Dipl.-Ing. Daniel Posch, BSc.

Towards Effective Multimedia Dissemination in Information-Centric Networks

DISSERTATION

submitted in fulfilment of the requirements for the degree of
Doktor der technischen Wissenschaften

Alpen-Adria-Universität Klagenfurt
Fakultät für Technische Wissenschaften

Mentor
Univ.-Prof. Dipl.-Ing. Dr. Hermann Hellwagner
Alpen-Adria-Universität Klagenfurt, Austria
Department of Informationtechnology (ITEC)

First Evaluator
Univ.-Prof. Dipl.-Ing. Dr. Hermann Hellwagner
Alpen-Adria-Universität Klagenfurt, Austria
Department of Informationtechnology (ITEC)

Second Evaluator
Prof. Dr. George Pavlou MSc.
University College London, United Kingdom
Department of Electronic and Electrical Engineering

Klagenfurt, Dezember 2016

Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich

- die eingereichte wissenschaftliche Arbeit selbständig verfasst und andere als die angegebenen Hilfsmittel nicht benutzt habe;
- die während des Arbeitsvorganges von dritter Seite erfahrene Unterstützung, einschließlich signifikanter Betreuungshinweise, vollständig offengelegt habe;
- die Inhalte, die ich aus Werken Dritter oder eigenen Werken wortwörtlich oder sinngemäß übernommen habe, in geeigneter Form gekennzeichnet und den Ursprung der Information durch möglichst exakte Quellenangaben (z.B. in Fußnoten) ersichtlich gemacht habe;
- die Arbeit bisher weder im Inland noch im Ausland einer Prüfungsbehörde vorgelegt habe und dass
- die zur Plagiatskontrolle eingereichte digitale Version der Arbeit mit der gedruckten Version übereinstimmt.

Ich bin mir bewusst, dass eine tatsachenwidrige Erklärung rechtliche Folgen haben wird.

Affidavit

I hereby declare in lieu of an oath that

- the submitted academic paper is entirely my own work and that no auxiliary materials have been used other than those indicated;
- I have fully disclosed all assistance received from third parties during the process of writing the paper, including any significant advice from supervisors;
- any contents taken from the works of third parties or my own works that have been included either literally or in spirit have been appropriately marked and the respective source of the information has been clearly identified with precise bibliographical references (e.g., in footnotes);
- to date, I have not submitted this paper to an examining authority either in Austria or abroad and that
- the digital version of the paper submitted for the purpose of plagiarism assessment is fully consistent with the printed version.

I am aware that a declaration contrary to the facts will have legal consequences.

Unterschrift/Signature: ________________________

Klagenfurt, 27. Dezember 2016

Table of Contents

Acknowledgments

Writing a thesis is a challenging task that is not easy to master without the support of other people. I consider my self lucky for having a lot of colleagues and friends who provided me support and patience. I want to express my sincere gratitude to:

> My first evaluator and mentor Univ.-Prof. Dipl.-Ing. Dr. Hermann Hellwagner. He has supported me since my early days at Alpen-Adria-Universität Klagenfurt and has not yet tired of doing so. Prof. Hellwagner guided my first steps as a researcher, eventually leading to this thesis. I want to thank him for the opportunity to write this thesis at ITEC, for the excellent cooperation, and for the generous assistance during my studies.
>
> My second evaluator Prof. Dr. George Pavlou MSc. for his willingness to review this thesis. Having such a highly distinguished expert as evaluator is a great honour for me. I want to thank Prof. Pavlou for his time and comments concerning this work.
>
> My colleagues and friends at ITEC. Especially, I want to thank Dipl.-Ing. Dr. Benjamin Rainer for the excellent collaboration resulting in many high quality research papers. I really appreciate his advices, in particular his support when dealing with mathematical challenges (and there were many). Furthermore, I want to thank Dipl.-Ing. Andreas Leibetseder and Sebastian Theuermann for their help in realizing and maintaining the NDN-based testbed.
>
> My family and friends. I want to thank my parents, siblings and grandparents for their generous understanding and support. Particularly, I want to thank Kirandeep, my other half. We all know that researchers are crazy people, but she probably knows best.

Refereed Publications

Parts of this thesis have been published in peer-reviewed journals, conferences and workshops. The following provides a chronological list of all publications that include contents correlated to this thesis:

- **Daniel Posch**, Benjamin Rainer, Hermann Hellwagner. Towards a Context-Aware Forwarding Plane in Named Data Networking Supporting QoS. In *Computer Communication Review*, ACM SIGCOMM, vol. 47, no. 1, New York, USA, pp. 9, 2017.

- **Daniel Posch**, Benjamin Rainer, Hermann Hellwagner. SAF: Stochastic Adaptive Forwarding in Named Data Networking. In *IEEE/ACM Transactions on Networking*, IEEE, vol. 25, no. 2, New York, USA, pp. 14, 2017. [Online] Available: `http://dx.doi.org/10.1109/TNET.2016.2614710`

- Benjamin Rainer, **Daniel Posch**, Andreas Leibetseder, Sebastian Theuermann, Hermann Hellwagner. A Low-Cost NDN Testbed on Banana Pi Routers. In *IEEE Communications Magazine: Network Testing and Analytics Series*, IEEE, vol. 54, no. 9, New York, USA, pp. 6, 2016. [Online] Available: `http://dx.doi.org/10.1109/MCOM.2016.7565256`

- Benjamin Rainer, **Daniel Posch**, Hermann Hellwagner. Investigating the Performance of Pull-based Dynamic Adaptive Streaming in NDN, In *Journal on Selected Areas in Communications (J-SAC): Special Issue on Video Distribution over Future Internet*, IEEE, vol. 34, no. 8, New York, USA, pp. 11, 2016. [Online] Available: `http://dx.doi.org/10.1109/JSAC.2016.2577365`

- **Daniel Posch**, Benjamin Rainer, Sebastian Theuermann, Andreas Leibetseder, Hermann Hellwagner. Emulating NDN-based Multimedia Delivery. In Proceedings of the *7th International Conference on Multimedia Systems* (Christian Timmerer, Ali Begen, eds.), ACM Digital Library, New York, pp. 4, 2016. [Online] Available: `http://dx.doi.org/10.1145/2910017.2910626`

- Cedric Westphal (ed.), Stefan Lederer, **Daniel Posch**, Christian Timmerer, et al. Adaptive Video Streaming over Information-Centric Networking (ICN). In *RFC 7933* (Informational), Internet Engineering Task Force, Information-Centric Networking Research Group (ICNRG), Aug. 2016. [Online] Available: `http://www.ietf.org/rfc/rfc7933.txt`

- Christian Kreuzberger, **Daniel Posch**, Hermann Hellwagner. A Scalable Video Coding Dataset and Toolchain for Dynamic Adaptive Streaming over HTTP. In Proceedings of the *6th ACM*

Multimedia Systems Conference (Tsang Ooi Wei, ed.), ACM, New York, USA, pp. 213-218, 2015. [Online] Available: `http://dx.doi.org/10.1145/2713168.2713193`

- **Daniel Posch**, Christian Kreuzberger, Benjamin Rainer, Hermann Hellwagner. Using In-Network Adaptation to Tackle Inefficiencies Caused by DASH in Information-Centric Networks. In Proceedings of the *10th International Conference on Emerging Networking Experiments and Technologies, VideoNext Workshop* (Colin Dixon, ed.), ACM Digital Library, New York, USA, pp. 1-6, 2014. [Online] Available: `http://dx.doi.org/10.1145/2676652.2676653`

- **Daniel Posch**, Christian Kreuzberger, Benjamin Rainer, Hermann Hellwagner. Client Starvation: A Shortcoming of Client-driven Adaptive Streaming in Named Data Networking. In Proceedings of the *1st ACM Conference on Information-Centric Networking* (Paulo Mendes, ed.), ACM Digital Library, New York, USA, pp. 1-2, 2014. [Online] Available: `http://dx.doi.org/10.1145/2660129.2660162`

- **Daniel Posch**, Hermann Hellwagner, Peter Schartner, On-Demand Video Streaming based on Dynamic Adaptive Encrypted Content Chunks, *In Proceedings of the 8th International Workshop on Secure Network Protocols (NPSec' 13)* (Jun Li, Olaf Maennel, eds.), IEEE, Los Alamitos, CA, USA, pp. 6, 2013. [Online] Available: `http://dx.doi.org/10.1109/ICNP.2013.6733673`

- Barry Crabtree, Tim Stevens, Brahin Allan, Stefan Lederer, **Daniel Posch**, Christopher Müller, Christian Timmerer, Video Adaptation in Limited or Zero Network Coverage, In *CCNxConn 2013 (Priya Mahadevan, ed.)*, PARC, Palo Alto, pp. 1-2, 2013.

Abstract

Real-time entertainment (mainly audio/video streaming) is responsible for the largest traffic share in today's networks. Social and entertainment platforms such as YouTube, Netflix and Facebook provide a tremendous amount of multimedia content to their global customers via the Internet. With the ever growing popularity of these services, the Internet is struggling to suffice the continuously increasing requirements demanded by applications. In particular, the demands go far beyond the intent of the Internet's original design. Architectural and legacy design choices lead to issues, the solutions to which are neither efficient nor elegant. One approach to tackle these challenges is Information-Centric Networking (ICN), a new concept for today's Internet. The idea is to base the network's principal communication model on the most important item, namely the content to be transferred. This novel concept provides significant opportunities to enhance networking. In this thesis we investigate how ICN can be used as an enabler for effective multimedia dissemination. As a first step we analyse the technology's characteristic capabilities and their potential benefits for content distribution in future networks. We develop an analytical model taking account of the main building blocks (network-inherent caching, multi-path forwarding) and compare the obtained upper bound to the current state of ICN considering the scenario of pull-based adaptive multimedia streaming. The results show that there exists a significant gap between the promised and the realized performance, largely caused by ineffective Interest forwarding strategies. Therefore, we design and implement a novel probability-based forwarding strategy named Stochastic Adaptive Forwarding (SAF), which provides effective multi-path forwarding, identifies unknown cached content replicas and deals with local topology changes without guidance from the routing plane. The results indicate that SAF brings ICN one step closer towards effective content distribution. In particular, we show that it is important to consider context information in the forwarding plane. This includes content characteristics and application demands. SAF is the first strategy that takes account of context information that can be supplied by the network operators. Furthermore, this work provides a framework for a testbed that can be used by researchers to readily deploy an ICN-based testbed. This allows researchers to conduct experiments on physical hardware providing deeper insights on proposed algorithms than network simulations or analytical methods could ever do. We use the testbed to validate our results concerning multimedia delivery in ICN, and conduct network emulations investigating the performance of SAF and its competitors. Furthermore, we compare the results of network emulations to the findings obtained from simulations to assess their validity. Both simulations and emulations show that our SAF approach provides a significant step towards effective multimedia content distribution in ICN.

Kurzfassung

Das Streaming von Audio- und Videoinhalten ist heutzutage für den Großteil des Internetverkehrs verantwortlich. Soziale Netzwerke und Unterhaltungsplattformen wie zum Beispiel YouTube, Netflix und Facebook bieten ihren global verteilten Kunden unvorstellbar umfangreiche Datenmengen multimedialen Inhalts an. Seit der aufkommenden Popularität dieser Services steht die heutige Internetinfrastruktur unter massivem Druck, da die Verteilung solch enormer Datenmengen im Internet ursprünglich nicht vorgesehen war. Architekturbedingte Herausforderungen und ein von Altlasten geprägtes Design führen zu Problemen, deren Lösungen weder effizient noch elegant sind. Ein Ansatz, diese Herausforderungen zu meistern, ist Information-Centric Networking (ICN), ein neuartiges Konzept für die Architektur des zukünftigen Internet. Die Idee dabei ist, das grundlegende Kommunikationsmodell auf das zentrale Element der Kommunikation zu fokussieren, nämlich auf die zu übertragenden Inhalte. Diese grundlegende Änderung ermöglicht umfangreiche Verbesserungen. Diese Arbeit untersucht, wie ICN-Technologie zur Effizienzsteigerung bei der Verteilung von Multimediainhalten hilfreich sein kann. Als ersten Schritt analysieren wir die Möglichkeiten dieser neuwertigen Architektur und deren potenziellen Einfluss auf die Inhaltsverteilung in zukünftigen Netzwerken. Wir entwickeln ein analytisches Modell zur Abschätzung der möglichen Performancesteigerung unter Berücksichtigung der wesentlichen Vorteile (Multi-Path-Forwarding und netzwerk-inhärentes Caching) am Beispiel von pullbasiertem adaptiven Multimedia-Streaming. Die Ergebnisse zeigen, dass es aktuell einen signifikanten Unterschied zwischen der versprochenen Effizienz und den tatsächlich erreichbaren Ergebnissen unter der Verwendung von ICN-Technologie gibt. Die Ursache dafür liegt in der schlechten Performance existierender Forwarding-Strategien. Aus diesem Anlass entwickeln wir Stochastic Adaptive Forwarding (SAF), eine neue wahrscheinlichkeitsbasierte Forwarding-Strategie, die effektives Multi-Path-Forwarding ermöglicht, Datenkopien im Netzwerk aufspüren kann und mit lokalen Veränderungen der Netzwerktopologie ohne Hilfestellung des Routing zurecht kommt. Die Resultate zeigen, dass SAF das ICN-Konzept einen Schritt näher zum gewünschten Ziel der effektiven Datenverteilung in Netzwerken bringen kann. Im Besonderen konnten wir zeigen, dass die Berücksichtigung von Kontextinformation beim Treffen der Forwarding-Entscheidung vorteilhaft ist. SAF ist die erste Strategie, die bereitgestellte Kontextinformation auswerten kann. Des Weiteren wird in dieser Arbeit ein Framework für eine ICN-basierte Testumgebung bereitgestellt. Die dargestellte Testumgebung ermöglicht Forschern zusätzlich zu analytischen Betrachtungen und Simulationen, Emulationen auf physischer Hardware durchzuführen. Wir nutzen solch eine Testumgebung für Experimente bezüglich der Perfomanceeinschätzung von SAF. Darüber hinaus vergleichen wir diese Ergebnisse mit jenen von Simulationen, um deren Korrektheit beurteilen zu können. Sowohl die Netzwerksimulationen als auch die Emulationen zeigen, dass SAF ICN einen Schritt näher zum gewünschten Ziel effizienter Verteilung von Multimediainhalten in zukünftigen Netzwerken führt.

CHAPTER 1

Introduction

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

— Alan Turing, 1912* - 1954†

This chapter motivates the importance of multimedia dissemination in today's and future networks. By looking back to the early days of the Internet, we state essential problems arising from the Internet's legacy architecture as well as from the increasing user and application demands resulting in Internet ossification. Subsequently, we state the research objectives and contributions of this thesis. These cover the topic of effective multimedia dissemination in Information-Centric Networking (ICN), where ICN is a nascent approach for the Future Internet. Finally, this chapter outlines the structure of this thesis.

1.1 Motivation

Let us put ourselves back to the 1960s, to the time when the foundation of today's Internet was laid. Back then, it was inconceivable that streaming multimedia content over packet-switching networks will ever be possible. Since then, the Internet evolved from the idea of sharing scarce resources within a limited geographical area to a service oriented global communication infrastructure. Nowadays, multimedia entertainment platforms such as YouTube and Netflix are no more fiction, but responsible for more than 67.4% of the peak traffic composition in North America for fixed access networks as reported by Sandvine [1]. Similar figures are valid for Asia and Europe, as well as for mobile access networks on these continents. Considering that the monthly global Internet traffic was 59.9 exabytes (10^{18}) in 2014 as reported by Cisco [2], more than 450 exabytes of multimedia content were distributed over the Internet in 2014. Furthermore, Cisco predicts that the monthly global Internet traffic will exceed 167 exabytes in 2019 [2], and as the popularity of multimedia entertainment (mainly audio-/video streaming, see Figure 1.1) is continuously increasing since the last decade [1], the yearly distributed multimedia content will very likely exceed the zettabyte (10^{21}) boundary.

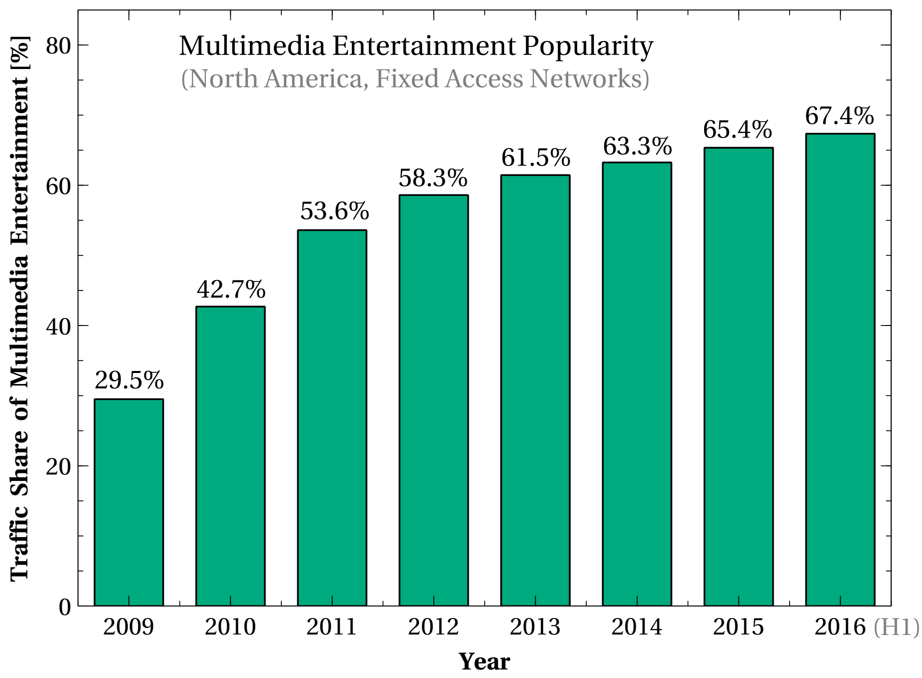


Figure 1.1: The traffic share caused by multimedia entertainment in North America for fixed access networks during peak traffic periods since 2009 [1].

The Internet is a network of networks. It evolved from the world's first experimental packet-switching network, which is denoted as the Advanced Research Projects Agency Network (ARPANET) [3] and was funded by the US Department of Defense. In conjunction with the Internet Protocol (IP) [4], the ARPANET provide the foundation of today's Internet. All Internet-based applications, e.g., the World Wide Web (WWW), electronic mail, and multimedia streaming, rest on IP's host-based communication principle. Actually, this architecture has never been intended to become as widespread as it is nowadays. The continuously growing amount of transmitted data, the trend to equip everyday items with communication capabilities (Internet of Things [5]), and the increasing user expectations push the Internet towards its operational limits [6]. A common example to reinforce this statement is the depletion of the IPv4 address space in 2011 [7]. Moreover, as a general purpose network, the Internet has to support a wide range of applications. It is a large challenge to fulfil the broad spectrum of requirements caused by varying application demands. This situation is further aggravated by the fact that the Internet is used by many applications in a way for which its underlying architecture is disadvantageous, and actually has never been intended for. For example, consider the video sharing platform YouTube. Their operators have to invest tremendous effort (and money) for content dissemination strategies, since today's Internet does not inherently provide the capabilities to distribute (on-demand) multimedia content at a large scale. Supporting measures such as Content Delivery Networks (CDN) [8] are required to tackle this and related challenges. Still, there are a vast number of applications that have different demands requiring various supporting measures and technologies leading to solutions that are neither efficient nor elegant [9].

> *The Internet only just works.* – Handley [9]

As Handley's article points out, today's Internet is working, providing humanity with a global infrastructure. Handley emphasizes that the Internet was never designed for a particular problem. Instead, it provides a general purpose network supporting a wide range of applications. However, he also points out that times are changing and that the Internet suffers from adopting new solutions to fulfil the continuously growing amount and variety of demands. Although the Internet provides most of the functionality required, it becomes harder and harder to adapt to new requirements as legacy design choices restrict the opportunities. The keyword is Internet ossification. It describes the fact that it is practically impossible to alter legacy standards as they are implemented by billions of devices (maybe even in hardware). For instance, the infamous and *long-lasting* transition from IPv4 to IPv6 is a striking example. Eventually, all these issues lead to the situation that researchers and industry investigate new architectural concepts to overcome current restrictions in the Future Internet (FI). The term FI encompasses all ongoing efforts [10–13] that deal with the design of new architectures, communication paradigms, and technologies for the future Internet. With this thesis we want to support the FI community by identifying challenges and proposing promising approaches that enable efficient multimedia content distribution in tomorrow's networks.

As previously indicated, real-time entertainment is the predominant traffic source in today's networks. Therefore, this thesis focuses in particular on multimedia dissemination in Information-Centric Networking (ICN) [14]. ICN is a nascent candidate for the Future Internet, especially as it got a lot of momentum from researchers and industry in recent years. In particular, we *investigate pull-based adaptive multimedia streaming* in *Named Data Networking* (NDN) [15], where NDN is a concrete implementation of the ICN concept. Information-centric networks promise effective content dissemination due to mechanisms such as *network-inherent caching*, *multi-path forwarding*, and *context- and content-aware dissemination strategies*. These mechanisms represent core components of information-centric networks that are aggregated in a novel strategy layer. This thesis investigates the interplay of the strategy layer with pull-based adaptive streaming, identifies performance gaps, and presents enhancements to ICN/NDN to further *improve multimedia dissemination* in these future networks.

1.2 Research Objectives and Contributions

The overall research objective of this thesis is to bring the concept of ICN one step closer to an effective Internet architecture for the dissemination of multimedia content. Therefore, this thesis analyses in particular the approach of Named Data Networking and its promises, and identifies open challenges. Furthermore, this dissertation tries to tackle some of them, especially in the research field of adaptive Interest forwarding strategies. The following list includes the main research objectives of this thesis:

(1) to investigate the main architectural concepts of ICN/NDN that are proposed as enablers for effective content delivery in the Future Internet;

(2) to provide an analytical assessment for the possible performance of content delivery in ICN/NDN networks;

(3) to match the state-of-the-art performance of ICN/NDN with the analytical assessment identifying a potential gap revealing yet unsolved/unknown challenges;

(4) to analyse strengths and weaknesses of the strategy layer in ICN/NDN concerning content distribution efficiency. In particular this includes the mechanisms of adaptive multi-path forwarding and in-network caching;

(5) to identify the role of content and context awareness for effective multimedia distribution concerning ICN's/NDN's strategy layer by investigating a multimedia delivery scenario encompassing applications with various demands;

(6) to assess and compare the performance of the relevant concepts in the strategy layer in both simulated and physical environments.

This thesis has made several contributions to the field of ICN, especially concerning multimedia dissemination in these future networks. More than 10 refereed scientific research papers have been published (cf. Page VII) that provide the contributions of this thesis. The following list includes the main research contributions of this thesis:

(1) an analytical model that provides the upper bound for pull-based multimedia delivery in NDN considering multi-path transport and in-network caching;

(2) a detailed investigation of pull-based adaptive streaming in NDN;

(3) the identification of open challenges in the strategy layer, especially concerning state-of-the-art approaches for multi-path forwarding;

(4) the design and implementation of a novel probability-based forwarding strategy that performs significantly better in multimedia and classical data delivery scenarios than state-of-the-art competitors;

(5) the first steps towards a context-aware forwarding plane, by providing a strategy that is able to consider user-supplied context information for packet forwarding;

(6) the design and implementation of a low-cost NDN-based testbed that can be readily set up and used to investigate ICN technologies under realistic settings;

(7) a detailed performance investigation and comparison of existing forwarding strategies in simulated and physical deployment scenarios.

1.3 Thesis Structure

This thesis is structured into seven chapters pursuing the objective of effective multimedia dissemination in ICN, as has been motivated in Chapter 1. Chapter 2 provides the necessary technical background and related work for this dissertation. The chapter introduces the basic principles of ICN, with on focus on Named Data Networking, which will be used as ICN representative for this work. Furthermore, Chapter 2 provides details about content dissemination and its classification in ICN. As this thesis focuses on the topic of pull-based adaptive streaming, we further introduce the principles of adaptive streaming and illustrate existing approaches in the ICN domain. Then, Chapter 3 investigates the promises of ICN that are put forward by the research community as Future Internet technology enabling effective content dissemination. First, the chapter illustrates the promises of network-inherent caching, adaptive multi-path forwarding, and context- and content-aware data delivery by investigating simplified use cases. Then, we model a more complex scenario trying to obtain the performance potential. The results show that ICN-based content delivery is superior to classical IP-based content delivery. However, we also identify that there is a significant gap between the potential performance (which is derived from analytical modelling) and the practically obtainable one. Therefore, Chapter 4 proposes a new adaptive Interest forwarding strategy, as we conclude that existing strategies are not able to fully utilize the opportunities provided by ICN. The strategy is designed to exploit all options and is based on a probabilistic approach. Chapter 5 discusses the importance of context awareness in the forwarding plane and investigates its influence on the content delivery performance with respect to different (multimedia) applications. Moreover, this chapter presents an extension to the developed strategy in Chapter 4, which uses the strategy's design to enable the consideration of context information in the forwarding plane. Chapter 6 provides a framework for an ICN testbed on so-called Banana Pi Routers, which are low-cost single board computers. We use the proposed testbed to validate our previously obtained results under real-world constraints. This also includes further evaluations considering new perspectives that are not accessible in simulated environments (e.g., power consumption, CPU usage, etc.). Finally, Chapter 7 summarizes our findings regarding effective multimedia dissemination in ICN and concludes this thesis.

CHAPTER

2 Technical Background and Related Work

"If I have seen further, it is by standing on the shoulders of giants."

— Bernard of Chartres, 1124†

This chapter discusses the necessary technical background and the related work relevant for this thesis. The reader is provided with a basic understanding and terminology of the relevant subjects. Specific approaches, algorithms, protocols, etc. relevant for the individual chapters are presented within the appropriate chapter's context. Section 2.1 introduces the concept of Information-Centric Networking (ICN), which proposes a paradigm shift from IP's host-based communication model to a content-oriented one. The paradigm shift is motivated by presenting challenges and inconveniences that users encounter in IP-based networks, that can be overcome by introducing the principles of ICN at the network layer. Since this thesis focuses on the specific ICN approach called Named Data Networking (NDN), Section 2.2 provides a detailed overview of the NDN architecture. Section 2.3 introduces related work regarding multimedia dissemination in ICN with a focus of pull-based multimedia streaming approaches in ICN. Especially, the principles of adaptive streaming will be discussed with a focus on MPEG-DASH. We argue that the concept of MPEG-DASH can be readily adopted to the ICN domain.

2.1 Information-Centric Networking

This section provides the basic idea of Information-Centric Networking (ICN) to the reader by: *i*) briefly motivating the paradigm shift from IP's host-based communication to a content-oriented one (cf. Subsection 2.1.1); *ii*) illustrating the ICN communication model (cf. Subsection 2.1.2); *iii*) discussing concrete ICN approaches and summarizing their key concepts and differences (cf. Subsection 2.1.3).

2.1.1 Motivation for a Content-Oriented Paradigm

Users appreciate the Internet for the possibility to share and fetch data easily. However, due to today's host-based communication model implemented by IP, there are several inconveniences. In the following we briefly discuss a (small) subset of them, because they provide a good motivation for the proposed paradigm shift foreseen by ICN.

Using IP, the first obstacle for fetching data or using a service is that consumers need to be aware of the endpoint (IP address) that provides the desired object or service. For instance, there is no inherent possibility for a consumer to express its demand for retrieving the latest episode of a prominent television series without additional search and resolution mechanisms that provide/resolve the corresponding endpoints (e.g., search engines, Domain Name System). Furthermore, the mapping of services to endpoints strongly impedes mobility, as services are addressed by endpoints that may change in some scenarios (e.g., a VoIP client that moves within a city may switch among different access networks frequently). Moreover, mobile devices that often employ multiple interfaces are restricted to use only a single one when relying on classical IP-based communication.

Another issue is the efficient usage of available network resources. For instance, if a consumer requests the same content that has been previously requested by a nearby host, IP is not able to fulfil the request by returning the close-by copy. Using IP, the network is ignorant about nearby replicas. Therefore, requests will be forwarded to the possible far away content origin wasting bandwidth resources and delaying service startup. This unfavourable behaviour is again due to IP's host-based communication principle that *i*) defines the only valid content source in the destination address field in the IP packet's header; and *ii*) provides no information about the content/service that has been actually requested preventing transparent network caching.

The last issue that we want to mention that motivates a paradigm shift in networking is the lack of inherent security in IP. Neither an IP packet's integrity nor its authenticity can be verified by its receiver or by any forwarding network entity. Again, this prohibits the idea of network-inherent caching (as consumers can not distinguish legitimate copies from manipulated data), but even worse, this also results in vulnerabilities to a broad range of attacks (e.g., (Distributed) Denial of Service attacks) since attackers may easily forge legitimate-looking fake packets.

Currently, the majority of these challenges can be counteracted only at the application or transport layers. However, the displacement of these issues on higher layers often results in solutions which are neither efficient nor elegant. For instance, recall the issue of efficient resource consumption that could be addressed by returning nearby content replicas. In IP-based networks, this challenge is tackled on the application layer by the usage of Content Delivery Networks (CDNs), which place (cache) popular content near the network edge on dedicated servers. Nevertheless, the provision and maintenance of CDNs is costly and requires a lot of effort, which could be greatly reduced if already the network itself provided some inherent caching mechanism. To overcome the previously listed challenges, ICN foresees a radical paradigm shift. The idea is to introduce a new architecture that focuses on the most important elements of today's computer networks, namely the content to be transferred. Introducing a content-oriented communication model that eliminates the assignment of endpoints to services and contents, provides significant opportunities to enhance networking. Furthermore, as will be discussed in the following subsections, ICN incorporates security within the content (transmitted packets) enabling authenticity and integrity verification already at the network layer.

2.1.2 The ICN-based Communication Model

The basic communication model of an ICN is illustrated in Figure 2.1 [14, 16]. In ICN every object that is shared via the network is given a name. For instance, in the example network illustrated in Figure 2.1 we have four named objects $\{A, B, C, D\}$. The aggregate of all objects is denoted as the content catalogue. Assuming that a user is interested in the object named B he/she emits a message representing his/her interest. Then the network is responsible to retrieve a *trustable* copy of B. This copy can either be retrieved from the content origin or from a nearby host maintaining a replica, as shown in Figure 2.1.

Note that the communication can be issued over untrusted connections and even the host providing the replica can be untrusted. But how is it possible that the receiver can be sure that a *valid* copy of the requested object has been obtained? The answer is content-based security. ICN introduces a new security paradigm to networking that is content-oriented rather than connection-oriented. The basic idea of content-based security [17–19] is illustrated in Figure 2.2 and rests on the idea of self-authenticating data objects ensuring integrity, authenticity and eventually the trust that is placed in received replicas. Content publishers can achieve this by creating an inextricable linking between the content, its given name and the publisher's identity, e.g., by using a cryptographic hash function. Then the

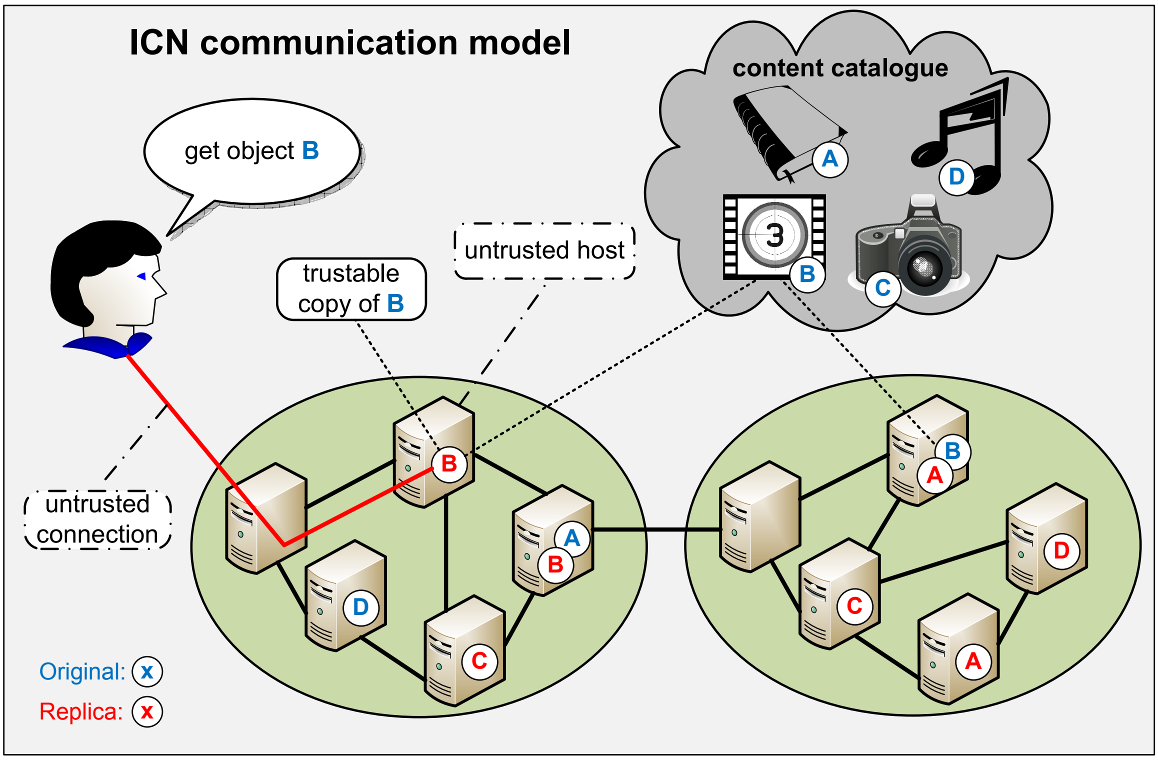


Figure 2.1: The communication model of an Information-Centric Network[14]. The content catalogue contains a set of named objects. A user requests objects by name and the network is responsible to return a *trustable* copy of the object.

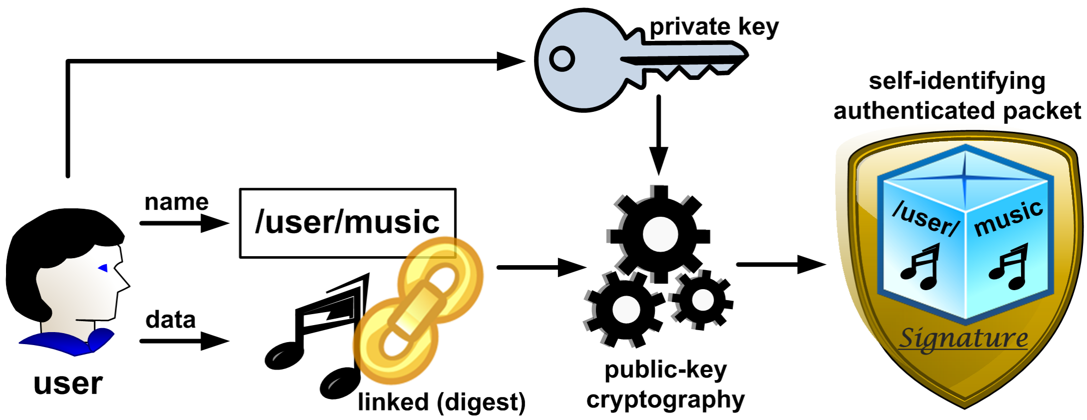


Figure 2.2: Content-based security foresees to protect the content itself by creating an inextricable linking between content name, data and its publisher forming a self-identifying authenticated packet [17–19].

linking is digitally signed using public-key cryptography [20]. The principle of content-based security provides ICN significant advantages over IP-based networks, e.g., it is one key enabler for network-inherent caching enhancing content dissemination.

2.1.3 ICN Implementation Approaches

As can be seen from the Figure 2.1, the general communication model of an ICN is actually quite simple. However, there are plenty of implementation and research challenges (efficient naming, routing, caching, object resolution, etc.). In research, a large set of challenges tend to lead to an even larger set of solutions, and therefore there is a decent amount of concrete implementations that define their own vision of an ICN (see below). Most approaches are results of extensive research projects that have been (or still are) carried out within the context of Future Internet research. We shortly outline the fundamentals of the most important approaches in chronological order. Therefore, this list also provides a historical review indicating the advancements of the ICN paradigm over time:

- **TRIAD:** The Translating Relaying Internet Architecture integrating Active Directories as proposed by Cheriton et al. [21, 22] from Stanford can be seen as the pioneering work on the ICN topic. TRIAD introduces a content layer that identifies content via Uniform Resource Locators (URLs). Content can be read or written via so-called *network pointers* realized via HTTP/TCP connections. The layer is implemented by content routers that are responsible for directing requests towards the content servers or content caches. Cheriton et al. already proposed the idea of *content transformers*, entities that transform content in response to characteristics of the requesting clients and their network connections.

- **DONA:** The Data-Oriented (and Beyond) Network Architecture as proposed by Koponen et al. [23] provides a new method of how Internet names are structured and resolved. The authors indicate that the concept of DONA is based on TRIAD [21], as well as on the Host Identity Protocol [24] and the idea of self-certifying pathnames [25]. The aim is to replace today's Domain Name System (DNS). DNS names are substituted with flat, self-certifying names, while DNS resolution is replaced by a name-based anycast primitive, which is located above the IP layer. DONA uses so-called *resolution handlers* that provide name resolution via the two basic primitives FIND and REGISTER. The REGISTER command is used to set up the necessary state in the *resolution handlers*, which is then used to look up the requested content via the FIND primitive. DONA considers multi-homing allowing FINDs to look up multiple paths, which can be used for data transfer simultaneously.

- **PSIRP/PURSUIT:** The Publish-Subscribe Internet Routing Paradigm as proposed by Lagutin et al. [26] decouples the sender and receiver of a data object in time and space by introducing a persistent, immutable association between an identifier and the corresponding data denoted as the *publication*. The PSIRP architecture consist of four distinct parts: *i*) **rendezvous**, a middleman between publishers and subscribers matching data sources hosting a publication with subscribers; *ii*) **topology**, a manager for the physical network, configures internal and external routes reacting to error conditions and considering load balancing; *iii*) **routing**, an administrator maintaining the delivery tree for each publication and caches popular content at branching points; *iv*) **forwarding**, a forwarder delivering the actual publications to the subscribers along the efficient delivery tree. Publish-Subscribe Internet Technologies (PURSUIT) [27] is the follow up EU FP7 projected of PSIRP with the aim to refine the PSIRP architecture. There is an open source implementation of PSIRP/PURSUIT denoted as *Blackhaw* available at [28].

- **NetInf:** The foundation of the Network of Information as proposed by Dannewitz et al. [29, 30] was laid in the EU FP7 project *4WARD* [31] and is continued in the Scalable and Adaptive Internet Solutions (SAIL) project [32]. The objective of NetInf is to introduce information as the first-class citizen in today's networks. So-called *Information Objects* represent information at a high level, e.g., a picture of the Eiffel Tower. *Data Objects* are bit-level specific instantiations of Information Objects, e.g., the picture of the Eiffel Tower hosted on the Wikipedia article about Paris. Location and data is separated using a flat but structured name space, and name resolution is implemented via Distributed Hash Tables. There is an open source implementation of NetInf denoted as *OpenNetInf* available at [33]. The software package includes a plugin for the Web browser Mozilla Firefox that provides browser access for NetInf-enabled Websites, and an extension to the email client Mozilla Thunderbird that enables the representation of peoples' email addresses as Information Objects.

- **CCN:** Content-Centric Networking as proposed by Jacobson et al. [34] aims at replacing IP as universal communication layer of today's protocol suite. So-called named content chunks are introduced as the universal communication principle. Content chunks are named based on tree-like structures enabling implicit addressing of chunks. Furthermore, CCN foresees two additional layers, a dedicated security layer that is based on content-based security methods, and a strategy layer that is responsible to optimize data transport, e.g., by means of path selection in multi-path scenarios. Xerox

PARC, the industrial maintainer of the CCN project, provides an implementation of their prototype software denoted as CCNx [35].

- **NDN:** For the approach Named Data Networking as proposed by Zhang et al. [15] we refer to Section 2.2. This approach will be discussed in detail as we take it as ICN representative for this thesis.

For a more detailed distinction among the aforementioned approaches we kindly refer the reader to the following two survey papers [14] and [16], and to the previously indicated references.

2.2 Named Data Networking

Named Data Networking (NDN) [15] is probably the most elaborated ICN approach today. The project was funded in 2010 by the US National Science Foundation and has common roots with the CCN [35] approach. We have chosen NDN as representative of the ICN concept because it provides the most functionality compared to the previously listed approaches in Subsection 2.1.3. The NDN project [36] offers an extensive open source software suite including a simulator [37], a network forwarder [38] and a variety of NDN-related applications [39–41] and libraries [42, 43] that can be used for experimentation. In this section we first introduce the fundamentals of NDN's communication architecture including a discussion of the two basic packet types *Interest* and *Data* (cf. Subsection 2.2.1). Second, we discuss NDN's hierarchical name space that provides efficient content naming and also illustrate the employed encoding (cf. Subsection 2.2.2). Then, we present the basic building blocks of a classical NDN node (cf. Subsection 2.2.3), and finally we illustrate the principles of packet processing (forwarding) and routing as performed by the individual nodes (cf. Subsection 2.2.4).

2.2.1 Communication Principles

NDN foresees to replace IP as the universal communication layer of today's Internet architecture with named chunks of data as illustrated in Figure 2.3. This changes the semantic of data communication from *delivering packets to endpoints* to *fetch data identified by a given name*. The transport of the chunks is guided by the newly introduced *strategy layer* that includes routing, forwarding, and in-network caching considerations. Inherent network

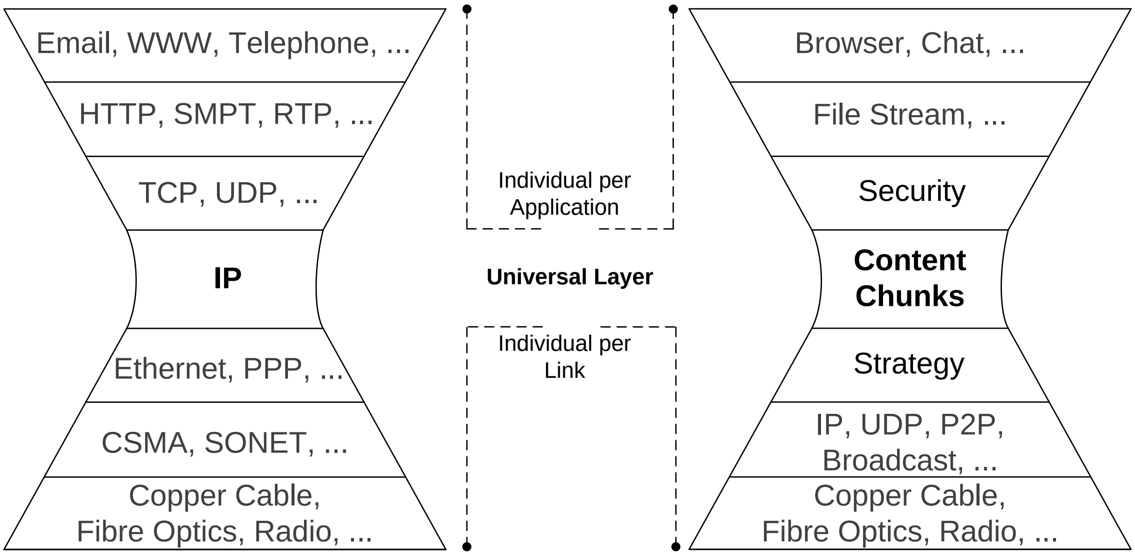


Figure 2.3: NDN foresees to replace IP as the universal communication layer using named chunks of data. The transport of the chunks is guided by the newly introduced strategy layer, while the security layer ensures data integrity and authenticity via content-based security [15].

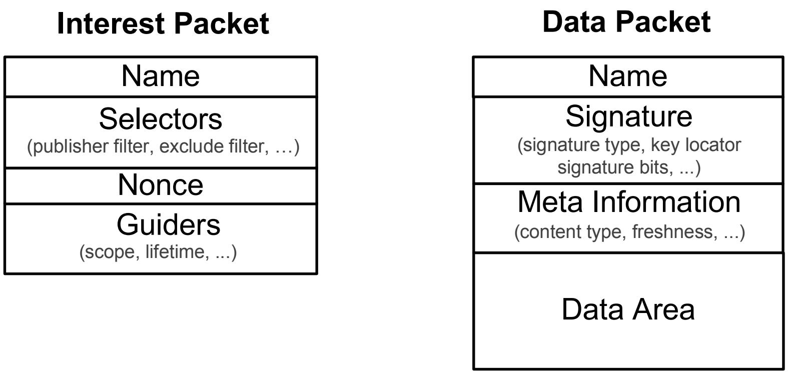


Figure 2.4: High-level representations of the two basic NDN packet types. An Interest may be satisfied by a Data packet if the Interest's and the Data's name match, and the selectors are consistent (e.g., publisher filter matches the publisher) [15].

caching is supported by the novel *security layer* that implements a contend-based security model ensuring integrity and authenticity of the transmitted content chunks.

In NDN communication is always receiver driven. The receiver emits a so-called *Interest* message that identifies the requested data chunk by name. Figure 2.4 illustrates the two packet types (Interest and Data) that are specified for information exchange. The ***Interest*** packet consists of: *i*) **Name**, identifying the requested data chunk; *ii*) **Selectors**, enabling the receiver to specify certain filters, e.g., specify a concrete content provider; *iii*) **Nonce**, a preferably unique bit pattern that is issued to identify looping/duplicate Interests; and *iv*) **Guiders**, specific constraints for an Interest, e.g., that it must not leave the local subnetwork. The ***Data*** packet consist of: *i*) **Name**, identifying the content of the packet; *ii*) **Digital Signature**, coupling name, data and their publisher inextricably; *iii*) **Data Area**, the raw bits of the content; and *iv*) **Meta Information** about the Data packet, e.g., freshness period, the time how long the data should be kept in the cache before it is marked as stale and should be evicted. For a more detailed discussion of all available packet fields we kindly refer the reader to [15] and [44].

2.2.2 Hierarchical Namespace and Packet Encoding

Although names play a very important role in NDN, they are in general opaque to the network. Names consist of an arbitrary number of components that are separated by a slash ('/'), similar to today's URLs. For example, `/icn.itec.aau.at/dash/demo.mpg/5000kbit/0` is a valid NDN name. An Interest packet matches a Data packet if the Interest's name is a prefix of the Data's name. NDN assumes a hierarchically structured name space that allows to model the context and the relationship between individual chunks in some logical, tree-like structure. Names are human readable and are divided into three parts as shown in Figure 2.5. The first part is the global-routable name (or prefix) that is used for longest prefix match routing, similar as in today's IP networks. The second part is denoted as the organizational name that allows the publisher to organize the data chunks

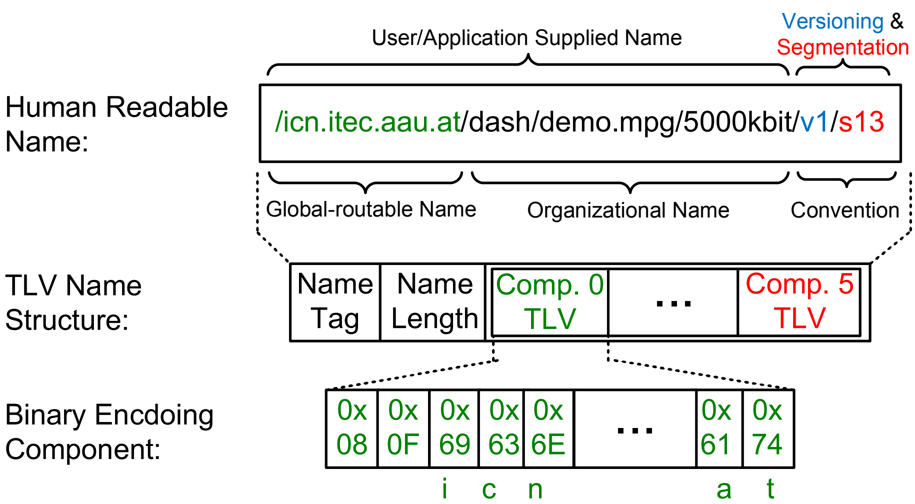


Figure 2.5: An NDN name is human readable and is divided into three parts, a global-routable name, an organizational name and additional versioning and segmentation information. For transmission it is translated to a binary TLV-based encoding [44].

in an appropriate way considering application demands. The last part of the name includes versioning and segmentation information. Segmentation is necessary since large files, e.g., a movie, need to be segmented into smaller chunks for network transmission. The versioning enables content providers to update their content, e.g., a newspaper article may be updated once additional details become apparent. For data transmission and storage purposes, the human readable form of the name is converted to a binary Tag-Length-Value-based (TLV) encoding as exemplarily indicated in Figure 2.5. TLV-based encoding is not solely used for the name, but also for the encoding of entire Interest and Data packets providing flexibility in adding new tags to the current NDN packet format specification [44].

2.2.3 Building Blocks of an NDN Node

The architecture of a typical NDN node is illustrated in Figure 2.6. Each operative node maintains three data structures that are used to perform packet forwarding via the so-called faces. Faces are a general concept of interfaces including (wireless) network interface controllers and ports to applications. The necessary data structures are:

- **CS:** The Content Store acts as an in-network storage for transmitted content chunks. It provides data replicas to Interests that ask for the same content.
- **PIT:** The Pending Interest Table keeps track of the forwarded but not yet satisfied Interests. These entries provide a breadcrumb-like return path for the Data packets. This is necessary as neither Data nor Interest packets carry location information. A Data packet follows the reverse path (breadcrumbs) of the corresponding Interest(s) to reach the requesting consumer(s).
- **FIB:** The Forwarding Information Base essentially maintains the routing information. For each known global-routable prefix, the FIB stores a list of possible outgoing faces that are likely to provide data for the given prefix. The forwarding strategy in the strategy layer decides which of the indicated face(s) are used for Interest forwarding.

2.2.4 Processing of Interest and Data Packets

The basic process of Interest and Data packet forwarding in NDN is sketched in Figure 2.7. In principle, it comprises lookup operations in the previously listed data structures (CS, PIT and FIB). The data structures are typically indexed by a common index with the purpose

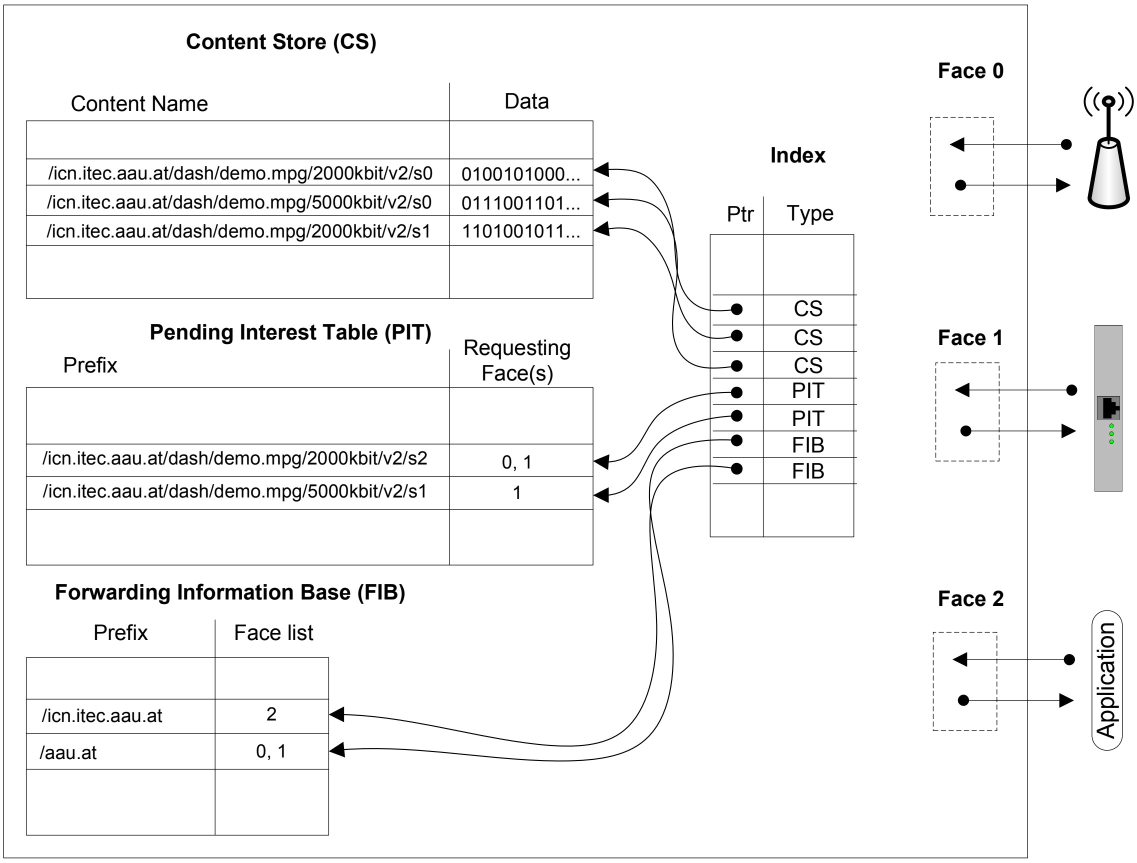


Figure 2.6: The architecture of a typical NDN node. The node maintains a list of faces used for forwarding and receiving packets. The common index data structures CS, PIT and FIB provide the necessary information for packet forwarding [15].

to simplify the implementation of the packet forwarding. Therefore, the index is organized in such a way that CS matches are always preferred over PIT matches, and PIT matches are always preferred over FIB matches. Therefore, the processing of an incoming Interest packet may result in the following situations:

- **CS match:** This is the optimal case for a node. The incoming Interest matches an already stored Data packet in the CS. The node can immediately satisfy the Interest by returning a replica of the cached object. Therefore, the replica is simply emitted on the Interest's incoming face and no further actions are required.

