An Information-Centric Approach to Quality of Service in a Highly Dynamic Edge Environment

by

Jessica McCarthy

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Doctor of Philosophy

University of Dublin, Trinity College

January 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

I, the undersigned, agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent / do not consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Jessica McCarthy

May 24, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Jessica McCarthy

May 24, 2019

Acknowledgments

There are a number of people without whom this thesis would not have been possible,and to whom I am greatly indebted and dedicate this thesis. First and foremost, I would like to thank my supervisor, Prof. Siobhán Clarke thank you for you endless support and encouragement and above all believing in me. I have learned so much from you over the past few years. Your talent, scientific rigour and your attention to detail continue to be a inspiration to me. Thank you to the leadership within Intel Labs for giving me the opportunity, scope and financial support to conduct this research and ensure it became an integral part of Intel Labs Europe ”Dependable Cyber Physical Systems” research agenda. Thank you to my mentor and friend Dr Vida Ilderem, you gave me the encouragement and confidence to believe I could do this and for continuing to support me as I take the next steps in my career. To my parents thank you for instilling the value of learning, and for your love and support even through difficult times. Thank you to my dear family for your constant encouragement, love, support and patience during this process, you are my rocks who ground me and give me the reality check I often needed during this process. Finally to anyone who has completed a PhD whilst working full time, I salute you, and to those in the process, you will get there, believe in yourself, a little progress every day adds up to big results.

Jessica McCarthy

University of Dublin, Trinity College
January 2019

Abstract

There is a fundamental paradigm shift in how today's communications networks are being used to deliver data. This has emerged from the proliferation of intelligent Internet of Things devices and corresponding volumes of data being generated by these devices. This data explosion is not just an increase in the sheer volume of the data. Data flows are dramatically different, with machines creating and consuming the data like never before. In addition, connectivity to the network is often heterogeneous and opportunistic.

The Internet Protocol (IP) is the universally-accepted networking protocol that has served us very well, transforming networks into the age of information. However, IP was designed with the sole purpose of connecting two machines over a fixed wired network. The requirements for the network when IP was designed were vastly different from the requirements we now have. Demand for heterogeneous content is rising sharply and more efficient content distribution is required to manage the corresponding traffic, and return requested content with acceptable Quality of Service (QoS). QoS provisioning and fulfilment is dependent on content recognition, which is not supported natively in IP host-centric networks. In IP, overlay mechanisms are used to capture content information to be used for QoS fulfilment, but this approach is inefficient given current network requirements. Information-Centric Networking (ICN) provides content-based delivery, but QoS concerns are not sufficiently addressed.

The vehicular application domain is used in this work as an exemplar of one with complex networking requirements. Modern vehicles include significant technology that could be exploited to improve safety and road efficiency, but making use of these technologies requires time-sensitive information delivery between vehicles. The capabilities of the network are of critical importance to provide the required quality of service of data delivery. This thesis presents a QoS-aware Information-Centric Network for vehicular applications. In particular, the work extends the Named-Data Network (NDN), a variant of ICN, focusing on data delivery deadline awareness. The new algorithms classify the priority of requests, with associated QoS requirements by encoding QoS information into the interest request packets and corresponding data reply packets, and extending the routing algorithm to use multihop forwarding to efficiently request and receive the requested content without pre-building routes. The work also explores extending network traffic control mechanisms at the data link layer, to assess the potential impact on network congestion management.

Evaluations have been carried out using extensive simulation, in particular using a combination of the ndnSIM and SUMO simulators. Deadline aware success rate and packet success rate are both measured under different network densities, vehicle speeds, proportions of vehicles in the environment acting as content producers, and experiment durations. The QoS-aware ICN algorithms are assessed against four baselines: UDP IP (with both DSRC and WiFi communication channels), and basic NDN (DSRC and WiFi).

The results of this study demonstrate that the QoS-aware approach generally achieves higher success rates at delivering different packet priority types within their deadlines, relative to the baselines. In addition, these success rates are achieved with fewer packet re-transmissions. However, adding QoS-awareness to network traffic control at the data link layer is less impactful.

Contents

List of Tables

List of Figures

List of Abbreviations

ICN	Information Centric Networking.
NDN	Named Data Networking.
VANETs	Vehicular Ad-hoc Networks.
CDN	Content Distribution Network.
MEC	Mobile Edge Computing.
MANETs	Mobile Ad-hoc Networks.
SDN	Software Defined Network.
QoS	Quality of Service.
MAC	Medium Access Control.
DSRC	Dedicated Short Range Communications.
CPS	Cyber Physical Systems.
QoSA-ICN	Quality of Service Aware Information Centric Networking.
QoSA TC-ICN	Quality of Service Aware Information Centric Networking with Traffic Control.

List of Publications

This thesis is based on the following peer-reviewed publications:

1. McCarthy J, Kuppuudaiyar P , Loomba R , Clarke S, 2019. QoSA-ICN: An Information-Centric Approach to Deadline-Aware Data Delivery in a Highly Dynamic Edge Environment. Transactions on Networking, Submitted Feb 2019
2. McCarthy J et al. INFORMATION CENTRIC NETWORK PACKET TRANSMISSION CONTROL, PROVISIONAL Patent Application filed, Jan 2019
3. H. Moustafa, E. M. Schooler and J. McCarthy, ”Reverse CDN in Fog Computing: The lifecycle of video data in connected and autonomous vehicles,” 2017 IEEE Fog World Congress (FWC), Santa Clara, CA, 2017, pp. 1-5. doi: 10.1109/FWC.2017.8368540

Chapter 1

Introduction

Modern cyber-physical systems (CPS) [Lee, 2008] include significant technology that could be exploited to improve safety and efficiency. For example, in the transport domain, communication technologies could support an advance hazard warning system while monitoring movements of vehicles, or in the manufacturing domain, intelligent systems could optimise production. Making use of these technologies often requires the delivery of time-sensitive information between the elements of the system. The capabilities of the network to provide the required quality of service of data between these elements is of critical importance. With the proliferation of intelligent Internet of Things (IoT) devices in our lives and in turn, a tidal wave of data being generated by these devices, there has been a fundamental paradigm shift in how today's networks are being used. The current architecture of the Internet is based on a host-centric communication model, but content-based traffic, and the demand for such content irrespective of its location, is rising sharply. Content-based traffic is growing not only in terms of volume but also in terms of heterogeneity and granularity [Al-Naday et al., 2014]. This has motivated the development of network architectures that focus on content and user interests to achieve more efficient content distribution. Information Centric Networking (ICN) is a next-generation network technology designed for this purpose, which looks to effectively provide highly scalable and efficient distribution of content. The principal idea behind ICN is to allow users to request data in a content-centric manner without concern for the physical location of the data. A data unit is requested, routed and delivered via its name rather than its address, making data migration easier under conditions of device mobility and intermittent connectivity. Secondly, ICN supports distributed content caching in the network, replication and facilitates one-to-many communications while offering native content identification in the network. ICN benefits are focused around improving data dissemination with regard to data and bandwidth demands and improving resiliency in challenging communication scenarios such as disconnections, disruptions or network black spots. In these areas, ICN has the potential to outperform IP. According to IDC it is expected that by 2019, 40% of IoT data will be created, stored, processed and analysed at the edge of the network [MacGillivray et al., 2016]. Latency issues will therefore be even more pronounced.Network latency is the time delay between sending information from one point to the next. This raises the question of whether the edge of the network will be able to provide the communication requirements needed to process this explosion of data. Coupled with this, cellular networks, which provide much of the communication technologies at the edge of the network, are fragmented and limited in channel size and serious questions have been raised as to its long term sustainability in its current form [Britz, 2014]. The need for efficient Quality of Service (QoS) [Al-Naday et al., 2014] provisioning for such environments is paramount and remains one of the key challenges facing current and future internet architectures. Effective QoS provisioning is highly dependent on the ability to identify content to fulfill expected service requirement. This is a complex task for today's host-centric IP networks as content identification is not a native feature, which in turn leads to many workarounds and fixes in place to achieve QoS provision for the network. ICN natively offers content identification in the network. However, QoS delivery has not been widely explored within ICN to date [Amadeo et al., 2016]. This work has been inspired by the NSF/Intel partnership on Information-Centric Networking in Wireless Edge Networks (ICN-WEN) [Intel and NSF, 2018] and the Software Infrastructure working group, which is part of the Open Fog Consortium [OCF, 2018].

1.1 Motivation and Challenges

Data has overtaken oil as the world's most valuable commodity. Yesterday's networks are not aligned with today's data creation, usage or movement realities (e.g., massive sensor data must now go upstream to the cloud), as they were not designed for this purpose. Information-Centric Networking (ICN) is an approach to transform networks by routing data by name (vs. address) and securing data independently of the underlying connection. Data is requested by

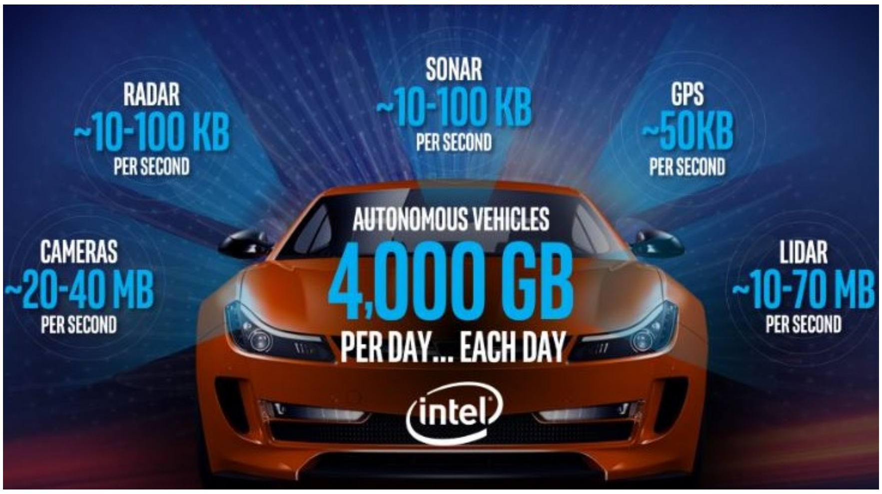


Figure 1.1: Flood of data in Autonomous Vehicles - source: Intel

name and delivered from wherever it resides. This approach has the potential to yield significant benefit for 5G/5G+ applications, like massive IoT networks, ADAS, and AR/VR.

1.1.1 Motivating scenario

According to Koptez [Kopetz, 2011], within real time systems, “A deadline is the instant when a result should/must be produced”. Deadlines can be either hard, firm or soft. If severe consequences occur when a deadline is missed it is called hard, these hard deadline are seen in hard real time or safety critical systems such as industrial process control systems or automotive safety critical systems. These deadlines must be guaranteed. A deadline is said to be firm if the results produced cease to be useful as soon as the deadline expires, but consequences of not meeting the deadline are not very severe [Shin and Ramanathan, 1994].Koptez goes on to point out that if a deadline passes and the result which the deadline produces is still useful to the system, the deadline is referred to as soft.

The vehicular application domain motivates this work due to its complex networking requirements for reliable data delivery (see Intel graphic, Fig. 1.1). In order to take full advantage of these technologies, vehicles need to be able to share information in a time-sensitive, reliable and consistent manner. The challenges for the network to perform such data-sharing tasks are exacerbated by the volume of data, which is predicted to be generated by these connected vehicles. Currently, a single autonomous vehicle produces the same amount of data as 2,666 users (i.e., people), daily [INTEL, 2019]

Users’ data usage patterns will continue to rise, but people are no longer

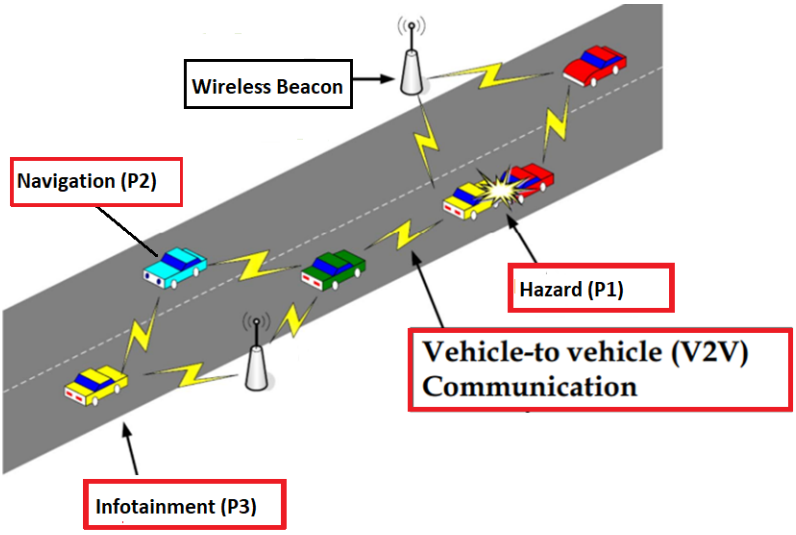


Figure 1.2: Motivating Scenario

the main generators of data. It is predicted that 13.7 billion Machine to Machine (M2M) devices will be connected by 2021 [Cisco, 2017]. 40% of the data, which these machines produce, will be stored, processed or actuated at the edge of the network [MacGillivray et al., 2016]. Coupled with this, it is predicted that 63% of ALL global IP traffic will be generated from wireless and mobile devices by 2021 [networking Index, 2016]. It is evident that the capabilities of the network are of critical importance to provide the required quality-of-service of data delivery. The capabilities of in-vehicle technology are such that there is significant potential for many different kinds of applications, with corresponding differences in quality of service requirements. For example, vehicle occupants are likely to prioritise road-side hazard information over navigation route guidance, though may also prefer navigation route guidance over infotainment or retail notifications (see Fig. 1.2). Networks do not currently distinguish between such different needs, which could result in a driver missing out on hazard warning information while the children in the back seat happily watch their favourite videos. In the motivating scenario for this thesis, there are potentially multiple applications in the vehicle, vying for network attention [Moustafa et al., 2017]. The motivating use case in this thesis has been inspired by the Mobileeye driver assistance system [Mobileeye, 2018]. This in car technology provides passive alerting concepts(warning messages) of potential dangers such as dangerous road conditions, forward or rear end collision advance warnings so the driver/car can take action to avoid this hazard. These warnings create an visual or audible alert often delivered via an in car infotainment system. Currently this system generates warning alerts of a collision up to 2.7 seconds in advance. The motivating scenario for this thesis assumes that the car is fitted with a WiFi/DSRC radio and sensor data from the cars on-board sensors such as gps, wipers, cameras, braking system is available to the application. It is also assumed that the car is not fully autonomous. This thesis looks to reduces the need for drivers to perform manual actions to report on dangerous road conditions. Instead the events are created automatically in conjunction with the data from the vehicle sensors to offer these warning alerts at low latency. With the overall aim to deliver these messages in milliseconds rather than seconds. The vehicles in the network subscribe to this service and adopt the role of either consumer, producer or forwarder of information and adopt the service QoS levels. The service aims to provides the mechanism to provide quality of service to the application with varying priorities (P1-P3) and round trip time (RTT) limit for the time-sensitive warnings to enable a more dependable Information centric approach for deadline sensitive network traffic. Examples of prioritised information types are as follows:

- Priority 1(P1) Hazard warning, broken down vehicle, dangerous road conditions, adverse weather conditions.
- Priority 2 (P2) Navigation Nudges, weather information, speed warning.
- Priority 3 (P3) Retail subscriptions, media streaming (video streaming is out of scope in this scenario)

The thesis does not provide hard real time deadline guarantees instead it looks to understand by implementing firm and soft real time deadlines can a higher percentage rate of QoS adherence be achieved for a dynamic edge based network compared to existing baselines. Uniform agreement on priority is based on the the assumption that a network service provider sets the priority and associated cost of data delivery based on that priority for all application using this service.

1.1.2 Challenges

The viability of delivering time-sensitive information in highly hostile dynamic environments e.g., vehicular networks, in a manner that is deadline-aware, is very challenging for the following reasons:

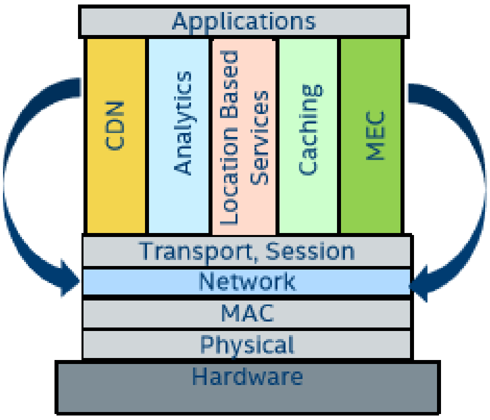


Figure 1.3: Bringing application layer functionality to the network layer - source: Intel

CH1 The dynamic, mobile nature of a vehicular environment leads to un-predictable network topology/patterns [Viriyasitavat et al., 2015].

CH2 Communication links have a short duration (average less than 6 secs in 97 percent of cases) [Toor et al., 2008].

CH3 The occurrence of communication scenarios such as disconnections, disruptions or network black spots is likely, but difficult to predict.

CH4 The network layer is not sufficiently optimised to support QoS-aware data delivery at the network edge.

The network layer [Zimmermann, 1980, Fall and Stevens, 2011]provides the common language for different networks to interact and transfer data. From a networking perspective, only the network layer is standardised such that the layers above and below can be developed independently. As such, networks have been evolving to better support data. However, many solutions to data management have been in the application layer. For instance, Content Distribution Network (CDN) [Dilley et al., 2002] are proprietary application-layer solutions that strategically distribute content over the network for faster and easier access. Similarly, application-layer solutions like Mobile Edge Computing (MEC) [Hu et al., 2015] have been developed to place compute and services closer to the edge. Instead of developing application-layer solutions to improve the efficiency of the networking layer, ICN aims to bring these constructs and patterns down to the network layer, where they are made available to all the nodes in the entire network (in network), removing the dependency on the application layer (see Fig. 1.3).

1.2 Existing solutions

ICN offers native content identification, which has the promise to be very effective when implementing QoS provisioning. Content identification could be implemented in a clean-slate ICN implementation or as an additional layer in an ICN/IP overlay approach. However, there is no common functional architectural implementation of content identification within ICN [Xylomenos et al., 2014]. The approaches vary from hierarchical content naming within Named Data Networking (NDN) [Wang et al., 2010], to COMET [COMET, 2010], which uses a block-based content identification structure, while PURSUIT [PURSUIT, 2010] uses an information management plane. The current QoS offerings within ICN also vary in approach and focus. Al-Naday et al., 2014 use QoS naming prefixes built on the PURSUIT architecture. Chu et al., 2016 employ optimised caching polices where data popularity is taken into consideration when making decisions on content placement. Wang et al., 2018 focus on energy efficiency within the network, with the approach tightly coupled to the communication network links. Tsilopoulos, 2011 explore not only the need for effective naming but also a clear understanding on specifics about the nature of the traffic. It is noted that all of these solutions' primary focus is on effective Interest packet delivery, which is only part of the solution needed. QoS-aware data delivery has not been widely explored within ICN, to date. Currently, ICN supports content delivery, though there has been initial discussion at IRTF [IRT, 2019] and ICNRG [ICN, 2019] recognizing the gaps in its support for QoS requirements for content delivery. In the latest draft, Jangam et al. propose using a differentiated services (DiffServ) approach [Jangam et al., 2018]. This solution has not been implemented to date and therefore no evaluation on the proposed approach has been published. In order to provide true data deadline awareness within ICN, both the interest and data paths must be considered. This is even more prevalent in a dynamic wireless environment where it cannot be assumed that the data packet will be returned on the same path as that travelled by interest packet. In addition, emerging applications such as autonomous vehicles, augmented reality or industrial robotics, which are most expected to benefit from the ICN approach, might contain sensitive control loops that are highly sensitive not only to delay, but also to the delivery of data in correct sequential order, with no errors.

1.2.1 Research Questions

This goal of this thesis is to solve the above challenges by exploring the question of how to enable an information-centric approach to deadline-aware data delivery in a highly dynamic edge environment. This question can be decomposed into:

1. **[RQ1]** *To what extent can the delivery of time-sensitive data in dynamic environments be improved by employing an information-centric, QoS-aware data delivery mechanism, compared to existing baselines?*

2. **[RQ2]** *To what extent can the delivery of time-sensitive data in dynamic environments be improved by adding QoS awareness to network traffic control mechanisms, compared to existing baselines?*

1.3 Thesis approach

This thesis presents a novel model for consideration of Quality of Service in information centric networks (hereafter called Quality of Service Aware Information Centric Networking (QoSA-ICN)), designed to provide an information-centric approach to deadline-aware data delivery at the network edge. The thesis explores the possibility of natively including QoS-awareness to ICN, and has the following research objectives:

[RO1] Design information centric QoS-aware data propagation algorithms to deliver time-critical data requests for optimized service assurance in dynamic Cyber Physical Systems (CPS) at the network edge.

[RO2] Design network traffic control modules to control data transmission and process the packet differently within queues at the data link layer with respect to its QoS requirements.

[RO3] Extend an existing ICN platform with the new QoS-aware data propagation algorithms.

[RO4] Design and implement evaluations of the design using appropriate simulators.

[RO5] Analyse, quantify and validate the impact of the implementations i.e., Deadline awareness improvement, network traffic control and overall algorithm efficiency.

Assumptions This thesis makes the following assumptions relating to the environment in which requests for data are sent:

1. The operating environment of a vehicular network is highly dynamic. Vehicle mobility changes operating conditions, and may lead to requests for data not being fulfilled with the desired quality of service.
2. The thesis focuses solely on Vehicle-to-Vehicle communication, and assumes that there is a communication infrastructure in place to support this.
3. The actors (vechicles) within this network have a shared view of message priority and associated deadlines.
4. The network and associated resources are dedicated. The thesis does not take into account a real-world scenario where network resources are shared.
5. It is assumed that vehicles' time are synchronised globally. This is implemented using a combination of UTC (Coordinated Universal Time) clock, time stamping of packets and allows for the use of on-board GPS devices.
6. While the solution allows for location based naming, this is not the focus of the thesis, instead focus is placed on delivery mechanism for priority based traffic.
7. This research does not assume that the global Internet will be solely ICN-enabled in future. The research assumes that IP host connectivity will remain prevalent. In order to create a fair evaluation, this research will look at existing IP-based technologies as baselines for evaluation in order to highlight the advantages and disadvantages of an information-centric approach to QoS aware fulfillment.

Hypothesis This thesis investigates how to make ICN QoS-aware. It frames its hypothesis as follows: Advancement in data-centric propagation models for dynamic deadline aware forwarding in Information-Centric Networking will lead to insight-driven QoS provisioning for ICN applications.

Basic Idea In short, the goal of this research is to explore, identify, implement and evaluate a suitable design for Information-Centric Networks to support time-sensitive applications at the network's edge. The research aims at refining and extending our current understanding of how applications, which reside on edge-based wireless networks, can benefit from Information-Centric Networking.

The approach taken is to introduce QoS-aware multi-hop packet forwarding for time-sensitive applications on vehicular networks, through assigning different priorities to different kinds of information, e.g., assign a P1 priority to road side hazards, P2 priority to navigation route guidance, and P3 to infotainment/context aware retail notification based of users service subscription preferences. The new algorithms classify the priority of requests P1-P3(P1 highest priority with shortest time to live value) relative to the associated QoS requirements, and encodes this QoS information into the native interest request packets and corresponding data reply packets. The native routing algorithm (i.e., best routing) is also extended to use multi-hop forwarding to efficiently request and receive the requested content, without pre-building routes. The work also explores extending the network layer traffic control mechanisms, to assess the potential impact on network congestion management. The overall objective of Quality of Service Aware Information Centric Networking with Traffic Control (QoSA-TC-ICN) is to prioritise high priority packets for further transmission and drop the low priority or the expired prioritised packets in order to reduce the congestion in the network resulting in better network utilisation. The congestion control mechanism at the data link layer intercepts the packets leaving the NDN forwarding stack to re-order the packets into dedicated priority queues. With the overall aim of ensuring prioritised transmission of packets that contain a QoS deadline by observing the packet time to live value. QoSA-TC-ICN implementation can be enabled (QoSA-TC-ICN) or disabled (QoSA-ICN) in the solution, when disabled QoS aware traffic management is not invoked. The algorithms have been implemented on top of the named data networking (NDN) ICN platform. NDN was selected because it is both open source, and has a suitable simulator, i.e., ndnSIM [Mastorakis et al., 2017]. The work is demonstrated and evaluated in the context of a simulated wireless edge network, to identify the solution's benefits and possible drawbacks. In particular, a combination of ndnSIM and SUMO simulators were used. ndnSIM is an open- source NDN simulator based on the NS-3 simulation framework [NS-3, 2019]. NS-3 is a widely-used, discrete-event network simulator for Internet systems, also open source, and licensed under the GNU GPLv2 license. ndnSIM is publicly available for research and development. The Simulator for Urban Mobility (SUMO) [Krajzewicz et al., 2012] was used, together with OpenStreetMap [Haklay and Weber, 2008], to generate realistic vehicle scenarios for our evaluations. SUMO is an open source, highly portable and continuous road traffic simulation package designed to build large road networks. SUMO was used to generate mobility traces for all the scenarios. Delivery success rate and packet re-transmission rate are both measured under different network densities, vehicle speeds, proportions of vehicles in the environment acting as content producers, and experiment durations. The QoS-aware ICN algorithms are assessed against four baselines: UDP IP (with both DSRC and wifi communication channels), and basic NDN (DSRC and WiFi).

1.4 Thesis Contribution

This thesis investigates an information-centric approach to supporting deadline-aware delivery of data in highly dynamic, distributed, decentralized infrastructure-less cyber physical systems. The main contribution is the design of a novel QoS-aware data delivery mechanism that extends information-centric technology, in particular the Named-Data Network (NDN), a variant of ICN. The work focuses on data delivery deadline awareness for vehicle applications with varying time sensitive data delivery requirements. Various and varying parameters are considered, such as road length and capacity, vehicle density, mobility rates, rate of content requesters and producers present in the environment, and communication medium. Each of the research questions and research objectives presented in the above section are instrumental in building these algorithms and have been addressed in detail in Chapter 3, Chapter 4, Chapter 5. The section extracts and summarises the high level contributions to the body of knowledge that have been made in this thesis.

1. **In-network support for Quality of Service:** Effective QoS provisioning is highly dependent on the ability to identify content to fulfill expected service requirement. This is a complex task for today's host-centric IP networks, as content identification is not a native feature. This leads to many workarounds and fixes to achieve QoS provisioning for the network. ICN natively offers data naming to allow for content identification in the network via naming, but QoS delivery in ICN has not been widely explored, to date. This thesis defines new algorithms to classify the priority of requests with associated QoS requirements by encoding QoS information into both the interest request packets and corresponding data reply packets (RQ1,RO1,RO3,RO4)

2. **Network layer traffic control:** The current architectures of ICN vari-

ants (e.g., NDN) focus on mechanisms for data forwarding for name-based content delivery only within the network layer, and adopt the default network traffic control mechanisms in the data link layer. NDN's stateful forwarding plane enables routers to control congestion at each hop, by either dropping Interest packets or diverting them to alternative paths. However, once the packet leaves the network layer, there is no mechanism in the data link layer to process the packet differently, with respect to QoS or deadline awareness. Without a QoS-aware congestion control mechanism at the MAC Layer which is a sub layer of the data link layer, there is no possibility of enforcing QoS policies in the management of network bandwidth/congestion, which might result in the random dropping of time-critical/ priority packets. These behaviors would lead to poor quality of service, which turn means that the network is incapable of delivering deadline awareness. This thesis reports on a design to allow for a cross layer approach (network & MAC) to include traffic control, which can be tailored by QoS requirements (RQ2, RO2, RO3 ,RO4)

1.5 Thesis Structure

State of the art Chapter 2 analyses how state of the art approaches to achieve Quality of Service met the challenges of delivering content for time sensitive applications in dynamic edge based environments. In particular the study explores a) the most common approaches to data delivery and dissemination b)how an information centric approach aims to optimize data delivery c) how this information centric approach to data delivery has been applied to dynamic environments d) how quality of service for data delivery in traditional network is realised e) how the concept differs when an information centric approach is taken

Design Chapter 3 returns to the challenges of data delivery in a dynamic edge network as outlined in Chapter 1 and describes the design objectives and system model of this thesis. Thereafter the chapter explains how the proposed algorithms for deadline aware data delivery address the research objectives in detail.

Implementation Chapter4 describes the implementation of the design of QoSA ICN which was called out in Chapter3. A C++ implementation integrated the QOSA algorithms with the network simulator ndnsim. It is realised as an extension to the ndnsim simulation platform. The chapter also covers the data pipeline which was written in python to carry out the extensive evaluation of the new algorithms against the baseline algorithms. The chapter also covers the experimental set up needed to carry out these evaluations.

Evaluation Chapter 5 evaluates how the QOSA-ICN algorithms fulfil the identified challenges and research questions by comparing to baseline solutions for the state of the art. Firstly it describes the experimental setup of the simulation-based study. The second part of the chapter presents and analyses the results showing that the proposed QoSA-ICN algorithms are a suitable alternative to achieve deadline aware data delivery for dynamic edge environments.

Conclusion Chapter 6 presents the thesis achievements, summarises this work and presents some open issues for further research.

Chapter 2

State of the art

12% of today's vehicles are connected to the Internet, with 34.3% projected for 2021 [STATISTA, 2018]. It is estimated that by 2020, a single car will generate approximately 3000x more data than a single person [INTEL, 2019].

In the context of automated and autonomous driving, a car is highly dependent on information about its environment and surroundings, which is now readily available from a myriad of sources. Considering the availability of all these new data, there is now an opportunity to extend the scope of the vehicular domain to develop new information-centric applications that are time sensitive in nature, for example, an application sharing information regarding road traffic hazards or traffic diversions to all vehicles within a certain geographical area. Robust QoS provisioning is essential to fulfill the requirements of such applications. According to current literature, existing QoS models fall short in supporting such demands [Al-Naday et al., 2014]. The literature presents empirical evidence that highlights an over-reliance on application layer resources, which often cannot respond effectively to spikes in network traffic (e.g., from flash crowds) for both cached and un-predicted dynamic requests [Wendell and Freedman, 2011]. Common requirements for vehicle applications, such as time-sensitive data delivery, have not been fully thought out in the ICN architecture, and extensions are needed to provide such QoS support. It is important to understand how current content delivery offerings manage and disseminate content in terms of QoS and mobility in order to highlight existing gaps.The following section of the thesis will looks and these in greater detail.

2.1 Content Delivery Networks

Today, Content Delivery Networks (CDN) [Dilley et al., 2002] are a common approach to managing and disseminating content. It is predicted that CDN traffic will carry 71% of all Internet traffic by 2021 [Cisco, 2017]. CDNs consist of a large number of distributed proxy servers which are deployed in multiple data centres. The CDN has two fundamental principles: replica allocation and content delivery. In order to allow for timely access to content, the CDN distributes content replicas to hosts in close proximity to end nodes. According to the literature, when designing a CDN, consideration must be given to the selection of the most optimum replica locations, the discovery of the most effective replica and the mechanism for content delivery itself [Salkuyeh and Abolhassani, 2016]. By having these replicas, CDNs offer caching closer to the end point, which allows for faster content delivery but this very much depends on the replica. Data flows from the content source to the content requester in a one-to-many fashion through the CDN. This is a one way flow of information. In a vehicular network, content is frequently generated at edge devices (on board sensors within vehicles) and these dynamic data flows follow a reverse data flow pattern. This is different to CDNs and is the opposite to how traditional networks have been provisioned to deliver data [Moustafa et al., 2017].

For communication and data dissemination purposes, vehicles form Vehicle Ad Hoc Networks (VANETs) that consist of vehicles with wireless capabilities that dynamically form a network with other vehicles, supporting vehicle to vehicle communication (V2V) [Piro et al., 2014]. They also communicate with city infrastructure such as road side units (RSUs) or cloud-based maps, weather or traffic information, all of which provide the car and driver with new on-road information and services. This is referred to as vehicle to infrastructure communication (V2I) [Su et al., 2017]. Using a VANET to deliver content is not a novel concept. However effective content storage methods, and managing/forwarding content, are challenges for such networks due to the type of content these networks now have to process and deliver. Content type has shifted in nature from simple messages to context-rich information such as real-time traffic information [Zaidi et al., 2017], which may include audio and maps. Traditional networking solutions are often challenged to support such a heterogeneous range of applications within VANETs, due to factors such as short communication link duration (avg < 6 secs), link disconnections, network disruption or black spots.

According to TalebiFard et al., 2015, VANETs present some unique features, which in turn create challenges for reliable content dissemination such as "fast changing topology, short-lived intermittent connectivity and a wide set of application with heterogeneous requirements often in harsh propagation conditions" [TalebiFard et al., 2015]. TalebiFard also points to the fact that these challenges are exacerbated by the heavy reliance on end-to-end host-centric networking i.e., the TCP/IP protocol stack.

Wireless Access in Vehicular Environments (WAVE) [IEEE, 2010] [Jiang and Delgrossi, 2008], also known as Dedicated Short Range Communications (DSRC) or the IEEE 802.11p protocol stack, was developed to overcome these specific challenges through the use of the WAVE Short Message Protocol (WSMP). This does not rely on IP for safety-critical messaging. WAVE, which is implemented at the Medium Access Control (MAC) layer, allows for communication in harsh communication environments with the aim to reduce access delay and signal overhead. Currently, the WAVE stack is used for the exchange of all data in VANETs and this brings with it the reliance on traditional networking protocols such as TCP/UDP/IPv6. With the introduction of IPv6, there have been some challenges with the "WAVE link model" as it does not provide for a direct link with the IPv6 link model [Céspedes et al., 2013] [Baccelli et al., 2010]. In today's host-based networking offerings, there have been various attempts to manage content effectively, including peer to peer (P2P) architectures, and cloud-based platforms, including Software Defined Network (SDN) and Content Delivery Networks (CDNs) [Schollmeier, 2001] [Piro et al., 2014] [Sadiku et al., 2014]. Content dissemination within vehicles is dynamic and constant, and the network must be able to support such constant data flow in an efficient manner. SDN and CDNs have applied their models at the application layer, which is not scalable. In contrast, Information-Centric Networking (ICN) was designed specifically for the network layer and to be efficient at data delivery. ICN has therefore been selected to anchor the research for this thesis. This state of the art is structured to first provide an introduction and analysis of ICN, in particular its Named Data Networking (NDN) implementation. The motivating scenario for this work is in the V2V domain, and so analysis of existing work on ICN's application to Vehicular Ad-hoc Networks (VANETs) and Mobile Ad-hoc Networks (MANETs) follows. Finally, this chapter analyses general approaches to Quality of Service within IP networks, mobile networks and ICN, finishing with a discussion of QoS-aware traffic control.

2.2 Information-Centric Networking

Information-Centric Networking (ICN) is a next-generation network that aims to provide highly scalable and efficient distribution of content [Vasilakos et al., 2015]. ICN networks offer many native advantages compared to IP, such as ease of mobility, faster content delivery, and support for multi-homing. The principal idea behind ICN is to allow users to request data in a content-centric manner, without concern for the physical location of the data. A data unit is requested, routed and delivered via its name rather than its address as seen in Fig. 2.1, making data migration easier under conditions of device mobility and intermittent connectivity.

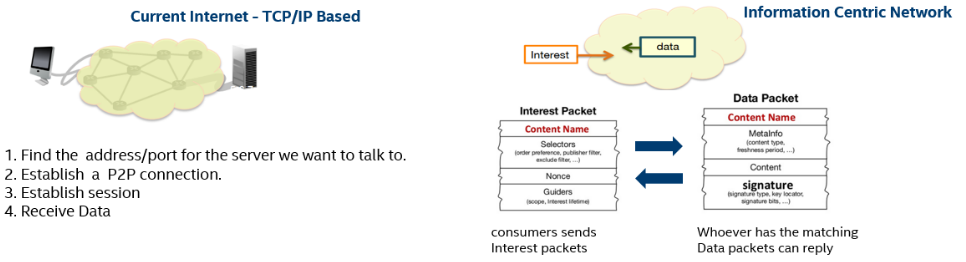


Figure 2.1: Content Delivery TCP-IP versus ICN

Research into the field of ICN has been growing sharply, with ICN data objects such as web pages, videos, documents or other information directly referred to by name instead of being retrieved through addressable endpoints. Data becomes independent from location, application, storage and means of transportation, enabling in-network caching and replication. Naming and name-based routing schemes are core elements to ICN.

