate privileges.

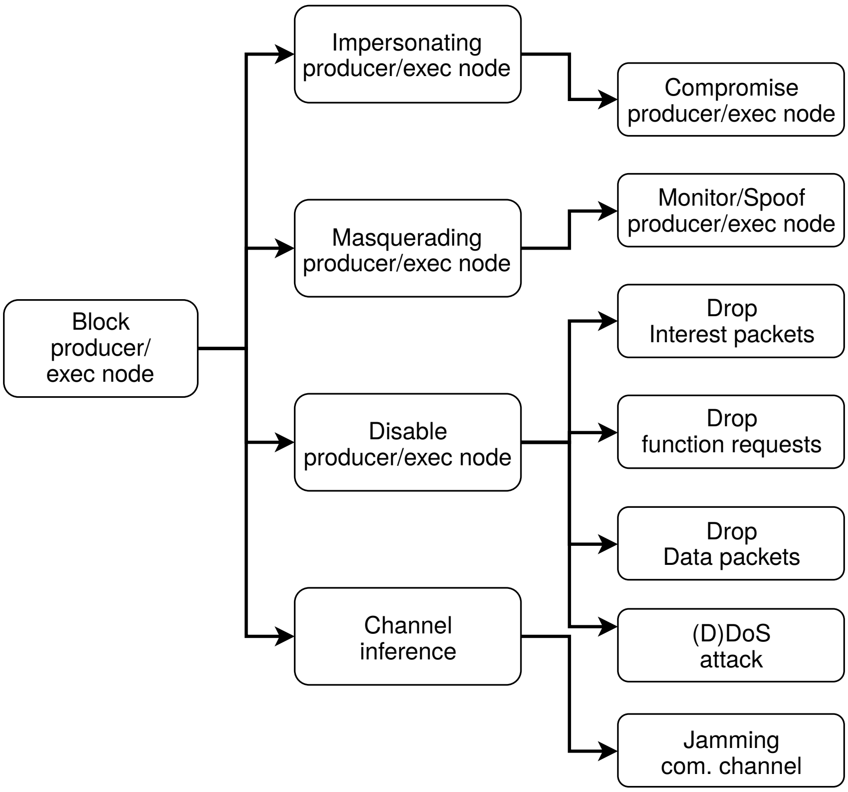

Block producer/ exec node
Impersonating producer/exec node
Compromise producer/exec node
Masquerading producer/exec node
Monitor/Spoof producer/exec node
Disable producer/exec node
Drop Interest packets
Drop function requests
Drop Data packets
(D)DoS attack
Channel inference
Jamming com. channel

Figure 8.10: Illustration of the attack tree to block a Producer/Execution node from receiving or sending any information object.

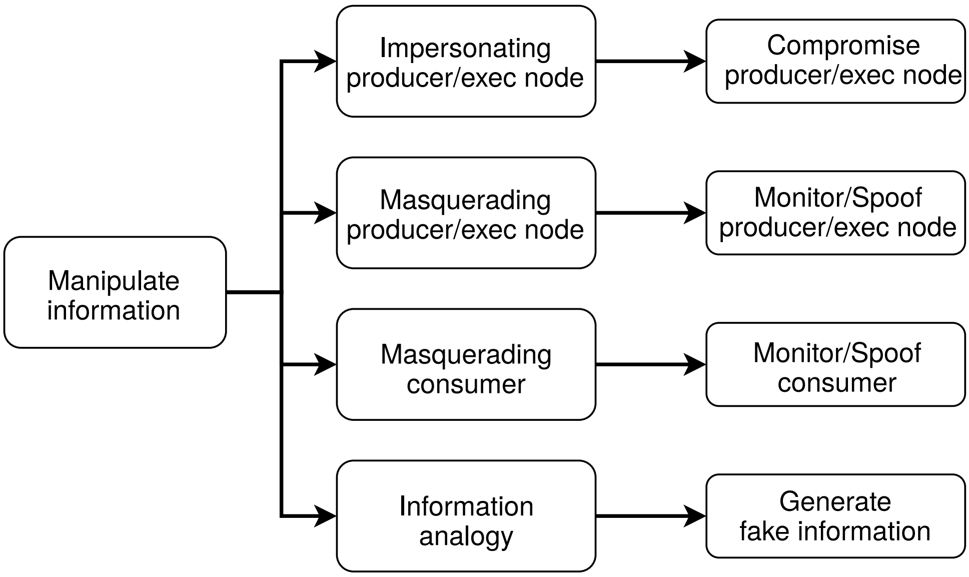

Manipulate information
Impersonating producer/exec node
Compromise producer/exec node
Masquerading producer/exec node
Monitor/Spoof producer/exec node
Masquerading consumer
Monitor/Spoof consumer
Information analogy
Generate fake information

Figure 8.11: Illustration of the attack tree to manipulate any information objects.

Manipulate/Spoof Information Threat: The attack tree to manipulate or spoof a packet is shown in Figure 8.11. When masquerading, injecting or generating fake data, it will only be accepted by a consumer when valid keys are used due to mandatory signatures. Without the keys an attacker can neither fabricate nor modify the data in the system. Thus, as long as the key material is secure, EnCIRCLE can protect the data.

The evaluation has shown that most of the attacks on the participant and data level are covered by the EnCIRCLE framework. Attacks on the network system itself such as (D)DoS or wormhole attacks (e.g., by dropping packets as part of a forwarding node), are not covered by the framework and therefore still possible. In the literature, there exist mechanisms, for example to detect flooding attacks (e.g., [313]) or to improve the resilience of the system by detecting (D)DoS (e.g., [314]).

Another sensitive point in the security framework of EnCIRCLE is the authentication of a consumer during the join process. If producers or execution nodes have restricted access, they would want to authenticate a consumer before issuing key material. Since there is already work done in this field, it is intentionally not covered in EnCIRCLE (cf. [312]).

Also, not surprisingly, similar to common Internet services, operational security in the backend is of importance. The operating system and software of producer's and execution nodes should be hardened and protected against attacks capable of taking over the system or leaking sensitive information. While the security capabilities of EnCIRCLE tackle the requirements regarding the access to data, there are still open security issues, in particular in the context of computation-centric vehicular networks. These issues are introduced in the following section.

8.3 Open Security Issues in Computation-Centric Vehicular Environments

The benefits of being able to invoke in-network computations (e.g., offloading a computation from the car to the infrastructure) poses opportunities for future connected vehicle networks (cf. Section 7). While the previously introduced EnCIRCLE framework addresses security challenges in ICN-based connected vehicles such as authentication, confidentiality and access control, there are specific challenges introduced by the concepts of *named functions* (e.g., [65, 66], cf. Section 3.1.4):

- **Secure Computation Input Submission**: A secure exchange of input parameters is required, if a in-network function requires such data from a requestor.
- **Secure Computation Invocation**: It must be ensured, that the right function is invoked, executed and not able to leak any private information.
- **Verification of Computation Results**: A requestor must be able to verify the correctness of computation results in order to detect fake data or malicious acting function provider.
- **Interoperability of Security Mechanisms**: Due to the diversity and evolution of device capabilities (e.g., different manufacturers, different hardware, etc.), connected vehicle environments are expected to be characterized by an existing heterogeneity of security solutions. These solutions must be aligned and periodically adjusted with the solutions in the infrastructure network.

In the following sections, the presented challenges are introduced, discussed in details and open issues identified. For each of the issues, mitigation approaches and open research directions are highlighted (cf. Table 8.1).

8.3.1 Secure Computation Input Submission

The *pull-based* communication model of `INTEREST`-based ICNs (e.g., NDN) challenges the submission of additional input parameters. According to this model, an execution node has to query the requestor for input parameters, securely transferred through the network via a encrypted `DATA` packet. Especially in connected vehicle environments, this procedure is challenging due to the high degree of mobility of the network participants. It requires that an execution node is able to reach the right requestor (e.g., using impracticable global unique prefixes for each requestor). An execution node might not be able to reach the vehicle, while it is moving from one AP to another. Furthermore, this problem is also valid if a in-network function has been moved from one execution node to another one, for example for efficient utilization of network resources. In both cases, creating or manipulating forwarding entries in the FIB provides a solution space. However, it opens the door for an attacker to run flooding attacks.

Another option to transfer input parameters can be achieved by adding them to the request. On the one hand, providing such information as part of the visible name may causes a leak of privacy. On the other hand, transferring parameters within the payload of an `INTEREST` packet opens the door for an objector to run DoS attacks. An attacker can occupy execution nodes by flooding the network by adding large amount of fake data to Information Objects.

8.3.2 Secure Computation Invocation

There are several security aspects regarding the secure invocation of an in-network function or querying for a computation result. First, the function needs to be verified before being executed by the execution node. This can be achieved by verifying the signature created by the function provider. Second, the correct behavior of the function has to be verified. This is more challenging, since not every function code is expressed in the network as open source. One potential solution is described by trusted execution environments. They allow applications to be executed in private memory enclaves (e.g., [315]). Functions can be tested beforehand within such private environments and published publicly after the verification. After a function has been verified successfully, it can attest to a requestor that it is trustworthy, for example by using a *remote attestation protocol*. However, such environments and protocols have not been conducted yet in the context of ICN-based computation networks, and therefore requires additional hardware and communication procedures in the network.

8.3.3 Result Verification

Due to the fact that it is mandatory to sign `DATA` packets in NDN, the verification of static content is achieved by checking the signature of a received packet. However, in the context of in-network computations, content can be modified as a result of in-network computations. Such processing invalidates the signature of a producer. Furthermore, the function is not able to sign the computation result on behalf of the originator, since it does not keep the private key material.

This also applies to function providers. As in-network functions are expected to be executed on remote hardware, they can be loaded and executed elsewhere in the network. Computation results should be signed by the function itself, however, it requires the key material in order to sign the result. Trusted execution environments can be used to store private key material, used by the function in order to sign the computation result and thus be verified by any consumer in the network.

8.3.4 Interoperability

The variety of existing security mechanisms as well as the variety of hard- and software components demand for conventions to coordinate and to agree to procedures for decryption, signature, and provenance verification. The transition to move computations from centralized, *walled-garden* cloud infrastructures towards federated untrusted nodes at the edge of the network challenges security interoperability. While first work is presented in the context of named functions to provide schematized trust and access control mechanisms (e.g., [316, 317]), they show that fully automated, flexible, and secure mechanisms are beneficial. However, the approaches are not presenting comprehensive solutions and therefore requires further effort in order to be applied to a distributed computing environment.

The introduction of in-network computation as part of a federation of untrusted nodes at the edge of the network raises many new security challenges. Mechanisms are required in order to construct a trusted system consisting of un-trusted components. The identified challenges and presented solution space need to be addressed in future efforts to establish such a trusted system for in-network function execution.

Table 8.1: Summary of identified problems with potential solutions and open issues

Issue	Potential Solution	Open Issues
Input Submission	FIB modification	Protect against flooding and DoS
Invocation	Remote Attestation Protocols	No attestation protocols for ICN, hardware vendor trust issue
Result Verification	Memory enclaves for private key material	No trust exec. env. for ICN, hardware vendor trust issues
Interoperability	Schematized security policies, explicit procedures	General-purpose language to express consumption logic

8.4 Summary

The work presented in this chapter addresses security related problems in ICN-based connected vehicle systems, caused by storing Information Objects multiple times within cache nodes in the network. Due to the distributed and decentralized fashion of NDN, the management of cached objects is challenging. For example, this includes authentication procedures, mechanisms to ensure confidentiality, integrity and the access management to information in a high dynamic environment.