- **PIT match:** A PIT match indicates that a previously received Interest has already requested the same content and still waits to be satisfied. The node checks whether the Interest's incoming face is already stored in the list of the requesting faces (cf. Figure 2.6). If this is not the case, the node adds the face to the list, which will trigger

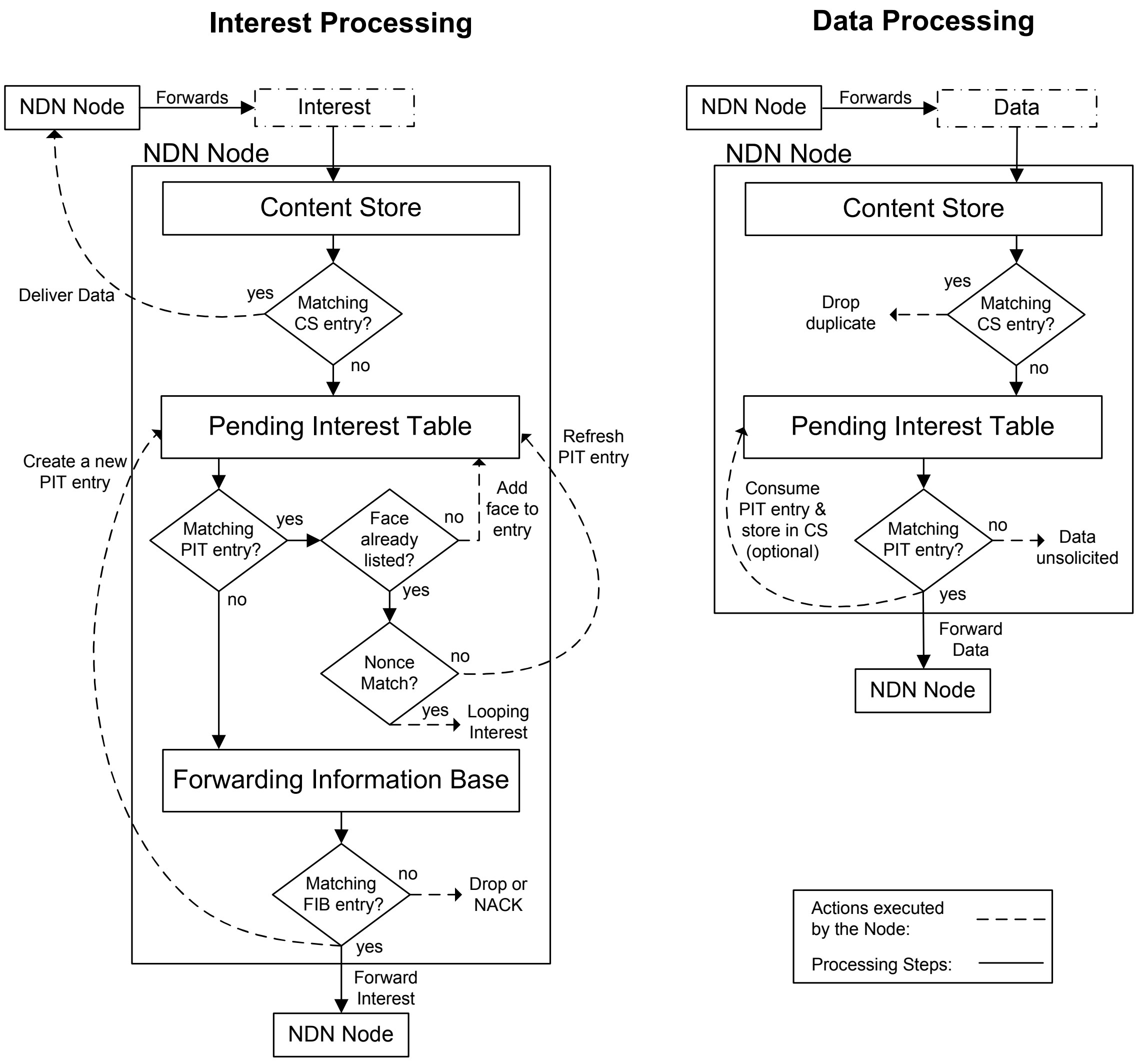


Figure 2.7: Interest and Data packet processing by an NDN node [15] using the basic building blocks CS, PIT and FIB (cf. Figure 2.6).

the transmission of a content replica once the corresponding Data packet is received. However, if the Interest's incoming face is already listed as one of the requesting faces, the node compares the Interest's Nonce field to all previously seen Nonces for this PIT entry. If a bit-exact match is detected, the incoming Interest is looping and must be discarded. Otherwise, if the Nonce is unknown, the inquiring node is still waiting for the Data packet and tries to prevent the current node from discarding the PIT entry due to a time out. The node should therefore refresh the PIT entry to delay the timeout, or it may reply with a Negative Acknowledgement (NACK) message signalling that waiting for a Data packet is in vain as it will eventually discard the pending Interest.

- **FIB match:** A FIB match indicates that the node knows possible outgoing faces that can provide Data given the Interest's prefix. The Node forwards the Interest on one or multiple faces supplied by the FIB and creates a new PIT entry. Which face(s) are selected is determined by the forwarding strategy in the strategy layer.
- **No match:** If there is neither a CS, PIT nor FIB match, the node is unable to satisfy this request and therefore discards the Interest; optionally indicating this by the transmission of a NACK message.

Processing an incoming Data packet may result in the following situations:

- **CS match:** A CS match indicates that the object has already been received once. The node discards the incoming Data packet as it is only a replica of an existing one.
- **PIT match:** A PIT match is the regular case for a Data packet. The node has requested this packet to satisfy a PIT entry. The packet is forwarded to all faces that are listed as requesting concerning the matching PIT entry, and optionally a replica is stored in the CS.
- **No match:** If the Data packet neither matches an entry in the CS nor in the PIT, it has been received unintentionally. The unsolicited packet is discarded by the node.

This section provided a detailed overview of NDN's communication principles. The interested reader may find additional information regarding forwarding [45–47], routing [48–50], security [18, 51–54] and NDN fundamentals [15, 36, 55] at the indicated references.

2.3 Multimedia Dissemination in ICN

This section gives an overview of content dissemination in ICN. First, Subsection 2.3.1 discusses a potential classification for content and its dissemination in the ICN domain. Then, Subsection 2.3.2 introduces the differences between push- and pull-based content delivery and emphasizes analogies to the previously discussed classification. As this thesis focuses on pull-based adaptive streaming in ICN, Subsection 2.3.3 introduces the principles of adaptive streaming and Subsection 2.3.4 presents MPEG-DASH as a concrete streaming approach. Finally, we discuss two existing NDN-specific multimedia dissemination approaches (NDNLive and NDNTube) in Subsection 2.3.5, and adaptive streaming approaches based on transcoding and scalable content encodings in Subsection 2.3.6.

2.3.1 Classifying the Dissemination of Content

ICN foresees efficient information dissemination in global networks. However, the content that represents the information is available in diverse forms. Information can be available in text, as a picture or as a continuous audio-/visual media stream. Furthermore, not only the kind of the content may differ, but also the manner of consumption. For instance, assume we have some audio data that is consumed by a user. If this data represents a piece of music, the content is usually consumed on-demand via streaming. In this case the user may fetch data in advance and cache it in a local buffer. This buffer can be used to deal with transmission issues such as packet loss or congestion giving the playback software some chance to react, (e.g., issue a retransmission) while preserving the feeling of a continuous media consumption for the end user. However, if the consumed audio data is part of a teleconferencing session, there is no chance of pre-fetching some data (it simply does not exist yet), and issuing retransmissions is not applicable because teleconferencing is a real-time application that discards late packets.

Considering these facts, Tsilopoulos and Xylomenos [56] propose a classification for content dissemination in ICN according to the applications' traffic type. Information is distinguished into *channels* and *documents*, where channels represent pieces of information that are loss tolerant, while documents represent pieces of information that must be transferred reliably. Tsilopoulos and Xylomenos emphasize that obtaining documents over an unreliable information delivery service requires some kind of error control, such as the TCP retransmission scheme [57]. In ICN this should be achieved by dividing large documents into named chunks, and if a packet is lost the receiver shall re-request the lost packet(s). In contrast to the receiver-driven approach for documents, Tsilopoulos and Xylomenos foresee a subscription-based communication model for channels since requesting each packet from a channel would be inefficient. Users shall express their interest on a channel by subscribing and the network shall forward the corresponding packets to the user, until the user revokes the channel subscription.

The second dimension that is relevant for content dissemination in ICN are timing constraints. Tsilopoulos and Xylomenos [56] differentiate between real-time transmissions (information that is transmitted/pushed in the moment it is generated) and on-demand transmissions (transmission of archived content). Real-time information includes both continuous media (e.g., live TV) and real-time notifications (e.g., Twitter updates). However, there is a fundamental difference between them. Continuous media is loss tolerant (channel) while real-time notifications such as emergency alerts (documents) require reliable transmission. Tsilopoulos and Xylomenos emphasize that there is an implicit synchronization for users receiving real-time information, which is not true for users requesting on-demand content. The classification proposes that there are three distinct traffic types that are relevant for information dissemination in ICN [56]: *i*) **channels**, *ii*) **on-demand documents,** *iii*) **real-time documents** as illustrated in Table 2.1. This thesis specifically focuses on on-demand pull-based adaptive streaming of continuous media, which belongs to the class of on-demand documents.

<table>
<tr><th></th><th>Channels</th><th>Documents</th></tr>
<tr><td>Real-time</td><td rowspan="2">Live TV, web radio, Voice over IP, teleconferencing, Skype, Twitch</td><td>Twitter updates, online gaming, chat rooms, emergency alerts</td></tr>
<tr><td>On-demand</td><td>File transfer, email, YouTube, Netflix, Amazon Video</td></tr>
</table>

Table 2.1: Classification of information dissemination types according to Tsilopoulos and Xylomenos [56] classifying application traffic either as channels, real-time documents or on-demand documents.

2.3.2 Principles of Push- and Pull-based Streaming

Interestingly, the specification of the terms *channels* and *documents* in [56] is conceptually very similar to what is classified in today's IP-based networks as push- or pulled-based streaming. Push-based streaming denotes a communication model where the server is actively transmitting (pushing) data towards the receiver(s). Probably the most common push-based streaming approach is the Real-time Transport Protocol (RTP) [58] on top of the User Datagram Protocol (UDP) [59]. Since with pushed-based streaming the server just pushes packets towards the client, some feedback must be provided indicating whether the packets have been received successfully by the client. Therefore, the Real-time Control Protocol (RTCP) [60] was introduced, which provides a control stream associated with the delivered data stream. With RTCP the client is able to provide feedback to the sender and if necessary the sender may adapt the transmitted data rate. Today RTP/RTCP is still used in applications such as Voice over IP and video conferencing.

Pull-based streaming instead matches the concept of a *document*. In this case the client/receiver requests each data object separately. The receiver is in full control of the request rate and may adapt it according to its needs. This makes a control protocol such as

RTCP for RTP obsolete. Nowadays, pull-based streaming is extremely popular for fetching on-demand content and data transport in peer-to-peer networks. In general whether to rely on push- or pull-based streaming is a matter of the application as correctly highlighted in [56]. As indicated before, this thesis focuses on pull-based content delivery, which brings us to the next subsection on adaptive streaming.

2.3.3 Dynamic Adaptive Streaming

Dynamic Adaptive Streaming (DAS) describes the idea to provide multimedia data in different representations (qualities) to the end users. Considering various constraints, e.g., available bandwidth capacities and device resources, the best fitting media stream is delivered to the client. Figure 2.8 illustrates three possible adaptation spaces for a digital video stream: *i*) **Spatial** domain, refers to the resolution (number of pixels) of the video frames; *ii*) **Temporal** domain, refers to the number of video frames per second; and *iii*) **Quality** domain, refers to quality of the video with respect to an objective measure, e.g., Structural Similarity [61]. A representation is therefore an encoding with a fixed set of constraints for the spatial, temporal and quality domains.

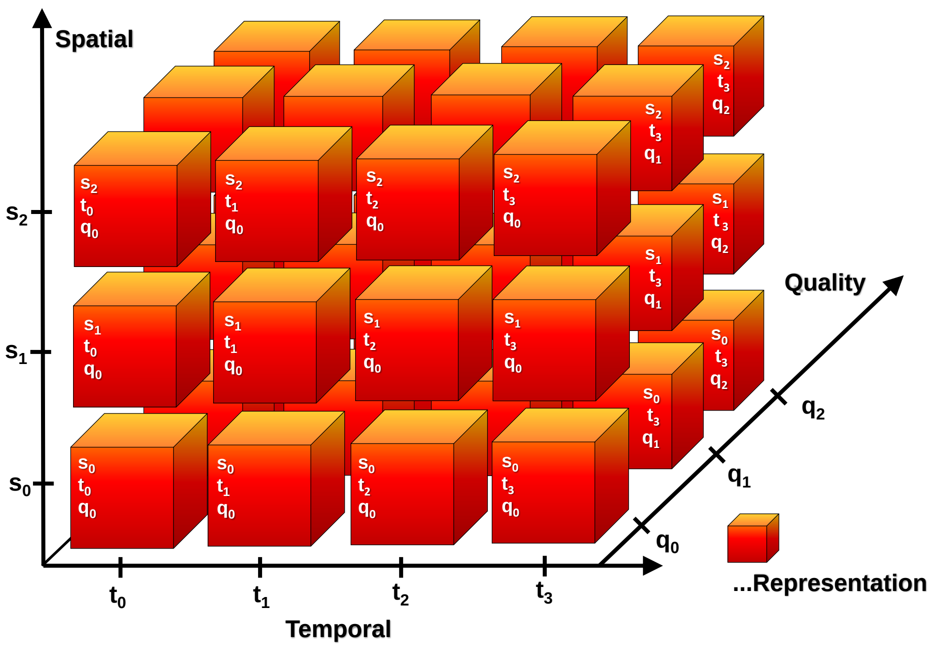


Figure 2.8: The possible adaptation space for a digital video includes changes in the spatial, temporal and quality domains.

Today adaptive streaming is state of the art for multimedia distribution systems, especially, since they have to handle device inhomogeneity (smart phones, tablets, large TV screens) and diverse transmission technologies (2/3/4/5G, 802.11a/b/g/n, DSL, etc.) that end users may use. In general adaptive streaming can be used with both, push- and pull-based communication models. With the increasing popularity of pull-based streaming during the recent decade, most practical approaches of adaptive streaming are implemented based on pull-based streaming. Several commercial solutions are currently available including Apple HTTP Adaptive Streaming [62], Microsoft Smooth Streaming [63] and Adobe Dynamic Streaming for Flash [64]. However, for this thesis we focus on MPEG's Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [65, 66], because it is the predominant standard for on-demand multimedia delivery nowadays. MPEG-DASH will be discussed in more detail in the next section, also providing information about its first steps in the ICN/NDN approach.

2.3.4 MPEG-DASH and First Steps Towards its Use in ICN

MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH ISO/IEC 23009-1) has been first ratified 2012 and has been recently revised in 2014 [67]. The objective of MPEG-DASH is to provide a standardized way of adaptive streaming over HTTP. The fundamental parts of MPEG-DASH are illustrated in Figure 2.9, where the red parts of the figure are covered by the standard and the green parts are left open for (commercial) competition. In MPEG-DASH an ordinary HTTP server provides multimedia content encoded in various representations (cf. Figure 2.8). Each representation consists of a set of so-called segments. A segment is a short part of an audio/video stream that can be decoded and displayed independently. The duration of a segment is typically in the range of 2 to 10 seconds. Usually, the audio and the video streams are separated, so that they can be be chosen independently. This readily enables multi-language support for multimedia streams.

The representations available for download are listed in the so-called *Media Presentation Description* (MPD), an XML-based document providing detailed information on the various representations (bitrate, spatial resolution, audio tracks, etc.). Basically, there are two versions of the MPD. The first one is a *static* approach, where for each representation the full catalogue of available segments is listed. The second version is *template-based.* Templates use place-holders that are substituted (e.g., by increasing numbers) by the consumer application on a regular basis to fetch the next segment with the corresponding

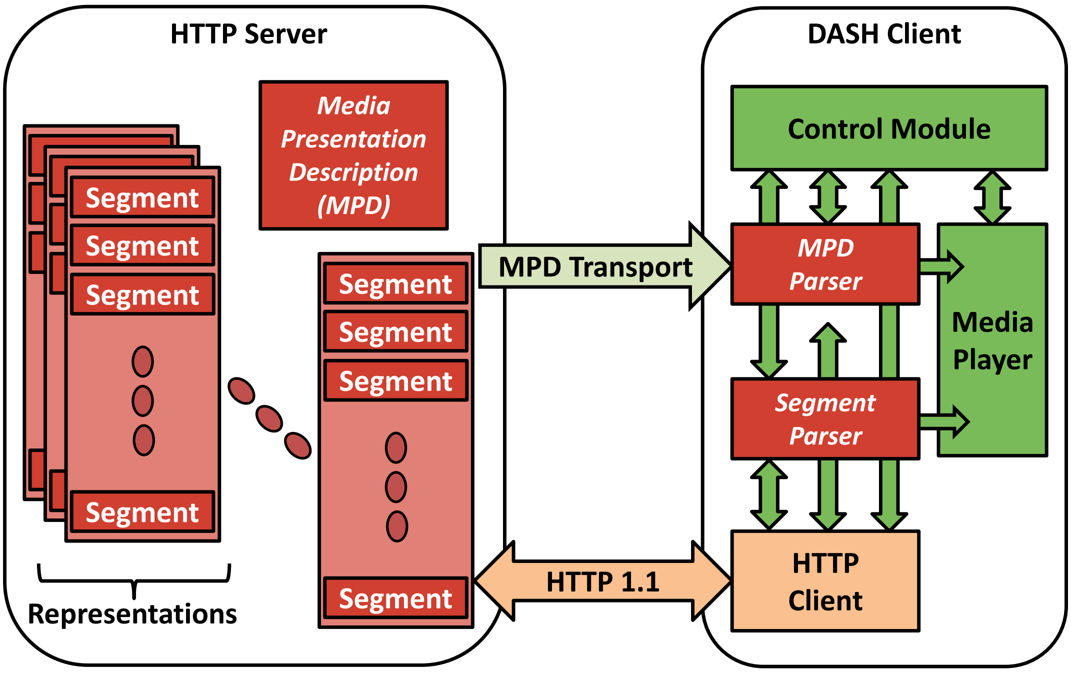


Figure 2.9: The MPEG-DASH architecture. The red parts are standardized, while the green parts are left open for (commercial) competition. [65]

identifier (number). While the first approach can be exclusively used for pre-recorded on-demand content, the second approach can also be used for streaming live content. Since the template-based MPD is more lightweight, it is often preferred. For more information regarding the MPD we kindly refer to the standard [67].

An MPEG-DASH client initiates a streaming session by downloading the MPD and parsing the available representation sets. Then, according to the client's device and resource constraints, a control module determines the best fitting representation for streaming. Once the representation is determined, the client starts downloading the segments from the corresponding representation via an ordinary HTTP 1.1 connection. If certain parameters change during the streaming session, e.g., less bandwidth resources are available, the control module is responsible to react accordingly to ensure a smooth playback. This is usually achieved by changing the representation set, e.g., to a set with lower quality, as illustrated in Figure 2.10. In this example the client starts streaming the lowest representation due to low available bandwidth. Then the resources are increased in time intervals two and three. Therefore, the client software chooses the higher representation(s). Eventually, in time interval four and five the bandwidth resources stabilize at a medium level and therefore the client software continues to stream the representation with fair quality.

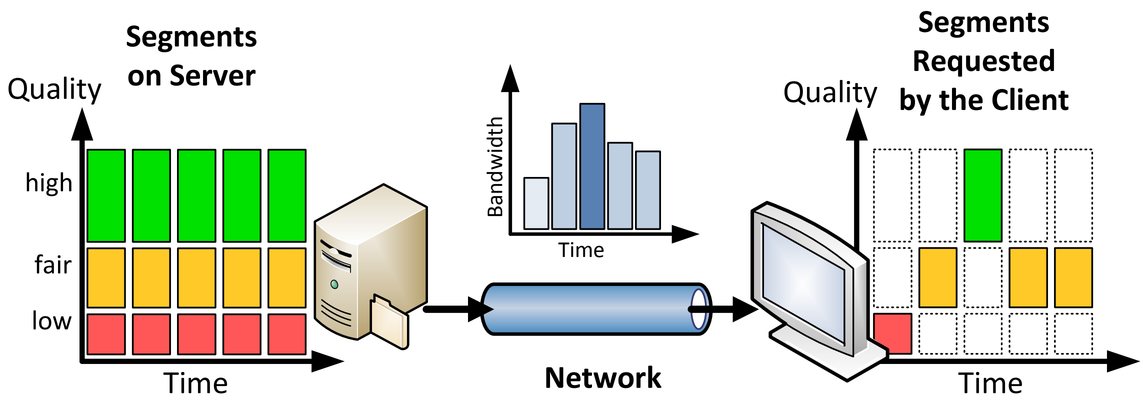


Figure 2.10: Example for adaptive streaming using MPEG-DASH. The client adapts the downloaded quality based on the available resources.

Lederer et al. [68] showed that MPEG-DASH streaming over an ICN infrastructure is possible. In [69] they argue that there are actually two major options to support MPEG-DASH in ICN. The first opportunity would be usage of a proxy translating HTTP GET requests to their ICN equivalent of Interest messages. For example, Detti et al. [70] implemented such a proxy to enable offloading in cellular networks using Information-Centric Networks. The second opportunity would be a complete integration of MPEG-DASH into ICN. Lederer et al. favour the direct integration and discuss the necessary architectural changes to the classical MPEG-DASH architecture. According to [69] the necessary changes are minimal, including the following three points: *i*) The MPD specification needs to be updated with respect to the segment URIs using an ICN naming scheme instead of HTTP URLs; *ii*) The DASH client is enhanced with an ICN access module enabling the request and delivery of segments via information-centric principles; *iii*) The HTTP server providing the segments is replaced by an ICN infrastructure possibly including multiple content sources.

Figure 2.11 illustrates the suggested translation from DASH-URIs to the DASH-ICN namespace [69]. Lederer et al. use the versioning and segment fields of the hierarchical ICN naming scheme (cf. Figure 2.5) to represent the individual representation sets and their segments. This seems to be an elegant approach to integrate DASH with ICN. However, the authors do not discuss possible negative implications of this idea. For instance, if the versioning information is used for representing the representation set, how can network elements such as router be aware of this fact? A network element may imply a caching policy that stores/caches only the newest version of an ICN packet. Changing the meaning of the versioning field may lead to unexpected and negative side effects.

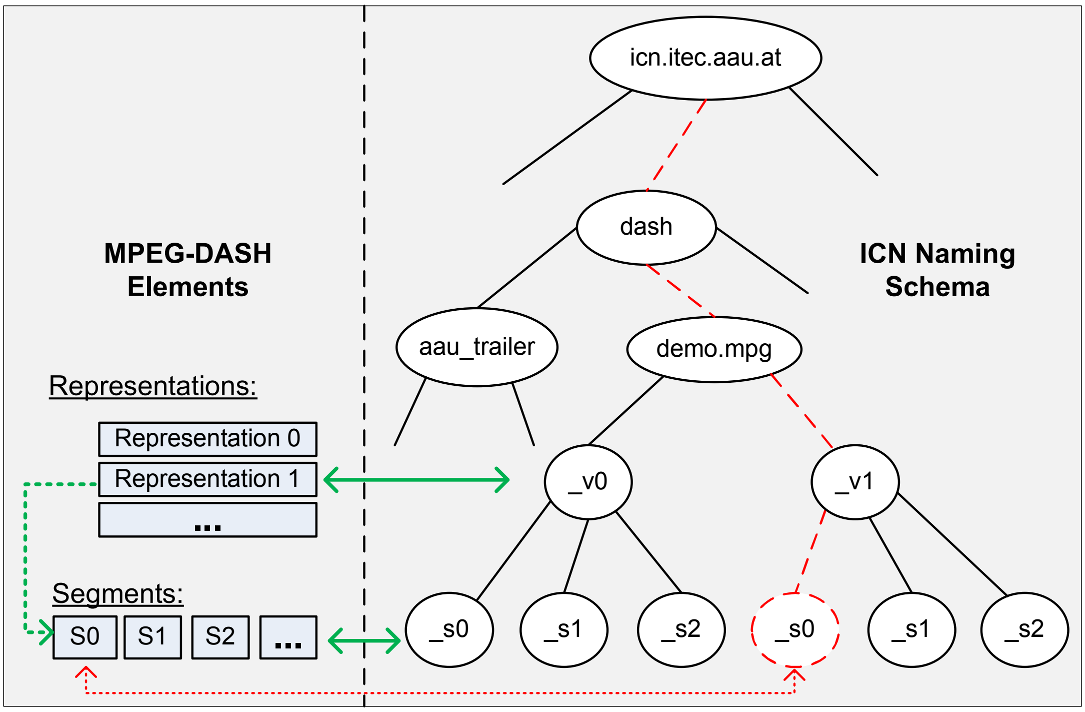


Figure 2.11: Suggested mapping for MPEG-DASH representations and segments to the ICN namespace [69].

Lederer at el. [69] implemented a prototype realizing content delivery based on named content chunks instead of relying on HTTP connections as foreseen in the ICN approach CCNx [34]. Furthermore, Lederer et al. [71] showed that the idea of using multiple links as envisaged in CCNx is beneficial for content delivery with MPEG-DASH, especially when considering mobile scenarios. Mobile devices tend to have multiple network interfaces resting on different transmission technologies, each of which having its advantages and disadvantages regarding signal coverage, transmission capacity or energy consumption. For mobile clients it is therefore important to have the opportunity to readily switch among the interfaces, which is easily possible in ICN, but challenging using classical HTTP/TCP connections.

2.3.5 NDN-Specific Approaches

The NDN community provides video distribution solutions apart from the MPEG-DASH standard. Still, those solutions share many common ideas with MPEG-DASH. The first NDN specific approach for video distribution was NDNVideo [72]. Nowadays, this approach is rather obsolete due to changes in NDN's packet format. NDNVideo is followed by

NDNLive and NDNTube [73]. This section focuses on the two successors, since they adopted most of the ideas and characteristics pioneered by NDNVideo. While NDNVideo supported both live and on-demand video streaming, the successors strictly separate these applications. NDNLive has been designed to support the usage of live streaming only, meaning that this approach is used for distributing content that is consumed in the moment it is generated. NDNTube is designed based on the model of YouTube and enables on-demand streaming of pre-recorded content. In contrast to MPEG-DASH, the NDN-based approaches do not explicitly use a manifest file such as the MPD. The NDN community rather tries to make use of the hierarchical namespace, encoding most of the relevant information within the content names. This is a similar approach as suggested by Lederer et al. [68].

Figure 2.12 illustrates the suggested namespace hierarchies for NDNLive and NDNTube. It can be seen from the figure that both namespaces separate the audio and video streams just as foreseen in MPEG-DASH. Furthermore, both approaches split content into small parts. However, while MPEG-DASH uses segments containing several seconds of content information, NDNLive and NDNTube split the content on a per-frame basis. Using NDNLive and NDNTube no segment lists must be provided, since an implicit template-based request schema is realized. A client may address the succeeding frame by specifying the correct sibling of the current frame, e.g., by increasing the frame number in the content name. According to the authors of [72], adaptive streaming can be provided by offering the content in various encodings. As the client is in full control of what it is requesting, it can easily adapt using the provided information carried by the content name. More detailed information regarding the content's attributes (resolution, bitrate, etc.) can be found under the prefix *stream_info* (cf. Figure 2.12). Basically, this namespace provides identical information as the definition of an MPEG-DASH representation. However, the authors of [72, 73] do not specify a policy in which format this information should be provided.

Due to the varying requirments for NDNLive and NDNTube their namespace hierarchy is slightly different. In NDNLive the basic entity of consumption is a stream, identified by the *stream_id*. For NDNTube the basic entity is a single video identified by its *video_name*. The concept of NDNTube foresees that all available videos are listed within the *playlist* namespace, which is also version-controlled by using time stamps. Each time a video is added or removed from the system, the playlist is updated and stored considering the current timestamp. In NDNLive the timestamp has a different purpose, and therefore it is located in a different position of the hierarchical namespace. Here it carries the information

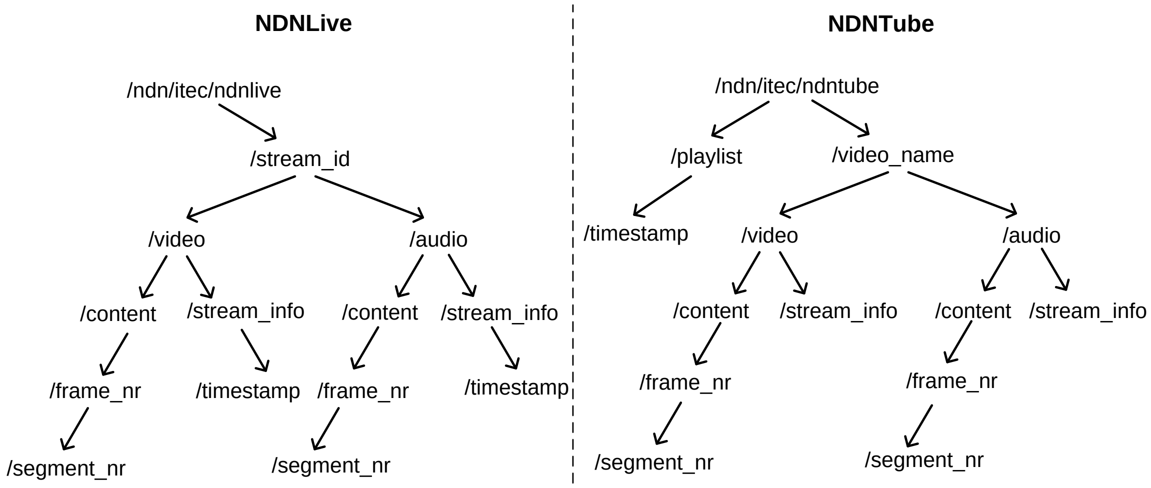


Figure 2.12: Suggested namespaces for NDNLive and NDNTube [73].

of the latest frame number that is provided by the content producer. This enables consumers to easily access the newest frames without watching all the previous ones. For more detailed information regarding the applications NDNVideo, NDNLive and NDNTube we kindly refer to [72, 73].

Note that in case of NDNLive and NDNTube the term segment denotes a completely different concept in contrast to MPEG-DASH. For NDNLive and NDNTube segments are the individual chunks of a single frame. Frames have to be segmented for network transmission if they do not fit in a single Data packet due to space restrictions. Although the authors of [72] claim that adaptive multimedia streaming is possible, NDNLive and NDNTube [73] currently do not support adaptive streaming. An explanation for the absence of adaptive mechanisms in these two approaches is that basically all modern video codecs use temporal prediction for a certain number of consecutive frames (also known as Group of Pictures (GOP)) to achieve a high data compression rate. Due to the inherent dependency of frames within a GOP, switching between different content representations is only possible at GOP boundaries. Therefore, switching to a different content representation is more complex in NDNLive and NDNTube and has not been yet realized in the available prototype software.

2.3.6 Adaptive Streaming by Use of In-Network Transcoding and In-Network Adaptation

Another approach for implementing adaptive multimedia streaming apart from MPEG-DASH in ICN is the use of in-network transcoding. Jin et al. [74] propose PAINT, a novel

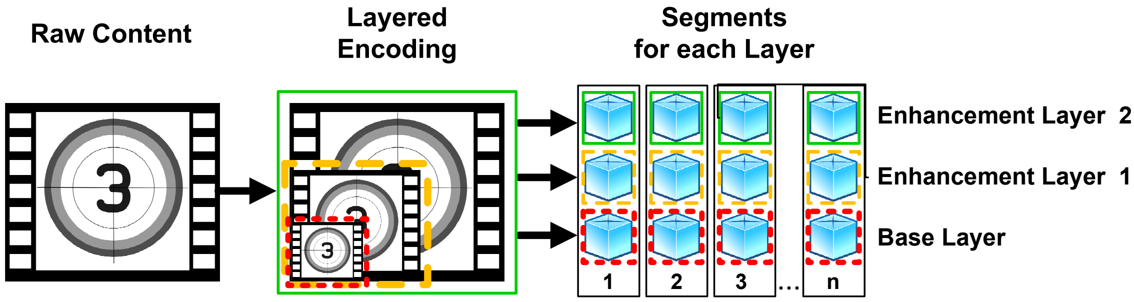


Figure 2.13: Scalable Content Encodings for ICN-based network delivery enable in-network adaptation by omitting enhancement layers that are optional for video decoding.

scheme for PArtial In-Network Transcoding in information-centric networks. This system employs ICN-enabled routers that use real-time in-network transcoding. The objective is to reduce the content distribution costs for adaptive multimedia streaming. This shall be achieved by ICN-enabled routers that only cache the highest content representation and employ in-network transcoding for requests of lower representations. The authors formulate a cost-minimization problem that optimizes the trade-off for bandwidth (transmission), storage (caching), and computational power (transcoding) consumption for the use case of adaptive video streaming. Jin et al. [74] claim that their approach achieves cost savings up to 50%. Nevertheless, they do not consider that almost all recent ICN approaches employ content-based security mechanisms. The authors of this thesis emphasize in [75] that in-network transcoding is inhibited if content-based security is employed. Using content-based security, network packets are digitally signed. Therefore, they can not be modified by third parties such as intermediate routers (cf. Figure 2.2). Hence, in [75] we indicate opportunities to circumvent this challenge. However, we consider none of them as applicable for a real deployment scenario. We suggest in [75] the use scalable content encodings such as the Scalable Video Coding (SVC) Extension of the H.264/AVC [76]. This allows to achieve the same advantages as provided by in-network transcoding as will be explained in the following.

In contrast to classical video encodings, SVC provides a layered encoding scheme as illustrated in Figure 2.13. Video content is delivered in multiple layers including a base layer and several (optional) enhancement layers. To decode a video, one requires at least the base layer. Every additional enhancement layer available at the time of decoding provides an enhancement in the spatial, temporal, and/or quality domains (cf. Figure 2.8). Using SVC-encoded video content provides the opportunity to enable in-network adaptation on ICN-enabled routers [75]. In contrast to the transcoding approaches (where packets have to be modified), routers are only required to drop the correct packets to realize content adaptation within the network. For instance, in the case of network congestion routers may decide to drop all packets carrying enhancement layers and deliver only base layer packets to the consumers. For instance, this approach is used by Liu et al. [77] to implement a hop-by-hop DAS-based video streaming approach. In [77], ICN-enabled routers are used to adjust the video quality in the network in order to avoid resource shortages. Similar to the transcoding approach, scalable content encodings reduce the required cache space compared to classical MPEG-DASH delivery using non-scalable content encodings [75]. In the case of non-scalable encodings, consumers requesting the same content, but in different representations, ask for different pieces of information. However, in the case of scalable content encodings, those consumers have at least the base layer in common providing a more efficient cache utilization.

CHAPTER 3 Promises and Challenges for Effective Content Distribution in ICN

"Things can change so fast on the Internet."

— Tim Berners-Lee, 1955*

This chapter investigates the promises of ICN concerning effective multimedia dissemination. Prominent keywords such as network-inherent caching, (adaptive) multi-path transmissions and context- and content-aware multimedia delivery are on everyone's lips in the research community. They are used to promote ICN as an enabler for effective content distribution providing significant enhancement over today's architecture. In the following we are going to outline these promises. Furthermore, we are challenging them and conduct investigations to assess whether ICN will really be able to enhance content distribution in future networks. Our evaluations show that this is truly the case, however, they also indicate that there is still much room for improvement.

The discussion of this chapter is structured into three sections. First, in Section 3.1 we discuss the promises and illustrate their potential benefits based on simplified scenarios. These simple use cases provide insight into the potential capabilities of ICN. Then, in Section 3.2 we examine the practicability of the presented approaches by sharpening the problem formulations. Here, the objective is to identify difficulties that emerge when the discussed promises are challenged with realistic problem definitions. Therefore, we model and investigate the concrete application of pull-based Dynamic Adaptive Streaming (DAS) in ICN. We perform a theoretical study modelling multimedia streaming as a Multi-Commodity Flow Problem [78]. This model provides us with the theoretical upper bound(s) that should be achievable for content delivery in ICN. Furthermore, we conduct network simulations and compare the obtained results to the aforementioned theoretical upper bound(s). Finally, we conclude this chapter in Section 3.3 by matching the promises, the theoretical upper bounds, and the simulation results. The findings obtained in this section are used as a starting point for further research to enhance multimedia delivery in ICN.

3.1 Content Distribution in ICN: The Promises

Studying ICN literature ([34, 47, 79–81]), for the community it seems to be evident that ICN is the enabler for effective content delivery. Taking Named Data Networking (NDN) [15] as representative, we are going to list and discuss in the following the most important promises with a focus on pull-based streaming. We start our discussion with the concept of in-network caching (cf. Subsection 3.1.1), followed by the promise of adaptive multi-path delivery (cf. Subsection 3.1.2) and give an outlook on context- and content-aware multimedia delivery (cf. Subsection 3.1.3).

3.1.1 In-Network Caching

As perfectly formulated in [82], ICN foresees a transformation of traditional, centralised caching overlays (e.g., CDNs) to a decentralised and uncoordinated environment. The idea is to enhance network elements (e.g., routers) with small buffers. Replicas of forwarded packets are stored in these buffers and are used to respond to subsequent requests for the same content without the need to forward the request to the content origin wasting bandwidth and delaying service time. Network-inherent caching is enabled by two major architectural design choices in the NDN approach (cf. Section 2.1.2). First, content is addressed by its name [15] decoupling content form its location (e.g., IP address). Second, content is secured using content-based security mechanisms [18] (digital signatures) providing trust for the users by enabling data integrity and authenticity verification.

Ubiquitous caching is probably the most important step towards efficient content delivery in ICN. Every networking element that is equipped with a small storage has an active part in distributing content. Van Jacobson, who played an important role in triggering the enthusiasm about content-oriented networking [34], thinks even one step further:

> *For content networking, a guy on a bicycle with a phone in his pocket is a networking element. He's doing a great job of moving bits.* [83]

So, even consumer mobility, which is a large challenge in today's networks, may support content distribution based on network-inherent caching. For instance, consider emergency response scenarios (e.g., a major earthquake), where large parts of the local available infrastructure are damaged or offline due to power failures. One of the major challenges in such scenarios is the rapid deployment of a communication infrastructure [84]. Using ICN-based technology, ad-hoc networks can be readily realised. First responders or drones may act as "data carriers" providing important information (e.g., image or video footage) to the incident commander that coordinates the rescue operation.

Nevertheless, this thesis is focused on classical content distribution. Therefore, we investigate a typical content delivery scenario. Figure 3.1 depicts a simple use case, that is used exemplarily to illustrate the significance of caching in the network for effective content delivery. We are going to compare the performance of NDN-based content delivery versus classical data transport using IP without network-inherent caching. For this scenario we have chosen a hierarchical tree-like topology, where on the top of the hierarchy a content server is placed. The content server is located within an ISP's network hosting relevant data for the ISP's customers. For instance, the content server could be part of a globally distributed CDN. On the right hand side of the figure the available link capacities (bi-directional) and delays for the links are indicated. For the sake of simplicity, we assume that each router maintains a cache of unlimited size to avoid side effects of different caching strategies for this illustrating example.

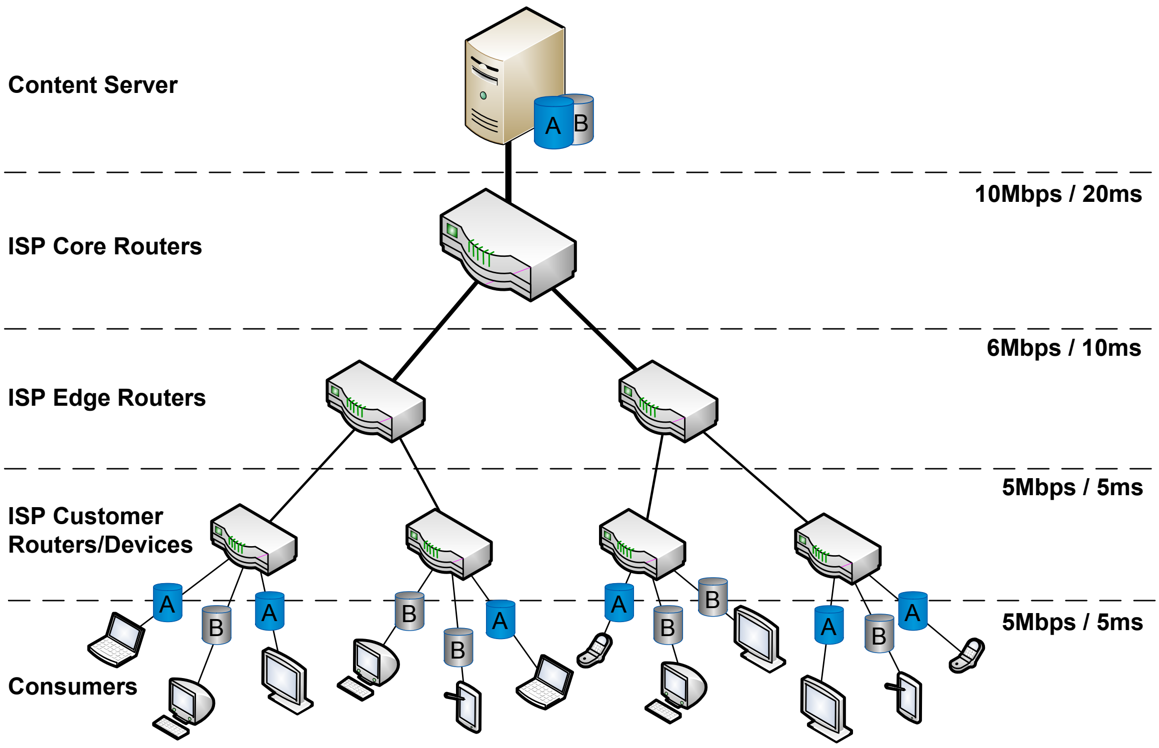


Figure 3.1: Tree-like network used to illustrate the benefits of network-inherent caching, comparing an NDN- and an IP-based delivery scenario.

For this scenario, we assume that consumers are interested in two different contents A and B (e.g., two popular videos) that are consumed by means of progressive download. We further assume that the consumers stream the contents simultaneously starting within a short time window of 5 seconds. The starting times were selected randomly following a uniform distribution. Note that for this comparison the small time window does not increase the cache hit ratio for the NDN-based delivery scenario, because we anyway assume unlimited cache for this simplified scenario. In the case of NDN, the clients are configured to consume the content with a constant bitrate of 2 Mbps, and we further assume that each Data packet carries a payload of 2048 bytes. For the TCP/IP scenario, we employ TCP New Reno [85] for congestion control. We have implemented this scenario in ns-3 [86], an event-based network simulator. For the NDN-based scenario we use the available ndnSIM plugin [37] for ns-3. The results are presented in Table 3.1 showing *i*) the average achieved goodput (throughput minus overhead) per client; *ii*) the average (one-way) hop count per received Data packet or transmitted TCP segment; and *iii*) the average round-trip time (RTT) for each Interest/Data packet pair, and for each TCP segment and its corresponding acknowledgment (ACK) message. Furthermore, the overall cache hit ratio for the NDN-based scenario is depicted (in the classical IP case there are no caches, thus no cache hit ratio is provided).

It is evident from the results that NDN-based content delivery outperforms classical TCP/IP-based delivery in this scenario. In the TCP/IP scenario, congestion is caused by the large amount of requests forwarded to the content server leading to an average goodput of only 737.38 kbps per consumer. In NDN, congestion can be avoided by the network-inherent caches, particularly from the caches deployed close to the consumers (ISP customer routers and edge routers, cf. avg. hop count in Table 3.1). Therefore, NDN is able to maintain a higher average goodput of 1945.68 kbps per consumer. This is more than 2.5 times better than in the TCP/IP scenario. The high cache hit ratio of 41.3% can be explained by the selected scenario, which represents a tree-like delivery scheme of popular content given unlimited caches. The benefits of network-inherent caching can also be observed from the required packet hop count. In case of TCP/IP all requests have to be satisfied by the content server, resulting in a high and constant (one-way) hop count of 4 hops per TCP segment. In NDN instead, a Data packet requires only 2.18 hops on average. This also has a positive influence on the average delivery time per packet. While the RTT for a request in NDN is only 45.92 ms, in IP it takes 85.50 ms.

Scenario	Goodput	(one-way) Hop Count	RTT	Cache Hit Ratio
TCP/IP	737.38 kbps	4.00	85.50 ms	–
NDN	1945.68 kbps	2.18	45.92 ms	0.413

Table 3.1: Average values of the investigated parameters comparing IP- and NDN-based content delivery for the scenario illustrated in Figure 3.1.

As previously indicated, the former example was idealized assuming infinite large caches hiding important challenges. In reality, caches are of limited size, which leads to the following important questions [87]: *i*) Which Data packets shall be placed in the cache, once they are received at a given node? *ii*) Once the cache capacity of a given node is exhausted, which Data packets shall be evicted in favour of others? The former challenge is known as the *caching policy* (or cache decision policy), while the latter challenge is known as the *cache replacement strategy.* In the context of ICN, when people talk about *caching strategies*, they usually mean the combination of both concepts, caching policy and cache replacement strategy. While in traditional caching systems cache locations are often predetermined [87], in ICN caches are ubiquitous requiring more sophisticated caching policies. Therefore, caching policies are a hot research topic in ICN leading to many published approaches as surveyed in [88]. Caching may not solely be performed on the delivery path of content (on-path caching), but could also be performed off-path [89]. The simplest available caching policy is *Cache Everything Everywhere (CEE)*, where a copy for each data packet is stored that traverses the current node. However, this can be ineffective due to the large redundancy that is caused by CEE as argued in [90]. *Probabilistic caching* [82] goes one step further trying to reduce cache redundancy by creating and storing copies with a certain probability only at each intermediate hop. There is a large variety of effective caching policies using even more sophisticated concepts like *Cooperative Caching* or cache coordination through implicit and/or explicit information exchange among the individual networks nodes (usually in a limited area) [87]. In [88] Zhang et al. provide an extensive list of available approaches indicating the individual advantages of each caching policy in an overview table. In [91] Ioannou et al. derive a taxonomy for on-path caching and discuss the interplay between caching and forwarding (forwarding is discussed in Subsection 3.1.2).

While caching policies are a hot research topic in the ICN community, cache replacement strategies are not of particular research interests. According to Zhang et al. [87] this has several reasons. First, replacement algorithms have already been extensively researched in the Web cache literature [92]. Second, ICN cache replacement has to be performed as fast as possible (probably at line-speed). Therefore, complex and resource intense replacement algorithms are not suitable. Third, it has been shown that simple random replacement strategies have performance comparable to, e.g., the well known strategy Least Recently Used (LRU) [93].

3.1.2 Adaptive Multi-Path Forwarding

As indicated in Section 2.2, NDN introduces a stateful forwarding plane [47]. While there are many additional benefits for networking arising from this concept (which will be discussed in detail in Section 4.1), especially multi-path forwarding is of considerable interest for effective multimedia delivery. Multi-path forwarding in NDN is possible due to a combination of the stateful forwarding plane and the NONCES in the Interest packets (cf. Figure 2.4). In contrast to IP, where looping packets at the network layer can not be identified, NDN achieves loop prevention by name and NONCE matching of received Interests to those already in the PIT. This basically enables forwarding on multiple paths without the risk of undetected looping packets and removes the strict limitations routing protocols enforce in classical IP-based networks. Note that in IP-based networks multi-path delivery is possible, however, not at the network layer, but rather at the transport layer, e.g., by using Multipath TCP (MPTCP) [94].

The benefits of multi-path forwarding become obvious when studying a simple scenario as indicated by Figure 3.2. In this scenario, a single consumer of network ISP_1 wants to consume a video offered by a server in network ISP_4. ISP_1 has no direct connection to ISP_4; it may reach ISP_4 via ISP_2 or ISP_3. Considering the classical TCP/IP-based case of today's Internet, the data is either delivered via ISP_2 or via ISP_3. However, in the case of NDN-based communication, the traffic load should be ideally split among the two available delivery paths ($\{ISP_1 \leftrightarrow ISP_2 \leftrightarrow ISP_4\}$ and $\{ISP_1 \leftrightarrow ISP_3 \leftrightarrow ISP_4\}$). The NDN-based approach has two major advantages: *i*) the combined available bandwidth of both delivery paths can be used if required; and *ii*) the inherent redundancy provides more resilience, e.g., to short-term effects such as link failures [47].

We have conducted the experiment illustrated in Figure 3.2 using the ns-3 simulator. We assume similar settings as for the previous experiment (cf. Subsection 3.1.1). For the TCP/IP case we again use TCP New Reno [85] for congestion control. For the NDN-based delivery scenario we assume a constant download bitrate of 5Mbps. In contrast to the previous scenario, this network provides multiple paths towards the content server.

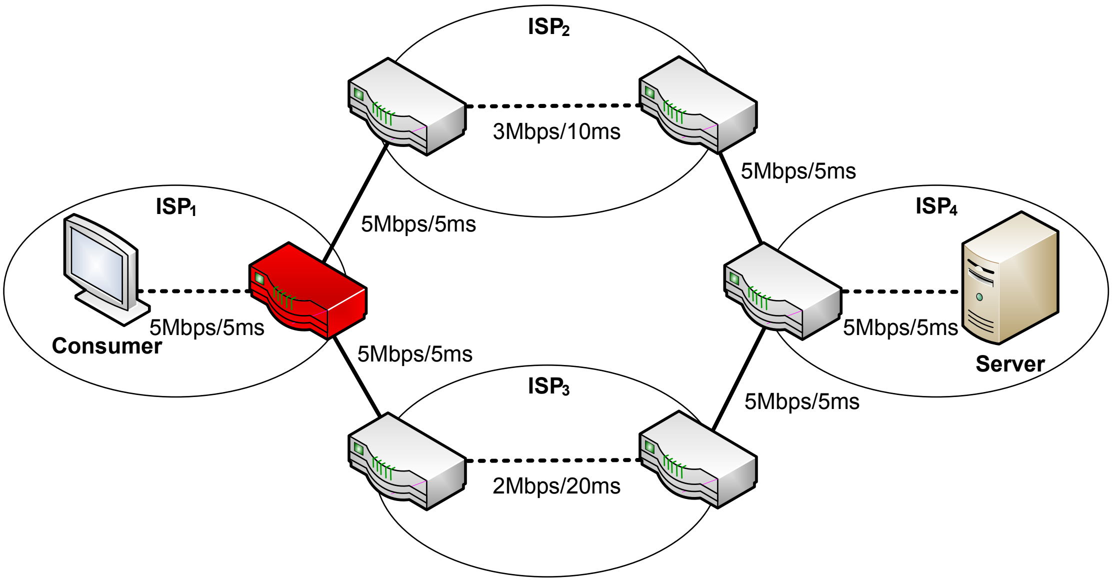


Figure 3.2: Example network with four ISPs. A consumer in the network of ISP_1 wants to stream a video from the server located in the network of ISP_4.

Scenario	Goodput	(one-way) Hop Count	RTT	Cache Hit Ratio
TCP/IP	2668.83 kbps	5.00	87.49 ms	–
NDN	4550.70 kbps	5.00	117.58 ms	0.0

Table 3.2: Average values of the investigated parameters comparing IP- and NDN-based content delivery for the scenario illustrated in Figure 3.2.

Therefore, we employ an Interest forwarding strategy [95] for NDN that balances the load on all available delivery paths (cf. Request Forwarding Algorithm (RFA) in Subsection 4.1.3). The results of the simulations are summarized in Table 3.2. The figures clearly show the advantage of multi-path forwarding. NDN is capable of using both paths at the same time for content delivery leading to an average goodput of 4550.70 kbps, while TCP/IP-based transmission is restricted to a single path ($\{ISP_1 \leftrightarrow ISP_2 \leftrightarrow ISP_4\}$) resulting in a goodput of only 2668.83 kbps. In this scenario, the average RTT for TCP/IP content delivery is lower than the one using NDN. The reason for this is that NDN also considers the additional path $\{ISP_1 \leftrightarrow ISP_3 \leftrightarrow ISP_4\}$, which has a higher delay than the other path.

As previously mentioned in Section 2.2, in NDN a Data packet always follows back the same way as its initial Interest. For this reason, the effective forwarding of Interests is of considerable importance and influences the performance of data transmission significantly. For instance, consider the aforementioned scenario depicted in Figure 3.2. Here the highlighted router in red has an important role. Incoming Interests from the consumer can be forwarded on multiple outgoing faces. Which face is actually used for forwarding an Interest, is decided by the so-called *Forwarding Strategy*. Recall Figure 2.7, which illustrated the processing of an Interest message. The final step before an Interest can be forwarded is a FIB match, which may return none, a single or also multiple matching faces (cf. FIB in Figure 2.6). Forwarding strategies may pursue various objectives (maximize throughout, minimize delay, achieve load balancing, etc.) when selecting the outgoing face(s). The set of choices that is left to the forwarding strategy in choosing the outgoing face(s) for a given Interest provides significant adaptivity to changes in the network and to application/content demands, which is also the reason for naming it adaptive forwarding.

3.1.3 Context- and Content-Aware Data Delivery

Another major promise of ICN is that it provides the opportunity to enhance networking devices (e.g., routers) with advanced capabilities. This especially includes the vision of context- and content-aware routers that provide the opportunity to effectively control and enhance content delivery [96, 97]. Here the terms *context* and *content awareness* represent a broad area of properties relevant for efficient content delivery and consumption. This information should be provided to the routers that deduce relevant actions to actively support content distribution. In general the term content awareness represents relevant information about the content. For example this information may include, but is not limited to: *i*) content characteristics (e.g., delay tolerant vs. intolerant); *ii*) required QoS for user satisfaction; and *iii*) available content representations (e.g., different encodings with varying quality (cf. Figure 2.8)). The term context awareness includes considerations about the network and its entities which can be used for traffic engineering purposes. Relevant information may include, but is not limited to: *i*) available bandwidth on the individual content delivery paths; *ii*) length/delay of the individual paths; *iii*) employed caching mechanism(s) by the networking entities on the delivery paths; *iv*) content source conditions (e.g., server load); and *v*) object/content popularity in the network.