As displayed in Fig.2.2, some ICN architectures aim to evolve the role of the "thin waist" such that packets can name data objects rather than host end points. Within ICN, content delivery is receiver driven. A receiver expresses interest in content, and the entire network is responsible for locating the information, identifying a path and delivering the information requested. ICN aims to improve efficiencies with regards to latency for information demand, and provide better robustness in challenging communication scenarios such as network black spots, where devices may be offline or network congestion or interference exists. The goal of ICN is to provide a network infrastructure service that is

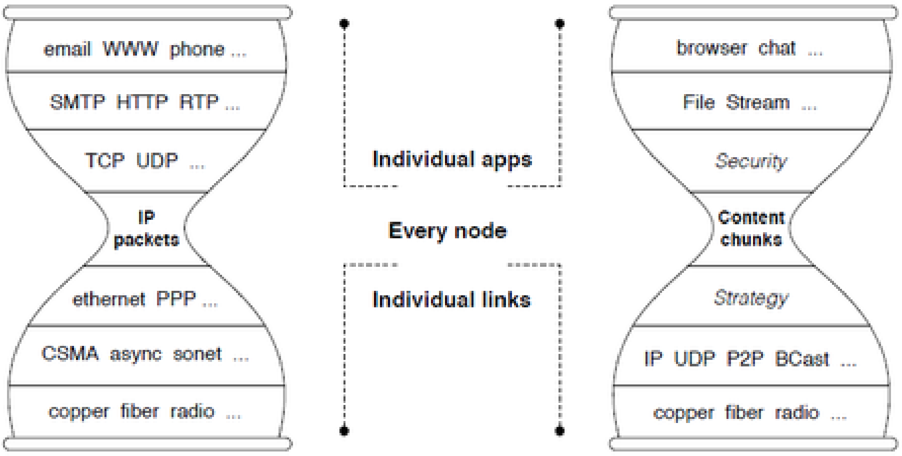


Figure 2.2: IP architecture vs. Named Data Networking(NDN architecture)

better suited to today's prevalent Internet usage model (in particular, content distribution and mobility) and more resilient to disruptions and failures. ICN concepts can be applied at different layers of the protocol stack. It has been a common approach to apply ICN name-based data access on top of existing IP networks as an overlay, thereby providing resource naming and in-network caching. However, overlay solutions, while they offer the advantages of traditional packet switches networks, also inherit the problems of the packet switching network and introduce complicated architecture layering, which leads to an increase in the processing time required for data routing and has a negative impact on latency. According to Liu et al., 2017 overlay approaches within mobile networks do not take advantage of wireless broadcasting capabilities for data propagation and also do not take advantage of in-network caching, which ICN offers natively. ICN has also been applied at the packet level of a network, but conceptually is very different in many aspects to traditional networks, e.g., data naming and forwarding. In the various ICN implementations, a key set of ICN functionalities exist, but they vary in some implementation decisions . Universally, ICN solutions focus on the content requested by the user. They do not focus on who is providing the content or its location. When a user makes a content request, the ICN architecture takes care of routing-by-name the content request towards the best source, which could be the data producer, a replica server, or an in-network cache, and sends the requested content back to the user. DONA [Koponen et al., 2007], NetInf [Dannewitz et al., 2013], PURSUIT [PURSUIT, 2010], CONVERGENCE [Convergence, 2010], NDN [Zhang et al., 2014], MobilityFirst [Seskar et al., 2011], COMET [García et al., 2011] all share this common communication approach. However, these project approach a number of other functionalities involved in information response flow differently, in particular, naming, routing and cache management, as follows.

2.2.1 Data Naming

The Named Data Networking [NDN] project "encourages the use of human-readable clear-text strings as name components, which resembles a file system naming scheme" [NDN, 2014]. Instead, DONA, PURSUIT, NetInf, and MobilityFirst projects following a flat naming structure. Convergence supports both hierarchical and flat naming schemes. COMET supports URL-based names. ICN-IOT [Zhang et al., 2013] encapsulates the name as part of the IP packet structure. With Flat, Hierarchical and IP encapsulation, tradeoffs must be made. For example, IP encapsulation introduces additional processing overhead. Hierarchical naming offers data aggregation and is human readable, but there is still an open question around scalability. On the other hand, flat naming is simpler and in theory allows for a fast information response time. However, for applications where real-time feedback loops rely on an efficient process to acquire, actuate and analyse content-rich data, a question arises as to what approach allows the data to be content rich enough to allow for a critical decision to be made effectively by such applications. This research will investigate the advantages offered, and tradeoffs which must be considered when considering data naming (or more specifically, what information should be included with the name), for such applications.

2.2.2 Routing

ICN solutions also differ in their routing mechanisms, which are used for both name resolution and data path creation. Name resolution usually refers to the mapping of a content name to possible locations of content sources [Piro et al., 2017]. The data path identifies the set of intermediate routers through which the data packet is forwarded towards the user that generated the request. Name resolution and data path creation may be coupled, whereby data is sent back to the user through the same path of the request or decoupled, where corresponding data packets follow different routing paths, often created by different logical nodes. CONVERGE, NDN, and COMET follow a coupled approach as they all install forwarding information in content routers as requests are resolved towards the publisher. PURSUIT and Mobility First follow a decoupled approach, where the request packets store the above information themselves and rely on source routing to return the data to the subscriber. DONA supports both.

2.2.3 Caching

A major advantage offered by ICN is in network caching, which allows information to be cached by the network rather than relying on the application layer, which is used by web caching today. Again, current ICN architectures differ in their approaches to caching. On-path and off-path caching are offered. In on-path caching, after the router receives a data request, it responds with a locally-cached copy, which has been opportunistically captured as the data previously flowed through. This scheme is natively supported by ICN, where name resolution and data path are offered in a coupled fashion, e.g., NDN and CONVERGENCE. With the off-path strategy, caches communicate their available content to the name resolution system so that incoming information requests can take advantage of the content, which may be closer to the subscriber thus allowing for faster information response. In both cases, "the ICN architecture uses nodes with cached content as parallel data producers, which may be taken into account during the name resolution process and/or routing-by-name operations" [Vasilakos et al., 2015]. In-network caching raises the question of what to cache, what to migrate, what to discard and when. When caching takes place inside the network, it is common for several types of traffic to compete for the same caching resources, and cache space management becomes critical to address. ICN in-network caching mechanisms were not designed originally to support the huge amounts of IoT data generated at the network edge nor the reverse flow of data flowing upstream vs downstream from the cloud. The success of efficient data propagation heavily relies on robust caching mechanisms and when deadline awareness is considered effective caching is essential as the success of packet delivery will be affected by cache hit and miss rates. A hybrid/overlay to TCP/IP approach has been taken in the following offerings ICN-IOT, POINT, COMET and GreenICN [Arumaithurai et al., 2013]. At the time of writing this state of the art there are several European Union sponsored and United States National Science Foundation ICN projects such as INTENT [INTENT, 2015], Ecousin [eCOUSIN, 2015], BONVOYAGE [BONVOYAGE, 2018], and UMOBILE [uMOBILE, 2018]. Intel and NSF launched a joint research center, ICN-WEN, in 2018, focused on Information-Centric Networking (ICN), and addressing discovery, movement, delivery, management, and protection of information within a network. The abstraction of an underlying communication plane is also under investigation in ICN-WEN, creating opportunities for new efficiencies and optimisations across communications technologies that could also address latency and scale requirements. An extensive survey on ICN offerings to date has been carried out by Xylomenos et al., 2014. This thesis focuses on the Named Data Network architecture (NDN) as compared with centralised architectures (such as DONA, NetInf and PURSUIT), as the distributed architecture of NDN conceptually aligns better with the distributed nature of ad-hoc networking, which is the focus of this work. The architecture of the Named Data Networking project (NDN) proposes a consumer-driven approach, where NDN fetches a named data package rather than delivering a packet to a destination host address [Zhang et al., 2010]. There are two types of packets within NDN, Interest and Data Packets. Interest packets are sent by a consumer, and include a request for a specific piece of content, identified by name. Routers forward these packets to data producers. Data packets are sent by producers in response to an interest packet and these packets contain the name and the content. The data packets are routed on the reverse path to the interest packet and the data are cached in all routers along the path. To allow this packet flow, each NDN router node contains three data structures:

- Pending Interest Table (PIT) stores all pending interests that have been forwarded but not yet satisfied. Each entry in the PIT contains the data name and the incoming and outgoing interface;
- Forwarding Information Base (FIB) is a routing table, which maps named components to interfaces;
- Content Store (CS) is a cache of information received;

NDN also uses a forwarding strategy, which makes decisions about forwarding packets.

It is important to understand the fundamentals of an NDN architecture and implementation, especially its packet structure, as this thesis will describe extensions to both the Interest and Data packets, to make them QoS aware.

2.3 ICN in mobile networks

Wireless mobility and the dynamic nature of edge-based applications, coupled with the often-time sensitive data delivery needs, have not been considered by the original NDN architecture. Subsequent offerings have looked to enhance NDN to support wireless communications [Amadeo et al., 2015] [Dynerowicz and Mendes, 2017] [Wu et al., 2018] [Amadeo et al., 2017]. However there has been a considerable body of work already done for taking a more content driven approach for data routing in MANETs [Meisel et al., 2010a, Oh et al., 2010, Meisel et al., 2010b, Amadeo and Molinaro, 2011, Lu and Zhu, 2013, Amadeo et al., 2013, Han et al., 2014, Liu and Yu, 2014, Kuang and Yu, 2015, Amadeo et al., 2015, Kim et al., 2016, Pham et al., 2015]. More recently Liu et al., 2017 have taken a conceptually hybrid approach with regard to ICN and MANETS (ICMANETS) which is an information centric MANET and is considered to be an evolving field. As the focus of the work in this thesis is on VANETs which is considered to be a subset of MANETs a more in depth evaluation in the next section of this thesis will be carried out on the current offerings in this space

2.4 ICN and VANETs

There has been considerable research on mechanisms for content dissemination in VANETS, and preliminary discussions and investigations of the advantages of ICN in vehicular networks [Xie et al., 2016a, Bai and Krishnamachari, 2010, TalebiFard and Leung, 2012, Wang et al., 2012, Yu et al., 2013, Grassi et al., 2015, Ahmed et al., 2015, Yu et al., 2015, TalebiFard et al., 2015, Wang et al., 2016b, Yaqub et al., 2016, Xu et al., 2017, Modesto and Boukerche, 2018, Li et al., 2018, Angius et al., 2012]. Each of these has looked at varying aspects of ICN, to help with content discovery and delivery, taking into account the localized and dynamic nature of a VANET. The benefits of ICN in vehicular networks were first discussed by Bai and Krishnamachari, 2010 and Wang et al., 2010 and there has been growing interest in the area since [Pentikousis et al., 2015] [TalebiFard et al., 2015]. Table 2.1 presents a summary of data dissemation mechanisms with VANETs, along with ICN VANET offerings. What is very evident is the lack of investigation to date in the area of ICN QoS offerings for VANETs. This is further discussed in Sections 2.5.2 and 2.5.3.

The implementation of ICN as an overlay over SDN was the first approach to incorporating evolving ICN functionality to existing networking technologies

Table 2.1: ICN and VANETs summary

Offering	Focus Area	Access Technology	QoS provisioning	Evaluation Platform
IC NoW [Bai and Krishnamachari, 2010]	Generic framework for VANET applications	WIFI/DSRC	Rule set based over IP, some real world implementation	US based highway
DMND [Wang et al., 2010]	Data collection from vehicles	802.16	Not discussed	Qualnet
CCDIVN [Talebi-Fard and Leung, 2012]	Data dissemination of information in VANETS -aim for better network resource management	WIFI/DSRC	Not discussed	Not discussed
BlooGO [Angius et al., 2012]	Neighbour-aware forwarding	802.11	Not discussed	Probabilistic Forwarding models
RUFS [Ahmed et al., 2015]	neighbour-aware forwarding	802.11p	Not discussed	ns2
CRoWN [Amadeo et al., 2012]	V2V/V2I forwarding-role of provider played by both car and infrastructure	802.11p	Not discussed	NS2
Navigo [Grassi et al., 2015]	location based packet forwarding	802.11p	Not discussed	ndnSIM
CCNHV [Arnould et al., 2011]	Self-organizing network model	802.11p	Not discussed	NS3

HBFR [Yu et al., 2015]	Hierarchical routing based on bloom filters	Not discussed	Not discussed	Qualnet
Hybrid VANETS [Xie et al., 2016b]	Safety applications	802.11b	Not discussed	NS2
Last Encounter Content Routing (LER) [Yu et al., 2013]	Opportunistic routing		Not discussed	ndnSIM
V-NDN [Wang et al., 2016b]	Traffic info	802.11a/.16	Not discussed	ndnSIM
SEVen [Modesto and Boukerche, 2018]	Service exchange and service management	802.11p/1609	Policy based	OMNeT++, Veins, SUMO
SDN , VCC Over ICN [TalebiFard et al., 2015]	Various	Various	Yes via logically centralized controller nodes	NS3
EcoMD [Xu et al., 2017]	Multimedia delivery	4G LTE	QOS/QOE focus-jitter,playback freeze etc	ndnSIM
CA-VNDN [Li et al., 2018]	Context aware forwarding	802.11a	Not discussed	ndnSIM, SUMO

[Salsano et al., 2013]. Recently, there has been further work incorporating ICN and SDN into VANETS [Jmal and Fourati, 2017] [Borcoci et al., 2017] [Su et al., 2017]. Here, as with any technology integration, complex trade-offs had to be made. In particular, there are evident trade-offs between ICN native content delivery and the centralised SDN approach to orchestration of heterogeneous tasks. Extensive extensions are not only needed to a packet's structure but also to the forwarding rules, and the complexity of operating such a network is high.

2.5 Quality of Service

Quality of service (QoS) in networks can be defined as the network's ability to delivery specific guarantees to the various services using that network. Most QoS approaches look to deliver these guarantees by optimally using the resources available within the network. Traditional QoS metrics focus on bandwidth, latency, jitter and packet loss. According to Pavlou and Psaras, 2018, despite many efforts to provide for QoS in networking technologies, very few solutions have been actually deployed effectively. This has resulted in a "best effort" approach to packet delivery, for the most part. The following sections look at existing QoS approaches in traditional IP networks, mobile ad hoc networks (including VANETs) and current QoS offerings in ICN. The section also explore QoS traffic control as this thesis puts forward the hypothesis that in order for an ICN network to be truly QoS aware, optimisations are required not only in the network layer, but also in the data link layer.

2.5.1 QoS in IP Networks

As a concept, QoS provisioning and fulfillment has a significant dependency on content recognition, which is not supported natively in IP host-centric networks. In current IP networking, application requirements are condensed into each packet via a 6-bit field, used to specify Differentiated Services in the IP header [Nichols et al., 1998]. The purpose of this information is to allow the network to classify and manage network traffic and provide desired quality of service. However, this is a limited number of bits within which an application can provide requirement information to the network. Multiprotocol label switching (MPLS) [Rosen et al., 2000b] and Resource Reservation Protocol (RSVP) [Wroclawski, 1997] with integrated/differential services (IntServ/Diffserv) [Wroclawski, 1997] [Blake et al., 1998], are applied as extensions over the base protocol [Yavatkar et al., 1999] for QoS consideration. Taking such an approach leads to the need to reserve resources at each hop between the source and content requester, which in turn leads to increased complexity for the network in terms of signaling, flow identification and queue processing. With increasing traffic volumes, this may become an even bigger issue for IP-based networks. Use of the IntServ field has led to significant scalability issues. The DiffServ field in the IP header can be very limiting and is only of use in access networks,it is implemented for flow priority. The same fields are applied to all the data generated by the application, but there may be parts of a given application layer data stream that require different quality of service than others (e.g., video vs. audio packets in a video conferencing application such as Skype). There is currently no support for such a distinction. This forces the networks to use deep-packet inspection engines that try to figure out the application-specific optimizations through a combination of IP source and destination addresses and the TCP/IP ports, or even decrypt and parse through application headers and then classify and prioritize them accordingly. These solutions are difficult to scale and are implemented in hardware due to the high processing requirements [Nichols et al., 1998] [Rosen et al., 2000b] [Rosen et al., 2000a].

2.5.2 QoS in mobile networks

With the advances in wireless technology, and in parallel the proliferation of mobile devices, applications are increasing served by mobile ad-hoc network (MANETs). Basagni et al., 2004 and Chen and Nahrstedt, 1999 called out the challenges for QoS within ad hoc networking i.e., a constantly changing network topology and imprecise routing information, and 15 years later, the very same challenges still ring true today. Indeed, they are exacerbated by the increasing number and mobility of devices, coupled with the exponential growth in data volume enerated at the network edge. QoS provisioning in MANETs is a challenging task, as MANETs by nature are dynamic, heterogeneous and often operate in constrained environments with limited resources [Loo et al., 2016]. While many solutions exist, many are single purpose solutions and focus on one aspect of providing QoS such as routing or location-based QoS or working to provide optimum QoS within a constrained environment e.g., bandwidth management or latency awareness. This results in a best-effort approach to network resource management, performance and optimsed data delivery, for overall network QoS. Considering the focus of this thesis, the analysis of MANETs (a very large research field) will focus on its approaches to QoS-aware routing. There have been some efforts to provide a more flexible solution and build on IntServ and DiffServ [Xiao et al., 2000] or introduce an end-to-end QoS framework [Lee et al., 2000] [Dharmaraju et al., 2002]. Both of these solutions, similarly to DiffServ, reserve network resources at each hop. This can be very challenging in a heterogeneous environment where several services may be vying for the same resources. Sinha et al. presents the Core-Extraction Distributed Ad Hoc Routing (CEDAR) algorithm [Sivakumar et al., 1999], which uses restricted flooding and source routing and is designed to select routes with sufficient bandwidth resources. Similar approaches have been taken by Lin and Liu, 1999 and Zhu and Corson, 2002, where bandwidth needs are considered by using shortest path routing. These offerings apply different methods to data routing, but consider only bandwidth as the primary QoS metric when routing data. Whereas Chen and Nahrstedt, 1999 and Shah et al., 2011 consider both bandwidth and latency as metrics for QoS adherence for data routing. Akkaya et al. put forward the argument that energy efficiency must be considered along with QoS-aware routing for new multimedia content, to ensure efficient usage of sensing resources [Akkaya and Younis, 2003]. There are many offerings where the focus is on providing energy-efficient QoS routing [Lu and Zhu, 2013] [Hassan and Muniyandi, 2017].

While Castellanos et al., 2016 recognise the need for end-to-end QoS support for time sensitive traffic such as video, major challenges remain with MANETs in providing an end-to-end QoS support for time-sensitive data. Most existing QoS multicast routing protocols for ad hoc networks were designed to support best effort services and are not end to end solutions. The literature points to the fact that providing multicasting subject to QoS constraints in wireless ad hoc networks needs further study [Zhang and Mouftah, 2005]. In contrast, by design, ICN natively supports multicast. Recently there have been some advances to considering ICN to create QoS efficencies for data delivery within 4G/5G networks. These solutions follow an overlay or hybrid approach with IP. This work is early stage with limited evaluation data available, so it is difficult to gauge the impact of such approaches [Ravindran et al., 2017] [Liang et al., 2015] [Yang et al., 2015] [Shannigrahi et al., 2018] [Suthar and Stolic, 2016] [Muscariello et al., 2018].

As discussed in the motivating scenario for this thesis, vehicles are now equipped with many different sensors, which can provide real world information and, considering the broad range of applications served by VANETs, ranging from safety information dissemination to infotainment, robust QoS mechanisms are essential to not only serve time-sensitive applications but also provide for effective traffic management to align with available compute and network resources. In terms of QoS offerings within VANETs, approaches for QoS are evolving to consider different information types and flows [Ning et al., 2018] [Al-Turjman, 2018] [Boulila et al., 2018] [Kaur et al., 2019] [Eiza et al., 2016] [Wang et al., 2016a] [Dua et al., 2015] [Eiza et al., 2015] [Hartenstein and Laberteaux, 2008] [Sun et al., 2006] [Chrysostomou et al., 2012]. None of these considers ICN as part of the offering. Vehicle to vehicle (V2V) communication in a vehicular network is still based on the TCP/UDP/IP protocol stack, which is prone to network fragmentation, and may result in vehicles not being able to communicate which each other, resulting in time-sensitive application information not being delivered.

2.5.3 QoS in ICN

ICN offers native content identification, which has the promise to be very effective when implementing QoS provisioning. Content identification could be implemented in a clean-slate ICN implementation or as an additional layer in an ICN/IP overlay approach. However, there is no common functional architectural implementation of content identification within ICN [Xylomenos et al., 2014]. The approaches vary from hierarchical content naming within Named Data Networking [NDN] [Wang et al., 2010], to COMET [COMET, 2010], which uses a block-based content identification structure, while PURSUIT [PURSUIT, 2010] uses an information management plane. The current QoS offerings within ICN also vary in approach and focus. Al-Naday et al., 2014 use QoS naming prefixes built on the PURSUIT architecture. Chu et al., 2016 employ optimised caching polices where data popularity is taken into consideration when making decisions on content placement. Wang et al., 2018 focus on energy efficiency within the network, with the approach tightly coupled to the communication network links. Tsilopoulos, 2011 explore not only the need for effective naming but also a clear understanding on specifics about the nature of the traffic. All of these solutions' primary focus is on effective Interest packet delivery, which is only part of the solution needed. In a dynamic environment where the interest packet's path cannot be relied on, the data packet's path must also be considered.

2.5.4 Qos-aware traffic control

The current architecture of ICN variants (e.g., NDN) focus only on the forwarding plane i.e., Network forwarding Daemon (NFD), to achieve name-based content delivery. However, differentiated (e.g., priority-aware/QoS-based) packet processing is also very important at the MAC layer (a sub layer of the data link layer) if you are to have a truly optimised traffic control system. The MAC layer is responsible for the sharing of the communication medium, and all the upper layer protocols are tied to this to ensure reliable communication. The current NDN design does not include traffic control to ensure optimised traffic and congestion control for QoS delivery. In addition, emerging applications such as autonomous vehicles, augmented reality, industrial robotics, which are most expected to benefit from an ICN approach, may contain sensitive control loops that are highly sensitive not only to delay, but also to the reliability of data delivery in correct sequential order with no errors and other requirements. Without a congestion control mechanism at the MAC layer for ICN QoS traffic, there is no possibility of enforcing QoS policies or managing network bandwidth/congestion, which might result in the random dropping of time-critical/ priority packets. These behaviours would lead to poor quality of service where data is not delivered to deadline requirements to the above mentioned application domains. Current solutions to traffic control within NDN are mainly centered around network forwarding, and also depend on performance metrics like Round Trip Time (RTT) and Recovery Time Object (RTO) from the applications in order to re-adjust the request rate [Schneider et al., 2016,Yang et al., 2018,Wang et al., 2013,Ren et al., 2016]. These approaches do not consider QoS or a packet's deliver deadline requirements. In addition, once packets enter the MAC layer, application-specific QoS information is no longer exposed and it is not a trivial matter to access this information to decide on differentiated processing of packets, based on priority assignment. NDNLP [Shi and Zhang, 2012] operates between the network layer and the 'link' , which supports fragmentation/reassembly and acknowledgement/retransmission of packets. NDNLP runs directly over Ethernet connectivity. The practical congestion control scheme recommended by Schneider et al [Schneider et al., 2016] measures the packet queuing time, it then signals nodes through explicit marking of selected packets. This action helps the downstream router to divert the traffic to alternate paths, and also allows consumers to re-adjust their interest sending rates. These approaches are suitable in a scenario where the multiple routes are pre-announced in routing table and the topologies are very stable and static. In this thesis, the objective of introducing congestion control at layer 2.5 is to intercept the packets, and re-order their transmission sequence by introducing priority queues and TTL based reshuffling. This approach effectively transmits high priority packets from the head of the queue and allows for the early dropping of packets which has a low probability of meeting the deadline. Since the environment is highly dynamic, it does not rely on the any alternate paths to divert the traffic in order to manage congestion. There is a body of work in the area of IP-based QoS-aware MAC protocols. These protocols offer static and dynamic service differentiation, which is done based on packet priority assignment. A static priority [Saxena et al., 2008, Liu and Elhanany, 2006,Tan et al., 2008,Firoze et al., 2007,Kim and Min, 2009,Yoon et al., 2007, Yahya and Ben-Othman, 2010] is assigned when a packet is created and remains unchanged until the destination is reached. With dynamic priorities, a packet's prioritisation [Slama et al., 2008] may change due to factors such as traffic load or remaining hop count. Liu et al., 2005 consider that a hybrid solution may exist, which is both dynamic and static in nature. Porting of existing IP network-based congestion control is not suitable for ICNs because there are only a limited number of bits available within which an application can provide QoS requirements information to the network. In order to apply service differentiation for QoS in IP, networks multiprotocol label switching (MPLS) and Resource Reservation Protocol (RSVP) under the integrated/differential services (IntServ/Diffserv) have been applied as extensions over the base protocol. Taking such an approach is based on having a known destination for packet delivery and leads to the need to reserve resources at each hop between the source and content requester. This leads to increased complexity for the network in terms of signalling, flow identification and queue processing. With increasing traffic volumes, this is likely to become an even bigger issue for IP networks in future. There has been some recent advances in the area of QoS-aware MAC protocols for VANETs. [Shah et al., 2018] propose a QoS-aware cluster-based MAC. Boulila et al., 2018 propose a hybrid solution to ensure service differentiation based on an extension to Time division multiple access (TDMA) and enhanced distributed channel access (EDCA). Both these approaches focus specifically on MAC layer QoS enhancements, rather than considering a cross-layer approach, which is proposed by this thesis.

2.6 Summary

The way in which the Internet is used today is very different to the original intended usage model. The original Internet was designed to dispatch data to the fixed IP address of a static computer. The corresponding hour glass shaped architecture centres around a thin waist universal network layer based on IP (See Fig.2.2). The thin waist fueled the Internet's growth because the technologies above and below this layer could advance independently. However, by design IP considers only host endpoints when transmitting packages. It does not focus on content, which is at odds with today's content rich applications' quality of service demands. The underlying networks of the Internet have been extensively reworked to cope with demands its original design never anticipated, such as supporting billions of devices or transporting huge amounts of multimedia content, which has led to major inefficiencies. A conclusion from the analysis carried out within this state of the art, is that research into protocols to support multiple applications with different QoS requirements running on the same network, is essential for today's evolving networks.

QoS-aware data delivery has not been widely explored within ICN, to date. Currently, ICN supports content delivery, though there has been initial discussion at IRTF [IRT, 2019] ICNRG [ICN, 2019] recognizing the gaps in its support for QoS requirements for content delivery. In the latest draft, Jangam et al. propose using a differentiated services (DiffServ) approach [Jangam et al., 2018]. This solution has not been implemented to date and therefore no evaluation on the proposed approach has been published. Current QoS offerings focus mainly on guaranteeing interest packet delivery or a specific QoS requirement e.g., bandwidth delivery [Al-Naday et al., 2014, Chu et al., 2016, Wang et al., 2010, Tsilopoulos, 2011, Kuang and Yu,]. While these solutions are interesting and directly related to this research thesis,in order to provide true data deadline awareness within ICN, both the interest and data paths must be considered which these solution do not do. This is even more prevalent in a dynamic wireless environment where it cannot be assumed that the data packet will be returned on the same path as the interest packet travelled. It is also evident that in order to support such QoS demand, compute resources must shift from a centralised to a decentralised approach to support the data explosion at the network edge [Pavlou and Psaras, 2018]. NDN allows for multipath forwarding of the interest packet to one or more next hop interfaces, and the interface information on which the packet arrived is recorded (ingress) in the PIT able. Considering the data packet flow, once the data is received, the packet is forwarded based on matching the interface from the original interest packet in the PIT table, along with the content name of the data packet. Forwarding of the data packet is not flexible and must forward based on the interface captured in the PIT entry. This creates a challenge as the same interface may be used for several data packets at the same time, as without QoS differentiation, there is no way to ensure that data delivery is treated with any prioritisation. This is a open issue within ICN for data delivery and traffic flow. ICN offerings in VANETs to date have not considered an end-to-end QoS mechanism for both content discovery and content delivery, and it remains a challenging task to maintain the QoS for various applications in vehicular adhoc networks (VANETs).

Since QoS provisioning is not a layer-specific issue and spans all layers in the communication protocol stack, cross-layer optimisation mechanisms may provide better QoS.

This research looks to address this gap in the current body of work by employing a QoS-aware solution for the entire round trip (interest and data packet delivery) of a data request and provides for optimised QoS traffic control.

Chapter 3

Design

The review of the state of the art, presented in the previous chapter, has shown that QoS-aware data delivery has not been widely explored within ICN, to date. From the current solutions, a number of limitations have been identified in the delivery of QoS service mechanisms for time sensitive data delivery, within dynamic environments at the network edge. Open issues with current research are i) current QoS offerings, although capable of content recognition, have limited support to deliver QoS mechanisms for dynamic applications ii) solutions to date have focused solely on interest packet delivery, and iii) in the current ICN traffic control mechanisms, once the packet leaves the network layer, there is no mechanism to distinguish between packets' different QoS or delivery deadline requirements, within queues at the data link layer. To address these gaps, this chapter describes a design for adding QoS awareness, in particular deadline awareness, to ICN, QoS-Aware ICN, or QoSA-ICN for short. In order to address these gaps identified within state of the art analysis the design will consider the addition of QoS Information object into the standard NDN packets for both the Interest and Data packets in the packet header. This QoS information object contain priorities, timestamp and round trip time limit for the time-sensitive information requests with different types of priority classification and also informs the forwarders that these requests must be treated differently. Secondly, to read QoS information from the packets and apply differentiated processing, an extension has to be made to the existing best-route algorithm to enable differentiated routing of QoS-Aware packets through multiple intermediate nodes (multihop) acting as forwarding hop until the packet reaches to the target node (producer or consumer). Optimising of congestion control and priority-based packet transmission to enable a more dependable Information centric approach for deadline sensitive network traffic at layer 2 must also be considered. Effective packet processing is important in order to ensure prioritised transmission of packets that have a QoS deadline. This design aims to prioritise high priority packet for further transmission and drop the low priority or priority expired packets, in order to reduce congestion in the network resulting in better network utilisation. It begins by discussing the design objectives for QoSA-ICN, followed by a description of the system model, and the deadline-awareness algorithms (RQ1). The chapter follows with a design of the traffic control algorithms added at the data link layer (RQ2), and concludes with a summary.

3.1 Design Objectives

As described in Chapter 1, the objective of this thesis is two fold: to design and build information-centric QoS-aware data propagation algorithms to deliver time-critical data requests in highly dynamic cyber physical systems at the network edge [R01] and to design and build a mechanism to effectively process the network packet within queues at the data link layer, with respect to QoS requirements [RO2]. To meet these challenges, this work aims to build a QoS-aware information centric network that supports the following design objectives, mapped to the challenges and research objectives outlined in Chapter 1. From the motivating scenario (see Section 1.1), the vehicular application domain presents complex networking requirements, particularly in the context of the challenges outlined in Section 1.1 and these challenges are exacerbated by the volume of data which is now being and will continue to be generated by these vehicles at the network edge. To address these challenges, the solution must:

- **Design Objective 1: Proactively gather information**. The topology of a dynamic network is constantly in flux because of mobility patterns of its members, in this case vehicles. Vehicles in the topology must be prepared to gather, cache and share information locally based on their world view [CH1, RQ1, RO1, RO3].

- **Design Objective 2: Support information sharing over short and long wireless communication links**. The use of both short(WiFi) and long links(DSRC) to communicate should reduce the impact of both mobility and network disconnects, as information can be requested from vehicles within range [CH2,CH3, RQ1, RO1, RO3]

- **Design Objective 3: Support QoS-aware content delivery**. QoS-aware content delivery is needed for time sensitive applications at the network edge [CH4, RQ1, RO1].
- **Design Objective 4: Distinguish between different content types in order to priortise data delivery based on application QoS requirements**. In order to fully support QoS, the system must be able to distinguish between different data types in line with QoS needs [CH4, RQ1, RO1].
- **Design Objective 5: Optimise packet processing at the data link layer with respect to its QoS requirements**. Once the packet leaves the network layer, a mechanism is needed to distinguish between packets based on their QoS or delivery deadline requirements, and process each differently within the queues at the data link layer [CH4, RQ2, RO2].

The system model in the next section formalises and introduces the terminology and assumptions of the design, which support its implementation (addressing RO4) as discussed in Chapter 4. The results outlined in Chapter 5 address RO5.

3.2 System Model

There are limited methods defined within the ICN framework, or NDN, to ensure that certain quality of service application requirements such as the reliability of the content delivery, the timeliness of data packet delivery are met. The QoS-aware ICN design in this thesis includes a means to classify the priority of requests, with associated QoS requirements and the work also explores extending network traffic control mechanisms to assess the potential impact on network congestion management. QoS information is encoded into the request packets and corresponding data packets, and the routing algorithm is extended to use multi hop/multi path forwarding to efficiently request and receive the requested content, while taking the QoS requirements into account. This is achieved without pre-building routes in FIB tables. The model is defined for deadline-aware data delivery in large-scale and highly dynamic topology scenarios in cyber physical systems. This is grounded in a vehicular network across sparse, medium and dense car scenarios, with cars moving at various speeds with varying numbers of producers and consumers of data. Considering models are usually defined for a given space, take into account time and contain objects, who may or may not be actors with roles. In the thesis scenario, the modelling of space is considered **??** and objects of space are the vehicles. Some of these become the actors and as they take the role of the consumer/forwarder or producer they also become entities. The model has been designed to be extendable to other QoS requirements. The priority index within the namespace in model allows for this (p1-pn). The baseline value and target metric for the QoS classifier allows a baseline QoS value and a target to be specified based on QoS requirements needs. Local evaluation of other QoS metrics would be very much dependent on the QoS requirements e.g. hop-count or consumption of resources such as CPU or memory could be evaluated locally whereas a QoS requirement dependent on throughput or link latency between nodes could not be evaluated locally. The model is formulated in the context of NDN, which is a commonly-used ICN architecture, and was selected as a basis for this work because it is both open source, and has a related simulator (i.e., ndnSim).

3.2.1 Terminology

Table 3.1: Notation used for Algorithm Specification.

Notation	Description
$\mathcal{T}$	Set of time steps
t	A single time-step
$\mathcal{S}$	Finite Topological Space
$\mathcal{E}$	Set of Actors in Space or Entities
e	One entity, often used as subscript to denote the entity
$\mathcal{S}_A$	Set of application-specific properties of space
$\mathcal{S}_B$	Set of boundary parameters of space
z	One object in space
S_0	Initialization of space
l	Length of Space or Entity when used with subscript e
w	Width of Space or Entity when used with subscript e
h	Height of Space or Entity when used with subscript e
δ	Depth of Space or Entity when used with subscript e
(x_e, y_e, z_e)	3D Cartesian Position in the X, Y and Z planes
$\mathcal{R}$	Set of all roles
$\mathcal{R}_e$	Set of roles applicable for entity e
r	One specific role

$\mathcal{J}$	Set of Incoming Interfaces
J	Subset of Incoming Interfaces
j	One incoming interface
$\mathcal{K}$	Set of Outgoing Interfaces
K	Subset of Outgoing Interfaces
k	One outgoing interface
$\mathcal{V}$	Set of Packets
$\mathcal{V}^i$	Set of Interest Packets
$\mathcal{V}^d$	Set of Data Packets
v	One packet
i	Superscript indicating interest packets
d	Superscript indicating data packets
λ	Rate of sending interest packets
v_η	Namespace of Packet v
v_ϕ	Selector for Packet v
v_ζ	Nonce for Packet v
v_σ	Digital Signatures for Packet v
v_M	Meta-Data and signed info for Packet v
v_γ	Data Payload for Packet v
$\mathcal{P}$	Set of priority values
p	One priority value
$\mathcal{Q}^*$	Set of all possible QoS classifiers
$\mathcal{Q}$	Set of QoS classifiers under consideration
q	One QoS classifier
v_p	Priority for Packet v
v_q	QoS metrics for Packet v for QoS classifier q
q_p^{target}	Target metric for QoS classifier q and priority p
$q_e^{baseline}$	Baseline value for QoS classifier q as defined by entity e
$q_e^{timestamp}$	Packet timestamp for QoS classifier q as defined by entity e
$q_e^{current}$	Current timestamp calculated by entity e for QoS metric q
ω	Set of Queues defined for traffic control
ω_0	Root Queue for traffic control
ω_p	Internal Queue defined for packets with priority p
$\mathcal{E}_{NDN}$	Set of entities with installed NDN stacks
$\mathcal{E}_{non-NDN}$	Set of entities with installed non-NDN stacks
θ	The coefficient of friction of the road segment

n	Number of lanes of the Road Segment
α_e	Speed of car e
β_e	Stopping distance for the car e
ρ	Density of Road Segment
$\mathcal{C}_e$	Content Store for entity e
$\mathcal{F}_e$	FIB for entity e
$\mathcal{U}_e$	PIT for entity e
τ_v	Timestamp at which the packet v was last seen

$\mathcal{T}$ is a set of discrete timesteps, with $t \in \mathcal{T}$ denoting one such timestep. A finite topological space $\mathcal{S}$, within which the model is applicable, a set of entities $\mathcal{E}$, and a set of communication packets being requested and received, represented as set $\mathcal{V}$ are also defined as detailed below. Table 3.1 explains each mathematical symbol.