The encryption-based access control framework EnCIRCLE contributes to this thesis by providing a solution for each of the introduced challenges. As it does not rely on a continuously available authority, it can be used in ICNs with high mobility and intermittent backend access. Access policies can be enforced, even for time limited access patterns in a distributed system. While the design of the framework is based on top of NDN, the approach should be applicable to other ICN architectures as well. Based on the introduction of attacker models, a security analysis has shown under which conditions the framework can protect Information Objects even if cached multiple times in the network.

While the security capabilities of EnCIRCLE have shown that the access to Information Objects can be protected, there still exist security related issues when a computation-centric network is considered. These issues have been presented and discussed in detail such as *secure submission of input parameters*, or the *verification of computation results*. The results of this discussion have shown open research challenges in the context of a secure distribution and execution of named functions at the edge of the network. Future work has to address the open research directions in order to improve the security and privacy mechanisms of computation-centric edge environments.

9 Conclusion

Knowledge has a beginning but no end.

Geeta Iyengar

Connected and smart vehicles will dominate the scene on the road, having a disruptive impact on future mobility solutions. One important aspect is the development of information and communication technologies, paving the way to interconnect different systems and devices together, including vehicular systems. However, such interconnected mobile systems challenge today's host-centric networks on multiple layers. For example, this includes the maintenance of host addresses, establishing end-to-end communication channels while participants move from one network access point to another, or the efficient dissemination of popular information to name just a few.

In the past years, the loosely coupled communication model of the Information-Centric Networking paradigm has attracted researchers in the field of vehicular networking (e.g., [225, 9, 10]). Especially the intrinsic in-network caching capabilities of ICNs are promising to balance and reduce traffic by placing valuable data nearby the consumers (e.g., [249, 136]) or to carry the information in areas uncovered by any network infrastructure (e.g., [206, 284]). Such placement is achieved using caching strategies such as data caching during the delivery reactively. However, reactive caching strategies in mobile networks also poses some challenges. For example, a mobile consumer may have lost the connection to an access point and hence is not able to receive the data. Proactive caching strategies promise to improve the performance of the network by placing the right data at the right elements in-time.

The identification of data worth to be placed within caches proactively describes a non-trivial task. Valuable information may be of interest to a group of vehicle consumers, but also for an individual one. On the one hand, this requires efficient mechanisms to monitor and detect automotive data traffic flows in order to select the right data worth to be cached proactively. On the other hand, the placement of data at nodes multiple times in connected vehicle networks raise questions regarding security and privacy.

This combined field of research has not been conducted yet by the research community, leading to three main research questions which are addressed in this thesis (cf. Section 1.4.1). The work present in this thesis envisions a proactive caching framework within information-centric/computation-centric connected vehicle environments in which active placement strategies play in important element. Based on the analysis of automotive services, such caching strategies are presented. The evaluation results show performance improvements of the network by increasing the availability of data, while decreasing delivery times, and thus, demonstrate the benefits of placing data pro-actively in the network.

The following sections discuss the level of achievements regarding the research questions and present the results. Future work and open research directions are outlined. Finally, the thesis closes with some final remarks.

9.1 Achievements and Comparison of the Research Questions

Q1: What are the benefits of placing automotive data proactively in the network and closer to consumers? From a network perspective, automotive data traffic can be categorized in terms of their (i) popularity, (ii) size, and duration of validity. In order to identify the benefits of active placement of automotive data, an analysis of automotive traffic classes has been made using real world mobile applications from the automotive domain. Data categories have been derived from the analysis, worth to be stored proactively in the network, namely *transient popular data*, as well as *transient/static personalized data*, both independent of the size (cf. Section 5.1.1). This provides an answers to the question of the first research objective: Q1.1 *What kind of automotive data is beneficial to be cached proactively in the network?*

In order to understand the shortcomings of today's host-centric network as well as of plain deployed NDN in the context of connected vehicles, an analysis of the communication behavior of both paradigms has been performed. Based on the real world automotive infrastructure deployment in Austria, extensive network simulations have been performed including varying traffic situations (cf. Section 5.6). From a host-centric networking perspective, the results presented in Section 1.3 have shown shortcomings in the network performance (e.g., request to response ratio) by always establishing an end-to-end connection between the vehicle and the producer of information. Furthermore, this also results in inefficient data delivery by transferring the same data multiple times through the network. The results of the analysis of reactive placement strategies in NDN presented in Section 5.2 have shown performance improvements by storing *static, popular* data closer to consumers. However, such strategies are not efficient for the other data traffic categories such as *transient, popular* or *transient, personalized* data. These results answer the question: Q1.2 *Why do current networks (e.g. IP-based, ICN-based) fall short?*

Another promising approach to overcome intermittent connectivity caused by sparse infrastructure deployments is the introduction of vehicles as "storage on wheels". Since ICN introduces intrinsic in-network caching capabilities, using mobile vehicular caches to carry information helps to overcome the challenge of sparse network deployments. The work presented in Section 6 introduces models from stochastic geometry to determine the potential of mobile vehicular caches. The models have been evaluated using the simulation environment presented in Section 5.6. The results have shown that the availability of data can be increased by using the cache capabilities of mobile nodes to carry data passively through the network. These results provide an answer to the question: Q1.3 *Can the availability of automotive data be increased when vehicles carry data passively through the network?*

For this reason, the first research objective of this thesis (Q_1) is fully achieved and can be summarized as follows: The active placement of *transient, popular/personalized* data increases its availability (e.g., in sparse network deployments), while reducing the delivery times, and thus improves the degree of quality in data delivery. This statement is valid for data which is part of the presented categories worth to be stored proactively in an automotive ICN, regardless of the transfer of plain Information Objects or in-network function results.

Q2: How to place automotive data proactively in data-oriented connected vehicle networks? As there is an evident benefit of placing content proactively in ICN-based connected vehicle networks, the second research objective is related to how this can be achieved. First, an extensive comparison of existing ICN architectures was performed in order to identify the most applicable ICN approach for connected vehicle environments. The result has shown that the decentralized fashion of *Interest-based* ICN architectures such as CCN or NDN are most appropriate for connected vehicular networks (cf. Section 4.1). Furthermore, a taxonomy of network caching has been presented in order to gain a better understanding about network caching architectures. As a result, the taxonomy has been used to identify the potential cache components and cache strategies useful for active content placement (cf. Section 3.2). This includes infrastructure nodes at a distance of 1-2 hops away from the consumers (e.g., RSUs, geo-location specific nodes as well as other vehicles heading in the same/opposite direction of the consumer). The results of this comparison as well as the discussion of network caching mechanisms answers the first part of the second research question: *Q*2.1 *What kind of network architectures are useful for proactive data placement?* In order to store the right content at the right cache node, mechanisms are required to identify which automotive data is needed.

This objective has been addressed for research question: *Q*2.2 *How to identify where automotive data is needed?* Based on the analysis of automotive data traffic classes, an adaptive framework for active content placement has been proposed in this thesis (cf. Section 5.2). As part of this framework, novel proactive caching strategies have been introduced, each addressing one of the data traffic categories. In each of the presented strategies, the identification of data items is described as an essential part. Since data is addressed directly in ICN, such networking paradigm simplifies the access/monitoring of data, in order to identify and extract valuable information. In this thesis, centralized (cf. PeRCeIVE in Section 5.3) as well as decentralized identification approaches (e.g., ADePt in Section 5.4) have been presented, either triggered by a consumer itself (e.g., by providing consumer specific information such as position and velocity), or by monitoring the traffic flows individually at edge nodes (e.g., such as RSUs). The approaches are presented and discussed in detail w.r.t. the identification mechanisms. The results of the options to identify data items worth to be cached proactively provide an answer to research question *Q*2.2.

The last element to be addressed are protocol mechanisms to actually place data items into cache components. It demands for a *push* like protocol design. However, most of the ICN architectures follows a *pull-based* communication model (including NDN). In this thesis, a detailed discussion about the options to place Information Objects into caches is presented in Section 5.3.2 as well as in Section 6. Protocol implementations in both simulation and a real world prototype have shown the applicability of the caching strategies. While the presented solutions keep the flow balancing principles of NDN intact, additional traffic overhead, caused due to additional round-trips and hence increased loading delay, are the results of the implementation. To this extent, the presented cache strategies provide protocol enhancements to load automotive data into network cache components. This answers the research question: *Q*2.3 *How to actually place automotive data within network component caches?*

For this reason, the second research objective of this thesis (Q_2) is fully achieved and can be summarized as follows: Based on the analysis of automotive applications, data categories have been derived and used to develop novel proactive cache strategies and protocol designs for each of these categories. Table 9.1 illustrate the achievements of the novel proactive placement strategies with respect to the automotive data traffic classes. The presented strategies have been evaluated against state-of-the-art mechanisms using simulation and implemented as part of a real world prototype in small scale. In order to optimize the network performance, further investigations regarding efficient models to extract and learn from transferred data are required.

Table 9.1: The novel proactive placement strategies presented in this thesis with respect to the automotive data traffic classes.

Data	popular	personalized
transient (small)	✓(ADePt)	✓(`PeRCeIVE`)
transient (large)	✓(Predictive Prefeching)	✓(`PeRCeIVE`)
static (small)	*reactive*	-
static (large)	*reactive*	-

Q3: What are *security* related implications when caching data proactively in data-oriented connected vehicle networks? Replicating and storing content and computation results multiple times in the network impacts security and privacy, which are addressed in this thesis as part of the last research objective *Q*3. Besides the content itself to be protected, this also includes the network entities and their personal goods, as well as in-network function and execution nodes. Based on the use cases presented in Section 1.2, actors and assets to be protected have been analyzed. By considering these actors and assets, security related challenges have been derived such as authentication, integrity, or access control. This analysis provides an answer to research question: *Q*3.1 *What kind of security related challenges exist when introducing named data/functions in connected vehicle environments?*

As one focus area, access control mechanisms which follows the decentralized networking model of NDN are depicted as one specific topic in this thesis. Based on the derived security challenges, the EnCIRCLE access control framework is created based on the principles of the CP-ABE encryption mechanism (cf. Section 8.2). The framework provides building blocks for each of the introduced challenges. In order to evaluate the security properties of EnCIRCLE, a security analysis has been performed. The results have shown that the framework can protect Information Objects against simple attacks, as long as the key material is not leaked. Based on the results of the analysis, this thesis answers research question: *Q*3.2 *How to restrict the access to automotive data which is placed proactively in the network only for eligible users?*

Since a creator of Information Objects in NDN/NFN (e.g., a producer or a in-network function) has no handle to the cached replicas in the network, controlling the access to these replicas is challenging. This includes the on- and off-boarding of consumers which received an already encrypted Information Objects. As part of the EnCIRCLE concept, this issue has been partly solved in the research question: *Q*3.3 *How can a (new) consumer access already cached automotive data in a non-trusted distributed environment while being highly mobile?* While the mechanism is able to on-board consumers using pre-provisioned key material (one-time access), revoking access privileges using time-based features defines a first solution. There is still potential for further improvements. There are still open challenges in secure distribution of content in information-centric connected vehicle environments, especially regarding computation-centric networks. These challenges have been introduced and discussed. As a result, open research directions have been presented in the context of secure distribution and execution of named functions in edge environments (cf. Section 8.3).

For this reason, the third research objective of this thesis (Q_3) is achieved and can be summarized as follows: Based on the analysis of the automotive use cases, security challenges have been derived for both transfer of plain content as well as in-network function results.

By using these challenges, an access control framework – EnCIRCLE – has been presented. Open research challenges in the context of secure distribution and execution of named functions in edge environments are outlined.

9.2 Discussions and Outlook

The work presented in this thesis forms the basis for directions of future work with respect to ICN in general as well as in the context of connected vehicle environments. The following section discusses the potential impact of this thesis and introduces some pointers for future work and their potential opportunities.