Currently there are no concrete implementations available to exploit context or content awareness in an NDN-based network. However, there are two theoretical approaches that propose how context and content information can be used in ICN to enhance content delivery. Pavlou et al. [96] propose a mediation approach for content access that takes into account content characteristics, server load and network distance when resolving the location of content replicas. With respect to these attributes the proposed system is able to obtain the "best" copy from the network. Kamel et al. [97] go one step further and provide the idea of a context-aware ICN ecosystem that additionally considers path load information and therefore facilitates further traffic engineering capabilities. Recently, CONCERT [98] (A Context-Adaptive Content Ecosystem Under Uncertainty), a CHIST-ERA project commenced that has the objective to design and develop a context-aware content ecosystem based on ICN technologies. The project goals include: *i*) the concrete specification of context information and its representation including communication interfaces for context information exchange; *ii*) the development of learning algorithms for coordinated decision making techniques; *iii*) cross-layer (network, content, control) and cross-player (end users, ISPs) content and network adaptations; with respect to uncertainty scenarios (changing network conditions, content popularity, etc.). As can be seen from these basic questions, research in this area is at an early stage. However, acquiring answers to these challenges will provide significant opportunities for effective content dissemination in future networks.

3.2 ICN Promises under Test: A Performance Investigation of DAS in NDN

The previous section clearly illustrated the individual building blocks (in-network caching, adaptive forwarding, context-/content-aware data delivery adaptation and/or object resolution) the promise of effective and efficient content delivery in ICN rests on. In this section we investigate if NDN (as a representative for an Information-Centric Network) is capable of fulfilling the promise of effective and efficient multimedia delivery, or whether expectations on the individual building blocks are too high. As evaluation scenario we use multimedia streaming based on the principles of MPEG-DASH. We consider this as a suitable and realistic use case, since today Dynamic Adaptive Streaming (DAS) is state of the art for on-demand and real-time multimedia streaming services. Furthermore, it has been shown that MPEG-DASH fits NDN's consumer-driven communication principle very well [99–101] (cf. Subsection 2.3.4) and should therefore be able to exploit NDN's full range content delivery capabilities. So, the ultimate objective of this section is to examine the performance of pull-based DAS in NDN,

- using different (Interest) *forwarding strategies* (at the network level),
- using different *caching strategies* (at the network level),

- using different *client-side adaptation mechanisms* (at the application level),

under non-optimal conditions (e.g., network congestion). Especially, investigating the interplay between application-level algorithms implementing the principles of DAS and the network's forwarding and caching strategies may reveal interesting insights. In our investigation we will determine the performance gap between the theoretically possible and the realized streaming performance by NDN considering multiple concurrently streaming DAS consumers.

In order to derive upper bounds for the multimedia streaming performance in NDN (without and with caching), we model the concurrent streaming activities by a given number of clients in a network as a Multi-Commodity Flow Problem (MCFP) [102]. The solution to the MCFP provides us with upper bounds taking multi-path transport into account. Both the theoretical investigations and the practical evaluations clearly state that NDN, as a candidate for a Future Internet architecture, is able to compete with current IP-based networks in the case of multimedia streaming. However, further improvements are possible. Please note that in this section we do *not focus on finding a near-optimal adaptation heuristic or forwarding strategy* for DAS in NDN. We rather carefully select representatives for each of the algorithms and compare every possible combination with respect to their general performance to deduce general conclusions.

The remainder of the section is organized as follows. Subsection 3.2.1 introduces the preliminaries for the conducted evaluation. The fractional MCFP providing the theoretical performance assessment is introduced in Subsection 3.2.2. The practical evaluation using an NDN-based network simulator is presented in Subsection 3.2.3.

3.2.1 Evaluation Preliminaries

In the following we discuss the preliminaries for the conducted evaluation. This includes a rationale for the employed content (dataset) and a discussion of the selected client-based adaptation mechanisms and network forwarding strategies.

3.2.1.a An SVC-Encoded MPEG-DASH-Compliant Dataset

For our experiments we use MPEG-DASH-compliant multimedia content that is encoded using the Scalable Video Coding (SVC) [76] extension of the H.264/AVC standard [103]. SVC offers the possibility to encode video content into a base layer and several enhancement layers. The enhancement layers build upon the base layer and provide scalability in the spatial and/or temporal and/or quality domain(s) (cf. Figure 2.8). In contrast to non-scalable encodings, this principle fits very well with NDN's inherent caching, since clients requesting different content representations at least have the base layer in common. As argued in [75], this increases the overall cache hit ratio in the network and, thus, enhances the delivered quality of the multimedia content. As test content we use MPEG-DASH-compliant SVC-encoded multimedia content with a segment size of two seconds. The multimedia content is taken from our SVC-DASH dataset [104]. The dataset provides four short movies with an average duration of about 12 minutes. We concatenated the short movies to obtain content with a duration of about 48 minutes, which is roughly the length of a typical TV episode. For the dataset presented in [104] we encoded multimedia content in various variants. A variant defines the encoding parameters as well as the scalability domains (temporal, spatial, quality) (cf. Figure 2.8). For this evaluation we have chosen a variant providing Signal-to-Noise Ratio (SNR) scalability only since we are solely interested in the objective streaming performance. We are not interested in the impact of possible content adaptations on the Quality of Experience (QoE). The chosen content is provided using a base layer and two enhancement layers. The base layer (henceforth denoted as $L0$) has an average bitrate of approximately 640 kbps. The first enhancement layer ($L1$) has a bitrate of approximately 355 kbps. In order to play back a segment at the quality of $L1$, one has to fetch the same segment of $L0$ and $L1$ yielding a combined multimedia bitrate of $L0 + L1 \approx 995$ kbps. The second enhancement layer ($L2$) has an average bitrate of approximately 407 kbps (resulting in a cumulative bitrate of $L0 + L1 + L2 \approx 1400$ kbps for the highest quality representation).

3.2.1.b Client-based Adaptation Mechanisms

The client-based adaptation mechanism decides which representation (quality) of the video is requested by the consumer application. In order to investigate the interplay of the forwarding strategies discussed in Section 3.2.1.c and the adaptation mechanisms at the clients, we select for each possible type of client-side adaptation mechanism (no adaptation, rate-based adaptation, and buffer-based adaptation) one representative as follows:

No Adaptation: Here, a client always tries to request each segment from each layer. Thus, it simply tries to retrieve the best representation. This is a greedy approach that very probably results in playback interruptions for scenarios with limited resources.

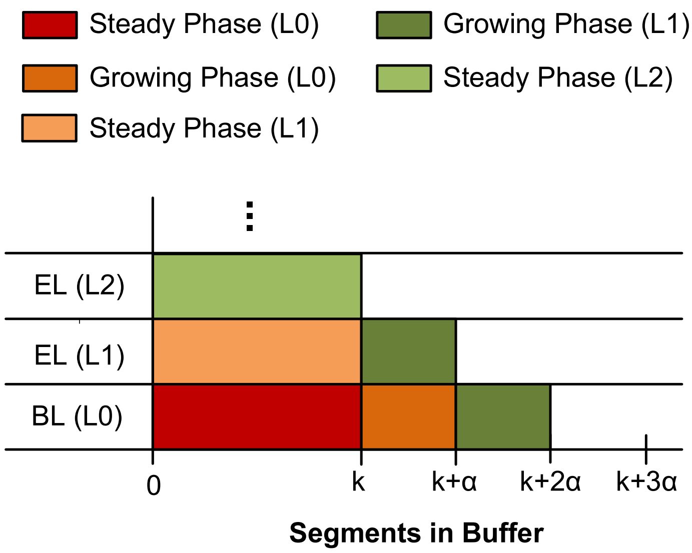


Figure 3.3: The basic idea of the buffer-based adaptation logic as proposed in [106].

Rate-based Adaptation: Here, a client measures the currently available bandwidth while downloading a segment. Then the client estimates the future available bandwidth using an exponential moving average given by $b_{k+1} = (1 - \alpha) \cdot b_k + \alpha \cdot b$, where b_{k+1} denotes the new estimate, b_k denotes the previous estimate, and b denotes the currently measured bandwidth [105]. For our experiments, we select $\alpha = 0.3$. The lower α, the more influence the historic measurements have on the estimated bitrate. The higher α, the more influence the recent measurement has. We consider $\alpha = 0.3$ as a moderate value providing a suitable balance between recent and historic measurements. Based on the estimated bitrate, the client selects a suitable representation from which it tries to download future segments.

Buffer-based Adaptation: Here, the decision which representation is selected to download a segment is based only on the client's playback buffer. We adopt the adaptation logic described in [106] that uses a deadline-based approach for selecting the appropriate representation and is optimized for SVC content. This adaptation logic tries to avoid playback interruptions at all costs by always having at least k (in our case $k = 8$) segments of the lowest representation (layer) in its playback buffer before considering higher representations. If this is the case, it changes from the so-called *steady phase* in the *growing phase* for L0, which means it downloads α (in our case $\alpha = k/2$) more segments of the base layer. If this has been accomplished, the adaptation logic tries to download the first enhancement layer for the first k segments in the buffer (steady phase for L1). Once this is achieved, it changes to the growing phase for L1 and downloads α more segments from the base and first enhancement layer, before switching to the steady phase for L2 (and so on). The quality (layer) that is downloaded for the available segments in the buffer follows a pattern of sloping stairs favoring segments from lower layers that are closer to the playback time stamp. This can be seen from Figure 3.3 that sketches the basic idea of the algorithm [106].

3.2.1.c Interest Forwarding Strategies

Since forwarding strategies have a significant influence on the performance of content delivery in NDN, we consider a variety of strategies for our investigations. For the experiments conducted in these evaluations, we use the network simulator ndnSIM 2.0 [37] which builds on top of *ns-3*. Currently ndnSIM 2.0 provides three forwarding strategies considering maximal coverage (Broadcast), minimal hop count (BestRoute) and minimal delivery time (NCC). We further extend this set by adding strategies that focus on effective cache utilization (iNRR) and throughput (SAF). In the following we briefly summarize the principles of each strategy (for a more detailed discussion we refer to Subsection 4.1.3, to Section 5.2, and to the indicated references):

BestRoute [37]: This strategy relies on routing information and forwards Interests on the path with the lowest costs considering a specific metric. We have chosen the distance (hop count) to the content origin as the relevant metric.

Broadcast [37]: This scheme forwards received Interests to all available faces (according to the faces that match the content name prefixes in the FIB, determined initially by the routing protocol), except the incoming face. Note that multiple copies of an Interest may be created if multiple faces are registered.

NCC [37]: Each node monitors the delays of its faces. The delay is defined as the time period that elapses until a forwarded Interest is satisfied by a Data packet. Interests are forwarded to the face that provides content with the lowest delay. This forwarding strategy is similar to the forwarding strategy used in CCNx 0.7.2 (`http://www.ccnx.org`). Its name was derived by flipping the initials of the term *Content-Centric Networking* (CCN) [34].

iNRR: Ideal Nearest Replica Routing [107] couples caching and forwarding. The approach makes use of an *oracle* that provides information on the availability of content in all caches in the network. The algorithm determines the nearest content replica (in terms of hop count) and forwards the Interest to the corresponding face to obtain the replica.

SAF [108]: This strategy, called *Stochastic Adaptive Forwarding*, mimics the behaviour of a water pipe system where each network node represents a crossing and distribution node with a pressure control valve. It forwards Interests based on a probability density function that is learned by observing traffic patterns. The pressure control valve is used to deal with network congestion by pro-actively discarding Interest packets that would otherwise exceed a node's transmission capabilities. SAF is the result of research that has been conducted for this thesis and has been developed considering the findings of this chapter. Therefore, SAF should be considered as out of competition with respect to the research objective (3) (cf. Section 1.2) that we are pursuing in this chapter. However, in order to not unnecessarily repeat an evaluation concerning multimedia distribution in Chapter 4, we decided to present the obtained results for SAF already in this chapter. The major benefit of SAF arises from the fact that it is not content/prefix-agnostic and, therefore, maintains a certain state for each content/prefix observed at the network nodes. The previously mentioned probability density function is learned by maximizing a given measure classifying Interests either as *satisfied* or *unsatisfied*. For this evaluation, we use a purely throughput-based measure that simply counts how many Interests are satisfied during a given time period. For more details on SAF, we kindly refer to Chapter 4.

3.2.2 Multimedia Streaming as a Multi-Commodity Flow Problem

Before we conduct network simulations investigating the performance of DAS in conjunction with different NDN forwarding strategies, we want to determine the theoretical upper bounds of the average multimedia streaming bitrate without and with *idealized* caching. Thus, we aim at finding the optimal selection of paths through the network (to the content origins or in-network caches) such that the average multimedia streaming bitrate is maximized for every client, constrained by the given network. In the context of NDN, we always speak of *multiple paths* because multi-path transmission is an inherent feature of NDN's architecture. Finding the optimal selection of paths is NP-complete if we want to solve it as an Integer Linear Program (ILP) [102]. However, if the fractional usage of the paths is allowed (solution will be real instead of integer) then the problem is solvable in polynomial time by modelling it as a Linear Program (LP) and solving it with well-known methods such as the interior-point method [109]. This LP will provide us with an upper bound for each client's multimedia bitrate assuming that the network and its characteristics are known in advance. In our scenario, we consider several clients that request different multimedia contents. Thus, we have multiple commodities. In the literature this problem is referred to as a Multi-Commodity Flow Problem (MCFP) [102]. Since the LP provides us an upper bound of the ILP, we focus on modelling and solving the LP allowing the fractional usage of the given network links.

3.2.2.a Modeling the Upper Bound without Caching

We model the *fractional* MCFP for a given network, clients and their corresponding servers using the paths from each client to its servers. One may also see the network as constrained to the maximization of the possible multimedia bitrate that each consumer may retrieve. As a preprocessing step, we compute every possible path from the clients to their servers. Let the three-tuple $G := (V, E, c)$ be a weighted graph that represents the underlying network topology, where V denotes the set of vertices, $E \subseteq V \times V$ denotes the set of edges, and $c : E \to \mathbb{R}$ assigns a bandwidth capacity to each edge. C denotes the set of clients. Then the paths from a client to a server can be enumerated by a slightly modified version of the classical breadth-first or depth-first search [110]. We denote P as the set of paths for all (s, t) pairs, where s denotes the client and t denotes the corresponding server for a given client s. We denote S as the set of all client-server pairs (s, t). Note that for a single client s multiple (s, t) pairs exist if a client's multimedia stream can be served by multiple servers. Further each (s, t) pair may have multiple delivery paths, hence, multiple sub-flows (cf. Figure 3.4 for an illustration of the terms *stream*, *flow*, and *sub-flow*). We further denote P_i as the set of paths for client i to all of its servers. For each path $p \in P$ we have a variable $x_p \in \mathbb{R}_+$ representing the bandwidth consumed on path p. This allows us to set up the LP 3.1 using vector $\boldsymbol{y} \in \mathbb{R}^{|C|}$ as auxiliary variable (henceforth vectors are denoted using bold math symbols) as follows:

$$\text{minimize} \ -||\boldsymbol{y}||_1 \tag{3.1a}$$

subject to

$$y_i \cdot \text{minBitrate}_i - \sum_{p \in P_i} x_p \leq 0 \ \ , \forall i = 1, \ldots, |C| \tag{3.1b}$$

$$\sum_{(u,v) \in p} x_p \leq c((u, v)) \ \ , \forall (u, v) \in E, p \in P \tag{3.1c}$$

$$\sum_{p \in P_i} x_p \leq \text{maxBitrate}_i \ \ , \forall i = 1, \ldots, |C| \tag{3.1d}$$

LP 3.1: A model for the upper bound of DAS in NDN without caching.

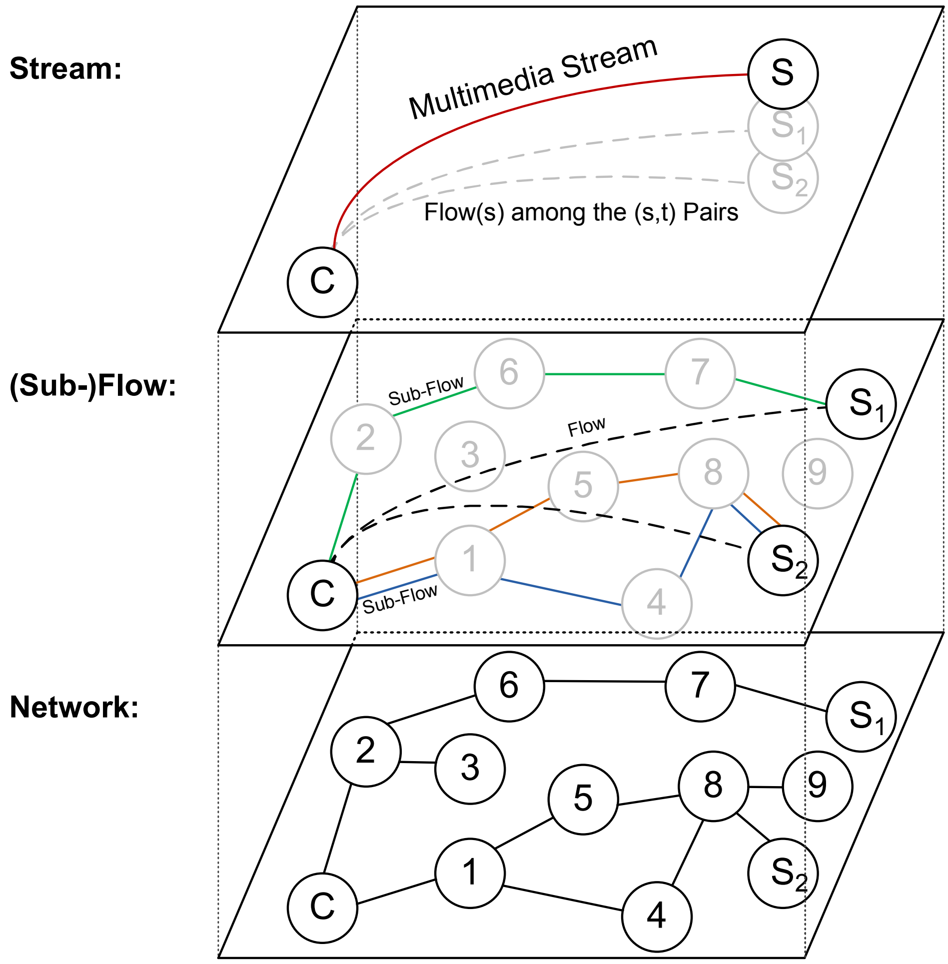


Figure 3.4: Illustration of the terms *multimedia stream*, *flow* and *sub-flow* indicating the basic concepts LP 3.1 uses. The illustration assumes a single client C consuming a multimedia stream (red line). The stream is provided by two servers, thus we have two (s,t) pairs (flows). The flows are indicated by the dashed lines connecting C with S_1 and S_2. Since there are multiple delivery paths among the nodes C and S_2 we have multiple sub-flows (blue and orange lines) for this (s,t) pair.

Equation 3.1a provides the objective function for the optimization problem. $||\cdot||_1$ denotes 1-norm, which is defined as $||\boldsymbol{x}||_1 := \sum_{i=1}^{n} |x_i|$. Here, x_i denotes the i-th element of vector $\boldsymbol{x}$ and n denotes the number of elements in $\boldsymbol{x}$. In LP 3.1, y_i represents the auxiliary variable for the i-th client, and minBitrate_i denotes the minimum bitrate of the i-th multimedia stream. Equation 3.1b denotes the constraint that each multimedia stream shall at least get the lowest possible media bitrate available. The LP becomes infeasible if this lower bound cannot be achieved by at least one of the clients (this is the case if we force $y_i \geq 1$, where $i = 1, \cdots, |C|$). This is a very strict criterion ensuring a smooth media playback for all clients. This constraint may be relaxed by choosing minBitrate_i lower than the lowest

Algorithm 3.1 Determine Upper Bound of the Average Multimedia Bitrates with *Idealized* Caching

$L \leftarrow getAllStreams(M, C)$
while $comb \leftarrow getNextDisjointCombination(L)$ **do**
    $\{rG, result_n\} \leftarrow solveMCFP(comb, G)$
    $S' \leftarrow createClientServerPairs(S, rG)$
    $\{sG, result_{|C|-n}\} \leftarrow solveMCFP(S', rG)$
    $R[comb] \leftarrow result_{|C|-n}\frac{(|C|-n)}{|C|} + result_n\frac{(n)}{|C|}$
end while
return $\max\{R\}$

available representation bitrate or by allowing $y_i < 1$. However, this might lead to clients receiving too few resources, even for streaming the base layer resulting in so-called media playback stalls (playback disruptions due to buffer drains of the video/audio buffer in the playback software). Equation 3.1c takes the edge capacities into account such that all paths that have an edge (u, v) in common do not consume more than the available capacity. Equation 3.1d denotes the constraint for restricting the maximum used media bitrate. A client cannot retrieve a higher representation bitrate than the highest available one (maxBitrate_i); this is again a very strict constraint. Allowing higher values than the highest available representation bitrate (i.e., arbitrarily high) would yield the highest possible streaming bitrate for each client.

LP 3.1 provides us with an upper bound for the case where we do not assume that any content is cached by the intermediate nodes on the paths. It further assumes that all clients start streaming at the very same time. An optimal solution to the introduced LP provides therefore also an upper bound for the streaming scenario in TCP/IP networks with the TCP extension of allowing multiple paths (MPTCP) [94] disregarding any overhead considerations.

3.2.2.b Calculating the Upper Bound with Idealized Caching

In order to take account of in-network caching when comparing client adaptation strategies and forwarding strategies to their theoretical upper bounds in NDN, we extend LP 3.1. This brings us to Algorithm 3.1. Here, we assume *idealized* caching along each path a Data packet has been sent and that intermediate nodes have unlimited cache size. M denotes the set of different multimedia contents that are retrieved by the clients. First, we add each client $c \in C$ that streams the same multimedia content $k \in M$ to a set M_k. $L \subseteq M \times C$

denotes the set of all tuples (k, M_k) and is obtained by calling the function $getAllStreams()$ (cf. Algorithm 3.1 line 1). Second, we pick a possible combination of $n = |M|$ clients from $M_k, 1 \leq k \leq n$ such that the clients request pairwise disjoint $(m_i \neq m_j, i \neq j)$ multimedia content (cf. Algorithm 3.1 line 2, denoted by $getNextDisjointCombination()$). Thus, in total we have $\prod_{k=1}^{|M|} |M_k|$ possible combinations. Let S denote the client-server pairs for the given n clients. Third, their paths are computed and LP 3.1 is solved for these n clients (cf. Algorithm 3.1 line 3, denoted by $solveMCFP()$). This yields the optimum for the average bitrate for these n clients streaming pairwise disjoint content denoted as $result_n$. Fourth, we use the residual graph rG as network graph for the remaining $|C| - n$ clients and we set all the vertices from all paths for each of the n clients as *servers* for the other clients that are about to stream the same multimedia content (cf. Algorithm 3.1 line 4, denoted by $createClientServerPairs()$). This provides us with S' that includes all client-server pairs for clients that are not part of the selected combination and all servers (including caches). We assume that all the nodes on the corresponding paths have *cached all the data* from their corresponding multimedia streams (if and only if LP 3.1 is feasible for the selected combination). Please note that this is an idealized scenario. For instance, if a client uses two paths to retrieve the desired data, not all nodes on the two paths will cache the same data (because the Interests may be forwarded arbitrarily on these two paths). Fifth, we solve the MCFP using the modified set of client-server pairs S' and the residual graph/network (cf. Algorithm 3.1 line 5, denoted by $solveMCFP()$), eventually providing the average multimedia streaming bitrate for the remaining $|C| - n$ clients. Sixth, both obtained results ($result_n$ and $result_{|C|-n}$) are averaged with respect to the number clients (cf. Algorithm 3.1 line 6). This value is then stored in the result vector R. This procedure is repeated for all possible combinations so R finally holds all results for all possible combinations of n clients. The highest average multimedia streaming bitrate, assuming *idealized* caching and unlimited cache size on each intermediate node, is given by the maximum element of the result vector R, which is returned by the algorithm.

We provide a MATLAB implementation of Algorithm 3.1 including the solution to LP 3.1 at `https://github.com/danposch/itec-ndn/` licensed under the General Public License (GPL). We further provide the source code that has been used for the evaluations in this chapter (cf. Subsection 3.2.3.a) at the previously mentioned URL (also licensed under GPL).

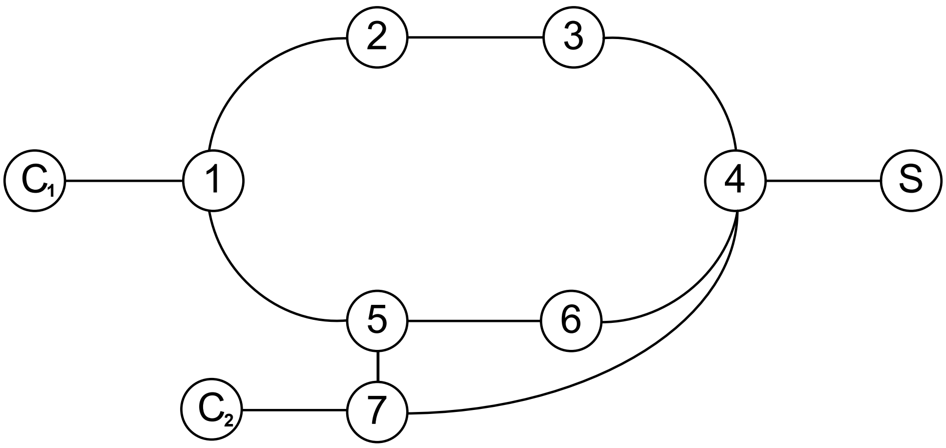


Figure 3.5: Example network with two clients (C_1 and C_2) interested in the same content available at the server (S).

3.2.2.c Example: Calculating the Upper Bounds

Figure 3.5 depicts an example network with two clients denoted as C_1 and C_2. They are interested in the same content available at a single server denoted as S. For the sake of simplicity we assume that the capacity of the links between the vertices is 1500 kbps and that the links are bidirectional. We further assume that the server provides SVC-encoded multimedia content with the following three layers/bitrates: $\{L0 = 640$ kbps, $L0+L1 = 995$ kbps, $L0+L1+L2 = 1400$ kbps$\}$. In order to solve the LP 3.1, which assumes that none of the nodes in the network caches content, we compute all paths for the client-server pairs. The paths for the two client-server pairs (C_1, S) and (C_2, S) are: $P = \{\{(C_1, 1), (1, 2), (2, 3), (3, 4), (4, S)\}, \{(C_1, 1), (1, 5), (5, 6), (6, 4), (4, S)\}, \{(C_1, 1), (1, 5), (5, 7), (7, 4), (4, S)\}, \{(C_2, 7), (7, 4), (4, S)\}, \{(C_2, 7), (7, 5), (5, 6), (6, 4), (4, S)\}, \{(C_2, 7), (7, 5), (5, 1), (1, 2), (2, 3), (3, 4), (4, S)\}\}$. So, each client has three possible paths to the server. Although the LP does not consider caching, it considers multi-path transmission as foreseen in NDN and MPTCP. The solution of the LP indicates that in the given network an average download bitrate of 750 kbps can be retrieved by the clients. This takes into account that the minimum bitrate of 640 kbps (the lowest representation/layer) shall be achieved by all clients, so that no stalls of the playback occur. If we take caching into account, we have to use Algorithm 3.1. In this case the achieved average download bitrate per client would increase to 1400 kbps. A more detailed investigation of the solution shows that Algorithm 3.1 suggests that C_1 shall request the multimedia content from the content origin S using the three available paths such that C_1 is able to achieve a media bitrate of 1400 kbps. Client C_2 then has 23 paths to all intermediate network nodes that are on the three paths from C_1 to S. Since we assume *idealized* caching, these nodes have the desired multimedia content in their local cache. Thus, C_2 is also able to maintain a download bitrate of 1400 kbps. This is for sure an artificial result since it assumes that, even though not all the Interests passed through a network node, that node still has every Data packet in its cache. However, it provides us with an upper bound for this scenario considering caching that can be used to compare against network simulations.

3.2.3 Performance of DAS in NDN

To evaluate and investigate the performance of pull-based multimedia streaming in NDN using the adaptation algorithms described in Subsection 3.2.1.b and forwarding strategies described in Subsection 3.2.1.c, we use *ndnSIM 2.0* [37], a simulation framework based on *ns-3*. First, we outline and justify the evaluation set-up. Then, we present the results comparing them to the theoretical upper bounds determined using the MCFP from Subsection 3.2.2 without and with *idealized* caching assuming unlimited cache sizes.

3.2.3.a Evaluation Setup

Figure 3.6 depicts the *fixed* network topology for the evaluation in order to investigate the pull-based streaming performance of forwarding strategies coupled with different client-based adaptation mechanisms. The network topology is *fixed* to ensure comparability among the simulations and the theoretical upper bounds provided by the MCFP. We are aware that the fixed topology is a limitation, however, otherwise the results could not be compared to the theoretical work from Section 3.2.2. In total 25 clients are placed in the network. Every five clients request the same multimedia content from the corresponding server. Thus, we have five groups of clients denoted by the colors (or numbers) red (1), green (2), blue (3), orange (4) and black (5) (cf. Figure 3.6). The servers are illustrated as rectangles labeled with S using the corresponding group color (number). The network nodes are equipped with a cache. The size of the cache is varied from 25 MB, 50 MB up to 100 MB per node, which corresponds to a cache size of approximately 1%, 2% and 4% of the total content catalogue, respectively. As suggested in [90], we consider different caching approaches: first, *Cache Everything Everywhere* (CEE), and second, *Probabilistic Caching* with a probability $p \in \{0.1, \quad 0.3, \quad 0.6\}$ of caching the seen content using a Least Recently Used (LRU) replacement strategy for both approaches.

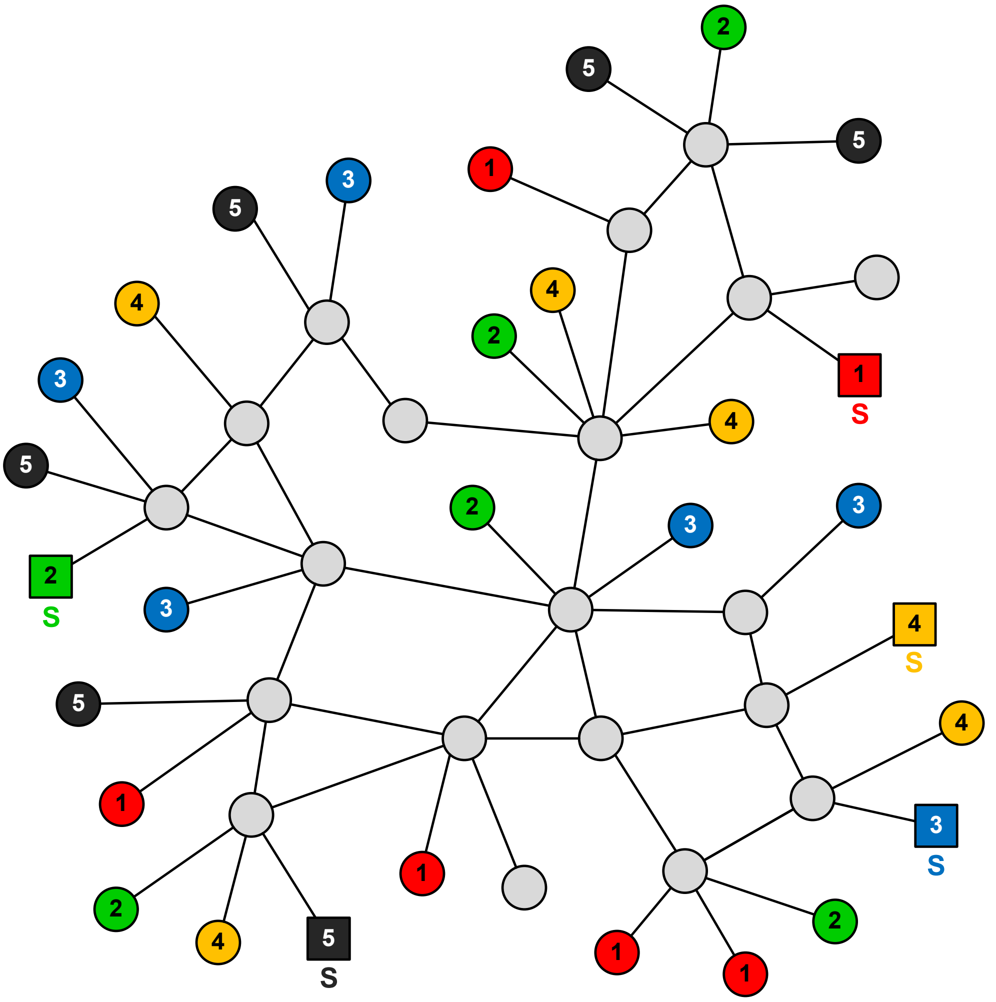


Figure 3.6: Topology used for evaluating the multimedia streaming performance.

We use two settings for the start times of the clients. The first setting exactly follows the problem description of the MCFP, which requires that all clients are configured to start simultaneously. This is again a limitation, yet required to ensure comparability to the theoretical results from Subsection 3.2.2. As the clients start at the same time, this may be beneficial for the overall caching performance, since requests for the same content are issued in a small time window and can be aggregated by the forwarding nodes. For the second setting, we draw the start time of a client from an exponential distribution (*mean=60s*, *max=180s*). This shall mimic the behavior of users joining streaming sessions, especially during prime time when a new movie or event is shown. We expect that the caching performance, and therefore the overall performance, will be worse than in the first setting, because the requests for the same content are issued in a larger time window and fewer requests can be aggregated or result in cache hits.

The links between the network nodes are bidirectional and have a bandwidth of 4 Mbps (in each direction). The links connecting the servers to their ingress/egress nodes have a bandwidth of 5 Mbps (bidirectional). The network links connecting the clients to their ingress/egress nodes have 2 Mbps. The presented topology with the given settings has been selected because it is likely that congestion will occur if all clients want to stream the multimedia content. For every configuration (forwarding strategy and adaptation algorithm) we conducted 25 simulation runs in order to reduce the influence of random variables on the sample means in the results. The duration of a single simulation run corresponds to the length of the employed multimedia content (roughly 48 minutes, cf. Subsection 3.2.1.a). For the forwarding algorithms we pre-computed all possible routes to evaluate all forwarding strategies under the same conditions. The algorithms Broadcast, BestRoute and NCC do consider the predetermined routes for forwarding only, while SAF and iNNR use the routing information merely as a starting point. The selected topology does not favor any of the forwarding strategies. All clients maintain a playback buffer that is capable of storing 50 seconds of multimedia content. Requests from clients are issued based on a constant bitrate model since congestion in NDN shall be handled by the forwarding plane [47].

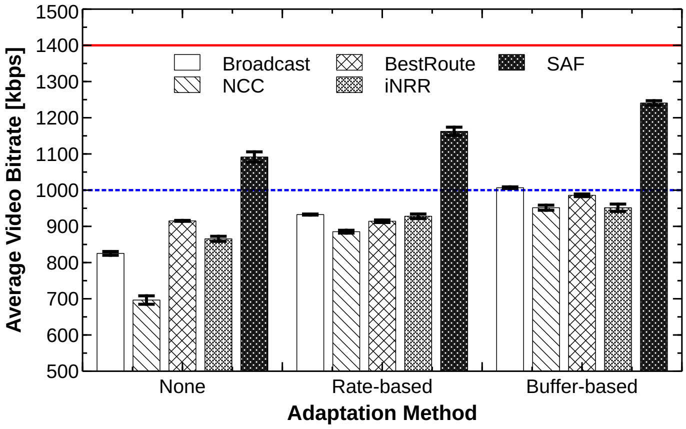
In addition to the NDN-specific simulations using ndnSIM 2.0, we provide a baseline evaluation of DASH using OMNeT++ as simulation environment utilizing the INET framework [111]. We use the presented topology (cf. Figure 3.6) and assume that the intermediate nodes are routers/switches. In order to obtain baseline results for HTTP adaptive streaming in TCP/IP-based networks, we use the rate-based adaptation logic introduced in Subsection 3.2.1 and we set the playback buffer size to 50 seconds. In analogy to the NDN scenario, we vary the starting times of the clients as described before. Instead of SVC-encoded multimedia content, we use AVC-encoded multimedia content since it is the most used video coding standard in conjunction with DASH. Furthermore, using SVC-based content in this case would only employ unnecessary overhead without providing any benefits (because no in-network caching is provided). We select *Big Buck Bunny* from the dataset [112] with a segment size of two seconds. In order to obtain the same duration as the content used for the NDN simulations, we extended the length of Big Buck Bunny by repeating it several times.

3.2.3.b Results

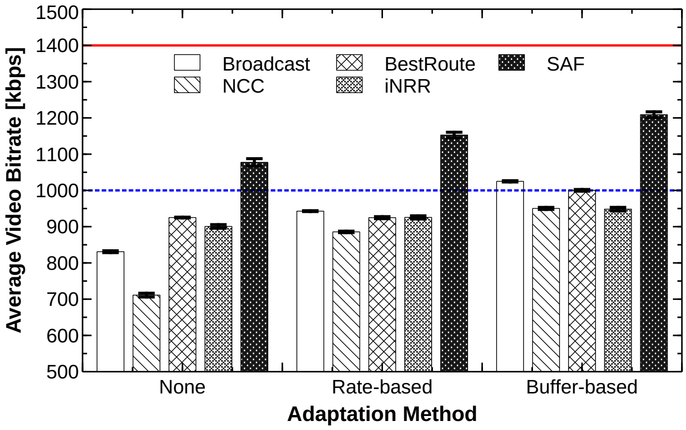
For the baseline evaluation of MPEG-DASH in a TCP/IP scenario, we obtain the following results for the average streaming bitrates. Considering simultaneous starting times of clients, we obtain an average streaming bitrate of 416 kbps and ± 0.382 kbps for the 95% confidence interval (CI). In the case of exponentially distributed starting times we obtain an average streaming bitrate of 423.441 kbps ± 0.736 kbps (95% CI). As expected, the performance is low. This is due to the fact that only single paths can be used by DASH and no caching of content takes place at network nodes.

Figures 3.7, 3.8, and 3.9 depict the 95% CI of the average video bitrates achieved by clients considering the combination of the different forwarding strategies, adaptation logics, caching strategies and starting times of the clients in NDN. The dashed (blue) line in the figures indicates the theoretical upper bound for the average video bitrate when solving the MCFP without consideration of caching as introduced in Subsection 3.2.2.a. The solid (red) line indicates the theoretical upper bound for the average video bitrate that is obtained by using Algorithm 3.1 assuming idealized caching as introduced in Subsection 3.2.2.b with unlimited cache sizes. In order to account for segments that are not retrieved in time causing stalls (the segment is not available until its associated playback timestamp), we penalize the average video bitrate by counting a zero bitrate segment in lieu thereof.

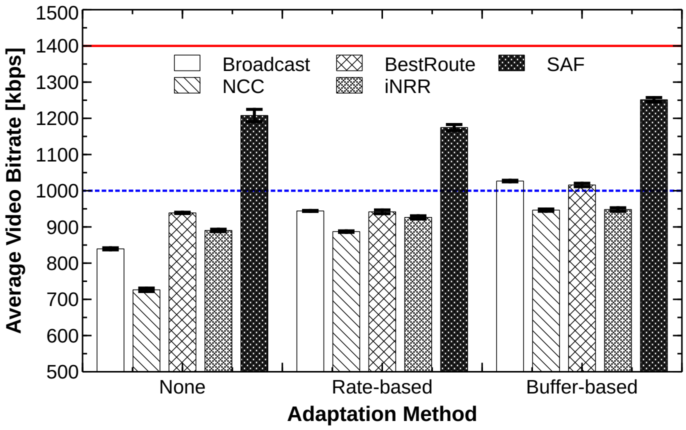
Having the results of the baseline DASH evaluation in mind, it is evident that any combination of cache size, forwarding strategy and adaptation logic (even no adaptation logic) is able to obtain a higher average video bitrate in NDN. Based on the figures, we make the following observations. BestRoute, which strongly focuses on the single *best* delivery path, benefits from caching in contrast to the standard TCP/IP scenario with DASH. The other forwarding strategies (particularly SAF), which make extensive use of multi-path forwarding, obtain an extra performance boost. Also Broadcast performs quite well in this scenario (although it is a very simple and resource demanding strategy), especially when clients use a buffer-based adaptation strategy. Furthermore, we can observe that all strategies obtain better results with larger caches. So, increasing the cache size has a positive impact on the average obtained video bitrate by the clients. Assuming an exponential distribution of the starting times of the clients has a negative impact on the average video bitrate obtained for both caching policies CEE and probabilistic caching. The performance of probabilistic caching is worse than that of CEE, particularly with small cache sizes. Please note that we only present results for probabilistic caching with parameter $p = 0.6$



(a) **25 MB** cache size per node

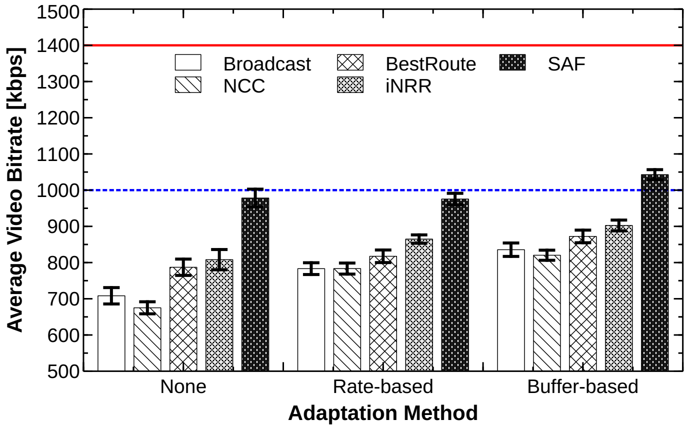


(b) **50 MB** cache size per node

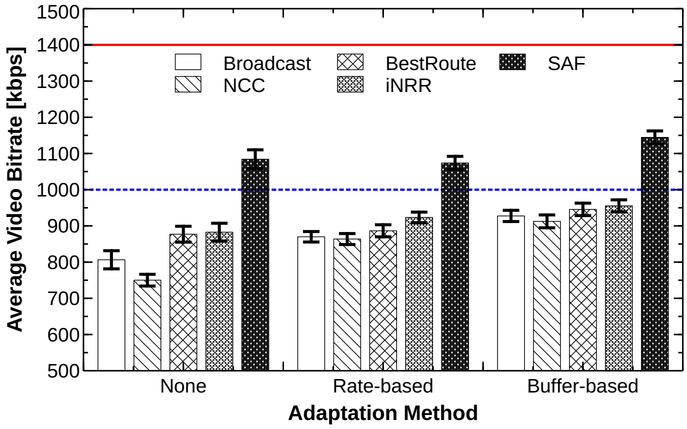


(c) **100 MB** cache size per node

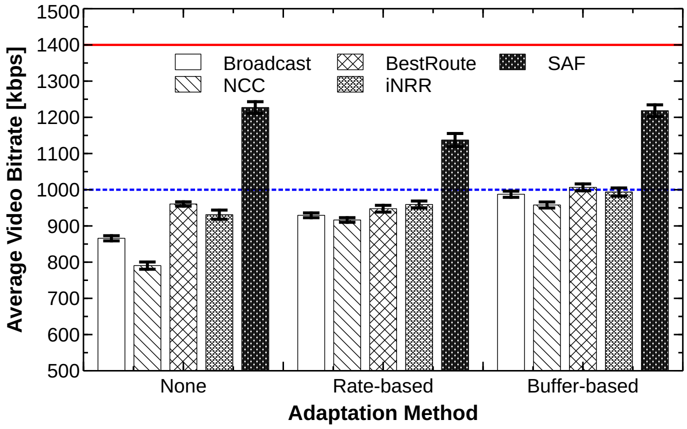
Figure 3.7: Average achieved **video bitrate** by the clients with **simultaneous start times** using a **CEE caching** policy.



(a) **25 MB** cache size per node

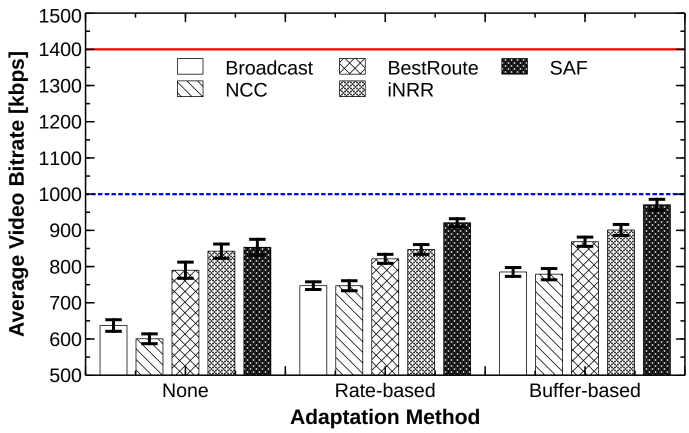


(b) **50 MB** cache size per node

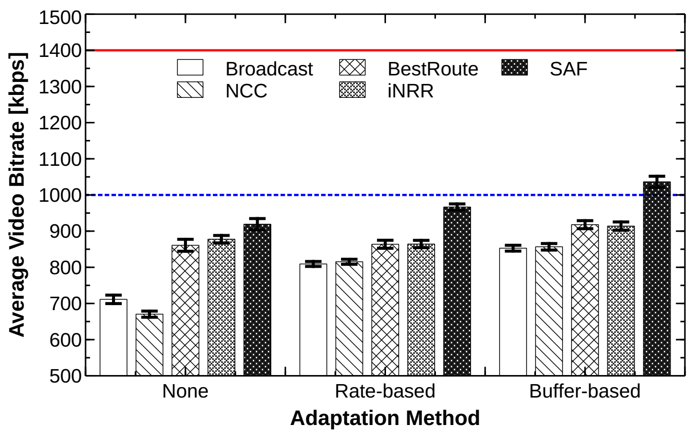


(c) **100 MB** cache size per node

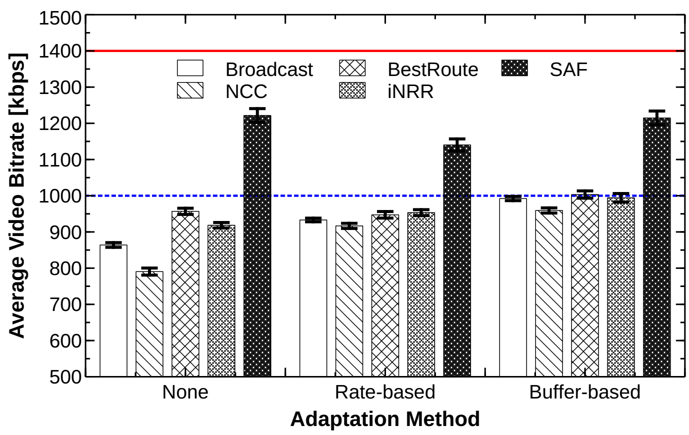
Figure 3.8: Average achieved **video bitrate** by the clients with **exponentially distributed start times** using a **CEE caching** policy.



(a) **25 MB** cache size per node



(b) **50 MB** cache size per node



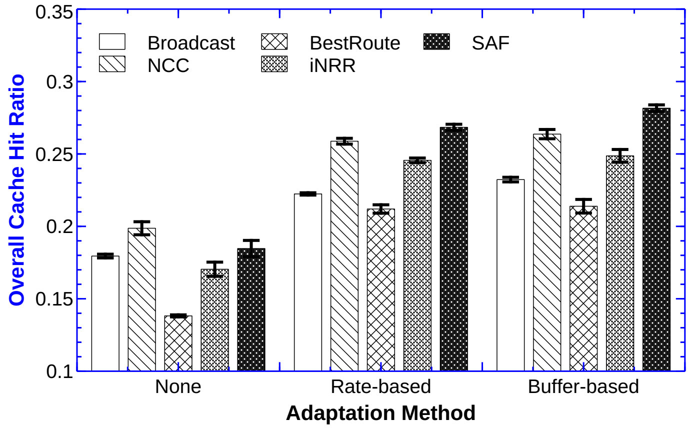
(c) **100 MB** cache size per node

Figure 3.9: Average achieved **video bitrate** by the clients with **exponentially distributed start times** using a **probabilistic caching** policy ($p = 0.6$).

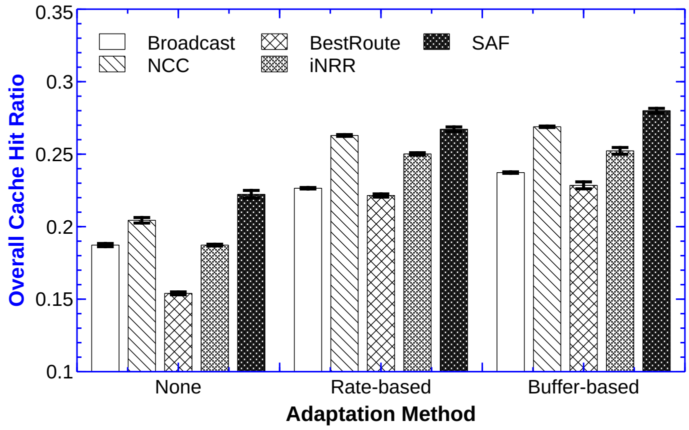
because lower values provide even lower cache hit ratios. This is due to the selected topology (cf. Figure 3.6) [90]. We further observe that the buffer-based adaptation logic obtains a higher average video bitrate compared to results of the rate-based adaptation logic. When we distribute the starting times of clients exponentially, we observe that SAF achieves the highest average video bitrate even without any adaptation logic when having bigger cache sizes (e.g., 50 MB or 100 MB). This is caused by the fact that SAF tries to maximize the throughput and in this case the retrieved representation of the multimedia content is not restricted by the adaptation logic.

These findings are affirmed by Figures 3.10, 3.11, and 3.12 which depict the 95% CI of the overall cache hit ratio with respect to the combination of forwarding strategies, caching strategies and starting times of the clients. It is evident from the figures that the larger the cache size, the higher the overall cache hit ratio. However, also the selected adaptation method and forwarding strategy have significant influence on the cache hit ratio. The buffer-based adaptation is able to obtain a higher cache hit ratio than the rate-based adaptation logic. Having no adaptation mechanism affects the cache hit ratio negatively leading to the worst results concerning this metric. Having a closer look at the influence of the forwarding strategies, it can be seen that particularly the strategie's NCC and iNRR maintain the highest cache hit ratios if cache sizes are low (e.g., 25 MB). This can be explained by the strategies design. NCC focuses on low-delay paths and therefore prefers nearby copies (nearby copies can be delivered faster), while iNRR uses an oracle to determine the nearest replica. However, SAF catches up when the cache size increases and is able to achieve the highest cache hit ratio among all forwarding strategies when the cache size approaches 100 MB.

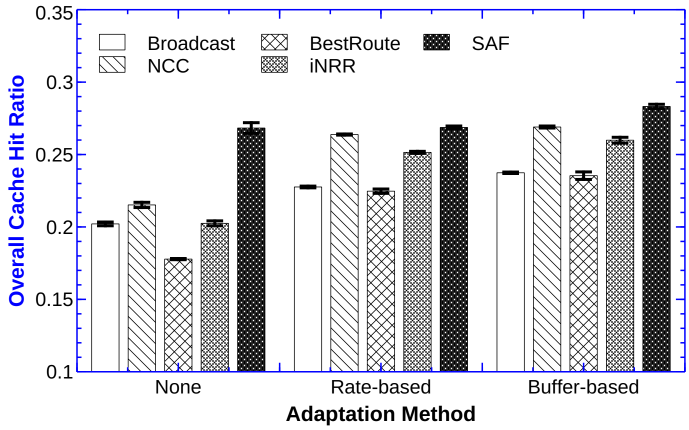
Still, the average video bitrate and cache hits do not tell the whole story. To further assess the performance of the adaptation logics in this NDN-based streaming scenario, we have a look at the clients' switching frequencies among the available representations and their playback stabilities with respect to the representations. Therefore, we study their behaviour in the case of letting the clients start streaming simultaneously, having CEE as the caching strategy and a cache size of 50 MB. The results are very similar for the other parameter settings (and are therefore not discussed in detail). Figures 3.13, 3.14, and 3.15 depict the number of clients that are able to retrieve a certain quality of a segment under different forwarding strategies and adaptation mechanisms for the mentioned settings, respectively. The x-axis denotes the segment numbers (which have a duration of two seconds)



(a) **25 MB** cache size per node

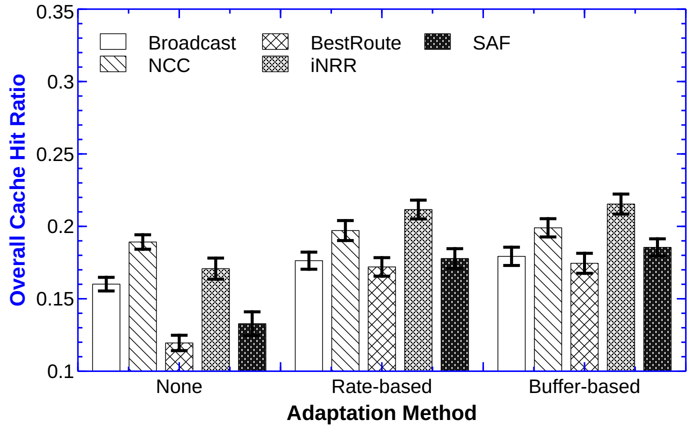


(b) **50 MB** cache size per node

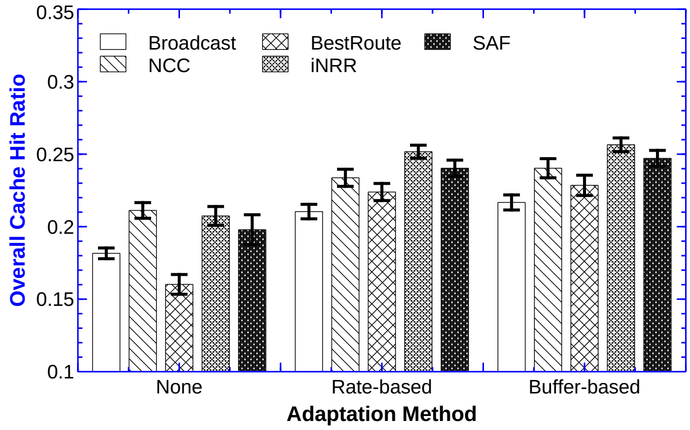


(c) **100 MB** cache size per node

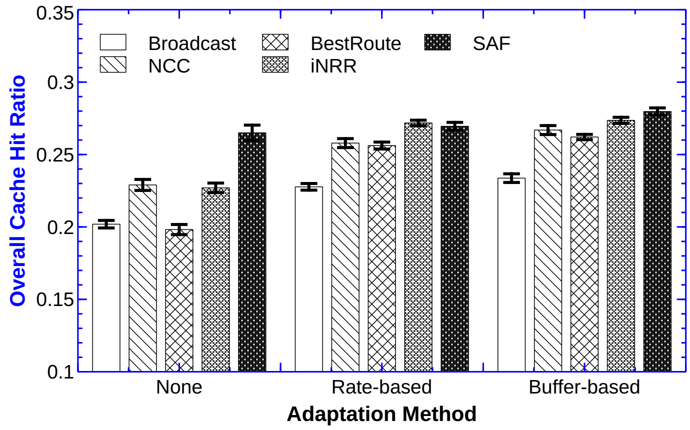
Figure 3.10: Average achieved **cache hit ratio** per node with **simultaneous start times** using a **CEE caching** policy.