Modelling Space

Space $\mathcal{S}$ is 3-dimensional, where each $z \in \mathcal{S}$ has a combination of application-specific properties, $\mathcal{S}_A$ and generic properties, $\mathcal{S}_B$. $\mathcal{S}_B$ represents the length, width and height of an object in the space, from a known quantified point in this space S_0, denoted as $\mathcal{S}_B = \{S_0, l, w, h\}$, where l is the length of the space, w is the width of the space, and h as the maximum height of the space. Application-specific examples of space properties, $\mathcal{S}_A$, are explained later, in Section 3.2.3. Given this, $z \in \mathcal{S}_A \cup \mathcal{S}_B$ are an object's properties applicable within this space.

Modelling Actors or Entities

Entities are actors in the space, and set $\mathcal{E}$ represents all entities in space $\mathcal{S}$. Entities are defined by their position in space, their roles, and their functional properties, which includes interfaces as well as application-specific properties.

Entity Position: Position is defined as a 3D Cartesian coordinate in space $\mathcal{S}$, $(x_e, y_e, z_e)_t$, at timestep t, where $e \in \mathcal{E}$. This includes static entities (i.e., the value of the coordinates remains constant in time) and mobile entities (i.e., the value of coordinates is dependent on the timestep).

Entity Role: The roles assumed by an entity can change between timesteps and are defined for timestep t as a set $(\mathcal{R}_e)_t$ for entity e, where $|(\mathcal{R}_e)_t| \geq 1$. These roles indicate whether an entity requests network packets as a *consumer*, sends network packets as a *producer* or forwards network packets as a *forwarder* in the system. Entities can have one or more roles, and $(\mathcal{R}_e)_t \subseteq \mathcal{R}, \forall t \in \mathcal{T}$ where $\mathcal{R} =$ ('C', 'P', 'F') defines possible roles that can be assumed as presented in Eq.3.1.

$$\forall r \in \mathcal{R}, r = \left\{ \begin{array}{ll} 'C', & e \text{ is a consumer} \\ 'P', & e \text{ is a producer} \\ 'F', & e \text{ is a forwarder} \end{array} \right\} \tag{3.1}$$

Entity Functional Properties: **The functional properties of each entity $e \in \mathcal{E}$ are defined as a set of network incoming interfaces $\mathcal{J}_e$, where one such incoming interface is defined as j, and network outgoing interfaces $\mathcal{K}_e$, where one such outgoing interface is defined as k.** Other application-specific properties may also be included, examples of which are defined later in Section 3.2.2.

Modelling Communication via Packets

Entities communicate with each other, via their interfaces, by sending packets (set $\mathcal{V}$), either as requests for specific information or in response to the requests. The set $\mathcal{V}$ of packets is composed of two types of packets, $\mathcal{V}^i$ which is the set of interest packets (i.e., requests) and $\mathcal{V}^d$ which is the set of data packets (i.e., responses).

Although, $\mathcal{V} = \mathcal{V}^i \cup \mathcal{V}^d$, it is important to note that $\mathcal{V}^i \cap \mathcal{V}^d = \emptyset$. All entities send interest packets at rate λ_e only for the duration that they assume the consumer role, and this rate is non-uniform over all entities (as defined in Eq. 3.2). They then receive corresponding data packets from the entities that have the role of a producer.

$$\forall e, t : (\mathcal{R}_e)_t \neq \emptyset, \lambda_e = \left\{ \begin{array}{ll} > 0, & \text{'C'} \in (\mathcal{R}_e)_t \\ 0, & \text{otherwise} \end{array} \right\} \tag{3.2}$$

This implies that $\left|\mathcal{V}^i\right| = \sum_{\forall e \in \mathcal{E}, \text{'C'} \in (\mathcal{R}_e)_t} \lambda_e$. It is important to note that $|\mathcal{V}^i| \geq |\mathcal{V}^d|$, as although all interest packets have an equivalent data packet, it might not be created if there is not enough time left, often leading to re-transmission of interest packets. The time constraint is defined during the creation of the interest packet, as a time to live (TTL). Within NDN the Interest Lifetime value indicates the (approximate) time remaining before the Interest times out. The value is in milliseconds. The timeout is relative to the arrival time of the Interest at the current node. Nodes that forward interest packets may decrease this lifetime value to account for the time spent in the node before forwarding, but are not required to do so within the current design of NDN and it is currently recommended that these adjustments be done only for relatively large delays (measured in seconds) [ns 3/src/ndnSIM, 2019] FreshnessPeriod: The optional FreshnessPeriod in the data packet indicates how long a node should wait after the arrival of a data before marking it “non-fresh”. The encoded value is in milliseconds. Note that the “non-fresh” data is still valid data; the expiration of FreshnessPeriod only means that the producer may have produced newer data. If the Data packet carries a FreshnessPeriod greater than zero, a node should initially consider it “fresh”. After the Data has resided in the node for FreshnessPeriod milliseconds, it will be marked as “non-fresh”. Both InterestLifetime and content freshness are computed based on the time spent on each node without taking into account of delays or latency incurred during transmission. Hence, a complete end to end time tracking mechanism is needed to include both time spent in the node as well as on transmission, QoSA-ICN addresses this gap by including a TTL value for round trip information request and delivery path.

An *interest packet* for entity e, represented as v_e^i is defined in literature as being composed of three values, $v_e^i = \left\{v_\eta, v_\phi, v_\zeta\right\}$ where v_η is the namespace of the packet, which includes the content name and the priority of the packet, v_ϕ is the selector including order preference, and v_ζ is the nonce of the packet, used to uniquely identify this packet.

A *data packet* for entity e, represented as v_e^d is defined in literature as being composed of four values, $v_e^d = \left\{v_\eta, v_\sigma, v_M, v_\gamma\right\}$, where v_η is the namespace of the packet as before, v_σ is the signature including the digest algorithm, v_M is the metadata and signed information associated with the packet and v_γ is the payload or data in the packet.

3.2.2 QoS Inclusion in NDN

The novelty of QoSA-ICN is the addition of quality of service (QoS) information to the packet distribution decision-making process, which is achieved in the context of NDN.

QoSA-ICN defines a set of priorities $\mathcal{P}$. Each priority $p \in \mathcal{P}$ is composed of a combination of pertinent QoS classifiers $\mathcal{Q} \subset \mathcal{Q}^*$ and their respective QoS targets, where $\mathcal{Q}^*$ is the set of all possible QoS classifiers, e.g., deadline-awareness, latency, throughput, hop count restrictions. These targets are represented as $q_p^{target}, \forall q \in \mathcal{Q}$ for the priority p.

For QoS-Aware ICN, each packet v is extended to include this priority value in its definition as represented by symbol v_p where $v_p = p : p \in \mathcal{P}$, and is added as part of the namespace v_η. Furthermore, the QoS classifiers are added to each packet for its priority, as $v_q, \forall q \in \mathcal{Q}$ as $|Q|$ elements, where each element is a duple $v_q^p = \left[q_{v_p}^{target}, q_e^{baseline}\right], \forall q \in \mathcal{Q}$ and $v_p = p$. The former defines the target of the QoS metric, for the given priority v_p of the packet (added as subscript) and the latter, $q_e^{baseline}$ is the associated baseline value of that QoS metric, which is calculated by a given entity e (added as subscript) creating the interest packet. Thus, the interest packet is composed of $v_e^i = \left\{v_\eta,\ v_s,\ v_\zeta,\ v_p,\ v_q\right\}$ and the data packet is composed of $v_e^d = \left\{v_\eta, v_S, v_M, v_\gamma, v_p, v_q\right\}$.

For example, the current work focuses on making ICN deadline aware, and we have identified three priorities $p_1, p_2, p_3 \in \mathcal{P}$, each associated with one QoS classifier indicating round-trip delay tolerance. As such, $|\mathcal{Q}| = 1$, and in this case, each priority relates to the deadline associated with the packet only. In other words, q_p^{target} defines the deadline, e.g., $p_1 = 100$ ms, $p_2 = 200$ ms and $p_3 = 300$ ms. Each packet sets a priority from one of these options. This priority is then interpreted by entities as a set v_q containing one element, which is a duple with $q_{v_p}^{target}$ as defined by the priority, and $q_e^{baseline}$ set to the packet timestamp. In this formulation, $q_e^{timestamp}$ is set to equal to $q_e^{baseline}$, and the symbols are used interchangeably. For QoS evaluation, a time to live (TTL) for packet v is computed by the entity e using target ($q_{v_p}^{target}$), baseline timestamp ($q_e^{timestamp}$) and the current timestamp of evaluation ($q_e^{current}$) as given by Eq. 3.6.

$$\forall v \in \mathcal{V}, v.\mathrm{TTL} = q_{v_p}^{target} - (q_e^{current} + q_e^{timestamp}) \tag{3.3}$$

This can be extended to include any QoS classifier of interest, for example, future work could implement a limit on the number of hops for forwarding the packet. In that case, priorities would be composed of the deadline target as well as the maximum hops permissible.

3.2.3 Formulating Real-World Scenarios

These definitions can be applied to multiple different real-world scenarios, including vehicular networks (VANETS) that rely on network communications for application specific data delivery. Other types of systems that have a notion of timesteps, space and entities could also be included - e.g., use-cases in Industry 4.0, logistics and transportation.

In the context of VANETs, space $\mathcal{S}$ is defined as a road segment with a dual carriage way. Space, as a road segment, uses the boundary parameters in set $\mathcal{S}_B = \{S_0, l, w, h, \delta\}$ and application specific parameters $\mathcal{S}_A = \{\theta, n\}$. Here S_0 is the starting coordinate represented in 3D as Cartesian coordinates (x_0, y_0, z_0). For the road segment, l is the length of the road, w is the width of the road, and the peaks and troughs of the road are defined using h as the maximum height of the road and δ is the maximum depth. θ defines the coefficient of friction of the road, and n defines the number of lanes of the road.

Set $\mathcal{E}$ defines relevant actors as cars on the road segment that exchange network packets for data delivery. Other examples of actors in this scenario not explicitly implemented in this work include static smart infrastructure elements (e.g., Road Side Units). The application-specific parameters for each car $e \in \mathcal{E}$ are the length l_e, width w_e and height h_e of the cars. They further include the speed of the car α_e, the reaction time τ_e^r, the reaction distance $\Delta_e^R = (\alpha_e * \tau_e^r)/c_1$ where $c_1 = 3.6$ to convert km/h to m/s, breaking distance $\Delta_e^B = \alpha_e^2/(c_2 * \theta)$ where $c_2 = 250$ and the stopping distance $\beta_e = \Delta_e^R + \Delta_e^B$, which are defined for a mobile car with movement restricted in road segment $\mathcal{S}$. The maximum road vehicle capacity ψ can then be calculated using the formulation in Eq. 3.4, and the maximum density ρ of the space or road segment, is defined in Eq. 3.5, given that $|\mathcal{E}| \geq \rho$ and $w \geq 4 \cdot \max w_e$. These are computed based on the average transport area across multiple countries [nationmaster.com,] and the formulation of stopping distance based on a summation of the reaction distance and braking distance [bra, 2018].

$$\psi = \frac{l * n}{\max \beta_e + \bar{l_e}} \tag{3.4}$$

$$\rho = F(l, \bar{l_e}, \max \alpha_e, \max \beta_e, \theta, n) \tag{3.5}$$

3.2.4 Packet Propagation Management

To manage packet propagation using the above definitions, each entity e has a content store of cached information and two tables, the Forwarding Information Base (FIB) and the Pending Interest Table (PIT) [Zhang et al., 2014] [Ullah et al., 2016].

The content store $\mathcal{C}_e$ is a collection of the namespace, the data payload and meta-data given by (v_η, v_γ, v_M), and the $|\mathcal{C}_e|$ can be configured based on available storage.

The FIB $\mathcal{F}_e$ is a collection of the namespace and set of outgoing interfaces connected to this namespace for all interest packets v_e^i that have a network path defined for entity e. This is defined as $\left(v_\eta, K_e\right)$ for each unique packet, where $K_e \subseteq \mathcal{K}_e$.

The PIT $\mathcal{U}_e$ is a collection of the namespace, the nonce, the set of incoming interfaces connected to this namespace for all interest packet v_e^i that have a known network path defined for entity e, and the timestamp τ_v at which this packet was last seen. This is defined using $\left(v_\eta,\ v_\zeta, J_e, \tau_v\right)$ for each unique packet, where $J_e \subseteq \mathcal{J}_e$.

There are multiple other parameters, e.g., freshness and placement policy defined in the content store or the time limits defined in the PIT, that are present in current NDN literature [Zhang et al., 2014] [Ullah et al., 2016]. This research has adopted the default definitions and configuration for these parameters.

3.3 Deadline-Aware Data Delivery

This section describes the design of the deadline-aware data delivery system in the context of highly dynamic vehicular networks. In the original NDN architecture, the routing information base (RIB) stores static or dynamic routing information registered by applications, operators, and the NDN Forwarding Daemon (NFD) itself. Each routing entry in the RIB [named data.net,] indicates a name, available next-hops, hop's face, the origin, a cost, etc. The RIB module processes all these routes to generate a consistent forwarding table and maintains it with the NFD's FIB, which contains only the minimal information needed for forwarding decisions. As such, the current design assumes the presence of an entry in the FIB in order to forward Interest packets to the next available entities. Otherwise, the packets will be dropped. In the scenarios for this thesis, network topologies are composed of entities (e.g., vehicles), which are mobile, and their location on the road at any specific time is difficult to predict accurately. In such a case, it is impossible to prepopulate the routing information in the FIB in order to route the incoming interest to a content source. In this work, the original NDN design is extended as shown in Fig. 3.1, which illustrates a check for QoS information in the packet. If found, packets get broadcasted using incoming interfaces subject to expiry of a packet's TTL duration. An overview of the whole desdline-aware data delivery process is illustrated in the process flow in Fig. 3.3.

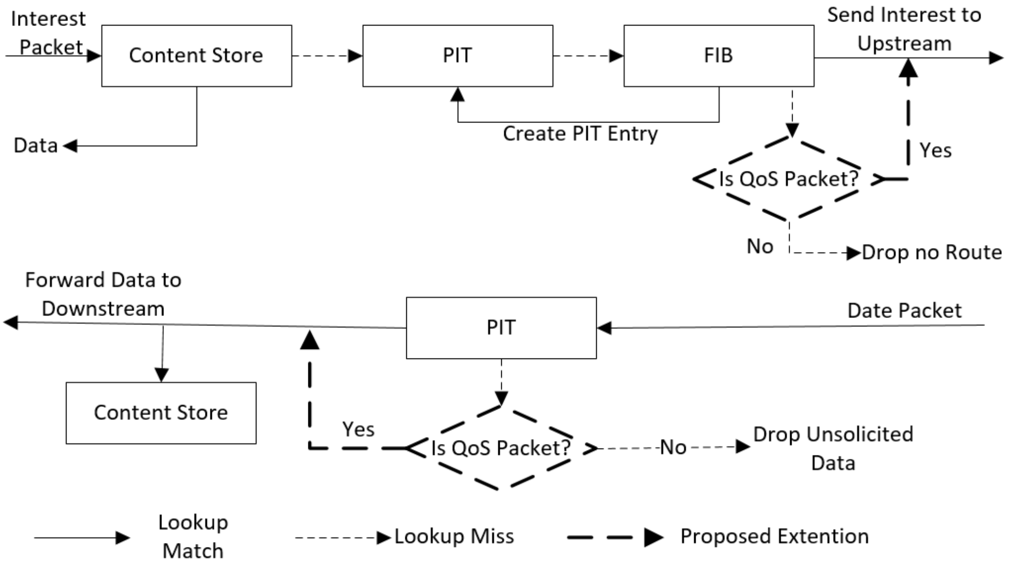


Figure 3.1: QoS extended NDN in high level abstract design upstream flow - Interest packet path Downstream flow - Data packet path.

3.3.1 QoS-Aware Interest Packet Generation

In this work, NDN Interest packets are classified into three priority groups (i.e., high, medium, and low) based on the QoS requirements of applications. When the Interest packet is generated by a consumer (requester) node, a set of QoS-related metrics are also added to the packet's header field. To achieve this, a QoS object of TLV (Type Length Value) wire-encoded format is created with the required QoS parameters such as priority, round trip time limit and the timestamp denoting the actual timestep when the packet was initially sent to the network. This QoS object is inserted into every Interest packet sent by the requester (i.e., consumer) nodes. The QoS-extended NDN interest packet is shown in Fig. 3.2(a) as a new field 'QoSInfo'. These are additional read-only parameters that stay in the packets throughout the lifetime of the Interest packets, the namespace also carries the priority and will be discussed fully in section 4.1.1 . When a node receives Interest packets it checks for QoS information. If found, then QoS metrics are retrieved for further processing. If no QoS information are found, then the packets will be processed normally by following the NDN default Interest forwarding procedures. The details of Interest packet generation are shown in Function 1. To enable this feature, the code base of the Interest class located in the 'ndn-cxx' module has been extended to accommodate the insertion of the new QoS object field.

Function 1 SendPackets()

```
$\forall e \in \mathcal{E}_{NDN}$, 'C' $\in \mathcal{R}_e$
function SendPackets( )
    Create $v_e^i = \left\{ v_\eta, v_\phi, v_\zeta \right\}$
    if $v_\eta$ contains $p : p \in \mathcal{P}$ then
        Calculate $q_e^{timestamp}$
        Initialize $v_p = p$
        Define $v_q = \left\{ q_p^{target}, q_e^{timestamp} \right\}$
        Update $v_e^i = v_e^i \cup v_p \cup v_q$
    end if
    Update $\mathcal{V}^i = \mathcal{V}^i \cup v_e^i$
end function
```

Figure 3.2: QoS Extended NDN Packets. a) Interest Packet b) Data Packets

3.3.2 QoS-Aware Multihop Interest Forwarding

The NDN packet forwarding process relies on the routing information presented in the FIB in order to resolve the outgoing interfaces. This approach works quite well in a wired network scenario with static topologies, as long as routes are pre-announced for Interest names that need to be forwarded further in the network. However, it is impractical to pre-populate the routing information in advance in a highly dynamic wireless environment where the network topology is rapidly changing. In such wireless scenarios, rebroadcasting of the packets on the incoming face is one of the viable solutions. But, the current NDN design does not allow the forwarding of packets on incoming faces. This restriction is not welcome, especially in the wireless mobile ad-hoc network scenarios. There have been some previous works [Amadeo et al., 2015, Kalogeiton et al., 2017] that have tried to rebroadcast the Interest using the incoming faces with some additional strategies. Amadeo et al., 2015 addressed the rebroadcasting of Interest through its incoming WiFi faces with new forwarding approaches (blind and provider aware forwarding). Kalogeiton et al., 2017 adopt a node identity-based forwarding approach, using MAC addresses in the Data and Interest packet to create a multihop forwarding path, where predicting the MAC correctly is important. However, in these works, large-scale simulation experiments are needed using highly-dynamic vehicular network scenarios, to assess the impact on network congestion, etc.

In this work, the controlled propagation of QoS-aware packets is considered, in the absence of routing information in the FIB. This has been achieved by re-broadcasting the Interest message into the incoming interface and the time-based packet suppression technique by extending the existing best-route strategy (see Function 2 and Function 3). Extending the best route strategy allows flexibility to be retained for different networking scenarios such as QoS-aware forwarding or non-QoS-aware packet forwarding, using a best route algorithm. However, the key criteria used here for rebroadcasting is based on the QoS parameters, i.e., priority and validity of time to live (TTL). The algorithm shows the calculation and validation of TTL. At each forwarding node, the packet's TTL is calculated using current time at node and the timestamp recorded (QoSInfo) in the packet when the packet was initially sent out in the network. If the TTL for the packet has expired i.e., $TTL <= 0$ then the packet will not be forwarded any further in the network. This helps to control packet propagation and avoid flooding the network. Further, an extension has also been made in the current NDN to process the Interest looping scenarios, to exempt the QoS-aware packets (as shown in Function 4) from sending the unwanted duplicate Nack messages resulting from rebroadcasting of packets.

3.3.3 Best Route Interest Forwarding Strategy Adoption

The current NDN design offers, as a default routing strategy, a best route forwarding strategy [Mastorakis et al., 2017, Afanasyev et al., 2014] to select the best possible route among the multiple routes that exist for a content name. This strategy compares the routing paths on various criteria and forwards Interest packets on the path(next hop) with the lowest cost and is implemented in the network forwarding daemon as nfd::fw::BestRouteStrategy2 class. This

Function 2 Process(Interest Packet, Incoming Interface)

```
function PROCESS($v_e^i \in \mathcal{V}^i$, $j_e$)
    Get $v_\eta, v_\zeta : v_\eta, v_\zeta \in v_e^i$
    if $v_\gamma \in \mathcal{C}_e(v_\eta)$ then
        return $(v_\gamma)$
    else if $\mathcal{U}_e$.contains($v_\eta$) then
        $u_e = \mathcal{U}_e$.Find($v_\eta$)
        if $v_\zeta \in u_e$.Get($v_\eta$) then
            return
        end if
        Update $J_e = J_e \cup j_e$
        if PitEntry.RetryTimer $\leq 0$ then
            FORWARD($v_e^i$, $u_e$)
            return
        end if
    else
        $u_e = \mathcal{U}_e$.Create(($v_\eta, v_\zeta, J_e$,current timestamp)
        FORWARD($v_e^i$, $u_e$)
    end if
end function
```

strategy was chosen for this thesis as it employs the lowest cost next hop an "exponential back-off" algorithms which allows consumer applications to control their own packet retransmission rates which was designed to prevent DDoS attacks. It is also an important feature for Quality of service where specific deadlines for RTT must be measured before retransmission of interest is invoked.In this thesis, the best route strategy is extended to include a multihop forwarding algorithm that supports both QoS-aware forwarding as well best route forwarding options, depending on the scenarios. This approach guarantees the execution of the current best route algorithm when a forwarder finds the next hop for an incoming interest packet, including both QoS and non-QoS aware ones, as shown in Function 2. If the interest packet contains QoS information, the new QoS-aware multihop strategy will be invoked. See Section 3.3.5 for more details on the multihop forwarding.

3.3.4 Producer Data Packet Composition

A data packet is created once an interest packet reaches an entity (i.e., producer) that has the requested data. By default in the original NDN architecture [Zhang et al., 2014], an NDN producer will return a data packet that is composed of both name and the content, along with a digital signature created using the producer's key to retain the integrity of the name and data. In this thesis, the information from the QoSInfo field of the interest packets are also copied into

Function 3 Forward (Interest Packet, Pit Entry)

```
function FORWARD($v_e^i$, $u_e$)
    $f_e = \mathcal{F}_e$.Find($v_\eta$)
    if $f_e$.nextHops().exists() then
        $K_e = f_e$.nextHops()
        for all $k_e \in K_e$ do
            SENDUPSTREAM($u_e, k_e, v_e^i$) return
        end for
    else if $v_e^i$.QoSInfo().exists() then
        $v_q = v_e^i$.getQoSInfo()
        $v_e^i$.TTL $= q_{v_p}^{target} - (q_e^{current} + q_e^{timestamp})$
        if $v_e^i$.TTL $\geq 0$ then
            $j_e$ = GetLast($J_e$)
            BROADCASTPACKET($u_e, j_e, v_e^i$)
        end if
        Send Nack(reason NoRoute) return
    else
        Send Nack(reason NoRoute) return
    end if
end function
```

Function 4 ProcessInterestLoop (Incoming Interface, Interest Packet)

```
function PROCESSINTERESTLOOP($j_e \in J_e, v_e^i$)
    if $j_e$.Type != 'LINK_TYPE_POINT_TO_POINT' then
        return
    end if
    if $v_q \in v_e^i$ then
        return
    end if
    Send Nack(reason Duplicate)
end function
```

the data packet. This is to inform the forwarding nodes in the downstream (i.e., data path) to process this data packet according to the QoS requirement defined in the Interest packet. Fig. 3.2(b) shows the extended version of an NDN data packet, and Function 5 describes the data packet composition in details.

3.3.5 Data Packet Multihop Forwarding

The current NDN design assumes that data packets flow back to the requesting consumer by following the breadcrumbs [Amadeo et al., 2015] left by the corresponding Interest packets in the intermediate nodes, along the Interest's upstream path, during the Interest packet's propagation. However, due to the dynamic nature of a mobile ad-hoc network, the topology formed by fast-moving entities is unlikely to be the same when the data packet flows back to the consumer. Such breadcrumbs may not exist to guide the data packet to reach the

Function 5 OnInterest (Interest Packet)

```
function OnInterest($v_e^i$)
    Create $v_e^d = \{\}$
    Get $v_\eta \in v_e^i, v_\gamma, v_M \in \mathcal{C}_e$
    $v_e^d = v_e^d \cup v_\eta$
    $v_e^d = v_e^d \cup v_\gamma \cup v_M$
    if $v_q \in v_e^i$ then
        $v_e^d = v_e^d \cup v_p \cup v_q$
    end if
    Create $v_\sigma$
    $v_e^d = v_e^d \cup v_\sigma$
    $\mathcal{V}^d = \mathcal{V}^d \cup v_e^d$
    Send $v_e^d$
end function
```

original requester (i.e., consumer). It is also highly likely that one or more entities that acted as forwarding hop(s) have exited the topology (e.g., when a car exits the road) and one or more new entities might also join the network, and form a new topology. In such a scenario, with the existing NDN implementation, the data packets will fail to reach the consumer unless there is a direct link between producer and consumer, which is very unlikely. In the proposed design, the QoS-aware data packets are pushed towards the consumer node without relying on the presence of Interest breadcrumbs. As shown in Function 6, the proposed design enables the opportunistic usage of new nodes entering the Data (NDN downstream) path to assist with rebroadcasting the data in the fast-changing network topology. Similar to Interest packets, the Data forwarding criteria also employ QoS metrics such as priority and packet round trip time limit to make sure that packets are not forwarded indefinitely in the network.

3.4 Traffic control

As described in Chapter 2, to date, approaches taken to congestion control have focused only on the network layer, without considering QoS. Figure 3.4 shows where this thesis proposes to insert a congestion control mechanism at the data link layer. The mechanism is designed to intercept packets leaving the NDN forwarding stack and re-order them in the forwarding queues, to ensure prioritized transmission of packets with shorter delivery deadlines.

The congestion control system retrieves the QoS information from the packet tag of each network packet. The NDN protocol converts the packet into a standard network packet for transmission before leaving the network layer to the

Function 6 OnIncomingData (Data Packet, Incoming Interface)

```
function OnIncommingData($v_e^d, j_e$)
    Get $v_\eta : v_\eta \in v_e^d$
    if $v_\eta$.ViolatingLocalHost() then
        return
    end if
    if $v_\eta \notin \mathcal{U}_e$ AND $v_p \notin v_e^d$ then
        ProcessUnSolicitedData($v_e^d$)
        return
    else
        $J_e = \mathcal{U}_e$.Lookup($v_e^d$)
        for all $j \in J_e$ do
            if $j$.Id == $j_e$.Id AND Face.LinkType ≠ Ad_Hoc then
                if $v_p \notin v_e^d$ then
                    continue
                end if
            end if
            ForwardData($v_e^d, j$)
        end for
        if $J_e = \emptyset$ then
            ForwardData($v_e^d, j_e$)
        end if
    end if
end function
```

data link layer. The QoS information consists of packet priority classification i.e., $p_1, p_3, p_3, \ldots, p_n$ (the smaller the subscript value, the higher the priority), timestamp t_0, the time the packet was first sent by a consumer, and the round trip time (RTT) limit t_{rtt}. As shown in Function 7, packets arriving from the NDN stack are first queued in the Root Queue, in the order of their arrival. Then, the packets are placed in internal queues, created for each priority class, as shown in Fig 3.5. The placement order of the packets in the internal queues is shuffled such that the packets with the least time to live t_{ttl} always stay at the head of each internal queue. The time to live values are calculated using Eq.3.6 and Eq.3.7, where $t_{current}$ is the present time at the queue and the elapsed time is the duration since the packet was created at the NDN application. If the elapsed time is greater than t_{rtt}, the packet will be dropped. The length of the queue can be limited with preconfigured values, for example a maximum of 1000 packets per queue. Packets get dropped if the number of packets exceeds the maximum limit, which controls congestion.

$$t_{ttl} = t_{rtt} - \text{elapsed time} \tag{3.6}$$

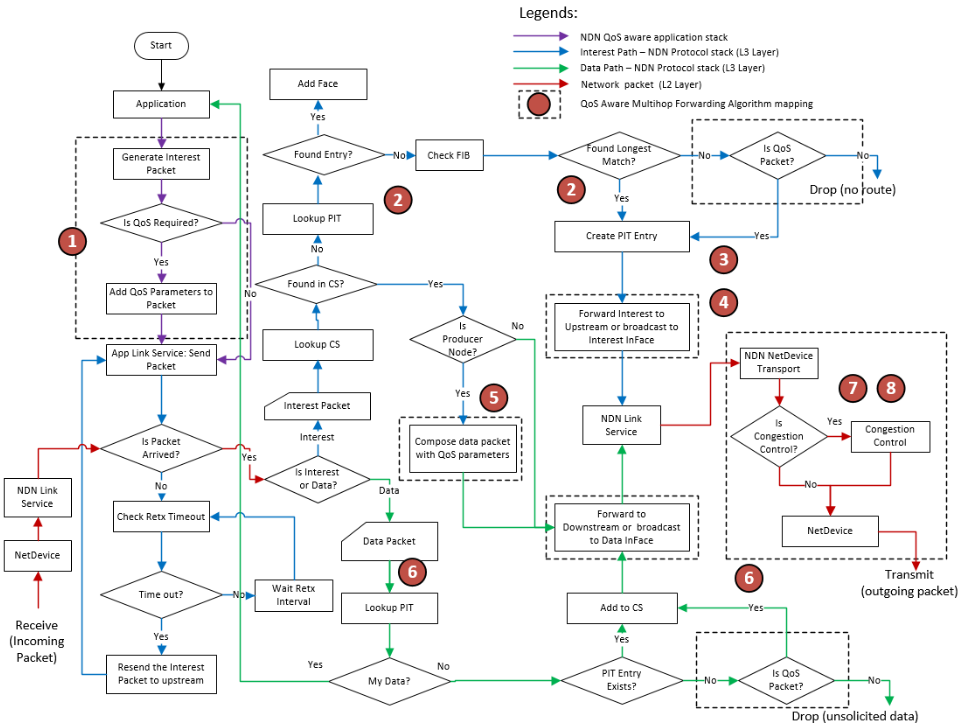


Figure 3.3: NDN QoS-aware multi-hop forwarding Process flow. Flowchart shows the activities of Packet generation with QoS parameters, Interest packet forwarding (upstream flow) and Data packet composition and forwarding (downstream flow), in NDN protocol stack at the network layer. Numbers on the diagram map to Algorithm (pseudo code) Function numbers in the thesis.

$$\text{elapsed time} = t_{current} - t_0 \tag{3.7}$$

The process for selecting packets to transmit is shown in Function 8. To choose the packets for transmission from the internal queues, the TTL values of packets that present at the head of the queue are read first. Then, comparisons are made such that the packets with the least TTL value and highest priority class are selected. If more than one packet shows same t_{ttl} values, then the high priority packet will be selected to transmit further down in the network (NetDevice). If more than one packet shows both the same t_{ttl} and priority, then they are transmitted in order of arrival. The flowchart shown in Fig. 3.3 describes the integration of the congestion control system with the QoS-aware multi-hop packet forwarding functions added to NDN. This enables the differentiated processing of packets at the data link layer using QoS metrics defined in the routing plane (NDN stack) to balance the network bandwidth. These mechanisms also allow for dropping packets from the tail of the queue (low priority packets), when the maximum limit has been exceeded, which reduces the congestion in the network.

Function 7 TrafficControl.SendPacket (Packet)

```
function TrafficControl.SendPacket($v$)
    Get $v_p, v_q : v_p, v_q \in v$
    $v$.TTL= $q_{v_p}^{target} - q_e^{current} + q_e^{timestamp}$
    if $v$.TTL $\neq 0$ AND $|\omega_0| <$ limit then
        $\omega_0$.add($v$)
    else
        Drop($v$)
    end if
    for all $v \in \omega_0 : v \in \mathcal{V}$ do
        Get $v_p, v_q : v_p, v_q \in v$
        $v$.TTL= $q_{v_p}^{target} - q_e^{current} + q_e^{timestamp}$
        if $v$.TTL $> 0$ then
            EnqueueToInternalQueue($v$)
        end if
    end for
end function
function EnqueueToInternalQueue($v$)
    Get $v_p : v_p \in v$
    Find($\omega_p$): $p = v_p$
    for position $\in [\omega_p$.head, $\omega_p$.tail$]$ do
        $v^* = \omega_p$.Get(position)
        if $v$.TTL $< v^*$.TTL then
            $\omega_p$.Insert($v$, position)
        end if
    end for
end function
```

3.5 UDP IP Application

In order to evaluate QoSA-ICN performance against UDP IP as a baseline a UDP application was designed and implemented. The UDP application follows a simple client server model. The UDP server listens for connections and the UDP client connects and sends a request. An application class was written to log consumer interest sent time, producer interest receive time, producer data packet and consumer data receive time for each packets. As this is an UDP application data centric security operations that match the signing and verification of NDN packets are not included in the design and implementation. Further details on implementation of this baseline application can be found in the IP application classes section of Chapter 4.

3.6 Design Summary

This chapter introduces QoSA-ICN, a novel model for addressing current challenges relating to the delivery of QoS within an information centric network.

Function 8 TrafficControl.SendToNetDevice ()

```
function TrafficControl.SendToNetDevice( )
    TTLPackets = []
    for $v = \omega_p$.Get(head), $\forall p \in \mathcal{P}$ do
        Get $v_p, v_q : v_p, v_q \in v$
        $v$.priority= $v_p$
        $v$.TTL= $q_{v_p}^{target} - q_e^{current} + q_e^{timestamp}$
        TTLPackets.Add($v$, $v$.TTL, $v$.priority)
    end for
    packet=TTLPackets.Get(head)
    for $v \in$ TTLPackets[] do
        if $v$.TTL < packet.TTL then
            packet=$v$
        else if $v$.TTL == packet.TTL AND $v$.priority > packet.priority then
            packet=$v$
        end if
    end for
    $\omega_p$.Pop(packet)
    NetDevice.Transmit(packet)
end function
```

QoSA-ICN includes 6 new algorithms that achieve QoS-aware multi-hop packet forwarding for time-sensitive applications on vehicular networks. Different priorities are assigned to different kinds of information, e.g., a P1 priority assigned to road side hazards, P2 priority to navigation route guidance, and P3 to infotainment/context aware retail notifications based of users' service subscriptions.

In addition, QosA-ICN proposes 2 further algorithms for a congestion control system, which retrieves QoS information from the packet tag of each network packet. The goal is to enable optimized congestion control and priority-based packet transmission, and achieve a more dependable information-centric approach for deadline-sensitive network traffic.

The proposed models and algorithms are fully implemented and integrated with NDN stacks, as described in Chapter 4. Chapter 5 describes experiments conducted by simulating scenarios using the ndnSIM simulator, which is built on NS3 network simulators. The results are evaluated with respect to baseline algorithms, to quantify the impacts resulting from the work within this thesis and to show the feasibility of the model and algorithms.