9.2.1 Traffic Monitoring and Analysis to Optimize Network Performance

The paradigm shift from addressing physical nodes to access content towards addressing content directly in ICN has resulted in transferring functionality from the higher communication layers directly to the network layer. The introduction of naming schemes has started a merging process of network data- and control-planes to bring both closer together. Such process opens up new opportunities to improve the performance of ICNs and increase the quality level of services. For example, every time an `INTEREST` is processed in the network, it offers access to information as well as leaving information as part of the forwarding state in the network. Another example are the opportunities to encode information as part of naming schemes. Since these names are used to route queries and content through the network, it is visible to all forwarding nodes.

Based on this facts, a network node or a set of nodes can monitor the network traffic. This enables the nodes to learn about content requested and transferred through the network. The mechanisms presented as part of the ADePt (cf. Section 5.4) and the predictive analytic approach (cf. Section 5.5) are only first directions to use the "floating" information in an ICN. Using such information allows for novel network features such as efficient congestion control and proactive network balancing, predictive self-healing mechanisms (e.g., in case of link/node failures), caching strategies, etc. just to name a few fields of application. Similar directions are currently under discussion within the ICN research community (cf. [318]).

9.2.2 Semantic Technologies to Improve Proactive Caching Decision Making

Existing ICN architectures offer mechanisms to transfer static content through the network (e.g., CCN or NDN). However, there is still research required to bring different content into context and therefore evolve towards a network of information.

In the past decades, research activities are concerned about information or knowledge networks, for example by introducing techniques from the Web of Data and the Semantic Web of Things (SWoT) domain. Such techniques allow enriched content with additional information such as the context used, the scope, validity, etc. First efforts of introducing such mechanisms in an ICN have been already done in the community. For example, the Semantic Object Discovery (SODi) framework [319] introduces a mechanism to enrich `DATA` packet using semantic annotations. While the scope of the framework is to simplify the access to information in an ICN, semantic annotations can be used for proactive caching decisions in order to actively load Information Objects to areas in which consumers will request the object soon. Furthermore, description mechanisms can be used to describe named function, for example, to simplify discovery of deployed functions, i.e., to describe their scope used, context, input parameters as well as their computation result.

9.2.3 Computation-Centric Mobile Networking

Named Function Networking and NFaaS are first efforts towards ubiquitous and pervasive networks. In the first place, in-network computations serve the needs of consumers' in order to provide access to dynamic content. Continuing the idea of a network capable of running any function at any node offers a high degree of flexibility as well as to optimize the network performance.

As part of this thesis, resolution strategies for in-network functions as well as a first vehicular named function prototype implementation has been presented. The work provides a first step towards the effort of introducing in-network function in mobile scenarios. An example use case of future in-network functions from the automotive domain includes mobile/traveling functions. In order to meet tight latency requirements of automated driving, a mobile function can travel along with the vehicle from access point to access point. However, there are still a lot of open research questions to be addressed in order to bring such a vision to reality. While the presented scenario sounds futuristic, in-network functions may have a huge impact on mobile applications to offload, share, distribute and store computation results everywhere in the network, not just in the automotive IoT, but also in other IoT domains.

9.3 Final Remarks

This thesis has contributed significantly to the topic of proactive content placement in order to increase the availability of data in connected vehicle environments using ICN as an underlying network technology. Novel caching strategies have been introduced and evaluated via extensive simulations based on a real world network deployment structure.

The development of a real world vehicular ICN prototype, including the development of a network stack for computation-centric vehicular networks, has shown the strengths of the ICN paradigm in mobile network scenarios compared to the traditional host-centric model. Furthermore, the development of the prototype has contributed significantly to investigate the capabilities of the NDN and NFN approaches under real world conditions.

Finally, the work has spearheaded the topic of Named Function Networking in the domain of IoT, especially the automotive IoT. As a result, further discussions regarding named function networking in mobile scenarios have been started in the research community.

List of Figures

List of Tables

References

[1] D. Grewe, M. Wagner, and H. Frey. “ICN-based open, distributed data market place for connected vehicles: Challenges and research directions”. In: *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. May 2017, pp. 265–270. DOI: 10.1109/ICCW.2017.7962668.

[2] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher. “Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions”. In: *Proceedings of the ACM Workshop on Mobile Edge Communications*. MECOMM ’17. 2017, pp. 7–12. DOI: 10.1145/3098208.3098210.

[3] I. N. Bohlin. *Safety belt*. US Patent: US3042625A. 1958.

[4] J. W. Hetrick. *Safety Cushion Assembly for Automotive Vehicles*. US Patent: US2649311A. 1953.

[5] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz. “Inter-vehicle communication: Quo vadis”. In: *IEEE Communications Magazine* 52.6 (2014), pp. 170–177.

[6] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. “Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions”. In: *IEEE Communications Surveys Tutorials* 13.4 (2011), pp. 584–616.

[7] dSPACE GmbH. *Developments on the Electronic Horizon*. 2010. URL: https://www.dspace.com/shared/data/pdf/dspace_magazine/2010-2/english/dSPACE-Magazine_ADAS_2010-02_en.pdf (visited on 05/13/2020).

[8] Pinsent Masons Lawyers. *Connectivity in the Automotive Sector*. 2016. URL: https://silo.tips/download/connectivity-in-the-automotive-sector (visited on 05/13/2020).

[9] M. Meisel, V. Pappas, and L. Zhang. “Ad Hoc Networking via Named Data”. In: *Proceedings of the ACM International Workshop on Mobility in the Evolving Internet Architecture*. MobiArch ’10. 2010, pp. 3–8.

[10] M. Amadeo, C. Campolo, and A. Molinaro. “Content-centric Networking: Is That a Solution for Upcoming Vehicular Networks?” In: *Proceedings of the ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*. 2012, pp. 99–102.

[11] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang. “VANET via Named Data Networking”. In: *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2014, pp. 410–415.

[12] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. “Networking Named Content”. In: *Proceedings of the International Conference on Emerging Networking Experiments and Technologies*. 2009, 1–12.

[13] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. “Named Data Networking”. In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), 66–73.

[14] I. Psaras, W. K. Chai, and G. Pavlou. "Probabilistic In-network Caching for Information-centric Networks". In: *Proceedings of the ACM Workshop on Information-centric Networking*. 2012.

[15] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack. "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks". In: *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM) Workshops*. 2012, pp. 316–321.

[16] N. Abani, G. Farhadi, A. Ito, and M. Gerla. "Popularity-based partial caching for Information Centric Networks". In: *Proceedings of the Mediterranean Ad Hoc Networking Workshop*. 2016, pp. 1–8.

[17] A. Mishra, M. Shin, and W. A. Arbaush. "Context caching using neighbor graphs for fast handoffs in a wireless network". In: *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. Vol. 1. 2004.

[18] D. Grewe, M. Wagner, and H. Frey. "PeRCeIVE: Proactive caching in ICN-based VANETs". In: *Proceedings of the IEEE Vehicular Networking Conference (VNC)*. 2016, pp. 1–8. DOI: 10.1109/VNC.2016.7835962.

[19] D. Grewe, S. Schildt, M. Wagner, and H. Frey. "ADePt: Adaptive Distributed Content Prefetching for Information-Centric Connected Vehicles". In: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. 2018, pp. 1–5. DOI: 10.1109/VTCSpring.2018.8417748.

[20] D. Grewe, M. Wagner, and H. Frey. "A Domain-Specific Comparison of Information-Centric Networking Architectures for Connected Vehicles". In: *IEEE Communications Surveys Tutorials* 20.3 (2018), pp. 2372–2388. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2817653.

[21] D. Grewe, K. P. P. Rao, S. Schildt, M. Wagner, D. Schoop, and H. Frey. "EnCIRCLE: Encryption-based access control for information-centric connected vehicles". In: *in Proceedings of the International Conference on the Network of the Future (NOF)*. Nov. 2017, pp. 114–119. DOI: 10.1109/NOF.2017.8251229.

[22] M. Król, C. Marxer, D. Grewe, I. Psaras, and C. Tschudin. "Open Security Issues for Edge Named Function Environments". In: *IEEE Communications Magazine* 56.11 (Nov. 2018), pp. 69–75. ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1701117.

[23] A. Festag. "Cooperative intelligent transport systems standards in europe". In: *IEEE Communications Magazine* 52.12 (2014), pp. 166–172.

[24] Robert Bosch GmbH. *Bosch Mediaspace*. 2020. URL: https://www.bosch-mediaspace.de/ (visited on 05/13/2020).

[25] Amazon Web Services, Inc. *Amazon Web Services AWS product website*. 2020. URL: https://aws.amazon.com/ (visited on 05/13/2020).

[26] Microsoft Corporation. *Microsoft Azure website*. 2020. URL: https://azure.microsoft.com/ (visited on 05/13/2020).

[27] I. Lequerica, P. M. Ruiz, and V. Cabrera. "Improvement of vehicular communications by using 3G capabilities to disseminate control information". In: *IEEE Network* 24.1 (Jan. 2010), pp. 32–38. ISSN: 0890-8044. DOI: 10.1109/MNET.2010.5395781.

[28] 3rd Generation Partnership Project. *Project page of the 3rd Generation Partnership Project (3GPP)*. 2020. URL: `http://www.3gpp.org/` (visited on 05/13/2020).

[29] 3rd Generation Partnership Project. *Roadmap of 3GPP Releases*. 2020. URL: `http://www.3gpp.org/specifications/releases` (visited on 05/13/2020).

[30] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro. "LTE for vehicular networking: a survey". In: *IEEE Communications Magazine* 51.5 (2013), pp. 148–157.

[31] 3rd Generation Partnership Project. *RP-161298: LTE-based V2X Services*. Tech. rep. Sept. 2016. URL: `https://portal.3gpp.org/ngppapp/CreateTDoc.aspx?mode=view&contributionUid=RP-161298` (visited on 05/13/2020).

[32] 3rd Generation Partnership Project. *RP-161272: Support for V2V services based on LTE sidelink*. Tech. rep. Sept. 2016.

[33] A. Bazzi, B. M. Masini, A. Zanella, and I. Thibault. "On the Performance of IEEE 802.11p and LTE-V2V for the Cooperative Awareness of Connected Vehicles". In: *IEEE Transactions on Vehicular Technology* 66.11 (2017), pp. 10419–10432.

[34] Y. J. Li. "An overview of the DSRC/WAVE technology". In: *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer. 2010, pp. 544–558.

[35] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". In: *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* (2016).

[36] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band". In: *IEEE Std 802.11a-1999* (Dec. 1999), pp. 1–102.

[37] "700 MHz Band Intelligent Transport Systems". In: *ARIB STD-T109 v1.3* (2017), pp. 1–245.

[38] "IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture". In: *IEEE Std 1609.0-2013* (2014), pp. 1–78. DOI: `10.1109/IEEESTD.2014.6755433`.

[39] European Telecommunications Standards Institute. *ETSI TS 102 636-3: Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetwork; Part 3: Network architecture v1.1.1*. Tech. rep. 2010.

[40] European Telecommunications Standards Institute. *ETSI EN 302 665: Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band v1.2.0*. Tech. rep. 2012.

[41] C. Li, J. Jiang, W. Chen, T. Ji, and J. Smee. "5G ultra-reliable and low-latency systems design". In: *2017 European Conference on Networks and Communications (EuCNC)*. 2017, pp. 1–5.

[42] 5G Automotive Association. *Project page of the 5G Automotive Association (5GAA)*. 2020. URL: `http://5gaa.org/` (visited on 05/13/2020).