(a) **25 MB** cache size per node

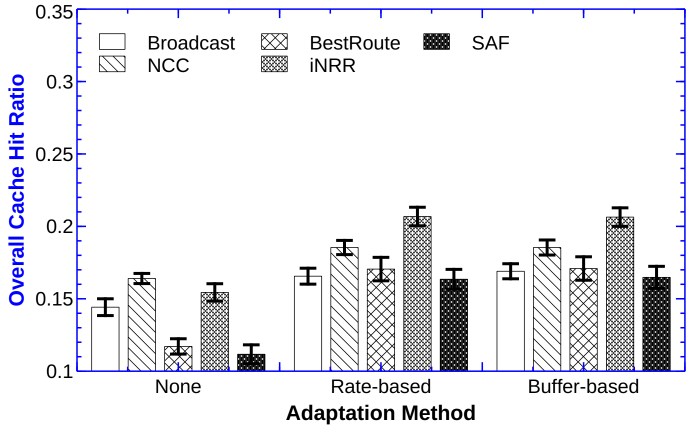


(b) **50 MB** cache size per node

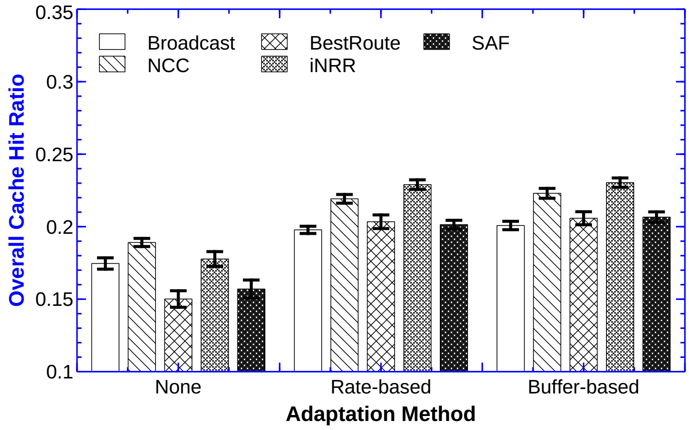


(c) **100 MB** cache size per node

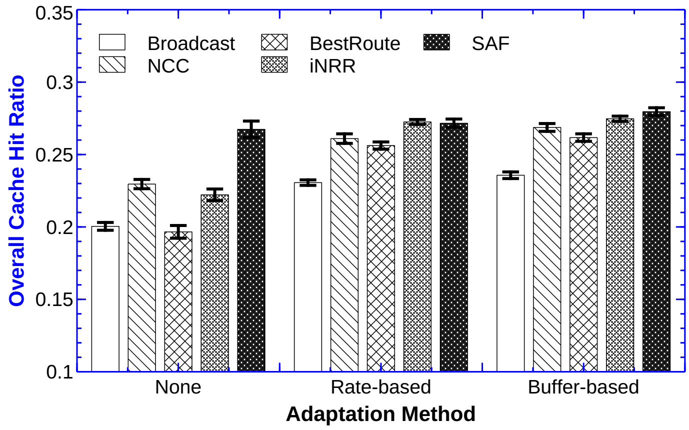
Figure 3.11: Average achieved **cache hit ratio** per node with **exponentially distributed start times** using a **CEE caching** policy.



(a) 25 MB cache size per node



(b) 50 MB cache size per node



(c) 100 MB cache size per node

Figure 3.12: Average achieved **cache hit ratio** per node with **exponentially distributed start times** using **probabilistic caching** ($p = 0.6$).

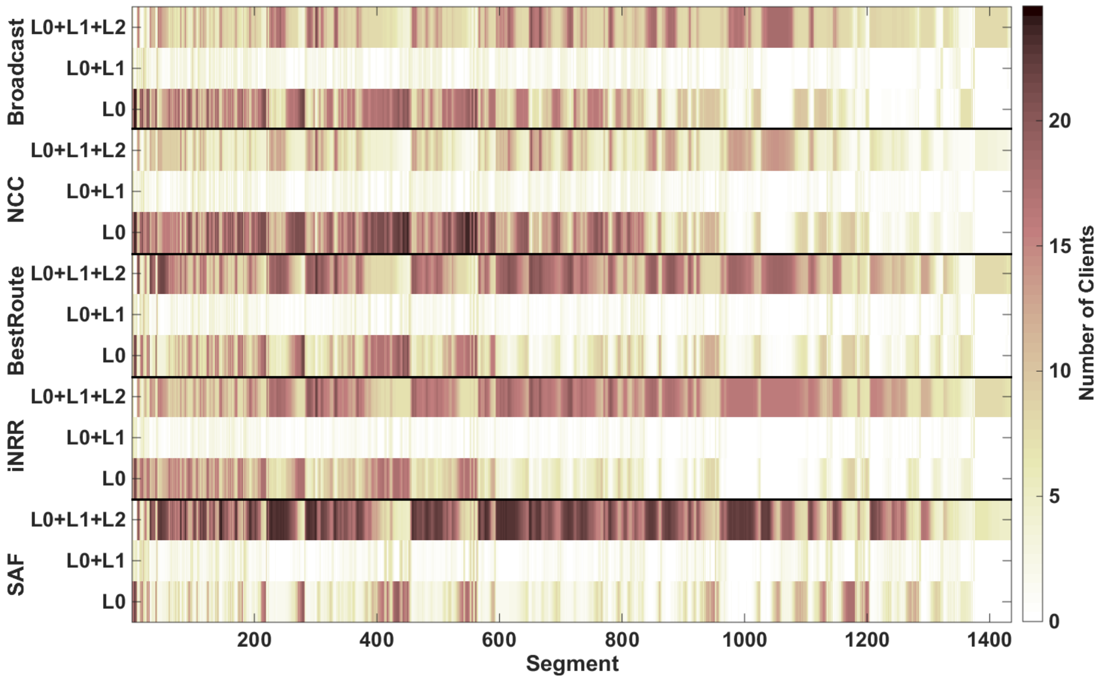


Figure 3.13: Number of clients that retrieve a given segment with a certain quality for playback under different forwarding strategies using **no adaptation**.

and the y-axis denotes the representations (layers) for every forwarding strategy. The figure depicts the number of clients receiving the different representations (layers) over time. The optimal case would occur if the row of $L0 + L1 + L2$ were black, and all others were white. This would indicate that all clients have got the highest available representation for all 1438 segments.

Figure 3.13 depicts the case where no adaptation strategy has been used. The figure clearly shows that the clients suffer from stalls if no adaptation algorithm is employed regardless of the forwarding strategy, indicated by the bright areas for the last segments. The bright area indicates that many of the clients are not able to retrieve these segments during the simulation time (48 minutes, cf. Subsection 3.2.3.a) due to previous playback interruptions (stalls). The forwarding strategy SAF clearly outperforms the other strategies as more clients are receiving a high quality layer (e.g., $L0 + L1 + L2$), followed by BestRoute and iNNR, which lie close together.

Figure 3.14 depicts the same case using a rate-based adaptation algorithm. The rate-based adaptation mechanism enables the clients to receive more segments during the simulation time for playback, so fewer playback stalls (indicated by darker tails) are encountered by the clients. This is due to the fact that fewer clients receive the best quality $(L0 + L1 + L2)$.

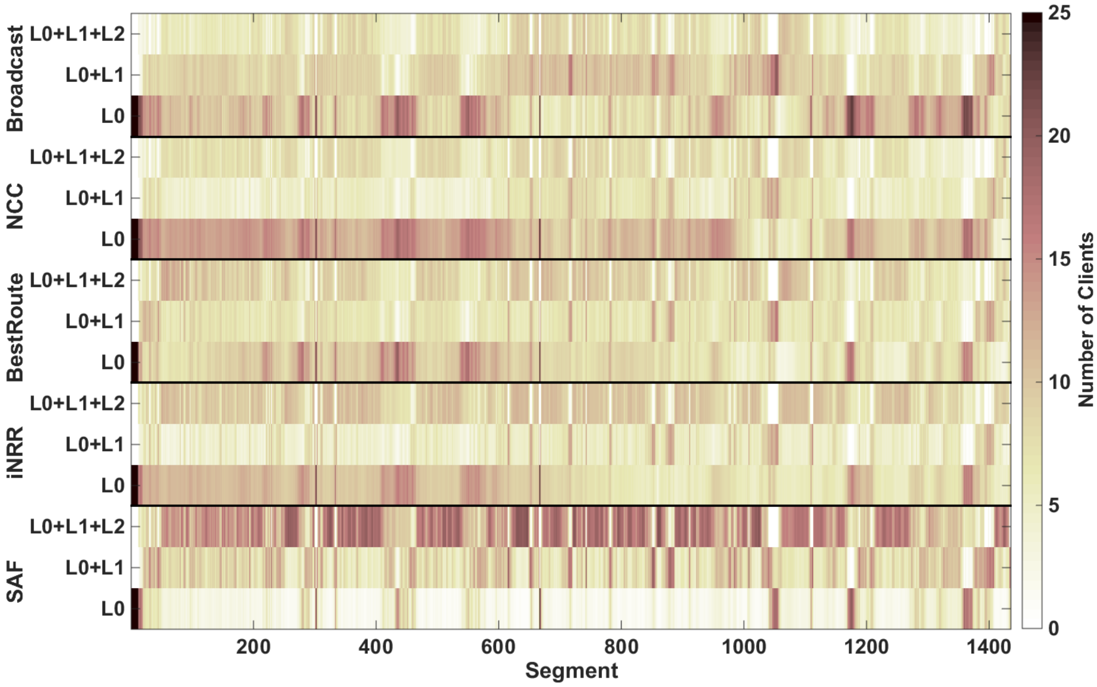


Figure 3.14: Number of clients that retrieve a given segment with a certain quality for playback under different forwarding strategies using **rate-based adaptation**.

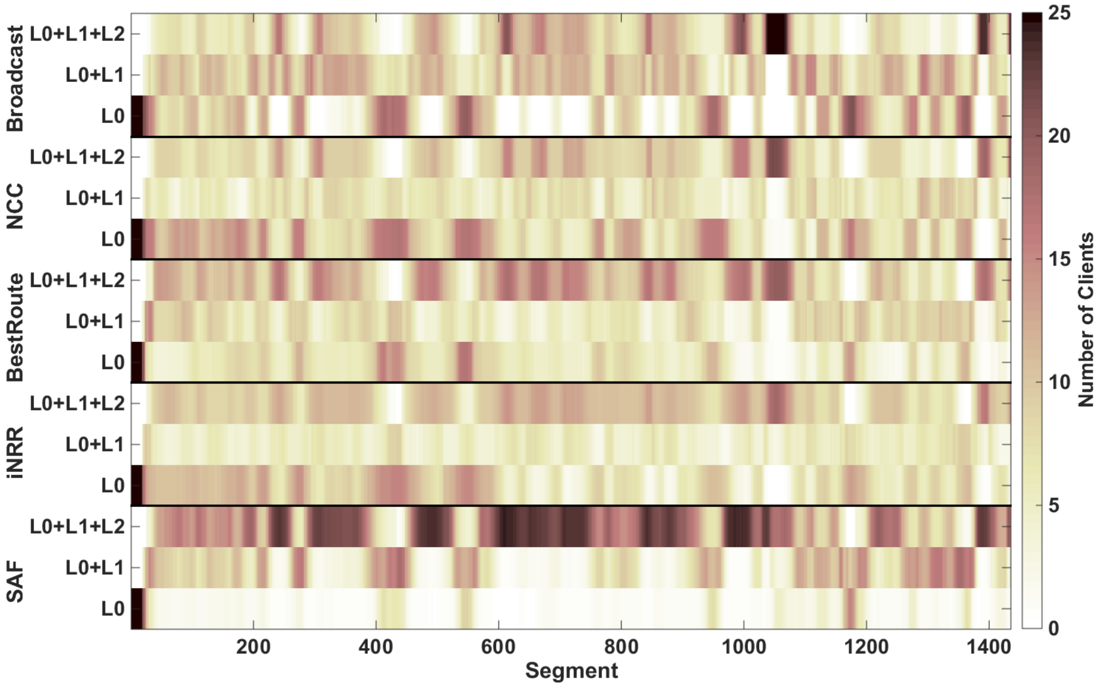


Figure 3.15: Number of clients obtaining a given segment with a certain quality for playback under different forwarding strategies using **buffer-based adaptation**.

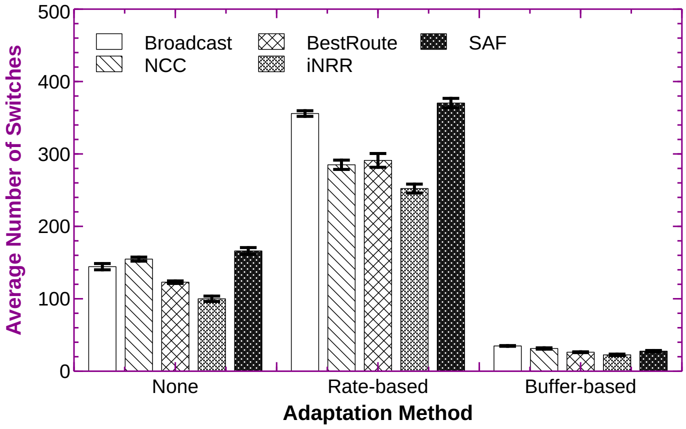
The available bandwidth is distributed more equally among the clients. Comparing Figure 3.13 and Figure 3.14 one observes that the latter shows more fine-grained variation patterns. This indicates that the clients in Figure 3.14 suffer from higher representation switching frequencies (will be discussed in detail, cf. Figures 3.16, 3.17, and 3.18).

Figure 3.15 depicts the received quality when the clients use a buffer-based adaptation mechanism. The first thing that attracts the attention is that the buffer-based adaptation provides a more stable quality to the clients, indicated by the very homogeneous colored areas (compared to no adaptation and rate-based adaptation). Furthermore, all forwarding strategies are able to provide a better quality to the clients compared to the rate-based and no adaptation approaches.

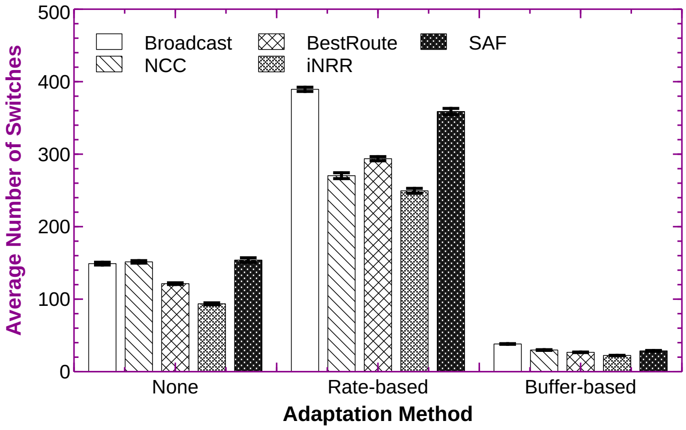
Figures 3.16, 3.17, and 3.18 depict the 95% CI of the average number of representation switches per client for the given parameter settings. Comparing Figures 3.16, 3.17, and 3.18 to Figures 3.13, 3.14, and 3.15, it follows that rate-based adaptation causes the clients to heavily oscillate between different representations. This has two reasons. First, NDN's multi-path transmission does not allow an accurate estimate of the available bandwidth. Second, if Interests cause a cache *hit* in a cache close to the client, the rate-based adaptation mechanism reacts and overestimates the bandwidth when requesting the next segment (most likely from a higher representation). However, the higher representation is not necessarily available in the nearby caches. This may lead to cache *misses* and a low download bitrate because only the previously requested representation is cached. Taking a look at the cache hit ratios (cf. Figures 3.10, 3.11, and 3.12) we see that with higher cache hit ratios the number of switches increases in the case of the rate-based adaptation logic. Thus, we can conclude that the oscillation effect caused by a rate-based adaptation logic is amplified if the cache size is increased. The results show that oscillation can be easily avoided by using a buffer-based adaptation logic instead of a rate-based one.

3.3 Conclusion and Further Challenges for ICN-based Content Delivery

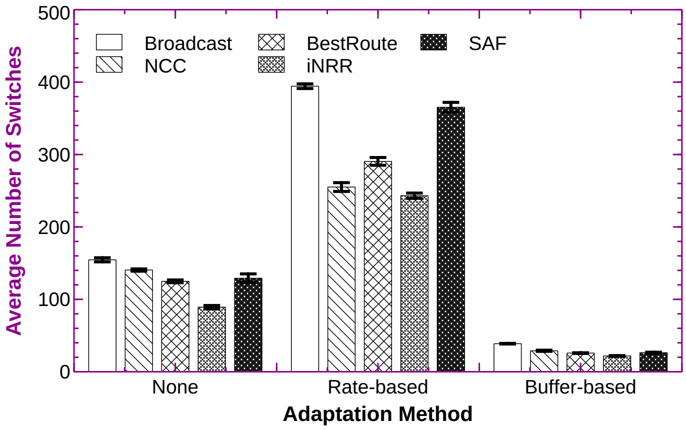
The objective of this chapter was to investigate the multimedia streaming performance in ICN/NDN relying on the principles of DAS. Therefore, in the first part of this chapter we discussed the potential beneficial mechanisms that are used to promote ICN as the future technology for effective (multimedia) content distribution. The presented simplified



(a) **25 MB** cache size per node

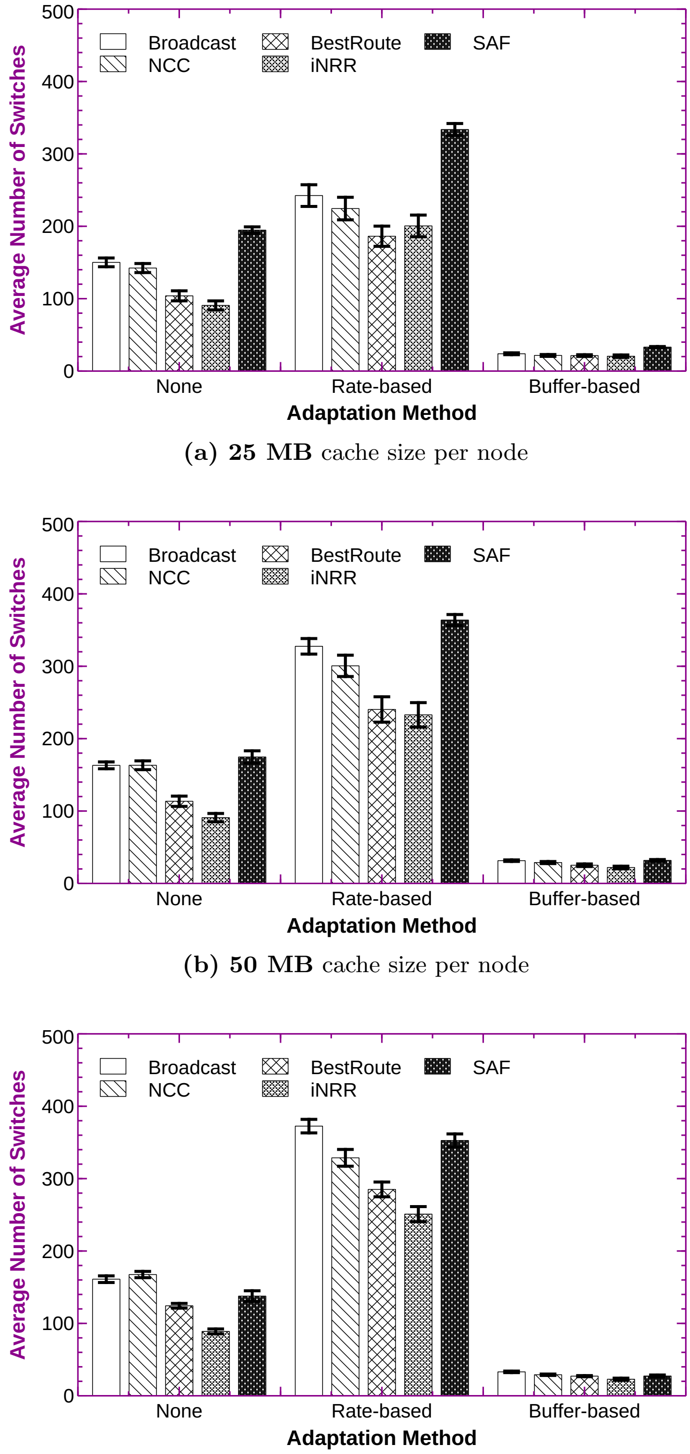


(b) **50 MB** cache size per node



(c) **100 MB** cache size per node

Figure 3.16: Average number of representation switches per client with **simultaneous start times** using a **CEE caching** policy.

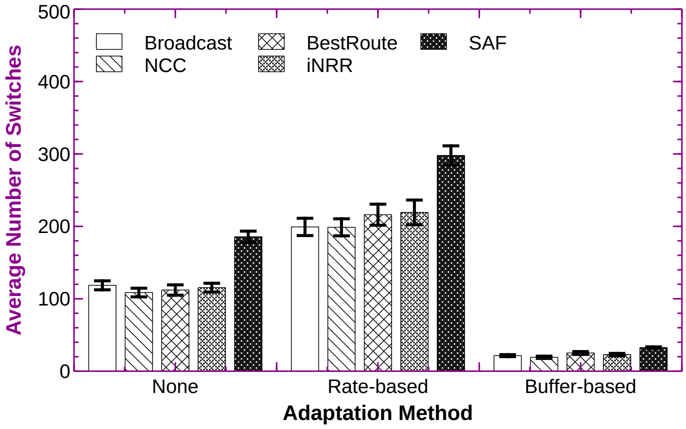


(a) 25 MB cache size per node

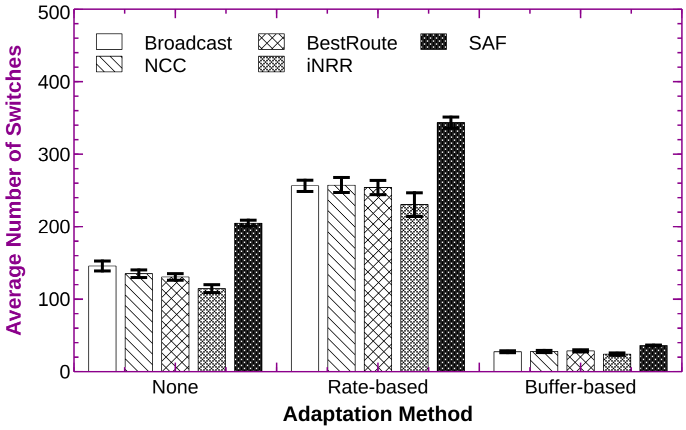
(b) 50 MB cache size per node

(c) 100 MB cache size per node

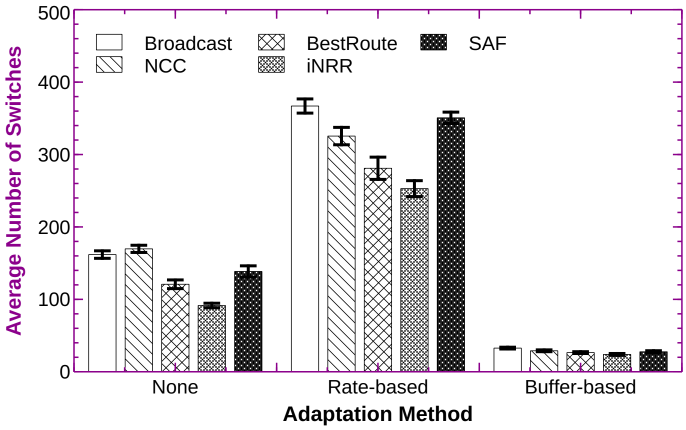
Figure 3.17: Average **number of representation switches** per client with **exponentially distributed start times** using a **CEE caching** policy.



(a) 25 MB cache size per node



(b) 50 MB cache size per node



(c) 100 MB cache size per node

Figure 3.18: Average **number of representation switches** per client with **exponentially distributed start times** using **probabilistic caching** ($p = 0.6$).

scenarios indicated the potential benefits of network-inherent caching, adaptive multi-path forwarding and context- and content-aware data delivery. In the second part of this chapter we have taken NDN as an ICN representative and investigated the actual achievable performance. Therefore, we first developed an MCFP that provides the theoretical upper bounds for multi-path multimedia transmission in NDN. The MCFP is capable of modelling DAS-based content delivery in NDN without and with idealized caching. The derived bounds *do not consider protocol overhead* introduced by NDN; thus, they are *purely theoretical*. Nevertheless, the bound obtained when solving the MCFP given by LP 3.1 also provides an upper bound for traditional IP-based networks using a multi-path enabled transmission protocol (e.g., MPTCP) without considering proxies acting as caches. In Subsection 3.2.3.b we showed that today's most prominent streaming technology MPEG-DASH over TCP/IP is far away from the optimum without caching, which is definitely due to a lack of multi-path support in IP networks. Considering NDN's inherent multi-path and caching capabilities, we had assumed that it would easily outperform DASH in the TCP/IP-based scenario and exceed the first theoretical bound that does not consider caching. The results clearly show that NDN-based DAS is definitely more effective than TCP/IP-based DASH, in particular if an appropriate forwarding strategy and sufficiently large caches are employed. However, NDN-based DAS is barely able to reach the first upper bound that does only consider multi-path transport without caching, especially when considering the proposed forwarding strategy SAF as out of competition for assessing the state-of-the-art performance of NDN (cf. Research Objective (3), Section 1.2). So, there is still a significant gap between the second theoretical bound that considers *idealized* caching and the results that can be reached practically using NDN.

Considering the fact that ICN caching strategies have been extensively researched (see Subsection 3.1.1), and also the variations of caching strategies (CEE, probabilistic caching with $p = \{0.1, 0.3, 0.6\}$) during the evaluation did not significantly improve the overall performance (although with iNRR we had a forwarding strategy that perfectly obtains cached replicas), we conclude that the performance gap has its basic roots in the existing forwarding strategies of exploiting all available paths. Additionally to this hypothesis, we believe that taking advantage of NDN's context and content awareness in the forwarding plane can provide significantly better results. In the following two chapters we are going to discuss these two hypotheses in detail and assess their validity. In Chapter 4 we design and implement SAF, a novel forwarding strategy that is able to consider context and content information in the forwarding plane. While Chapter 4 focuses on the overall design and performance challenges of the strategy, Chapter 5 is dedicated to exploiting context and content awareness within the forwarding plane.

Another important result of this chapter is that the evaluations showed that buffer-based adaptation mechanisms should be preferred in ICN/NDN. Since multi-path transmissions and network-inherent caching do not allow for an accurate estimation of the available bandwidth, rate-based adaptation logics should not be used. DAS clients in NDN will suffer from the same oscillation behaviour as in traditional IP-based networks using HTTP proxies as caches, if rate-based adaptation is employed [113]. This oscillation behaviour is amplified by increasing cache sizes (cf. Figures 3.16, 3.17 and 3.18). Grandl et al. [114] provide a detailed discussion of this topic.

CHAPTER 4

Stochastic Adaptive Forwarding in NDN

"One cannot really argue with a mathematical theorem."

— Stephen Hawking, 1942*

Forwarding decisions in classical IP-based networks are predetermined by routing. This is necessary to avoid loops, inhibiting opportunities to implement an adaptive and intelligent forwarding plane. Consequently, content distribution efficiency is reduced due to a lack of inherent multi-path transmission. In Named Data Networking (NDN) instead, routing shall hold a supporting role to forwarding, providing sufficient potential to enhance content dissemination at the forwarding plane. In this chapter we design, implement, and evaluate a novel probability-based forwarding strategy, called Stochastic Adaptive Forwarding (SAF) for NDN. SAF imitates a self-adjusting water pipe system, intelligently guiding and distributing Interests through network crossings circumventing link failures and bottlenecks. Just as real pipe systems, SAF employs overpressure valves enabling congested nodes to lower pressure autonomously. Through an implicit feedback mechanism it is ensured that the fraction of the traffic forwarded via congested nodes decreases. By conducting simulations we show that our approach outperforms existing forwarding strategies in terms of the Interest satisfaction ratio in the majority of the evaluated scenarios. This is achieved by extensive utilization of NDN's multi-path and content-lookup capabilities without relying on the routing plane. SAF explores the local environment by redirecting requests that are likely to be dropped anyway. This enables SAF to identify new paths to the content origin or to cached replicas, circumventing link failures and resource shortages without relying on routing updates.

This chapter is structured into four sections. Section 4.1 provides an introduction to the topic of forwarding. It discuses the different roles of forwarding and routing in IP-based and NDN-based networks. Once the specific tasks of forwarding are identified, we use them to define the design goals for SAF. Furthermore, Section 4.1 includes an extensive discussion of existing competitors (forwarding strategies) and their employed techniques. We use this discussion to emphasize the differences between SAF and its competitors.

Furthermore, we may identify important concepts that can be considered for the design of SAF. Section 4.2 introduces the principles of SAF. We present the content, network and node models the design rests on. Furthermore, we prove the correctness of the employed approach. Section 4.3 presents an extensive evaluation comparing the performance of SAF to that of its competitors. We investigate two scenarios modelling a content delivery scenario under uniform and Zipf-like content popularity. Therefore, we generate multiple network topologies based on the Barabási-Albert model [115] including the considerations of topology changes over time (e.g., link failures). Finally, Section 4.4 concludes the findings of this chapter.

4.1 Introduction

This section provides *i*) a discussion of the roles of forwarding and routing in NDN and IP, with a focus on their clear separation in NDN (cf. Subsection 4.1.1); *ii*) the basic design principles for SAF and a short outline that presents the basic ideas of the algorithm (cf. Subsection 4.1.2); and *iii*) a detailed discussion of existing forwarding strategies with a focus on their basic principles and differences to SAF (cf. Subsection 4.1.3).

4.1.1 Clear Separation of Forwarding and Routing

Today's Internet is based on a legacy host-based architecture, resulting in limitations (cf. Subsection 2.1.1). One key issue is forwarding, which is strictly predetermined by routing to ensure a loop-free packet transmission. There are few opportunities to implement an adaptive forwarding plane in classical IP-based networks since routing dictates the forwarding options. In IP, forwarding planes are stateless and routing protocols are responsible to deal with all kinds of short- and long-term topology changes. Routing protocols struggle with all the imposed responsibility. For instance, Labovitz et al. [116, 117] showed that BGP convergence times up to 15 minutes recovering from a single multi-homed fault (switching to an alternative, redundant route) are possible, if no additional mechanisms (e.g., backup configurations providing alternative routes [118]) are used.

The concept of NDN instead rests on a content-centric communication model, where content only is addressed. Using the three previously introduced data structures CS, PIT and FIB (cf. Subsection 2.2.3) NDN provides a stateful forwarding plane that is capable of performing adaptive multi-path forwarding. This is possible in contrast to IP, because NDN is able to identify duplicate or looping Interests packets. In order to detect these packets, a NONCE (unique bit pattern) is added to each Interest. Loop prevention is achieved by matching name and NONCE of received Interests to those already in the PIT. This enables an adaptive forwarding plane with inherent multi-path delivery, in contrast to routing in IP that has to avoid potential loops in the first place impeding content distribution efficiency.

It has been shown by Yi et al. [47] that NDN's stateful forwarding plane handles typical network issues, such as short-term link failures and congestion, more effectively than IP networks. Furthermore, in [48] Yi et al. argue that *routing* in NDN *shall hold a supporting role to forwarding*. Routing should only provide a reasonable starting point for the forwarding plane, which then should explore different multi-path opportunities. In return, adaptive forwarding enables a more scalable routing plane with relaxed requirements in terms of convergence time and completeness. For this chapter our understanding of routing, forwarding and caching and their clear separation is in accordance with Yi et al. [47] and will drive our design goals for SAF. Please note that the previous assumption does not restrict the coupling of these mechanisms as suggested by other work (e.g., by [107] and [119]), but clearly separates their areas of responsibility. The existing adaptive forwarding strategies (cf. Subsection 4.1.3) in NDN do not provide all mechanisms foreseen in [47] and [48], which can also be one explanation for the only moderate performance of NDN-based DAS in Section 3.2.3 in comparison to the theoretical upper bound considering network-inherent caching. For this reason, we propose *Stochastic Adaptive Forwarding* (SAF), a probability-based forwarding strategy. As application field for this strategy we envisage the infrastructure of Internet Service Providers (ISPs) interconnecting several autonomous systems and/or access networks. Furthermore, since today's networks have to support applications with varying demands (cf. Subsection 2.3.1), SAF should be able to provide content and context-aware forwarding for the individual applications/contents (e.g., Interests requesting delay sensitive services/contents should be preferably forwarded on low latency links). This brings us to the basic design principles of this strategy.

4.1.2 SAF – Design Principles

This strategy has been designed to meet the following objectives:

- Perform stochastic adaptive forwarding on a per-content/per-prefix basis.
- Provide effective forwarding even with incomplete or partly invalid routing information

as suggested by the clear separation of routing and forwarding [48].

- Deal with unexpected network topology changes, e.g., link failures, without relying on the routing plane.
- Discover unknown paths to cached replicas.
- Enable content- and context-aware considerations.

While these attributes are beneficial for the rather versatile edge networks, they are not suitable for the rigid Internet core. Here, simpler approaches may perform better considering the well known scalability principle for networks: "*complexity at the edge, aggregation at the core*" [120].

SAF accomplishes the previously mentioned objectives by imitating a water pipe system. Network nodes act as crossings for an incoming flow of water (Interests). Returning Data packets act as input for a probability distribution which determines the share of the flow that is forwarded via the different pipes (outgoing interfaces). Each crossing maintains an overpressure valve. If the pressure on a node increases, e.g., due to network congestion, it may use the overpressure valve to lower the pressure autonomously. The Interests that pass the overpressure valve can be used for different purposes. For instance, they can be used as scouts to investigate unknown paths to complement routing information. However, in some cases it is best to simply discard these Interests. The discarded and therefore unsatisfied Interests provide implicit feedback to the requesting nodes, indicating that they should reduce the number of requests forwarded to the considered node.

SAF is based on an measure that defines the target of the adaptive forwarding. The objective is to maximize the Interest/Data satisfaction ratio with (optional) respect to delay, hop-count, and/or transmission cost considerations. Therefore, SAF does not depend on a concrete measure, but provides flexibility for individual scenarios. Based on the requested content, the specific service and/or the network operator's ambitions, this measure has to be chosen carefully. It determines the preferred paths for forwarding. However, adaptivity provided by SAF is not only achieved by intelligent multi-path transmission using redundant paths. SAF is also able to exploit content-based information to further improve the forwarding decisions. For instance, this is beneficial for multimedia scenarios where the relative priority of packets (e.g., VoIP vs. file transfer) is more important than a purely throughput- and/or delay-based metric may indicate (cf. Chapter 5).

4.1.3 State of the Art Forwarding Strategies

As this chapter focuses on forwarding in NDN, we first discuss the proposed forwarding strategies introduced by the NDN community. Yi et al. [45, 47] classify (inter-)faces based on a simple color scheme. Faces can be marked as Green, Yellow and Red, which corresponds to the meaning that faces return data (Green), may or may not return data (Yellow), or they do not work at all (Red). Within this classification, faces are ranked, e.g., based on the delay of receiving Data packets. This basic scheme is used by all proposed forwarding strategies that are provided within version 1.0 of the ns3/ndnSIM simulator [121]:

- Flooding: Interests are forwarded to all Green and Yellow faces supplied by the FIB. Note that this implies that on every traversed node the Interest may get replicated several times (depends on the number of matching FIB entries).
- SmartFlooding: Interests are forwarded to the highest-ranked Green face. If no Green face is available, an Interest is forwarded via all Yellow faces.
- BestRoute(1): Interests are forwarded to the highest-ranked Green face. If no Green face is available, an Interest is forwarded via the highest-ranked Yellow face.

Furthermore, forwarding strategies in ns3/ndnSIM v1.0 can be supplemented by several enhancements [121]. One example are *Interest Limits*, which conceptually are Token Bucket filters [122]. For instance, if the highest ranked interface reaches its transmission limit, an *Interest Limit* ensures that another face is selected for forwarding further requests. Further supplements are negative acknowledgement messages (NACKs), which can be returned to an Interest issuer to provide immediate feedback if a request cannot be satisfied (e.g., due to the imposed Interest Limits).

Recently ns3/ndnSIM v2.0 [37] was released. The newer version no more re-implements basic NDN primitives, such as forwarding, but uses code from the *NDN Forwarding Daemon* (NFD) [38]. This allows near-realistic simulations since the code-base of the NFD is used, which has been developed for physical hardware. The step towards more realistic simulations resulted in major changes in the simulator also affecting the implemented forwarding strategies. In ns3/ndnSIM v2.0 strategies no longer rest on the aforementioned color scheme and cannot take advantage of additional features such as *Interest Limits* or NACKs. We shortly outline the forwarding strategies available in the NFD [38]:

- Broadcast: Interests are forwarded to all faces supplied by the FIB (Since the latest patch for ndnSIM v2.1 was published, this strategy is also known as *Multicast Strategy*).
- BestRoute(2): Interests are forwarded to the lowest-cost (e.g., in terms of hop count) upstream face indicated by routing. Actually, there is a name collision between this strategy and the BestRoute(1) provided by [121]. As we are going to consider only this strategy as competitor to SAF (will be justified later on), we use the name BestRoute for BestRoute(2) in the remainder of this thesis.
- NCC: Interests are forwarded to those faces that provide Data packets with the lowest delay. NCC is not an acronym. Its name was derived from flipping the initials of the term Content-Centric Networking (CCN) [34]. NCC was the default forwarding strategy implemented in PARC's CCNx (v0.7.2) and has been ported to the NFD.

As the aforementioned strategies are available in the ns3/ndnSIM simulator, they can easily be used for comparison with new approaches. Since we are focusing on a realistic approach with SAF, we regard ns3/ndnSIM v.2.0 as the platform most suitable for evaluations and performance measurements. For this reason, we consider the algorithms Broadcast, BestRoute, and NCC for comparison to SAF, and do not take into account their predecessors, Flooding, SmartFlooding, and BestRoute available in ndnSIM v.1.0.

In [107] Rossini and Rossi propose [ideal] Nearest Replica Routing ([i]NRR), an approach to couple caching and forwarding extending aNET [123]. The iterative algorithm provided in [107] makes use of an oracle providing information on the availability of content in all caches in the network. The strategy selects the face with the shortest distance to the content, which prefers nearby caches rather than forwarding the Interest towards the content origin. iNRR is implemented in the ccnSIM simulator [124]. Although the authors indicate that an implementation of a perfect oracle is not feasible in a real environment, we consider iNRR with perfect knowledge about the individual content chunks in the caches as a competitor to SAF. Please note that in [107] some practical approaches using off-path exploration to assess the necessary information provided by the oracle for the concept of NRR are proposed.

Chiocchetti et al. [125] developed INFORM, which is an adaptive hop-by-hop forwarding strategy using reinforcement learning inspired by the Q-routing framework. INFORM is able to discover temporary copies of content not present in the routing table, thus increasing the effectiveness of forwarding. The authors have implemented and evaluated INFORM within ccnSIM [124]. Unfortunately, they indicate on their website [1] that due to a lack of manpower no release candidate or source code package for this strategy can be provided. However, the authors also indicate that one should prefer iNRR over INFORM for comparison, since it outperforms INFORM. Since iNRR is already on our list as a competitor to SAF, we do not consider INFORM as a competitor.

In [95] Carofiglio et al. derive a set of optimal dynamic multi-path congestion control protocols and request forwarding strategies from a multi-commodity flow problem. The proposed Request Forwarding Algorithm (RFA) in [95] is outlined in detail and is a good candidate for comparison. The idea of the algorithm is simple yet effective. For each content-prefix and for each face, RFA monitors the PIT entries. The forwarding probability of a face is then determined by a weight, which is actually a moving average over the reciprocal count of the PIT entries. We re-implemented this algorithm for ndnSIM v2.0 and for the NFD for comparison.

Qian et al. [126] proposed the concept of Probability-based Adaptive Forwarding. The basic idea is to select faces based on a probability distribution, which is also similar to our approach. However, noticeable differences are that [126] is inspired by ant colony optimization and focuses on delay minimization. As described later on in the thesis, SAF is generic providing opportunities to adapt forwarding with respect to further circumstances. We introduce a virtual face that enables content- and context-aware adaptation. Furthermore, we do not introduce distinguished Interest packets for probing only, which keeps our approach compatible to other forwarding strategies used in the network. Since we consider a possible interplay among the individual strategies as mandatory, we do not consider this approach as a competitor.

Yeh et al. [119] proposed VIP, a framework for *joint dynamic forwarding and caching* in NDN. In this system, *Virtual Interest Packets* (VIPs) capture the measured demand for respective data objects. The VIP count in a part of a network represents the local level of interest in a given object. The VIP framework employs a virtual control plane which operates on the VIPs. Distributed control algorithms are used to guide caching and forwarding strategies. We do not consider the VIP framework as a competitor due to: *i*) absence of a reference implementation; *ii*) difficulties to precisely reproduce the results: the illustration in [119] leaves room for interpretation regarding the individual parts (virtual control plane, communication protocols, control algorithms) and their interplay; *iii*) no

[1] `http://perso.telecom-paristech.fr/~drossi/index.php`, accessed 2016/07)

estimation regarding the communication overhead is provided.

Recently, Udugama et al. [127] published On-demand Multi-Path Interest Forwarding (OMP-IF). This forwarding strategy uses multiple node disjoint paths for Interest forwarding simultaneously. Each router may only use a single face (from the FIB) for forwarding per content-prefix to ensure node disjointness. The client is responsible for triggering the multi-path transmission by utilizing a weighted round-robin mechanism based on path delays distributing Interests over multiple faces. However, considering only node disjoint paths may leave some network resources unused. We consider OMP-IF as a competitor and implemented the strategy as presented in [127] for ndnSIM v2.0.

4.2 Implementing Stochastic Adaptive Forwarding

This section deals with the terminology and design of SAF. First, in Subsection 4.2.1 the network, content and node models are discussed, which state the necessary preconditions for the presented approach. Subsequently, we show that SAF enables adaptive forwarding based on a given measure. The design of $\mathcal{M}_T$, an exemplary measure maximizing the Interest satisfaction ratio, is illustrated in Subsection 4.2.2. Subsection 4.2.3 shows how SAF identifies unsatisfied traffic. SAF defines update operations that modify the forwarding probabilities for the individual (inter-)faces on a per-content basis to avoid unsatisfied traffic (cf. Subsection 4.2.4). The ultimate goal of SAF is to optimize a node's forwarding behavior such that it performs optimally in terms of a given measure. The design of these update operations is presented and exemplarily discussed based on $\mathcal{M}_T$ in Subsection 4.2.5. Furthermore, Subsection 4.2.6 discusses SAF's probing mechanism, which enables to identify paths to nearby content replicas unknown to the routing plane. Finally, two examples are presented in Subsection 4.2.7 illustrating the functionality of SAF.

4.2.1 Network, Content and Node Models

SAF rests on the following network model: $\mathcal{N}(\mathcal{V}, \mathcal{E})$ denotes a network consisting of a set of nodes $\mathcal{V}$ and a set of edges/links $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each node $v \in \mathcal{V}$ maintains a physical face $F_{(v,u)}$ (e.g., wireless network interface) for each tuple $(v, u) \in \mathcal{E}$. We define the list of physical faces on v as $\mathcal{F}'_v := \bigcup_{(v,u)\in\mathcal{E}} F_{(v,u)}$, where $|\mathcal{F}'_v|$ denotes the number of physical links/faces of v. A node may receive Interests on any $F_{in} \in \mathcal{F}'_v$ and tries to satisfy these requests by either returning a locally stored copy of the requested data or by forwarding the Interest to a suitable face $F_{out} \in \mathcal{F}'_v \setminus \{F_{in}\}$. The content catalogue in $\mathcal{N}$ is determined by a set $\mathcal{C}$. Each $c \in \mathcal{C}$ denotes content that can be retrieved using a common prefix.

In addition to the physical faces $\mathcal{F}'_v$, each node maintains a distinguished virtual face F_{D_v}, which acts as overpressure valve. The set $\mathcal{F}_v = \mathcal{F}'_v \cup \{F_{D_v}\}$ denotes the entire set of faces known to v. SAF is an algorithm local to each node, requiring no explicit communication between nodes. Therefore further discussion on SAF focuses only on a single node, which allows us to omit the subscripts v identifying a specific node for the remainder of this chapter. The virtual face F_D is treated as an ordinary face by SAF, however, any Interest forwarded to this face is discarded; the use of this *dropping face* will be made clear in the course of the section.

Every node maintains a so-called *Forwarding Table* (FWT). This table can be represented by a two-dimensional matrix, where the rows correspond to the set of faces $\mathcal{F}$ and the columns correspond to the different contents from the catalogue $\mathcal{C}$. The elements of the matrix indicate the confidence (probability) with which a certain outgoing physical face can provide data for a certain prefix. For instance, the following matrix represents an example FWT for a node with $\mathcal{F} = \{F_D, F_0, F_1, F_2\}$ and $\mathcal{C} = \{c_0, c_1, c_2\}$:

$$\text{FWT} = \begin{array}{c} \\ {\scriptstyle F_D} \\ {\scriptstyle F_0} \\ {\scriptstyle F_1} \\ {\scriptstyle F_2} \end{array} \begin{array}{c} \begin{array}{ccc} {\scriptstyle c_0} & {\scriptstyle c_1} & {\scriptstyle c_2} \end{array} \\ \begin{pmatrix} 0 & 0 & 1/3 \\ 1/3 & 1/2 & 0 \\ 2/3 & 0 & 2/3 \\ 0 & 1/2 & 0 \end{pmatrix} \end{array}$$

The FWT provides the probability of forwarding an Interest for content c_l on face F_i. We denote $p(F_i, c_l)$ as the forwarding probability that an Interest asking for c_l will be forwarded on F_i, for instance, $p(F_1, c_0) = \frac{2}{3}$. Note that, for any c_l, the corresponding column of the FWT specifies a discrete probability distribution: $\sum_{F_i \in \mathcal{F}} p(F_i, c_l) = 1$. The decision to forward a given Interest on a face is as simple as drawing a random number from a uniform distribution $U(0, 1 - p(F_{in}, c_l))$. Algorithm 4.1 sketches the face selection process, which is also known as *inverse transform sampling*. The function *nextDouble()* (cf. Alg. 4.1 line 2) draws a number from the distribution U and *pop()* (cf. Alg. 4.1 line 4) removes and returns the top element of a list (F_{list}). The algorithm requires $O(|\mathcal{F}|)$ steps. As the number of faces is usually constant on a node, the algorithm is actually in $O(1)$.

Algorithm 4.1 Select outgoing Face for an Interest

$F_{list} \leftarrow \mathcal{F} \setminus \{F_{in}, F_D\}$, $limit \leftarrow 0.0$
$rand \leftarrow U(0, 1 - p(F_{in}, c_l)).nextDouble()$
while $F_{list} \neq \emptyset$ **do**
  $F_{cur} \leftarrow F_{list}.pop()$
  $limit \leftarrow limit + p(F_{cur}, c_l)$
  if $rand \leq limit$ **then**
    return F_{cur}
  end if
end while
return F_D

Figure 4.1 illustrates an NDN node using SAF. The design of the FWT provides two opportunities to perform adaptive forwarding, which are implemented by the *Adaptation Engine*. First, modifications of the probabilities within a single column (*solid, red lines* in Figure 4.1) of an FWT change the forwarding probabilities for Interests asking for a specific c_l. These updates modify which faces/paths are preferred for forwarding Interests. Second, shifting forwarding probabilities among different columns (*dashed, blue lines* in Figure 4.1) allows prioritization of specific content types/prefixes. For instance, assume c_l is more important than c_k. In this case it can be beneficial to increase the probability of dropping Interests for c_k in favor of c_l. The necessary statistical information for these operations is provided by the *Statistic Collector*, which monitors the Interests received and satisfied by faces.

This chapter focuses on the core forwarding part of SAF (cf. *solid, red lines* in Figure 4.1). Additional context- and content-aware adaptations (cf. *dashed, blue lines* in Figure 4.1) are inherently supported by the design of SAF. However, since this kind of operations are more complex we defer the discussion of such mechanisms to Chapter 5. So, they are considered as out of scope for this chapter. Note that the forwarding core of SAF operates separately within a single column (content prefix) of the FWT. Therefore, and for the sake of readability we omit the notion of different $c \in \mathcal{C}$ for the remainder of this chapter.

4.2.2 A Throughput-Based Forwarding Measure

As previously mentioned, SAF is based on a measure. In the following we discuss the necessary fundamentals of measure theory. Furthermore, we argue that the selected (signed) measure can be expressed using a probability density function (cf. Theorem 4.1), and

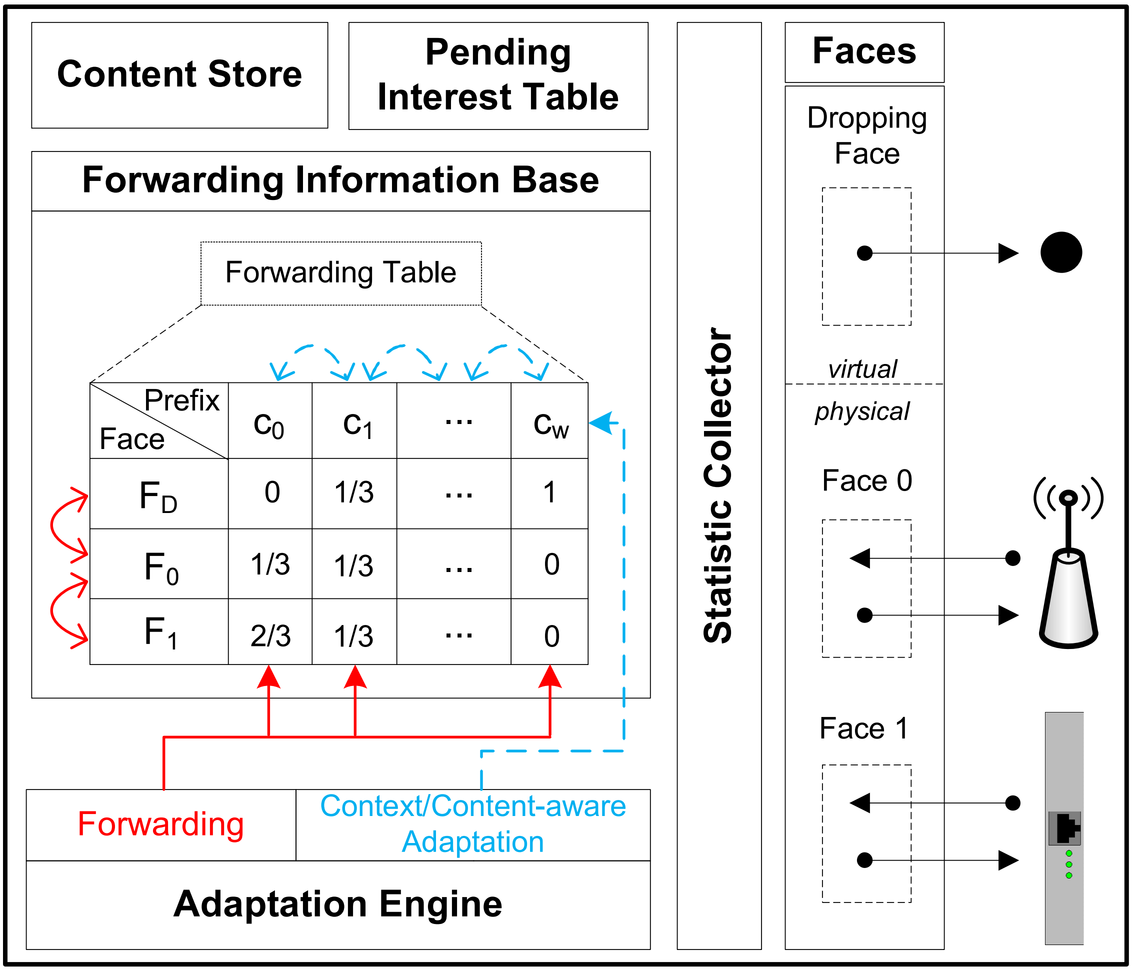


Figure 4.1: The model of an NDN node using SAF. The adaptation engine provides two levels of adaptation. The red, solid lines indicate adaptations considering operations concerning a single prefix, while the blue, dashed lines indicate adaptation over multiple contents considering context and content information.

therefore, a definite and optimal forwarding solution exists that is obtained by our approach (cf. Theorem 4.2). So, recapitulate the definition of a σ-Algebra. A σ-Algebra is a set of collections that suffices Definition 4.1. The definition of a (signed) measure is given in Definition 4.2. SAF is based on the measure $\mathcal{M}$ as defined in Definition 4.3.