Furthermore, the design choices made within the QoS aware ICN can influence the ICN community to towards a general framework for broader set of QoS approaches. This could be done by standardise packets classification mechanisms and trust mechanisms to enable the effective processing of QoS in-

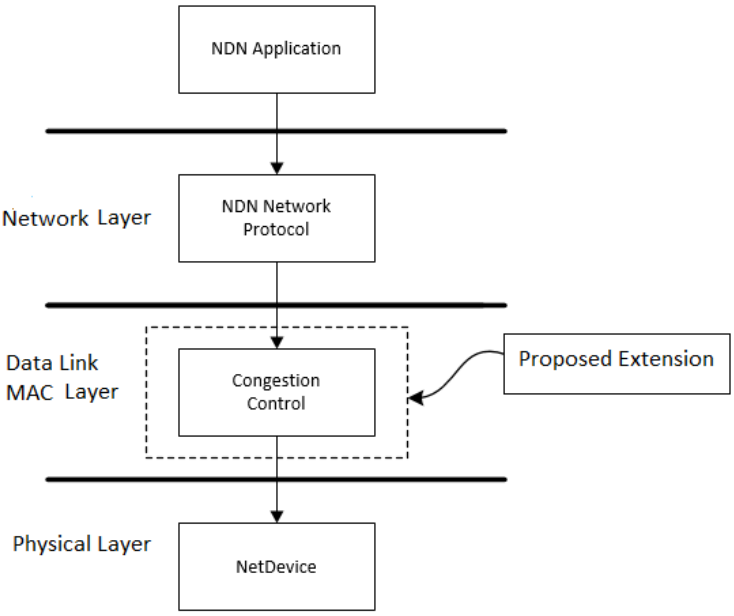


Figure 3.4: NDN Network flow with Congestion Control proposed extension

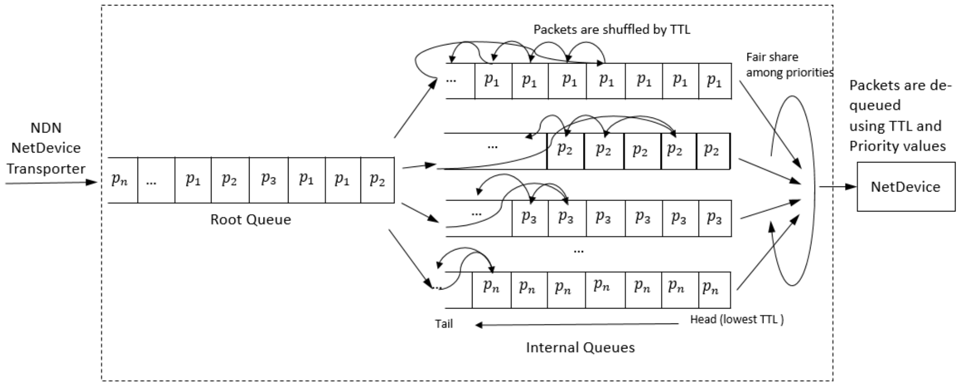


Figure 3.5: Proposed Congestion Control Design

formation. This is this is analogous to an emergency vehicle on the road where every other vehicles pull over and provide clear access. The motivation for this thesis is to design and implement a mechanism to ensure QoS aware packet delivery. Vehicles which are assigned a producer role are configured to produce all three types of priority packets. Focus is not on the content itself rather the delivery within deadlines. These design choices are implemented and evaluated using ndnSIM 2.0 [Mastorakis et al., 2015] which allows for simulation of all NDN protocol operations and maintains packet level interoperability with CCNx implementation [Mahadevan, 2014] [Afanasyev et al., 2014]

Chapter 4

Implementation

This chapter describes the implementation of the deadline-aware data delivery model and algorithms. The implementation is written in C++, integrates with the Information-Centric Networking simulator ndnSIM [Mastorakis et al., 2017] and builds the NDN specification discussed in the previous chapters. Focusing on the key concepts of the QoSA-ICN implementation, the chapter is broken down into three sections. 4.1 presents the implementation approach of QoSA-ICN. Section 4.2 describes the implementation of the extensions to the traffic control mechanisms. Section 4.3 describes the code base extensions which were necessary to implement the IP-UDP application classes. Section 4.4 explains the setup and generation of the realistic mobility scenarios, used as input to the simulations discussed in Chapter 5, it also describe the challenges encountered during the implementation. Section 4.5 describes the implementation of a data pipeline which was implemented to allow for the effective evaluation of experimentation data. Section 4.6 presents an implementation summary for QoSA-ICN. All of these implementations were carried out to fulfil research objectives R01-R04 as described in Chapter 1 of this thesis.

4.1 Implementation Approach

The following section describes the implementation of the QoSA-ICN design objectives in Chapter 3. Both QoSA-ICN and the UDP/IP baseline simulation environments have been built on NS3, NDN and ndnSIM from the following codebases: ns-3 [NS-3, 2019], pybindgen [pybindgen, 2019], ndnSIM [ns 3/src/ndnSIM, 2019]. The implementation approach of QoSA-ICN introduces the following enhancements and extensions to above codebases:

- Interest and Data packets are extended to include TLV (Type Length Value) which are encoded with QoS parameters (priority, timestamp and round trip time limit) and are exposed by the packet header and data naming mechanisms further detail can be found in Data Naming section of this chapter 4.1.1.
- To allow for effective interest packet forwarding, the best-route algorithm as describe in chapter 3 of this thesis is extended to extract and evaluate QoS information from the packet and allow for the broadcast of this information (via node wireless interface subject to communication range limitation) in the absence of pre-built routes. The implementation allows for dynamic forwarding to nodes that were not on the original interest path which is implemented by extending the NDN packet forward service which is the network forwarding daemon (NFD).
- In order to allow for a QoS aware Data packet implementation, the implementation ensures that the QoS info is copied from the original interest packet to the data packet as part of the data packet formation process.
- Optimised traffic control for these packets is included through the introduction of a cross layer method to optimise congestion control and priority-based packet transmission with the aim to enable a more dependable Information centric approach for deadline sensitive network traffic (note this optimised traffic control solution can be enabled or disabled via a code flag within this implementation).
- In order to evaluate QoSA-ICN performance against UDP IP baseline a UDP application was designed and implemented.
- Many challenges were encountered during this implementation especially in the areas of mobility simulation and tools for efficient experimental data evaluation further details of which can be found in Section 4.4.1 and 4.5 of this thesis.

As discussed in Chapter 1 of this thesis the chosen use case environment for this implementation is a dynamic, infrastructure less, vehicular ad-hoc networking (VANET). In such a scenario it is not feasible to establish data delivery paths in a guaranteed manner. This implementation broadcasts the interest packet to a number of moving vehicles within communication range. In turn the first data packet send from a producer fulfilling the consumers request will be accepted. The embedded QoS information with specified deadline introduces a time to live and avoids flooding the network. The implementation leverages the default NDN security features. Simulation of communication technologies including features such as propagation loss model and delay propagations model within the WiFi stack are included by default to evaluate the performance of the QoSA-ICN algorithms aganist comparative baselines. The QoSA-ICN implementation also allows for the processing on non QoS aware packets. These packets are processed as NDN packets (interest and data) and may generate a large number NACK responses in a dynamic wireless scenario with a no-route message being passed back to the consumer due to the absence of routing information in the FIB. The implementation of the design specifically focuses on the opportunity to embed QOS information in content and have the network act on it, and it does not attempt to simultaneously evaluate other potential benefits of NDN such as multipath forwarding and data centric security, or other extensions for mobility support such as WLDR.

4.1.1 Data Naming

QoSA-ICN uses NDN's default feature for matching long prefixes, there has been no alteration carried out as part of this thesis to override the prefix matching process that exists in the current NDN implementation. For example the NDN consumer application generates unique namespace with prefixes consists of base prefix 'ndn-qos', priority type 'p1' and the unique identification string '/%FE/%01'. The NDN producer application in the simulator matches '/ndn_qos/p1/' to generate data packet for '/ndn_qos/p1/%FE%01'. In this implementation once a node has the role of a producer they are configured to supply any type of priority packets (P1-P3). The new algorithms classify the priority of requests P1-P3 (P1 highest priority with shortest time to live (TTL)) value.

4.1.2 Code base implications

A substantial amount of new code was added to, and existing code modified in, the NDN code base, to implement the design. All new QoSA-ICN code has been fully integrated with NDN with the aim of fulfilling all research and design objectives, and realising a QoS-aware information-centric network. The code also provides an environment where the evaluation of the research hypothesis can be performed, to answer the research questions. Appendix D describes thiS new code in detail and also the modifications needed to baseline code base for this implementation.

4.1.3 QoSA-ICN Class Descriptions

The following section contains a description of the new and extended classes, needed for the QosA-ICN implementation.

1. **Class QoSInfo**: This class implements the QOSInfo objects by inheriting TLV:Error. The QoSInfo object is used to set the metrics, such as packet timestamp, priority type, qos time limit. See Fig.4.1.

2. **Class Data**:

   This class represents a data packet. See Fig.4.1.

3. **Class Interest**:

   This class represents an Interest packet.

4. **Class NDNQoSPriorityQueueDisc**: As illustrated in Fig.4.1, Linux $pfifo_fast$ is the default priority queue enabled on Linux systems. Packets are enqueued in three FIFO droptail queues, according to the three priority bands from the packet priority definitions. The queue disc capacity, i.e., the maximum number of packets that can be enqueued in the queue disc, is set through the limit attribute, which plays the same role as txqueuelen in Linux. This requires an internal queue of type QoSPriorityQueue, each with a capacity equal to the limit, created by default. A user can provide queues, but there must be three, and operate in packet mode, each have a capacity not less than the limit. No packet filter can be provided.

5. **Class QoSPriorityQueue**:

   A QoS priority queue distributes packets to the internal queues as per priority type, and also shuffles the order of the packet in the queue based on the packet's time to live (TTL). Packets are dequeued from the head of the queues and compared for TTL and priority for transmission.

6. **Class NetDeviceTransport**: ndnSIM-specific transport

The relationships between these classes can be seen in Fig.4.1 and Fig.4.2.

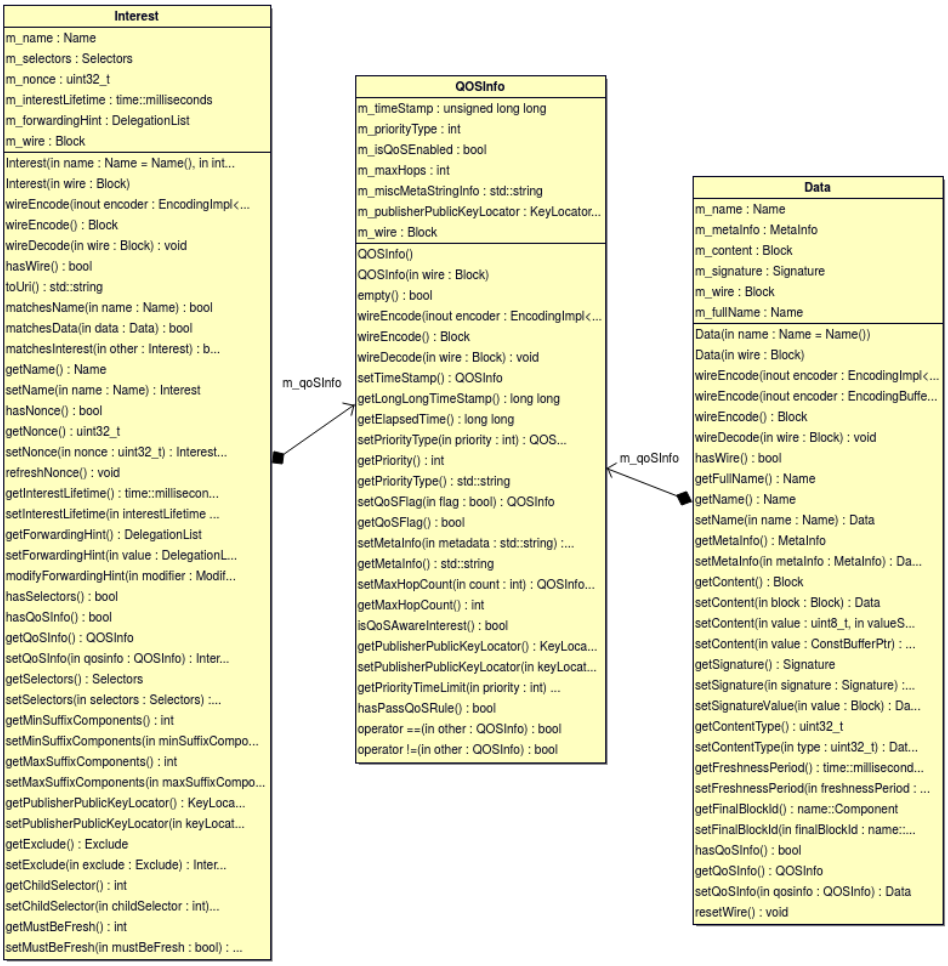


Figure 4.1: QoSA-ICN class diagram for Interest and Data Packet Extensions.

4.1.4 Network Forwarding Daemon(NFD)

QoSAwareBestRouteStrategy inherits the Strategy class and implements the QoS-aware forwarding rules. Strategies can be configured using StrategyChoiceHelper. The NDN forwarder loads the QoSAwareBestRouteStrategy during run time, for making forwarding decisions. See Fig.4.2.

4.1.5 QOSA-ICN packet sequence flow

Fig.4.3 illustrate the flow of packets within the basic implementation of NDN (this flow is applicable for both interest and data packets). An augmented flow is presented in Fig.4.4 for QoSA-ICN, which illustrates that QoS is considered and processed by the system at each stage of packet delivery.

4.2 Traffic Control

NDN NetDeviceTransport aggregates the TrafficControlLayer. QoSPriorityQueue and NDNQoSPriorityQueueuDisc extend the NS3 Queue class and QueueDisc class, respectively. NdnQoSQueuDiscItem is inherited from QueueDiscItem, which holds the packet with associated metadata information. See Fig.4.5.

Mobility Parser Class MobilityTclFileParser implements mobility-tcl-file-parser.h, pre-processing of mobility trace files is essential to read and save start/stop time of each node in the simulation environment. In order to enable and disable the node network activities when the node enter and leave the road segments respectively. See Fig.4.6

Packet Tag Class NDNQosTag implements ndn-qos-tag.h to set and retrieve QoS metrics in order to expose the metrics at data link (MAC layer) for traffic control. See Fig. 4.7

4.3 IP application classes

The **AppDelayTracerQos** implements ip-app-delay-tracer-qos-exp.hpp, main roles includes logging of consumer interest sent time, producer interest receive time, producer data sent time and consumer data receive time for each packets in IP simulation environment. See Fig.4.8

The **IPApp**is a base class definition for IP app by extending the Application class. IPConsumer is derived from IPApp which implements the UDP IP

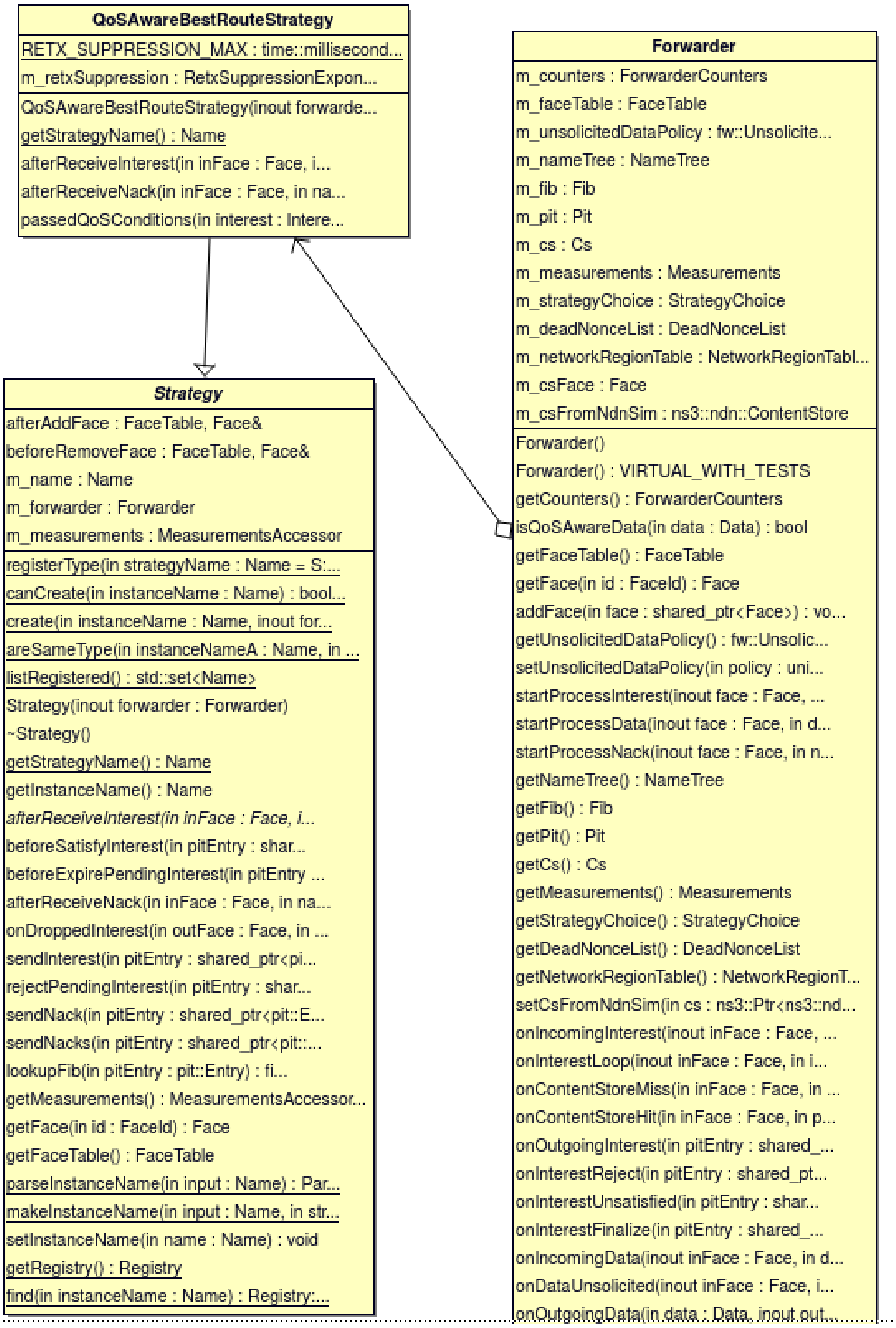


Figure 4.2: QoSA-ICN class diagram for forwarding.

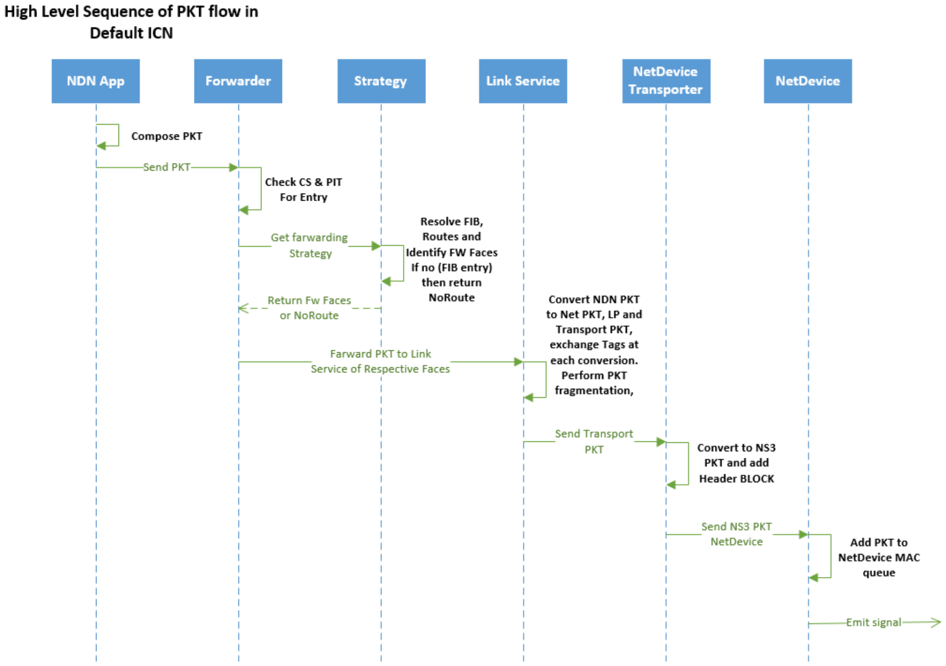


Figure 4.3: Packet flow for NDN.

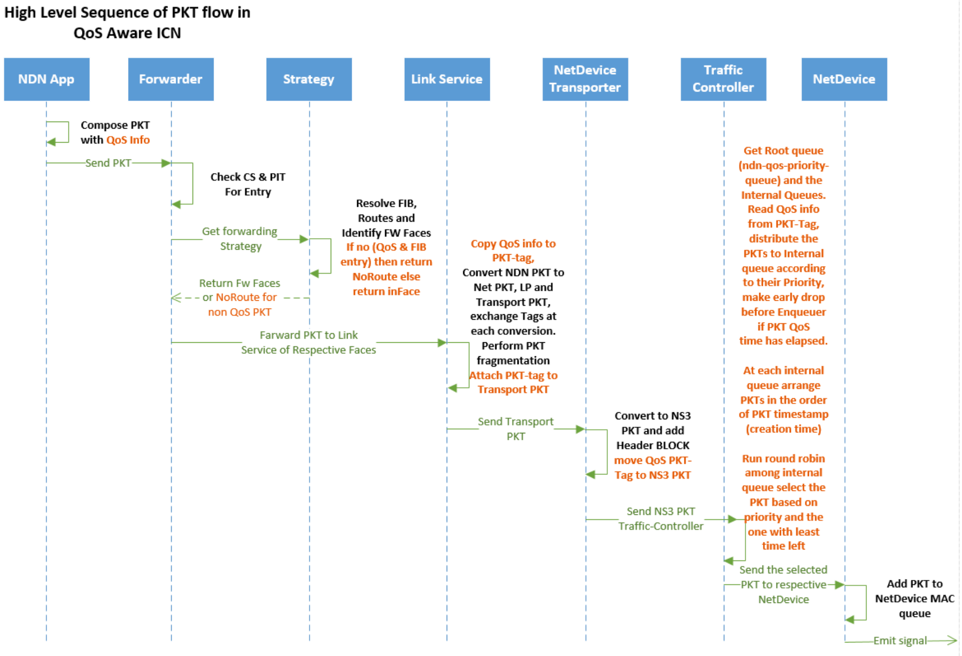


Figure 4.4: Packet flow for QoSA-ICN.

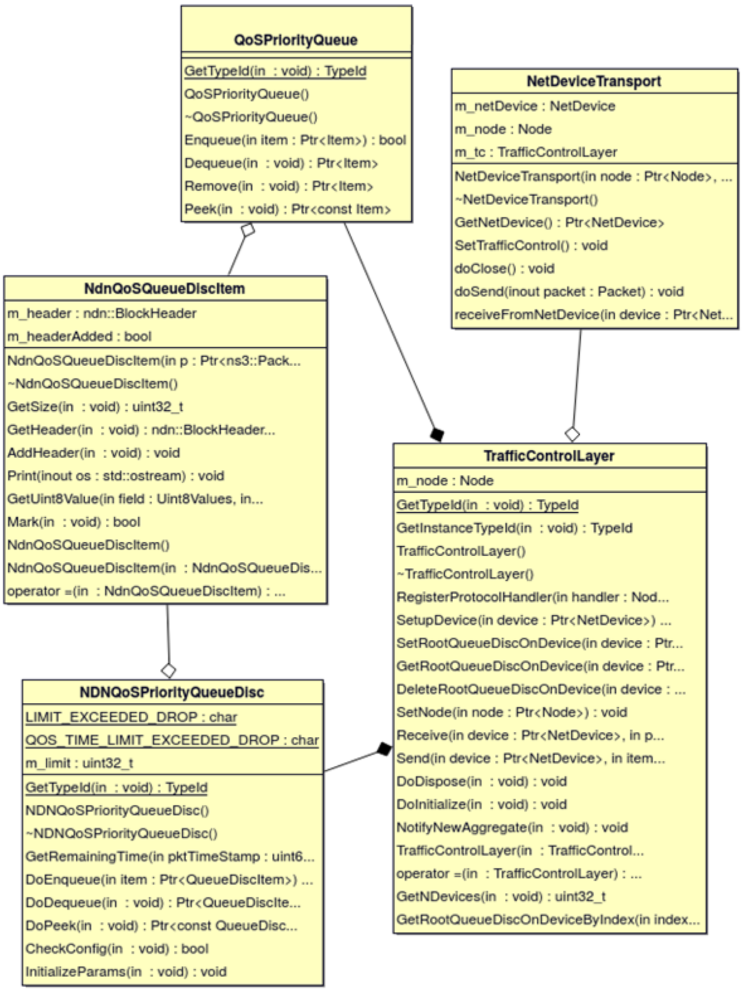


Figure 4.5: QoSA-ICN class diagram for traffic control

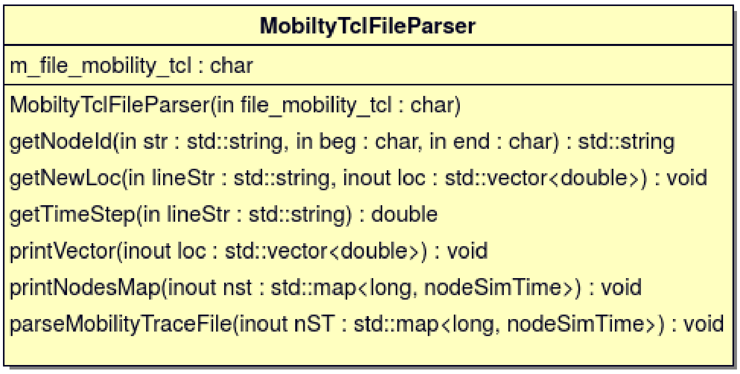


Figure 4.6: Mobility Parser Class Diagram

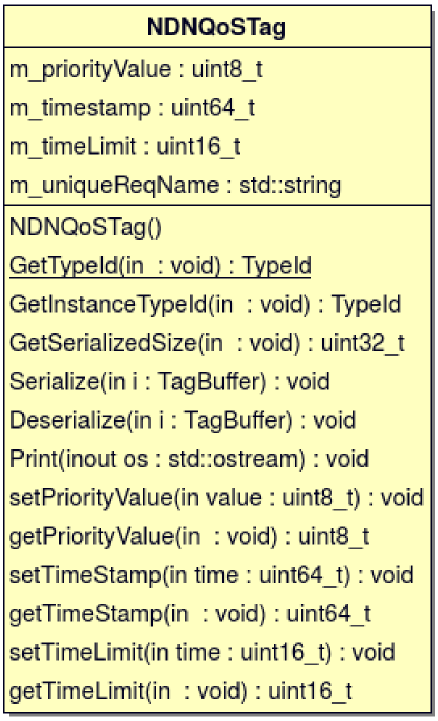


Figure 4.7: Class to expose QoS to MAC for traffic control

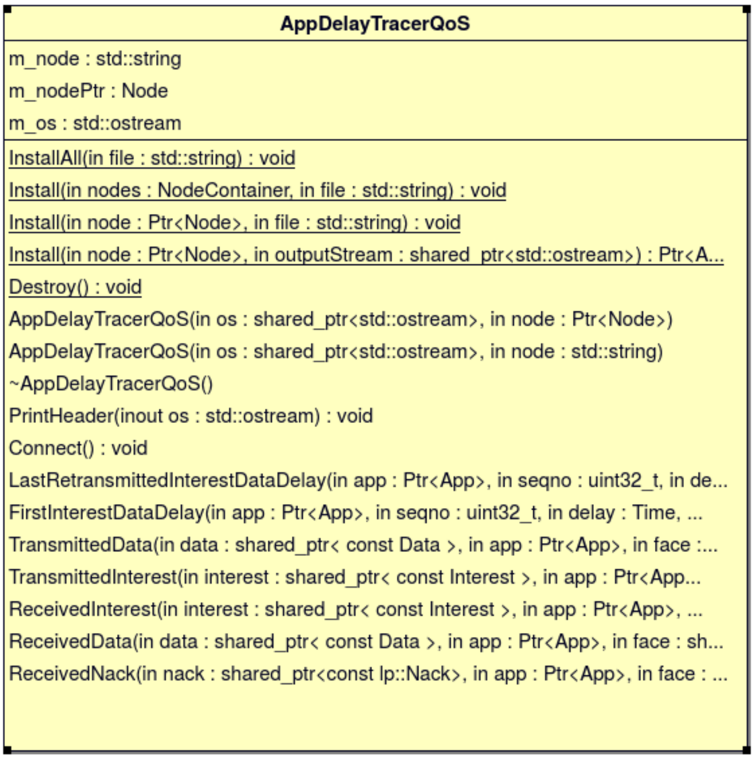


Figure 4.8: AppDelayTracerLog

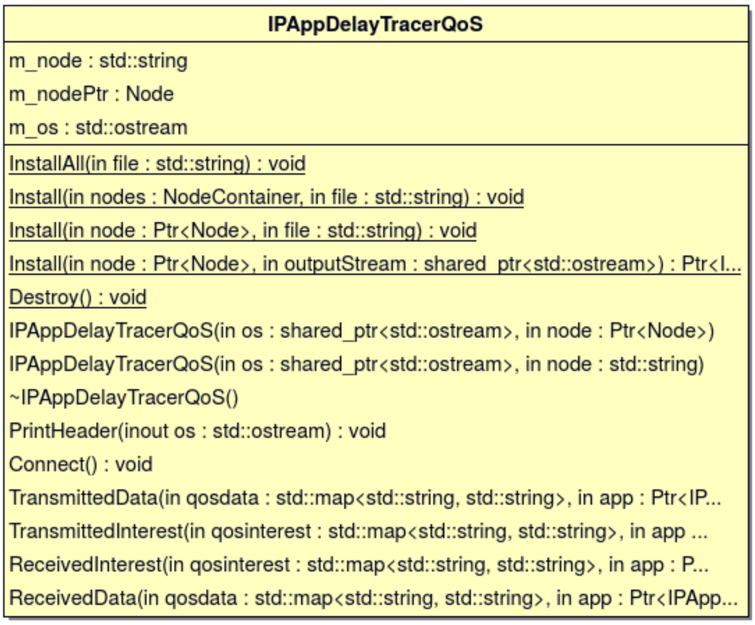


Figure 4.9: AppDelayTracerLogIP class

consumer application. IPConsumerCbr is extended using NDNConsumerCbr which is responsible for scheduling the Interest packet generation. See Fig.4.10

4.4 Mobility implementation

The research for this thesis used the ndnSIM simulator [Mastorakis et al., 2017] to realise the vehicular network scenario. ndnSIM is an an open-source NDN simulator, based on the NS-3 simulation framework [NS-3, 2019]. NS-3 is a widely-used, discrete-event network simulator for Internet systems, also open source, and licensed under the GNU GPLv2 license. ndnSIM is publicly available for research and development, and includes an NDN Network Forwarding Daemon, ndn-cxx source code, and an NDN simulation layer, with a number of plug-and-play simulation scenarios. Figure 4.11 shows the process of exchanging packets between two simulated nodes, through NDN, NDN-cxx and the NS-3 stacks. The ndnSIM implementation was extended to support QoSA-ICN. The NDN simulation environments were built using ndnSIM version v2.5 and the UDP/IP environment was built on NS-3 version 3.27.

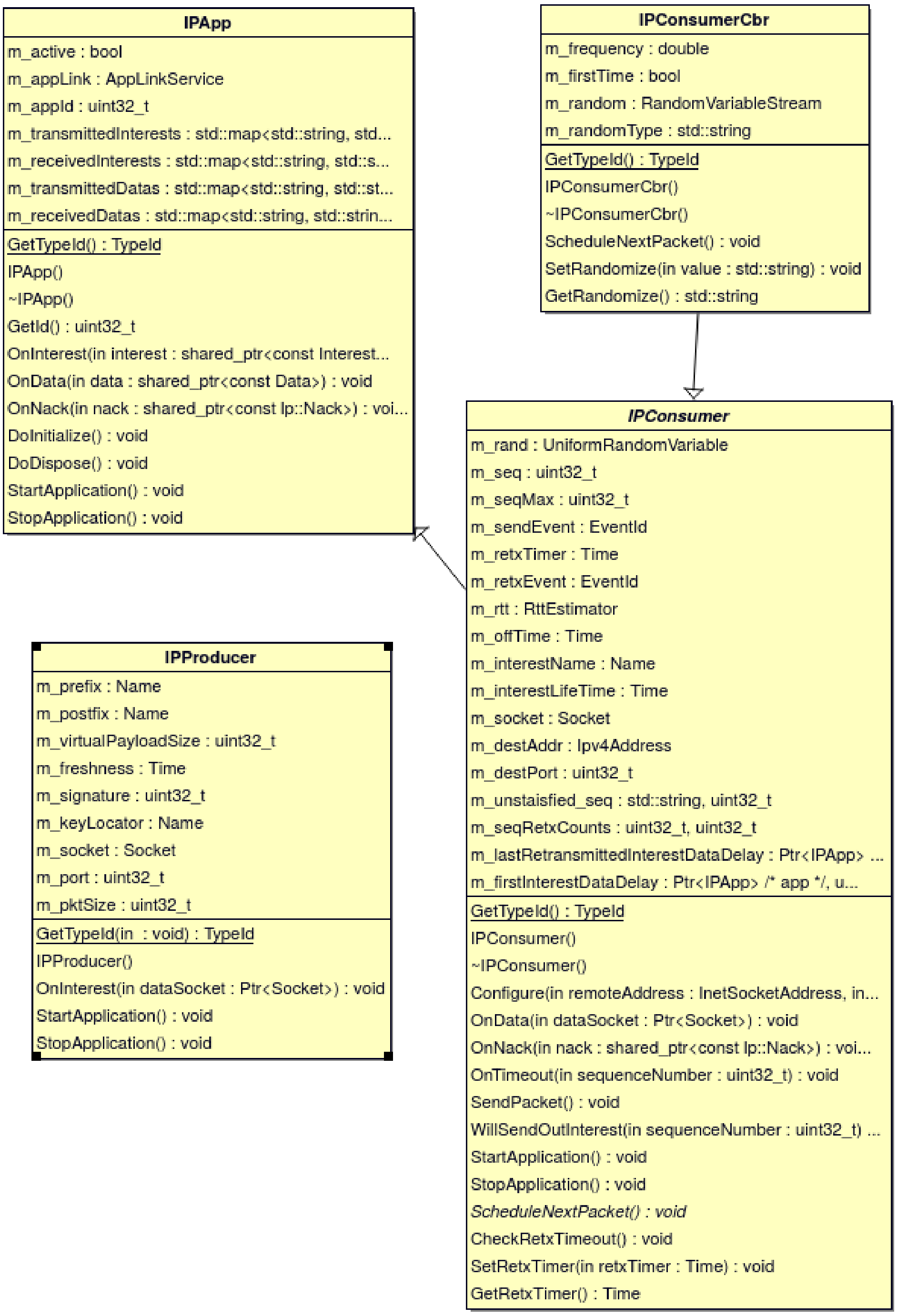


Figure 4.10: IP(UDP)application class

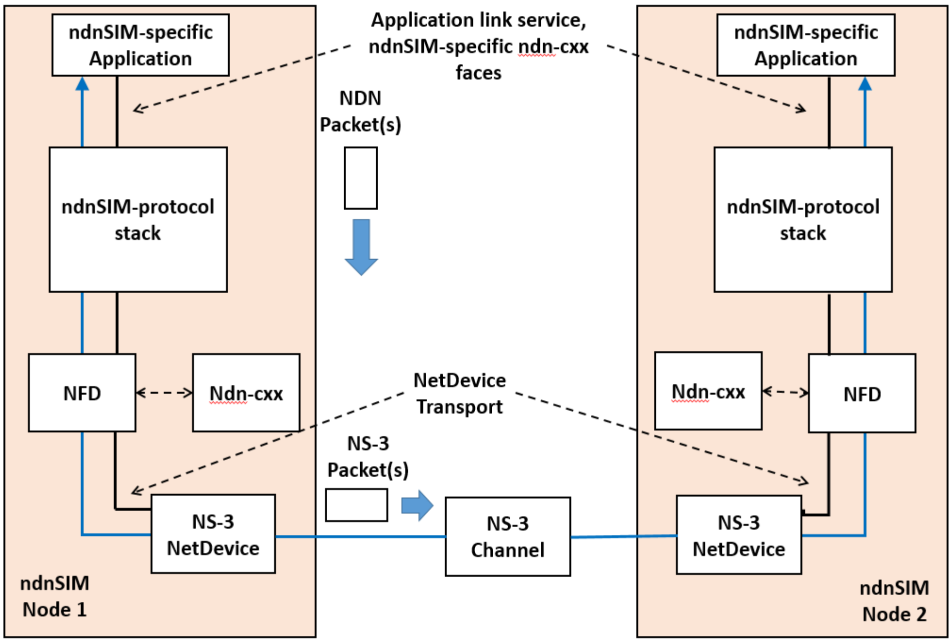


Figure 4.11: NDN packet exchange between two simulated nodes [Kalogeiton et al., 2017].

The Simulation of Urban Mobility (SUMO) simulator [Krajzewicz et al., 2012] and OpenStreetMap (OSM) [Haklay and Weber, 2008], were used to generate realistic mobility scenarios. SUMO is an open source, highly-portable and continuous road traffic simulation package, designed to build large road networks. For this work, SUMO was used to generate mobility traces with the goal to have realistic vehicle movement and in-vehicle application behaviour [Sommer et al., 2008]. Open Street Map was used to create a real-world map of a 2km dual-carriageway road section (divided highway), which is a class of highway with carriageways for traffic travelling in opposite directions, separated by a central reservation. This section of road did not have any exits. For the required scenarios, 27 mobility trace files were created, simulating traffic flow for the 2km road length, with varying vehicle density, speed, and experiment duration. Vehicle densities are calculated based on maximum road vehicle capacity, as follows.

Road Vehicle Capacity

Estimation of maximum road vehicle capacity is important to simulate practical road traffic scenarios of varied vehicle density (e.g., dense, medium and sparse), and validate the model at all possible boundaries within the VANET. Given the definitions of the cars and the road segment in Section 3.2.3 of Chapter 3, the road capacity ψ and the density ρ were calculated following Eq. 3.4 and Eq. 3.5, respectively.

Mobility trace generation

The traffic flow traces generated in SUMO are used as input to ndnSIM, which creates the experimental output data file for analysis. The SUMO traces model a number of attributes relating to each vehicle's movements and speeds, and the length and shape of the road they run on. These are input to ndnSIM at the start of each simulation, with different values that influence how the simmulation is executed. The simulator then installs an NDN software stack on each node and starts the vehicle flow.

Traffic Flow Attributes

Each traffic flow trace models a different traffic scenario, created by varying the flow attributes. The flow attributes are:

- Road Length
- Experiment Duration
- Vechicle Speed
- Vechicle Density

Chapter 5 describes the variation of these attributes, relevant to the particular results under analysis.