[43] Electronic Communications Committee. *ECC Decision (09)01: Harmonised use of the 63-64GHz frequency band for Intelligent Transportation Systems*. Tech. rep. Mar. 2016.

[44] M. Armbrust et al. “A View of Cloud Computing”. In: *Commun. ACM* 53.4 (2010), pp. 50–58.

[45] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. “Fog Computing and Its Role in the Internet of Things”. In: *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*. 2012, pp. 13–16.

[46] M. Gerla. “Vehicular Cloud Computing”. In: *Proceedings of the Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. 2012, pp. 152–155.

[47] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen. “Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures”. In: *IEEE Transactions on Vehicular Technology* 65.6 (2016), pp. 3860–3873.

[48] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. “A survey and comparison of peer-to-peer overlay network schemes”. In: *IEEE Communications Surveys Tutorials* 7.2 (2005), pp. 72–93.

[49] G. Pallis and A. Vakali. “Insight and Perspectives for Content Delivery Networks”. In: *Commun. ACM* 49.1 (2006), pp. 101–106.

[50] M. Pathan and R. Buyya. “A Taxonomy of CDNs”. In: *Content Delivery Networks*. Springer Berlin Heidelberg, 2008, pp. 33–77.

[51] Akamai Technologies, Inc. *Akamai Technologies, Inc. website*. 2020. URL: `https://www.akamai.com/` (visited on 05/13/2020).

[52] Amazon Web Services, Inc. *Amazon CloudFront: Highly programmable, secure content delivery network*. 2020. URL: `https://aws.amazon.com/cloudfront/` (visited on 05/13/2020).

[53] Google LLC. *Peering with Google's data centers*. 2020. URL: `https://peering.google.com/` (visited on 05/13/2020).

[54] K. Fall. “A Delay-tolerant Network Architecture for Challenged Internets”. In: *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM ’03. 2003, pp. 27–34.

[55] K. Fall and S. Farrell. “DTN: an architectural retrospective”. In: *IEEE Journal on Selected Areas in Communications* 26.5 (2008), pp. 828–836.

[56] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues. “Improving Vehicular Delay-Tolerant Network Performance with Relay Nodes”. In: *Proceedings of the Next Generation Internet Networks*. 2009, pp. 1–5.

[57] P. R. Pereira, A. Casaca, J. J. P. C. Rodrigues, V. N. G. J. Soares, J. Triay, and C. Cervello-Pastor. “From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks”. In: *IEEE Communications Surveys Tutorials* 14.4 (2012), pp. 1166–1182.

[58] K. Scott and S. Burleigh. *RFC 5050: Bundle Protocol Specification (Experimental)*. Request for Comments 5050. 2007.

[59] M. Demmer, J. Ott, and S. Perreault. *RFC 7242: Delay-Tolerant Networking TCP Convergence-Layer Protocol*. Request for Comments 7242. 2014.

[60] M. Nottingham. *RFC 7320: URI Design and Ownership*. Request for Comments 7320. 2014.

[61] M. Gritter and D. R. Cheriton. "An Architecture for Content Routing Support in the Internet." In: *Proceedings of the 3rd Usenix Symposium on Internet Techologies and Systems*. 2001, pp. 37–48.

[62] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. "A survey of information-centric networking". In: *Communications Magazine, IEEE* 50.7 (2012), pp. 26–36.

[63] G. Xylomenos, C.N. Ververidis, V.A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K.V. Katsaros, and G.C. Polyzos. "A Survey of Information-Centric Networking Research". In: *IEEE Communications Surveys Tutorials, IEEE* 16.2 (2014), pp. 1024–1049.

[64] M. Król, S. Mastorakis, D. Oran, and D. Kutscher. "Compute First Networking: Distributed Computing Meets ICN". In: *Proceedings of the ACM Conference on Information-Centric Networking*. 2019, 67–77.

[65] M. Sifalakis, B. Kohler, C. Scherb, and C. Tschudin. "An information centric network for computing the distribution of computations". In: *Proceedings of the International Conference on Information-centric Networking*. 2014, 137–146.

[66] M. Król and I. Psaras. "NFaaS: Named Function As a Service". In: *Proceedings of the ACM Conference on Information-Centric Networking*. 2017, pp. 134–144. ISBN: 978-1-4503-5122-5.

[67] M. Yan, P. Castro, P. Cheng, and V. Ishakian. "Building a Chatbot with Serverless Computing". In: *Proceedings of the International Workshop on Mashups of Things and APIs*. MOTA '16. 2016, 5:1–5:4.

[68] A. Madhavapeddy and D. J. Scott. "Unikernels: Rise of the Virtual Library Operating System". In: *Queue* 11.11 (2013), pp. 30–44.

[69] Statista, Inc. *Number of internet users worldwide from 2005 to 2019 (in millions)*. 2018. URL: `https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/` (visited on 05/13/2020).

[70] J. Wang. "A Survey of Web Caching Schemes for the Internet". In: *SIGCOMM Comput. Commun. Rev.* 29.5 (Oct. 1999), pp. 36–46.

[71] K. Wong. "Web cache replacement policies: a pragmatic approach". In: *IEEE Network* 20.1 (2006), pp. 28–34.

[72] P. Krishnan, D. Raz, and Y. Shavitt. "The Cache Location Problem". In: *IEEE/ACM Trans. Netw.* 8.5 (Oct. 2000), pp. 568–582.

[73] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker. "Packet Caches on Routers: The Implications of Universal Redundant Traffic Elimination". In: *Proceedings of the ACM SIGCOMM Conference on Data Communication*. SIGCOMM '08. 2008, pp. 219–230.

[74] P. Rodriguez, C. Spanner, and E. W. Biersack. "Web caching architectures: hierarchical and distributed caching". In: *in Proceedings of WCW*. Vol. 99. 1999.

[75] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwarz, and K. J. Worrel. "A Hierarchical Internet Object Cache". In: *Proceedings of the Usenix Technical Conference*. 1996, pp. 153–164.

[76] R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay. “Design considerations for distributed caching on the Internet”. In: *Proceedings of the IEEE International Conference on Distributed Computing Systems*. 1999, pp. 273–284.

[77] D. Wessels and K. Claffy. *RFC 2186: Internet Cache Protocol (ICP)*. Request for Comments 2186. 1997.

[78] V. Khatri and C. V. Brown. “Designing Data Governance”. In: *Commun. ACM* 53.1 (2010), pp. 148–152.

[79] M. R. Korupolu and M. Dahlin. “Coordinated placement and replacement for large-scale distributed caches”. In: *IEEE Transactions on Knowledge and Data Engineering* 14.6 (2002), pp. 1317–1329.

[80] S. Podlipnig and L. Böszörmenyi. “A Survey of Web Cache Replacement Strategies”. In: *ACM Comput. Surv.* 35.4 (Dec. 2003), pp. 374–398.

[81] A. Silberschatz, P. B. Galvin, and G. Gagne. *Operating system concepts essentials*. John Wiley & Sons, Inc., 2014. ISBN: 978-1-118-80492-6.

[82] J. T. Robinson and M. V. Devarakonda. “Data Cache Management Using Frequency-based Replacement”. In: *SIGMETRICS Perform. Eval. Rev.* 18.1 (Apr. 1990), pp. 134–142.

[83] M. Abrams, C. Standridge, G. Abdulla, S. Williams, and E. Fox. In: *Proceedings of the International World Wide Web Conference*. 1995.

[84] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, and T. Jin. “Evaluating Content Management Techniques for Web Proxy Caches”. In: *SIGMETRICS Perform. Eval. Rev.* 27.4 (Mar. 2000), pp. 3–11.

[85] P. Cao and S. Irani. “Cost-Aware WWW Proxy Caching Algorithms”. In: *Proceedings of the Usenix Symposium on Internet Technologies and Systems*. 1997.

[86] J. E. Pitkow and M. Recker. In: *Proceedings of the International World Wide Web Conference*. 1994, pp. 1039–1046.

[87] G. Zhang, Y. Li, and T. Lin. “Caching in information centric networking: A survey”. In: *Computer Networks* 57.16 (2013), pp. 3128–3141.

[88] A. Ioannou and S. Weber. “A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking”. In: *IEEE Communications Surveys Tutorials* 18.4 (2016), pp. 2847–2886.

[89] N. Laoutaris, H. Che, and I. Stavrakakis. “The LCD interconnection of LRU caches and its analysis”. In: *Performance Evaluation* 63.7 (2006), pp. 609–634.

[90] S. Arianfar, P. Nikander, and J. Ott. “On Content-centric Router Design and Implications”. In: *Proceedings of the Re-Architecting the Internet Workshop*. ReARCH ’10. 2010, 5:1–5:6.

[91] W. K. Chai, D. He, I. Psaras, and G. Pavlou. “Cache “Less for More” in Information-Centric Networks”. In: *NETWORKING 2012*. Springer Berlin Heidelberg, 2012, pp. 27–40. ISBN: 978-3-642-30045-5.

[92] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. “How to model an internetwork”. In: *Proceedings of IEEE Conference on Computer Communications (INFOCOM ’96)*. Vol. 2. 1996, 594–602 vol.2.

[93] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. Davies, A. Molinaro, and S. Eum. *RFC 7476: Information-Centric Networking: Baseline Scenarios*. Request for Comments 7476. 2015.

[94] V. Jacobson. "A new way to look at networking". In: *Google Tech Talk*. 2006.

[95] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. "A Data-oriented (and Beyond) Network Architecture". In: *SIGCOMM Computer Communication Review* 37.4 (2007), 181–192.

[96] The FP7 Publish-Subscribe Internet Routing Paradigm (PSIRP) project. *PSIRP project page*. 2016. URL: `http://www.psirp.org/` (visited on 05/13/2020).

[97] FP7 (4WARD) project. *4WARD - Architecture and Design for the Future Internet*. 2016. URL: `https://cordis.europa.eu/project/id/216041/de` (visited on 05/13/2020).

[98] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander. "LIPSIN: Line Speed Publish/Subscribe Inter-networking". In: *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*. 2009, 195–206.

[99] FP7 PUblish SUbscribe Internet Technolgy (PURSUIT) project. 2016. URL: `https://cordis.europa.eu/project/id/257217/de` (visited on 05/13/2020).

[100] FP7 Scalable and Adaptive Internet Solutions (SAIL) project. *Network of Information (NetInf)*. 2016. URL: `https://cordis.europa.eu/project/id/257448` (visited on 05/13/2020).

[101] FP7 Scalable and Adaptive Internet Solutions (SAIL) project. 2016. URL: `http://www.sail-project.eu/` (visited on 05/13/2020).

[102] Xerox Palo Alto Research Center. *The CCNx Project*. 2016. URL: `https://www.parc.com/blog/project-ccnx-announces-the-ccnx-v1-0-protocol-roadmap/` (visited on 05/13/2020).

[103] Andrea Detti, Nicola Blefari Melazzi, Stefano Salsano, and Matteo Pomposini. "CONET: A Content Centric Inter-networking Architecture". In: *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*. 2011, 50–55.

[104] FP7 CONVERGENCE project. 2016. URL: `http://www.ict-convergence.eu/` (visited on 05/13/2020).

[105] FP7 COnvergence of fixed and Mobile BrOadband access/aggregation networks (COMBO) project. 2016. URL: `http://www.ict-combo.eu/` (visited on 05/13/2020).

[106] M. Amadeo and A. Molinaro. "CHANET: A content-centric architecture for IEEE 802.11 MANETs". In: *Proceedings of the International Conference on the Network of the Future (NOF)*. 2011, pp. 122–127.

[107] NSF MobilityFirst project. 2016. URL: `http://mobilityfirst.winlab.rutgers.edu/` (visited on 05/13/2020).