Definition 4.1. Let $\mathcal{A}$ be some set and $\mathcal{P}(\mathcal{A})$ denotes its power set. Then $\mathcal{A}$, with $\mathcal{A} \subseteq \mathcal{P}(\Omega)$ is denoted as σ-Algebra with respect to the universal set Ω if and only if it holds that: *i)* $\emptyset, \Omega \in \mathcal{A}$; *ii)* $A \in \mathcal{A} \Rightarrow \overline{A} \in \mathcal{A}$, (where $\overline{A}$ denotes the complement $\overline{A} = \Omega \setminus A$); and *iii)* $\forall A_i \in \mathcal{A}, 1 \leq i \leq n, n \in \mathbb{N} : \bigcup_{i \geq 1} A_i \in \mathcal{A}$. [128]

Definition 4.2. Let $(\Omega, \mathcal{A})$ be a measurable space with a universal set Ω, $\mu : \mathcal{A} \rightarrow \mathbb{R} \cup$

$\{\infty, -\infty\}$ is a (signed) measure if and only if: *i)* $\mu(\emptyset) = 0$; *ii)* $\forall A_i, A_j \in \mathcal{A}, i \neq j, A_i \cap A_j = \emptyset : \mu(A_i \cup A_j) = \mu(A_i) + \mu(A_j)$; and *iii)* $\forall A \in \mathcal{A} : \mu(A) \geq 0$ (only if not signed). [128]

Definition 4.3. Let $(\Omega, \mathcal{A})$ be a measurable space with a universal set Ω, the corresponding σ-Algebra $\mathcal{A} \subseteq \mathcal{P}(\Omega)$ and $\mathcal{M}, S, U : \mathcal{A} \to \mathbb{R}$: $\forall F_i \in \mathcal{F} \setminus \{F_D\}, \forall A \in \mathcal{A} : \mathcal{M}_{F_i}(\emptyset) = 0$, $\mathcal{M}_{F_i}(A) = S_{F_i}(A) - U_{F_i}(A)$ and $\forall A \in \mathcal{A} : \mathcal{M}_{F_D}(A) = 0$.

$S_{F_i}(A)$ provides a measure for *satisfying Interests* and $U_{F_i}(A)$ defines a measure for *not satisfying Interests*. Note that, since $U_{F_i}(A)$ is the complementary measure to $S_{F_i}(A)$, it suffices to define $S_{F_i}(A)$. $S_{F_i}(A)$ is not predefined by SAF, which allows to use an arbitrary measure. For instance, one may define $S_{F_i}(A)$ as the number of Interests that are satisfied by Data packets below a certain delay threshold. The objective of $\mathcal{M}$, therefore the definition of $S_{F_i}(A)$, guides the update operations for the FWT. These operations are issued periodically, e.g., once every second. SAF finds the optimal forwarding strategy by maximizing $\mathcal{M}$ over the periods. The forwarding core of SAF operates separately on each column (prefix) of the FWT, thus providing for decoupling of periods for the different contents (prefixes). This allows to spread the node's work load over time. It remains to be shown that $\mathcal{M}$ suffices the conditions for a signed measure:

Proposition 4.1. Let $(\Omega, \mathcal{A})$ be a measurable space with a universal set Ω. Given two measurable functions S, U and $\forall A \in \mathcal{A}$ it holds that the linear combination $\mathcal{M}(\mathcal{A}) = S(A) - U(A)$ suffices the definition of a signed measure.

Proof [Proposition 4.1] It is easy to show that any canonical combination of measures $(\sum_{i=1}^{n} \alpha_i \cdot \mu_i, \alpha_i \in \mathbb{R}^+)$ is again a (positive) measure, but not the linear combination $(\alpha_i \in \mathbb{R})$ of positive measures (thus, positive measures are only closed under the canonical combination and not under linear combination). However, signed measures are the linear closure of positive measures. Thus, it immediately follows that $\mathcal{M}_{F_i}(A) = S_{F_i}(A) - U_{F_i}(A)$ is a signed measure because $S_{F_i}(A)$ and $U_{F_i}(A)$ are measures, and $\mathcal{M}$ is a linear combination of these measures. ■

In the following we show that for $\mathcal{M}$ a probability density function exists that can be maximized. Therefore, we require Definition 4.4 that defines the relation of absolutely continuous measures. Our given measures S and U are therefore absolutely continuous on a measurable space $(\Omega, \mathcal{A})$ with respect to the counting measure (denoted as μ_c, with $A \in \mathcal{A}$)

defined by Equation 4.1.

$$\mu_c(A) = \begin{cases} |A| & \text{if } A \text{ is finite,} \\ \infty & \text{if } A \text{ is infinite.} \end{cases} \tag{4.1}$$

Definition 4.4. Let $(\Omega, \mathcal{A})$ be a measurable space, given two (signed) measures μ and ν. Then ν is called absolutely continuous concerning μ ($\nu \ll \mu$) if $\forall A \in \mathcal{A} : \mu(A) = 0 \Rightarrow \nu(A) = 0$. [128]

Theorem 4.1. (Radon-Nikodym [129]). Let $(\Omega, \mathcal{A})$ be a measurable space, given two $\sigma-$finite measures μ and ν, with $\nu \ll \mu$. Then there exists a definite non-negative measurable function $f \in f : \Omega \to \mathbb{R}^+$ that holds $\nu(A) = \int_A f d\mu, \forall A \in \mathcal{A}$.

Considering Theorem 4.1, we know that there exist probability functions for S and U with respect to μ_c such that we can express S and U by $S(A) = \int_A f_s d\mu_c$, and $U(A) = \int_A f_u d\mu_c$. Thus, we can express $\mathcal{M}$ using Definition 4.3 by $\forall A \in \mathcal{A} : \mathcal{M}(A) = \int_A f_s d\mu_c - \int_A f_u d\mu_c = \int_A (f_s - f_u) d\mu_c$, and therefore, a probability density function for $\mathcal{M}$ exists that can be maximized.

In this chapter we use the measure $\mathcal{M}^T$, which maximizes the throughput by investigating the Interest satisfaction ratio on individual faces. Therefore, we consider for each node the set of forwarded Interests $\mathcal{I}_n$ during period n, for which a Data packet has been received (*satisfied* Interest) or a timeout has occurred (*unsatisfied* Interest) within that period n. Interests neither satisfied nor unsatisfied within a period are termed *pending*. We define $S_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is satisfied by a Data packet on } F_i\}|$ and $U_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is not satisfied on } F_i\}|$ $\forall F_i \in \mathcal{F} \setminus \{F_D\}$, and $S_{F_D}(\mathcal{I}_n) = |\{j \in \mathcal{I}_n : j \text{ is satisfied by } F_D\}|$ (F_D satisfies Interests by definition, but note that $\mathcal{M}_{F_D}(A) = 0$); $U_{F_D}(\mathcal{I}_n) = 0$). Pending Interests are not considered within the current period and their classification is postponed until the next period. This implicitly defines $\mathcal{M}^T$.

Remark 4.1. The ongoing discussion of SAF is agnostic to a concrete instantiation of a measure $\mathcal{M}$. One may use other measures than $\mathcal{M}^T$ considering the hop count or delay of received Interests. For instance, one may define a hop count-based measure $\mathcal{M}^H$ as follows: $S_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is satisfied by a Data packet } d \text{ on } F_i, \text{ where } d \text{ traversed fewer than } h \text{ hops}\}|, \forall F_i \in \mathcal{F} \setminus \{F_D\}$, and $S_{F_D}(\mathcal{I}_n) = |\{j \in \mathcal{I}_n : j \text{ is satisfied by } F_D\}|$.

For the sake of simplicity we write S_{F_i} for $S_{F_i}(\mathcal{I}_n)$ and U_{F_i} for $U_{F_i}(\mathcal{I}_n)$ for the remainder of the chapter. Table 4.1 depicts variables and expressions which are used for the definition of $\mathcal{M}$ and for SAF's update operations. Since these operations are executed during the transition from one period to another, variables hold the observed system state that is observable at the *end* of a given period.

Note that SAF is also scalable concerning space complexity. While classical approaches maintain a list of outgoing faces per prefix, SAF requires a vector of forwarding probabilities instead (cf. Figure 4.1). For instance, assume that those probabilities are quantized into the range of a byte. Furthermore, for each combination of prefixes and faces, SAF requires two counting variables (e.g., 4 bytes) representing S_{F_i} and U_{F_i}. Then the space complexity for SAF is given by $|\mathcal{C}| \cdot |\mathcal{F}| \cdot 9$ bytes. Since $\mathcal{F}$ is usually fixed on a node, space complexity is in $O(|\mathcal{C}|)$, as for most forwarding strategies (cf. Section 4.1.3).

4.2.3 Identifying Unsatisfied Traffic

For all update operations, SAF requires knowledge about the total unsatisfied traffic fraction δ. It provides information about the traffic percentage that has been forwarded towards wrong faces, given the measure $\mathcal{M}$. Using the expressions from Table 4.1, δ can be defined as given by Equation 4.2.

$$\delta = \begin{cases} 1 - \sum\limits_{F_i \in \mathcal{F}} ST_{F_i} = \sum\limits_{F_i \in \mathcal{F}} UT_{F_i} & \text{if } I > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

δ denotes the amount of traffic that should be forwarded on other faces during the next period. However, it is not yet known which faces provide a *poor service* and should therefore receive less traffic, and vice versa. For this reason, SAF splits the set of all physical faces $\mathcal{F} \setminus \{F_D\}$ into two disjoint subsets: (i) $\mathcal{F}_\mathcal{R}$, the set of reliable faces, and (ii) $\mathcal{F}_\mathcal{U}$, the set of unreliable faces. This partitioning is based on the definition of the reliability of a face R_{F_i} and the dynamic reliability threshold t. The threshold t is adapted based on a node's *health* status in the interval $t \in [t_{min}, t_{max}]$, which will be discussed later. Note that the definition of R_{F_i} is based only on the defined measure S_{F_i}.

The partitioning of $\mathcal{F} \setminus \{F_D\}$ into $\mathcal{F}_\mathcal{R}$ and $\mathcal{F}_\mathcal{U}$ provides a starting point to improve a node's forwarding decisions. SAF's update operations focus on shifting traffic from the unreliable faces towards the reliable faces. The next step for SAF is to evaluate the amount of traffic from faces in $\mathcal{F}_\mathcal{U}$ that can be shifted to faces in $\mathcal{F}_\mathcal{R}$ without overloading those. For this purpose, we define $\delta_\mathcal{U}$ (cf. Equation 4.3), which specifies the accumulated unsatisfied traffic from faces in $\mathcal{F_U}$ only.

$$\delta_{\mathcal{U}} = \begin{cases} 1 - \sum_{F_i \in \mathcal{F_U}} ST_{F_i} = \sum_{F_i \in \mathcal{F_U}} UT_{F_i} & \text{if } I > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

VAR./EXP.	DEFINITION / EXPLANATION
S_{F_i}	Number of satisfied Interests on F_i. F_D satisfies Interests by definition.
U_{F_i}	Number of unsatisfied Interests on F_i.
I	$\sum_{F_i \in \mathcal{F}} [S_{F_i} + U_{F_i}]$, satisfied and unsatisfied without pending Interests.
ST_{F_i}	$\begin{cases} \frac{S_{F_i}}{I} & \text{if } I > 0 \\ 0 & \text{otherwise.} \end{cases}$ Satisfied traffic fraction on F_i.
UT_{F_i}	$\begin{cases} \frac{U_{F_i}}{I} & \text{if } I > 0 \\ 0 & \text{otherwise.} \end{cases}$ Unsatisfied traffic fraction on F_i.
R_{F_i}	$\begin{cases} \frac{S_{F_i}}{S_{F_i} + U_{F_i}} & \text{if } S_{F_i} + U_{F_i} > 0 \\ 1 & \text{otherwise.} \end{cases}$ Reliability of F_i.
$p(F_i)$	Forwarding probability for face F_i.
t	$t \in [t_{min}, t_{max}]$, reliability threshold. $t_{max} \in]0,1[$, and $t_{min} \in]0, t_{max}[$.
$\mathcal{F_R}$	$\{F_i \in \mathcal{F} \setminus \{F_D\} \mid R_{F_i} \geq t\}$, reliable faces.
$\mathcal{F_U}$	$\mathcal{F} \setminus (\mathcal{F}_R \cup \{F_D\})$, unreliable faces.
$\mathcal{F_S}$	$\{F_i \in \mathcal{F_R} \mid ST_{F_i} + UT_{F_i} > 0\}$, faces that may take additional traffic.
$\mathcal{F_P}$	$\mathcal{F_R} \setminus \mathcal{F_S}$, faces used for probing.
$\delta \in [0,1]$	Total unsatisfied traffic fraction.
$\delta_{\mathcal{U}} \in [0,1]$	Unsatisfied traffic on faces in $\mathcal{F_U}$.
$\alpha_{F_i} \in]0,1]$	$\frac{1}{1+\sqrt{Var(X)}}$, traffic stability indicator, where X is a window over S_{F_i} with length N.
$\Delta \in [0,1]$	$\delta_{\mathcal{U}}$ with respect to $\alpha_{F_i} \forall F_i \in \mathcal{F_U}$.
σ_{F_i}	Resources for additional Interests on F_i.
$\rho \in [0,1]$	Traffic fraction used for probing.
$\tau \in (0, \infty)$	Duration of a period in seconds.

Table 4.1: Variables and expressions for SAF.

4.2.4 Update Operations

At the end of each period, SAF's objective is to shift the traffic fraction $\delta_{\mathcal{U}}$ from $\mathcal{F}_{\mathcal{U}}$ to $\mathcal{F}_{\mathcal{R}}$, or, in the worst case, to the virtual face F_D. In order to neglect short-term effects, the shifting of traffic is relaxed by $\alpha_{F_i} \in]0, 1]$. α_{F_i} is an indicator for the stability of the satisfied traffic over F_i and is defined in Table 4.1. It is determined by the standard deviation of the satisfied Interests over a given number of periods. Note that, the larger the standard deviation of S_{F_i} over the periods, the smaller α_{F_i}, and vice versa. This ensures that SAF balances the updates of the FWT taking into account traffic stability, which allows stronger changes if the observed state is steady over the periods. Plugging α_{F_i} into Equation 4.3 provides the relaxed unsatisfied traffic fraction Δ, as denoted in Equation 4.4.

$$\Delta = \begin{cases} \sum\limits_{F_i \in \mathcal{F}_{\mathcal{U}}} UT_{F_i} \cdot \alpha_{F_i} & \text{if } I > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

Algorithm 4.2 Pseudocode for the FWT updates in SAF

```
1: Δ ← determineUnsatisfiedTraffic()
2: Γ ← Δ + p(F_D)
3: if Γ > 0 then
4:    F_S, F_P ← splitSet(F_R)
5:    shiftTraffic(F_U, F_S, Γ)
6:    p(F_D) = Γ − Γ'
7:    if p(F_D) > 0 then
8:       probeOnFaces(F_P)
9:       if p(F_D) > (1 − t) then
10:         decreaseReliability(t)
11:      end if
12:   end if
13: else if I > 0 then
14:   increaseReliability(t)
15: end if
```

Algorithm 4.2 outlines SAF's update procedure, which can be used as the road map for Subsections 4.2.4 to 4.2.6. In line 2 of Algorithm 4.2, Γ is introduced. Γ denotes the sum of the unsatisfied traffic fraction Δ and the current forwarding probability $p(F_D)$ of the virtual face F_D. It is important to consider the sum of Δ and $p(F_D)$, as Δ does not consider the discarded traffic fraction on F_D. Note that $\Delta > 0 \Leftrightarrow \mathcal{F}_U \neq \emptyset$ as indicated by Proposition 4.2. *Remark*: $\Delta = 0 \nRightarrow UT_{F_i} = 0 : \forall F_i \in \mathcal{F}$.

Proposition 4.2. Suppose $\forall F_i \in \mathcal{F} : \alpha_{F_i} \in]0,1]$ and $I > 0$. Then, it holds that $\Delta > 0 \iff \mathcal{F}_{\mathcal{U}} \neq \emptyset$.

Proof [Proposition 4.2] "$\Rightarrow$", suppose that $\mathcal{F}_{\mathcal{U}} = \emptyset$, we have

$$0 < \Delta = \underbrace{\sum_{F_i \in \mathcal{F}_{\mathcal{U}}} UT_{F_i} \cdot \alpha_{F_i}}_{=0,\text{ since } \mathcal{F}_{\mathcal{U}} = \emptyset} = \underbrace{\sum_{F_i \in \mathcal{F}_{\mathcal{U}}} \frac{U_{F_i}}{I} \cdot \alpha_{F_i}}_{=0,\text{ since } \mathcal{F}_{\mathcal{U}} = \emptyset} = 0$$

which is a contradiction to the assumption. Thus, $\Delta > 0 \Rightarrow \mathcal{F}_{\mathcal{U}} \neq \emptyset$. "$\Leftarrow$", suppose that $\Delta = 0$. Since $\mathcal{F}_{\mathcal{U}} \neq \emptyset$ and according to the definition of $\mathcal{F}_{\mathcal{U}}$ and $\mathcal{F}_{\mathcal{R}}$, we have

$$0 = \Delta = \alpha \cdot \delta_{\mathcal{U}} = \underbrace{\sum_{F_i \in \mathcal{F}_{\mathcal{U}}} \frac{U_{F_i}}{I} \cdot \alpha_{F_i}}_{>0,\text{ since } \mathcal{F}_{\mathcal{U}} \neq \emptyset \text{ and } \alpha_{F_i} > 0} > 0,$$

which is a contradiction. Thus, $\mathcal{F}_{\mathcal{U}} \neq \emptyset \Rightarrow \Delta > 0$. ■

Algorithm 4.2 has a trivial case, which eventuates if $\Gamma = 0$. In this case no changes in the FWT are required, since neither unsatisfied traffic exists nor any traffic is dropped in advance. In this favorable case, t is increased if $I > 0$ in this period. However, if $\Gamma > 0$, SAF resolves the unsatisfied traffic using the two following approaches: *i*) adaptation of the forwarding probabilities within the FWT (cf. Alg. 4.2 line 4-5); *ii*) identification of yet unknown paths to the desired content via probing (cf. Alg. 4.2 line 8). For the sake of simplicity, we separate the further discussion of Algorithm 4.2 into these two parts. The adaptation of the reliability threshold (cf. Alg. 4.2 lines 10 and 14) is discussed at the end of the second part.

Note that Algorithm 4.2 is executed for each content prefix in the FWT (cf. Figure 4.1). If we assume the worst case for Algorithm 4.2 ($\Gamma > 0$, $p(F_D) > 0$, and $p(F_D) > (1-t)$), then the asymptotic time complexity is given by $O(|\mathcal{C}|)$. This is due to the fact that only simple arithmetic operations are used (cf. Subsection 4.2.5 and 4.2.6), and their quantity solely depends on the number of faces $|\mathcal{F}|$, which is usually constant.

4.2.5 Adaptation of Forwarding Probabilities

SAF shifts traffic between faces only if $\Gamma > 0$. The principal objective in this step is to shift the unsatisfied traffic from the unreliable faces $\mathcal{F}_{\mathcal{U}}$ towards the reliable faces $\mathcal{F}_{\mathcal{R}}$ (cf. Alg. 4.2 line 5) without overloading them. Of course, this is not always possible, which in the worst case forces the algorithm to forward some Interests towards the virtual face F_D (cf. Alg. 4.2 line 6). Before any actions are taken, SAF splits the set $\mathcal{F_R}$ into two disjoint subsets. The subset $\mathcal{F_S} \subseteq \mathcal{F_R}$ includes only faces from $\mathcal{F_R}$ which have successfully forwarded Interests in the current period. The second subset, $\mathcal{F_P} = \mathcal{F_R} \setminus \mathcal{F_S}$ (cf. Table 4.1), includes faces that are considered as reliable only because they have not forwarded any Interests ($\forall F_i \in \mathcal{F_P} : S_{F_i} = U_{F_i} = 0$) in the current period. So it is very likely that faces in $\mathcal{F_P}$ cannot fulfill requests, which is why we do not consider them for *attracting additional* traffic.

Before SAF performs the shifting, it determines how much additional traffic the faces in $\mathcal{F_S}$ may take, without decreasing their reliability below t. Proposition 4.3 provides σ_{F_i}, which denotes the number of additional Interests the face $F_i \in \mathcal{F_S}$ may take without dropping R_{F_i} below t.

Proposition 4.3. For a given reliability t, every $F_i \in \mathcal{F_S}$ can satisfy $0 \leq \sigma_{F_i} \leq \lfloor \frac{S_{F_i}}{t} - S_{F_i} - U_{F_i} \rfloor$ additional Interests.

Proof [Proposition 4.3] Since $F_i \in \mathcal{F_S}$ and based on the definition of R_{F_i} we have,

$$\frac{S_{F_i}}{S_{F_i} + U_{F_i} + \sigma_{F_i}} \geq t \Rightarrow \sigma_{F_i} \leq \frac{S_{F_i}}{t} - S_{F_i} - U_{F_i}$$

■

Given Γ and $\sigma_{F_i} \forall F_i \in \mathcal{F_S}$, SAF is able to determine the maximum traffic that can/should be shifted from $\mathcal{F_U}$ to $\mathcal{F_S}$. We denote this amount as Γ' as defined in Equation 4.5.

$$\Gamma' = \min\left(\frac{1}{I} \cdot \sum_{F_i \in \mathcal{F}_S} \sigma_{F_i}, \Gamma\right) \tag{4.5}$$

The next step for SAF is to determine the forwarding probabilities for period $n+1$ by: *i*) decreasing the forwarding probabilities for faces in $\mathcal{F_U}$ by Γ', Equation 4.6; *ii*) increasing the forwarding probabilities for faces in $\mathcal{F_S}$ by Γ', Equation 4.7.

$$\forall F_i \in \mathcal{F_U} : p_{n+1}(F_i) \leftarrow p_n(F_i) - UT_{F_i} \cdot \alpha_{F_i} \tag{4.6}$$

$$\forall F_i \in \mathcal{F_S} : p_{n+1}(F_i) \leftarrow p_n(F_i) + \Gamma' \cdot \frac{\sigma_{F_i}}{\sum\limits_{F_i \in \mathcal{F}_S} \sigma_{F_i}} \tag{4.7}$$

Note that Equation 4.6, does not use Γ' to determine the amount of the traffic reduction. Instead, the complete unsatisfied traffic from a face $F_i \in F_{\mathcal{U}}$ considering α_{F_i} is removed, which exactly adds up to Δ. As the residual traffic $(\Gamma \; - \Gamma')$ can not be satisfied by any $F_i \in \mathcal{F} \setminus \{F_D\}$, it is beneficial to drop this portion of the traffic. Forwarding those Interests would likely cause congestion and impair the performance. Therefore, SAF simply determines the amount of residual unsatisfied traffic and puts it on the virtual face F_D, denoted by Equation 4.8 (cf. Alg. 4.2 line 6). Theorem 4.2 shows that SAF converges to a steady state, which is defined as a state where $\mathcal{F_U} = \emptyset$.

$$p(F_D) = \Gamma - \Gamma' \tag{4.8}$$

Theorem 4.2. Given any state with $\mathcal{F_U} \neq \emptyset$, $t \in [t_{min}, t_{max}]$ and without the loss of generality set α sufficiently small (i.e., $\alpha = \min(\alpha_{F_i} : F_i \in \mathcal{F_U})$). d_{F_i} denotes the number of Interests that can be satisfied on face F_i, $p^*(F_i) = \frac{d_{F_i}}{I}$ denotes the optimal forwarding probability for face F_i and I denotes the number of Interests that shall be forwarded in every period. Assume that I is constant for every period and that $I \cdot p(F_i) \geq \frac{d_{F_i}}{t}$. According to Equations 4.6, 4.7 and 4.8 SAF converges to $\mathcal{F_U} = \emptyset$ after n periods (iterations) bounded by:

$$n \leq \max_{F_i \in \mathcal{F_U}} \left(\left\lceil \frac{ln(\frac{d_{F_i}}{t} - d_{F_i}) - ln(p_0(F_i) \cdot I - d_{F_i})}{ln(1-\alpha)} \right\rceil \right), \tag{4.9}$$

where p_0 denotes the initial forwarding probability.

Proof [Theorem 4.2] In the case of $\mathcal{F_U} \neq \emptyset$, SAF uses Equation 4.6 to reduce the forwarding probabilities on every $F_i \in \mathcal{F_U}$ until all become reliable. According to the definition of R_{F_i}, $F_i \in \mathcal{F_U}$ iff $p_n(F_i) \cdot I > \dfrac{d_{F_i}}{t}$. In this case we either shift the probabilities $\forall F_i \in \mathcal{F_U}$ to faces in $\mathcal{F_S}$, or we drop the traffic by increasing the forwarding probability of F_D. Since we may express UT_{F_i} as $p_n(F_i) - \dfrac{d_{F_i}}{I}$ assuming a perfect random distribution for Algorithm 4.1, and according to Equation 4.6 we have, $\forall F_i \in \mathcal{F_U}$:

$$p_{n+1}(F_i) = p_n(F_i) \cdot (1-\alpha) + \alpha \cdot \frac{d_{F_i}}{I}. \tag{4.10}$$

Expanding the recursion,

$$
\begin{aligned}
p_{n+1}(F_i) &= p_n(F_i)\cdot(1-\alpha)+\alpha\frac{d_{F_i}}{I}\\
p_{n+1}(F_i) &= \left(p_{n-1}(F_i)\cdot(1-\alpha)+\alpha\frac{d_{F_i}}{I}\right)\cdot(1-\alpha)+\alpha\frac{d_{F_i}}{I}\\
&\dots\\
p_{n+1}(F_i) &= \left(\dots\left(p_0(F_i)\cdot(1-\alpha)+\alpha\frac{d_{F_i}}{I}\right)\cdot\dots\cdot(1-\alpha)+\alpha\frac{d_{F_i}}{I}\right)\cdot(1-\alpha)+\alpha\frac{d_{F_i}}{I}
\end{aligned}
$$

suggests that Equation 4.11 holds. We proof this assumption by induction:

Basis: $n = 1$

$$
\begin{aligned}
&p_0(F_i)\cdot(1-\alpha)^1+\alpha\cdot\frac{d_{F_i}}{I}\cdot\underbrace{\sum_{j=0}^{1-1}(1-\alpha)^j}_{=1}\\
&= p_0(F_i)\cdot(1-\alpha)+\alpha\cdot\frac{d_{F_i}}{I}
\end{aligned}
$$

Induction: $n \to n+1$:

$$
\begin{aligned}
p_{n+1}(F_i) &= \left(p_0(F_i)\cdot(1-\alpha)^n+\alpha\cdot\frac{d_{F_i}}{I}\cdot\sum_{j=0}^{n-1}(1-\alpha)^j\right)\cdot(1-\alpha)+\alpha\cdot\frac{d_{F_i}}{I}\\
&= p_0(F_i)\cdot(1-\alpha)^{n+1}+\alpha\cdot\frac{d_{F_i}}{I}\cdot\sum_{j=0}^{n-1}(1-\alpha)^j\cdot(1-\alpha)+\alpha\cdot\frac{d_{F_i}}{I}\\
&= p_0(F_i)\cdot(1-\alpha)^{n+1}+\alpha\cdot\frac{d_{F_i}}{I}\cdot\sum_{j=0}^{n-1}(1-\alpha)^{j+1}+\alpha\cdot\frac{d_{F_i}}{I}\cdot(1-\alpha)^0\\
&= p_0(F_i)\cdot(1-\alpha)^{n+1}+\alpha\cdot\frac{d_{F_i}}{I}\cdot\sum_{j=1}^{n}(1-\alpha)^j+\alpha\cdot\frac{d_{F_i}}{I}\cdot(1-\alpha)^0\\
&= p_0(F_i)\cdot(1-\alpha)^{n+1}+\alpha\cdot\frac{d_{F_i}}{I}\cdot\sum_{j=0}^{n}(1-\alpha)^j.
\end{aligned}
$$

$$
p_n(F_i) = \underbrace{p_0(F_i)\cdot(1-\alpha)^n}_{\lim\limits_{n\to\infty}\to 0}+\underbrace{\alpha\cdot\frac{d_{F_i}}{I}\cdot\sum_{j=0}^{n-1}(1-\alpha)^j}_{\lim\limits_{n\to\infty}\to\alpha\cdot\frac{d_{F_i}}{I}\cdot\frac{1}{\alpha}}. \tag{4.11}
$$

Equation 4.11 and $\lim\limits_{n\to\infty} p_n(F_i) = \frac{d_{F_i}}{I}$ provides the claim that $p_n(F_i) \to p^*(F_i)$ and with $t \to 1$, $p^*(F_i) = \frac{d_{F_i}}{I}$ denotes the optimum for face F_i. Further, one easily shows that the convergence speed ($|p_{n+1}(F_i) - p^*(F_i)| \leq M \cdot |p_n(F_i) - p^*(F_i)|$) is linear with $M = (1-\alpha)$:

$$\begin{aligned}
|p_n(F_i) - p^*(F_i)| &= \\
|p_0(F_i) \cdot (1-\alpha)^n + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=0}^{n-1} (1-\alpha)^j - \frac{d_{F_i}}{I}| &= \\
|(p_0(F_i) \cdot (1-\alpha)^{n-1} + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=0}^{n-2} (1-\alpha)^j) \cdot (1-\alpha) + \alpha \cdot \frac{d_{F_i}}{I} - \frac{d_{F_i}}{I}| &= \\
|(p_0(F_i) \cdot (1-\alpha)^{n-1} + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=0}^{n-2} (1-\alpha)^j) \cdot (1-\alpha) - (1-\alpha) \cdot \frac{d_{F_i}}{I}| &= \\
|(1-\alpha)| \cdot |(p_0(F_i) \cdot (1-\alpha)^{n-1} + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=0}^{n-2} (1-\alpha)^j) - \frac{d_{F_i}}{I}| &\leq \\
|1-\alpha| \cdot |p_{n-1}(F_i) - p^*(F_i)|.
\end{aligned}$$

By plugging the expression given in Equation 4.11 for $p_n(F_i)$ into $p_n(F_i) \cdot I > \frac{d_{F_i}}{t}$ and using the formula for the geometric series ($0 < 1-\alpha < 1$) we have,

$$\begin{aligned}
p_n(F_i) \cdot I &> \frac{d_{F_i}}{t} \\
\left(p_0(F_i) \cdot (1-\alpha)^n + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=0}^{n-1} (1-\alpha)^j \right) \cdot I &> \frac{d_{F_i}}{t} \\
(p_0(F_i) \cdot (1-\alpha)^n) \cdot I + \alpha \cdot d_{F_i} \cdot \sum_{j=0}^{n-1} (1-\alpha)^j &> \frac{d_{F_i}}{t} \\
(p_0(F_i) \cdot (1-\alpha)^n) \cdot I + \alpha \cdot d_{F_i} \cdot \frac{1-(1-\alpha)^{n-1}}{1-(1-\alpha)} &> \frac{d_{F_i}}{t} \\
(p_0(F_i) \cdot (1-\alpha)^n) \cdot I + \frac{\alpha \cdot d_{F_i} - \alpha \cdot d_{F_i} \cdot (1-\alpha)^{n-1}}{\alpha} &> \frac{d_{F_i}}{t} \\
(p_0(F_i) \cdot (1-\alpha)^n) \cdot I + d_{F_i} - d_{F_i} \cdot (1-\alpha)^{n-1} &> \frac{d_{F_i}}{t} \\
(1-\alpha)^n \cdot (p_0(F_i) \cdot I - d_{F_i}) + d_{F_i} &> \frac{d_{F_i}}{t} \qquad (4.12)
\end{aligned}$$

If we re-arrange Equation 4.12, then the number of periods n until all unreliable faces become reliable is bounded by:

$$n \leq \max_{F_i \in \mathcal{F}_\mathcal{U}} \left(\left\lceil \frac{ln(\frac{d_{F_i}}{t} - d_{F_i}) - ln(p_0(F_i) \cdot I - d_{F_i})}{ln(1-\alpha)} \right\rceil \right)$$

■

Remark 4.2. In Theorem 4.2 we assumed a constant number of Interests in each period. In order to show that Theorem 4.2 is still valid given a varying number of Interests for each time dependent period, we may define a period by the number of Interests. Thus, we again have a constant number of Interests during a period and Theorem 4.2 holds.

Remark 4.3. Let $(\mathbb{R}, |x-y|)$ be a metric space. According to Equation 4.10, $p^*(F_i)$ is a fixed point. This fixed point is globally asymptotic stable for $\alpha \in]0,1]$.

Proof [Remark 4.3] Let $\Phi(k,\kappa,\xi) = \xi(1-\alpha)^{k-\kappa} + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=\kappa}^{k-1}(1-\alpha)^{j+1} \cdot \mathbb{1}_{\{k \geq \kappa\}}$ with initial pair $(\xi,\kappa) \in \mathbb{R} \times \mathbb{N}$, $p_\kappa = \xi$ and $k \geq \kappa$ be the general solution to our autonomous difference equation given in Equation 4.10. First, we show that $p^*(F_i)$ is a fixed point and it is globally attractive.

$$p_n(F_i) \cdot (1-\alpha) + \alpha \cdot \frac{d_{F_i}}{I} = p_n(F_i) \iff p_n(F_i) = \frac{d_{F_i}}{I}.$$

According to Theorem 4.2, it follows that for $\alpha \in]0,1]$ the fixed point $p^*(F_i)$ is globally attractive $(\forall(\xi,\kappa) \in \mathbb{R} \times \mathbb{N} : lim_{n\to\infty}|p_n(F_i) - p^*(F_i)| = 0)$. Second, we show that the solution $p* = \frac{d_{F_i}}{I}$ is stable for all $\alpha \in]0,1]$. For $\alpha \in]0,1[$, $\forall k \in \mathbb{N}, k \geq \kappa$ and for any $\varepsilon > 0$ we have,

$$\begin{aligned} &|\Phi(k,\kappa,\xi) - p^*(F_i)| = \\ &|\xi \cdot (1-\alpha)^{k-\kappa} + \underbrace{\alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=\kappa}^{k-1}(1-\alpha)^{k-(j+1)} \cdot \mathbb{1}_{\{k \geq \kappa\}}}_{*} - \frac{d_{F_i}}{I}| \end{aligned} \tag{4.13}$$

We may express * by using a index shift as:

$$\begin{aligned} \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=0}^{k-\kappa-1}(1-\alpha)^j &= \\ \alpha \cdot \frac{d_{F_i}}{I} \cdot \frac{1-(1-\alpha)^{(k-\kappa)}}{1-(1-\alpha)} &= \\ \frac{d_{F_i}}{I} \cdot (1-(1-\alpha)^{(k-\kappa)}) \end{aligned}$$

Plugging this into Equation 4.13 we have,

$$
\begin{aligned}
|\xi \cdot (1-\alpha)^{k-\kappa} + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=\kappa}^{k-1} (1-\alpha)^{k-(j+1)} \cdot \mathbb{1}_{\{k \geq \kappa\}} - \frac{d_{F_i}}{I}| &= \\
|\xi \cdot (1-\alpha)^{k-\kappa} + \frac{d_{F_i}}{I} \cdot (1-(1-\alpha)^{(k-\kappa)}) - \frac{d_{F_i}}{I}| &= \\
|\xi \cdot (1-\alpha)^{k-\kappa} + \frac{d_{F_i}}{I} - \frac{d_{F_i}}{I}(1-\alpha)^{(k-\kappa)} - \frac{d_{F_i}}{I}| &= \\
|\xi \cdot (1-\alpha)^{k-\kappa} - \frac{d_{F_i}}{I}(1-\alpha)^{(k-\kappa)}| &= \\
|\underbrace{(1-\alpha)^{k-\kappa}}_{\leq 1} \cdot \Big(\xi - \frac{d_{F_i}}{I}\Big)| &\leq \\
|\xi - \frac{d_{F_i}}{I}| < \varepsilon \quad \forall \xi \in B_\delta\Big(\frac{d_{F_i}}{I}\Big)&,
\end{aligned}
$$

with $\delta = \frac{\varepsilon}{2}$, where $B_\delta(y) := \{x \in \mathbb{R} : |x-y| < \delta\}$. For $\alpha = 1$ the proof is analogues. Thus, $p^*(F_i)$ is globally asymptotic stable. ■

4.2.6 Searching for Unknown Paths and Cached Replicas

After shifting traffic in the previous step, $p(F_D)$ holds the residual traffic that cannot be forwarded on the physical faces. This traffic is going to be dropped by the virtual face F_D. SAF's probing mechanism takes a share of this traffic and uses it to discover new paths towards the content origin or to discover nodes holding cached replicas. The fraction of the traffic that is used for probing is limited by ρ as denoted in Equation 4.14. The probe is defined by Equation 4.15.

$$\rho = 1 - \sum_{F_i \in \mathcal{F} \setminus F_D} ST_{F_i} = \delta - ST_{F_D} \tag{4.14}$$

$$probe = p(F_D) \cdot \rho \tag{4.15}$$

ρ increases based on the proportion of the satisfied and the unsatisfied traffic on physical faces. The larger the fraction of the unsatisfied traffic, the larger ρ. The idea behind this is that the more unsatisfied traffic we have, the more important it is to discover additional paths to the content. In the worst case where $p(F_D) = 1$ the entire traffic is used for probing. For instance, this can happen when a previously working path to the content suffers from link failure(s). In this case probing may help to circumvent the broken link or to identify yet unknown path(s) to cached content replicas. The probe is uniformly distributed on all faces in $\mathcal{F_P}$ as denoted in Equation 4.16. Finally, the forwarding probability of the virtual face F_D needs to be adjusted as outlined in Equation 4.17.

$$\forall F_i \in \mathcal{F_P} : p_{n+1}(F_i) = p_n(F_i) + \tfrac{probe}{|\mathcal{F_P}|} \tag{4.16}$$

$$p_{n+1}(F_D) = p_n(F_D) - probe \tag{4.17}$$

After the probing has been carried out, SAF checks if it has to decrease the reliability threshold (cf. Algorithm 4.2 lines 9–10). If $p(F_D)$ is larger than $(1-t)$, the reliability threshold has to be decreased since it cannot be retained. In contrast to this, the reliability threshold should be increased if currently all Interests can be satisfied with a reliability of at least t (cf. Algorithm 4.2 lines 13–14). We suggest Equations 4.18 and 4.19 for the adjustment of t, where λ denotes the rate of change.

$$t_{n+1} = (1-\lambda) \cdot t_n + \lambda \cdot t_{max} \tag{4.18}$$

$$t_{n+1} = (1-\lambda) \cdot t_n - \lambda \cdot t_{min} \tag{4.19}$$

4.2.7 SAF Examples

We present two examples, one illustrating how SAF approaches the optimal FWT, and a second one illustrating the probing mechanism that allows SAF to recover from a link failure in the given scenario. Figure 4.2 depicts the example network with the corresponding link capacities in Mbps. Our object of investigation is the router R. This router has to forward Interests for a given content prefix c, which can be retrieved at the content provider. R maintains three physical faces which have links to the routers F_0, F_1, and F_2 (we use faces from R interchangeably with routers F_0, F_1, and F_2). Interests forwarded via $F1$ and $F2$ reach the content provider. We assume that R has to satisfy a constant flow of Interests requesting 3 Mbps of Data packets. Furthermore, we assume that the initial value for $t = 0.5$ and $\forall F_i \in \mathcal{F} : \alpha_{F_i} = 1$ for all periods. The initial FWT for R can either be provided by the routing layer, or, as in this example, the traffic is uniformly distributed among the faces (cf. Figure 4.3a). During the first period each of the faces F_0, F_1, and F_2 receives Interests

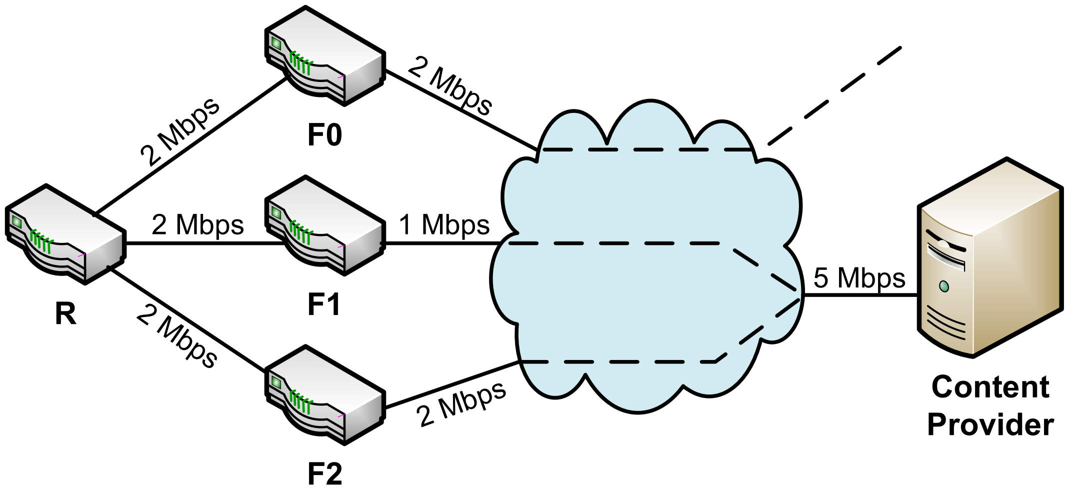


Figure 4.2: Example network to illustrate the principles of SAF.

$$\begin{array}{cc} t{=}0.5 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \end{array}$$

(a) Initial

$$\begin{array}{cc} t{=}0.5 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 0 \\ 0 \\ 1/2 \\ 1/2 \end{pmatrix} \end{array}$$

(b) Iteration 1

$$\begin{array}{cc} t{=}0.75 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 0 \\ 0 \\ 1/2 \\ 1/2 \end{pmatrix} \end{array}$$

(c) Iteration 2

$$\begin{array}{cc} t{=}0.75 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 0 \\ 0 \\ 0.42 \\ 0.58 \end{pmatrix} \end{array}$$

(d) Iteration 3

Figure 4.3: FWTs of router R for the **first** example.

requesting Data packets for a bitrate of 1 Mbps. While F_1 and F_2 satisfy all Interests, F_0 is not able to satisfy any. Therefore, F_0 is classified as an unreliable face and since $\alpha_{F_0} = 1$ the entire unsatisfied traffic is removed (cf. Eq. 4.6). The forwarding probability of F_0 is shifted towards F_1 and F_2 (cf. Eq. 4.7 and Figure 4.3b). In the second period both, F_1 and F_2, receive Interests requesting Data packets for 1.5 Mbps. While F_2 is able to satisfy all Interests, F_1 satisfies only $\frac{2}{3}$. However, based on t both faces are considered as reliable and no changes are performed on the FWT (cf. Figure 4.3c). Only the threshold t is increased (cf. Alg. 4.2 line 14), assume t is increased to 0.75. In the third period the traffic is distributed as in the second period, with the same issue that F_1 can satisfy only $\frac{2}{3}$ of the Interests. However, this time $t > \frac{2}{3}$, which classifies F_1 as unreliable. Therefore, the unsatisfied traffic (≈ 0.08) is shifted towards F_2 (cf. Figure 4.3d). With each further period, the FWT approaches the optimal distribution ($p(F_1) = \frac{1}{3}$, $p(F_2) = \frac{2}{3}$).

For the second example we consider the same network as before with one major change.

$$\begin{array}{cc} t{=}0.99 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 0 \\ 0 \\ 1/3 \\ 2/3 \end{pmatrix} \end{array}$$

(a) Initial

$$\begin{array}{cc} t{=}0.75 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 2/9 \\ 4/9 \\ 1/3 \\ 0 \end{pmatrix} \end{array}$$

(b) Iteration 1

$$\begin{array}{cc} t{=}0.75 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 0 \\ 0.57 \\ 0.43 \\ 0 \end{pmatrix} \end{array}$$

(c) Iter. 2 *and* 3 ($t = 0.85$)

$$\begin{array}{cc} t{=}0.85 & c \\ \begin{array}{c} F_D \\ F_0 \\ F_1 \\ F_2 \end{array} & \begin{pmatrix} 0 \\ 2/3 \\ 1/3 \\ 0 \end{pmatrix} \end{array}$$

(d) Iteration 4

Figure 4.4: FWTs of router R for the **second** example.

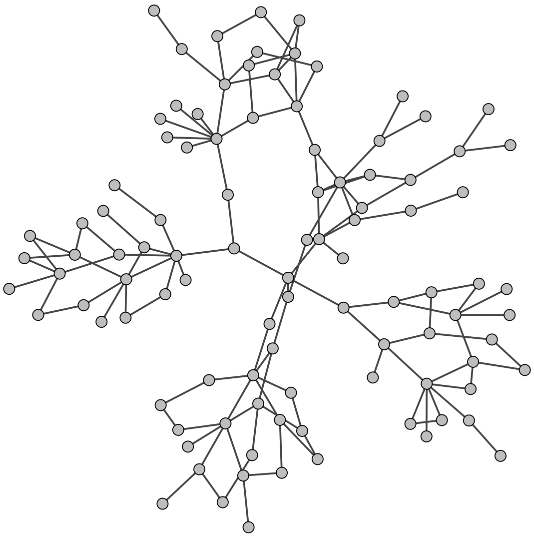
The path traversing F_0 ends at a content replica of the content provider. We use the optimal FWT from the previous example as the starting point for the second example, $t = 0.99$ and $\forall F_i \in \mathcal{F} : \alpha_{F_i} = 1$ for all periods. We further assume that, at the beginning of the first period, a link failure on the path between F_2 and the content provider occurs. Therefore, none of the Interests can be satisfied via F_2, which causes this face to be marked as unreliable. As $t = 0.99$, F_1 is not able to take any additional Interests from F_2 causing this traffic portion ($\frac{2}{3}$) to be shifted to the virtual face F_D. As $p(F_D) > 0$, probing is issued (cf. Alg. 4.2 line 7). This results in the situation that a share of $p(F_D) \cdot \rho = \frac{2}{3} \cdot \frac{2}{3}$ Interests is used for probing on face F_0 (cf. Eq 4.14 and 4.16). Figure 4.4b illustrates the FWT after the first iteration. Since $p(F_D) > 1 - t$, t is decreased. Assume t is decreased to $t = 0.75$. In the second period, all Interests forwarded via F_0 and F_1 can be satisfied and SAF is able to discover a new path to the content via F_0. After the first iteration, still 22.2% of the Interests are dropped in advance. Since in the second period all Interests forwarded via F_0 and F_1 can be satisfied and $\sigma_{F_0} + \sigma_{F_1} > p(F_D)$ ($\approx 0.148 + 0.111 > 0.222$) the forwarding probabilities for F_D are distributed according to Eq. 4.7 (cf. Figure 4.4c). As the capacity of F_1 was already exhausted, a part of the traffic ($\approx 0.43 - 0.33 \approx 0.1$) is dropped during the third period due to congestion. However, according to $t = 0.75$, F_1 is still reliable and since all faces are considered as reliable during the third period the FTW is not modified between iterations two and three. Only the reliability threshold is increased, assume t is increased to 0.85. Therefore, after the fourth period F_1 is considered as unreliable and the unsatisfied traffic is completely taken away according to Eq. 4.6 (as $\forall F_i \in \mathcal{F} : \alpha_{F_i} = 1$ is assumed). This traffic portion is then distributed among the reliable face(s) in $\mathcal{F_S}$ (in this case only $\mathcal{F_S} = \{F_0\}$), and since $\sigma_{F_0} > UT_{F_1}$, SAF reaches the optimal FTW as shown in Figure 4.4d.

4.3 Performance Evaluation

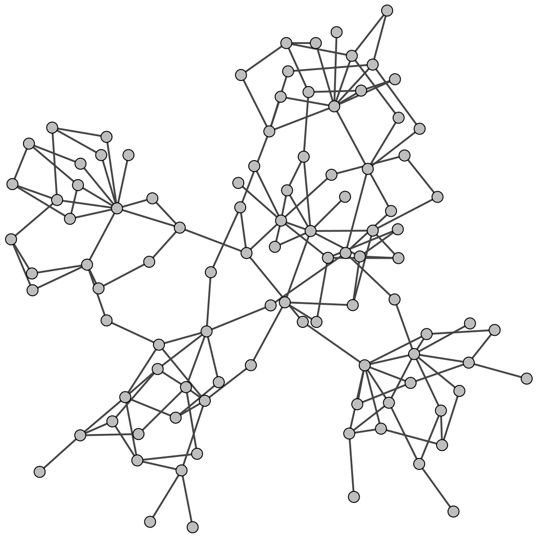
In this section we investigate the performance of SAF by comparing it to related work and standard forwarding algorithms already present in the ns3/ndnSIM 2.0 simulator [121]. The selected algorithms from related work are: ideal Nearest Replica Routing (iNRR) [107], the Request Forwarding Algorithm (RFA) [95], and the On-demand Multi-Path Interest Forwarding (OMP-IF) [127]. As already mentioned in Section 4.1.3, iNRR uses an oracle to determine for every Interest the nearest node which holds a cached copy of the corresponding Data packet and to provide the shortest path to this node [107]. RFA has been implemented as described in [95]. OMP-IF has been implemented according to the descriptions given in [127]. We provide an open source implementation of SAF together with the implementations of iNRR, RFA, and OMP-IF in ns3/ndnSIM 2.0 at `http://icn.itec.aau.at`. For SAF we use the presented measure $\mathcal{M}^T$, thus, maximizing the throughput for specific content prefixes at every node in the network.

4.3.1 Network Topology Generation

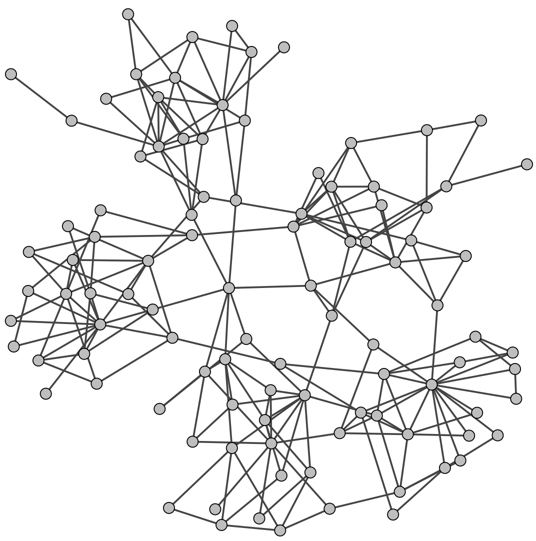
For generating random network topologies we employ the network topology generator BRITE [130]. BRITE was configured to build scale-free networks in a top-down fashion since the Internet topology is likely to be described by power-laws [131]. We model the infrastructure of large ISPs, interconnecting several autonomous systems and access networks. The top level represents $\mu = 5$ autonomous systems (AS). Each AS maintains $\nu = 20$ nodes (bottom level) acting as ICN routers (in total we have $5 \cdot 20 = 100$ routers) and serving as access nodes for later on added client and server nodes. Both the top- and the bottom-level graphs are randomly generated based on the Barabási-Albert model. The model was configured to connect each node with exactly one neighbour. This generates scale-free networks with no redundant paths. As ICN characteristics such as multi-path delivery and link-failure recovery can only be evaluated properly if redundant paths exist, we extended the generated graphs with additional random edges. This was necessary because BRITE only enables to set an integer number of links per node, and therefore does not support a "probability-based" link insertion. The alternative, to set the number of neighbours to values greater or equal two, results in already very well-connected networks inhibiting a fine-granular investigation of the influence of redundant paths on the forwarding strategies.



(a) Sample topology *LowCon*



(b) Sample topology *MediumCon*



(c) Sample topology *HighCon*

Figure 4.5: Sample topologies for the different network **connectivities** as defined in Table 4.2 (without client and server nodes).

Based on the aforementioned considerations, we generate nine different topology variants. These topology variants differ in graph connectivity and available bandwidth resources. Table 4.2 specifies the configured graph connectivities, where the second and third columns specify the number of additionally added edges at the top and at the bottom level, respectively. We define the connectivity $C(\mathcal{N})$ of a network $\mathcal{N}$ as $C(\mathcal{N})$ =

IDENTIFIER	TOP LEVEL	BOTTOM LEVEL	$C(\mathcal{N})$
LowCon	$\lfloor \mu/2 \rfloor$	$\lfloor \nu/3 \rfloor$	0.0265
MediumCon	μ	$\lfloor \nu/2 \rfloor$	0.0311
HighCon	$\lfloor \mu * 2 \rfloor$	ν	0.0422

Table 4.2: Additional edges per connectivity variant.

IDENTIFIER	TOP LEVEL	BOTTOM LEVEL
LowBW	[2, 4] Mbps	[1, 2] Mbps
MediumBW	[3, 5] Mbps	[2, 4] Mbps
HighBW	[4, 6] Mbps	[3, 5] Mbps

Table 4.3: Assigned link capacities.

$\frac{1}{(|\mathcal{V}|-1)\cdot|\mathcal{V}|} \cdot \sum_{v\in\mathcal{V}} deg(v)$, where $deg(v)$ denotes the (edge) degree of node v. Figure 4.5 illustrates three sample topologies for the chosen connectivity values. Table 4.3 lists the bandwidth resources provided to the links. The capacity of each link is randomly drawn from a uniform distribution limited by the indicated intervals. The nine topology variants arise from the cross product of Table 4.2 and Table 4.3. For instance, if we refer to variant *LowConLowBW*, this refers to a topology defined by settings given in the first two lines of Table 4.2 and Table 4.3.

4.3.2 Scenario Description

We evaluate the mentioned algorithms under two different request scenarios. First, the popularity of content is uniformly distributed among all clients. This leads to the fact that most of the content is concentrated at the caches in the core of the network [132]. Second, we distribute the popularity of content according to a Zipf distribution which has the opposite effect (caches at the edges are utilized) [132]. Before simulations are conducted, we perform a limited parameter investigation in Subsection 4.3.3 regarding the duration of the period (the time between two iterations of Algorithm 4.2) for SAF, as Theorem 4.2 suggests that the duration of the period has a significant influence on SAF's performance.