Vehicle Density

Vehicle density is the number of vehicles on the road at any one point. Different proportions of the road's vehicle capacity are used to calculate road density. All traces maintain a strict vehicle density for the lifetime of the simulation. There are three different densities per speed, namely:

- Dense (100% of max density)
- Medium (50% of max density)
- Sparse (10% of max density)

As per equation 3.4 and 3.5 respectively k represents density values as various scenario speeds 4.12 Density strictness is enforced by how much the density is allowed to vary at any point in the flow. Most traces allow a variance of plus or minus 5% from the target density. See further discussion in Section 4.4.1.

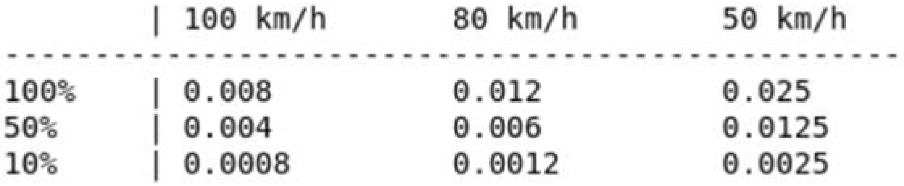
Traffic Flow Trace Scenarios

Initially three different road lengths were considered for the trace generation: 200m, 2,000m and 60,000m, these road lengths were selected to simulate suitable conditions for WiFi and DSRC communication ranges. In addition, three different vehicle speeds were considered: 50km/h, 80km/h and 100km/h, selected to evaluate slow, medium and fast driving speeds.

For example, below are the list of all traces that are used for the 2,000m implementation. As per equation 3.4 and 3.5 respectively k represents density

- 2000m-100kmh-0.0083k
- 2000m-100kmh-0.00415k
- 2000m-80kmh-0.0124k
- 2000m-80kmh-0.0062k
- 2000m-80kmh-0.00124k
- 2000m-50kmh-0.0263k
- 2000m-50kmh-0.01315k
- 2000m-50kmh-0.00263k

Each trace above is created with three different experiment durations, 300sec, 600sec, 1200sec. So the trace 2000m-100kmh-0.0083k, is in fact, 2000m-100kmh-0.0083k-300s, 2000m-100kmh-0.0083k-600s and 2000m-100kmh-0.0083k-1200s.



```
        | 100 km/h       80 km/h        50 km/h
--------------------------------------------------
100%    | 0.008          0.012          0.025
50%     | 0.004          0.006          0.0125
10%     | 0.0008         0.0012         0.0025
```

Figure 4.12: Density values

Creating Traffic Flow Traces

The output of each of the above scenarios creates a trace which is formatted as a floating car data (FCD) file which is used as input to ndnSim. One of SUMO's strengths is its support for creating realistic traffic movements, and it is used here only to create the vehicle flow models. It is not used to actually simulate the traffic scenarios, as it is more appropriate to use ndnSIM for the simulations.

The traces relying on SUMO are created with a number of python packages, which guide the process to ensure correct traffic flow traces are created, as follows:

- **SUMO generator**: This package takes a road template and modifies the flow attribute, to create the different vehicle scenarios. The output of this package is a SUMO mobility trace, which is formatted as an FCD file, with summary data of the traffic flow.
- **Trace Validator**: After each trace is created, it is passed to the SUMO validator. The SUMO validator uses the summary data output from the SUMO generator to create csv files for analysis. These summary files have values for vehicle speeds, traffic density and average traffic leaving and entering a road for each step of the model. The csv files are validated to ensure the traces conform to the speed and density requirements. Typically at this point, the traces have a duration of 30 minutes, which ensures there is a ten minute window in the whole simulation where the density is most consistent. Those traces with the best time ranges for density consistency are output.
- **Trim trace**: Finally, the traces are split into three durations. This is done using the output from the trace validator.

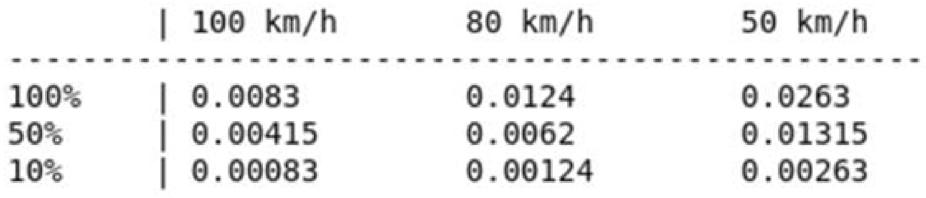
Input to simulation

In addition to the traffic flow traces, the experiments need to be configured in order to run the large-scale simulations. An experiment runner package has been implemented, which runs the simulations. Written in python, it takes the traffic flow traces and the configuration details and uses these to execute the simulation. The experiment runner counts the number of cores on the server, and launches a different experiment onto each core. This process needs to be balanced with the amount of RAM on the machine, to ensure that a machine does not halt through over consumption of resources.

Simply, the simulation involves taking the traffic flow traces and passing them to the ns-3 NDN simulator (i.e., ndnSIM) to be executed. Once finished, the simulation will produce a number of output traces containing the resulting data from the simulation execution. This experimental data is the key output, and is used for the analysis discussed in Chapter 5.

4.4.1 Simulation Challenges

Most traces allow a variance of plus or minus 5% from the target density. This is a very tight density variance, especially on roads with shorter lengths. A short road has less capacity, which means that vehicles leaving or entering the road have an increased impact on the rate of change of the proportion of the total capacity, i.e., the density. In fact, this becomes impossible for some traffic flow scenarios with high speed, sparse density and short length. Though SUMO is tightly controlled, it was not designed to enforce such tight restrictions. The vehicles flexed a little, varying their speeds slightly. Though this variation was slight, it was enough to push many of the densities outside of the 5% requirement. To accommodate these scenarios, a new method for creating the traces was developed, with the strictness relaxed to 10%. This method which was implemented as a python module does not allow with vehicle flex and every car that leaves the road is immediately replaced by a new vehicle entering the road, all at a consistent rate. This method adopted the following densities which were consistent with Fig. 4.12 with the along with a 10% variance of density, was applied to one mobility trace 200m-100kmh-0.00083k. All other traces followed the 5% density variation and were generated using SUMO. During the design phase of this thesis , it was considered to vary the road lengths to include 200m, 2000m and 60000m. However it became very evident that certain scenarios would not be feasible such as 200m-100kmh-0.00083k and 200m-80kmh-0.00124k .Due to speed and short road distance. Simulation failures on certain scenarios after a few days of execution which has a large number of nodes and dense topology structure on both NDN and OoSA-ICN environments was encountered. The error message is “free(): invalid pointer: —”.This has been reported to ndnSIM support for further investigation. This resulted in 15 failed experiments out of 1080 experiments which is a 1.39% experiment failure rate overall.



```
        | 100 km/h      80 km/h       50 km/h
-----------------------------------------------
100%    | 0.0083        0.0124        0.0263
50%     | 0.00415       0.0062        0.01315
10%     | 0.00083       0.00124       0.00263
```

Figure 4.13: Adjusted density values

4.5 Data Pipeline

Due to the volume of experimental produced by the 1065 experiments, which were run 10 times, it became very evident that a structured data pipeline was needed to complete the evaluation in an effective and efficient manner. The function of the pipeline, which is written in Python, is broken down as follows: a) parse all data log files generated from simulation; b) analyse all algorithms; c) generate statistical analysis and plots.

The Pipeline File Tree is broken down as follows and the following section will explain the components in greater detail.

Pipeline/

- masterpipeline.py
- common.py
- Config.py
- Generatestatisticalanalysis.py
- Parselogfiles.py
- Analysealgorithms.py
- Path.py
- Processtracefile.py
- Walk.py
- PlotsData/
- ProtocolSummary.csv
- Plots

- TestScripts
- config.cfg
- Go.py

4.5.1 Pipeline Implementation

Go.py is the main starting point of the application. Executing go.py will set up the log file and call masterpipeline.py to begin the pipeline. This log file setup will only be used in the current process space. while spawned processes also have a log configuration **Config.cfg** The configuration file for the pipeline, containing the configurable options for the pipeline. **General**

- protocol
- runsections
- minTraceSizetoProcess

Comparsion

- iterations
- roadsize
- heatmaps
- protocols
- algorithms

MasterPipeline.py is called from the main starting point (go.py) and is used to orchestrate what to run. The sequence is as follows:

- Reads in the protocol list from the config file.
- Reads the sections of the pipeline to run from the config file, which allows specific actions to be performed.
- Walk Directory calls: a) walk.py, to walk mounted protocol directories for new trace files and save them to a txt list; b) parselogfiles.py, to find the expected number of cars for each scenario; c) processtracefiles.py, to copy, load and analyse new trace files. The number of cars list is passed to this script to allow a sanity check for u trace files.

- Combines all the summary files for each protocol in the protocols list to the combined summary csv for further analysis.
- Calls analysealgorithms.py for analysis of the experimental data, in particular, the algorithms' performance with regard to the experiments' parameters
- Calls generatestatisticalanalysis.py to compare each algorithm's the performance, listed in config file

Walk.py

For a given protocol, this script will walk the directories in the mounted directory and find all trace files (Note: the custom qos trace string is hard coded - this can be changed if the user wants to pull a difference trace file). Once all trace files are found, the script compares this list with the list obtained from the most recent spreadsheet in the results directory (if one exists) and incomplete and failed trace lists. If a file exists, the script will compare the lists to ensure it was not already processed and save the new traces to a local protocol directory list "qosFileList.txt" (automatically created). If the file does not exist, all trace files found will be copied to a local directory list "qosFileList.txt" for analysis. The script also counts the number of files found and the total number of files for that protocol. The config.cfg has a configuration of "minimumTraceSizeToProcess", which can be used to limit the number of traces to process (if 0 then no filtering is applied). This was introduced to allow fast processing of smaller-sized files initially, as there were some traces over 1GB in size. The larger files were loaded after having a dataset to perform initial analysis on.

Parselogfiles.py points to the mounted directory for a given protocol and parses log files for experiment names and their associated number of cars. The experiment data frame is returned.

Processtracefiles.py creates a results directory for a given protocol (if it doesn't already exist). The latest summary spreadsheet is either loaded, or a new one created. The list of trace files to load is read, and a check performed that the trace has not already been processed. If the list is not empty, the script will iterate through it, performing the following:

- The trace file is copied from source to local storage
- The trace file is loaded, and summary statistics calculated

- The spreadsheet dataframe updates every 20th record (to reduce locking in the multi-threading environment)
- The local trace file is removed after processing is complete.

The updated spreadsheet is merged with the given list of cars and a check is done to list any partial experiments. There is a check that the number of consumers matches the expected proportion (threshold +/- 5% with exceptions of +/- 10%). Any partial traces are removed from the spreadsheet before it is saved to file.

Analysealgorithms.py The algorithms are analysed independently of each other with trends associated with the experiment parameters exposed. Logs for these separate processes are not visible in the main log. They are logged in separate log files with the following filename "pipelineanalysealgorithmsPID.log". For each algorithm, a summary spreadsheet is loaded, and the following Linear model subsets calculated:

- Implements linear model with regard to Parameters for each individual algorithm;
- Returns coefficients for each parameter and R-score;
- Subsets the data by most influential parameter (highest coefficient) and re-evaluates linear model to see if R-score improves;
- - Results are output to the results directory in a txt file labelled by the algorithm.

Parameters are density, speed, producers and experiment length. A full list of data pipeline parameters are listed in Chapter 5, as part of evaluation discussion.

Generatestatisticalanalysis.py This script combines specific algorithms to compare performance across the scenarios. This method is multiprocessed (by processing each road length in a separate process) to speed up graph generation by optimizing processing and bandwidth usage. Logs for these separate processes are not visible in the main log. They are logged in separate log files with the following filename "pipelinegeneratestatisticalanalysis{PID}.log". ANOVA and Tukey tests were implemented to identify the scenarios where the algorithms performed significantly differently from one another (p-value 0.05 or 95% confidence interval). The mean differences were also plotted using heat maps to visualise trends associated with difference scenarios. These results will be discussed in the next chapter.

4.6 Implementation Summary

This chapter describes the overall artefacts were implemented to realise QoSA-ICN and fulfil the design objectives which were defined in chapter 3 of this thesis.

QoSA-ICN is mainly realised as a number of C++ classes which when integrated with NDN enable a) A method to allow for QoS aware packet extension to both interests and data packets b) A method to allow for QoS Aware best route forwarding withing the Network Forward Daemon (NFD) c) A method to allow for a cross layer approach to achieving optimised traffic control in the datalink MAC layer

Due to the volume of data produced by experiments and to aid with statistical analysis a comprehensive data pipeline was implemented.

The next chapter based on this implementation describes the evaluation of QoSA-ICN through ndn-sim based simulation and presents the evaluations of the results.

Chapter 5

Evaluation

This chapter evaluates how well QoSA-ICN addresses the thesis research objectives(see Section 1.3) and hypothesis by providing an information-centric approach to deadline-aware data delivery at the network edge. The combined results will indicate the extent to which the delivery of time-sensitive data in dynamic environments can be improved by employing an information-centric, QoS-aware data delivery mechanism, compared to existing baselines (RQ1) and also the extent to which the delivery of time-sensitive data in dynamic environments can be improved by adding QoS awareness to network traffic control mechanisms, compared to existing baselines (RQ2). This chapter outlines the evaluation of the protocol in comparison to other solutions and for a variety of settings. In particular, it examines the impact of density, speed, connectivity, proportion of data producers and experimentation duration have on the performance of QoSA-ICN and the baselines. For this, the chapter is organised in two parts: The first part, Section 5.1 Experimental Setup, describes the general settings, evaluation scenarios, metrics and baselines used in the experiments as well as threats to the study's validity. The second part, Section 5.2 Results and Analysis, presents and analyses the results of the experiments.

5.1 Experimental Setup

The evaluation of QoSA-ICN is based on simulations as they allow for controllable, repeatable and scalable experiments. Despite being widely used, simulations limit the validity and this will be discussed further in Section 5.1.3. SUMO and ndnSIM were used to simulate a dynamic vehicle network. SUMO trace generation is used as an input to ndnSIM and has been discussed at length in Chapter 4. The focus in this experimental setup will be on ndnSIM, which

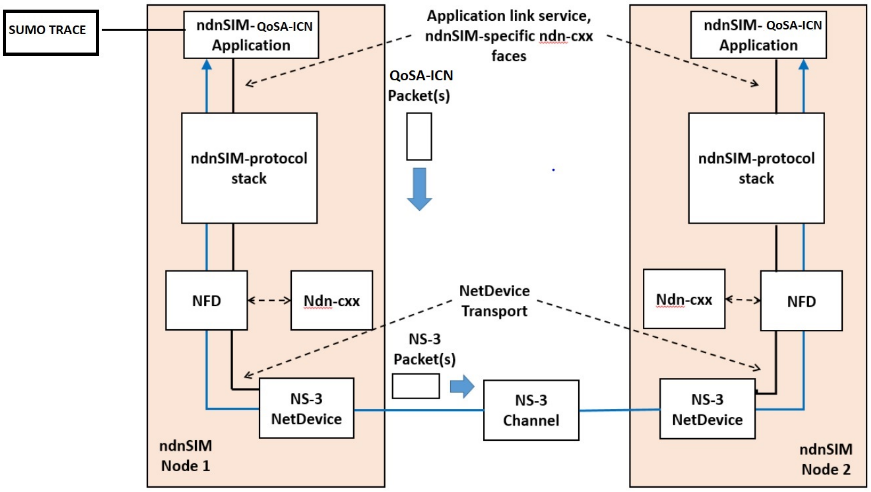


Figure 5.1: Experimental setup

runs the experiments. The following sections explains the general settings and the specifics of the evaluation scenarios. It then outlines performance metrics, and introduces the baseline implementations. WiFi stacks in the simulation are configured with default settings, which are captured in the simulation parameters table. The simulation models are built such that there is significant parameterisation. This allows for the testing of algorithms for a range of input values, as listed in Table 5.1, and described further in this section. Figure5.1

5.1.1 Evaluation scenarios

108 evaluation scenarios were developed as a result of combining four variables, i.e., vehicle density, vehicle speed, percentage of vehicles acting as producers, and simulation duration on a 2km road length. For example, one scenario is a sparse environment with vehicles travelling at 50 km/h, with 20% of vehicles acting as producers, and a simulation length of 300 seconds. See Section 4.4 for discussion on the calculations for sparse, medium and dense vehicle densities. Interest packets are sent by consumers at a rate of 5 per second. This consistent rate was chosen as it became very challenging within ndnsim to send more than this number is a dense environment. The 5 packets are generated based on the following ratio 3:2:1 (P1:P2:P3). Consumers will send request information to all vehicles in their geographical range. Data packet rates are generated inline with interest rates as a data packet is never generated unless a corresponding interest has been sent. ndnSIM imports SUMO trace file which

Table 5.1: Experimentation - Simulation Parameter Settings

Parameters	**Value**
Number of mobility Traces	27
Number of scenarios	108
Number of iterations	10
Vehicle density	sparse, medium, dense
Speed (km/h)	50,80,100
Experiment duration (secs)	300,600,1200
Wireless Connectivity	WiFi, DSRC
Wireless Operating Mode	Adhoc Wifi Mac
Transmission Power	10 dBm
Propagation Loss Model	Nakagami Propagation Loss Model
WiFi Transmission Range	100 meters
DSRC Transmission Range	1000 meters
WiFi Model	802.11a
OFDM	5 GHz band
Propagation Delay Model	Constant Speed Propagation Delay Model
Bandwidth	54 Mbps
Consumer/Producer (%)	20/80, 40/60, 60/40, 80/20
Consumer Distribution	Random
Producer Distribution	Random
Content Store (CS) Policy	LRU
Content Store MaxSize	1000
Data Packet Payload Size	1200 Byte
Packet Priority Types	P1, P2, P3
Packet QoS RTT Limits (ms)	100 (P1), 200 (P2), 300 (P2)
Interest frequency	5 packets/sec
Packet priority distribution	50% (P1), 30% (P2), 20% (P3)

contains nodes' location relative to time steps. NS3 mobility module parse the trace files and use it for various operations. Location based data naming is not the focus for this thesis instead focus has been directed towards priority based packet delivery. The solution allows for location based naming to be included in the namespace by vehicles on board gps sensors. In the simulation setup producer nodes are configured to produce and serve all three types of priority data. The data generation patterns are following the same Interest expression pattern however, the NDN application used for producing the data (1200 bytes of virtual payload) manages the identification of data, freshness, signing and verification by their unique namespace as shown in the example below. In addition to the virtual payload, the producer node inserts a copy of QoSInfo object into the data packet from the subsequent Interest packet. The forwarding algorithm on the data return path extracts the QoSInfo and processes the information to make forwarding decision. As discussed in Section 4.4.1, there were simulation failures on certain scenarios after a few days of execution, particularly where there were a large number of nodes and dense topology structure. This happened for both NDN and OoSA-ICN environments. In total, 1,080 experiments were run for the 2000m scenarios. 1,065 executed successfully. Post analysis was carried out on the 15 failed experiments using an internal Intel statistical analysis tool to examine if it was possible to predict the missing values. The tool takes a gradient-based tree approach (GBT), which takes 70% of the data set for training, then predicts the values for the next 30%. The process improves its prediction over a loop of 50 iterations, with the goal to get the prediction error as low as possible. See Appendix A, Figure 5. As the failed experiments represent only 1.39% of the total experiments, a decision was made to focus evaluation efforts on the successful 1065 experiments. Ensuring effective Cache utilisation for satisfying information request is not in scope for this thesis. Each nodes Content Store (CS) which is a cache of information received is dependent on experimentation simulation parameter settings per scenario 5.1. When analysing the impact of experiment duration, the results would suggest in practice caches of nodes using wifi connectivity within a sparse environment would take 10 minutes (600sec) to achieve a performance increase. However as cache utilisation is not in scope for this thesis further experimentation would be needed to focus on caching behavioural patterns and utilisation to understand this in more detail. 5.4

5.1.2 Baselines

The experiments measure the end-to-end data delivery response time of QoSA-ICN, compared against two baselines, for each scenario. In particular, the network simulation protocols evaluated are QoSA-ICN, NDN and UDP/IP. For each, the connectivity types implemented were standard WiFi 802.11a (100m range) and DSRC 802.11p (1000m range), giving six results sets, providing an answer to RQ1.

In addition, a further four result sets capture the experiments to measure the impact of the traffic control implementation at the data link layer, designed to answer RQ2. In particular, traffic control was added to the UDP/IP DSRC and WiFi algorithms, and to the QoSA-ICN DSRC and WiFi algorithms. It did not make sense to measure a traffic control version of base NDN, as traffic control requires a notion of QoS to be added to the system, which, for NDN, is QoSA-ICN.

NDN Protocols

The NDN implementation used as a baseline is the standard version downloaded from the NDN open source repository[1]. QoSA-ICN is an extension to this implementation. See Section 2.2 for background information on NDN and Section 2.5.3 for a discussion on the design of the QoSA-ICN extensions to NDN.

UDP/IP Network Protocol

Producer/consumer scenarios were designed to execute over UDP/IP with the corresponding applications and network environments configured to match the behaviour of the NDN applications (see Table 5.1). UDP sockets are created at each node by combining an IP address and port from a configured IP address range and port numbers. When the consumer nodes want to send a request, they broadcast to a range of configured IP addresses. When a request is received, only producer nodes send the corresponding data back to the requested consumer directly by using its IP address.

Metrics

The research questions require an answer that indicates the extent, or otherwise, of improvements in the delivery of time-sensitive data. To this end, the percentage of successful packet deliveries achieved within deadlines is measured, as a proportion of the total successful packet deliveries, for all priorities and all scenarios. The measurements are taken in the context of varying proportions of producers in the environment, vehicle speed, network density and experimentation duration. The percentage of network packets delivered within the deadline, as a proportion of total network packets (including packets not delivered), is also measured. It was observed during the experimentation that a low value in this second metric meant there was a corresponding higher rate of re-transmissions of packets, while algorithms continued in their attempts to

---

[1] https://github.com/named-data

find and deliver data packets. Such a situation is therefore an indication of poor efficiency.

5.1.3 Threats to validity

Each experiment was repeated 10 times for consistency. However, at every iteration, the roles (producer or consumer) of vehicles are randomly selected to mitigate any bias resulting from one specific pattern of consumer and producer distribution. It is outside the scope of this paper to consider all possible combination of roles, as this would require a huge number of iterations for each scenario and is quite difficult to achieve in large-scale parameterised experiments involving thousands of vehicles (nodes).

It is assumed that there is a global uniform network understanding of deadline awareness and associated prioritisation within name spaces. This may not reflect behaviour in a real-world implementation.

5.2 Results and analysis

Experiment results across the range of parameters captured in Table 5.1 are shown in Figs 5.4-5.14. In order to assess algorithm efficiency for overall packet delivery, a statistical analysis was performed to verify whether the observed differences in the results for the evaluated algorithms are statistically significant. First, a Shapiro-Wilk normality test was performed to determine whether the results follow a Gaussian distribution. To further validate the results of the Shapiro-Wilk test, a Kurtosis and Skewness test was performed across the samples. As the results follow a Gaussian distribution, an ANOVA parametric analysis was used in the second step. The confidence level was set to 95 % in all statistical tests (i.e., p-value 0.05), which means that the differences are unlikely to have occurred by chance with a probability of 95 %. Tukey HSD was used as a post hoc method on the result. Where the observed differences are significant, the effect size was measured using, η^2 (eta squared) test. Eta squared measures the proportion of the total variance in a dependent variable and the partial eta squared is considered as a measure of effect size [Richardson, 2011]. Table 6 in Appendix B shows the test results for Eta Squares per packet, group (speed, producers and experiment length) across each algorithm and density. The η^2 values are highlighted in three different colours by following the effect size groups (small - 0.02, medium- 0.13 and large 0.26), as rule of thumb given by CohenD. [Cohen, 1977]

The graphs show the overall performance and efficiency of QoSA-ICN compared to the baselines. In all, there are six result sets, captured as bars, showing two wireless connectivity versions for each of QoSA-ICN, NDN and UDP/IP, across the 108 scenarios. From the results, it is evident that QoSA-ICN performs consistently better in successful packet delivery relative to the NDN and UDP/IP algorithms. In the following sections, the impacts of density, speed, proportion of data producers and connectivity are discussed.

5.2.1 Impact of Speed

In general, as shown in Figure 5.2, QoSA-ICN performs better on both WiFi and DSRC connectivity links, when compared with the baselines i.e., NDN and IP on the corresponding connectivity. Interestingly, the performance of QoS-ICN on WiFi decreases at a higher rate on the sparse environment, when compared to medium and dense environments, when speed is increased. This shows that a shorter link (WiFi, 100m range) is vulnerable to speed increases, as disconnections are likely to be frequent, and link durations very short when vehicles are moving fast. On the other hand, with a DSRC link, QoSA-ICN's performance increases at a small percentage when speed increases, as it also does for NDN in medium and sparse environments, and for IP protocols in every density. Overall, an increase in vehicle speed does not adversely affect the delivery of data within a deadline, when the connectivity is long range (DSRC, 1000m range). Based on the QOSA-ICN DSRC scenarios, it can be estimates that it will take 18 secs to break the link once the connection is established, even when vehicles are moving in opposite directions, at a max speed of 100km/h. On the other hand, increasing speed has a positive influence on the UDP/IP-based algorithms, especially in a sparse environment. This may be because UDP/IP is connectionless and the application design is lightweight i.e., it does not go through any additional processing like TCP/IP (3 way handshake SYN, SYN-ACK, ACK, packet seq. check, etc.) or NDN (check cache, PIT, FIB, etc.), before responding to requests. Even though fast-moving traffic produces a more uncertain network topology, with increasing speed a packet reaches more vehicles in a shorter duration. This means that UDP/IP sees an improvement in data retrieval. However, the performance UDP/IP with respect to dense environment is low, this could be due to too many concurrent connections, interference and pocket collisions arising from dense population of vehicles.

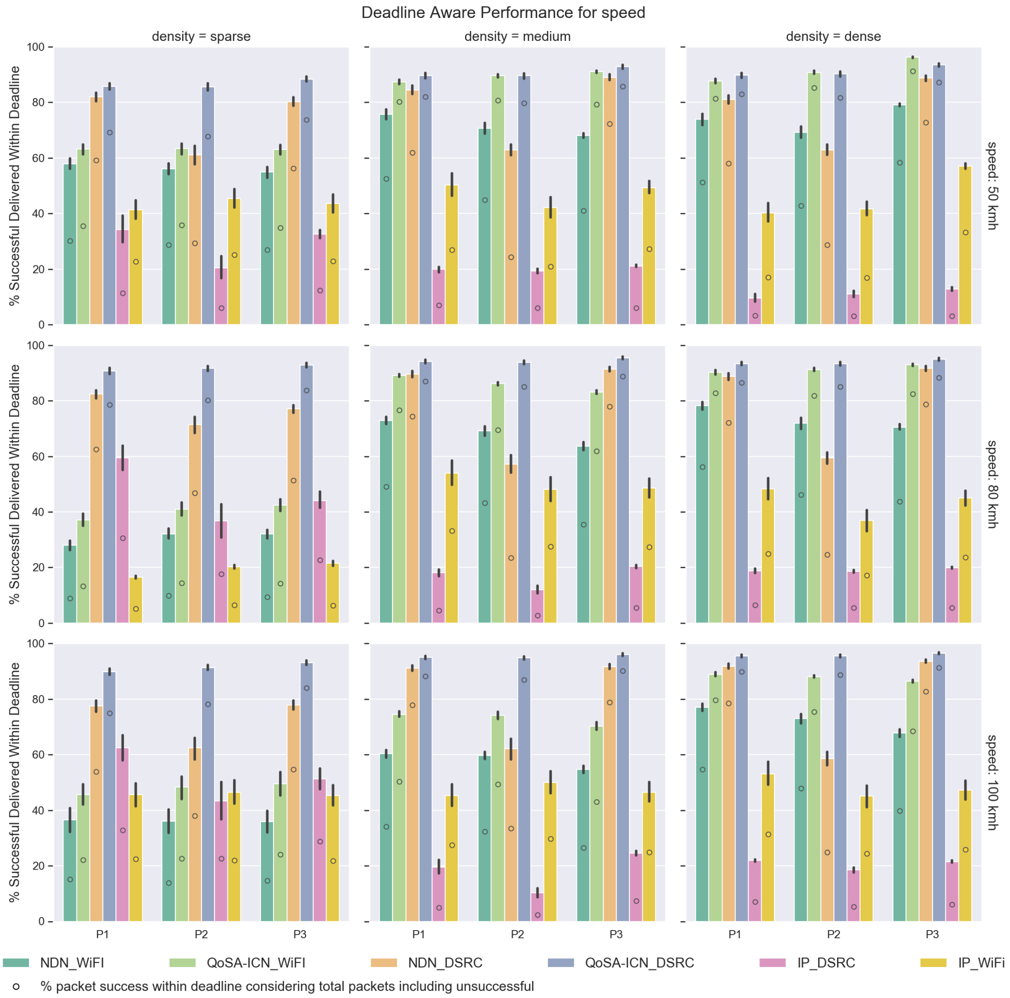


Figure 5.2: QoS Success for Density vs.Speed

5.2.2 Impact of Data Producers

Figure 5.3 illustrates that, in general, increasing vehicle density leads to a higher percentage deadline awareness for all the QoSA-ICN wifi-based algorithms, across all scenarios, independent of increasing the % of data producers in the environment. This is most evident when moving from sparse to medium densities, as the proportion of successful packets delivered within deadline increases by 40% for all traffic types. A similar trend is observed for NDN WiFi, though the increase of percentage success is not as high, at 30%. On the other hand, increasing density has very little impact on the QoSA-ICN DSRC algorithm, given the enhanced connectivity range (1000m on a 2km road segment) and QoSA-ICN's multi-hop communication implementation. It is likely that the content will be found successfully within the network, even if the proportion of producers is low. QoSA-ICN DSRC outperforms NDN DSRC across all densities and producer percentage, which is most evident in sparse environments for both metrics. However, even where the difference is small (e.g., 2-5% in dense environments), QoSA-ICN's successful packet delivery within deadline is consistently higher across all producer scenarios. This implies that QoSA-ICN spends less time on packet re-transmission in comparison to NDN, and is therefore more efficient. It can also be observes that increasing the percentage of data producers positively affects the overall success rate for UDP/IP, indicating that it requires more producers in the environment to successfully deliver packets within the deadline. Neither of the NDN algorithms is as affected by the proportion of producers, indicating that there is not the same dependence on a large number of producers in an ICN environment. Of the two, QoS-ICN shows the better results.

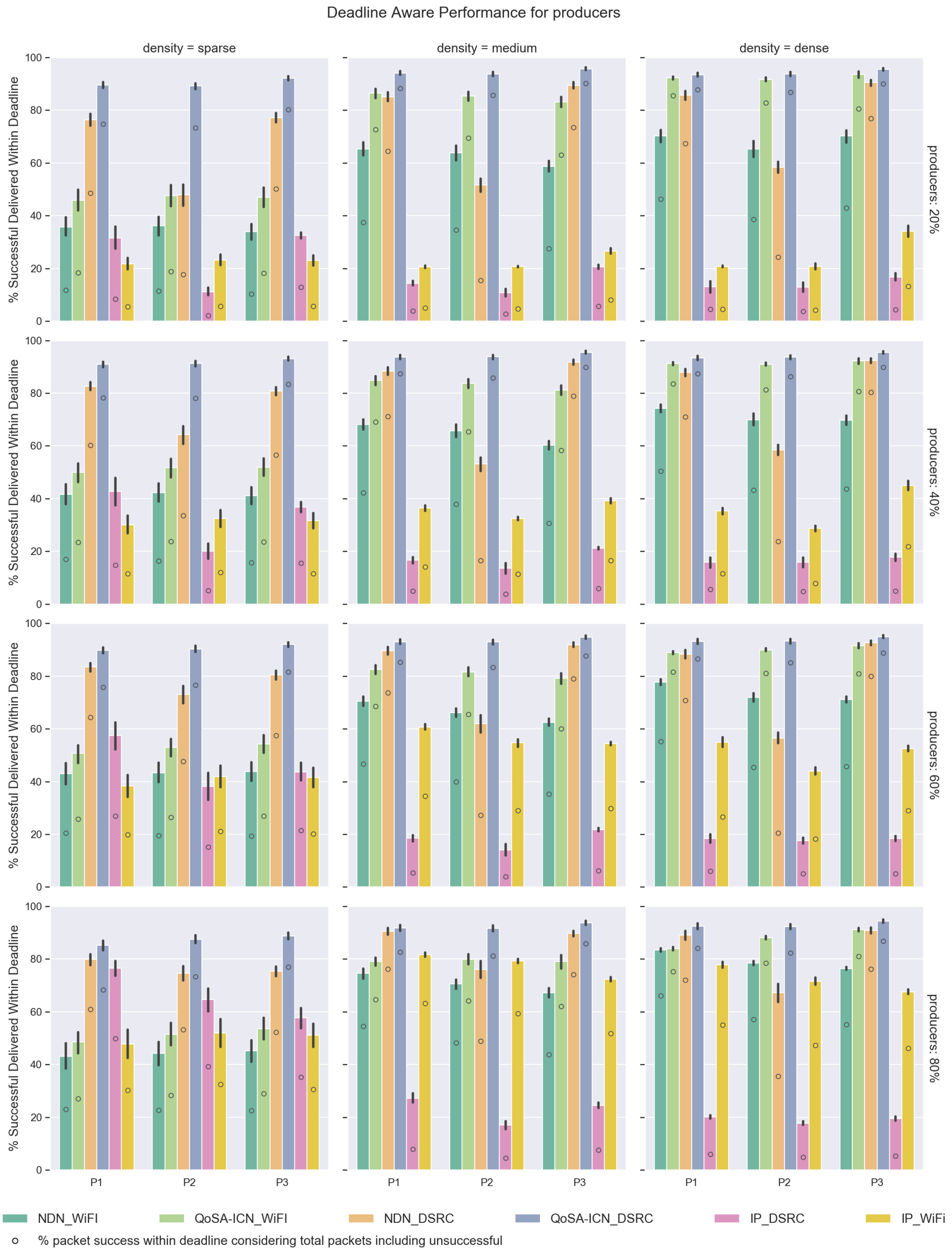


Figure 5.3: QoS Success for Density vs. Producers

5.2.3 Impact of Experiment Duration

ICN's premise is that data are likely to propagate over a network over time, and this will impact retrieval, and hence delivery success. To determine the length of time it would take to influence data delivery within deadline, the experiment durations were set at 5 minutes (300 seconds), 10 minutes (600 seconds) and 20 minutes (1200 seconds). The experiment duration directly correlates to real time. This means that experiment duration is an indicator of how long it takes for packets to propagate across a network. Fig 5.4 illustrates that an increase in experiment duration positively affects the percentage of successful data delivered for QosA-ICN WiFi. This increase occurs across all densities, though is most evident in sparse environments and moving from sparse to medium density. The biggest percentage increase is from 300 seconds to 600 seconds, where there is an average of 20% increase in success rate for all traffic priorities. When the experiment duration is increased from 600 seconds to 1200 seconds, the upward trend in the percentage successful data delivered within deadline continues. However, the percentage increase is much less. Across the experiments, a run of 600 seconds is needed for QoS-ICN WiFi to create optimal data propagation of all packet types across the network. Increasing the duration from 600s to 1200s will indeed increase percentage success but at a lower rate across all densities, with the highest increase occurring in sparse environments. The trend is also evident for NDN Wifi, though QoSA-ICN WiFi out-performs NDN WiFi consistently over all experiment durations for both metrics. The same behaviour is exhibited for QoSA-ICN DSRC compared to NDN DSRC, but percentage success is greater. The results indicate that a minimum experiment duration of 600 seconds is needed, which is most evident based on the gain in performance in a sparse environment using wifi connectivity. This means that in practice within a sparse environment for algorithms using WIFI connectivity it takes 10 minutes for caches of nodes to be populated to a level which allows for deadline awareness performance gain.