[108] FP7 GreenICN project. 2016. URL: `http://www.greenicn.org/` (visited on 05/13/2020).

[109] C. Tschudin and M. Sifalakis. "Named functions and cached computations". In: *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*. 2014, pp. 851–857.

[110] FP7 COntent Mediator architecture for content-aware nETworks (COMET) project. 2016. URL: http://www.comet-project.org/index.html (visited on 05/13/2020).

[111] Architecture for an Internet for everybody (RIFE) project. 2015. URL: https://rife-project.eu/ (visited on 05/13/2020).

[112] EU H2020 UMobile project. *Universal, mobile-centric and opportunistic communications architecture (UMobile*. 2015. URL: http://www.umobile-project.eu/ (visited on 05/13/2020).

[113] Bonvoyage H2020 project. 2015. URL: http://bonvoyage2020.eu/ (visited on 05/13/2020).

[114] H2020 POINT (iP Over IcN - the betTer IP) project. 2016. URL: https://cordis.europa.eu/project/id/643990/de (visited on 05/13/2020).

[115] D. Trossen, A. Sathiaseelan, and J. Ott. "Towards an Information Centric Network Architecture for Universal Internet Access". In: *SIGCOMM Comput. Commun. Rev.* 46.1 (2016), pp. 44–49.

[116] EU H2020 UMobile project. *UMobile - Deliverable: D3.2: UMOBILE architecture report (2)*. Tech. rep. 2017. URL: http://www.umobile-project.eu/phocadownload/deliverables/D3.2_-_UMOBILE_architecture_report_(2).pdf (visited on 05/13/2020).

[117] M. Al-Khalidi et al. *Deliverable D2.4: Scenarios, Requirements, Specifications and KPIs, final version*. Tech. rep. 2017. URL: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5a01ee764&appId=PPGMS (visited on 05/13/2020).

[118] Palo Alto Research Center. 2017. URL: https://www.parc.com/blog/cisco-acquires-parcs-content-centric-networking-ccn-platform/ (visited on 05/13/2020).

[119] FD.io Linux Foundation Project. *Community ICN (CICN)*. 2017. URL: https://wiki.fd.io/view/Cicn (visited on 05/13/2020).

[120] National Science Foundation and Intel Labs. *NSF/Intel Partnership on Information-Centric Networking in Wireless Edge Networks (ICN-WEN)*. 2017. URL: https://www.nsf.gov/pubs/2016/nsf16586/nsf16586.htm (visited on 05/13/2020).

[121] Cefore Project Web page. *Cefore: Information Centric Networking Platform*. 2018. URL: https://cefore.net/ (visited on 05/13/2020).

[122] NSF/Intel ICN-AR project. *ICN-Enabled Secure Edge Networking with Augmented Reality (ICE-AR)*. 2017. URL: http://ice-ar.named-data.net/team.html (visited on 05/13/2020).

[123] EU H2020 ICN2020 project. *ICN2020: Advancing ICN towards real-world deployment through research, innovative applications, and global scale experimentation*. 2016. URL: http://www.icn2020.org/ (visited on 05/13/2020).

[124] IKT 2020 i3 project. *Information Centric Networking for the Industrial Internet*. 2016. URL: http://i3.realmv6.org/ (visited on 05/13/2020).

[125] G. Tyson, A. Mauthe, S. Kaune, P. Grace, and T Plagemann. "Juno: An adaptive delivery-centric middleware". In: *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*. 2012, pp. 587–591.

[126] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. K. Ramakrishnan. "COPSS: An Efficient Content Oriented Publish/Subscribe System". In: *Proceedings of the ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. 2011, pp. 99–110.

[127] ANR Connect Project. *Content-Oriented Networking: a New Experience for Content Transfer*. 2010. URL: `https://anr.fr/Project-ANR-10-VERS-0001` (visited on 05/13/2020).

[128] D. Pesavento, G. Grassi, C.E. Palazzi, and G. Pau. "A naming scheme to represent geographic areas in NDN". In: *Proceedings of the IEEE IFIP Wireless Days (WD)*. 2013, pp. 1–3.

[129] S. H. Bouk, S. H. Ahmed, and D. Kim. "Vehicular Content Centric Network (VCCN): A Survey and Research Challenges". In: *Proceedings of the Annual ACM Symposium on Applied Computing*. SAC '15. 2015.

[130] M. Amadeo, C. Campolo, and A. Molinaro. "Information-centric networking for connected vehicles: a survey and future perspectives". In: *IEEE Communications Magazine* 54.2 (2016).

[131] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang. "Named Data Networking of Things (Invited Paper)". In: *Proceedings of the IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 2016, pp. 117–128.

[132] IHS Markit, Ltd. *Vehicles Getting Older: Average Age of Light Cars and Trucks in U.S. Rises Again in 2016 to 11.6 Years, IHS Markit Says*. 2016. URL: `https://news.ihsmarkit.com/press-release/automotive/vehicles-getting-older-average-age-light-cars-and-trucks-us-rises-again-201` (visited on 05/13/2020).

[133] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe. "A Survey of Mobility in Information-centric Networks: Challenges and Research Directions". In: *Proceedings of the ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*. NoM '12. 2012, pp. 1–6.

[134] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang. "A survey of mobility support in Named Data Networking". In: *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Apr. 2016, pp. 83–88.

[135] A. Detti, C. Pisa, and N. Blefari Melazzi. "Modeling multipath forwarding strategies in Information Centric Networks". In: *Proceeding of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2015, pp. 324–329.

[136] N. Abani, T. Braun, and M. Gerla. "Proactive Caching with Mobility Prediction Under Uncertainty in Information-centric Networks". In: *Proceedings of the ACM Conference on Information-Centric Networking*. ICN '17. 2017, pp. 88–97.

[137] European Telecommunications Standards Institute (ETSI). *ETSI TR 102 638: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions*. Tech. rep. 2009.

[138] D. Kutscher, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, and M. Waehlisch. *RFC 7927: Information-Centric Networking (ICN) Research Challenges*. Request for Comments 7927. July 2016.

[139] MarkLines Co., Ltd. *Top 30 global supplier rankings for FY 2016*. July 2017. URL: `https://www.marklines.com/en/report/rep1614_201706` (visited on 05/13/2020).

[140] Y. Yu, A. Afanasyev, Z. Zhu, and L. Zhang. *NDN-0023, Revision 1: NDN Technical Memo: Naming Conventions*. Tech. rep. 2014. URL: `https://named-data.net/publications/techreports/ndn-tr-22-ndn-memo-naming-conventions/` (visited on 05/13/2020).

[141] T. Kaerkkaeinen et al. *Deliverable 3.3: Final platform design and set of dissemination strategies*. Tech. rep. 2017. URL: `https://rife-project.eu/wp-content/uploads/sites/31/2017/08/RIFE_D3.3_final.pdf` (visited on 05/13/2020).

[142] I. Psaras, S. Reñé, K. V. Katsaro, V. Sourlas, G. Pavlou, N. Bezirgiannidis, S. Diamantopoulos, I. Komnios, and V. Tsaoussidis. "Keyword-based Mobile Application Sharing". In: *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*. MobiArch '16. 2016, pp. 1–6.

[143] EU H2020 UMobile project. *UMobile - Deliverable: D3.4: UMOBILE ICN layer abstraction final specification*. Tech. rep. 2017. URL: `http://www.umobile-project.eu/phocadownload/deliverables/D3.4_-_UMOBILE_ICN_layer_abstraction_final_specification.pdf` (visited on 05/13/2020).

[144] N. B. Melazzi, A. Detti, M. Arumaithurai, and K. K. Ramakrishnan. "Internames: A name-to-name principle for the future Internet". In: *Proceedings of the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. 2014, pp. 146–151.

[145] FP7 PUblish SUbscribe Internet Technolgy (PURSUIT) project. *Deliverable: D2.3 Architecture Definition, Components Descriptions and Requirements*. Tech. rep. 2013. URL: `http://www.fp7-pursuit.eu/PursuitWeb/?page_id=158` (visited on 12/31/2018).

[146] FP7 Scalable and Adaptive Internet Solutions (SAIL) project. *Network of Information (NetInf) - Deliverable: D.B.3 (D-3.3) Final NetInf Architecture*. Tech. rep. 2013. URL: `https://sail-project.eu/wp-content/uploads/2013/01/SAIL-DB3-v1.1-final-public.pdf` (visited on 05/13/2020).

[147] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri. "MobilityFirst Future Internet Architecture Project". In: *Proceedings of the Asian Internet Engineering Conference*. 2011, 1–3.

[148] A. Baid, S. Mukherjee, T. Vu, S. Mudigonda, K. Nagaraja, J. Fukuyama, and D. Raychaudhuri. "Enabling vehicular networking in the MobilityFirst future internet architecture". In: *Proceedings of the IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–3.

[149] S. C. Nelson, G. Bhanage, and D. Raychaudhuri. "GSTAR: Generalized Storage-aware Routing for Mobilityfirst in the Future Mobile Internet". In: *Proceedings of the International Workshop on MobiArch*. 2011, 19–24.

[150] FP7 COntent Mediator architecture for content-aware nETworks (COMET) project. *Deliverable D3.2 Final Specification of Mechanisms, Protocols and Algorithms for the Content Mediation System*. Tech. rep. 2012. URL: `http://www.comet-project.org/deliverables.html` (visited on 05/13/2020).

[151] M. Mosko, I. Solis, and C. Wood. *RFC 8609: CCNx Messages in TLV Format draft-irtf-icnrg-ccnxmessages-08*. Request for Comments 8609. 2019.

[152] NSF Named Data Networking project. *NDN Packet Format Specification v0.2.1*. Tech. rep. 2016. URL: `http://named-data.net/doc/NDN-packet-spec/current/changelog.html#version-0-2-1` (visited on 05/13/2020).

[153] FP7 CONVERGENCE project. *Deliverable D3.2 System architecture*. Tech. rep. 2016. URL: `http://www.ict-convergence.eu/deliverables/` (visited on 05/13/2020).

[154] Z. Li, J. Point, S. Ciftci, O. Eker, G. Mauri, M. Savi, and G. Verticale. "ICN Based Shared Caching in Future Converged Fixed and Mobile Network". In: *Proceedings of the IEEE International Conference on High Performance Switching and Routing (HPSR)*. 2015.

[155] FP7 GreenICN project. *Deliverable D3.4.3 Final Specification of cross-layer designs and trade-off management for video delivery in in-network caching mobile environments*. Tech. rep. 2015. URL: `http://www.greenicn.org/deliverables/deliverables/` (visited on 05/13/2020).

[156] FP7 PUblish SUbscribe Internet Technolgy (PURSUIT) project. *Deliverable: D3.5 Final Integrated Prototype*. Tech. rep. 2016. URL: `http://www.fp7-pursuit.eu/PursuitWeb/?page_id=158` (visited on 12/31/2018).

[157] FP7 COntent Mediator architecture for content-aware nETworks (COMET) project. *Deliverable D4.2 Final Specification of Mechanisms, Protocols and Algorithms for Enhanced Network Platforms*. Tech. rep. 2010. URL: `http://www.comet-project.org/deliverables.html` (visited on 05/13/2020).

[158] FP7 CONVERGENCE project. *Deliverable D5.3 Final protocol architecture*. Tech. rep. 2012. URL: `http://www.ict-convergence.eu/deliverables/` (visited on 05/13/2020).

[159] X. Vasilakos, M. Al-Khalidi, V. A. Siris, M. J. Reed, N. Thomos, and G. C. Polyzos. "Mobility-based Proactive Multicast for Seamless Mobility Support in Cellular Network Environments". In: *Proceedings of the SIGCOMM Workshop on Mobile Edge Communications*. MECOMM '17. 2017, pp. 25–30.