Given a generated topology we randomly placed $a = 100$ clients and $b = 10$ servers in the network. Clients are configured to request content from a single server with a rate of 30 Interests per second, uniformly distributed during a second. This corresponds to a download rate of approximately 1 Mbps, as a single Interest always requests a 4 kiB Data packet. The uniformly distributed request rate represents a steady consumption of content, e.g., streaming video at a certain bit-rate. As already mentioned, for the first scenario we assume a perfect uniform content popularity. A client randomly starts to consume content within the first 30 seconds of the simulation. Each server provides unique content identified by an arbitrary prefix, e.g., */server_id/*. Each node in the network is equipped with a 25 MB large cache which corresponds to approximately 1% of the available content catalogue. We consider this size as sufficient and realistic for the following reason. For instance, consider the *MediumConMediumBW* scenario. A node in this scenario maintains on average 3.11 links (cf. Table 4.2). The maximum link capacity for this scenario is limited by 5 Mbps (cf. Table 4.3), which leads to an average traffic of $3.11 \cdot 5\ Mbps = 15.55\ Mbps$. So, given a node that fully utilizes all links may still cache more than 10 seconds of traffic on average. Considering the findings in [133] that 40% of all cache hits occur in the first 10 seconds, and that clients in the selected scenarios start to request content within a 30 second window, we consider the cache to be sufficiently large. We use the Least Recently Used (LRU) cache replacement strategy [132] and nodes cache every packet they receive.

We further introduce 0, 50, or 100 random link failures during each simulation run. A link failure's point of occurrence and its duration are distributed uniformly. The duration of a link failure is drawn from the interval of $[0, \lfloor \frac{SimTime}{10} \rfloor]$, where $SimTime$ denotes the simulation time. Nodes are configured such that they know all possible routes to any content server at the beginning of each simulation run. This is necessary because many of SAF's competitors require that this information is provided by the routing layer. In this experiment, SAF uses the routing information only as the starting point for the initial FWT. During the simulation no routing updates are enforced, which puts the responsibility to deal with short-term topology changes to the forwarding strategies. We simulate 1800 seconds of network traffic using the aforementioned parameters and conduct 50 runs per setting.

4.3.3 SAF: Influence of the Period on the Performance

In order to investigate the influence of different values for the duration of a period (τ), we considered the scenario *MediumConMediumBW* with 50 link failures given uniform content popularity. Figure 4.6 depicts the 95% confidence intervals (CIs) and the average Interest satisfaction ratio, the cache hit ratio and the hop count for $\tau \in \{0.1,\ 0.5,\ 1.0,\ 2.5,\ 5.0, 10.0, 50.0, 100.0\}$ seconds. The Interest satisfaction ratio denotes the ratio between received Data packets and generated Interests by all clients. The cache hit ratio is averaged over all

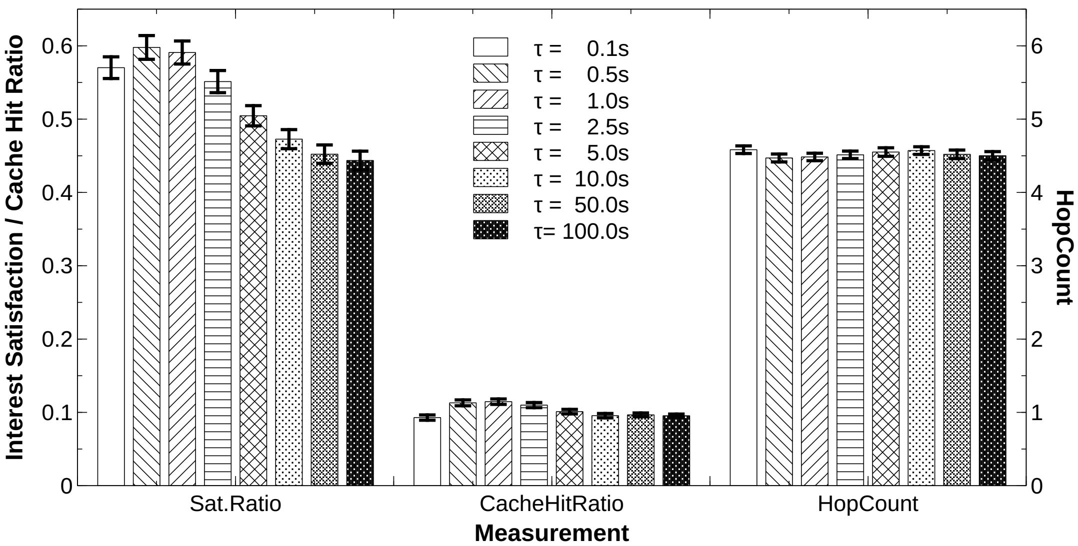


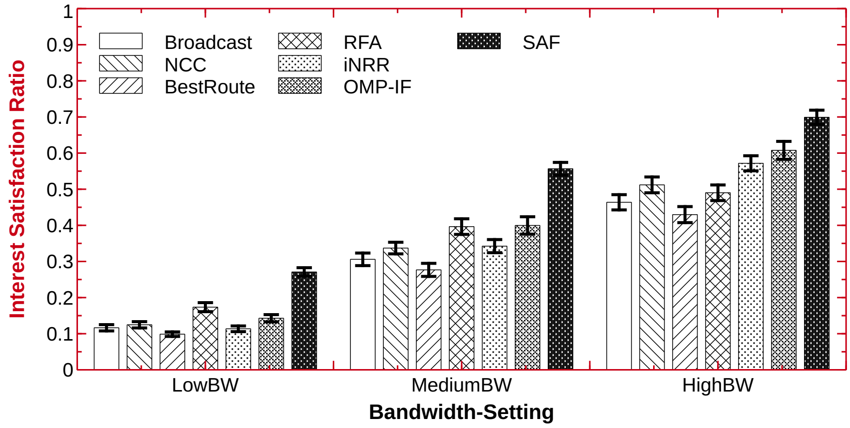
Figure 4.6: Influence of different period durations (τ) on SAF, given scenario *MediumBWMediumCon* with 50 link failures.

network nodes (clients and servers maintain no cache). The hop count provides the number of links a Data packet traversed to satisfy an issued Interest by a client considering cache hits.

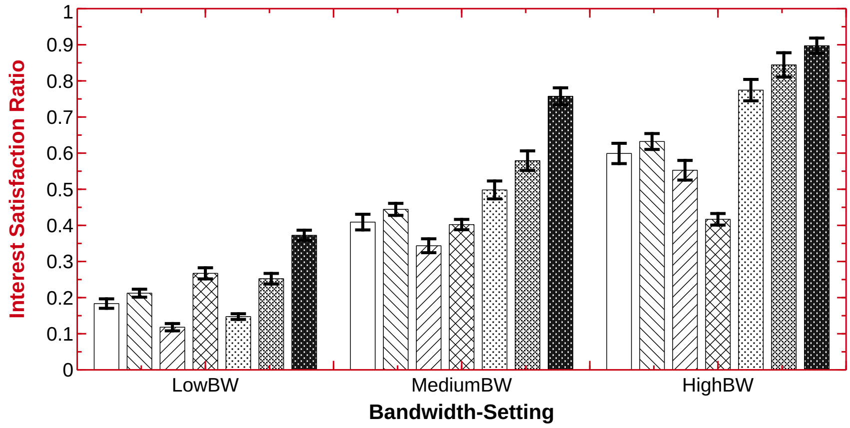
One can observe in Figure 4.6 that in general shorter periods increase the performance regarding the Interest satisfaction ratio. This is consistent with Theorem 4.2, which provides the number of steps for SAF to converge to the optimum (shorter periods, faster steps). However, one can also see that a too small period, e.g. $\tau = 0.1s$, may have a negative influence on the performance. This is due to the fact that the observed traffic within extremely short periods is not representative enough to deduce a suitable decision. The cache hit ratio varies only slightly considering different period durations. Nevertheless, one can observe that the highest ratios are achieved with period durations of few seconds. The average hop count does not change significantly, regardless of the selected period. We use $\tau = 1.0$ *seconds* for all further evaluations, because this setting achieves a good performance with acceptable computational overhead (cf. Subsection 4.2.4).

4.3.4 Performance under Uniform Content Popularity

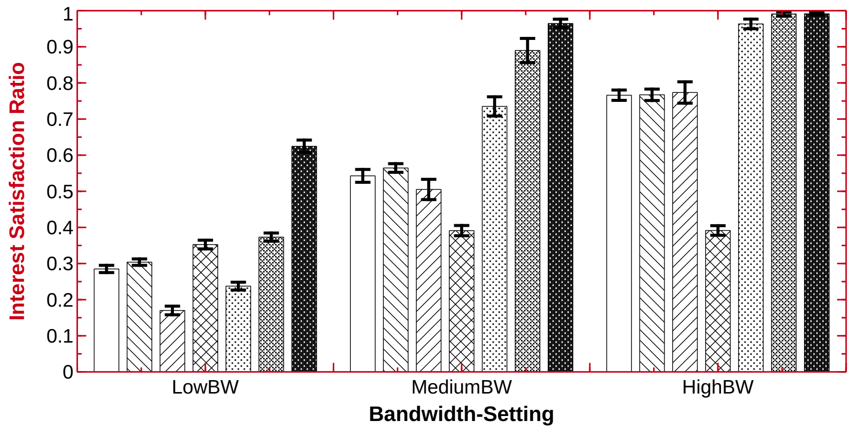
Figures 4.7, 4.8, and 4.9 depict the 95% CIs of the Interest satisfaction ratio for each pairing (topology variant, forwarding strategy) having 0, 50, or 100 random link failures during each simulation. It is evident that SAF outperforms its competitors in nearly all



(a) Results for LowCon

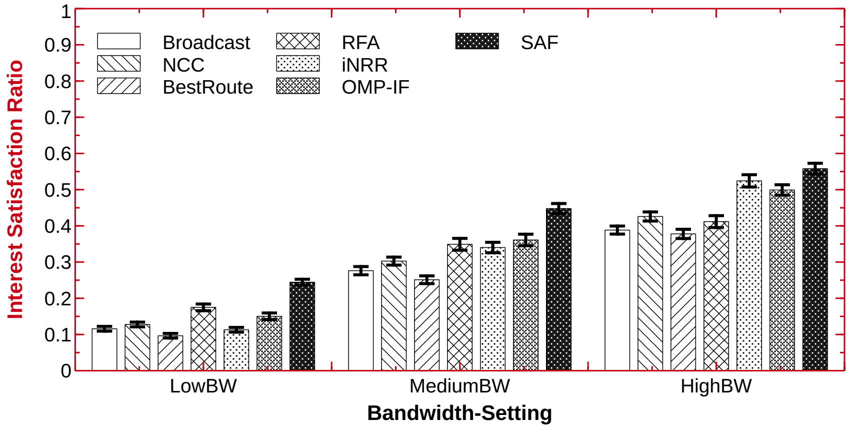


(b) Results for MediumCon

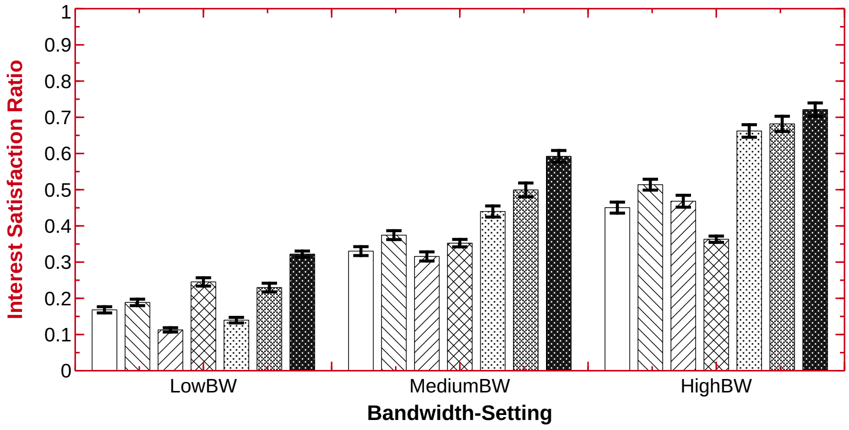


(c) Results for HighCon

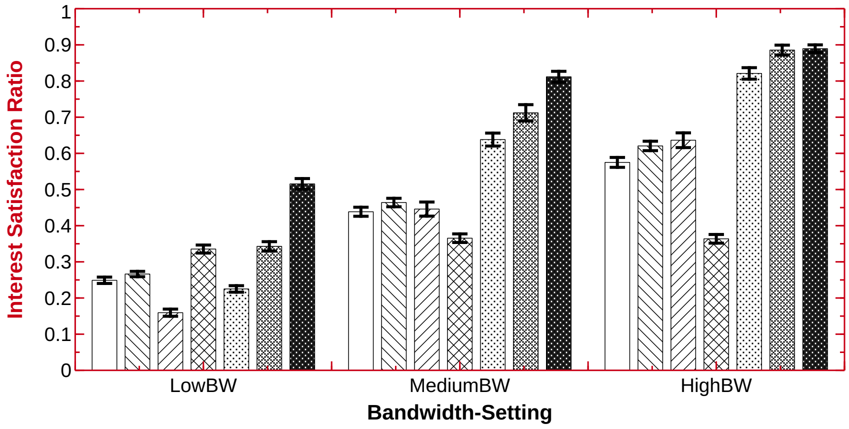
Figure 4.7: Average **Interest satisfaction ratio** and 95% CI with **0 link failures** per simulation run (higher is better) [uniform popularity].



(a) Results for LowCon

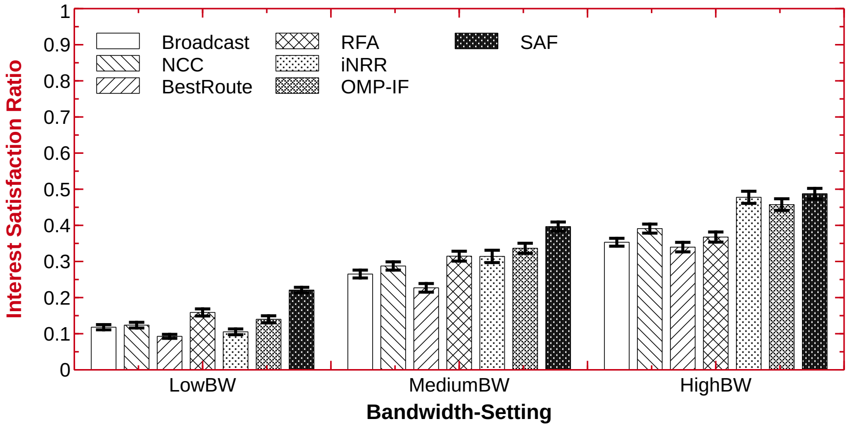


(b) Results for MediumCon

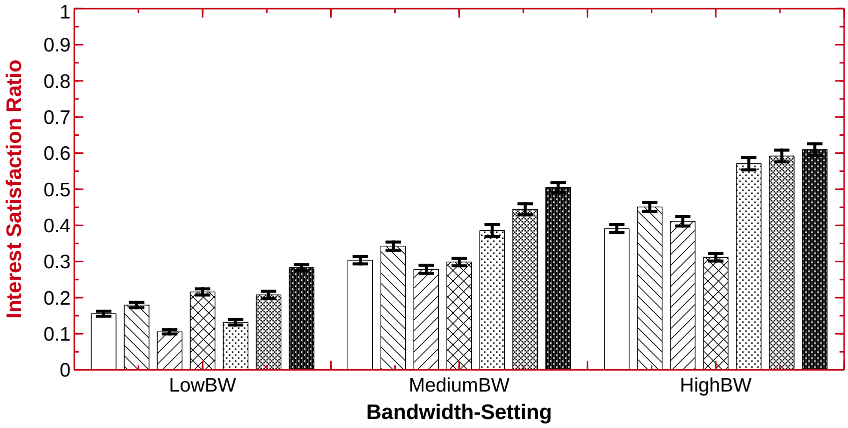


(c) Results for HighCon

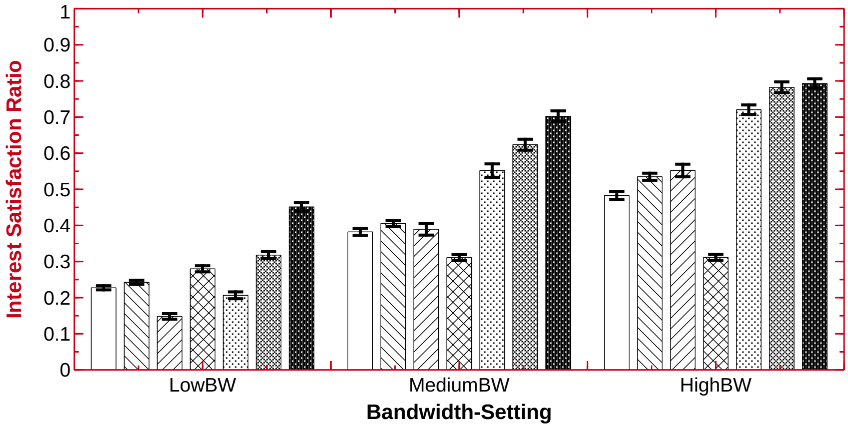
Figure 4.8: Average **Interest satisfaction ratio** and 95% CI with **50 link failures** per simulation run (higher is better) [uniform popularity].



(a) Results for LowCon



(b) Results for MediumCon



(c) Results for HighCon

Figure 4.9: Average **Interest satisfaction ratio** and 95% CI with **100 link failures** per simulation run (higher is better) [uniform popularity].

simulation scenarios in terms of Interests satisfied, regardless of the number of link failures, the connectivity of the underlying network and/or network topology. The two strongest competitors to SAF are iNRR and OMP-IF, especially in scenarios with more resources available. SAF's good performance considering all scenario settings can be explained by its ability to: *i*) smartly circumvent congested nodes and link failures by taking detours into account; *ii*) prevent further transmission of Interests on congested links by redirecting them to the virtual dropping face; *iii*) discover paths to cached content that are not indicated by the FIB.

An increasing number of link failures has a significant negative impact on the Interest satisfaction ratio for all algorithms. For instance, consider Figures 4.7b and 4.9b illustrating the results for 0 and 100 link failures with *medium* graph connectivity. Table 4.4 depicts the absolute and relative performance loss in terms of satisfied Interests for each strategy. SAF has the highest absolute performance loss, however, the relative loss is in a similar range as for its competitors. In these scenarios SAF is not able to fully circumvent link failures due to the resource shortages that are inherent in the selected scenarios. iNRR also suffers from link failures as the oracle used for finding cached replicas does only consider the distance to the content and does not consider link attributes such as reliability or capacity. Interestingly, BestRoute maintains the smallest absolute and relative performance losses. First, these numbers must be considered with respect to the absolute number of satisfied Interests, and second, this can be explained due to the rather short paths used to satisfy Interests (cf. Figure 4.11b), which of course have a lower chance of being affected by randomly emerging link failures. As will be shown later, Broadcast and NCC cannot take advantage of their low hop counts due to extensive Interest replication (cf. Figure 4.16).

The previously discussed findings are also reflected by Figure 4.10 depicting the average cache hit ratio for the scenarios with zero link failures. We omit the results for 50 and 100 link failures because they show a similar picture. In most of the cases SAF outperforms the other algorithms, except for iNRR, which has perfect knowledge about the content chunks in the individual caches. In general, we can observe that the three best performing algorithms SAF, OMP-IF and iNRR concerning the Interest satisfaction ratio also maintain the highest cache hit ratios in most of the cases. However, as already identified in Subsection 3.2.3.b, a high cache hit ratio does not necessarily guarantee to a high Interest satisfaction ratio (cf. setting *LowCon* for BestRoute in Figures 4.7a and 4.10a)

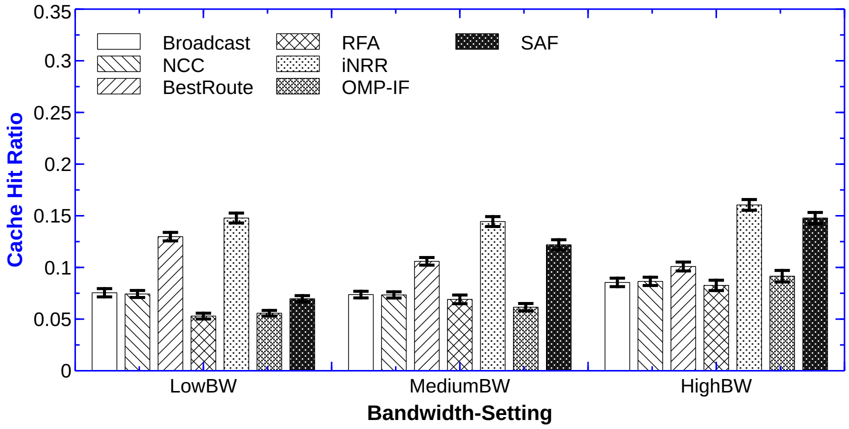
Figure 4.11 illustrates the average hop count per satisfied Interest for the scenarios with

Strategy	LowBW	MediumBW	HighBW
Broadcast	0.024 | 0.133	0.097 | 0.242	0.195 | 0.333
NCC	0.033 | 0.155	0.102 | 0.229	0.181 | 0.287
BestRoute	0.012 | 0.098	0.065 | 0.188	0.135 | 0.247
RFA	0.060 | 0.218	0.109 | 0.267	0.111 | 0.263
iNRR	0.016 | 0.107	0.112 | 0.226	0.204 | 0.263
OMP-IF	0.045 | 0.177	0.135 | 0.233	0.252 | 0.298
SAF	0.090 | 0.240	0.253 | 0.334	0.288 | 0.320

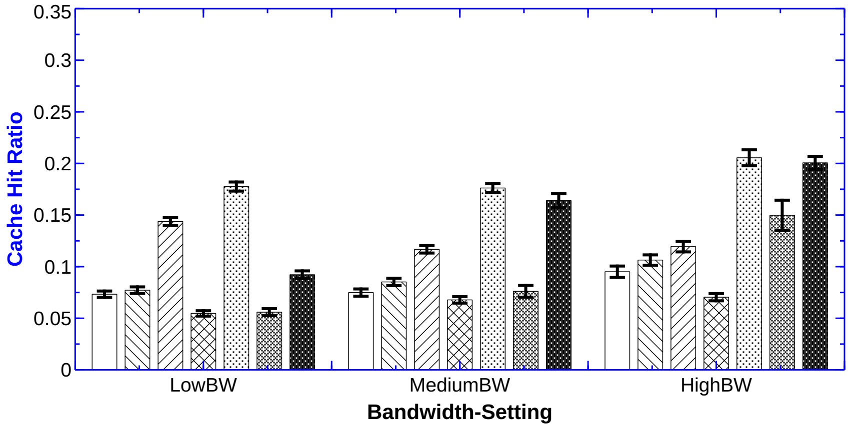
Table 4.4: Absolute (first value) and **relative** (second value) **performance loss** in terms of satisfied Interests comparing scenarios with *medium* graph connectivity having **0** and **100** link failures (cf. Figure 4.7b and 4.9b).

zero link failures. As expected, SAF maintains a higher hop count than some of the other algorithms due to the detours it takes for maximizing the Interest satisfaction according to $\mathcal{M}^T$. The very low hop count of Broadcast and NCC show that both algorithms are able to obtain nearby replicas, however, due to their poor Interest satisfaction and cache hit ratio the low hop count is of minor importance. RFA performs worst concerning the hop count metric, which is definitely due to its principle of load balancing utilizing all available links. This leads to a low cache hit ratio and therefore to a higher hop count compared to its competitors.

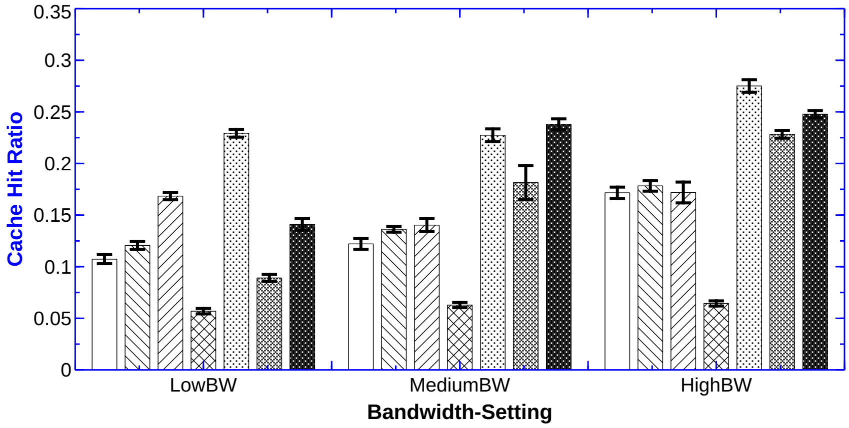
However, especially if the network resources and the connectivity are very limited, RFA is the closest competitor to SAF considering the Interest satisfaction ratio. This is due to the extensive use of all provided routes to the content origins by RFA. Thus, it is able to distribute the load equally on congested nodes/paths. Nevertheless, in better connected scenarios and with higher network resources, RFA is not anymore able to reach the performance of the *standard* algorithms (Broadcast, NCC, BestRoute). RFA uses the number of pending Interests on a face (the one with the lowest number pending is selected) for deciding to which one of the faces in the FIB an Interest shall be forwarded. It does not store any *long term* state or classification of a face. Therefore, the more routes are available to RFA and the higher the network resources, the less RFA is able to determine which faces are the most appropriate for specific content prefixes. For instance, this can be observed in Figure 4.7c. This is also reflected by the low cache hit ratio and high hop count shown in Figures 4.10 and 4.11 indicating that RFA is not able to select those faces that provide good service (e.g., cache hits), but rather takes unnecessary detours.



(a) Results for LowCon

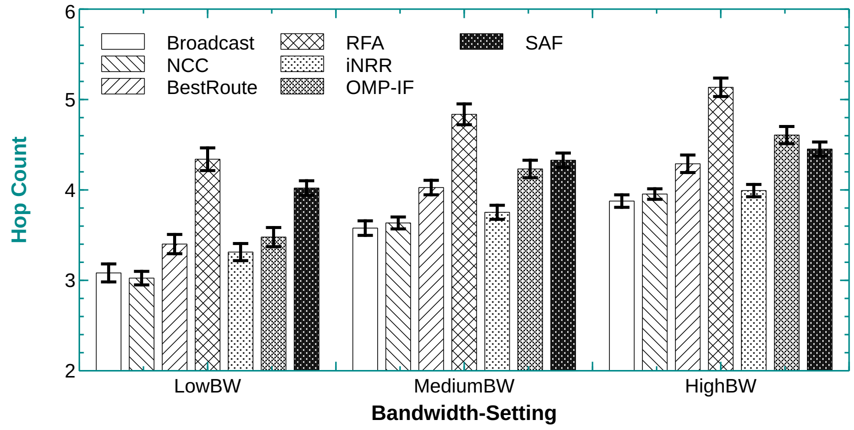


(b) Results for MediumCon

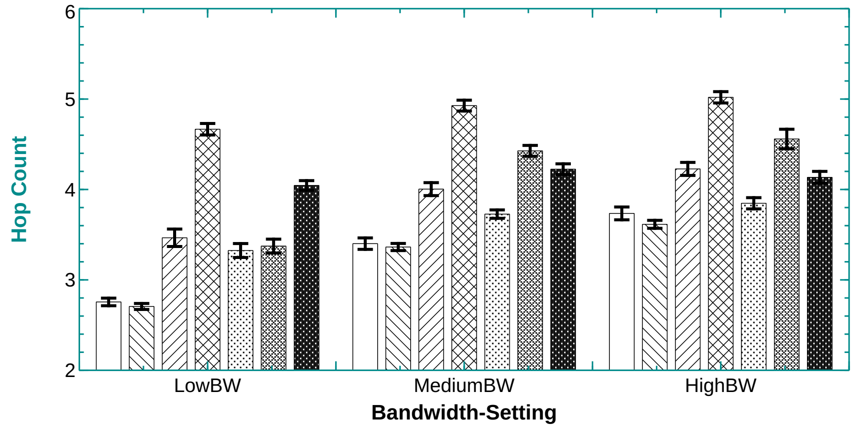


(c) Results for HighCon

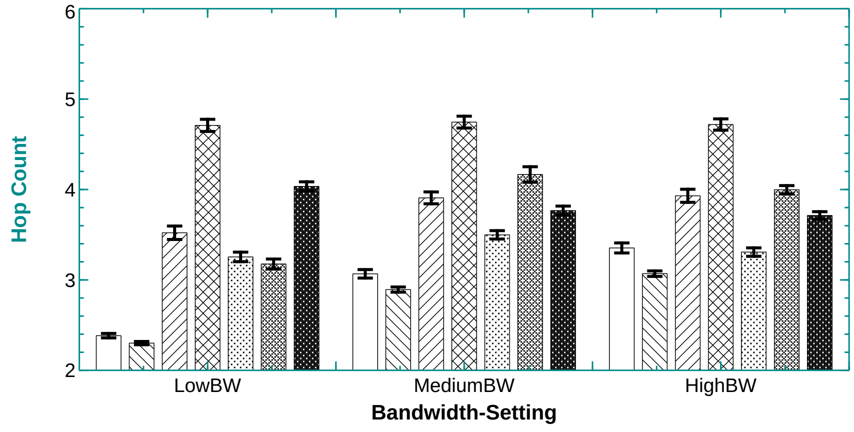
Figure 4.10: Average **cache hit ratio** and 95% CI in the network with **0 link failures** per simulation run (higher is better) [uniform popularity].



(a) Results for LowCon



(b) Results for MediumCon



(c) Results for HighCon

Figure 4.11: Average **hop count** per *satisfied* Interest and 95% CI with **0 link failures** per simulation run (lower is better) [uniform popularity].

In scenarios with plenty resources available, iNRR and OMP-IF are able to catch up with SAF. Figure 4.10 shows that iNRR achieves the highest cache hit ratio, which also is an explanation for iNRR's good performance. The higher hop count compared to Broadcast and NCC (cf. Figure 4.11), can be explained by the deletion of cache entries and the resulting detours iNRR imposes on Interests. In order to understand iNRR's behavior, suppose three nodes $a, b, c \in \mathcal{V}$, whereof c is the content origin. Assume that a receives an Interest and searches for caches that can satisfy it. The nearest cache is b satisfying the aforementioned constraints (cf. Subsection 4.1.3). Therefore, a forwards the Interest to b, but at arrival or on the way to b, b has already evicted the desired content. Thus, the Interest is forwarded to c. Therefore, the triangle inequality $(d(a, c) \leq d(a, b) + d(b, c))$ explains the higher hop count of iNRR compared to Broadcast and NCC. This leads to the assumption that larger caches will improve iNRR's performance.

The strong performance of OMP-IF in scenarios with many resources can be explained by its principle of node disjointness. Subsequent Interests requesting the same content will be forwarded with high probability on the same paths resulting in a high cache hit ratio (cf. Figure 4.10c). Furthermore, the usage of probing Interests (which are broadcasted on all faces in the FIB to identify the best performing path) performs well in these scenarios. However, as resources become scarce (cf. Figure 4.7b) these probing Interests amplify congestion. This leads to the fact that the used paths are switched frequently impairing the cache hit ratio (cf. Figures 4.10b and 4.10c) and the overall performance significantly.

4.3.5 Performance under Zipf-like Content Popularity

In order to compare the results using a uniform content popularity and the results using a Zipf distribution for popularity, we use the same network topologies and simulation parameters as for the previous evaluation (cf. Section 4.3.4). Only the distribution of the content popularity was changed to a Zipf distribution with $\alpha = 0.668$ according to [134]. Figure 4.12 illustrates the content popularity considering 10 items for different values for α, including $\alpha = 0.668$. We again conduct 50 simulation runs for each scenario. In Section 4.3.4 we presented the results for scenarios with 0, 50, and 100 link failures. It can be observed that the performance of the algorithms decreases proportionally with an increase in link failures (cf. Figures 4.7, 4.8 and 4.9). A very similar behavior can be observed for the scenarios where a Zipf-distributed content popularity is assumed. Therefore, and for the sake of clarity we do not depict figures for 50 and 100 link failures, but only focus on

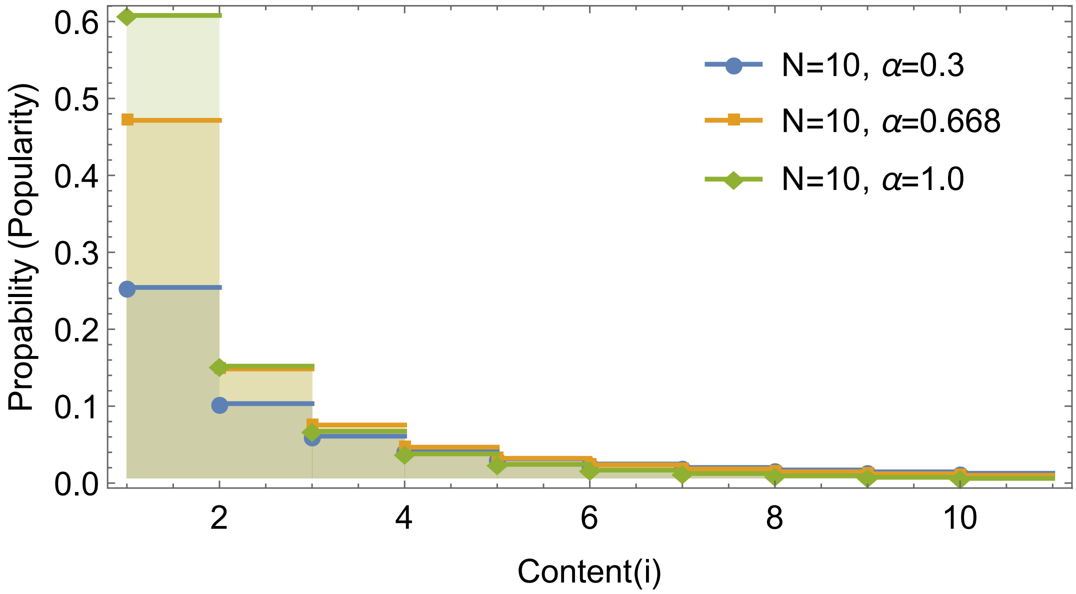
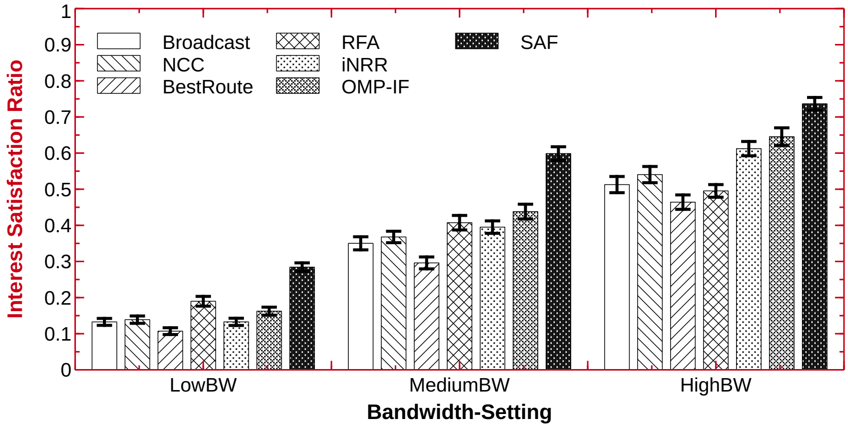


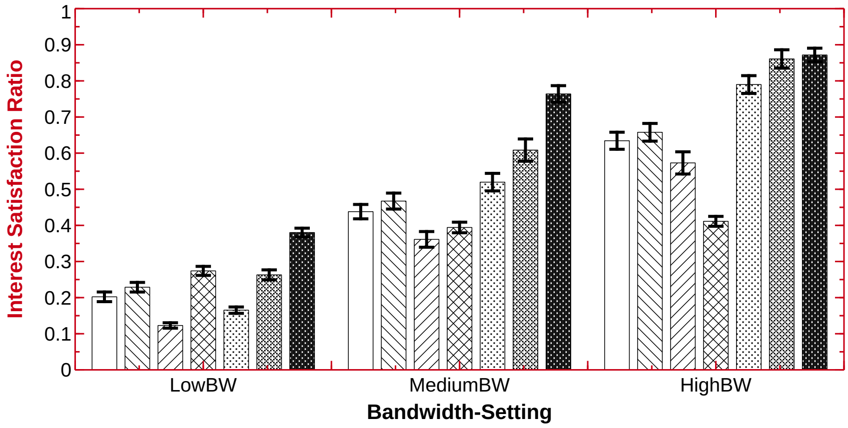
Figure 4.12: Zipf density for 10 content items and varying $\alpha = 0.3$, 0.668 and 1.0.

the scenario with 0 link failures.

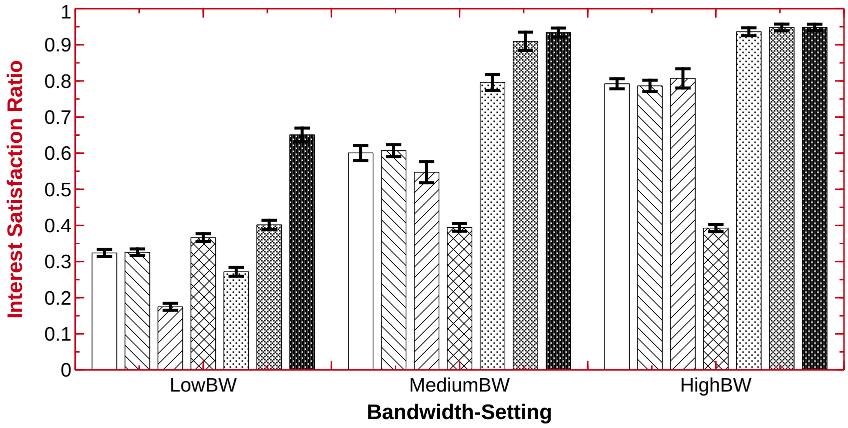
Figures 4.13, 4.14, and 4.15 depict the average Interest satisfaction ratio per node, the average cache hit ratio, and the average hop count per satisfied Interest, respectively. The Interest satisfaction ratio for Zipf-distributed content popularity (cf. Figure 4.13) paints a picture similar to Figure 4.7 for uniform content popularity. SAF outperforms its competitors in terms of satisfied Interests. Again iNRR and OMP-IF are the strongest competitors to SAF and in well-connected scenarios with many resources available these algorithms achieve similar performance. As expected, iNRR beats SAF in terms of cache hit ratio since iNRR is designed to forward Interests to the nearest cache holding the desired Data packet. Broadcast and NCC again provide the lowest average hop count (cf. Figure 4.15) outperforming the other algorithms regarding this metric for the very same reasons as outlined in Section 4.3.4. Although iNRR always maintains a lower hop count and higher cache hit ratio than SAF, SAF outperforms iNRR regarding the number of satisfied Interests. This is due to the extensive usage of multi-path forwarding considering paths which may not provide optimal performance regarding cache hit ratio or hop count, however, maximize the number of satisfied Interests. OMP-IF also focuses on the usage of multiple but node-disjoint paths per content. Nevertheless, as can be seen from Figures 4.14 and 4.15 in most of the cases it maintains a higher hop count and lower cache hit ratio than SAF resulting in a smaller number of overall satisfied Interests.



(a) Results for LowCon

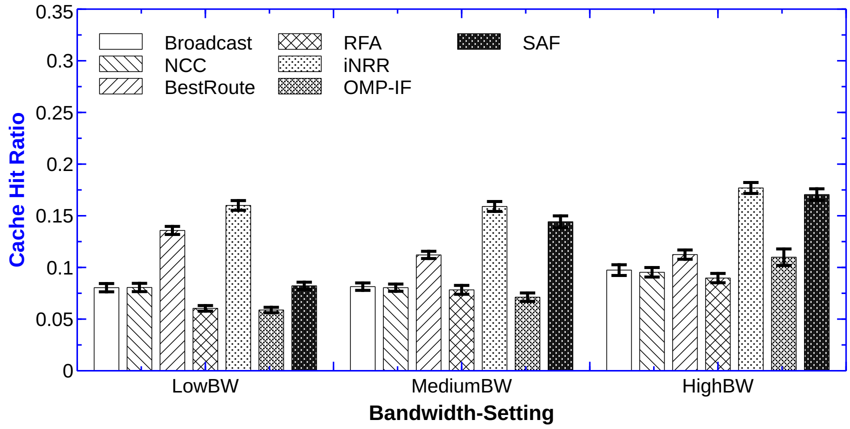


(b) Results for MediumCon

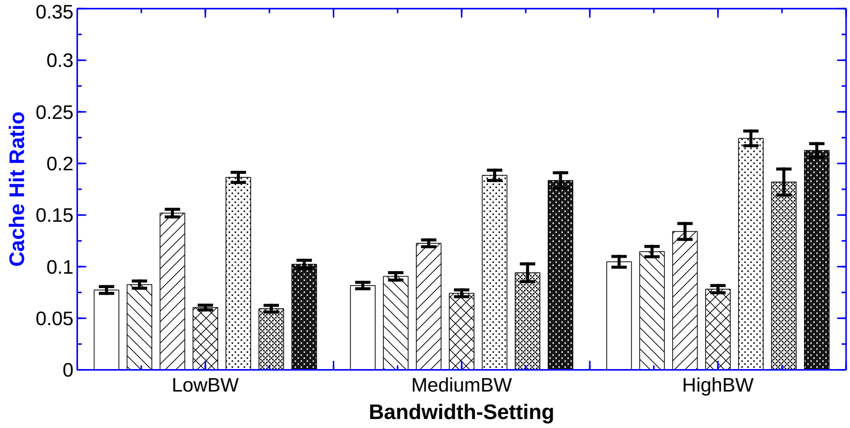


(c) Results for HighCon

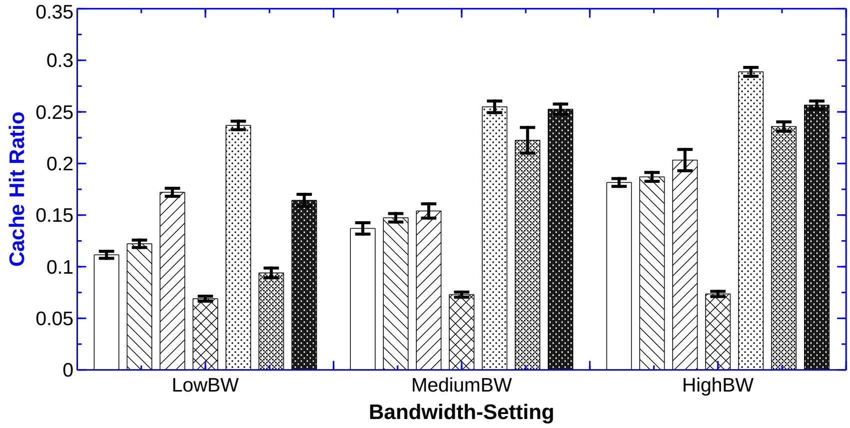
Figure 4.13: Average **Interest satisfaction ratio** and 95% CI with **0 link failures** per simulation run (higher is better) [zipf popularity].



(a) Results for LowCon

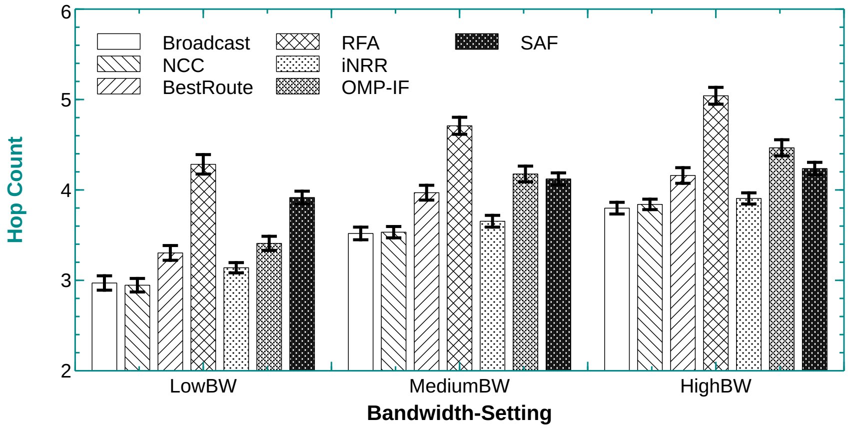


(b) Results for MediumCon

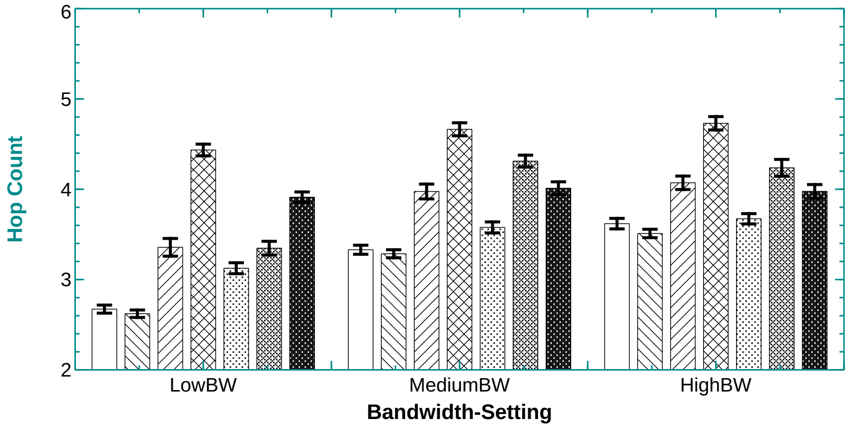


(c) Results for HighCon

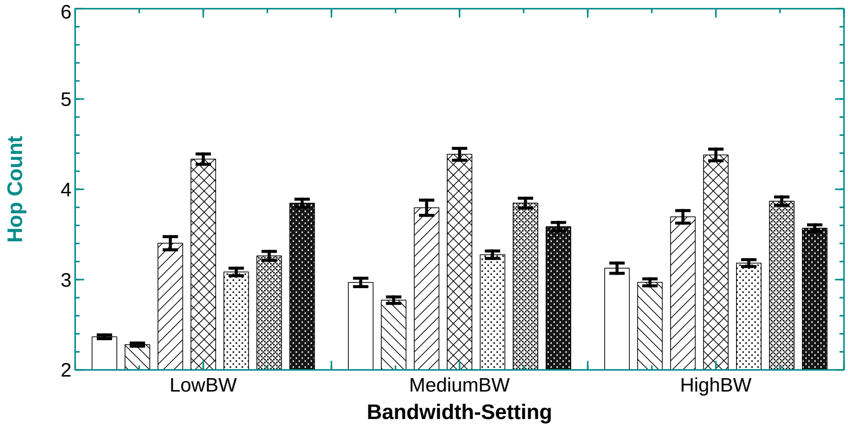
Figure 4.14: Average **cache hit ratio** and 95% CI in the network with **0 link failures** per simulation run (higher is better) [zipf popularity].



(a) Results for LowCon

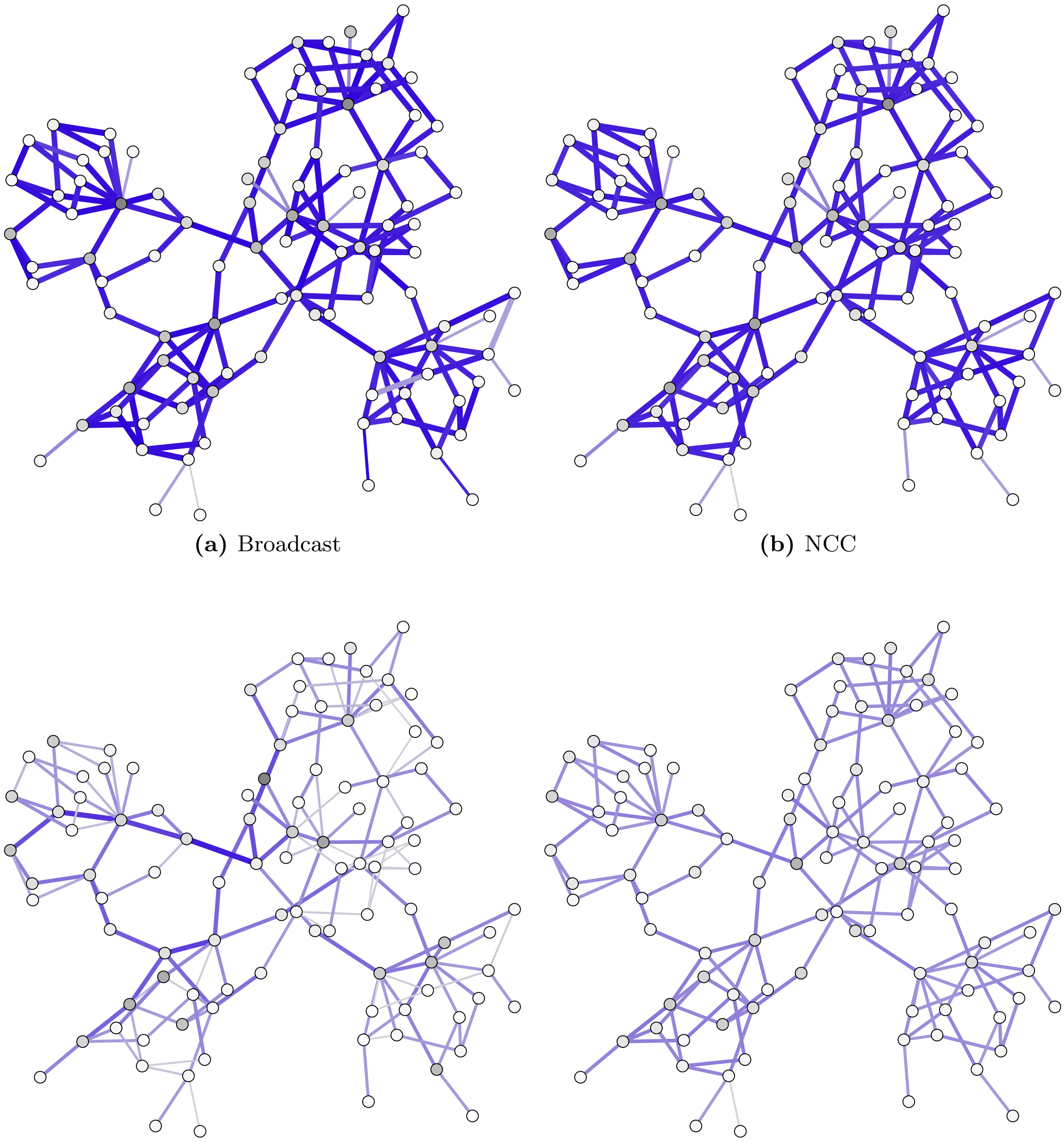


(b) Results for MediumCon



(c) Results for HighCon

Figure 4.15: Average hop count per *satisfied* Interest and 95% CI with **0 link failures** per simulation run (lower is better) [zipf popularity].

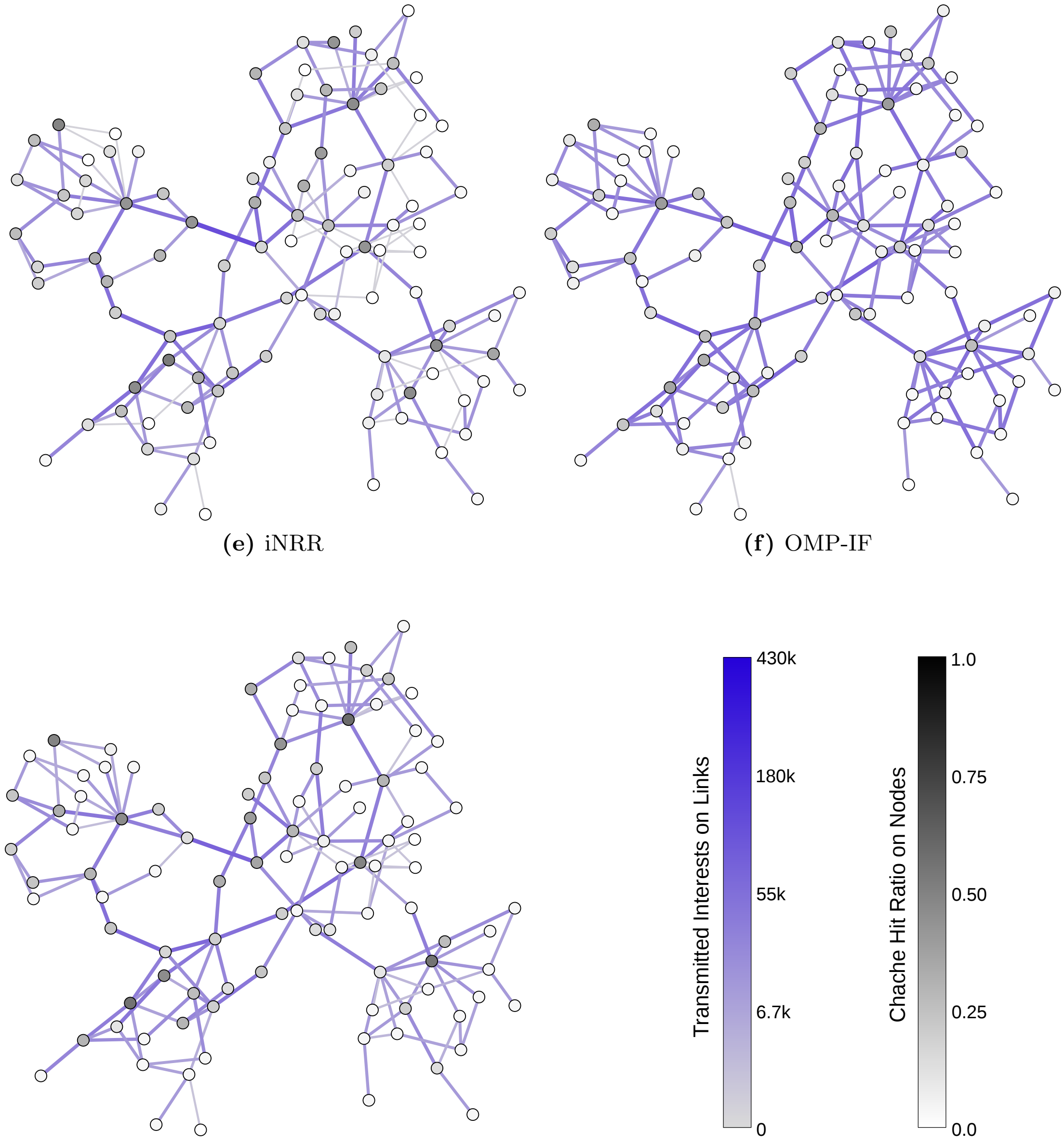


(a) Broadcast **(b)** NCC

(c) BestRoute **(d)** RFA

Figure 4.16: Heat maps illustrating the distribution of the transmitted Interests and cache hits. The wider and the more saturated a link, the more Interests are transmitted over the given link. The darker the coloring of a node, the higher the cache hit ratio. *The figure is continued on page 113.*

Figure 4.16 depicts the distribution of the transmitted Interests and cache hits in the network. For each of the evaluated algorithms, a single simulation run is depicted considering a fixed scenario (*MediumConHighBW*; cf. Figure 4.5b) with a Zipf-distributed



(e) iNRR **(f)** OMP-IF

(g) SAF **(h)** Key

Figure 4.16 (cont.)**:** Heat maps illustrating the distribution of the transmitted Interests and cache hits. The wider and the more saturated a link, the more Interests are transmitted over the given link. The darker the coloring of a node, the higher the cache hit ratio.

content popularity. This figure illustrates how differently the forwarding algorithms behave. Broadcast replicates Interests on every node and pushes them to all neighbouring nodes (cf. Figure 4.16a). NCC shows a similar behaviour due to its inherent and unlimited retransmissions for late packets on alternative paths. BestRoute focuses on the shortest paths from a client to the content provider and, therefore, the caches on the shortest paths are heavily used. RFA performs a kind of load balancing distributing Interests equally on all routes. Thus, it achieves a low cache hit ratio and Interest satisfaction ratio (cf. Figure 4.13). iNRR tries to maximize the cache hit ratio by explicitly preferring nearby cached Data packets in the network instead of retrieving them from the content origin. This circumstance is reflected in Figure 4.16e showing several nodes with high cache hit ratios (dark coloring). OMP-IF also achieves an acceptable cache hit ratio due to its principle of node disjointness. Frequent probing and path switching leads to a similar load balancing behavior as in RFA. SAF is able to provide both, effective load balancing and high cache hit ratios at the relevant nodes (cf. Figure 4.16g). Compared to OMP-IF, which balances the load very equally on all available links, SAF focuses on paths that provide better performance, leading to a better cache hit ratio and an overall better performance.