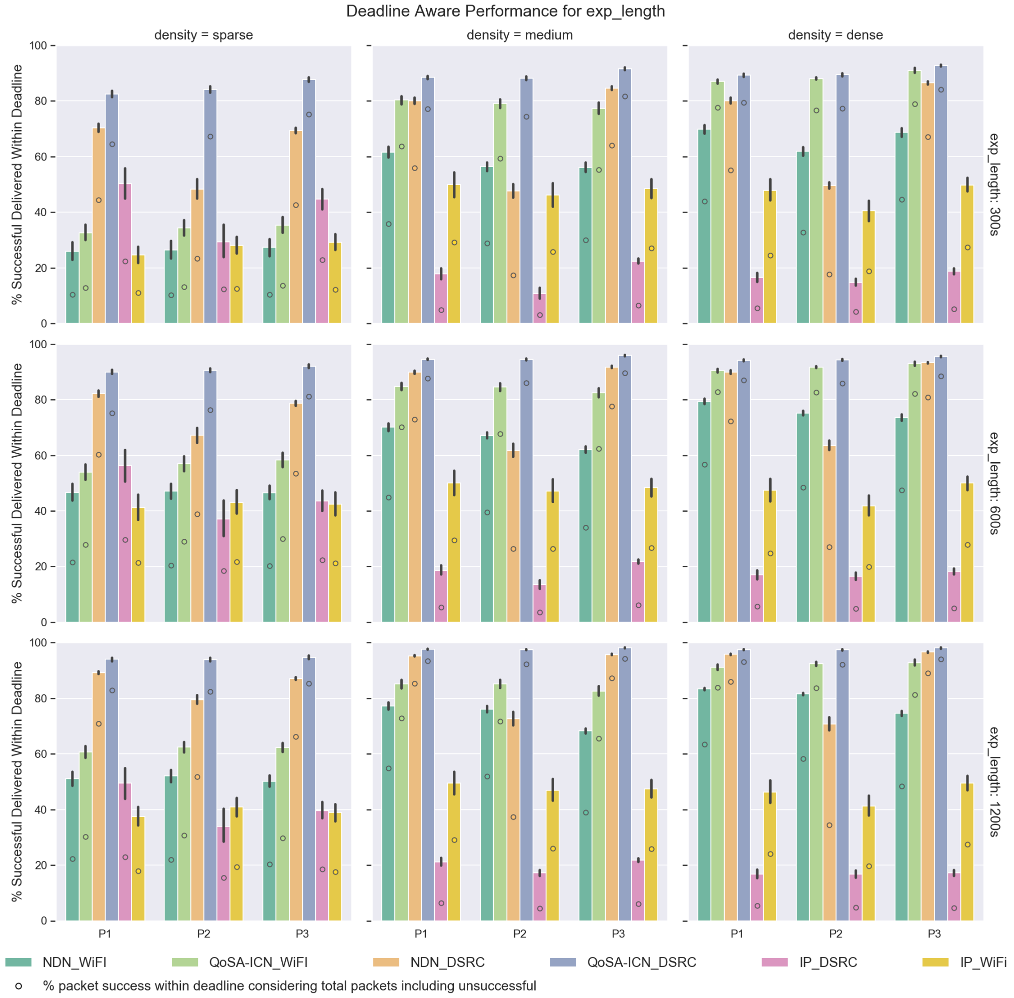


Figure 5.4: QoS Success for Density vs. Experiment Duration

5.2.4 Impact of Connectivity

The results from Figures 5.2, 5.3 and 5.4 validate that connectivity range plays a major role in successful packet return. As seen from the results, QoSA-ICN on a DSRC link has performed the best in all scenarios, which include all ranges of speed, density, percentage of producers and durations. Similarly, with WiFi connectivity, QoSA-ICN performs the best among the other algorithms using WiFi. However, the percentage of gains are significantly higher, especially in medium to dense environments, and speed range between 50 to 80KM/h. This shows that even with short-range links, QoSA-ICN's multi-hop forwarding algorithm performs better in a dense environment, where density and the reduced speed limits are favourable for achieving successful packet delivery. On the other hand, QoSA-ICN with DSRC (1 KM range) helps in environments where there are likely to be fewer vehicles i.e., a sparse environment, with minimal variations in performance for speed ranges 50-100km/h and proportion of producers. The UDP/IP algorithms perform poorly in comparison with both QoSA-ICN and NDN in most scenarios, especially when used with a DSRC link. This is likely to be because of the broadcasting nature of UDP/IP, making it susceptible to interferences, packet collision and also the packet drop caused by queue overflow [udp,].

5.2.5 Algorithm efficiency for packet delivery

Each o symbol on Figures 5.2, 5.3 and 5.4 represent the % of packets delivered within the deadline, as a proportion of the overall number of network packets, including unsuccessful ones. Higher values for this proportion indicate lower re-transmission rates and lower failed packets on the network. In most instances, both versions of the QoS-ICN algorithms achieve a higher rate of packet success within specified deadlines, compared to their counterpart baseline algorithms. The results indicate that these algorithm are not only more successful at delivering requested data within deadline, but are also more efficient.

5.2.6 Mean difference in performance

Figures 5.5 and 5.9 shows the summary mean percentage values for deadline-aware data delivery, for each of the algorithm comparisons under evaluation i.e. QoSA-ICN and baselines with no traffic control Fig. 5.5, QoSA-ICN and QoSA with traffic control Fig. 5.9 Appendix A enabled and all algorithms Fig. 4. The colour on the figures are based on summary mean performance for all evaluation scenario densities, where the colour bar centres at 0, with positive summary means indicated with a red hue. For example in the case of 5.5 It can be seen that the QoSA-ICN DSRC algorithm deliver a higher deadline-aware performance for all traffic priorities, compared to the baseline algorithms. The lowest mean difference values occur when compared to NDN DSRC. The highest mean difference occurs between QoSA-ICN DSRC and the IP-based baselines.

5.2.7 Impact of QoS-aware traffic Control

The QoS-aware ICN algorithms have also been assessed against QoS-aware ICN with traffic control enabled (Figures 5.6, 5.7 and 5.8). The results are less favourable to deadline-awareness success when traffic control is enabled for

QoSA-ICN. QoSA-ICN without traffic control out performs QoSA-TC-ICN for the most part. This is particularly evident for WiFi-based QoSA algorithms in sparse and medium environments and less so in dense environments for speed, data producers and algorithm length where the performance of QosA-ICN DSRC and QoSA TC-ICN DSRC is similar. The heat map illustrated in Figure 5.9 reinforces these findings. In addition, see further graphs in Appendix A, which illustrate all baseline algorithms together, combining all results discussed previously. The traffic control mechanism at layer 2 intercept the packets leaving NDN forwarding stack to re-order the packets in the queues in order to ensure prioritised transmission of packets that has QoS deadline. This is achieved by extracting and processing the packet priority classification i.e. p1, p2, p3 (smaller the subscript value higher the priority), timestamp t0 (the time value when the packet was first sent) and the RRT limit t-rtt. Internal queues are created for each priority types and the packets are queued by reshuffling the packet in the queue based on their time-to-live value i.e., least time to live t-ttl packets always stays at the head of the each internal queue. To choose the packets for transmission from the internal queues, the TTL values of packets that present at the head of the queue are read first. Then, comparisons are made among them such that the packet has the least TTL value and high priority class. If more than one packets showing same t-ttl values then the high priority packet will be selected to transmit further down in the network (NetDevice). It is likely processing overhead has been introduced as by the design as it includes management of a root and three packet queues, one for each of the three priorities.To choose the packets for transmission from the internal queues, the Time to live (TTL) values of packets that present at the head of the queues are read first. Then, comparisons are made amongst them to determine which packet has the lowest TTL value and highest priority class.The interruption of packet flow to perform this comparison and the processing associated with transferring packets from the root queue, ensuring they are in the right queue, and selecting the most appropriate next packet to send from the three queues, is likely to introduce delays rather than short comings in simulation platform. See Figure 3.5 in Chapter 3, for more information.

Based on this outcome, a focused experiment was carried out to compare the bandwidth consumption of the QosA-ICN DSRC and QoSA-TC ICN DSRC algorithms. Results indicate that QoSA-TC ICN does not consume as much bandwidth as QoSA-ICN, while there is a 1.8x difference in consumption for high

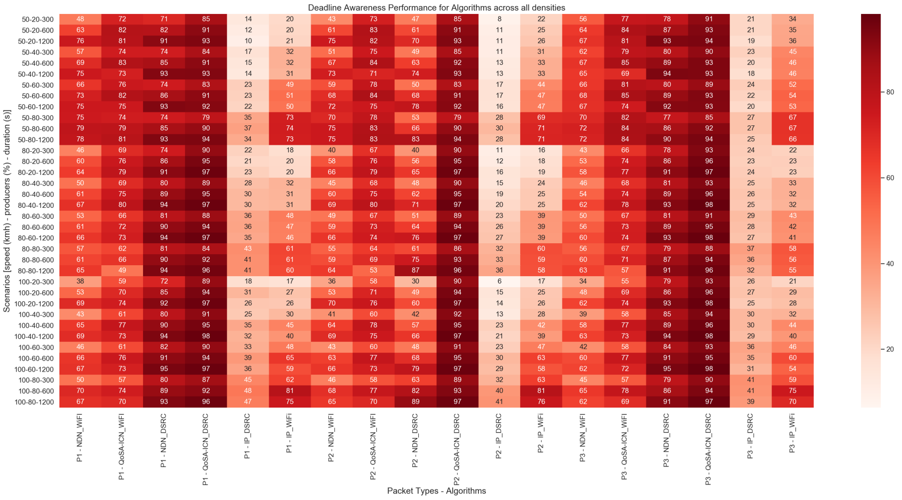


Figure 5.5: Summary Mean deadline-aware performance all packet types and all densities QoSA-ICN vs. baselines

and very high usage (see Figure 5.12). This is encouraging, and makes it worth investigating, in future work, whether the design of the deadline awareness algorithm on the data link layer can be streamlined to improve delivery success.

5.3 Result Summary

Figs 5.10a, 5.10b, and 5.11 illustrate an averaging of the results, where relevant parameters are averaged across all scenarios. The figures show that the QoSA-ICN algorithms perform consistently better in successful packet delivery. In Fig. 5.10a, it can be seen that QoSA-ICN achieves 3.5x and 1.6x rates of improvement over Dedicated Short Range Communication (DSRC) links, and 1.18x and 1.2x rates of improvement on a WiFi link, when compared to UDP-IP and NDN, respectively. In addition, it is evident from Fig. 5.10b and Fig. 5.11 that QoSA-ICN delivers higher volumes of packets with fewer re-transmissions under equivalent scenarios, indicating greater algorithm efficiency for packet delivery. The QoS-aware ICN algorithms have also been assessed against QoS-aware ICN with traffic control enabled. Results have been less favourable for deadline-aware data delivery for the QoSA-ICN algorithms when traffic control is enabled.

The parameters included in these summaries are range, vehicle density, speed, proportion of data producers in the environment, and connectivity links.

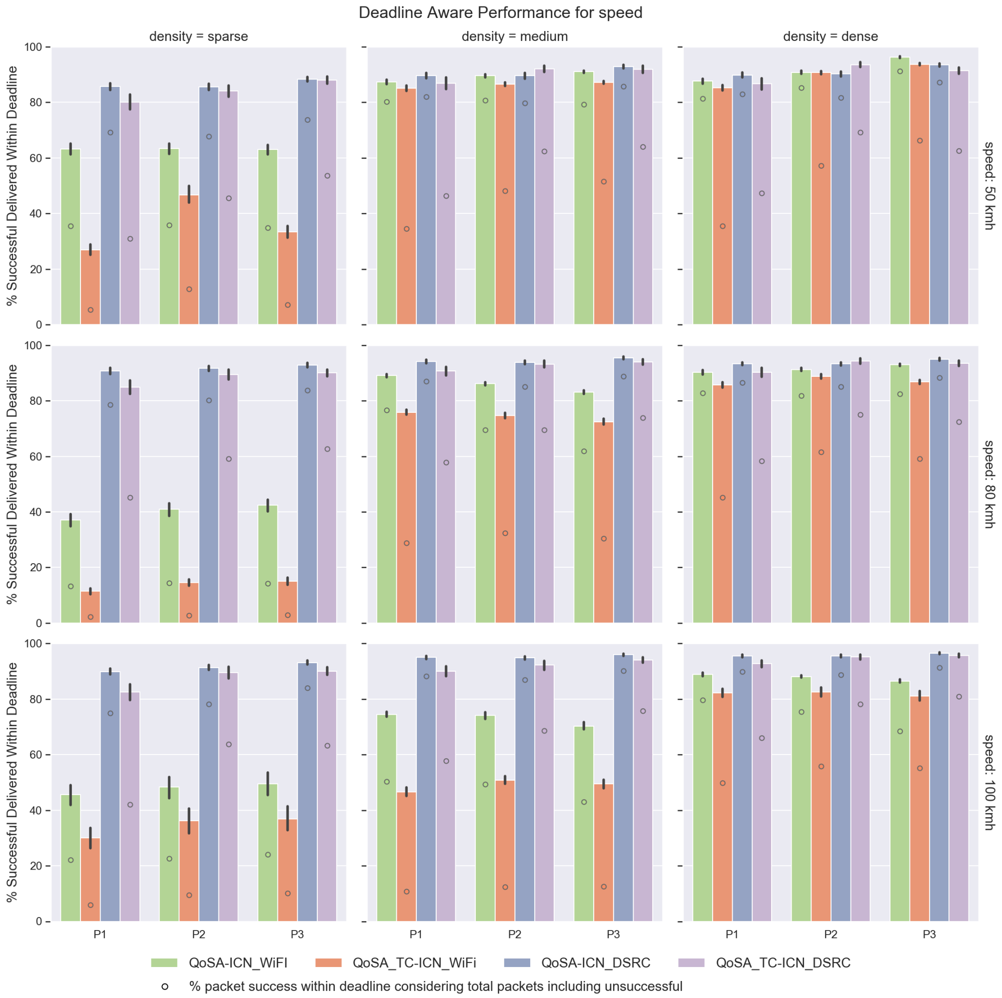


Figure 5.6: QoS Success for Density vs. Speed with traffic control

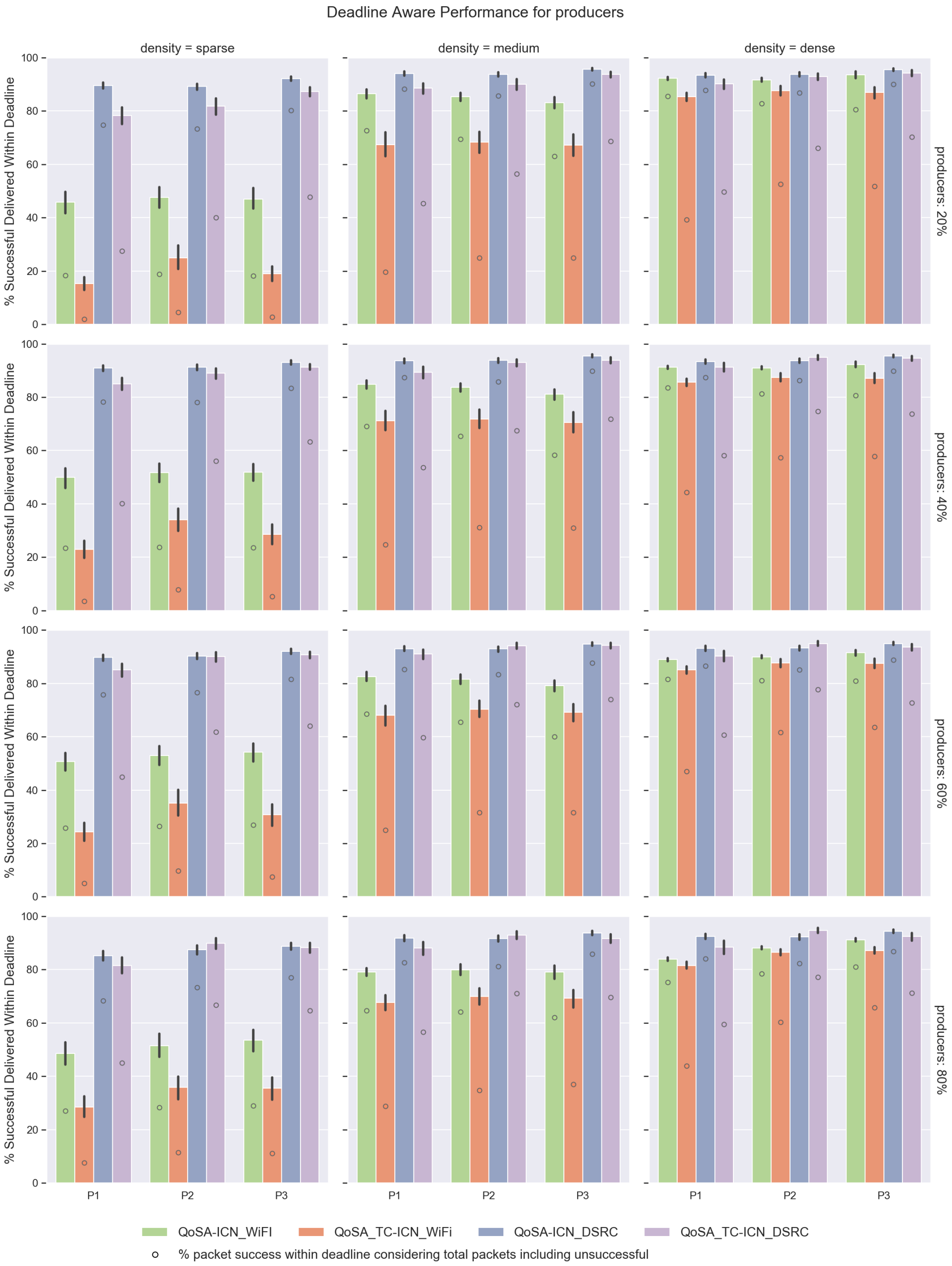


Figure 5.7: QoS Success for Density vs. % Producers, with traffic control

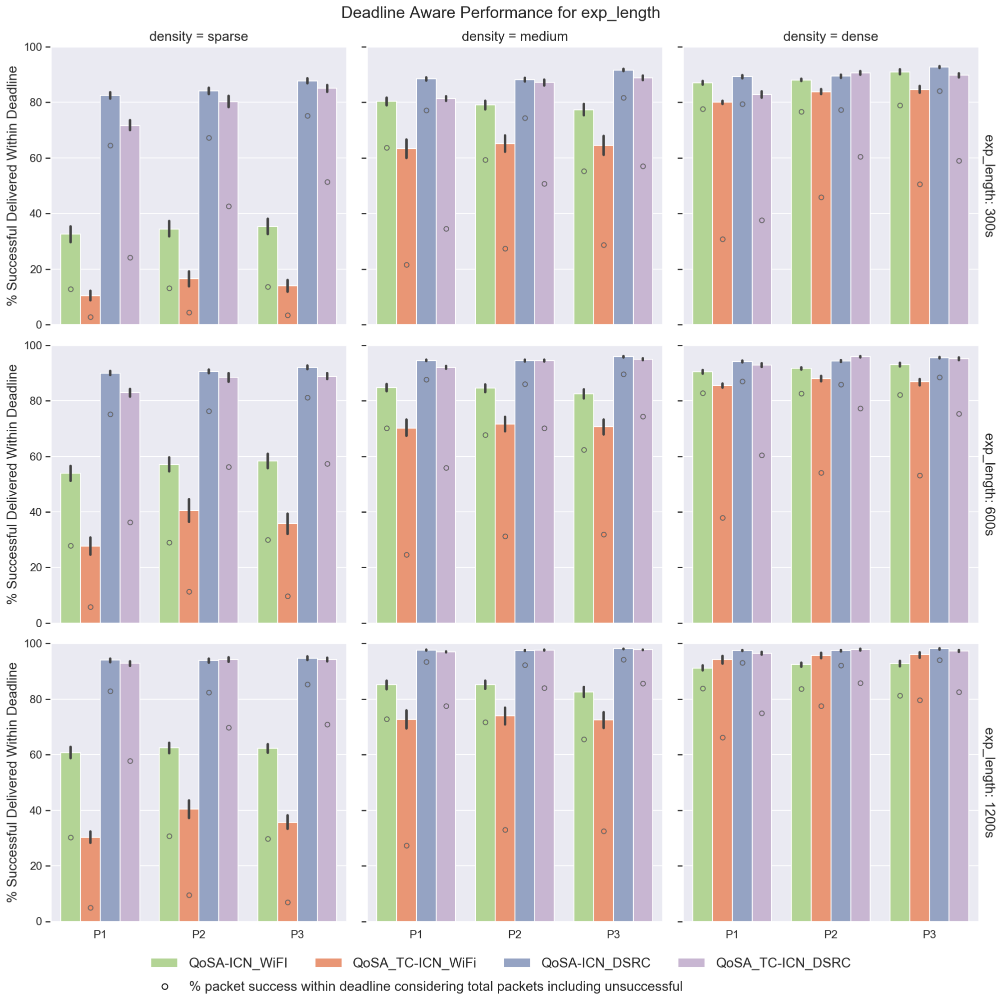
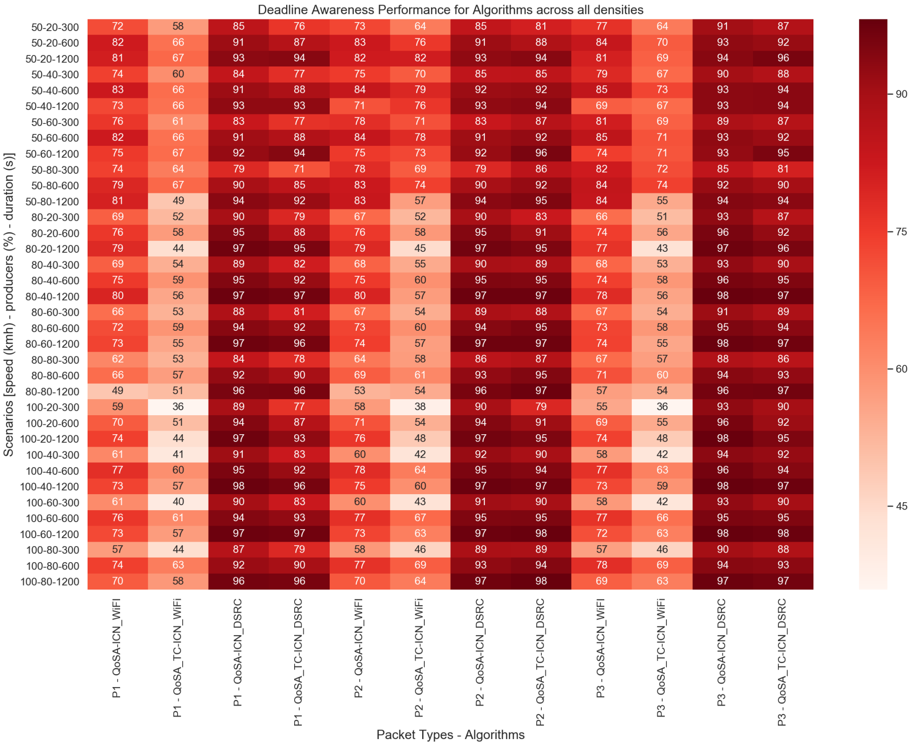


Figure 5.8: QoS Success for Density vs. Experiment Length with traffic control



Scenarios [speed (kmh) - producers (%) - duration (s)]	P1 - QoSA-ICN_WiFI	P1 - QoSA_TC-ICN_WiFi	P1 - QoSA-ICN_DSRC	P1 - QoSA_TC-ICN_DSRC	P2 - QoSA-ICN_WiFI	P2 - QoSA_TC-ICN_WiFi	P2 - QoSA-ICN_DSRC	P2 - QoSA_TC-ICN_DSRC	P3 - QoSA-ICN_WiFI	P3 - QoSA_TC-ICN_WiFi	P3 - QoSA-ICN_DSRC	P3 - QoSA_TC-ICN_DSRC
50-20-300	72	58	85	76	73	64	85	81	77	64	91	87
50-20-600	82	66	91	87	83	76	91	88	84	70	93	92
50-20-1200	81	67	93	94	82	82	93	94	81	69	94	96
50-40-300	74	60	84	77	75	70	85	85	79	67	90	88
50-40-600	83	66	91	88	84	79	92	92	85	73	93	94
50-40-1200	73	66	93	93	71	76	93	94	69	67	93	94
50-60-300	76	61	83	77	78	71	83	87	81	69	89	87
50-60-600	82	66	91	88	84	78	91	92	85	71	93	92
50-60-1200	75	67	92	94	75	73	92	96	74	71	93	95
50-80-300	74	64	79	71	78	69	79	86	82	72	85	81
50-80-600	79	67	90	85	83	74	90	92	84	74	92	90
50-80-1200	81	49	94	92	83	57	94	95	84	55	94	94
80-20-300	69	52	90	79	67	52	90	83	66	51	93	87
80-20-600	76	58	95	88	76	58	95	91	74	56	96	92
80-20-1200	79	44	97	95	79	45	97	95	77	43	97	96
80-40-300	69	54	89	82	68	55	90	89	68	53	93	90
80-40-600	75	59	95	92	75	60	95	95	74	58	96	95
80-40-1200	80	56	97	97	80	57	97	97	78	56	98	97
80-60-300	66	53	88	81	67	54	89	88	67	54	91	89
80-60-600	72	59	94	92	73	60	94	95	73	58	95	94
80-60-1200	73	55	97	96	74	57	97	97	74	55	98	97
80-80-300	62	53	84	78	64	58	86	87	67	57	88	86
80-80-600	66	57	92	90	69	61	93	95	71	60	94	93
80-80-1200	49	51	96	96	53	54	96	97	57	54	96	97
100-20-300	59	36	89	77	58	38	90	79	55	36	93	90
100-20-600	70	51	94	87	71	54	94	91	69	55	96	92
100-20-1200	74	44	97	93	76	48	97	95	74	48	98	95
100-40-300	61	41	91	83	60	42	92	90	58	42	94	92
100-40-600	77	60	95	92	78	64	95	94	77	63	96	94
100-40-1200	73	57	98	96	75	60	97	97	73	59	98	97
100-60-300	61	40	90	83	60	43	91	90	58	42	93	90
100-60-600	76	61	94	93	77	67	95	95	77	66	95	95
100-60-1200	73	57	97	97	73	63	97	98	72	63	98	98
100-80-300	57	44	87	79	58	46	89	89	57	46	90	88
100-80-600	74	63	92	90	77	69	93	94	78	69	94	93
100-80-1200	70	58	96	96	70	64	97	98	69	63	97	97

Figure 5.9: Summary Mean deadline aware performance QoSA-ICN vs. QoSA TC-ICN all packet types and all densities

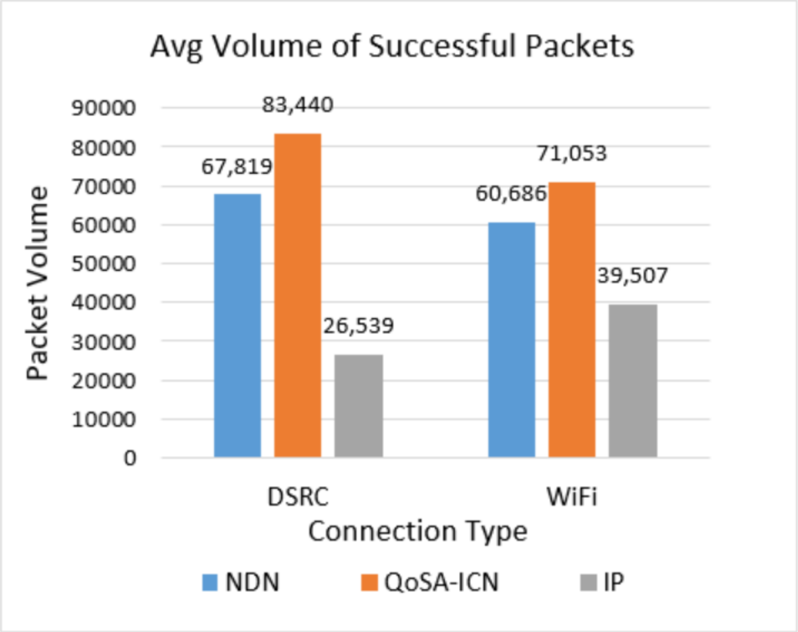
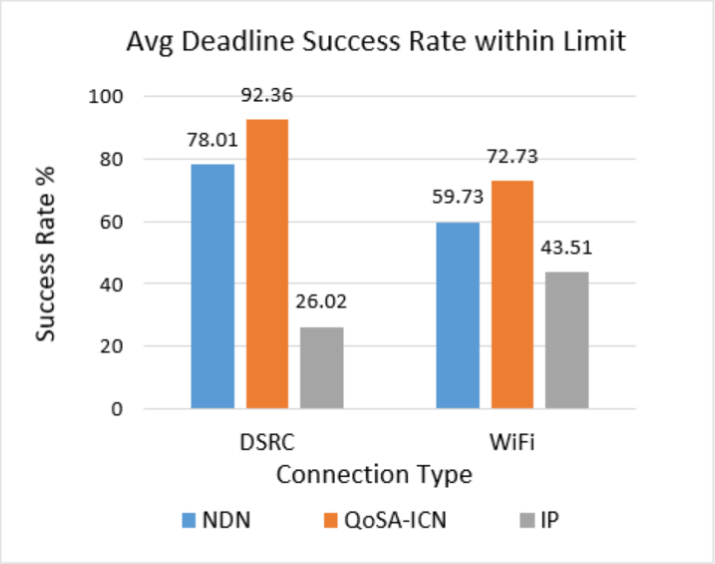
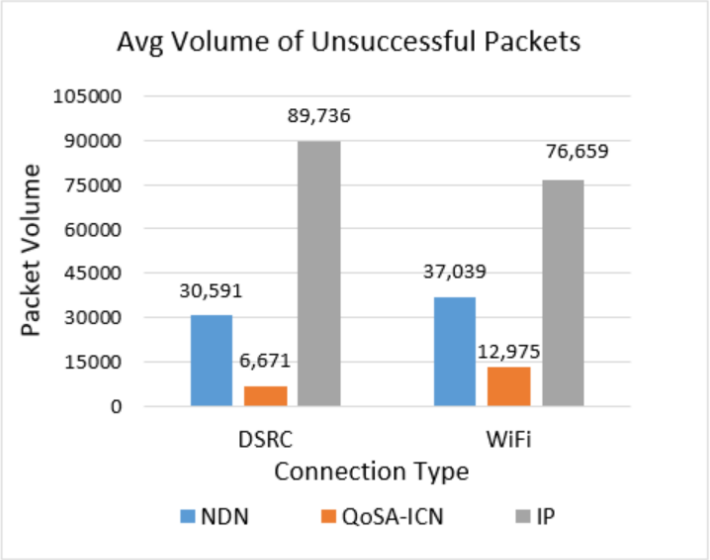


Figure 5.11: Overall Average Volume of Successful Packets.



(a) Overall QoS adherence Rate Comparison.



(b) Overall Average Volume of Unsuccessful Packet Return.

Figure 5.10: (a) Overall QoS adherence Rate Comparison (b) Overall Average Volume of Unsuccessful Packet Return

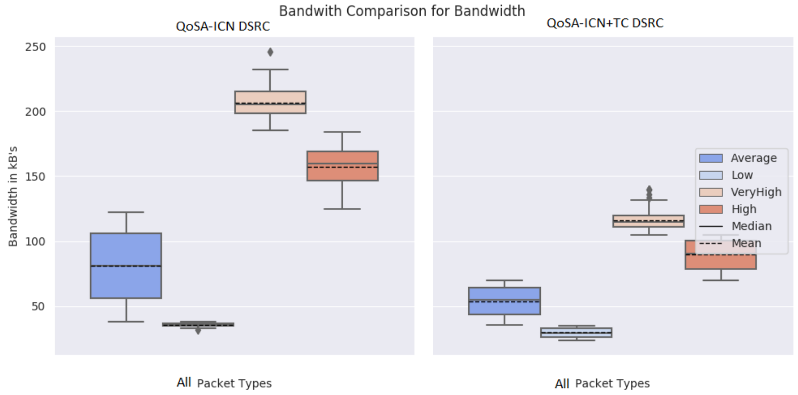


Figure 5.12: Comparsion of Bandwidth consumption for QoSA-ICN vs QoSA-TC-ICN.

5.4 Chapter summary

Evaluations have been carried out using extensive simulation, in particular using a combination of the ndnSIM and SUMO simulators. Deadline-aware success rate and packet delivery success rate are both measured under different network densities, vehicle speeds, proportions of vehicles in the environment acting as content producers, and experiment duration's. The QoS-aware ICN algorithms are assessed against four baselines: UDP IP (with both DSRC and wifi communication channels), and basic NDN (DSRC and WiFi).

The results of this study demonstrate that the QoS-aware approach generally achieves higher success rates at delivering different packet priority types within their deadlines, relative to the baselines. In addition, QoSA-ICN requires fewer re-transmissions to achieve successful delivery within deadlines, which leads to greater efficiency within algorithms itself and the network overall, as resource are consumed on value-add tasks i.e., successful packet delivery vs. packet re-transmission.

The QoS-aware ICN algorithms have also been assessed against QoS-aware ICN with traffic control enabled. Results are less favourable for deadline-aware data delivery for the QoSA-ICN algorithms when traffic control is enabled. QoSA-ICN without traffic control out performs QoSA-ICN+TC in the majority of scenarios, see Figures 4. Results also indicate that Quality of Service Aware Information Centric Networking with Traffic Control (QoSA TC-ICN) does not consume as much bandwidth as QoSA-ICN in terms of average consumption, while there is a 1.8x difference in consumption for high and very high usage (see Figure 5.12). These results are applicable to dynamic cyber physical systems where quality of service awareness is a requirement.

Chapter 6

Conclusion

This chapter summarises the thesis and its achievements. It discusses the trade-offs of the QoSA-ICN design, and highlights potential areas for future work.

6.1 Thesis Summary

This thesis presents a QoS-aware Information Centric Network for vehicular applications. In particular, the work extends the Named-Data Network (NDN), a variant of ICN, focusing on data delivery deadline awareness. The new algorithms classify the priority of requests, with associated QoS requirements, by encoding QoS information into the interest request packets and corresponding data reply packets. The NDN routing algorithm was extended to use multi-hop forwarding to efficiently request and receive the requested content, without pre-building routes

Evaluations have been carried out using extensive simulation, in particular using a combination of the ndnSIM and SUMO simulators. Deadline-aware success rate and packet delivery success rate are both measured under different network densities, vehicle speeds, proportions of vehicles in the environment acting as content producers, and experiment durations. The QoS-aware ICN algorithms are assessed against four baselines: UDP IP (with both DSRC and wifi communication channels), and basic NDN (DSRC and WiFi).

The results of this study demonstrate that the QoS-aware approach generally achieves higher success rates at delivering different packet priority types within their deadlines, relative to the baselines. In addition, QoSA-ICN requires fewer re-transmissions to achieve successful delivery within deadlines, which causes less congestion on the network.

Introduction Chapter 1 describes the motivation of this thesis which arose from the new challenges which exist for deadline sensitive cyber physical systems which sit in a dynamic edge environment. It also described the overall research objectives and research questions of this thesis.

State of the Art Chapter 2 reviewed the information centric, host based and ad-hoc networks in terms of approaches to content delivery. It examined different approaches to QoS provisioning in these network types. The analysis highlighted the gap for end to end QoS provisioning within information-centric networks. ICN offerings in VANETs have not, to date, considered an end-to-end QoS mechanism for both content discovery and content delivery, and it remains a challenging task to maintain QoS for applications in vehicular ad hoc networks (VANETs). The chapter also highlights that a cross-layer approach is needed for QoS provisioning, as QoS is not a layer-specific issue, and spans all layers in the communication protocol stack.

Design Chapter 3 focused on the design of QoSA-ICN. Chapter 3 returns to the challenges of data delivery in a dynamic edge network, as outlined in Chapter 1 and describes the design objectives and system model of this thesis. Thereafter, the chapter explains in detail how the proposed algorithms for deadline-aware data delivery address the research objectives.

Implementation Chapter 4 describes the overall artefacts that were implemented to realise QoSA-ICN and fulfil the design objectives in chapter 3 of this thesis. The implementation delivers: a) A method to allow for QoS-aware packet extension to both interests and data packets; b) A method to allow for Qo-aware best route forwarding within the Network Forward Daemon (NFD); and c) A method to allow for a cross layer approach to achieving optimised traffic control in the data link MAC layer. It also highlights a comprehensive data pipeline, which was implemented to aid with statistical analysis.

Evaluation Chapter 5 evaluated how well QoSA-ICN achieves the overall objective of this thesis, i.e., of improving data delivery in dynamic environments by employing an information-centric, QoS-aware data delivery mechanism, compared to existing baselines. It describes simulations of various scenarios, designed to expose the algorithms to varying parameters such as network densities, vehicle speeds, proportions of vehicles in the environment acting as content producers, and experiment durations. The evaluation metrics of deadline-aware success rate and packet delivery success rate are both measured against UDP IP (with both DSRC and wifi communication channels), and basic NDN (DSRC and WiFi).

6.2 Discussion

Returning to the motivating scenario of this thesis which looks to understand if by implementing firm and soft real time deadlines a higher percentage rate of QoS adherence be achieved for a dynamic edge based network (VANET) compared to existing baselines. As discussed extensively in Chapter 5 of this thesis, it is evident that QoSA-ICN performs consistently better in successful packet delivery within deadlines, relative to the NDN and UDP/IP algorithms (RQ1). Looking at the impact of speed, while delivery success is generally lower at high speeds in a sparse density for WiFi connections, QoSA-ICN nonetheless performs better than NDN, and slightly better than IP. In medium to dense environments, all algorithms perform better than in sparse, though again, QoSA-ICN shows better results when compared against corresponding connection types. Similar patterns are evident when taking experiment duration into account, where in general shorter experiment length means a lower proportion of successful delivery for all algorithms (except the DSRC ones), and QoSA-ICN performs better. Similar patterns are again evident when taking the proportion of producers in the environment into account. In general, the proportion of producers is not hugely influential, and QoSA-ICN performs better when compared against corresponding network types.