[160] I. Psaras, L. Saino, M. Arumaithurai, K. K. Ramakrishnan, and G. Pavlou. "Name-based replication priorities in disaster cases". In: *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2014, pp. 434–439.

[161] N. Blefari, P. Boccadoro, A. Detti, L. A. Grieco, M. Losciale, G. Piro, G. Ribezzo, and G. Tropea. *Deliverable D3.2: Publish/Subscribe and security functionality*. Tech. rep. 2017. URL: `http://bonvoyage2020.eu/results/deliverables/` (visited on 05/13/2020).

[162] N. Blefari, P. Boccadoro, A. Detti, L. A. Grieco, G. Piro, G. Ribezzo, and G. Tropea. *Deliverable D3.1: Networking*. Tech. rep. 2016. URL: `http://bonvoyage2020.eu/results/deliverables/` (visited on 05/13/2020).

[163] J. Lee, S. Cho, and D. Kim. "Device mobility management in content-centric networking". In: *IEEE Communications Magazine* 50.12 (2012), pp. 28–34.

[164] F. Hermans, E. Ngai, and P. Gunningberg. "Global Source Mobility in the Content-centric Networking Architecture". In: *Proceedings of the ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*. 2012, 13–18.

[165] FP7 Scalable and Adaptive Internet Solutions (SAIL) project. *Network of Information (NetInf) - Deliverable: D.B.1: The Network of Information: Architecture and applications*. Tech. rep. 2013. URL: `https://sail-project.eu/wp-content/uploads/2011/08/SAIL_DB1_v1_0_final-Public.pdf` (visited on 05/13/2020).

[166] M. Mosko, I. Solis, and E. Uzun. *CCN 1.0 Protocol Architecture*. Tech. rep. 2016. URL: `https://wiki.fd.io/view/Cicn` (visited on 05/13/2020).

[167] Y. Rao, H. Zhou, D. Gao, H. Luo, and Y. Liu. "Proactive Caching for Enhancing User-Side Mobility Support in Named Data Networking". In: *Proceedings of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. 2013.

[168] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang. "A Case for Stateful Forwarding Plane". In: *Computer Communications* 36.7 (2013).

[169] NSF Named Data Networking project. *Packet Fragmentation in NDN: Why NDN Uses Hop-By-Hop Fragmentation (NDN Memo)*. Tech. rep. 2015. URL: `http://named-data.net/publications/techreports/ndn-0032-1-ndn-memo-fragmentation/` (visited on 05/13/2020).

[170] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi. "Transport-layer Issues in Information Centric Networks". In: *Proceedings of the ICN Workshop on Information-centric Networking*. 2012, 19–24.

[171] H. Nakazato, S. Zhang, Y. J. Park, A. Detti, D. Bursztynowski, Z. Kopertowski, and I. Psaras. "On-Path Resolver Architecture for Mobility Support in Information Centric Networking". In: *Proceedings of the IEEE Globecom Workshops (GC Wkshps)*. 2015, pp. 1–6.

[172] K. Kanai, T. Muto, H. Kisara, J. Katto, T. Tsuda, W. Kameyama, Y. J. Park, and T. Sato. "Proactive content caching utilizing transportation systems and its evaluation by field experiment". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*. 2014, pp. 1382–1387.

[173] M. Al-Naday, N. Fotiou, A. Karila, K. Katsaros, G. Petropoulos, A. Phinikarides, M. J. Reed, S. Robitzsch, Y. Thomas, and G. Xylomenos. *POINT System Architecture and Specifications*. Tech. rep. 2018. URL: `https://www.point-h2020.eu/wp-content/uploads/2018/04/POINT_TR_001_1.00.pdf` (visited on 10/24/2018).

[174] M. Al-Naday et al. *Deliverable D4.3: First System Evaluation Report*. Tech. rep. 2016. URL: `https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5a8c79796&appId=PPGMS` (visited on 05/13/2020).

[175] B. A. Alzahrani, M. J. Reed, J. Riihijärvi, and V. G. Vassilakis. "Scalability of information centric networking using mediated topology management". In: *Journal of Network and Computer Applications* 50 (2015), pp. 126–133.

[176] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. "Accountable Internet Protocol (Aip)". In: *ACM SIGCOMM Computer Communication Review* 38.4 (2008), 339–350.

[177] N. Blefari Melazzi, S. Salsano, A. Detti, G. Tropea, L. Chiariglione, A. Difino, A. C. G. Anadiotis, A. S. Mousas, I. S. Venieris, and C. Z. Patrikakis. "Publish/subscribe over information centric networks: A Standardized approach in CONVERGENCE". In: *Proceedings of the Future Network Mobile Summit (FutureNetw)*. 2012, pp. 1–8.

[178] C. Sarros et al. “Connecting the Edges: A Universal, Mobile-Centric, and Opportunistic Communications Architecture”. In: *IEEE Communications Magazine* 56.2 (2018), pp. 136–143.

[179] Information-Centric Networking Research Group (ICNRG) at IETF. *Design Choices and Differences for NDN and CCNx 1.0 Implementations of Information-Centric Networking*. 2017. URL: `https://icnrg.github.io/draft-icnrg-harmonization/draft-icnrg-harmonization-00.txt` (visited on 05/13/2020).

[180] I. Choi, B. Lee, H. Jeon, H. Song, and Y Jeong. “VSCCN: CCN with a very simple control plane”. In: *Proceedings of the IEEE International Conference on Advanced Communication Technology (ICACT)*. 2012, pp. 690–693.

[181] FP7 PUblish SUbscribe Internet Technolgy (PURSUIT) project. *Deliverable: D2.2 Conceptual Architecture: Principles, patterns and sub-components descriptions*. Tech. rep. 2016. URL: `http://www.fp7-pursuit.eu/PursuitWeb/?page_id=158` (visited on 12/31/2018).

[182] FP7 PUblish SUbscribe Internet Technolgy (PURSUIT) project. *Source Code Repository of the PURSUIT Blackadder implementation on GitHub*. 2012. URL: `https://github.com/fp7-pursuit/blackadder` (visited on 05/13/2020).

[183] NSF Named Data Networking project. 2018. URL: `http://named-data.net/` (visited on 05/13/2020).

[184] H2020 POINT (iP Over IcN - the betTer IP) project. *Source Code Repository of the POINT project on GitHub*. 2016. URL: `https://github.com/point-h2020/` (visited on 05/13/2020).

[185] H2020 Architecture for an Internet for everybody (RIFE) project. *Source Code Repositories of the RIFE project*. 2018. URL: `https://www.rife-project.eu/publications/open-source-resources/` (visited on 05/13/2020).

[186] EU H2020 UMobile project. *Source Code Repository of the project on GitHub*. 2017. URL: `https://github.com/umobileproject` (visited on 05/13/2020).

[187] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang. “Data naming in Vehicle-to-Vehicle communications”. In: *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2012, pp. 328–333.

[188] W. Drira and F. Filali. “NDN-Q: An NDN query mechanism for efficient V2X data collection”. In: *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking Workshops (SECON Workshops)*. 2014, pp. 13–18.

[189] M. Amadeo, C. Campolo, and A. Molinaro. “Named data networking for priority-based content dissemination in VANETs”. In: *Proceedings of the International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016, pp. 1–6.

[190] S. H. Bouk, S. H. Ahmed, and D. Kim. “Hierarchical and hash based naming with Compact Trie name management scheme for Vehicular Content Centric Networks”. In: *Computer Communications* 71 (2015), pp. 73–83.

[191] R. Ravindran, S. Lo, X. Zhang, and G. Wang. “Supporting seamless mobility in named data networking”. In: *Proceedings of the IEEE International Conference on Communications (ICC)*. 2012, pp. 5854–5869.

[192] J. M. Duarte, T. Braun, and L. A. Villas. “Receiver Mobility in Vehicular Named Data Networking”. In: *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*. MobiArch ’17. 2017, pp. 43–48.

[193] J. M. Duarte, T. Braun, and L. A. Villas. “Source Mobility in Vehicular Named-Data Networking: An Overview”. In: *Ad Hoc Networks*. 2018, pp. 83–93.

[194] P. TalebiFard and V. C.M. Leung. “A Content Centric Approach to Dissemination of Information in Vehicular Networks”. In: *Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*. DIVANet ’12. 2012, pp. 17–24.

[195] Y. Yu, Y. Li, X. Ma, W. Shang, M. Y. Sanadidi, and M. Gerla. “Scalable opportunistic VANET Content Routing with encounter information”. In: *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*. 2013, pp. 1–6.

[196] M. Amadeo, C. Campolo, and A. Molinaro. “Design and analysis of a transport-level solution for content-centric VANETs”. In: *Proceedings of the IEEE International Conference on Communications Workshops (ICC)*. 2013, pp. 532–537.

[197] M. Amadeo, C. Campolo, and A. Molinaro. “A novel hybrid forwarding strategy for content delivery in wireless information-centric networks”. In: *Computer Communications* 109 (2017), pp. 104–116.

[198] Y. Yu, M. Gerla, and M. Y. Sanadidi. “Scalable VANET content routing using hierarchical bloom filters”. In: *Wireless Communications and Mobile Computing* 15.6 (), pp. 1001–1014.

[199] Q. Wang, D. Xie, and X Ji. “Network codes-based content-centric transmission control in VANET”. In: *Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE)*. 2015, pp. 157–162.

[200] C. Bian, T. Zhao, X. Li, and W. Yan. “Boosting named data networking for efficient packet forwarding in urban VANET scenarios”. In: *Proceedings of the IEEE International Workshop on Local and Metropolitan Area Networks*. 2015, pp. 1–6.

[201] X. Wang, W. Liu, L. Yang, W. Zhang, and C. Peng. “A new content-centric routing protocol for Vehicular Ad Hoc Networks”. In: *Proceedings of the Asia-Pacific Conference on Communications (APCC)*. 2016, pp. 552–558.

[202] S. H. Ahmed, S. H. Bouk, M. A. Yaqub, D. Kim, H. Song, and J. Lloret. “CODIE: Controlled Data and Interest Evaluation in Vehicular Named Data Networks”. In: *IEEE Transactions on Vehicular Technology* 65.6 (2016), pp. 3954–3963.

[203] M. Kuai, X. Hong, and Q. Yu. “Density-Aware Delay-Tolerant Interest Forwarding in Vehicular Named Data Networking”. In: *Proceedings of the 84th Vehicular Technology Conference (VTC-Fall)*. 2016, pp. 1–5.

[204] X. Yu, R. W. L. Coutinho, A. Boukerche, and A. A. F. Loureirol. “A distance-based interest forwarding protocol for vehicular information-centric networks”. In: *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–5.

[205] A. Boukerche, R. W. L. Coutinho, and X. Yu. “LISIC: A Link Stability-Based Protocol for Vehicular Information-Centric Networks”. In: *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 2017, pp. 233–240.

[206] Y. Hui, Z. Su, and T. H. Luan. "Content in Motion: A Novel Relay Scheme for Content Dissemination in Urban Vehicular Networks". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*. 2016, pp. 1–5.

[207] M. F. Majeed, S. H. Ahmed, and M. N. Dailey. "Enabling Push-Based Critical Data Forwarding in Vehicular Named Data Networks". In: *IEEE Communications Letters* 21.4 (2017), pp. 873–876.

[208] C. Anastasiades, J. Weber, and T. Braun. "Dynamic Unicast: Information-centric multihop routing for mobile ad-hoc networks". In: *Computer Networks* 107 (2016), pp. 208–219.