4.4 Conclusion and Future Work for SAF

This chapter introduced Stochastic Adaptive Forwarding (SAF), a novel forwarding strategy for Named Data Networking. SAF provides probability-based forwarding on a per-content/per-prefix basis. It ensures effective forwarding with incomplete and/or invalid routing information, and resolves unexpected network topology changes without relying on the routing plane. The extensive usage of multi-path transmission is the foundation for SAF's performance. SAF is flexible in that it can be configured with various measures defining the forwarding objectives. The effectiveness of content distribution of SAF was illustrated by conducting extensive simulations using the ns-3/ndnSIM framework. We presented the throughput-based measure $\mathcal{M}^T$ optimizing the Interest satisfaction ratio in a given network. In both scenarios, considering uniform and Zipf-like content popularity, SAF outperforms all competitors. Our results can be reproduced as we provide an open source implementation of SAF and its competitors (iNRR, OMP-IF, RFA) at `http://icn.itec.aau.at`.

SAF performs also excellent in real multimedia streaming scenarios. The evaluation of DAS-based multimedia streaming conducted in the previous chapter (cf. Subsection 3.2.3.b) showed that SAF is able to outperform existing strategies in such scenarios. Recall the Multi-Commodity Flow Problem stated by LP 3.1 and Algorithm 3.1. They provide the theoretical upper bound(s) for NDN-based multimedia streaming considering multi-path forwarding and idealized caching. The results showed that SAF's competitors are barely able to approach the first upper bound (cf. blue lines in Figures 3.7, 3.8, and 3.9), which considers multi-path forwarding only. However, SAF fulfills the expectations of NDN-based content delivery considering the upper bound that incorporates both in-network caching and multi-path forwarding. In particular, this is the case when sufficiently large caches are deployed in the network (cf. red lines in Figures 3.7, 3.8, and 3.9).

SAF optimizes its forwarding decisions with respect to a measure. The only assumption that SAF has on the measure is that it allows classifying Interests into satisfied and unsatisfied ones. Thus, SAF can be easily tailored to specific use cases. For example, if Interests shall not exceed a specific hop count, one may introduce a measure that classifies Interests as satisfied if the hop count is below this specific threshold. SAF further allows making decisions on a context and content level. Since SAF keeps track of the forwarding probabilities for all content prefixes, it is possible to adapt the forwarding probabilities with respect to content prefixes. This can be used to prioritize traffic from specific content prefixes. For instance, (real-time) multimedia traffic that takes the same path through the network as a bulk file transfer can be assigned higher priority. In this case, one may introduce a weighting such that the probability of the virtual dropping face (F_D, cf. Subsection 4.2.1) for bulk file transfer is increased. This leads to a decrease of the forwarding probability of F_D for multimedia traffic and, therefore, to an increase of the forwarding probability on the desired faces. We are going to investigate mechanisms for content- and context-aware forwarding in NDN in Chapter 5. There, we present an approach that uses SAF to realize relative content priorities considering the context of the consumed contents/applications.

Although we analytically indicated the resource consumption (complexity) of SAF (cf. Section 4.2), we think it is important to investigate this in more detail. Therefore, we present a framework for an NDN-based testbed in Chapter 6. We will use the proposed testbed to evaluate the practicality of SAF and its competitors, and present further performance measurements with respect to CPU load and power consumption. Furthermore, we will investigate to which extent the observed performance in a simulated environment of SAF can be accomplished in real deployment scenarios on physical devices. Future work concerning SAF could include an investigation of a more sophisticated probing mechanism. The currently proposed probing mechanism is quite simple and basically distributes a given probe equally on all existing faces with unknown performance. The size of a probe is determined by the ratio of satisfied and unsatisfied Interests with respect to a given prefix. Very recently, Lei et al. [135] proposed an entropy-based forwarding strategy for NDN, which is an interesting concept. One approach to enhance SAF's probing mechanism could include an entropy-based probing solution that does not split the probe uniformly, but based on the entropy of the relevant faces.

CHAPTER 5

Towards a Context-Aware Forwarding Plane in NDN Supporting QoS

"Sometimes your greatest strength can emerge as a weakness if the context changes."

— Harsha Bhogle, 1961*

The emergence of Information-Centric Networking (ICN) provides considerable opportunities for context-aware data distribution. While packet forwarding in classical IP-based networks is basically predetermined by routing, ICN/NDN foresees an adaptive forwarding plane considering the requirements of network applications. As research in the area of adaptive forwarding is still at an early stage, most of the work so far focused on providing the basic functionality, rather than on considering the available context information to improve Quality of Service (QoS). This chapter investigates to which extent existing forwarding strategies take account of the available context information and can therefore increase service quality. Furthermore, this chapters shows how the design of SAF (cf. Chapter 4) can be used to incorporate context information in the forwarding plane. The chapter is structured into four sections. First, in Section 5.1 we provide a motivation for context-/content-aware forwarding in ICN/NDN. Then, in Section 5.2 we discuss the existing forwarding strategies with respect to their context-/content awareness. Subsequently, Section 5.3 examines a typical multimedia content distribution scenario encompassing different user applications (Voice over IP, video streaming, and classical data transfer) with varying demands (context). We will investigate how well the applications' requirements are met by the existing forwarding strategies by assessing the achieved user satisfaction. Furthermore, we introduce the presented scenario's context to NDN's forwarding plane by implementing a context/content-aware adaptation mechanism for SAF's adaptation engine (cf. Figure 4.1). Finally, we conclude our findings in Section 5.4, and provide an outlook on this challenging topic.

5.1 Motivation for QoS in NDN's Forwarding Plane

It is evident that today's Internet has to deal with a large variety of applications. Each application requires a specific kind of service, which in general is opaque to the network layer of classical IP-based network infrastructures. On the one hand, there are applications like Voice over IP (VoIP) that demand low latency and jitter while consuming a moderate amount of bandwidth resources. On the other hand, there are applications such as video streaming that consume a large amount of bandwidth resources, but have relaxed requirements with respect to delay and jitter (due to possible pre-buffering capabilities). For an Internet Service Provider (ISP) it would be beneficial to become aware of the individual application requirements, so each transmitted packet can be delivered within the given constraints. This would lead to a high consumer satisfaction, while providing the opportunity to deliver packets in a cost-effective way (e.g., using cheap, but high-delay paths, for classical data traffic).

While the necessary context information is not easily accessible at IP's network layer (in particular without additional mechanisms, such as Differentiated Services [136, 137], and even then only partially), the emergence of Information-Centric Networking (ICN) is turning the tide. As discussed in Subsection 2.1.3, there are multiple approaches to implement ICN. For this chapter the understanding of ICN is again coincident with the approach of Named Data Networking (NDN) [15]. Recall from Section 2.2.1, in NDN, data is requested by its name following a strictly receiver-driven communication model. The employed names may include additional data providing information about the application and/or the requested content characteristics. For instance, name prefixes could be used as indicator for content/application requirements (real-time, delay-tolerant, etc.) (cf. Subsection 2.3.1). This chapter does not investigate on how (or where) to represent this information, but rather focuses on how it can be used by NDN's forwarding plane to support Quality of Service (QoS) considerations.

NDN's adaptive forwarding plane is outlined in [47] and its basic technical background has been discussed in Subsection 2.2.4 and 4.1.1. Recall that the forwarding plane is flexible, and already a variety of strategies [38, 95, 108, 127] have been proposed to realize adaptive forwarding. In general each of the strategies pursues a different objective (maximize throughput, minimize delay, load balancing, optimal cache utilization, etc.). This raises the following two questions:

Q1: *Are forwarding strategies in NDN able to consider different application requirements, or do they just focus on their narrow forwarding objective, oblivious to additional context information provided in the network?*

Q2: *Does context awareness in NDN's forwarding plane support the fulfillment of Quality of Service demands?*

In order to answer these questions, we investigate and evaluate selected forwarding strategies with respect to three different types of applications: *VoIP*, *video streaming* and *classical data transfer*. Furthermore, we discuss the adaptation engine of Stochastic Adaptive Forwarding (SAF) (the forwarding strategy introduced in Chapter 4) in detail. The adaptation engine enables the consideration of extensive context information in the forwarding plane. We present how this is possible and show the benefits of this approach by conducting network simulations using the network simulation framework ns-3/ndnSIM v2.0 [37]. Therefore, we define an evaluation scenario encompassing the aforementioned user applications. We measure the relevant QoS parameters for each application type and use them as input for existing models to obtain the actual user satisfaction. For the application VoIP we use the well known E-Model [138] to map the QoS values to a Mean Opinion Score (MOS). In order to measure the user satisfaction of video streaming, we use the model suggested in [139]. This model considers relevant parameters for adaptive video streaming (video quality, playback interruptions, etc.) and provides a measure for the user satisfaction. Concerning the classical data transfer application, we consider the average achieved goodput as a sufficiently accurate measure. The detailed investigation of the presented scenario will provide substantial insight into how the individual forwarding strategies and their context awareness influence the user satisfaction with respect to a concrete application. The results indicate that context awareness in the forwarding plane is relevant and can lead to QoS improvements (cf. Subsection 5.3.4).

5.2 Forwarding Strategies: How Context-Aware Are They?

This section discusses existing forwarding strategies with respect to their context awareness concerning their decision policies. Recall that an NDN node may register multiple outgoing faces per name prefix, and the forwarding strategy is responsible to select the "best" outgoing face(s) for each individual Interest that traverses a node. To obtain a good decision it is necessary to consider a certain amount of context available in the forwarding plane. The context may include the individual face performances with respect to delay, capacity, packet loss and/or also content specific information; e.g., the name prefix */voip* may tell that this is a delay sensitive VoIP packet. The majority of the existing strategies focus solely on face performance metrics, potentially leading to non-optimal decisions. In the following, we discuss the most prominent strategies available for NDN and sketch their basis of decision making (context) on what is (are) the "best" face(s).

Broadcast [38] is a simple strategy that does not consider any context except the information that is provided by the FIB. Interests are forwarded on all outgoing faces that are registered for the given name prefix. It is evident that this strategy causes a lot of unnecessary overhead that increases with the number of available nodes/links leading to bad performance in scenarios with scarce resources. However, if sufficient resources are available, Broadcast guarantees optimal delivery in terms of many relevant performance metrics (e.g., hop count and data delivery delay).

BestRoute [38] forwards Interests to the lowest-cost (e.g., in terms of hop count) upstream face. This context information has to be provided by a third component. For instance, this could be the routing plane that provides the number of hops (distance) per outgoing face to the content origin stored in the Routing Information Base (RIB). This setting is the default configuration as implemented in the NFD [38]. Alternative configurations are possible considering the imposed latency, or also a combination of both. Furthermore, the strategy considers "short-term context“ information about the individual faces on a node. If Interests forwarded to the best face are not satisfied, BestRoute temporarily tries to avoid it preferring other sub-optimal faces regarding the concerned metric. This allows dealing with congestion and link-failures on individual links.

NCC [38] forwards Interests to those faces that provide the lowest delay for receiving data packets. Instead of BestRoute that depends on context information provided by a third component, NCC gathers and maintains the latency statistics on its own making it independent from third components (besides the FIB). Therefore, it measures for each Interest the time it takes to satisfy it on the outgoing face and continuously updates the selected face's assessment. While BestRoute temporarily avoids non-performing faces, NCC uses a different approach. It maintains an individual clock per forwarded Interest. If the time to satisfy an Interest defers too strongly from the expected delivery time, the strategy actively performs a retransmission on an alternative sub-optimal face (if the lifetime of the Interest has not already been expired).

The **Request Forwarding Algorithm (RFA)** [95] is part of a set of optimal dynamic multi-path congestion control protocols and request forwarding strategies derived from multi-commodity flow problems. This algorithm monitors for each available prefix the number of PIT entries per face listed in the FIB. Using this information, the forwarding probability of a face is determined by a weight that is actually a moving average over the reciprocal count of the PIT entries. Simply put, the relevant context for RFA is the current utilization of a face as indicated by the PIT. As RFA does not consider any further context information, traffic of different applications will be heavily scattered among the available faces. This can be extremely unfavourable, especially for applications like VoIP, which have low latency and low jitter demands.

On-demand Multi-Path Interest Forwarding (OMP-IF) [127] suggests the forwarding of Interests on node-disjoint paths. In the proposed approach each network node may only use a single face (from the FIB) for forwarding per name prefix to ensure node disjointness. The consumer nodes trigger the multi-path transmission by utilizing a weighted round-robin mechanism based on the path delays, distributing Interests over multiple faces. If a router encounters packet loss on the selected face, subsequent Interests of the corresponding name prefix are broadcasted. The first face satisfying a broadcasted Interest is selected for further transmission. Generally, this strategy puts the decision to use multiple paths close to the user, allowing the application context to decide if multiple paths should be used. For instance, this could be beneficial for voice VoIP applications to ensure low jitter. However, the strict rule of switching the path as soon as a single packet loss is encountered in the network may lead to negative side effects in congestion scenarios (e.g., frequent path switching).

Stochastic Adaptive Forwarding (SAF) as presented in Chapter 4 imitates a self-adjusting water pipe system, intelligently guiding and distributing Interests through the network. SAF uses the returning Data packets as input for deriving a probability distribution, determining forwarding probabilities for the individual face per name prefix. Additionally, SAF employs a virtual (dropping) face that is used to encounter congestion by actively dropping Interests that could cause congestion. The aforementioned probabilities are stored in a table, which are modified by an adaptation engine. SAF is the first strategy that enables an operator to incorporate specific context-aware considerations. Operators may define per name prefix the conditions for considering an Interest as satisfied (e.g., returning Data packets must be retrieved in less than d milliseconds). Furthermore, SAF

Strategy \ Context	FACE METRICS/ CONSIDERATIONS	CONTENT/APPLICATION CHARACTERISTICS	OTHER CONTEXT INFORMATION
Broadcast	—	—	FIB entries*
NCC	Interest/Data satisfaction delay†	—	FIB entries*
BestRoute	use of suboptimal face(s) if best face fails†	—	FIB entries*, routing cost metric*
RFA	face utilization† in terms of PIT entries	per-prefix forwarding probabilities†	FIB/PIT entries*
OMP-IF	network nodes use only one face per prefix (node disjointness)†	user/application decides on multi-path usage*	FIB entries*
SAF	face reliability (satisfied versus unsatisfied Interests)†	per-prefix forwarding probabilities†, pre-prefix constraints*	FIB entries*, relative priorities per prefix*, dropped traffic share†

Table 5.1: The table summarizes (in a broad sense) the context information that is used to derive the forwarding decisions for each strategy. Entries labeled with $\star$ are gathered/provided/specified by third parties, while entries labeled with $\dagger$ are obtained/maintained/ensured by the strategies themselves.

enables operators to set different weights per name prefix considering their relative importance. The adaptation takes account of the employed context information and arranges the forwarding probabilities in the table accordingly. We present a possible extension to SAF in Subsection 5.3.3 that considers context information of the presented scenario (cf. Subsection 4.3.2).

The discussion on context awareness of the forwarding strategies show that there has not been invested much effort in considering content/application characteristics and further context information in NDN's forwarding plane. The results are summarized in Table 5.1 Only SAF foresees a dedicated mechanism (adaptation engine, cf. Figure 4.1) that can be fed with additional application information. In the following we investigate if this is an advantage compared to the other strategies.

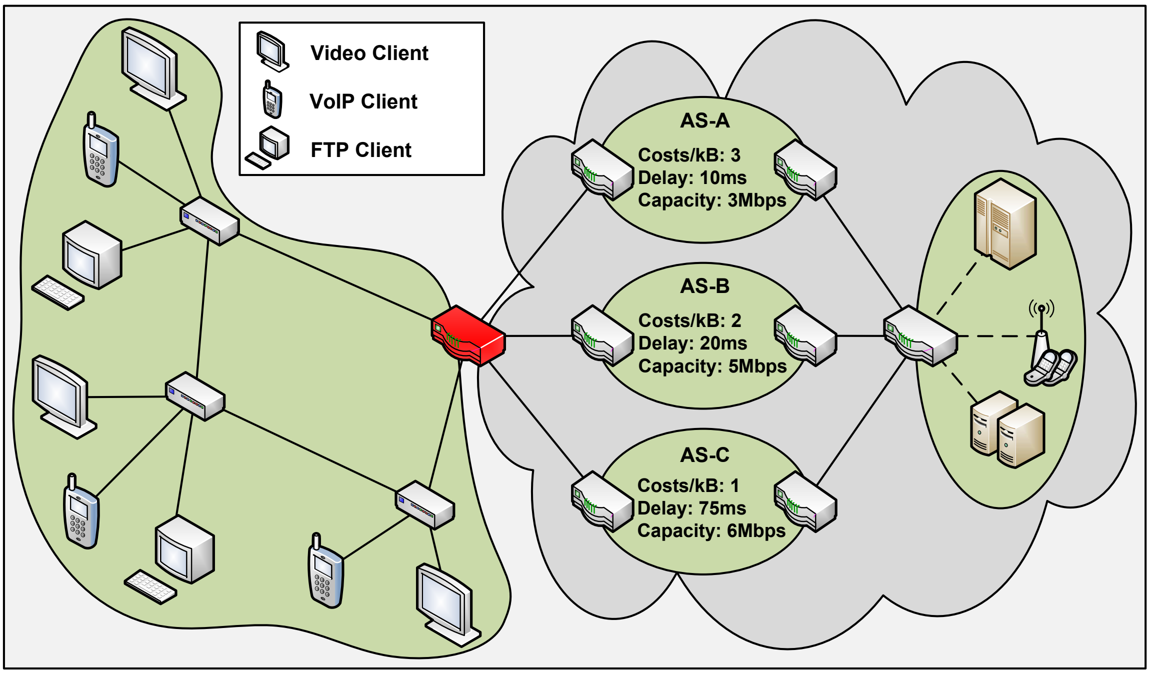


Figure 5.1: The evaluation scenario considers three different applications: video streaming, VoIP, and data (file) transfer.

5.3 NDN's Context-Aware Forwarding Plane: Does it Enhance QoS?

This section investigates the opportunities of introducing context awareness in the forwarding plane to enhance QoS considering different applications. As a first step, a scenario is sketched that encompasses network applications with varying requirements (cf. Subsection 5.3.1). Then Subsection 5.3.2 discusses the employed evaluation methods. Afterwards, in Subsection 5.3.3 we introduce the necessary context information into SAF's adaptation engine. Finally, in Subsection 5.3.4 the results are presented that have been obtained by conducting network simulations using ns-3/ndnSIM [37].

5.3.1 Scenario Description

Figure 5.1 depicts the proposed content-delivery scenario in an NDN network. The left hand side of the figure illustrates an autonomous system (AS) consisting of several routers and clients representing a typical ISP access network. The clients employ three applications: video streaming, VoIP, and classical file transfer (FTP). The relevant files or communication partners for the clients are located in another AS that is depicted on the right hand side of Figure 5.1. As there is no direct connection between those two autonomous systems, traffic has to be routed/forwarded through the networks of other autonomous systems. We assume that the ISP access network maintains service level agreements (SLAs) with three other (intermediate) providers (AS-A, AS-B, AS-C) that may bridge the gap between the networks. AS-A provides a capacity of 3 Mbps for traffic, imposes a one-way delay of 10 ms and charges 3 cost units per transmitted kilobyte. The respective values for AS-B are 5 Mbps, 20 ms delay, 2 cost units, and for AS-C 6 Mbps, 75 ms delay, 1 cost unit. In the given scenario all remaining links provide sufficient resources and low latencies having negligible negative impact on data transmission. Therefore, the investigated strategies are only deployed on the router highlighted by the red color in Figure 5.1 that connects the access network with the autonomous systems AS-A, AS-B and AS-C. All other routers always use the BestRoute strategy for forwarding Interests.

The video streaming application on the clients is implemented by the principles of Dynamic Adaptive Streaming (DAS) (cf. Subsection 2.3.3). We employ a buffer-based adaptation logic [106] (cf. Subsection 3.2.1.b) for the DAS clients as suggested by our results obtained in Subsection 3.2.3.b. The video content for this experiment is taken from the SVC-DASH dataset [104], which provides four short movies with an average duration of about 12 minutes. We concatenated the short sequences to obtain content with an average duration of about 48 minutes. Recall from Subsection 3.2.1.a that in [104] the video content is encoded in various variants. A variant defines the encoding parameters as well as the scalability domains (temporal, spatial, quality). For this evaluation we have chosen the variant providing Signal-to-Noise-Ratio (quality) scalability only, with a segment duration of 2 seconds. The chosen content is provided using a base layer and two enhancement layers. The base layer (henceforth denoted as $L0$) has an average bitrate of approximately 640 kbps. The first enhancement layer ($L1$) has a bitrate of approximately 355 kbps. The second enhancement layer ($L2$) has an average bitrate of approx. 407 kbps ($L0 + L1 + L2 \approx 1400$ kbps). The request pattern of the video streaming clients is bursty, thus challenging the individual forwarding strategies. There are two reasons for the bursty request pattern of the video streaming clients: *i*) The implemented buffer-based adaptation logic requests segments in a consecutive manner without requesting different segments in parallel (cf. Figure 3.3). This allows the adaptation logic to continuously assess the current situation after each successfully download segment adapting to changes with minimal delay. However, this leads to short request pauses between two consecutive requests of segments with fast ramp up at the

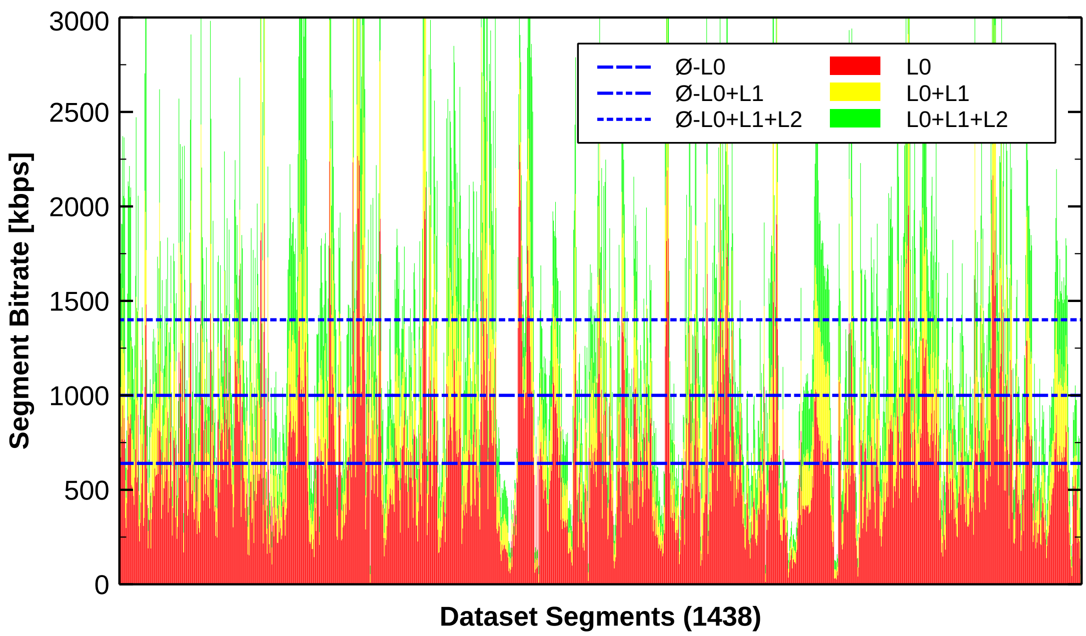


Figure 5.2: The **cumulative segment bitrates** of the available representations for the concatenated video sequence taken from [104] including their averages.

beginning of a download phase (many Interests are issued in the beginning) and a slow ramp down at the end of a segment download phase. *ii*) The segments and their individual layers may significantly vary in size or bitrate, respectively, depending on the video content. The cumulative segment bitrates for the concatenated video sequence is depicted in Figure 5.2. The average segment bitrates and their standard deviations for the individual layers are given in Table 5.2.

Layer	Avg Segment Bitrate	Standard Deviation
(L0) Base Layer	643.13 kbps	390.67 kbps
(L1) Enhancement Layer 1	355.02 kbps	219.78 kbps
(L2) Enhancement Layer 2	407.83 kbps	266.62 kbps

Table 5.2: The **average segment bitrates** and their standard deviations for the individual layers of the concatenated video sequence taken from [104].

While IP-based VoIP applications are implemented in a push-based fashion, this is not possible in NDN. NDN follows a strictly pull-based communication approach. A Data packet can only be delivered in response to an Interest. However, this would roughly double the typical one-way delay imposed by push-based approaches impairing user Quality of Experience (QoE) significantly. Pioneering work by Jacobson et al. [140] exploits NDN's hierarchical name space to request data that does not yet exist. This resolves the problem by transmitting an Interest to the producer before the corresponding VoIP Data packet is created. The Interest keeps pending until the data is generated, and then the Data packet can be transmitted immediately to the requesting consumer application. Note that in this case the client application needs to know the packet generation rate, which can be easily predicted from the employed audio codec (e.g, G.711 [141]). We implement the VoIP client following these principles and choose G.711 with Packet Loss Concealment (PLC) as audio codec. Furthermore, we assume a fixed jitter buffer of 50 ms and employ a fixed audio codec bitrate of 64 kbps, which is a typical setting for G.711. The request pattern of the VoIP clients is steady with low demands on bandwidth capacity. However, the VoIP communication demands that the forwarding strategies choose only low latency links. For instance, routing VoIP traffic through AS-C will lead to late packets having a negative impact on user satisfaction.

The File Transfer Protocol (FTP) client is modeled by a simple consumer/producer application provided by [37]. We assume that each FTP client requests a large file from a server demanding roughly 3 Mbps of bandwidth capacity. The traffic pattern of this application is steady (constant bitrate), with very low demands only focusing on throughput.

As this chapter focuses on the capabilities of context-aware forwarding, we assume that each of the clients depicted in Figure 5.1 requests unique content. Therefore, caching can be disabled eliminating possible side effects on the data transmission performance. This allows to perfectly investigate the pure forwarding capabilities of the investigated strategies.

5.3.2 Evaluation Method

To evaluate the performance of the different forwarding strategies and their context-aware capabilities, we investigate each of the applications using a different evaluation method that is focused on the user's satisfaction:

5.3.2.a Simplified E-Model for VoIP

For the evaluation of the VoIP performance we use a simplified version of the E-Model [138, 142] that is applicable when only packet loss and delay impairments are considered. This model provides the so called R-value that can be mapped to a Mean Opinion Score (MOS) using Equation 5.1. The R-value is basically calculated as $R = 93.2 - I_d - I_{e-eff}$, where

I_d represents the impairments caused by the one-way delay d as defined in Equation 5.2. I_{e-eff} is defined by Equation 5.3 considering the codec impairments (I_e) and the influence of the packet loss percentage (P_{pl}) and its burstiness ($BurstR$) on the employed codec. As in our scenario VoIP clients use G.711 with PLC, we have $I_e = 0$ and $B_{pl} = 34$, which denotes the codec's built-in packet loss concealment ability. We determine the so-called burst ratio ($BurstR$) as suggested in [138]. $BurstR$ is specified in Equation 5.4, where p denotes the transitional probability from packet loss to no loss, and q the transitional probability from no loss to loss. We obtain p and q by employing the 2-state Markov model as depicted in Figure 5.3. For more details on the E-Model and the selected parameters we refer the interested reader to ITU recommendations G.107 [142], G.711 [141], G.113 [143], and to [138].

$$MOS = \begin{cases} 1 & \text{if } R \leq 0, \\ 1 + 0.035R + R \cdot (R-60)(100-R) \cdot 7 \cdot 10^{-6} & \text{if } 0 < R < 100, \\ 4.5 & \text{if } R > 100. \end{cases} \tag{5.1}$$

$$I_d = \begin{cases} 0.024d & \text{if } d < 177.3, \\ 0.024d + 0.11(d - 177.3) & \text{if } d \geq 177.3. \end{cases} \tag{5.2}$$

$$I_{e-eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{\frac{P_{pl}}{BurstR} + B_{pl}} \tag{5.3}$$

$$BurstR = \frac{1}{p+q} \tag{5.4}$$

5.3.2.b User Satisfaction Model for Video Clients

To evaluate the performance of the video streaming applications, we use the proposed user satisfaction model from [139]. The model considers the video quality, the quality variations

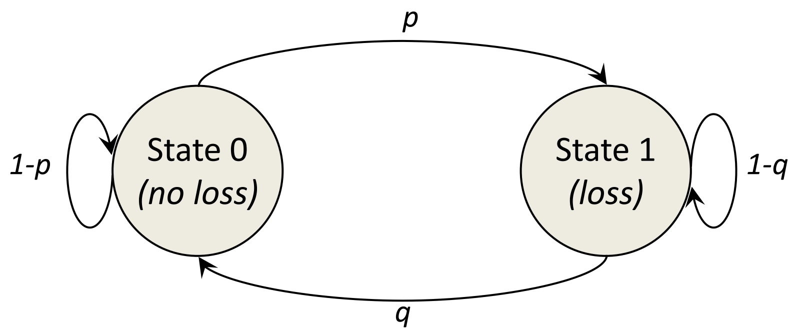


Figure 5.3: The 2-state Markov model used to calculate the parameters p (transitional probability from *no loss* to loss) and q (transitional probability from *loss* to *no loss*) for the so-called burst ratio (cf. Equation5.4) [138].

and the re-buffering events:

$$Q = \underbrace{\sum_{k=1}^{K} q(R_k)}_{\text{video quality}} - \lambda \underbrace{\sum_{k=1}^{K-1} |q(R_{k+1}) - q(R_k)|}_{\text{quality variations}} - \mu \underbrace{\sum_{k=1}^{K} b(R_k)}_{\text{re-buffering time}} \tag{5.5}$$

K denotes the number of segments received by a DAS client. R denotes the available representations (in our case $R = \{L0, L0 + L1, L0 + L1 + L2\}$) and R_k denotes the consumed representation of segment k on a client. $q(R_k)$ denotes the quality of a segment, which we define as the average corresponding representation bitrate. $b(R_K)$ denotes the number of re-buffering seconds (video stalling time) before segment R_k is ready for play-out. λ and μ are non-negative parameters modeling the particular influence of quality variations and re-buffering events, respectively. In [139] different combination of values for λ and μ are suggested. We vary the parameters in the suggested ranges and provide a 3D plot indicating the user satisfaction considering various user preferences (more/less sensitive to re-buffering events versus more/less sensitive to representation switches). Please note that we re-formulated the model slightly because in [139] it is used as an objective function for a maximization problem imposing a more complex formulation.

5.3.2.c Download Bitrate for FTP Clients

The performance of the FTP clients is measured by the pure download bitrate. We consider the goodput (throughput minus overhead) as the relevant performance indicator. Other metrics, like packet delivery delay or jitter have no noticeable influence for the user.

5.3.3 Adding Context Information to SAF

As already mentioned, SAF's design allows considering additional context information. In the following we show how to introduce the scenario's context into SAF. We consider the following facts as relevant context information: VoIP clients cause the lowest amount of traffic, however, their users suffer heavily from packet loss and late packets (cf. Equations. 5.1, 5.2, and 5.3). Therefore, VoIP traffic should be prioritized at the forwarding plane, especially on low latency links. Let us assume that we order the importance of the content classes as: $VoIP >_c video >_c data$, where $a >_c b$ denotes that a should be prioritized over b. Then our objective is to introduce a weighting mechanism that ensures this ordering by influencing the performance assessment of faces. The weights shall be selected in such a way that low priority content will be dropped in favour of high priority content due to SAF internals (using the dropping face F_D). In the following we show how this demand can be introduced and realized by SAF.

Recall that SAF periodically performs updates of the forwarding probabilities for each registered name prefix considering all faces $F_i \in \mathcal{F}$ (cf. Subsection 4.2.4). The updates maximize a combined measure $\mathcal{M}$ that is given by Equation 5.6. $\mathcal{I}_n$ denotes the set of Interests in a given period n. $S_{F_i}(\mathcal{I}_n)$ denotes a measure for the satisfied Interests, $U_{F_i}(\mathcal{I}_n)$ a measure for the unsatisfied Interests. SAF's default configuration defines $S_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is satisfied by a Data packet on } F_i\}|$ and $U_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is not satisfied on } F_i\}| \; \forall F_i \in \mathcal{F} \setminus \{F_D\}$, where F_D denotes the virtual dropping face that satisfies Interests by definition [108]. Thus, SAF maximizes the throughput for the individual name prefixes.

$$\mathcal{M} = \sum_{F_i \in \mathcal{F}} M_{F_i}(\mathcal{I}_n) = \sum_{F_i \in \mathcal{F}} (S_{F_i}(\mathcal{I}_n) - U_{F_i}(\mathcal{I}_n)) \tag{5.6}$$

In order to consider the relative content priorities, we are going to re-define the measure $U_{F_i}(\mathcal{I}_n)$. This leads us to SAF-CAA (<u>C</u>ontex-<u>A</u>ware <u>A</u>daptation) which introduces an additional weight ω to the definition of U_{F_i}. We re-define $U'_{F_i}(\mathcal{I}_n) := U_{F_i}(\mathcal{I}_n) \cdot \omega_{F_i} := \omega_{F_i} \cdot |\{j \in \mathcal{I}_n : j \text{ is not satisfied by a Data packet on } F_i\}|$. As ω_{F_i} is chosen differently for each content (and also specifically per face as indicated by the subscript F_i, which will be disscues later in more detail) it can be used to realize the prioritization. In general, if network congestion is encountered, a large ω leads to earlier pro-active packet dropping of the corresponding content. In the following we provide a rationale for the selection of the individual weights based on the definition of a *reliable* face [108]:

According to SAF, a reliable face can be defined as given by Definition 5.1. So, a face is reliable if and only if $t_{c_j} \leq \frac{S_{F_i,c_j}}{S_{F_i,c_j}+U_{F_i,c_j}}$, where c_j denotes the j-th content (SAF operates on a per content basis). t_{c_j} denotes the reliability threshold for the j-th content. This dynamic threshold tells us how much reliable traffic a specific content currently has, assuming SAF has already converged. The threshold itself does only vary between t_{min,c_j} and t_{max,c_j}. We want to introduce an ordering on the specific contents such that we can influence the reliability accordingly. Therefore, we use the weight ω_{F_i,c_j} for the calculation of the reliability of face F_i for a given content c_j as follows: $t_{c_j} \leq \frac{S_{F_i,c_j}}{S_{F_i,c_j}+U_{F_i,c_j} \cdot \omega_{F_i,c_j}}$.

Definition 5.1. A face $F_i \in \mathcal{F}$ is *reliable* for content $c_j \in \mathcal{C}$ if and only if $R_{F_i,c_j} := \frac{S_{F_i,c_j}}{S_{F_i,c_j}+U_{F_i,c_j} \cdot \omega_{F_i,c_j}} \geq t_{c_j}$, where t_{c_j} denotes the reliability threshold of SAF (cf. Table 4.1) for the j-th content, and $\omega_{F_i,c_j} \in [1, \infty[$ is a weight for the j-th content on F_i.

Definition 5.2 defines an *adaptable set* of contents with respect to a given face $F_i \in \mathcal{F} \setminus \{F_D\}$. Here, the term *adaptable* indicates that for these contents/prefixes adaptation among the individual columns of SAF's FWT is reasonable (cf. blue arrows in Figure 4.1 suggesting the *shifting* of forwarding probabilities among different prefixes). Colloquially, we define a set of contents as *adaptable* with respect to F_i if that face performs reliable data delivery for every content in that set. Furthermore, a secondary condition must hold that says that there must exist unsatisfied traffic for each content in that set. In the following we discuss why contents in an adaptable set have to satisfy both requirements so that SAF's adaptation engine may perform effective adaptations.

Definition 5.2. We define a set of contents $\mathcal{C}_{F_i}$ as *adaptable* with respect to a given face F_i if and only if for a given content catalogue $\mathcal{C}$, $\mathcal{C}_{F_i}$ contains only contents that are considered as reliably transmitted on F_i, although a number (greater than 0) of Interests cannot be satisfied by F_i, $\mathcal{C}_{F_i} := \{c_j \in \mathcal{C} | R_{F_i,c_j} \geq t_{c_j} \wedge U_{F_i,c_j} > 0 \wedge S_{F_i,c_j} > 0\}$ $\forall F_i \in \mathcal{F} \setminus \{F_D\}$.

The reason for solely encapsulating contents that are reliably transmitted via F_i in an adaptable set is apparent when considering the following. SAF has learned the optimal amount of traffic for each content in the adaptable set that should be forwarded via F_i considering the individual content's reliability threshold t_{c_j}. Only for these contents SAF has converged (cf. Theorem 4.2). Therefore, adaptation among the individual contents in the adaptable set is possible. Contents that are not part of an adaptable set are not transmitted reliably on F_i. Thus, SAF is in an unstable state concerning these contents making meaningful adaptation decisions impossible. The secondary condition ensures that only contents are considered for adaptation that have unsatisfied Interests on F_i. Therefore, only contents interfering with each other on the given face F_i are considered for adaptation. Given an adaptable set and an ordering of contents, we can state Theorem 5.1 that provides a rationale for selecting the individual weights ω.

Theorem 5.1. Given an *adaptable set* $\mathcal{C}_{F_i}$ with $|\mathcal{C}_{F_i}| > 1$ and an ordering on the contents $(\mathcal{C}, >_c)$ (denoting the importance of the contents), one obtains the following result for determining the weights such that the ordering of the contents is established by SAF on F_i:

$$\begin{aligned}
&\forall c_k, c_j, c_m \in \mathcal{C}_{F_i}, c_k >_c c_j >_c c_m : \\
&\omega_{F_i,c_j} \in \left]\max\left\{\frac{S_{F_i,c_j} \cdot U_{F_i,c_k} \cdot \omega_{F_i,c_k}}{S_{F_i,c_k} \cdot U_{F_i,c_j}}, \frac{S_{F_i,c_j} \cdot (1-t_{c_j})}{U_{F_i,c_j} \cdot t_{c_j}}\right\}, \frac{S_{F_i,c_j} \cdot U_{F_i,c_m} \cdot \omega_{F_i,c_m}}{S_{F_i,c_m} \cdot U_{F_i,c_j}}\right[
\end{aligned} \tag{5.7}$$

We further have two degrees of freedom, ω_{F_i,c_1} and $\omega_{F_i,c_{|C_{F_i}|}}$.

Proof [Theorem 5.1] It suffices to show that the selection of the weights ω_{F_i,c_j} as given in Theorem 5.1 results in the same ordering of reliabilities as the ordering of the contents such that $R_{F_i,c_1} > R_{F_i,c_2} > \cdots > R_{F_i,c_{n-1}} > R_{F_i,c_n}$, where R_{F_i,c_1} corresponds to the reliability of the most important content with $c_1 >_c c_2 >_c \cdots >_c c_{n-1} >_c c_n$. SAF then shifts the traffic such that the faces become reliable again. The lower bound can be easily obtained from the required ordering of the reliabilities and that we want to force SAF to shift traffic from the given face to other faces or the dropping face. Therefore, we have according to Definitions 5.1 and 5.2:

$$\begin{aligned}
&\forall c_j, c_{j+1} \in \mathcal{C}_{F_i}, c_j >_c c_{j+1} : \\
\frac{S_{F_i,c_j}}{S_{F_i,c_j} + U_{F_i,c_j} \cdot \omega_{F_i,c_j}} &> \frac{S_{F_i,c_{j+1}}}{S_{F_i,c_{j+1}} + U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}} \\
S_{F_i,c_j} &> \frac{S_{F_i,c_{j+1}} \cdot S_{F_i,c_j} + S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j}}{S_{F_i,c_{j+1}} + U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}} \\
S_{F_i,c_j} \cdot S_{F_i,c_{j+1}} + S_{F_i,c_j} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}} &> S_{F_i,c_j} \cdot S_{F_i,c_{j+1}} + S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j} \\
S_{F_i,c_j} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}} &> S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j} \\
\omega_{F_i,c_{j+1}} &> \frac{S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j}}{S_{F_i,c_j} \cdot U_{F_i,c_{j+1}}}
\end{aligned}$$

We derive the upper bound analogously as follows:

$$\forall c_{j+1}, c_{j+2} \in \mathcal{C}_{F_i}, c_{j+1} >_c c_{j+2} :$$

$$
\begin{aligned}
\frac{S_{F_i,c_{j+1}}}{S_{F_i,c_{j+1}} + U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}} &> \frac{S_{F_i,c_{j+2}}}{S_{F_i,c_{j+2}} + U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}}} \\
S_{F_i,c_{j+1}} &> \frac{S_{F_i,c_{j+2}} \cdot S_{F_i,c_{j+1}} + S_{F_i,c_{j+2}} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}}{S_{F_i,c_{j+2}} + U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}}} \\
S_{F_i,c_{j+1}} \cdot S_{F_i,c_{j+2}} + S_{F_i,c_{j+1}} \cdot U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}} &> S_{F_i,c_{j+1}} \cdot S_{F_i,c_{j+2}} + S_{F_i,c_{j+2}} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}} \\
S_{F_i,c_{j+1}} \cdot U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}} &> S_{F_i,c_{j+2}} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}} \\
\omega_{F_i,c_{j+1}} &< \frac{S_{F_i,c_{j+1}} \cdot U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}}}{S_{F_i,c_{j+2}} \cdot U_{F_i,c_{j+1}}}
\end{aligned}
$$

Taking into account the requirement that the reliability should be below the corresponding reliability threshold. It follows that:

$$
\begin{aligned}
&\forall c_j \in \mathcal{C}_{F_i}, c_j >_c c_{j+1} : \\
&\frac{S_{F_i,c_j}}{S_{F_i,c_j} + U_{F_i,c_j} \cdot \omega_{F_i,c_j}} < t_{c_j} \\
S_{F_i,c_j} &< t_{c_j} \cdot S_{F_i,c_j} + t_{c_j} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j} \\
\frac{S_{F_i,c_j}}{t_{c_j}} &< S_{F_i,c_j} + U_{F_i,c_j} \cdot \omega_{F_i,c_j} \\
-U_{F_i,c_j} \cdot \omega_{F_i,c_j} &< S_{F_i,c_j} - \frac{S_{F_i,c_j}}{t_{c_j}} \\
-U_{F_i,c_j} \cdot \omega_{F_i,c_j} &< \frac{S_{F_i,c_j} \cdot (1 - t_{c_j})}{t_{c_j}} \\
\omega_{F_i,c_j} &> \frac{S_{F_i,c_j} \cdot (1 - t_{c_j})}{t_{c_j} \cdot U_{F_i,c_j}}
\end{aligned}
$$

$\omega_{F_i,c_1} \geq 1$ can be chosen arbitrarily and by calculating all the lower bounds we finally get to $\omega_{F_i,c_{|\mathcal{C}_{F_i}|}}$. The upper bound for $\omega_{F_i,c_{|\mathcal{C}_{F_i}|}}$ can be chosen arbitrarily big with the restriction that it has to be bigger than the indicated lower bound. Then we may calculate the upper bounds for all other weights and choose the weights within the determined bounds. This will instruct SAF to prioritize the contents as given by the ordering $(\mathcal{C}_{F_i}, >_c)$ on the network level. This concludes the proof. ■

Algorithm 5.1 outlines an algorithmic procedure to obtain the weights ω_{F_i,c_j}. The algorithm shall be executed after each iteration of SAF's FWT updates (cf. Algorithm 4.2) assuming that the contents are ordered by importance (descending) in $\mathcal{C}_{F_i}$. Recall that the following operations are executed for every face $F_i \in \mathcal{F} \setminus \{F_D\}$ (cf. Alg 5.1, line 1). First, we initialize the lower bound for the content with the highest priority with 1 (cf. Alg. 5.1, line 2). This, will also be the first degree of freedom deduced in Theorem 5.1 (cf. Alg. 5.1, line

Algorithm 5.1 Context-Aware Adaptation for SAF

1: **for each** $F_i \in \mathcal{F} \setminus \{F_D\}$ **do**
2: $\quad w^{(L)}_{F_i,1} \leftarrow 1$
3: $\quad$ **for** $1 \leq j \leq |\mathcal{C}_{F_i}| - 1$ **do**
4: $\quad\quad w^{(L)}_{F_i,c_{j+1}} \leftarrow \max \left\{ \dfrac{S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot w^{(L)}_{F_i,c_j}}{S_{F_i,c_j} \cdot U_{F_i,c_{j+1}}}, \dfrac{S_{F_i,c_{j+1}} \cdot (1 - t_{c_{j+1}})}{U_{F_i,c_{j+1}} \cdot t_{c_{j+1}}} \right\}$
5: $\quad$ **end for**
6: $\quad w^{(U)}_{F_i,c_{|\mathcal{C}_{F_i}|}} \leftarrow w^{(L)}_{F_i,c_{|\mathcal{C}_{F_i}|}} + 1$
7: $\quad$ **for** $|\mathcal{C}_{F_i}| \geq j \geq 2$ **do**
8: $\quad\quad w^{(U)}_{F_i,c_{j-1}} \leftarrow \dfrac{S_{F_i,c_{j-1}} \cdot U_{F_i,c_j} \cdot w^{(U)}_{F_i,c_j}}{S_{F_i,c_j} \cdot U_{F_i,c_{j-1}}}$
9: $\quad$ **end for**
10: $\quad \omega_{F_i,1} \leftarrow w^{(L)}_{F_i,1}$
11: $\quad \omega_{F_i,|\mathcal{C}_{F_i}|} \leftarrow w^{(U)}_{F_i,c_{|\mathcal{C}_{F_i}|}}$
12: $\quad$ **for** $2 \leq j < |\mathcal{C}_{F_i}|$ **do**
13: $\quad\quad \omega_{F_i,c_j} \leftarrow \dfrac{w^{(U)}_{F_i,c_j} + w^{(L)}_{F_i,c_j}}{2}$
14: $\quad$ **end for**
15: $\quad$ **for each** $c_j \in \mathcal{C} \setminus \{\mathcal{C}_{F_i}\}$ **do**
16: $\quad\quad \omega_{F_i,c_j} \leftarrow 1$
17: $\quad$ **end for**
18: **end for**

10). Then, the remaining lower bounds for the weights are determined, which are denoted as $w^{(L)}$ (cf. Alg. 5.1, lines 3-5). Subsequent, we set the second degree of freedom, which is the upper bound for the content with the lowest priority (cf Alg. 5.1, line 6 and line 11). Then, the remaining upper bounds are calculated, which are denoted as $w^{(U)}$ (cf. Alg. 5.1, lines 7-9). Now, the weights ω are chosen. Here we take a value in the middle of the open interval $(w^{(L)}_{F_i,c_j}, w^{(U)}_{F_i,c_j})$ (cf. Alg. 5.1, lines 12-14). Finally, the algorithm resets the weights for all contents that do not satisfy the conditions for an adaptable set (cf. Alg 5.1, lines 15-17). We have implemented Algorithm 5.1 for SAF's adaptation engine as presented. The source code can be found at `http://icn.itec.aau.at`.

Remark 5.1. Note that contents with the same importance can be represented as a single content. However, weights for these contents have to be calculated and the greater ones have to be taken for all contents with the same importance.

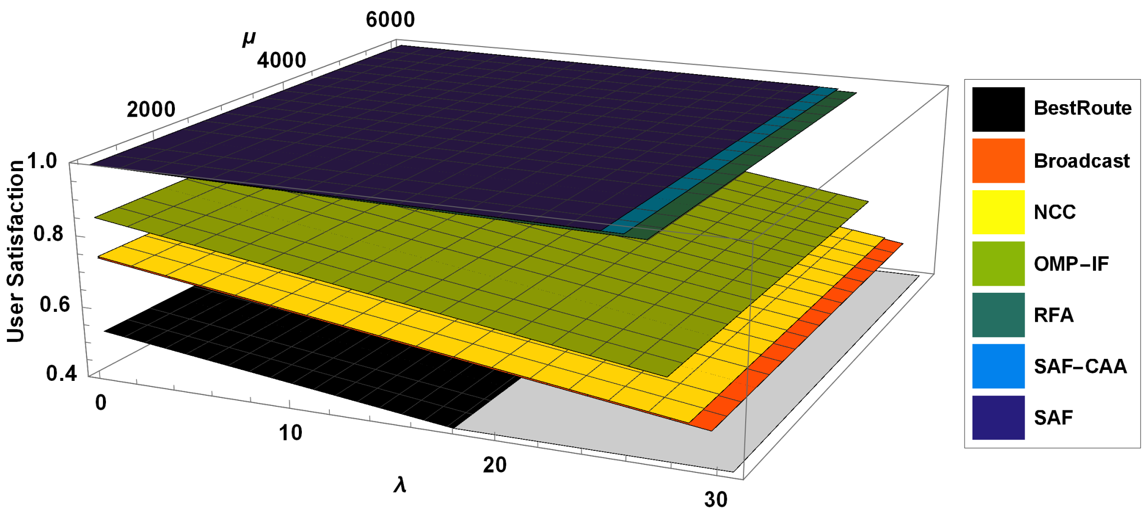
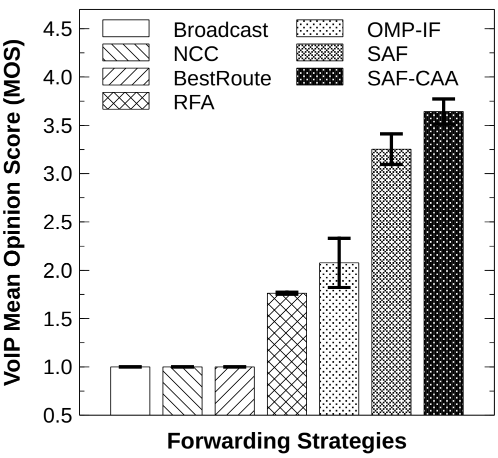


Figure 5.4: Video streaming satisfaction considering different preferences (λ denotes a weight for a user's sensitivity to quality variations, and μ denotes a weight for a user's sensitivity to re-buffering time) [139].

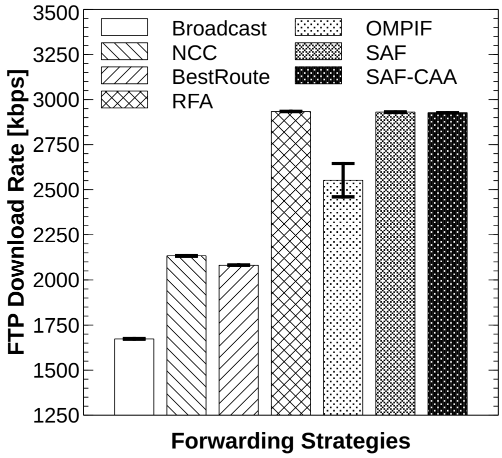
5.3.4 Results

Figures 5.4 and 5.5 summarize the results for the individual applications considering all forwarding strategies discussed in Section 5.2. Figure 5.4 depicts the user satisfaction of the video streaming users obtained by Equation 5.5 normalized by the highest possible score. Figure 5.5a depicts the MOS value for the VoIP clients. Figure 5.5b depicts the achieved download rate of the FTP clients. Figure 5.5c depicts the total traffic (incoming and outgoing) triggered by the ISP access network illustrated on the left hand side of Figure 5.1. Figure 5.5d provides the average costs per transmitted kilobyte. In the following we discuss the results for each strategy in detail:

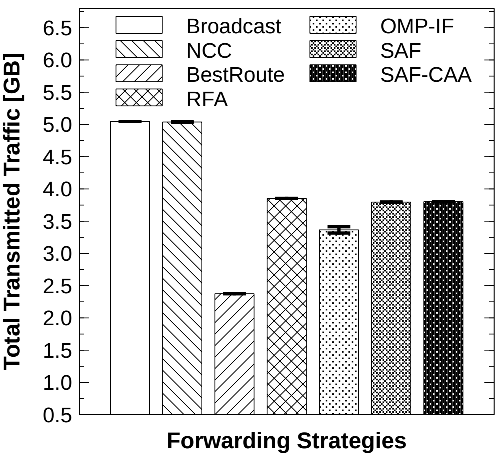
Broadcast: It is evident from Figure 5.5 that Broadcast performs worst. This can be explained by the strategy's nature to excessively replicate Interests and by the rather limited available bandwidth resources. Broadcast introduces a lot of unnecessary overhead. Although the strategy causes with 5 GB the highest amount of traffic (and therefore also high costs for the provider, Figure 5.5c) the consumer/application demands are not fulfilled. The congestion caused by Interest replication leads to a high packet loss rate and heavily delayed packet delivery. This results in the worst possible MOS for the VoIP clients (cf. Figure 5.5a) and also in the worst achieved goodput (1672 kbps) for the FTP clients (cf. Figure 5.5b). Nevertheless, the user satisfaction of the video streaming clients is at an acceptable level, even higher than for BestRoute (cf. Figure 5.4).