However, the delivery of time-sensitive data in dynamic environments has not been improved by adding QoS-awareness to the network traffic control mechanism (RQ2). The introduction of traffic control causes extra overhead when packets are processed because dedicated priority queues (3 in total) are introduced, and packet flow is interrupted to accommodate theses queues and perform comparisons. First, the new queues are fed from a root queue. The decision-making as to which packets to choose for transmission from the internal queues requires first reading the Time to live (TTL) values of packets at the head of the queues. Then, comparisons are made to determine which packet has the lowest TTL value and highest priority class. If more than one packet has the same ttl values, then the high priority packet will be selected to transmit further down in the network (NetDevice), see Figure 3.5 in Chapter 3. This is evident in short link communication scenarios, i.e., WiFi, in both sparse and medium environments, where the performance for QoSA-ICN without traffic control is superior. This trend is not as evident in longer range communication i.e., DSRC, where the difference between QoSA-ICN and QoSA-TC-ICN is smaller. Deadline-aware success rates are high for both, however QoSA-ICN outperforms QoSA-TC-ICN for both of our evaluation metrics (see Figures 5.6, 5.7 and 5.8). This poor performance might be improved by introducing parallel processing of queues and would make for an interesting piece of future work.

The QoSA-TC-ICN algorithms has shown some promise for network bandwidth consumption, and in a scenario where bandwidth is the primary QoS consideration e.g., resource constrained environments, these algorithms may show promise (see Figure 5.12).

6.3 Future Work

This thesis examined a new research direction for an information-centric approach to deadline awareness. Notwithstanding its contribution to knowledge, the thesis may serve as a starting point for further investigations in areas such as:

6.3.1 Consideration of other QoS properties

Although QoSA-ICN provides a general model and design for QoS requirements relating to information-centric networking, only deadline awareness has been implemented and evaluated for this thesis. It would be interesting to assess the manner in which other QoS properties (e.g., number of hops, bandwidth sharing) could be handled, and their implementation optimised.

6.3.2 Vehicle to everything (V2X)

The topology and composition of vehicular networks change much faster than the traditional cellular networks. The fast-changing topology, the short-lived intermittent connectivity, the wide range of possible applications with heterogeneous requirements, and the harsh propagation conditions, heavily challenge the traditional information-centric network deployment for vehicular networks. When an edge infrastructure is available, vehicles could leverage it to enhance the reliability of data transfer. QoSA-ICN takes a V2V approach, but it would be interesting to investigate whether the addition of fixed point edge-based infrastructure enhances data delivery rates beyond the current results.

6.3.3 Integration with host-based networks

QoSA-ICN takes a clean slate ICN design approach which means it is a novel approach to achieving QoS service assurance within ICN and is not limited by backward capability or integration requirements for existing IP based networking solutions [Intel and NSF, 2018] and, while evaluations are performed against host-based networks (UDP/IP), it does not integrate with host-based networks. It may be interesting to combine the two to understand the benefits and trade-offs of such an approach.

6.3.4 Integration with learning techniques

The experimentation of the QoSA-ICN algorithm has allowed for the capture of a large data set capturing features including round trip time delivery , per hop delivery, geo location information, link level qualities, network resource utilisation etc. Currently, QoSA-ICN does not check for an optimised path for data packet delivery. It would be very interesting to combine this work with learning techniques for future information-centric data sharing optimisations e.g., if an interest packet is generated for geo-location associated content (such as a request for an intersection HD map or specific parking information), learning techniques could look to data producer location, consumer location, road topology and neighbouring nodes location/direction/link qualities to determine the next hop for optimised data delivery. This function could be trained offline by large simulations. The function could also be trained in an online manner to capture the inherent characteristics of the network in which it is implemented, with collaborative learning agents.

6.3.5 Network and Service Providers

QoSA-ICN builds on a number of underlying assumptions a) As QoSA-ICN is a clean-slate ICN, it is assumed that network and service providers are equipped to support information centric networks; b) network service providers adopt a uniform policy to information centric naming and packet prioritisation; c) cross service sharing of data occurs; and d) all network resources are dedicated. It would be interesting to remove these assumptions, and evaluate the efficacy of QoSA-ICN accordingly.

6.3.6 Privacy and Security

The work in this thesis relies on data services that are likely to be offered by third-party providers. With regard to privacy and security concerns, a trustworthy service provider and trustworthy producer of data i.e., a vehicle, is assumed in this research. In terms of security QoSA-ICN has adopted NDN privacy and security mechanisms. An interesting piece of future work would be to investigate all privacy and security considerations related to the QoSA-ICN model. For example, would there be any privacy issue with adding priority information to the packet headers? There are likely other important questions not considered in this thesis.

Appendices

Appendix A

The following figures show results against the two evaluation metrics for QoSA-ICN algorithms and all baselines broken out by density and packets type (p1-p3) considering speed, proportion of producers and experimentation duration. These graphs collate graphs 5.2 5.3 5.4 5.6 5.7 5.8 but also include IP based baselines which do not perform well when compared to QOSA-ICN algorithms.

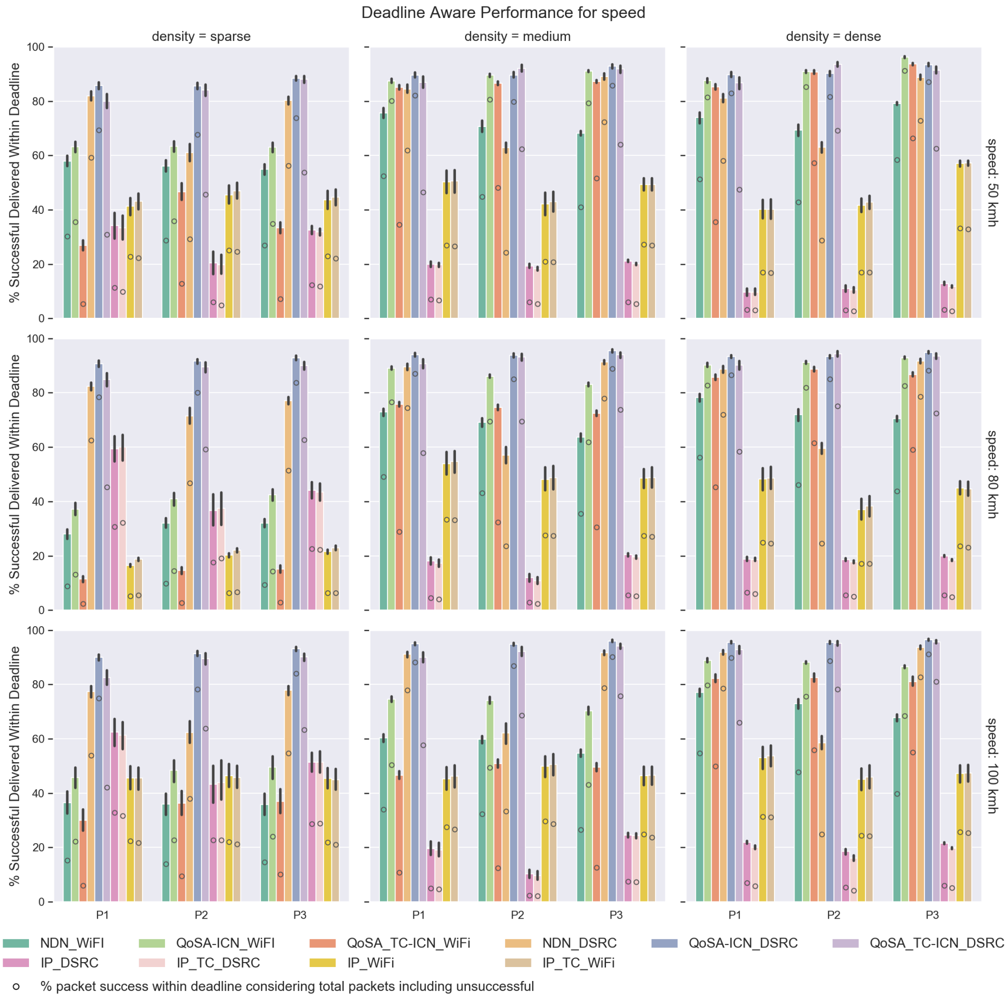


Figure 1: QoS Success for Density vs Speed all algorithms

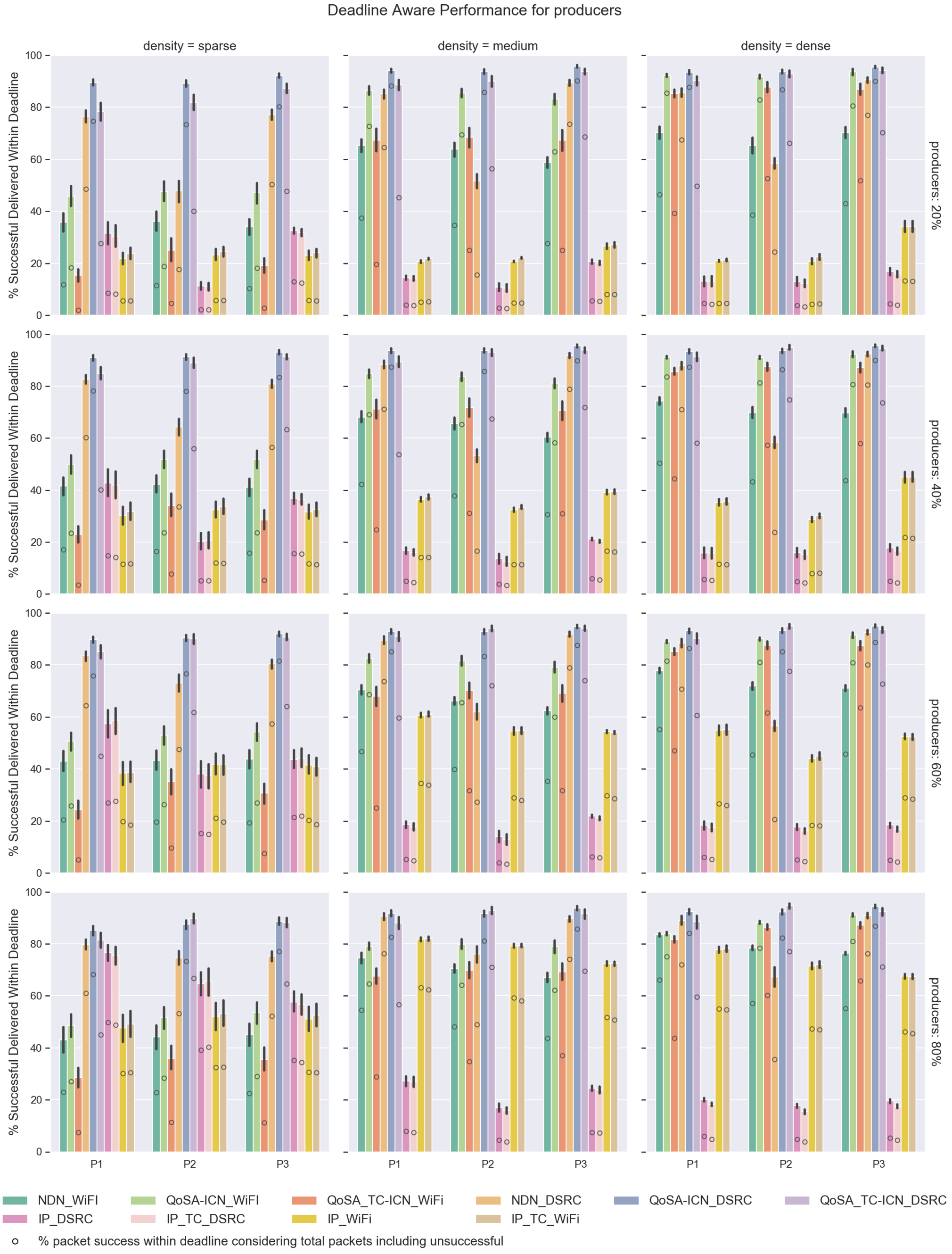


Figure 2: QoS Success for Density vs Producers all algorithms

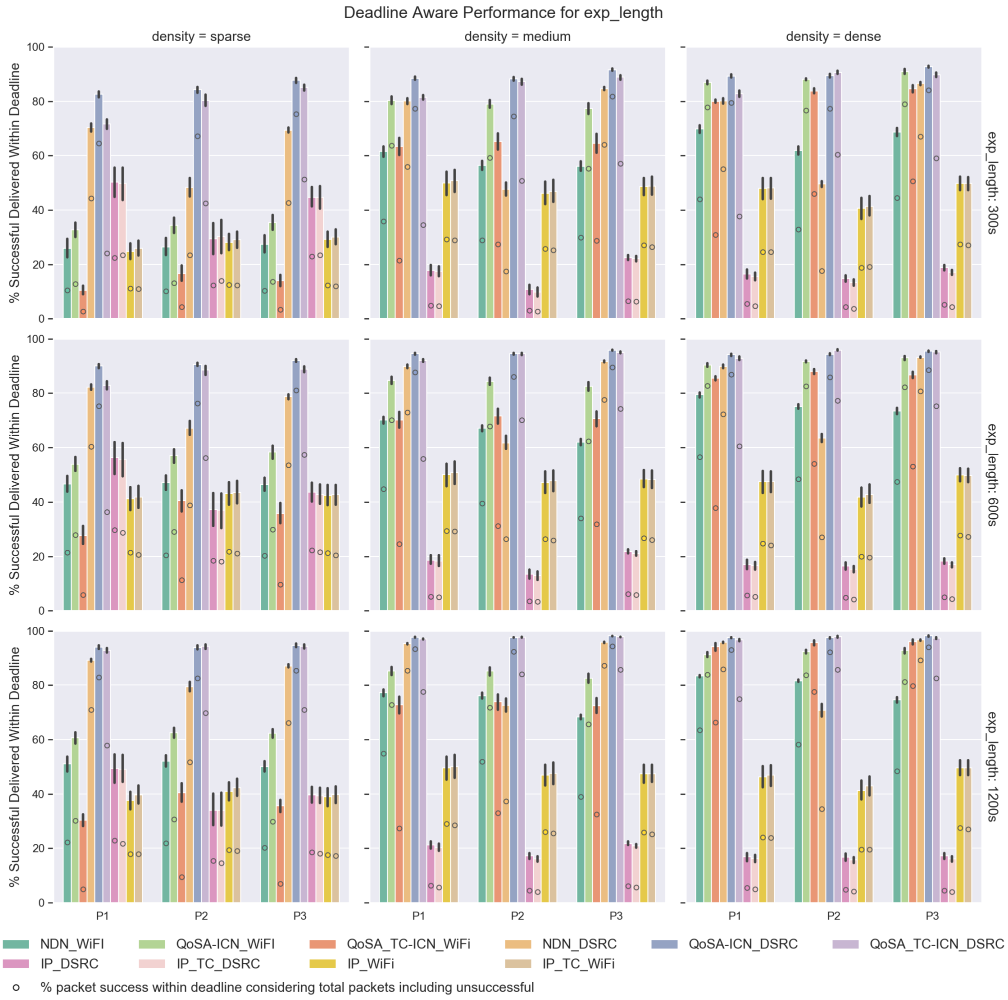


Figure 3: QoS Success for Density vs Exp Duration all algorithms

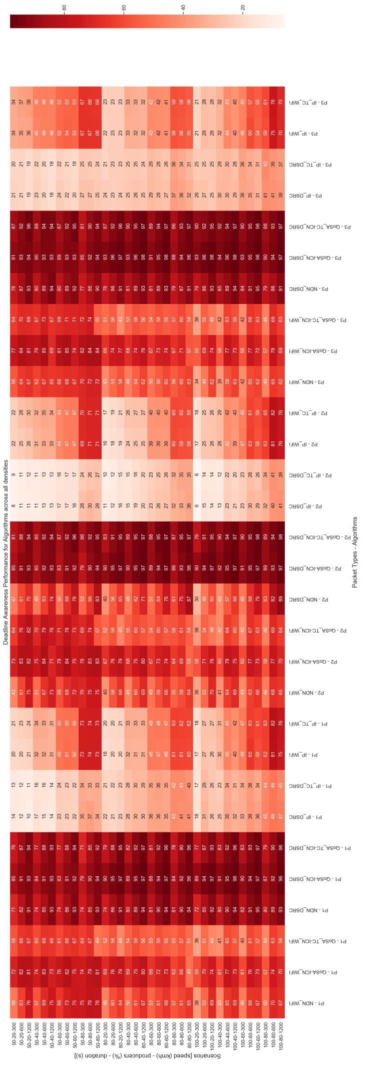
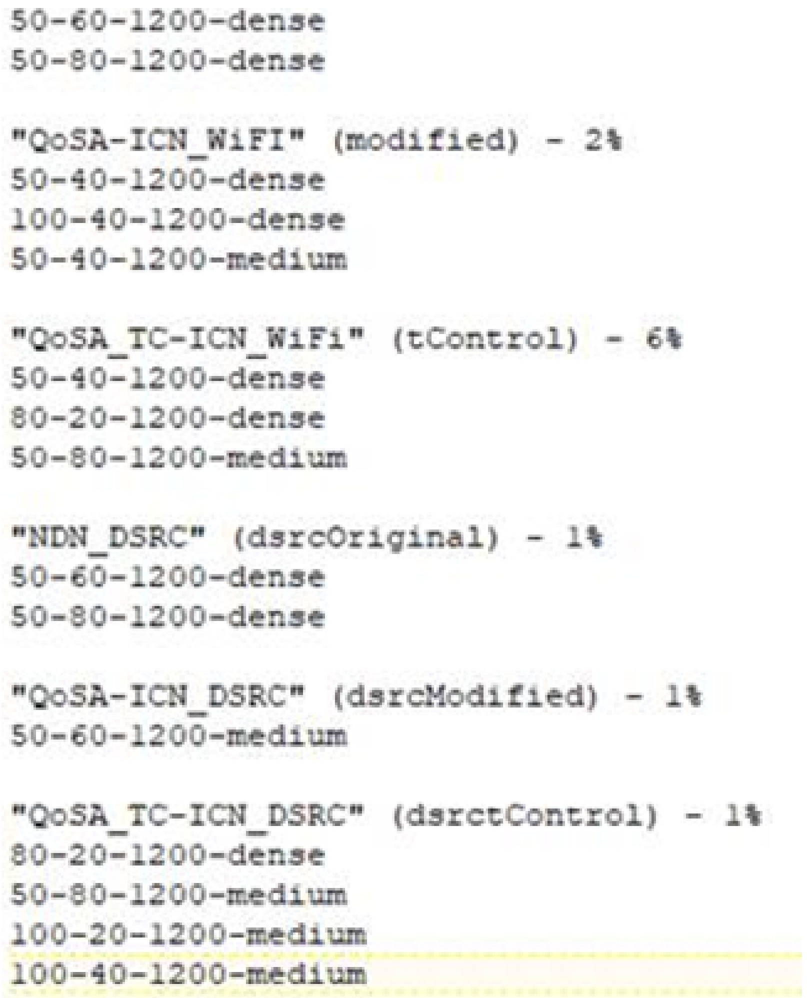


Figure 4: Mean values for all experiments



```
50-60-1200-dense
50-80-1200-dense

"QoSA-ICN_WiFI" (modified) - 2%
50-40-1200-dense
100-40-1200-dense
50-40-1200-medium

"QoSA_TC-ICN_WiFi" (tControl) - 6%
50-40-1200-dense
80-20-1200-dense
50-80-1200-medium

"NDN_DSRC" (dsrcOriginal) - 1%
50-60-1200-dense
50-80-1200-dense

"QoSA-ICN_DSRC" (dsrcModified) - 1%
50-60-1200-medium

"QoSA_TC-ICN_DSRC" (dsrctControl) - 1%
80-20-1200-dense
50-80-1200-medium
100-20-1200-medium
100-40-1200-medium
```

Figure 5: Prediction rates for missing experiments

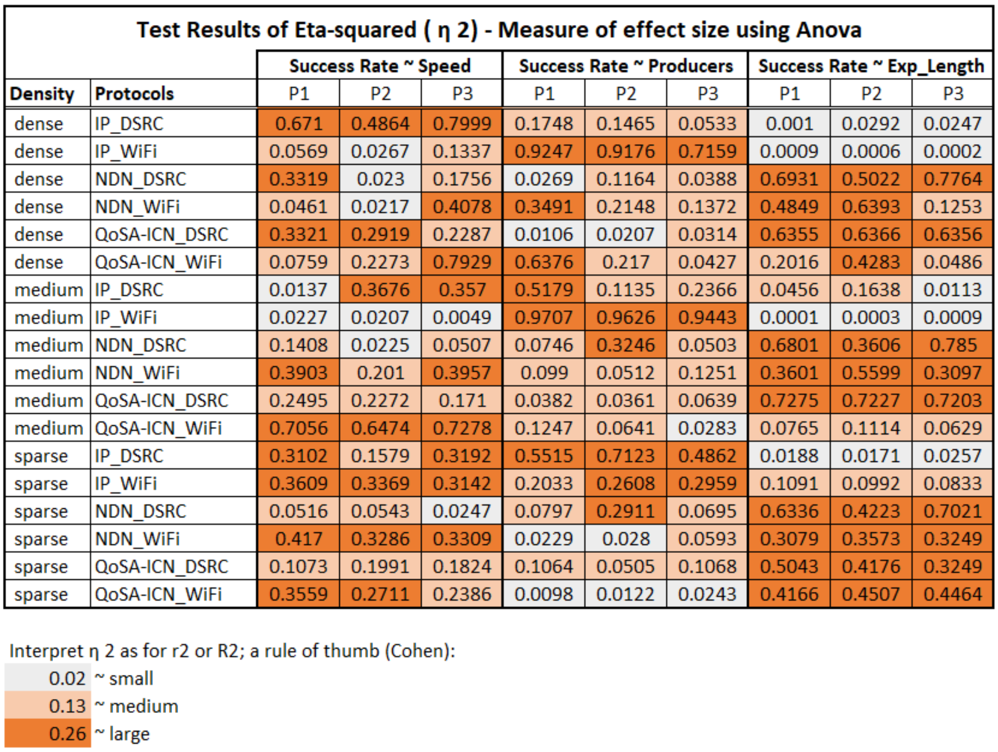
Appendix B

Evaluation Results' Validity

In order to statistically confirm and quantify the impacts of changing parameters such as speed, producer % and experiments lengths on deadline aware success rate as shown in Figures 5.2 5.3 5.4, eta squared tests were conducted to measure the effect size. The results can be seen in Table 6.

In considering the impact of these effects on the overall results, note, for example, how speed impacts the various algorithms in different ways. So, in sparse environments for P1 packets, NDN.WiFi is more affected than QoSA-ICN.WiFi, as indicated by the 0.417 value versus 0.35 value, respectively, Looking at the speed graph (see Fig 5.2), this higher impact for NDN-WiFi can be seen. Across all the algorithms, each is affected differently by different parameters.

Figure 6: Results of η^2 - Measure of effect size using Anova: Statistical testing of effect size on packet success rate at varying speeds, producers and experiment lengths are shown in the table.



Test Results of Eta-squared (η 2) - Measure of effect size using Anova

		Success Rate ~ Speed			**Success Rate ~ Producers**			**Success Rate ~ Exp_Length**		
Density	**Protocols**	P1	P2	P3	P1	P2	P3	P1	P2	P3
dense	IP_DSRC	0.671	0.4864	0.7999	0.1748	0.1465	0.0533	0.001	0.0292	0.0247
dense	IP_WiFi	0.0569	0.0267	0.1337	0.9247	0.9176	0.7159	0.0009	0.0006	0.0002
dense	NDN_DSRC	0.3319	0.023	0.1756	0.0269	0.1164	0.0388	0.6931	0.5022	0.7764
dense	NDN_WiFi	0.0461	0.0217	0.4078	0.3491	0.2148	0.1372	0.4849	0.6393	0.1253
dense	QoSA-ICN_DSRC	0.3321	0.2919	0.2287	0.0106	0.0207	0.0314	0.6355	0.6366	0.6356
dense	QoSA-ICN_WiFi	0.0759	0.2273	0.7929	0.6376	0.217	0.0427	0.2016	0.4283	0.0486
medium	IP_DSRC	0.0137	0.3676	0.357	0.5179	0.1135	0.2366	0.0456	0.1638	0.0113
medium	IP_WiFi	0.0227	0.0207	0.0049	0.9707	0.9626	0.9443	0.0001	0.0003	0.0009
medium	NDN_DSRC	0.1408	0.0225	0.0507	0.0746	0.3246	0.0503	0.6801	0.3606	0.785
medium	NDN_WiFi	0.3903	0.201	0.3957	0.099	0.0512	0.1251	0.3601	0.5599	0.3097
medium	QoSA-ICN_DSRC	0.2495	0.2272	0.171	0.0382	0.0361	0.0639	0.7275	0.7227	0.7203
medium	QoSA-ICN_WiFi	0.7056	0.6474	0.7278	0.1247	0.0641	0.0283	0.0765	0.1114	0.0629
sparse	IP_DSRC	0.3102	0.1579	0.3192	0.5515	0.7123	0.4862	0.0188	0.0171	0.0257
sparse	IP_WiFi	0.3609	0.3369	0.3142	0.2033	0.2608	0.2959	0.1091	0.0992	0.0833
sparse	NDN_DSRC	0.0516	0.0543	0.0247	0.0797	0.2911	0.0695	0.6336	0.4223	0.7021
sparse	NDN_WiFi	0.417	0.3286	0.3309	0.0229	0.028	0.0593	0.3079	0.3573	0.3249
sparse	QoSA-ICN_DSRC	0.1073	0.1991	0.1824	0.1064	0.0505	0.1068	0.5043	0.4176	0.3249
sparse	QoSA-ICN_WiFi	0.3559	0.2711	0.2386	0.0098	0.0122	0.0243	0.4166	0.4507	0.4464

Interpret η 2 as for r2 or R2; a rule of thumb (Cohen):

0.02	~ small
0.13	~ medium
0.26	~ large

Appendix C

Function1 Pseudo-code: NDN-Consumer Send QoS Aware Interest Packet (with QoSInfo)

Function 1 NDN-Consumer Send QoS Aware Interest Packet (with QoSInfo)

```
Precondition: QoS Aware Packets are classified as Critical, High, Medium and
    Low priorities and are denoted with QoSPrefixes such as /ndn_qos/p1,
    /ndn_qos/p2, /ndn_qos/p3, respectively.
Precondition: Special class called QoSInfo was created which carries the data
    member for parameters such as the timeStamp, metadata, priority-level,
    hopCounts, QoSTimeLimit in milliseconds etc.
function SendPacket
    create new Interest
    if Interest.Name contains QoSPrefixes then
        create new QoSObject
        SET Parameters to QoSObject
        SET QoSInfo to Interest
    end if
    SET Signature to Interest
    send Interest
end function
```

Function2 Pseudo-code: NDN-BestRoute Interest Processing

Function3 Pseudo-code: QoS-Aware Multi-Hop Interest Forwarding

Function4 Pseudo-code: Make exception to Interest Loop sending Duplicate NACK

Function5 Pseudo-code: NDN-Producer Return QoS Aware Data (with QoSInfo)

Function6 Pseudo-code: QoS Aware Multi-Hop Data Forwarding

Function7 Pseudo-code: NDN Traffic Control: Fair-Share, Shuffled Queuing

Function 2 NDN-BestRoute Interest Processing

```
function Process(Interest)
    Name ← Interest.Name
    if Data ← ContentStore.Find(Name) then
        Return (Data)
    else if PitEntry ← PIT.Find(Name) then
        if Interest.Nonce $\in$ PitEntry.NonceLi then
            Return
        end if
        Add Interest.Interface to PitEntry.Incoming
        if PitEntry.RetryTimer is expired then
            Forward(Interest, PitEntry)
            Return
        end if
    else
        PitEntry ← PIT.Create(Interest)
        PitEntry.Incoming ← Interest.Interface
        Forward(Interest, PitEntry)
    end if
end function
```

Function 3 QoS-Aware Multi-Hop Interest Forwarding

```
function Forward(Interest, PitEntry)
    FibEntry ← FIB.Find(Interest.Name)
    if NextHops ← FibEntry.nextHops() then
        for interface in NextHops do
            Process(PITEntry, Interface, Interest)
            Return
        end for
    else if QoSInfo ← Interest.getQoSInfo then
        QoSTimeLeft ← QoSInfo.QoSRTT − QoSInfo.TimeElapsed
        if QoSTimeLeft > 0 then
            Send(PITEntry, Inface.Interface, Interest)
            Return
        end if
        Send Nack(reason NoRoute)
        Return
    else
        Send Nack(reason NoRoute)
        Return
    end if
end function
```

Function 4 Make exception to Interest Loop sending Duplicate NACK

```
function ProcessInterestLoop(InFace, Interest)
    if InFace.Type $\neq$ LINK_TYPE_POINT_TO_POINT then
        Return
    end if
    if QoSEnabled ← Interest.getQoSInfo then
        Return
    end if
    Send Nack(reason Duplicate)
end function
```

Function 5 NDN-Producer Return QoS Aware Data (with QoSInfo)

```
function OnInterest(Interest)
    create new Data
    Name ← Interest.Name
    SET Name to Data
    SET Content to Data
    if QoSInfo ← Interest.getQoSIn then
        SET QoSInfo to Data
    end if
    SET Signature to Data
    Send Data
end function
```

Function 6 QoS Aware Multi-Hop Data Forwarding

```
function OnIncommingData(InFace, Data)
    Name ← Data.Name
    if Name.ViolatingLocalHost then
        Return
    end if
    if Name Not in PITEntry And Data Not QoSAware then
        ProcessUnSolicitedData(Data)
        Return
    end if
    pendingDownStream ← PITEntry.Lookup(Data)
    for Face in pendingDownStream do
        if InFace.Id == Face.Id And Face.LinkType ≠ Ad_Hoc then
            if Data Not QoSAware then
                continue
            end if
        end if
        Forward(Data, Face)
    end for
    if pendingDownStream Is Empty then
        Forward(Data, InFace)
    end if
end function
```

Function 7 NDN Traffic Control: Fair-Share, Shuffled Queueing

```
function TrafficControl.SendPacket(packet)
    Get QoSObject (packet)
    Compute packet.TimeToLive (reuse previous formula)
    if Not RTT elapsed and Root Queue Length < Limit then
        add to Root Queue
    else
        Drop packet
    end if
    for each packet in the Root Queue do
        Peek timestamp and priority
        if packet.TimeToLive > 0 then
            Function EnqueueToInternalQueue(packet) {

            for queue.position = head to Tail (InternalQueue of ID = packet.priority) do
                Compute queued.packet.TimeToLive queued.position.packet
                if packet.TimeToLive < queued.packet.TimeToLive  then
                    Insert packet into queue.position
                end if
            end for
            }
        end if
    end for
    Peek a Packet from each Internal Queue
    Get TTL and Priority level
    select packet with lowest TTL when TTL of all packets are unequal
    or
    select packet with lowest TTL and High priority when TTL of some packets are equal
    Dequeue the selected packet
    Netdevice.transmit(packet)
end function
```

Appendix D

QoSA-ICN new files

The following new source code are added to the codebase to support the QoSA-ICN implementation.

1. ../ns-3/src/ndnSIM/ndn-cxx/src/qos-info.cpp:

   Contains the QoS information class definition: data member declarations and methods

2. ../ns-3/src/ndnSIM/ndn-cxx/src/qos-info.hpp:

   Implements qos-info.hpp class definitions, with the main roles of including accessor methods for the QoS metrics and code to convert them into TLV format, in preparation of their communication across the network.

3. ../ns-3/src/ndnSIM/NFD/daemon/fw/qos-aware-best-route-strategy.hpp:

   Contains the definition of the QoS-aware best route strategy. The best route strategy is extended, inheriting from the Strategy class.

4. ../ns-3/src/ndnSIM/NFD/daemon/fw/qos-aware-best-route-strategy.cpp:

   Implements qos-aware-best-route-strategy.hpp. The main roles include checking packets for QoS metrics. If no routes/next hops are found, the message is re-broadcast. If the packet's QoS conditions qualify, the best route strategy is followed .

5. ../ns-3/src/mobility/helper/mobility-tcl-file-parser.h:

   This is a utility class defined for parsing the SUMO generated mobility TCL file.

6. ../ns-3/src/mobility/helper/mobility-tcl-file-parser.cc:

   Implements mobility-tcl-file-parser.h. Pre-processing of mobility trace files is essential to read and save the start/stop time of each node in the simulation environment. This enables and disables the node network activities when the node enters and leaves the road segments, respectively.

7. ../ns-3/src/ndnSIM/model/ndn-qos-queue-disc.hpp:

   Definition of class to interface and interact with QoS NDN queues for traffic control operation.

8. ../ns-3/src/ndnSIM/model/ndn-qos-queue-disc.cpp:

   Implements ndn-qos-queue-disc.hpp. Most of the functions are ported from NS3 queue-disc.hpp to implement traffic control in ndnSIM.

9. ../ns-3/src/ndnSIM/ns-3/src/network/model/ndn-qos-tag.h:

   Definition of class that provides interfaces for the NDN QoS tag used in traffic control

10. ../ns-3/src/ndnSIM/ns-3/src/network/model/ndn-qos-tag.cc:

    Implements ndn-qos-queue-disc.hpp, with accessors for the QoS metrics, which expose the metrics at the data link layer for traffic control

11. ../ns-3/src/ndnSIM/ns-3/src/network/utils/qos-priority-queue.h:

    Class definition and interface for QoS priority queue

12. ../ns-3/src/ndnSIM/ns-3/src/network/utils/qos-priority-queue.cc:

    Implements qos-priority-queue.h. The main roles include enqueuer, dequeuer and peek operations on packets.

13. ../ns-3/src/ndnSIM/ns-3/src/traffic-control/model/ndn-qos-priority-queue-disc.h:

    Class definition and interfaces for the root queue type i.e., NDN QoS priority queue.

14. ../ns-3/src/ndnSIM/ns-3/src/traffic-control/model/ndn-qos-priority-queue-disc.cc:

    Implements ndn-qos-priority-queue-disc.h. The main roles include packet screening for TTL and early drops (i.e., packets no longer considered for forwarding), and distribution of packets to individual internal queues, based on the priority.

15. ../ns-3/src/dnSIM/ns-3/src/ndnSIM/utils/tracers/ip-app-delay-tracer-qos-exp.hpp:

Class definition and interfaces for tracing application delays in the UDP/IP experiments

16. ../ns-3/src/ndnSIM/ns-3/src/ndnSIM/utils/tracers/ip-app-delay tracer-qos-exp.cpp:

    Implements ip-app-delay-tracer-qos-exp.hpp The main role is to log consumer interest sent time, producer interest receive time, producer data sent time and consumer data receive time for each packet in the IP simulation environment.

17. ../ns-3/src/ndnSIM/ns-3/src/ndnSIM/utils/tracers/ndn-l3-rate-tracer-qos-exp.hpp:

    Class definition and interfaces for tracing application delays in the NDN experiments.

18. ../ns-3/src/ndnSIM/ns-3/src/ndnSIM/utils/tracers/ndn-l3-rate-tracer-qos-exp.cpp:

    Implements ndn-app-delay-tracer-qos-exp.hpp. The main role is to log consumer interest sent time, producer interest receive time, producer data sent time and consumer data receive time for each packets in default NDN, QoSA-ICN simulation environment.

19. ../ns-3/src/ndnSIM/ns-3/src/ndnSIM/apps/ip-app.hpp:

    Base class definition for IP application, which extends the Application class

20. ../ns-3/src/ndnSIM/ns-3/src/ndnSIM/apps/ip-app.cpp:

    Implements ip-app.hpp. The main role is to create and sent interest packets from IP consumers, using the UDP/IP protocol

21. ../ns-3/src/ndnSIM/ns-3/src/ndnSIM/apps/ip-consumer.hpp:

    Class definition for the IP producer, with interfaces to create and receive interest packets and reply with data packets, using UDP/IP protocol

22. ../ns-3/src/ndnSIM/ns-3/src/ndnSIM/apps/ip-consumer.cpp:

    Implements the ip-consumer.hpp. The main roles include: to stop, start and configure an IP application on the UDP server; to compose and send requests as per the configured rate; and to receive and process incoming data packets.

23. /ns-3/src/ndnSIM/ns-3/src/ndnSIM/apps/ip-consumer-cbr.cpp:

    Implements the ip-producer.hpp. The main roles include: to stop, start and configure an IP application on the UDP server; to receive and process interest request packets; and to send reply using the UDP/IP protocol.

24. ip-sumo-mobility-parameterised-experiments.cpp:

    This is the IP simulation's main script, which takes a range of inputs as command line parameters (speed, producer/consumer percentages, interest request rates, mobility trace file, etc.). It creates nodes, network devices and configures/assigns network devices to nodes. t generates IP consumer and IP producer distributions among the nodes and assigns nodes using the mobility trace file. It defines each node's start and stop time. It configures tracing and runs the simulation.

25. ndn-sumo-mobility-parameterised-experiments.cpp:

    This is the NDN simulation's main script. It takes a range of inputs as command line parameters (speed, producer/consumer percentages, interest request rates, mobility trace file, etc.). It creates nodes, network devices and configures/assigns network devices to nodes. It generates NDN consumer and NDN producer distributions among the nodes, and assigns nodes using the mobility trace file. It defines each node's start and stop time. It configures tracing and runs the simulation.