[209] E. Kalogeiton, T. Kolonko, and T. Braun. "A multihop and multipath routing protocol using NDN for VANETs". In: *Proceedings of the Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. 2017, pp. 1–8.

[210] Y. Wang, H. Liu, L. Huang, and J. Stankovic. "Efficient and proactive V2V information diffusion using Named Data Networking". In: *Proceedings of the IEEE/ACM International Symposium on Quality of Service (IWQoS)*. 2016, pp. 1–10.

[211] S. H. Ahmed, S. H. Bouk, and D. Kim. "RUFS: RobUst Forwarder Selection in Vehicular Content-Centric Networks". In: *IEEE Communications Letters* 19.9 (2015), pp. 1616–1619.

[212] X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis. "Proactive Selective Neighbor Caching for Enhancing Mobility Support in Information-centric Networks". In: *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*. ICN '12. 2012, pp. 61–66.

[213] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco. "Social cooperation for information-centric multimedia streaming in highway VANETs". In: *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. 2014, pp. 1–6.

[214] S. Abdelhamid, H. S. Hassanein, G. Takahara, and H. Farahat. "Caching-assisted access for vehicular resources". In: *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*. 2014, pp. 28–36.

[215] H. Tian, M. Mohri, Y. Otsuka, Y. Shiraishi, and M. Morii. "LCE in-network caching on vehicular networks for content distribution in urban environments". In: *Proceedings of the International Conference on Ubiquitous and Future Networks*. 2015, pp. 551–556.

[216] V. A. Siris, X. Vasilakos, and G. C. Polyzos. "Efficient proactive caching for supporting seamless mobility". In: *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2014.

[217] G. Mauri, M. Gerla, F. Bruno, M. Cesana, and G. Verticale. "Optimal Content Prefetching in NDN Vehicle-to-Infrastructure Scenario". In: *IEEE Transactions on Vehicular Technology* 66.3 (2017), pp. 2513–2525.

[218] W. Zhao, Y. Qin, D. Gao, C. H. Foh, and H. Chao. "An Efficient Cache Strategy in Information Centric Networking Vehicle-to-Vehicle Scenario". In: *IEEE Access* 5 (2017), pp. 12657–12667.

[219] J. M. Duarte, T. Braun, and L. A. Villas. "Addressing the Effects of Low Vehicle Density in Highly Mobile Vehicular Named-Data Networks". In: *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. DIVANet '17. 2017, pp. 117–124.

[220] S. Fang and P. Fan. "A Cooperative Caching Algorithm for Cluster-Based Vehicular Content Networks with Vehicular Caches". In: *Proceedings of the IEEE Globecom Workshops (GC Wkshps)*. 2017, pp. 1–6.

[221] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu. "A Cooperative Caching Scheme Based on Mobility Prediction in Vehicular Content Centric Networks". In: *IEEE Transactions on Vehicular Technology* 67.6 (2018), pp. 5435–5444.

[222] Z. Wei, J. Pan, K. Wang, L. Shi, Z. Lyu, and L. Feng. "Data Forwarding and Caching Strategy for RSU Aided V-NDN". In: *Wireless Algorithms, Systems, and Applications*. Springer International Publishing, 2019, pp. 605–612. ISBN: 978-3-030-23597-0.

[223] M. Chowdhury, A. Gawande, and L. Wang. "Secure Information Sharing among Autonomous Vehicles in NDN". In: *Proceedings of the IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 2017, pp. 15–26.

[224] V. Jain, R. S. Kushwah, and R. S. Tomar. "Named Data Network Using Trust Function for Securing Vehicular Ad Hoc Network". In: *Soft Computing: Theories and Applications*. 2019, pp. 463–471.

[225] F. Bai and B. Krishnamachari. "Exploiting the wisdom of the crowd: localized, distributed information-centric VANETs [Topics in Automotive Networking]". In: *IEEE Communications Magazine* 48.5 (2010), pp. 138–146.

[226] J. Wang, R. Wakikawa, and L. Zhang. "DMND: Collecting data from mobiles using Named Data". In: *Proceedings of the IEEE Vehicular Networking Conference*. 2010, pp. 49–56.

[227] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus. "CarSpeak: A Content-centric Network for Autonomous Driving". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. SIGCOMM '12. 2012, pp. 259–270.

[228] M. Amadeo, C. Campolo, and A. Molinaro. "Content-centric vehicular networking: An evaluation study". In: *Proceedings of the Third International Conference on The Network of the Future (NOF)*. 2012, pp. 1–5.

[229] M. Amadeo, C. Campolo, and A. Molinaro. "Enhancing content-centric networking for vehicular environments". In: *Computer Networks* 57.16 (2013), pp. 3222–3234.

[230] M. Amadeo, C. Campolo, and A. Molinaro. "CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks". In: *IEEE Communications Letters* 16.9 (2012), pp. 1380–1383.

[231] M. Chen, D. O. Mau, Y. Zhang, T. Taleb, and V. C. M. Leung. "VENDNET: VEhicular Named Data NETwork". In: *Vehicular Communications* 1.4 (2014), pp. 208–213.

[232] M. Tavan, R. D. Yates, and D. Raychaudhuri. "Connected vehicles under information-centric architectures". In: *Proceedings of the IEEE Vehicular Networking Conference (VNC)*. 2016, pp. 1–8.

[233] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang. "Rapid Traffic Information Dissemination Using Named Data". In: *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*. NoM '12. 2012, pp. 7–12.

[234] C. De Castro, C. Raffaelli, and O. Andrisano. “A dynamic hierarchical VANET architecture for Named Data Networking applications”. In: *2015 IEEE International Conference on Communications (ICC)*. 2015, pp. 3659–3665.

[235] S. H. Ahmed, S. H. Bouk, D. Kim, D. B. Rawat, and H. Song. “Named Data Networking for Software Defined Vehicular Networks”. In: *IEEE Communications Magazine* 55.8 (2017), pp. 60–66.

[236] M. Tarroumi and I. Jabri. “EVNDN: Enhanced vehicular named data networking”. In: *International Symposium on Networks, Computers and Communications (ISNCC)*. 2017, pp. 1–6.

[237] Z. Yan, S. Zeadally, and Y. Park. “A Novel Vehicular Information Network Architecture Based on Named Data Networking (NDN)”. In: *IEEE Internet of Things Journal* 1.6 (2014), pp. 525–532.

[238] J. Chen, M. Jahanian, and K. K. Ramakrishnan. “Black ice! Using Information Centric Networks for timely vehicular safety information dissemination”. In: *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. 2017, pp. 1–6.

[239] J. M. Duarte, E. Kalogeiton, R. Soua, G. Manzo, M. R. Palattella, A. Di Maio, T. Braun, T. Engel, L. A. Villas, and G. A. Rizzo. “A Multi-Pronged Approach to Adaptive and Context Aware Content Dissemination in VANETs”. In: *Mobile Networks and Applications* 23.5 (2017), pp. 1247–1259.

[240] J. M. Duarte, T. Braun, and L. A. Villas. “MobiVNDN: A distributed framework to support mobility in vehicular named-data networking”. In: *Ad Hoc Networks* 82 (2019), pp. 77–90.

[241] R. Hussain, S. H. Bouk, N. Javaid, A. M. Khan, and J. Lee. “Realization of VANET-Based Cloud Services through Named Data Networking”. In: *IEEE Communications Magazine* 56.8 (2018), pp. 168–175.

[242] A. Boukerche and R. Coutinho. “LoICen: A novel location-based and information-centric architecture for content distribution in vehicular networks”. In: *Ad Hoc Networks* 93 (2019).

[243] P. TalebiFard, V. Leung, M. Amadeo, C. Campolo, and A. Molinaro. “Information-Centric Networking for VANETs”. In: *Vehicular ad hoc Networks*. Springer International Publishing, 2015.

[244] X. Liu, Z. Li, P. Yang, and Y. Dong. “Information-centric mobile ad hoc networks and content routing: A survey”. In: *Ad Hoc Networks* 58 (2017), pp. 255–268.

[245] E. Kalogeiton and T. Braun. “Vehicular communication: a survey”. In: *Proceedings of the IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. 2018, pp. 1–10.

[246] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, and A. Ksentini. “Named Data Networking in Vehicular Ad hoc Networks: State-of-the-Art and Challenges”. In: *IEEE Communications Surveys Tutorials* 22.1 (2020), pp. 320–351.

[247] S. Y. Oh, D. Lau, and M. Gerla. “Content Centric Networking in tactical and emergency MANETs”. In: *Proceedings of the IFIP Wireless Days*. 2010, pp. 1–5.

[248] N. Aloulou, M. Ayari, M. F. Zhani, and L. Saidane. "A popularity-driven controller-based routing and cooperative caching for named data networks". In: *Proceedings of the International Conference on the Network of the Future (NOF)*. 2015, pp. 1–5.

[249] V. A. Siris, X. Vasilakos, and D. Dimopoulos. "Exploiting mobility prediction for mobility popularity caching and DASH adaptation". In: *Proceedings of the IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*. 2016, pp. 1–8.

[250] A. Mahmood, C. Casetti, C. F. Chiasserini, P. Giaccone, and J. Harri. "Mobility-aware edge caching for connected cars". In: *Proceedings of the Conference on Wireless On-demand Network Systems and Services (WONS)*. 2016, pp. 1–8.

[251] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, and F. Naït-Abdesselam. "An Optimized Proactive Caching Scheme Based on Mobility Prediction for Vehicular Networks". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*. 2018, pp. 1–6.

[252] D. Grewe, A. Tan, M. Wagner, S. Schildt, and H. Frey. "A Real World Information-Centric Connected Vehicle Testbed Supporting ETSI ITS-G5". In: *Proceedings of the European Conference on Networks and Communications (EuCNC)*. June 2018, pp. 219–223. DOI: 10.1109/EuCNC.2018.8442818.

[253] Named Data Networking Project. *NDN Packet Format Specification 0.3*. Tech. rep. 2018. URL: http://named-data.net/doc/NDN-packet-spec/current/ (visited on 05/13/2020).

[254] C. Tschudin and C. Wood. *File-Like ICN Collection (FLIC) draft-irtf-icnrg-flic-02*. Nov. 2019. URL: https://tools.ietf.org/html/draft-irtf-icnrg-flic-02 (visited on 05/13/2020).

[255] R. Ravindran, A. Chakraborti, S. Amin, M. Mosko, and I. Solis. *Support for Notifications in CCN draft-ravi-ccn-notification-01*. Tech. rep. Mar. 2016. URL: https://tools.ietf.org/html/draft-ravi-ccn-notification-01 (visited on 05/13/2020).

[256] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, 2006. ISBN: 978-0-387-31073-2.

[257] T. T. T. Nguyen and G. Armitage. "A survey of techniques for internet traffic classification using machine learning". In: *IEEE Communications Surveys Tutorials* 10.4 (2008), pp. 56–76.

[258] G. Singh and F. Al-Turjman. "Learning Data Delivery Paths in QoI-Aware Information-Centric Sensor Networks". In: *IEEE Internet of Things Journal* 3.4 (2016), pp. 572–580.

[259] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, S. H. Ahmed, and M. Guizani. "Bringing Deep Learning at the Edge of Information-Centric Internet of Things". In: *IEEE Communications Letters* 23.1 (2019), pp. 52–55.

[260] I. Moiseenko and D. Oran. *Flow Classification in Information Centric Networking v05*. Tech. rep. Jan. 2020. URL: https://tools.ietf.org/html/draft-moiseenko-icnrg-flowclass-05 (visited on 05/13/2020).

[261] G. Bolch, S. Greiner, H. De Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. John Wiley & Sons, 2006. ISBN: 978-0-471-56525-3.