Files modified -NDN

The following source code has been modified/extended to integrate the QoS-aware ICN implementation with the default NDN build.

1. ../ns-3/src/ndnSIM/ndn-cxx/src/interest.hpp:

    Extends the NDN Interest packet to include QoSInfo objects at type-length-value (TLV) number 200.

2. ./ns-3/src/ndnSIM/ndn-cxx/src/interest.cpp:

    Implements interest.hpp. Its main role is to wire encode and wire decode the additional QOS information.

3. ../ns-3/src/ndnSIM/ndn-cxx/src/data.hpp:

    Extends the NDN Data packet to include QoSInfo objects at TLV number 200.

4. ../ns-3/src/ndnSIM/ndn-cxx/src/data.cpp:

   Implements data.hpp. The send data method is extended to query QoS metrics from an Interest packet and add to the corresponding Data packet.

5. ../ns-3/src/ndnSIM/NFD/core/common.hpp:

   New and extended class definition header files related QoS NDN implementation are added.

6. ../ns-3/src/ndnSIM/NFD/daemon/fw/forwarder.hpp:

   New methods added to Forwarder to query the QoS packet.

7. ../ns-3/src/ndnSIM/NFD/daemon/fw/forwarder.cpp:

   forwarder.cpp is extended to query a packet for for its QoSInfo, retrieve/process QoS metrics and make forwarding decisions.

8. ../ns-3/src/ndnSIM/NFD/daemon/face/generic-link-service.cpp:

   The implementation is extended to copy QoS metrics from NDN packets (Interest/Data) to NDN QoS tag, for traffic control.

9. ../ns-3/src/ndnSIM/model/ndn-common.hpp:

   Updated with NDN QoSInfo namespace.

10. ../ns-3/src/ndnSIM/model/ndn-net-device-transport.hpp:

    The implementation is extended to include traffic control member.

11. ../ns-3/src/ndnSIM/model/ndn-net-device-transport.cpp:

    The doSend method is extended to send the NS3 packet to traffic controller instead of net device.

12. ../ns-3/src/ndnSIM/apps/ndn-consumer.cpp:

    The consumer implementation is extended to instantiate QoSInfo instance, set values and attach to the Interest packet. The trace sources are also enabled for transmitted interest and received data.

13. ../ns-3/src/ndnSIM/apps/ndn-producer.cpp:

    The ndn-producer implementation is extended to enable the trace sources for interest packet received and data packets sent. Also, the QoSInfo Object is sent from the interest to the data packet.

14. ../ns-3/src/ndnSIM/helper/ndn-stack-helper.hpp:

    Extends the stack helper, included traffic control data member to aggregate the traffic control module.

15. ../ns-3/src/ndnSIM/helper/ndn-stack-helper.cpp:

    Instantiates the traffic control object and aggregates with NDN protocol.

16. ../ns-3/src/ndnSIM/ndn-cxx/encoding/tlv.hpp: QoS Extension to TLV values are added as follows: QoSAwareInfo = 200, QoSFlag = 201, QoSTimeStamp = 202, QoSPriority = 203, QoSMaxHopCount = 204, QosHopCount = 205, QoSMetaInfo = 206

17. /ns-3/src/ndnSIM/ndn-cxx/lp/fields.hpp: Field declarations are wire encoded QoS-related TLV values

18. ../ns-3/src/ndnSIM/ndn-cxx/lp/packet.hpp:

    Extended to include NDN QoS packet tag.

19. ../ns-3/src/ndnSIM/ns-3/src/network/model/packet.h:

    Inline member GetTimeStamp(void) and GetTimeStamp are added to work with the QueuDiscItem template for traffic control at the data link layer.

20. ../ns-3/src/ndnSIM/ns-3/src/traffic-control/model/queue-disc.cc: NS3 network simulator object template class defined with QoSPriorityQueue and QueueDiscItem.

Bibliography

[udp,] Advantages of udp and disadvantages of udp. `http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-UDP.html`. Last accessed: 2019-01-25.

[bra, 2018] (2018). Stopping distance and reaction time. `http://korkortonline.se/en/theory/reaction-braking-stopping`. Last accessed 22 January 2019.

[ICN, 2019] (2019). Information-centric networking research group. `https://trac.ietf.org/trac/irtf/wiki/icnrg`. Last accessed: 2019-01-25.

[IRT, 2019] (2019). The internet research task force. `https://https://irtf.org/`. Last accessed: 2019-01-25.

[Afanasyev et al., 2014] Afanasyev, A., Shi, J., Zhang, B., Zhang, L., Moiseenko, I., Yu, Y., Shang, W., Huang, Y., Abraham, J. P., DiBenedetto, S., et al. (2014). Nfd developers guide. *Dept. Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0021*.

[Ahmed et al., 2015] Ahmed, S. H., Bouk, S. H., and Kim, D. (2015). Rufs: Robust forwarder selection in vehicular content-centric networks. *IEEE Communications Letters*, 19(9):1616–1619.

[Akkaya and Younis, 2003] Akkaya, K. and Younis, M. (2003). An energy-aware qos routing protocol for wireless sensor networks. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 710–715. IEEE.

[Al-Naday et al., 2014] Al-Naday, M. F., Bontozoglou, A., Vassilakis, V. G., and Reed, M. J. (2014). Quality of service in an information-centric network. In *2014 IEEE Global Communications Conference*, pages 1861–1866. IEEE.

[Al-Turjman, 2018] Al-Turjman, F. (2018). Qos-aware data delivery framework for safety-inspired multimedia in integrated vehicular-iot. *Computer Communications*, 121:33–43.

[Amadeo et al., 2012] Amadeo, M., Campolo, C., and Molinaro, A. (2012). Crown: Content-centric networking in vehicular ad hoc networks. *IEEE Communications Letters*, 16(9):1380–1383.

[Amadeo et al., 2015] Amadeo, M., Campolo, C., and Molinaro, A. (2015). Forwarding strategies in named data wireless ad hoc networks: Design and evaluation. *Journal of Network and Computer Applications*, 50:148–158.

[Amadeo et al., 2017] Amadeo, M., Campolo, C., and Molinaro, A. (2017). A novel hybrid forwarding strategy for content delivery in wireless information-centric networks. *Computer Communications*, 109:104 – 116.

[Amadeo et al., 2016] Amadeo, M., Campolo, C., Quevedo, J., Corujo, D., Molinaro, A., Iera, A., Aguiar, R. L., and Vasilakos, A. V. (2016). Information-centric networking for the internet of things: challenges and opportunities. *IEEE Network*, 30(2):92–100.

[Amadeo and Molinaro, 2011] Amadeo, M. and Molinaro, A. (2011). Chanet: A content-centric architecture for ieee 802.11 manets. In *Network of the Future (NOF), 2011 International Conference on the*, pages 122–127. IEEE.

[Amadeo et al., 2013] Amadeo, M., Molinaro, A., and Ruggeri, G. (2013). E-chanet: Routing, forwarding and transport in information-centric multihop wireless networks. *Computer communications*, 36(7):792–803.

[Angius et al., 2012] Angius, F., Gerla, M., and Pau, G. (2012). Bloogo: Bloom filter based gossip algorithm for wireless ndn. pages 25–30.

[Arnould et al., 2011] Arnould, G., Khadraoui, D., and Habbas, Z. (2011). A self-organizing content centric network model for hybrid vehicular ad-hoc networks. In *Proceedings of the first ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 15–22. ACM.

[Arumaithurai et al., 2013] Arumaithurai, M., Seedorf, J., Tagami, A., Ramakrishnan, K., and Melazzi, N. B. (2013). Using icn in disaster scenarios. *draft-seedorf-icn-disaster-01*.

[Baccelli et al., 2010] Baccelli, E., Clausen, T., and Wakikawa, R. (2010). Ipv6 operation for wave - wireless access in vehicular environments. In *2010 IEEE Vehicular Networking Conference*, pages 160–165.

[Bai and Krishnamachari, 2010] Bai, F. and Krishnamachari, B. (2010). Exploiting the wisdom of the crowd: localized, distributed information-centric vanets [topics in automotive networking]. *IEEE Communications Magazine*, 48(5).

[Basagni et al., 2004] Basagni, S., Conti, M., Giordano, S., and Stojmenovic, I. (2004). *Mobile ad hoc networking*. John Wiley & Sons.

[Blake et al., 1998] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1998). An architecture for differentiated services.

[BONVOYAGE, 2018] BONVOYAGE (2018). Bonvoyage. `http://bonvoyage2020.eu/`. Last accessed 22 January 2019.

[Borcoci et al., 2017] Borcoci, E., Ambarus, T., and Vochin, M. (2017). Distributed control plane optimization in sdn-fog vanet. *ICN 2017*, page 135.

[Boulila et al., 2018] Boulila, N., Hadded, M., Laouiti, A., and Saidane, L. A. (2018). Qch-mac: A qos-aware centralized hybrid mac protocol for vehicular ad hoc networks. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 55–62. IEEE.

[Britz, 2014] Britz, D. M. (2014). Wireless big data are our wireless edge networks ready? In *Infrared, Millimeter, and Terahertz waves (IRMMW-THz), 2014 39th International Conference on*, pages 1–4. IEEE.

[Castellanos et al., 2016] Castellanos, W. E., Guerri, J. C., and Arce, P. (2016). A qos-aware routing protocol with adaptive feedback scheme for video streaming for mobile networks. *Computer Communications*, 77.

[Céspedes et al., 2013] Céspedes, S., Lu, N., and Shen, X. (2013). Vip-wave: On the feasibility of ip communications in 802.11 p vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):82–97.

[Chen and Nahrstedt, 1999] Chen, S. and Nahrstedt, K. (1999). Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected areas in Communications*, 17(8):1488–1505.

[Chrysostomou et al., 2012] Chrysostomou, C., Djouvas, C., and Lambrinos, L. (2012). Dynamically adjusting the min-max contention window for providing quality of service in vehicular networks. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, pages 16–23. IEEE.

[Chu et al., 2016] Chu, W., Wang, L., Xie, H., Zhang, Z.-L., and Jiang, Z. (2016). Network delay guarantee for differentiated services in content-centric networking. *Computer Communications*, 76:54–66.

[Cisco, 2017] Cisco, V. (2017). The zettabyte era: trends and analysis. updated (07/06/2017).

[Cohen, 1977] Cohen, J. (1977). Statistical power analysis for the behavioral sciences, rev.

[COMET, 2010] COMET (2010). Content mediator architecture for content-aware networks. `http://www.comet-project.org/index.html`. Last accessed 22 January 2019.

[Convergence, 2010] Convergence (2010). The convergence project. `http://www.ict-convergence.eu/`. Last accessed 22 January 2019.

[Dannewitz et al., 2013] Dannewitz, C., Kutscher, D., Ohlman, B., Farrell, S., Ahlgren, B., and Karl, H. (2013). Network of information (netinf)–an information-centric networking architecture. *Computer Communications*, 36(7):721–735.

[Dharmaraju et al., 2002] Dharmaraju, D., Roy-Chowdhury, A., Hovareshti, P., and Baras, J. S. (2002). Inora-a unified signaling and routing mechanism for qos support in mobile ad hoc networks. In *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, pages 86–93. IEEE.

[Dilley et al., 2002] Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., and Weihl, B. (2002). Globally distributed content delivery. *IEEE Internet Computing*, 6(5):50–58.

[Dua et al., 2015] Dua, A., Kumar, N., and Bawa, S. (2015). Qos-aware data dissemination for dense urban regions in vehicular ad hoc networks. *Mobile Networks and Applications*, 20(6):773–780.

[Dynerowicz and Mendes, 2017] Dynerowicz, S. and Mendes, P. (2017).

named-data networking in opportunistic network. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, pages 220–221. ACM.

[eCOUSIN, 2015] eCOUSIN (2015). Content mediator architecture for content-aware networks. Last accessed 22 January 2019.

[Eiza et al., 2016] Eiza, M. H., Owens, T., and Ni, Q. (2016). Secure and robust multi-constrained qos aware routing algorithm for vanets. *IEEE Transactions on Dependable and Secure Computing*, 13(1):32–45.

[Eiza et al., 2015] Eiza, M. H., Owens, T., Ni, Q., and Shi, Q. (2015). Situation-aware qos routing algorithm for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 64(12):5520–5535.

[Fall and Stevens, 2011] Fall, K. R. and Stevens, W. R. (2011). *TCP/IP illustrated, volume 1: The protocols*. addison-Wesley.

[Firoze et al., 2007] Firoze, A., Jun, L., and Kwong, L. (2007). Pr-mac a priority reservation mac protocol for wireless sensor networks. In *Electrical Engineering, 2007. ICEE'07. International Conference on*, pages 1–6. IEEE.

[García et al., 2011] García, G., Beben, A., Ramón, F. J., Maeso, A., Psaras, I., Pavlou, G., Wang, N., Śliwiński, J., Spirou, S., Soursos, S., et al. (2011). Comet: Content mediator architecture for content-aware networks. In *Future Network & Mobile Summit (FutureNetw), 2011*, pages 1–8. IEEE.

[Grassi et al., 2015] Grassi, G., Pesavento, D., Pau, G., Zhang, L., and Fdida, S. (2015). Navigo: Interest forwarding by geolocations in vehicular named data networking. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, pages 1–10. IEEE.

[Haklay and Weber, 2008] Haklay, M. and Weber, P. (2008). Openstreetmap: User-generated street maps. *Ieee Pervas Comput*, 7(4):12–18.

[Han et al., 2014] Han, H., Wu, M., Hu, Q., and Wang, N. (2014). Best route, error broadcast: a content-centric forwarding protocol for manets. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, pages 1–5. IEEE.

[Hartenstein and Laberteaux, 2008] Hartenstein, H. and Laberteaux, L. (2008). A tutorial survey on vehicular ad hoc networks. *IEEE Communications magazine*, 46(6):164–171.

[Hassan and Muniyandi, 2017] Hassan, M. H. and Muniyandi, R. C. (2017). An improved hybrid technique for energy and delay routing in mobile ad-hoc networks. *International Journal of Applied Engineering Research*, 12(1):134–139.

[Hu et al., 2015] Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., and Young, V. (2015). Mobile edge computing-a key technology towards 5g. *ETSI white paper*, 11(11):1–16.

[IEEE, 2010] IEEE (2010). Ieee draft standard for wireless access in vehicular environments (wave) - networking services. *IEEE P1609.3/D9, August 2010*, pages 1–133.

[INTEL, 2019] INTEL (2019). Data is the new oil. `https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving`.

[Intel and NSF, 2018] Intel and NSF, N. S. (2018). Nsf/intel partnership on information-centric networking in wireless edge networks.

[INTENT, 2015] INTENT (2015). Information-centric network management and traffic engineering. `https://cordis.europa.eu/project/rcn/188247/factsheet/en`. Last accessed 22 January 2019.

[Jangam et al., 2018] Jangam, A., Suthar, P., and Stolic, M. (2018). Supporting QoS aware Data Delivery in Information Centric Networks. Internet-Draft draft-anilj-icnrg-icn-qos-00, Internet Engineering Task Force. Work in Progress.

[Jiang and Delgrossi, 2008] Jiang, D. and Delgrossi, L. (2008). Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2036–2040.

[Jmal and Fourati, 2017] Jmal, R. and Fourati, L. C. (2017). Emerging applications for future internet approach based-on sdn and icn. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 208–213. IEEE.

[Kalogeiton et al., 2017] Kalogeiton, E., Kolonko, T., and Braun, T. (2017). A multihop and multipath routing protocol using NDN for VANETs. *2017 16th Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2017*.

[Kaur et al., 2019] Kaur, S., Aseri, T. C., and Rani, S. (2019). Qos-aware routing in vehicular ad hoc networks using ant colony optimization and bee colony optimization. In *Proceedings of 2nd International Conference on Communication, Computing and Networking*, pages 251–260. Springer.

[Kim et al., 2016] Kim, D., Kim, J.-h., Moon, C., Choi, J., and Yeom, I. (2016). Efficient content delivery in mobile ad-hoc networks using ccn. *Ad Hoc Networks*, 36:81–99.

[Kim and Min, 2009] Kim, H. and Min, S.-G. (2009). Priority-based qos mac protocol for wireless sensor networks. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8. IEEE.

[Kopetz, 2011] Kopetz, H. (2011). *Real-time systems: design principles for distributed embedded applications*. Springer Science & Business Media.

[Koponen et al., 2007] Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I. (2007). A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 181–192. ACM.

[Krajzewicz et al., 2012] Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.

[Kuang and Yu,] Kuang, J. and Yu, S.-Z. Bandwidth-based qos-aware multisource architecture for information-centric wireless multihop networks.

[Kuang and Yu, 2015] Kuang, J. and Yu, S.-Z. (2015). Csar: A content-scent based architecture for information-centric mobile ad hoc networks. *Computer Communications*, 71:84–96.

[Lee, 2008] Lee, E. A. (2008). Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE.

[Lee et al., 2000] Lee, S.-B., Ahn, G.-S., Zhang, X., and Campbell, A. T. (2000). Insignia: An ip-based quality of service framework for mobile ad hoc networks. *Journal of Parallel and distributed Computing*, 60(4):374–406.

[Li et al., 2018] Li, Y., Shi, X., Lindgren, A., Hu, Z., Zhang, P., Jin, D., and Zhou, Y. (2018). Context-aware data dissemination for icn-based vehicular ad hoc networks. *Information*, 9(11):263.

[Liang et al., 2015] Liang, C., Yu, F. R., and Zhang, X. (2015). Information-centric network function virtualization over 5g mobile wireless networks. *IEEE Network*, 29(3):68–74.

[Lin and Liu, 1999] Lin, C. R. and Liu, J.-S. (1999). Qos routing in ad hoc wireless networks. *IEEE Journal on selected areas in communications*, 17(8):1426–1438.

[Liu and Yu, 2014] Liu, W. and Yu, M. (2014). Aasr: Authenticated anonymous secure routing for manets in adversarial environments. *IEEE Trans. Vehicular Technology*, 63(9):4585–4593.

[Liu et al., 2017] Liu, X., Li, Z., Yang, P., and Dong, Y. (2017). Information-centric mobile ad hoc networks and content routing: A survey. *Ad Hoc Networks*, 58:255 – 268. Hybrid Wireless Ad Hoc Networks.

[Liu et al., 2005] Liu, Y., Elhanany, I., and Qi, H. (2005). An energy-efficient qos-aware media access control protocol for wireless sensor networks. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 3–pp. IEEE.

[Liu and Elhanany, 2006] Liu, Z. and Elhanany, I. (2006). Rl-mac: A qos-aware reinforcement learning based mac protocol for wireless sensor networks. In *Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on*, pages 768–773. IEEE.

[Loo et al., 2016] Loo, J., Mauri, J. L., and Ortiz, J. H. (2016). *Mobile ad hoc networks: current status and future trends*. CRC Press.

[Lu and Zhu, 2013] Lu, T. and Zhu, J. (2013). Genetic algorithm for energy-efficient qos multicast routing. *IEEE Communications Letters*, 17(1):31–34.

[MacGillivray et al., 2016] MacGillivray, C., Turner, V., Clarke, R., Feblowitz, J., Knickle, K., Lamy, L., Xiang, M., Siviero, A., and Cansfield, M. (2016). Idc future scape: Worldwide internet of things 2017 predictions. In *IDC Web Conference*.

[Mahadevan, 2014] Mahadevan, P. (2014). Ccnx 1.0 tutorial. *Mar*, 16:1–11.

[Mastorakis et al., 2015] Mastorakis, S., Afanasyev, A., Moiseenko, I., and Zhang, L. (2015). ndnsim 2.0: A new version of the ndn simulator for ns-3. *NDN, Technical Report NDN-0028*.

[Mastorakis et al., 2017] Mastorakis, S., Afanasyev, A., and Zhang, L. (2017). On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation. *ACM SIGCOMM Computer Communication Review*, 47(3).

[Meisel et al., 2010a] Meisel, M., Pappas, V., and Zhang, L. (2010a). Ad hoc networking via named data. In *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*, pages 3–8. ACM.

[Meisel et al., 2010b] Meisel, M., Pappas, V., and Zhang, L. (2010b). Listen first, broadcast later: Topology-agnostic forwarding under high dynamics. In *Annual conference of international technology alliance in network and information science*, page 8.

[Mobileeye, 2018] Mobileeye (2018). Advanced driver assistance systems. Last accessed 23 April 2019.

[Modesto and Boukerche, 2018] Modesto, F. M. and Boukerche, A. (2018). Seven: A novel service-based architecture for information-centric vehicular network. *Computer Communications*, 117:133–146.

[Moustafa et al., 2017] Moustafa, H., Schooler, E. M., and McCarthy, J. (2017). Reverse cdn in fog computing: The lifecycle of video data in connected and autonomous vehicles. In *2017 IEEE Fog World Congress (FWC)*, pages 1–5.

[Muscariello et al., 2018] Muscariello, L., Carofiglio, G., Auge, J., and Papalini, M. (2018). Hybrid information-centric networking. Technical report, Internet-Draft draft-muscariello-intareahicn-00. Internet Engineering Task.

[named data.net,] named data.net. Nfd - named data networking forwarding daemon documentation. `https://named-data.net/doc/NFD/current/manpages/nfdc-route.html`. Last accessed: 2019-01-25.

[nationmaster.com,] nationmaster.com. Countries compared across road densities. `http://www.nationmaster.com/country-info/stats/Transport/Road-density/Km-of-road-per-100-sq.-km-of-land-area`. Last accessed: 2019-01-25.

[NDN, 2014] NDN (2014). Ndn technical memo: Naming conventions. Last accessed 23 April 2019.

[networking Index, 2016] networking Index, C. V. (2016). Forecast and methodology, 2016-2021, white paper. *San Jose, CA, USA*, 1.

[Nichols et al., 1998] Nichols, K., Blake, S., Baker, F., and Black, D. (1998). Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. Technical report.

[Ning et al., 2018] Ning, Z., Hu, X., Chen, Z., Zhou, M., Hu, B., Cheng, J., and Obaidat, M. S. (2018). A cooperative quality-aware service access system for social internet of vehicles. *IEEE Internet of Things Journal*, 5(4):2506–2517.

[NS-3, 2019] NS-3 (2019). ns-3. `https://github.com/named-data-ndnSIM/ns-3-dev.git`. Last accessed: 2019-01-25.

[ns 3/src/ndnSIM, 2019] ns 3/src/ndnSIM (2019). ns-3/src/ndnsim. `https://github.com/named-data-ndnSIM/ndnSIM.git`. Last accessed: 2019-01-25.

[OCF, 2018] OCF (2018). OpenFog (OCF). `https://www.openfogconsortium.org`. Last accessed: 2019-01-25.

[Oh et al., 2010] Oh, S.-Y., Lau, D., and Gerla, M. (2010). Content centric networking in tactical and emergency manets. *Wireless days*, 10.

[Pavlou and Psaras, 2018] Pavlou, G. and Psaras, I. (2018). The troubled journey of qos: From atm to content networking, edge-computing and distributed internet governance. *Computer Communications*, 131:8 – 12. COMCOM 40 years.

[Pentikousis et al., 2015] Pentikousis, K., Ohlman, B., Corujo, D., Boggia, G., Tyson, G., Davies, E., Molinaro, A., and Eum, S. (2015). Information-centric networking: baseline scenarios. Technical report.

[Pham et al., 2015] Pham, Q.-V., Kim, H., and Hwang, W.-J. (2015). Globally optimal solutions for cross-layer design in fast-fading lossy delay-constrained manets. 18(2):168–177.

[Piro et al., 2014] Piro, G., Grieco, L. A., Boggia, G., and Chatzimisios, P. (2014). Information-centric networking and multimedia services: present and future challenges. *Transactions on Emerging Telecommunications Technologies*, 25(4):392–406.

[Piro et al., 2017] Piro, G., Signorello, S., Palattella, M. R., Grieco, L. A., Boggia, G., and Engel, T. (2017). Understanding the social impact of icn: between myth and reality. *AI & SOCIETY*, 32(3):401–419.

[PURSUIT, 2010] PURSUIT (2010). Pursuit icn. `http://www.fp7-pursuit.eu/PursuitWeb/`. Last accessed 22 January 2018.

[pybindgen, 2019] pybindgen (2019). pybindgen. `https://github.com/named-data-ndnSIM/pybindgen.git`. Last accessed: 2019-01-25.

[Ravindran et al., 2017] Ravindran, R., Chakraborti, A., Amin, S. O., Azgin, A., and Wang, G. (2017). 5g-icn: Delivering icn services over 5g using network slicing. *IEEE Communications Magazine*, 55(5):101–107.

[Ren et al., 2016] Ren, Y., Li, J., Shi, S., Li, L., Wang, G., and Zhang, B. (2016). Congestion control in named data networking–a survey. *Computer Communications*, 86:1–11.

[Richardson, 2011] Richardson, J. T. (2011). Eta squared and partial eta squared as measures of effect size in educational research. *Educational Research Review*, 6(2):135–147.

[Rosen et al., 2000a] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and Conta, A. (2000a). Mpls label stack encoding. Technical report.

[Rosen et al., 2000b] Rosen, E., Viswanathan, A., and Callon, R. (2000b). Multiprotocol label switching architecture. Technical report.

[Sadiku et al., 2014] Sadiku, M. N., Musa, S. M., and Momoh, O. D. (2014). Cloud computing: opportunities and challenges. *IEEE potentials*, 33(1):34–36.

[Salkuyeh and Abolhassani, 2016] Salkuyeh, M. A. and Abolhassani, B. (2016). An adaptive multipath geographic routing for video transmission in urban vanets. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2822–2831.

[Salsano et al., 2013] Salsano, S., Blefari-Melazzi, N., Detti, A., Morabito, G., and Veltri, L. (2013). Information centric networking over sdn and openflow: Architectural aspects and experiments on the ofelia testbed. *Computer Networks*, 57(16):3207–3221.

[Saxena et al., 2008] Saxena, N., Roy, A., and Shin, J. (2008). Dynamic duty cycle and adaptive contention window based qos-mac protocol for wireless multimedia sensor networks. *Computer Networks*, 52(13):2532–2542.

[Schneider et al., 2016] Schneider, K., Yi, C., Zhang, B., and Zhang, L. (2016). A practical congestion control scheme for named data networking. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pages 21–30. ACM.

[Schollmeier, 2001] Schollmeier, R. (2001). A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*, pages 101–102. IEEE.

[Seskar et al., 2011] Seskar, I., Nagaraja, K., Nelson, S., and Raychaudhuri, D. (2011). Mobilityfirst future internet architecture project. In *Proceedings of the 7th Asian Internet Engineering Conference*, pages 1–3. ACM.

[Shah et al., 2018] Shah, A. S., Ilhan, H., and Tureli, U. (2018). qcb-mac: Qos aware cluster-based mac protocol for vanets. In *Science and Information Conference*, pages 685–695. Springer.

[Shah et al., 2011] Shah, N., Qian, D.-P., and Wang, R. (2011). An efficient unstructured p2p overlay for file sharing over manets using underlying reactive routing. 12(3):517–530.

[Shannigrahi et al., 2018] Shannigrahi, S., Fan, C., and White, G. (2018). Bridging the icn deployment gap with ipoc: An ip-over-icn protocol for 5g networks. In *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*, pages 1–7. ACM.

[Shi and Zhang, 2012] Shi, J. and Zhang, B. (2012). Ndnlp: A link protocol for ndn. *The University of Arizona, Tucson, AZ, NDN Technical Report NDN-0006*.

[Shin and Ramanathan, 1994] Shin, K. G. and Ramanathan, P. (1994). Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24.

[Sivakumar et al., 1999] Sivakumar, R., Sinha, P., and Bharghavan, V. (1999). Cedar: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in communications*, 17(8):1454–1465.

[Slama et al., 2008] Slama, I., Shrestha, B., Jouaber, B., and Zeghlache, D. (2008). A hybrid mac with prioritization for wireless sensor networks. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pages 274–281. IEEE.

[Sommer et al., 2008] Sommer, C., Yao, Z., German, R., and Dressler, F. (2008). On the need for bidirectional coupling of road traffic microsimulation and network simulation. *Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models - MobilityModels '08*, page 41.

[STATISTA, 2018] STATISTA (2018). Connected car world outlook. `http://www.ict-convergence.eu/`. Last accessed 22 January 2019.

[Su et al., 2017] Su, Z., Hui, Y., and Yang, Q. (2017). The next generation vehicular networks: A content-centric framework. *IEEE Wireless Communications*, 24(1):60–66.

[Sun et al., 2006] Sun, W., Yamaguchi, H., Yukimasa, K., and Kusumoto, S. (2006). Gvgrid: A qos routing protocol for vehicular ad hoc networks. In *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, pages 130–139. IEEE.

[Suthar and Stolic, 2016] Suthar, P. and Stolic, M. (2016). Deploying information centric networking in lte mobile networks. In *Wireless and Mobile (APWiMob), 2016 IEEE Asia Pacific Conference on*, pages 130–136. IEEE.

[TalebiFard and Leung, 2012] TalebiFard, P. and Leung, V. (2012). 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*, pages 17–24. ACM.

[TalebiFard et al., 2015] TalebiFard, P., Leung, V. C., Amadeo, M., Campolo, C., and Molinaro, A. (2015). Information-centric networking for vanets. In *Vehicular ad hoc Networks*, pages 503–524. Springer.

[Tan et al., 2008] Tan, J., Chan, M.-C., Tan, H.-X., Kong, P.-Y., and Tham, C.-K. (2008). A medium access control protocol for uwb sensor networks with qos support. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pages 289–296. IEEE.

[Toor et al., 2008] Toor, Y., Muhlethaler, P., Laouiti, A., and Fortelle, A. D. L. (2008). Vehicle ad hoc networks: applications and related technical issues. *IEEE Communications Surveys Tutorials*, 10(3):74–88.

[Tsilopoulos, 2011] Tsilopoulos, C. (2011). Supporting diverse traffic types in information centric networks. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 13–18. ACM.

[Ullah et al., 2016] Ullah, S., Thar, K., Alam, M. G. R., Son, J. H., Lee, J. W., and Hong, C. S. (2016). NDN Developer's Guide. *18th Asia-Pacific Network Operations and Management Symposium, APNOMS 2016: Management of Softwarized Infrastructure - Proceedings*, pages 1–72.

[uMOBILE, 2018] uMOBILE (2018). Universal, mobile-centric and opportunistic communications architecture. `http://umobileproject.eu/`. Last accessed 22 January 2019.

[Vasilakos et al., 2015] Vasilakos, A. V., Li, Z., Simon, G., and You, W. (2015). Information centric network: Research challenges and opportunities. *Journal of network and computer applications*, 52:1–10.

[Viriyasitavat et al., 2015] Viriyasitavat, W., Boban, M., Tsai, H.-M., and Vasilakos, A. (2015). Vehicular communications: Survey and challenges of channel and propagation models. *arXiv preprint arXiv:1505.06004*.

[Wang et al., 2010] Wang, J., Wakikawa, R., and Zhang, L. (2010). Dmnd: Collecting data from mobiles using named data. In *Vehicular networking conference (VNC), 2010 IEEE*, pages 49–56. IEEE.

[Wang et al., 2012] Wang, L., Wakikawa, R., Kuntz, R., Vuyyuru, R., and Zhang, L. (2012). Data naming in vehicle-to-vehicle communications. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 328–333. IEEE.

[Wang et al., 2016a] Wang, S., Lei, T., Zhang, L., Hsu, C.-H., and Yang, F. (2016a). Offloading mobile data traffic for qos-aware service provision in vehicular cyber-physical systems. *Future Generation Computer Systems*, 61:118–127.

[Wang et al., 2018] Wang, X., Lv, J., Huang, M., Li, K., Li, J., and Ren, K. (2018). Energy-efficient icn routing mechanism with qos support. *Computer Networks*, 131:38–51.

[Wang et al., 2016b] Wang, Y., Liu, H., Huang, L., and Stankovic, J. (2016b). Efficient and proactive v2v information diffusion using named data networking. In *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on*, pages 1–10. IEEE.

[Wang et al., 2013] Wang, Y., Rozhnova, N., Narayanan, A., Oran, D., and Rhee, I. (2013). An improved hop-by-hop interest shaper for congestion control in named data networking. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 55–60. ACM.

[Wendell and Freedman, 2011] Wendell, P. and Freedman, M. J. (2011). Going viral: flash crowds in an open cdn. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 549–558. ACM.

[Wroclawski, 1997] Wroclawski, J. (1997). The use of rsvp with ietf integrated services. Technical report.

[Wu et al., 2018] Wu, F., Yang, W., Guo, Q., and Xie, X. (2018). Energy efficient multicast in wlan via ndn. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6.

[Xiao et al., 2000] Xiao, H., Seah, W. K. G., Lo, A., and Chua, K. C. (2000). A flexible quality of service model for mobile ad-hoc networks. In *VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No.00CH37026)*, volume 1, pages 445–449 vol.1.

[Xie et al., 2016a] Xie, Y., Bao, J., Song, Z., and Zhou, Y. (2016a). A hybrid opportunistic routing scheme based on nodes grouping strategy for vanets in urban scenarios. In *2016 25th Wireless and Optical Communication Conference (WOCC)*, pages 1–5.

[Xie et al., 2016b] Xie, Y., Bao, J., Song, Z., and Zhou, Y. (2016b). A hybrid opportunistic routing scheme based on nodes grouping strategy for vanets in urban scenarios. In *Wireless and Optical Communication Conference (WOCC), 2016 25th*, pages 1–5. IEEE.

[Xu et al., 2017] Xu, C., Quan, W., Vasilakos, A. V., Zhang, H., and Muntean, G.-M. (2017). Information-centric cost-efficient optimization for multimedia content delivery in mobile vehicular networks. *Computer Communications*, 99:93 – 106.

[Xylomenos et al., 2014] Xylomenos, G., Ververidis, C. N., Siris, V. A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K. V., Polyzos, G. C., et al. (2014). A survey of information-centric networking research. *IEEE Communications Surveys and Tutorials*, 16(2):1024–1049.

[Yahya and Ben-Othman, 2010] Yahya, B. and Ben-Othman, J. (2010). Energy efficient and qos aware medium access control for wireless sensor networks. *Concurrency and Computation: Practice and Experience*, 22(10):1252–1266.

[Yang et al., 2015] Yang, C., Chen, Z., Xia, B., and Wang, J. (2015). When icn meets c-ran for hetnets: an sdn approach. *IEEE Communications Magazine*, 53(11):118–125.

[Yang et al., 2018] Yang, W., Qin, Y., and Yang, Y. (2018). An interest shaping mechanism in ndn: Joint congestion control and traffic management. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

[Yaqub et al., 2016] Yaqub, M. A., Ahmed, S. H., Bouk, S. H., and Kim, D. (2016). Interest forwarding in vehicular information centric networks: a survey. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 724–729. ACM.

[Yavatkar et al., 1999] Yavatkar, R., Pendarakis, D., and Guerin, R. (1999). A framework for policy-based admission control. Technical report.

[Yoon et al., 2007] Yoon, S., Qiao, C., Sudhaakart, R. S., Lit, J., and Talty, T. (2007). Qomor: A qos-aware mac protocol using optimal retransmission for wireless intra-vehicular sensor networks. In *2007 Mobile Networking for Vehicular Environments*, pages 121–126. IEEE.

[Yu et al., 2015] Yu, Y.-T., Gerla, M., and Sanadidi, M. (2015). Scalable vanet content routing using hierarchical bloom filters. *Wireless Communications and Mobile Computing*, 15(6):1001–1014.

[Yu et al., 2013] Yu, Y.-T., Li, Y., Ma, X., Shang, W., Sanadidi, M., and Gerla, M. (2013). Scalable opportunistic vanet content routing with encounter information. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, pages 1–6. IEEE.

[Zaidi et al., 2017] Zaidi, S., Bitam, S., and Mellouk, A. (2017). Enhanced user datagram protocol for video streaming in vanet. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE.

[Zhang and Mouftah, 2005] Zhang, B. and Mouftah, H. T. (2005). Qos routing for wireless ad hoc networks: problems, algorithms, and protocols. *IEEE Communications Magazine*, 43(10):110–117.

[Zhang et al., 2010] Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J. D., Smetters, D. K., Zhang, B., Tsudik, G., Massey, D., Papadopoulos, C., et al. (2010). Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 157:158.

[Zhang et al., 2014] Zhang, L., Jacobson, V., Diego, S., Crowley, P., Louis, S., and Wang, L. (2014). Named Data Networking. *ACM Computer Communication Review*, 44(3):66–73.

[Zhang et al., 2013] Zhang, Y., Raychadhuri, D., Ravindran, R., and Wang, G. (2013). Icn based architecture for iot. *IRTF contribution, October*.

[Zhu and Corson, 2002] Zhu, C. and Corson, M. S. (2002). Qos routing for mobile ad hoc networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 958–967. IEEE.

[Zimmermann, 1980] Zimmermann, H. (1980). Osi reference model–the iso model of architecture for open systems interconnection. *IEEE Transactions on communications*, 28(4):425–432.