[262] ASFINAG Maut Service GmbH. *European Corridor - Austrian Testbed for Cooperative Systems*. 2017. URL: http://eco-at.info/home.html (visited on 05/13/2020).

[263] Autobahnen- und Schnellstraßen-Finanzierungs-Aktiengesellschaft (ASFINAG). *Website of the ASFINAG company*. 2020. URL: https://www.asfinag.at/ (visited on 05/13/2020).

[264] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. "A First Look at Traffic on Smartphones". In: *Proceedings of the ACM Conference on Internet Measurement*. 2010, pp. 281–287.

[265] N. Vallina-Rodriguez, A. Auçinas, M. Almeida, Y. Grunenberger, K. Papagiannaki, and J. Crowcroft. "RILAnalyzer: A Comprehensive 3G Monitor on Your Phone". In: *Proceedings of the ACM Conference on Internet Measurement*. 2013, pp. 257–264.

[266] R. Fielding, M. Nottingham, and J. Reschke. *Hypertext Transfer Protocol (HTTP/1.1): Caching*. Request for Comments 7234. 2014.

[267] T. Henderson, M. Lacage, G. Riley, M. Watrous, G. Carneiro, and T. Pecorella. *ns-3 Homepage*. Mar. 2018. URL: https://www.nsnam.org/ (visited on 05/13/2020).

[268] A. Afanasyev, S. Mastorakis, I. Moiseenko, and L. Zhang. *ndnSIM project page v.2.4*. Dec. 2017. URL: https://ndnsim.net/2.4/index.html (visited on 05/13/2020).

[269] NDN Project Team. *ndn-CXX Overview*. Mar. 2017. URL: http://named-data.net/doc/ndn-cxx/current/README.html (visited on 05/13/2020).

[270] NDN Project Team. *Named Data Networking Forwarding Daemon*. 2017. URL: http://named-data.net/doc/NFD/current/overview.html (visited on 05/13/2020).

[271] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. "Recent Development and Applications of SUMO - Simulation of Urban MObility". In: *International Journal On Advances in Systems and Measurements* 5.3&4 (2012), pp. 128–138.

[272] OpenStreetMap Foundation. *OpenStreetMap project page*. 2020. URL: https://www.openstreetmap.org/ (visited on 05/13/2020).

[273] C. J.C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". In: *Data Mining and Knowledge Discovery* 2.2 (1998), pp. 121–167.

[274] R Lisový, M Sojka, and Z Hanzálek. *IEEE 802.11p Linux Kernel Implementation*. Tech. rep. 2014. URL: https://rtime.felk.cvut.cz/publications/public/ieee80211p_linux_2014_final_report.pdf (visited on 05/13/2020).

[275] European Telecommunications Standards Institute. *Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band (ETSI EN 302 663 V1.2.1)*. Tech. rep. 2013.

[276] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer, and F. Dressler. "Demo: OpenC2X - An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5". In: *Proceedings of the IEEE Vehicular Networking Conference, Demo Session*. 2016, pp. 1–2.

[277] D. Grewe, M. Wagner, S. Schildt, M. Arumaithurai, and H. Frey. "Caching-as-a-Service in Virtualized Caches for Information-Centric Connected Vehicle Environments". In: *Proceedings of the Vehicular Networking Conference (VNC)*. Dec. 2018.

[278] F. Baccelli and B. Blaszczyszyn. "Stochastic geometry and wireless networks - Volume I: Theory". In: *Foundations and Trends in Networking* 3 (2009), pp. 249–449.

[279] M. Haenggi. *Stochastic Geometry for Wireless Networks*. Cambridge University Press, Cambridge, 2012. DOI: `https://doi.org/10.1017/CBO9781139043816.002`.

[280] J.F. Coeurjolly, J. Moller, and R. Waagepetersen. "A Tutorial on Palm Distributions for Spatial Point Processes". In: *International Statistical Review* 85.3 (2016), pp. 404–420.

[281] Z. Tong, H. Lu, M. Haenggi, and C. Poellabauer. "A Stochastic Geometry Approach to the Modeling of DSRC for Vehicular Safety Communication". In: *IEEE Transactions on Intelligent Transportation Systems* 17.5 (2016), pp. 1448–1458.

[282] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa. "Supporting Cooperative Caching in Disruption Tolerant Networks". In: *Proceedings of the International Conference on Distributed Computing Systems*. 2011, pp. 151–161.

[283] E. Hyytiae, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott. "When does content float? Characterizing availability of anchored information in opportunistic content sharing". In: *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. 2011, pp. 3137–3145.

[284] C. Anastasiades, T. Schmid, J. Weber, and T. Braun. "Information-centric content retrieval for delay-tolerant networks". In: *Computer Networks* 107 (2016), pp. 194–207.

[285] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. "Web caching and Zipf-like distributions: evidence and implications". In: *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. Vol. 1. 1999, pp. 126–134.

[286] J. Benin, M. Nowatkowski, and H. Owen. "Vehicular Network simulation propagation loss model parameter standardization in ns-3 and beyond". In: *Proceedings of IEEE Southeastcon*. 2012, pp. 1–5.

[287] C. Scherb, D. Grewe, M. Wagner, and C. Tschudin. "Resolution Strategies for Networking the IoT at the Edge via Named Functions". In: *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*. Las Vegas, USA, Jan. 2018.

[288] D. Grewe, C. Marxer, C. Scherb, M. Wagner, and C. Tschudin. "A Network Stack for Computation-Centric Vehicular Networking". In: *Proceedings of the ACM Conference on Information Centric Networking (ICN)*. Boston, MA, USA, Sept. 2018.

[289] C. Scherb, B. Faludi, and C. Tschudin. "Execution state management in named function networking". In: *Proceedings of the IFIP Networking Conference (IFIP Networking) and Workshops*. 2017, pp. 1–6.

[290] pICN - Named Function Networking project. *Source Code Repository of the pICN project on GitHub*. 2020. URL: `https://github.com/cn-uofbasel/PiCN` (visited on 05/13/2020).

[291] Jason Andress. *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*. 2nd. Syngress Publishing, 2014. ISBN: 978-0-128-00812-6.

[292] G. J. Simmons. "Symmetric and Asymmetric Encryption". In: *ACM Comput. Surv.* 11.4 (1979), pp. 305–330.

[293] R. L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-key Cryptosystems". In: *Commun. ACM* 21.2 (1978), pp. 120–126.

[294] N. Koblitz. "Elliptic Curve Cryptosystems". In: *Mathematics of Computation* 48.177 (1987), pp. 203–1209.

[295] A. Shamir. "Identity-Based Cryptosystems and Signature Schemes". In: *Advances in Cryptology*. Springer Berlin Heidelberg, 1985, pp. 47–53.

[296] V. Goyal, O. Pandey, A. Sahai, and B. Waters. "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data". In: *Proceedings of the ACM Conference on Computer and Communications Security*. CCS '06. 2006, pp. 89–98.

[297] J. Bethencourt, A. Sahai, and B. Waters. "Ciphertext-Policy Attribute-Based Encryption". In: *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*. 2007, pp. 321–334.

[298] S. Mueller, S. Katzenbeisser, and C. Eckert. "Distributed Attribute-Based Encryption". In: *Proceedings of Information Security and Cryptology – ICISC*. Springer Berlin Heidelberg, 2009, pp. 20–36.

[299] S. Misra, R. Tourani, and N. E. Majd. "Secure Content Delivery in Information-centric Networks: Design, Implementation, and Analyses". In: *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*. 2013, pp. 73–78.

[300] R. S. da Silva and S. D. Zorzo. "An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges". In: *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC)*. 2015, pp. 128–133.

[301] M. Ion, J. Zhang, and E. M. Schooler. "Toward Content-centric Privacy in ICN: Attribute-based Encryption and Routing". In: *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*. 2013, pp. 39–40.

[302] B. Hamdane, A. Serhrouchni, and S. G. El Fatmi. "Access control enforcement in Named Data Networking". In: *Proceedings of the International Conference for Internet Technology and Secured Transactions*. 2013, pp. 576–581.

[303] G. Mauri and G. Verticale. "Up-to-date key retrieval for information centric networking". In: *Computer Networks* 112 (2017).

[304] Y. Yu, A. Afanasyev, J. Seedorf, Z. Zhang, and L. Zhang. "NDN DeLorean: An Authentication System for Data Archives in Named Data Networking". In: *Proceedings of the ACM Conference on Information-Centric Networking*. 2017, pp. 11–21.

[305] A. Lewko and B. Waters. "Decentralizing Attribute-Based Encryption". In: *Proceedings of the Advances in Cryptology – EUROCRYPT 2011*. Springer Berlin Heidelberg, 2011, pp. 568–588.

[306] M. Mosko, E. Uzun, and C. Wood. *IETF Internet Draft: CCNx Key Exchange Protocol Version 1.0*. Tech. rep. Mar. 2017. URL: `https://tools.ietf.org/html/draft-wood-icnrg-ccnxkeyexchange-02` (visited on 05/13/2020).

[307] J. Hur and D. K. Noh. "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems". In: *IEEE Transactions on Parallel and Distributed Systems* 22.7 (2011), pp. 1214–1221.

[308] S. Rafaeli and D. Hutchison. "A Survey of Key Management for Secure Group Communication". In: *ACM Comput. Surv.* 35.3 (2003), pp. 309–329.

[309] M. Raya and J.P. Hubaux. “The Security of Vehicular Ad Hoc Networks”. In: *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks*. 2005, pp. 11–21.

[310] D. Dolev and A. Yao. “On the security of public key protocols”. In: *IEEE Transactions on Information Theory* 29.2 (1983).

[311] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott. “Security and Privacy in Device-to-Device (D2D) Communication: A Review”. In: *IEEE Communications Surveys Tutorials* 19.2 (2017), pp. 1054–1079.

[312] NSF Named Data Networking project. *Documentation of the ndn-cxx library: Signed Interest*. Tech. rep. 2018. URL: `https://named-data.net/doc/ndn-cxx/current/specs/signed-interest.html` (visited on 05/13/2020).

[313] A. Compagno, M. Conti, P. Gasti, and G. Tsudik. “Poseidon: Mitigating interest flooding DDoS attacks in Named Data Networking”. In: *Proceedings of the IEEE Conference on Local Computer Networks*. 2013, pp. 630–638.

[314] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. “DoS and DDoS in Named Data Networking”. In: *Proceedings of the International Conference on Computer Communication and Networks (ICCCN)*. 2013, pp. 1–7.

[315] S. Gueron. “Memory Encryption for General-Purpose Processors”. In: *IEEE Security & Privacy* 14.6 (2016), pp. 54–62.

[316] Y. Yu, A. Afanasyev, D. Clark, kc Claffy, V. Jacobson, and L. Zhang. “Schematizing Trust in Named Data Networking”. In: *Proceedings of the ACM Conference on Information-Centric Networking*. 2015, pp. 177–186.

[317] C. Marxer and C. F. Tschudin. “Schematized Access Control for Data Cubes and Trees”. In: *Proceedings of the ACM Conference on Information-Centric Networking*. 2017, pp. 170–175.

[318] D. Oran. *Considerations in the development of a QoS Architecture for CCNx-like ICN protocols v04*. Tech. rep. Dec. 2019. URL: `https://tools.ietf.org/html/draft-oran-icnrg-qosarch-04` (visited on 05/13/2020).

[319] D. Grewe, M. Wagner, S. Schildt, A. Nordmann, and J. Laverman. “Towards Semantic Object Discovery for Vehicular Named Data Networks”. In: *Proceedings of the 87th IEEE Vehicular Technology Conference (VTC Spring)*. June 2018, pp. 1–5. DOI: `10.1109/VTCSpring.2018.8417783`.