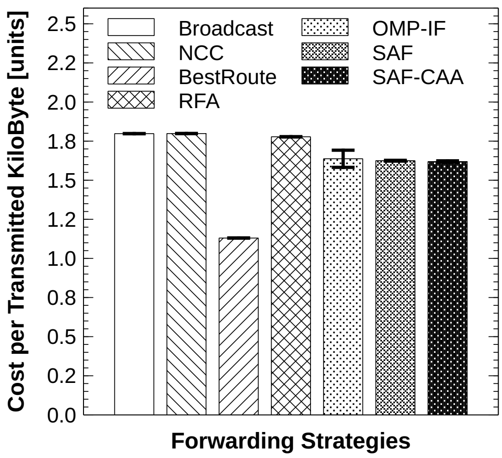
(a) MOS for VoIP clients.



(b) FTP download bitrates.



(c) Total transmitted traffic.



(d) Average costs per kilobyte.

Figure 5.5: Results considering the individual applications' performance with respect to the discussed forwarding strategies showing 95% confidence intervals.

NCC: The overall performance of NCC is similar to Broadcast. The only major distinction is the performance of the FTP clients, which are able to achieve a 460 kbps higher goodput. The bad performance of NCC can be explained by the following facts. NCC focuses only on the link that provides the lowest data delivery delay. Therefore, the network traffic is preferably routed through AS-A. The limited resources of AS-A are overburdened resulting in packet loss due to network congestion. Internally, NCC issues retransmission for late or lost packets using alternative paths. This excessive retransmission policy introduces a similar amount of overhead like Broadcast (cf. Figure 5.5c), resulting in the worst possible MOS of 1 for the VoIP clients and high costs for the ISP operator(s). Also the achieved user satisfaction of the video streaming consumers is only marginally better than achieved by Broadcast (cf. Figure 5.4).

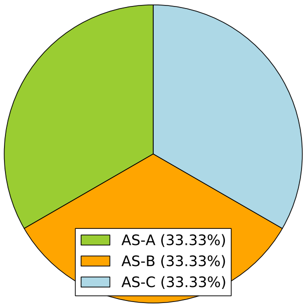
BestRoute: This strategy, configured to consider the ISP's costs as the relevant metric, is no great improvement to Broadcast or NCC. Instead of NCC that focuses on the lowest-delay connection through AS-A, BestRoute focuses on the lowest-cost connection through AS-C (cf. Figure 5.1). Only if a consumer application issues a retransmission due to late or lost packets, BestRoute considers the next "best" (cheapest) route for data transmission. The low transmitted amount of traffic presented in Figure 5.5c indicates that BestRoute is unable to effectively use multiple paths for delivery leading to the lowest user satisfaction for the video streaming clients (cf. Figure 5.4). Furthermore, BestRoute is unable to deliver a single VoIP packet in time due to its strict focus on the "cheapest" route, which is also reflected by the worst possible MOS of 1 achieved by the VoIP clients. However, it achieves the lowest cost per transmitted kilobyte (cf. Figure 5.5d).

RFA: This is the first algorithm that achieves a MOS greater than 1 for the VoIP clients (cf. Figure 5.5a). As previously mentioned, RFA basically performs a kind of load balancing by considering the number of pending Interests per face. Considering this scenario, the strategy performs fine, in particular the FTP clients are able to achieve their target goodput of about 3000 kbps (cf. Figure 5.5b). Also, the user satisfaction for the video streaming clients is acceptable (cf. Figure 5.4). Nonetheless, a better performance for the VoIP clients is desirable. On the one hand, the results for RFA indicate that it is perfectly able to exploit multiple paths. However, on the other hand, it is ignorant to the application demands resulting in non-optimal forwarding decisions (e.g., a non-negligible share of VoIP Interests is forwarded via AS-C).

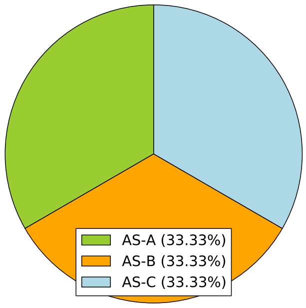
OMP-IF: This strategy is able to obtain a MOS of about 2 for the VoIP clients, which is significantly better than RFA (cf. Figure 5.5a). Nevertheless, OMP-IF is inferior with respect to the data transfer (cf. Figure 5.5b) and video streaming (cf. Figure 5.4) performance, for the following reasons. While RFA is focused on load balancing, and therefore tries to maximize the throughput, OMP-IF considers only node-disjoint paths for individual name prefixes. As in this scenario only three different prefixes are employed (*/voip*, */data*, */video*) OMP-IF is capable of separating the individual content streams. However, taking the burstiness of the video traffic into account, none of the autonomous systems AS-A, AS-B and AS-C can fulfill the traffic demands during the bursts. These bursts lead to packet loss or late packets, which will be countered by OMP-IF by frequent path switching. The frequent path switching leads to a lower performance with respect to the video and transfer applications, though, it gives the VoIP clients the opportunity to obtain a slightly better service than in the case of RFA.

SAF(-CAA): It is evident from Figures 5.4 and 5.5 that SAF outperforms all other algorithms. It reaches a MOS for VoIP of more than 3, while maintaining the target goodput of the data streaming clients and delivering excellent service to the video streaming clients. The key to success for SAF is the consideration of context that allows to maintain this high QoS and QoE levels. As previously mentioned, SAF evaluates for each name prefix and for each face the Interest satisfaction ratio. Considering this ratio, it derives the optimal forwarding strategy. For instance, it is able to deliver the VoIP traffic successfully since it implicitly considers the lifetime flag that is provided by every Interest packet (cf. Figure 2.4). If an Interest times out in the forwarding plane due to a lifetime expiry it is counted as unsatisfied. SAF uses this information to deduce better decisions in the future. Therefore, it is able to guide the VoIP traffic over the low-latency links. As previously introduced, SAF-CAA considers VoIP traffic as more important than video or data traffic, which are in general more resilient to low values of packet loss. These applications may use retransmission to re-request missing/lost packets. As can be seen from Figure 5.5, introducing this context information into the forwarding plane further increases the MOS value for the VoIP clients by about 0.4 without resulting in significant (negative) side-effects for the data and video streaming applications. SAF-CAA performs better than classical SAF because in the case of congestion on the low-latency links, video and data traffic are penalized stronger and earlier than the VoIP traffic. As the traffic share of the VoIP traffic is significantly smaller than the video and the data shares, the prioritizing of the VoIP traffic has no noticeable effect on the video and data streamers' user satisfaction in this scenario.

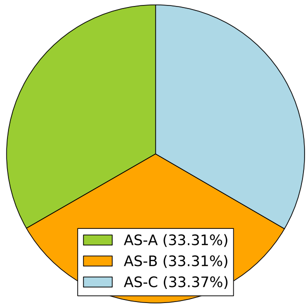
To further investigate the individual behavior of the algorithms in the presented scenario (cf. Figure 5.1), Figure 5.6 depicts the individual shares of Interests that are forwarded for each content/application per autonomous system. The figure shows the result from an exemplarily chosen simulation run. Each row of pie charts presents the Interest shares for the applications VoIP (1st column), video (2nd column), and FTP (3rd column) with respect to the investigated forwarding strategies which may forward Interests via AS-A (green), AS-B (orange), and AS-C (light blue). Figure 5.6 shows that Broadcast and NCC are oblivious to the individual application demands, simply transmitting equally sized shares



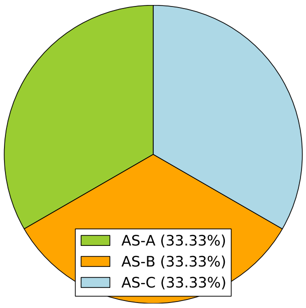
(a) VoIP (Broadcast).



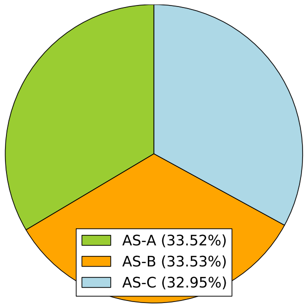
(b) Video (Broadcast).



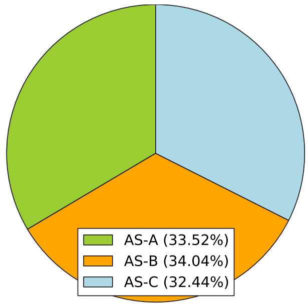
(c) FTP (Broadcast).



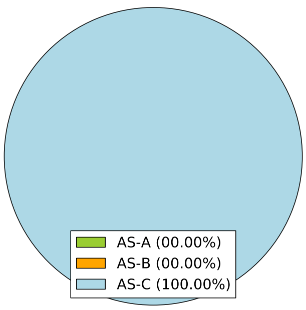
(d) VoIP (NCC).



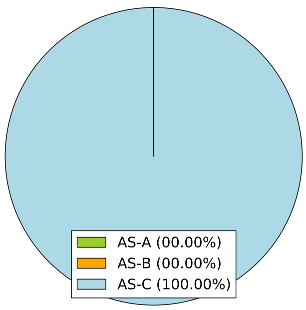
(e) Video (NCC).



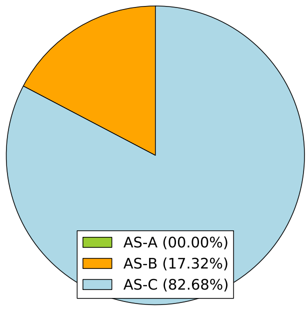
(f) FTP (NCC).



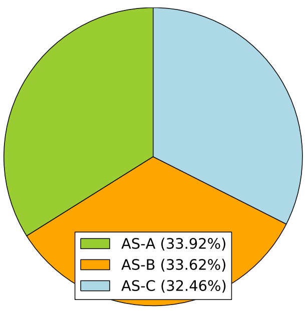
(g) VoIP (BestRoute).



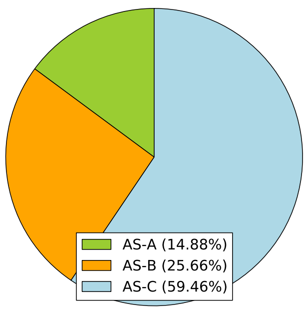
(h) Video (BestRoute).



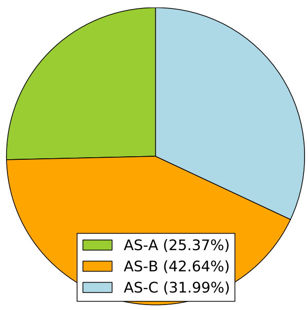
(i) FTP (BestRoute).



(j) VoIP (RFA).

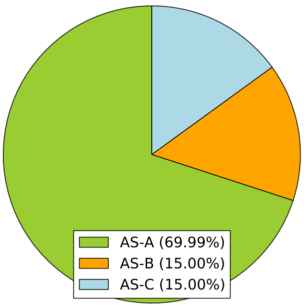


(k) Video (RFA).

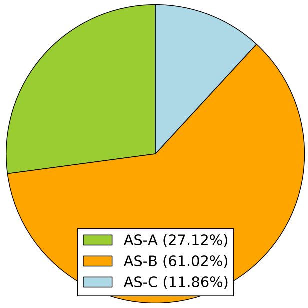


(l) FTP (RFA).

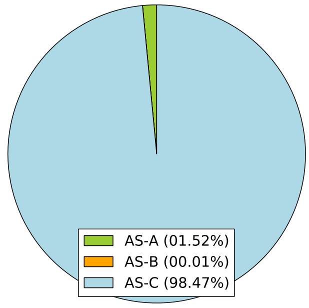
Figure 5.6: Share of Interests forwarded via the available autonomous systems per content/application (cf. Figure 5.1). *The figure is continued on page 139.*



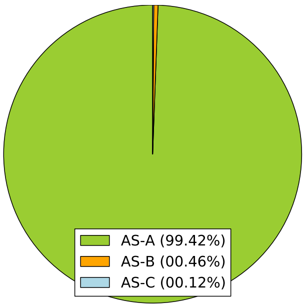
(m) VoIP (OMP-IF).



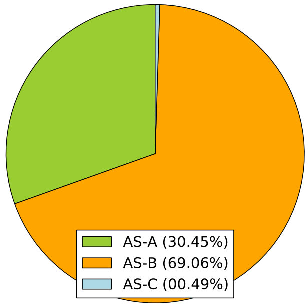
(n) Video (OMP-IF).



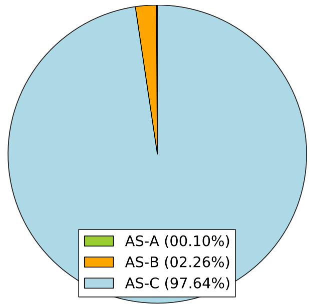
(o) FTP (OMP-IF).



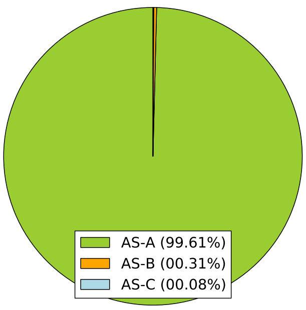
(p) VoIP (SAF).



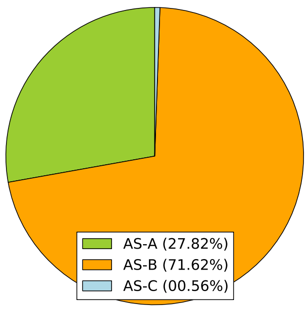
(q) Video (SAF).



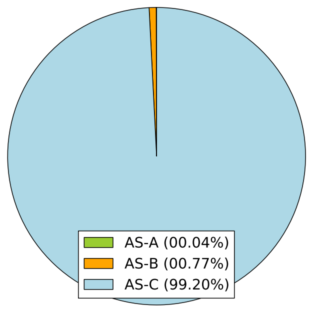
(r) FTP (SAF).



(s) VoIP (SAF-CAA).



(t) Video (SAF-CAA).



(u) FTP (SAF-CAA).

Figure 5.6 (cont.)**:** Share of Interests forwarded via the available autonomous systems per content/application (cf. Figure 5.1).

on all available autonomous systems. Although both strategies forward approximately two-thirds of the VoIP Interests via AS-A and AS-B, they are not able to achieve a MOS greater than 1. In contrast, RFA achieves a MOS of about 1.5 (cf. Figure 5.5a) showing a similar traffic share pattern. Although also RFA forwards one-third of the VoIP traffic via AS-C (that is not able to deliver VoIP packets in time), it is able to outperform Broadcast and NCC. The reason for this is that RFA causes less congestion because of: *i*) not excessively replicating Interests (cf. Figure 5.5c); and *ii*) by effective load balancing of traffic considering the number of PIT entries as an indicator for face utilization (faces with many PIT entries are avoided). BestRoute basically considers only AS-C for forwarding, because it has been configured to prefer the lowest-cost path. Only for lost and/or retransmitted Interests an alternative path via AS-B is used. Therefore, this strategy fails in satisfying the demands of the VoIP clients and also can not take advantage of NDN's multi-path capabilities. In contrast, OMP-IF does a much better job. OMP-IF only forwards 15% of the VoIP Interests via AS-C, and therefore, obtains a MOS of about 2. We can also see that OMP-IF forwards the largest part of the data traffic to AS-C. However, when compared to SAF(-CAA), we observe that better results are possible. SAF performs excellently, forwarding more than 99% of the latency sensitive VoIP traffic via AS-A and forwarding more than 99% of the latency tolerant FTP traffic via the lowest-cost path (AS-C). SAF forwards the biggest share of the video traffic via AS-B, using remaining resources of AS-A that are not required by the VoIP traffic. The results show that SAF optimally separates and distributes the Interests for the individual applications on the available autonomous systems. Furthermore, we can observe that the weighting of contents has a significant positive effect on the MOS (cf. Figure 5.5a), and it has basically no influence on the traffic shares. The weighting only ensures that data and video packets are dropped earlier than (in favour of) VoIP packets.

5.4 Conclusion and Outlook for Context-Aware Forwarding

In this chapter we investigated the importance of considering context information in NDN's forwarding plane. First, we outlined existing forwarding strategies and discussed their context awareness (their basis of decision making). Furthermore, we defined a scenario encompassing different network applications (VoIP, video streaming, data transfer) with various demands and evaluated their performance using prominent forwarding strategies. We conducted network simulations using ns-3/ndnSIM [37] and obtained the relevant QoS parameters. We mapped these values to user satisfaction models to asses their actual benefit. The results indicate that the more context information is considered by the forwarding strategies, the better becomes the provided QoS, resulting in higher user satisfaction. In particular, the strategy Stochastic Adaptive Forwarding (SAF), which can be easily configured to consider context information, performs excellently in the presented scenario. We presented an enhancement for SAF's adaptation engine that is capable of considering supplied context information for the forwarding operations. Our approach rests on a weighting mechanism that orders contents according to their user/operator-supplied priority. According to the internals of SAF, we were able to derive the weights according to the definition of a *reliable* face. The weights are used to influence the performance assessment of faces in order to drop low priority content in favour of high priority content on exhausted faces.

Our results indicate that further research in the area of forwarding should focus on the available context information to unlock the full capabilities of NDN's forwarding plane. The presented scenario was of limited size, and therefore a more extensive investigation of this topic is necessary in future work. While our approach solely is intended for the forwarding plane, also overlapping approaches between the routing and the forwarding layers are possible. For instance, future work could investigate how routing could help to introduce context awareness in the forwarding plane. One idea would be to reserve or block certain paths for specific traffic/applications considering a broader notion of context [96, 97]. For instance, given the previously presented scenario we could already decide at the moment the routing layer provides/installs the available routes in the FIB that for latency intolerant traffic the path through AS-C is not suitable. Therefore, we may not add this face to the possible outgoing faces in the first place.

CHAPTER

6 From Simulations to Physical Networks

"The problems of the real world are primarily those you are left with when you refuse to apply their effective solutions."

— Edsger W. Dijkstra, 1930* - 2002†

The computer communication research community shows significant interest in the paradigm of Information-Centric Networking (ICN). Continuously, new proposals for ICN-related challenges (caching, routing, forwarding, etc.) are published. However, due to a lack of a readily available testbed, the majority of these proposals is either evaluated by theoretical analysis and/or by conducting network simulations potentially masking further challenges that are not observable in synthetic environments. Therefore, this chapter presents a framework for an ICN testbed using low-budget physical hardware with little deployment and maintenance effort for the individual researcher; specifically, Named Data Networking (NDN) is considered. The employed hardware and software are powerful enough for most research projects, but extremely resource intensive tasks may push both components towards their limits (e.g., cryptographic algorithms). The testbed framework is based on well established open source software and provides the tools to readily investigate important ICN characteristics on physical hardware emulating arbitrary network topologies.

This chapter is divided into five sections. Section 6.1 provides an introduction motivating the need for a testbed in ICN/NDN research. This also includes a brief discussion about existing testbeds, and why they are not sufficient for a researcher's daily work. Section 6.2 illustrates the testbed architecture and discusses the necessary preparation steps to conduct significant network emulations. The following two sections present evaluations that have been obtained from the testbed. The first set of emulations investigates the performance of various forwarding strategies (cf. Section 6.3). The objective of this evaluation is to validate the results obtained for SAF and its competitors (cf. Section 4.3), which have been obtained conducting networking simulations using ns-3/ndnSIM. The evaluations on the testbed are performed on a smaller scale (fewer networking nodes due to the limited number of nodes available), however, the results provide an estimate how well SAF and its competitors perform on physical hardware. Moreover, the results may indicate further challenges that have to be overcome when pursuing a real-world deployment of ICN-based communication. Section 6.4 provides an evaluation of DAS-based multimedia streaming on the testbed. Here, the experiments are designed similar to the experiments conducted in Section 3.2.3, however, also on a smaller scale due to the limited number of devices at hand. Finally, Section 6.5 provides a conclusion of the chapter giving an outlook on further research activities that may benefit from the proposed testbed framework.

6.1 Introduction – Yet Another Testbed?

In recent years, the ICN research community has grown continuously, positioning data-oriented communication as promising architecture for the Future Internet. Today, there is a variety of ICN candidates (cf. Section 2.1.3) interpreting and implementing data-oriented communication in their own ways, as surveyed in [14] and [144]. Among them is Named Data Networking (NDN) [15], probably the most elaborated approach. The NDN community provides an extensive software suite with the objective to further advance research in this field. This includes the network simulator *ndnSIM* [37], which is based on the well-known *ns-3* framework. Furthermore, the software suite includes the *Networking Forwarding Daemon (NFD)* [38], an implementation of the NDN principle on Linux-based systems. Although the Linux-based implementation of [38] is freely available, evaluations on physical networks are rarely conducted in research papers. The majority of proposals have been evaluated in a synthetic environment, either relying on theoretical analysis or using network simulations. While both methods are valid and necessary, they will always model the environment, protocols, and constraints imperfectly, simplifying and disregarding potential important aspects of the real world. Therefore, these methods should not be the final evaluation step, and they definitely can not substitute experiments on physical networks. Hence, it would be desirable that researchers additionally conduct experiments on physical hardware. This may reveal weak components, performance bottlenecks, and further challenges that are not observable in synthetic environments. However, there are a number of impediments for experiments on physical networks/testbeds:

- Setting up a testbed is a tedious and work-intensive task.
- Building a testbed of significant size can be very costly.

- Conducting a large batch of evaluations using different parameters in a testbed is more time-consuming/error-prone than using a flexible simulation environment.

In order to overcome these impediments and to support researchers in their daily work, the provision of a testbed is necessary. Currently, the NDN community provides a testbed [145] available to its community members. At the time of writing, the community encompasses about 30 members consisting of universities and industrial companies interested in ICN research. The testbed interconnects 32 NDN nodes and encompasses 87 links around the world realized as an overlay on top of today's Internet providing the opportunity to evaluate NDN-based applications. Figure 6.1 provides the topology of the testbed. Nevertheless, this testbed has several inconveniences for the individual researcher. First, the researcher must be a (paying) member of the community to get access to the testbed nodes. Furthermore, the testbed is actually a shared resource as it can be accessed by all community members, and therefore the availability for the individual researcher may be limited. Moreover, researchers may not be able to experiment with the core functionality of the NFD (even if it would be necessary for their research objectives), since the NDN testbed needs to stay functional at all times. The testbed is rather designed for experiments with NDN-enhanced applications evaluating the benefits of NDN-based data communication, rather than for experiments with

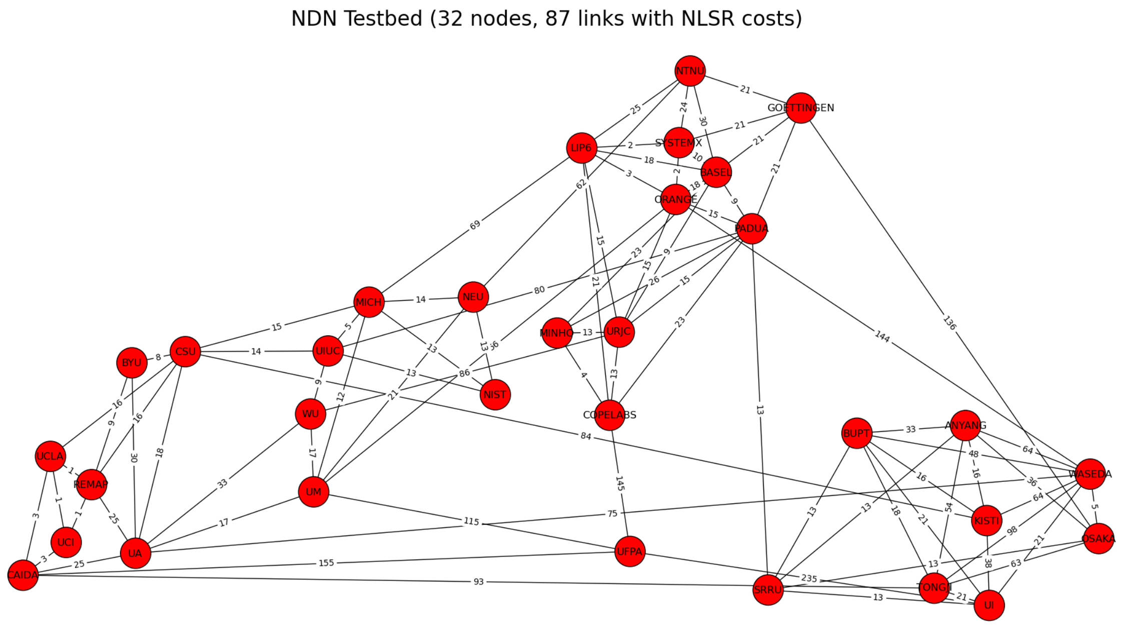


Figure 6.1: Screenshot taken from [145] illustrating the topology of the NDN testbed. The numbers on the links provide the estimated costs by the routing plane using the Named-Data Link State Routing (NLSR) protocol [50]. *(Screenshot taken 2016/07)*

the NDN core functionalities. Another issue is also that the NDN testbed is realized as an overlay network on top of today's Internet using ordinary computers as software routers. Since the researchers do not have full control over the used infrastructure (devices, links, etc.), side effects may influence their evaluations leading to distorted results and aggravate reproducibility.

To tackle the above mentioned challenges, this chapter presents a framework enabling researchers to readily deploy their own NDN testbed at low costs. We suggest a testbed framework that is based on a set of single-board devices, so-called Banana Pi Routers. While focusing on a low budget, the selected hardware is powerful enough to conduct significant evaluations in all research areas of interest regarding ICN. An exception is the use of computationally expensive cryptography; however, this would also challenge recent off-the-shelf CPUs such as used in ordinary desktop computers without dedicated cryptographic hardware support. The costs for an NDN Banana Pi testbed of significant size, e.g., 20 nodes, is approximately 3400 USD. Costs scale linearly with approximately 160 USD per added testbed node. We provide pre-configured disk images, scripts and installation guidelines for download (`http://icn.itec.aau.at`) enabling researchers to easily realize their own NDN testbed with little effort. Once the images are copied to the required disks and the testbed hardware is assembled and connected to a network, the testbed is ready for use. The proposed framework allows to quickly deploy arbitrary network overlays on the physical topology. Researchers may specify every detail of the topology including link capacities, delays, queuing disciplines, queue sizes, and more. The framework further provides a Web interface for observing the health status of the testbed and allows the tracking of ongoing emulations in near real-time.

6.2 A Framework for an NDN-based Banana Pi Testbed

This section discusses the testbed architecture. First, insights on the selected hardware are given (cf. Subsection 6.2.1). Second, the tools and applications necessary to realize an arbitrary network topology on the physical devices are introduced, and their configuration is discussed (cf. Subsection 6.2.2). Furthermore, the installation and configuration of the NDN-based communication layer is presented (cf. Subsection 6.2.3). Finally, the classical emulation cycle and the Web-based monitoring possibilities are presented (cf. Subsections 6.2.4 and 6.2.5).

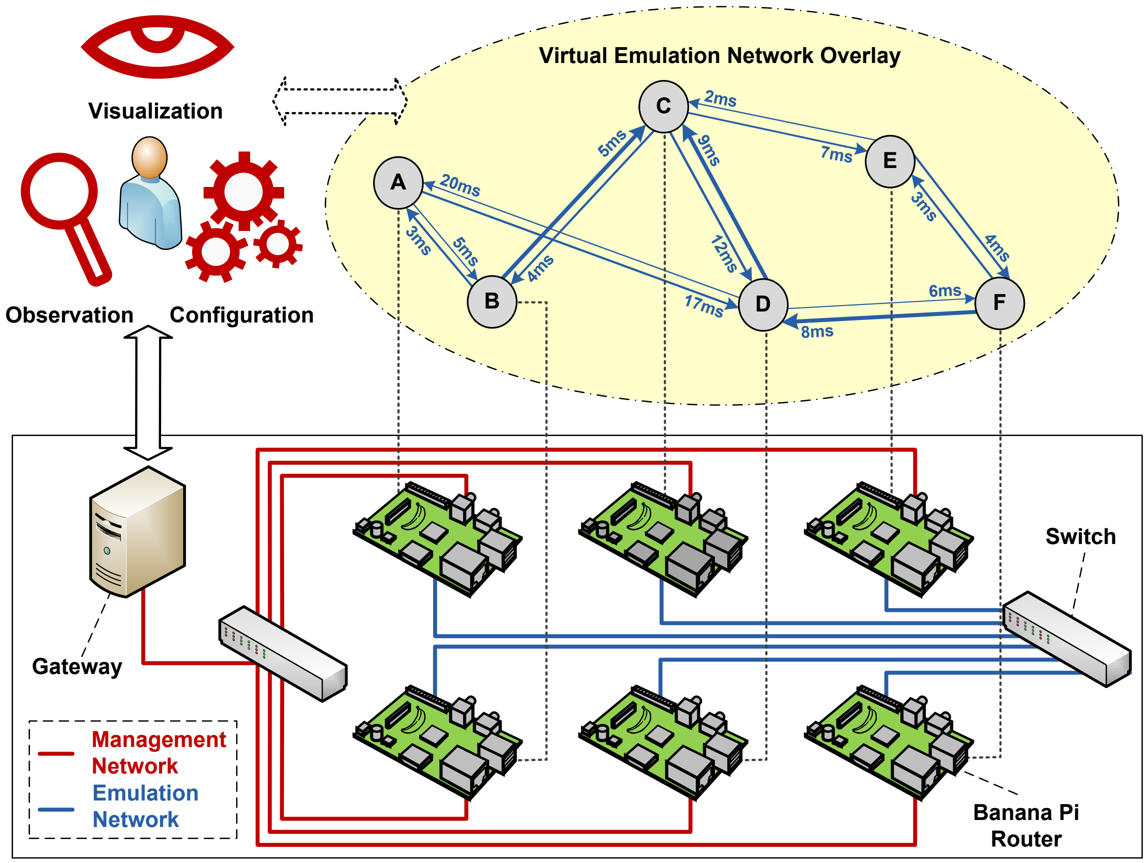


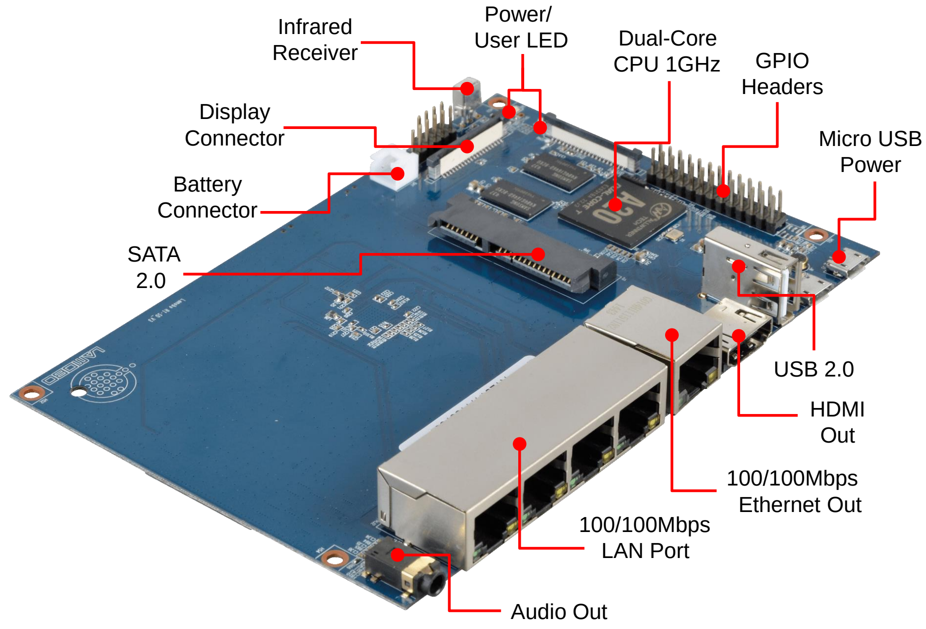
Figure 6.2: An overview of the testbed architecture. The testbed consists of two dedicated physical networks. The *Management Network* (red lines) is used to set up and control the testbed nodes, while the *Emulation Network* (blue lines) realizes the virtual overlay network. The gateway is used as control server and enables users to configure, observe and visualize emulation tasks.

6.2.1 Testbed Architecture and Hardware Requirements

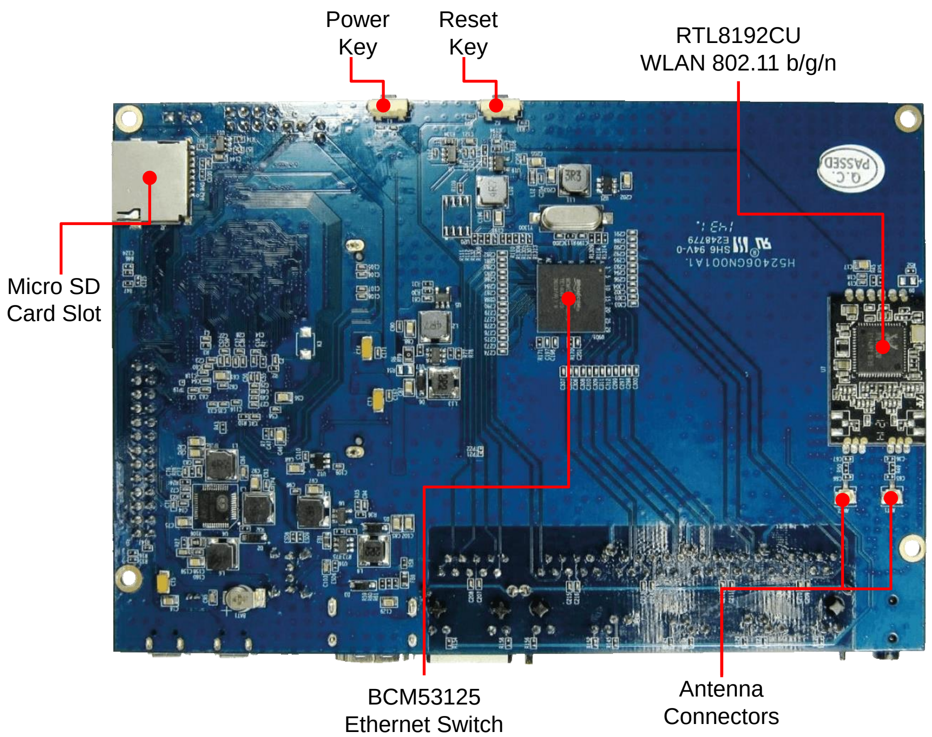
Figure 6.2 depicts an overview of the testbed architecture. The testbed is based on an IP network and realizes a virtual NDN overlay atop. In general, the testbed consists of a number of single-board computers, (at least) two network switches and a gateway. The testbed architecture requires two dedicated networks, each forming a star topology. The first network is denoted as the *Management Network* (MN), while the second one is denoted as the *Emulation Network* (EN). This clear separation ensures that traffic in the management network (setting up nodes, monitoring devices, deploying software, etc.) does not interfere with, or influence, active emulations. An ordinary Personal Computer (PC) provides enough resources to manage the tasks of the gateway. The gateway is connected to the MN only, providing external communication and control functionality. It acts as the control server for conducting emulations and is therefore the entry point for the users to the testbed. Furthermore, it hosts several (optional, but very useful) services, which allow the monitoring of the testbed (discussed in detail later, cf. Subsection 6.2.5). The single-board computers should be able to access the Internet via the gateway, so software updates can be retrieved easily.

Since the testbed architecture foresees two dedicated networks, the testbed nodes (single-board computers) are required to provide at least two network interfaces. We have chosen the Banana Pi Routers (BPI-R1) (cf. Figure 6.3) over all other available single-board computers, because they are equipped with a four-port LAN switch including a dedicated switching circuit. Moreover, these devices provide a WLAN interface (supporting IEEE 802.11 b/g/n), which additionally gives the opportunity to support mobility scenarios. Another advantage of the BPI-R1 is that it offers a SATA 2.0 port. We consider this as an important feature, especially for conducting emulations in the area of ICN. Users may want to conduct experiments requiring large in-network caches. When using only ordinary Micro SD cards (default disk storage for embedded devices and single-board computers), such experiments could not be performed since the cache size would be constrained by the main memory (1 GB DDR3-SDRAM). Paging memory to the Micro SD cards with rather low read/write data rates would drastically decrease the performance below the required line speed. Therefore, we strongly suggest equipping the Pis with a Solid State Disk (SSD). We summarize the most important hardware specifications of a BPI-R1 in Table 6.1. Furthermore, Table 6.2 lists the required hardware components (including approximate costs per unit) for a testbed with 20 nodes. Note that we did not include hardware for the control server because any available PC should be able to handle the required workload and does not burden one's budget.

Once all hardware components are available, the Pis need to be assembled (they are usually shipped without a case), and the two networks illustrated in Figure 6.2 are created connecting the Pis with the network switches. The testbed should be placed in an air-conditioned environment, and the Pis should not be stacked on top of each other to avoid thermal issues. We suggest placing them upstanding with a small space between the individual cases as shown in Figure 6.4. Furthermore, we suggest to uniquely label the Pis with stickers and to use network cables in two colours. This allows an easy identification of the individual devices and also provides a quick way to distinguish whether a cable belongs to the emulation or management network.



(a)



(b)

Figure 6.3: The Banana Pi Router (BPI-R1): (a) top view, (b) bottom view indicating the most important components/interfaces.

Component	Description (http://www.bananapi.com)
CPU	A20 ARM-Cortex-A7 Dual-Core (2x1.0 GHz)
GPU	ARM Mali-400MP2 with Open GL ES 2.0/1.1
Memory	1 GB DDR3-SDRAM
Storage	1x Micro SD (max. 64 GB), 1x SATA 2.0 (max. 2 TB)
Network	1x Ethernet RJ45, 4-Port-Switch, WLAN 802.11b/g/n
Power Source	5 Volt / 2 Ampere via Micro USB
Interfaces	Video In(CSI for video cameras)/Out(HDMI,CVBS,LVDS), Audio In(on-board micro)/Out(3.5 mm Jack, HDMI), 2x USB 2.0, GPIO, UART, I2C Bus, SPI Bus, CAN bus, etc.
Circuit Board	92mm x 60mm, 52g

Table 6.1: Specification of a Banana Pi Router (BPI-R1).

Component	Advice/Comment	≈ Cost/Unit
20x BPI-R1		80 USD
20x Case for BPI-R1	optional, but handy	15 USD
20x SSD	size of at least 120 GB	50 USD
20x Micro SD	size of at least 8 GB	4 USD
20x USB power cable	dimensioned for 2 ampere	3 USD
4x USB power hub	at least 6 ports	20 USD
2x Gigabit switch	at least 24 ports	100 USD
40x Ethernet cable	CAT6, two colors, 5 ft	2 USD
	Total:	**3400 USD**

Table 6.2: Components for a testbed with 20 nodes.

6.2.2 Deployment of the Virtual Network Overlay

Having the Pis assembled and connected, the next step is to deploy the desired virtual network overlay on top of the testbed. The BPI-R1 is shipped with its own operating system called Bananian, a customized Debian Linux composed by the hardware manufacturers. So, basically all networking tools available for Linux, e.g, *iptables* [146] and *traffic control (tc)* [147], can be used to realize the desired overlay network. However, in order to use some more advanced features of the *tc*, the Linux Kernel has to be re-configured and re-compiled since the default disk image is optimized with respect to space constraints rather than for networking functionality. Therefore, we provide a modified Bananian image for download at `http://icn.itec.aau.at`. There are two images available, one for the Micro SD

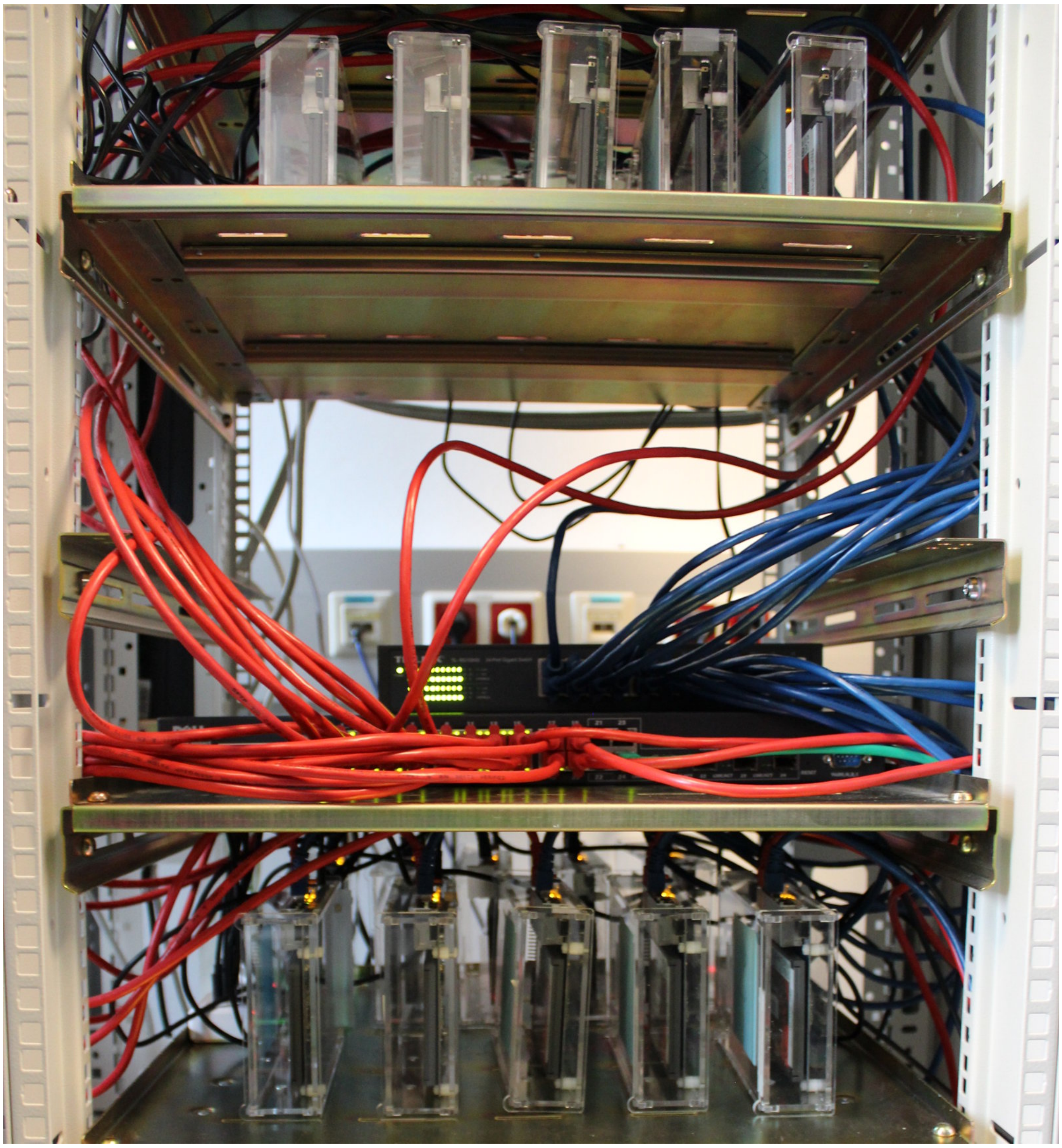


Figure 6.4: The Banana Pi NDN testbed deployed at ITEC `http://icn.itec.aau.at/`. It consists of 20 nodes and was assembled from the parts listed in Table 6.2.

card, and another one for the SSD. Please note that even when using the SSD, one still requires the Micro SD card as boot device (the BPI-R1 is only able to boot from a storage media in the SD card slot).

We continue the discussion by providing the technical details to deploy a virtual overlay on the testbed. All discussed steps are implemented in Python and Linux Bash Scripts that can be downloaded at the previously indicated Web page. Researchers using the framework may specify arbitrary overlay networks (in an ordinary text file, or use the provided random network topology generator) and apply them on the physical devices instantaneously. Note that only the EN will be modified. The MN remains always unchanged since it is only used for configuration and monitoring tasks.

Let us assume that the overlay network illustrated in Figure 6.2 shall be deployed. The directed edges define the virtual links connecting the individual nodes (thicker lines indicate higher link capacities), and the link delays are indicated next to the directed edges' heads in milliseconds. The first step is to realize the virtual connection(s). This can be achieved using the application *iptables* [146] that enables the configuration of the tables provided by the Linux kernel firewall. Initially, we configure the tables to block all IP packets received or transmitted over the EN. This is achieved by setting the default action for all packet processing chains (INPUT, FORWARD, OUTPUT) to DROP (cf. Listing 6.1, lines 1,6 and 13). Then for each virtual link in the overlay network, an exception is inserted allowing to forward/receive IP packets. For instance, in the case of node A these are the upstream and downstream connections to/from B and D (cf. Figure 6.2). Listing 6.1 depicts the required exceptions to realize the connections for node A. Once the *iptables* are configured on all nodes, the basic topology of the overlay network is reflected by the EN on the IP layer.

```
1  Chain INPUT (policy DROP)
2  target  protocol  option  source  destination
3  ACCEPT    all       --      B        A
4  ACCEPT    all       --      D        A
5
6  Chain FORWARD (policy DROP)
7  target  protocol  option  source  destination
8  ACCEPT    all       --      A        B
9  ACCEPT    all       --      B        A
10 ACCEPT    all       --      A        D
11 ACCEPT    all       --      D        A
12
13 Chain OUTPUT (policy DROP)
14 target  protocol  option  source  destination
15 ACCEPT    all       --      A        B
16 ACCEPT    all       --      A        D
```

Listing 6.1: Entries in the Linux firewall tables for node A (cf. Figure 6.2).

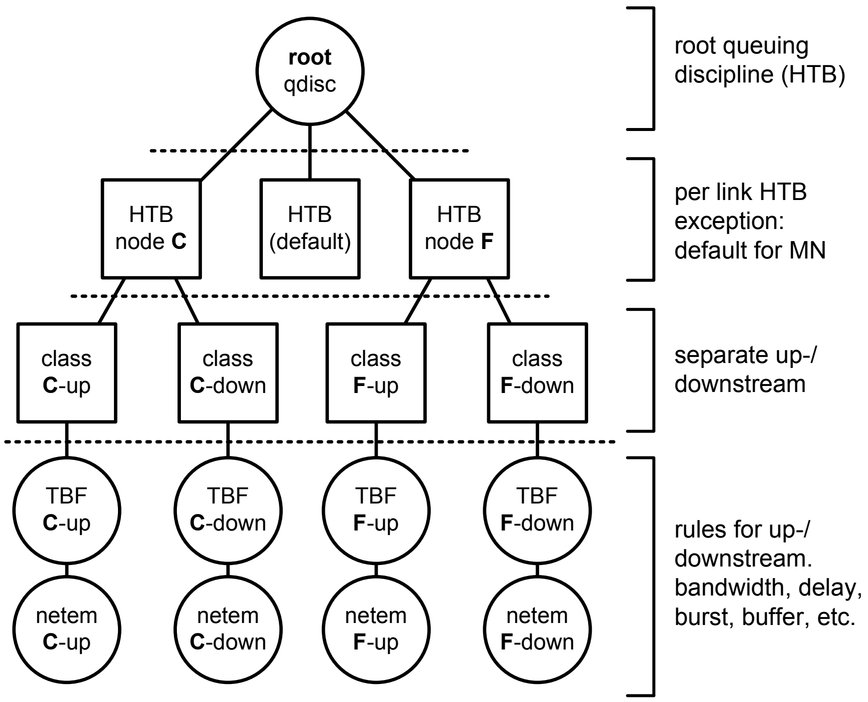


Figure 6.5: Linux *traffic control* classification for packet scheduling using the example of node E (cf. Figure 6.2).

The next step is to enforce the link capacities and delays as indicated by the overlay description. For this task the framework relies on *traffic control (tc)* [147], an application to configure and control the Linux kernel's network scheduler as sketched in Figure 6.5. The figure illustrates the employed traffic control classification for packet scheduling by the example of node *E*. We use Hierarchical Token Bucket (HTB) filters, with multiple child classes (one class per up- or downstream connection). The bandwidth capacities are not limited at the HTB level, but at the individual HTB's child classes. Each child is equipped with an additional queuing discipline, a classical Token Bucket Filter (TBF) controlling the link capacities and queue sizes. Furthermore, each child class is equipped with a *netem* discipline realizing the individual link delays. One can go even further and use *netem* [148] to introduce artificial packet loss or packet corruption facilities if desired; however, we do not discuss this further and refer to [146], [147] and [148] for the interested reader.

6.2.3 Installing the NDN Communication Layer

The final step before emulations can be conducted is to set up the Networking Forwarding Daemon (NFD) [38] accordingly on the Pis. This application implements the NDN-based communication layer. To make this task as convenient as possible, the testbed framework (NFD plus our extensions) takes also over this part. The user may specify the configuration of the NFDs (caching strategies, cache sizes, forwarding strategies and entries, etc.) and deploy it on the individual Pis using a deployment script. The NFD supports two baseline caching strategies (Least Recently Used (LRU) and First In – First Out (FIFO)), and several Interest forwarding strategies (BroadCast, BestRoute, and NCC) [38]. Nevertheless, one may be interested in implementing more sophisticated approaches. To that end, the articles [87] and [88] provide good surveys of recent caching strategies, while the related work in [108] gives a good overview of existing forwarding strategies.

To complete the configuration, the installation of the routing and forwarding entries in the Routing- and Forwarding Information Base of the NFD [38] is required. The framework provides two choices when users decide to automatically deploy routes, similar as implemented in ndnSIM [37]. Either the user may choose to use all possible routes, or only the shortest paths between each pair of nodes may be used. The framework implements the installation of the entries as follows. Each node is given a unique identifier. Then for each forwarding strategy and for each other node, one forwarding entry of the following structure is installed: */fw-strategy/source-node-id*. If using all routes has been selected, one entry may point to multiple outgoing faces. So, any user-defined application within the testbed can easily choose the source from which it may consume data by specifying the producer's node identifier. One further chooses the forwarding strategy by indicating the strategy's name in the emitted request message.

Listing 6.2 shows the available channels and faces for node A and the configuration of the FIB for the given network overlay depicted in Figure 6.2. The NFD [38] supports various channels type including TCP and UDP channels. Unfortunately, (raw) Ethernet channels are not supported at the time of writing. At this time our testbed uses UDP channels only, however, if needed one can readily switch to TCP channels. However, we prefer UDP channels because this much simpler transport protocol introduces less side effects (packet retransmission, congestion control, etc.) than TCP. The configuration shows that in this case we have decided to use all available routes to any available content source (node), since we have multiple possible next hops per prefix. Furthermore, it tells us that two forwarding strategies are installed (SAF and BestRoute) and their corresponding prefixes, which are used for forwarding Interests. Note that besides the “physical“ faces providing a connection to the two neighbouring nodes B and D, the NFD instance [38] additionally maintains “virtual“ faces, e.g., to applications. For example, there exists a face to the installed content store (*face 254*) and also a null face (*face 255*) that drops every packet. A face is always identified by a local and a remote endpoint (not necessarily physically remote). For instance, *face 258* (the physical connection between node A and B) has the remote endpoint *udp4://192.168.1.11:6363* (UDP-channel, remote IP, remote port) and the local endpoint *udp4://0.0.0.0:6363* (UDP-channel, local IP, local port).

```
Channels:
udp4://0.0.0.0:6363        /*UDP Channel*/
unix:///run/nfd.sock       /*Unix Stream Channel*/
tcp4://0.0.0.0:6363        /*TCP Channel*/
ws://[::]:9696             /*Web Socket Channel*/
Faces:
faceid=1    remote=internal:// local=internal:// local persistent p2p
faceid=254  remote=contentstore:// local=contentstore:// local persistent p2p
faceid=255  remote=null:// local=null:// local persistent p2p
faceid=256  remote=fd://18 local=unix:///run/nfd.sock local on-demand p2p
faceid=258  remote=udp4://192.168.1.11:6363 local=udp4://0.0.0.0:6363 non-local persistent p2p
faceid=260  remote=udp4://192.168.1.13:6363 local=udp4://0.0.0.0:6363 non-local persistent p2p
faceid=270  remote=fd://21 local=unix:///run/nfd.sock local on-demand p2p
Forwarding Information Base:
/localhost/nfd    nexthops={faceid=1   (cost=0)}
/best-route/B     nexthops={faceid=258 (cost=1), faceid=260 (cost=3)}
/best-route/C     nexthops={faceid=258 (cost=2), faceid=260 (cost=2)}
/best-route/D     nexthops={faceid=260 (cost=1), faceid=258 (cost=3)}
/best-route/E     nexthops={faceid=258 (cost=3), faceid=260 (cost=3)}
/best-route/F     nexthops={faceid=260 (cost=2), faceid=258 (cost=4)}
/saf/B            nexthops={faceid=258 (cost=1), faceid=260 (cost=3)}
/saf/C            nexthops={faceid=258 (cost=2), faceid=260 (cost=2)}
/saf/D            nexthops={faceid=260 (cost=1), faceid=258 (cost=3)}
/saf/E            nexthops={faceid=258 (cost=3), faceid=260 (cost=3)}
/saf/F            nexthops={faceid=260 (cost=2), faceid=258 (cost=4)}
Strategy Choices:
/best-route   strategy=/localhost/nfd/strategy/best-route/%FD%03
/saf          strategy=/localhost/nfd/strategy/saf/%FD%01
```

Listing 6.2: Channels, faces and FIB entries for node A (cf. Figure 6.2) considering two forwarding strategies SAF and BestRoute. Each FIB entry shows the available next hops for a given prefix including their estimated transmission costs (distance/hops) and face identifiers.

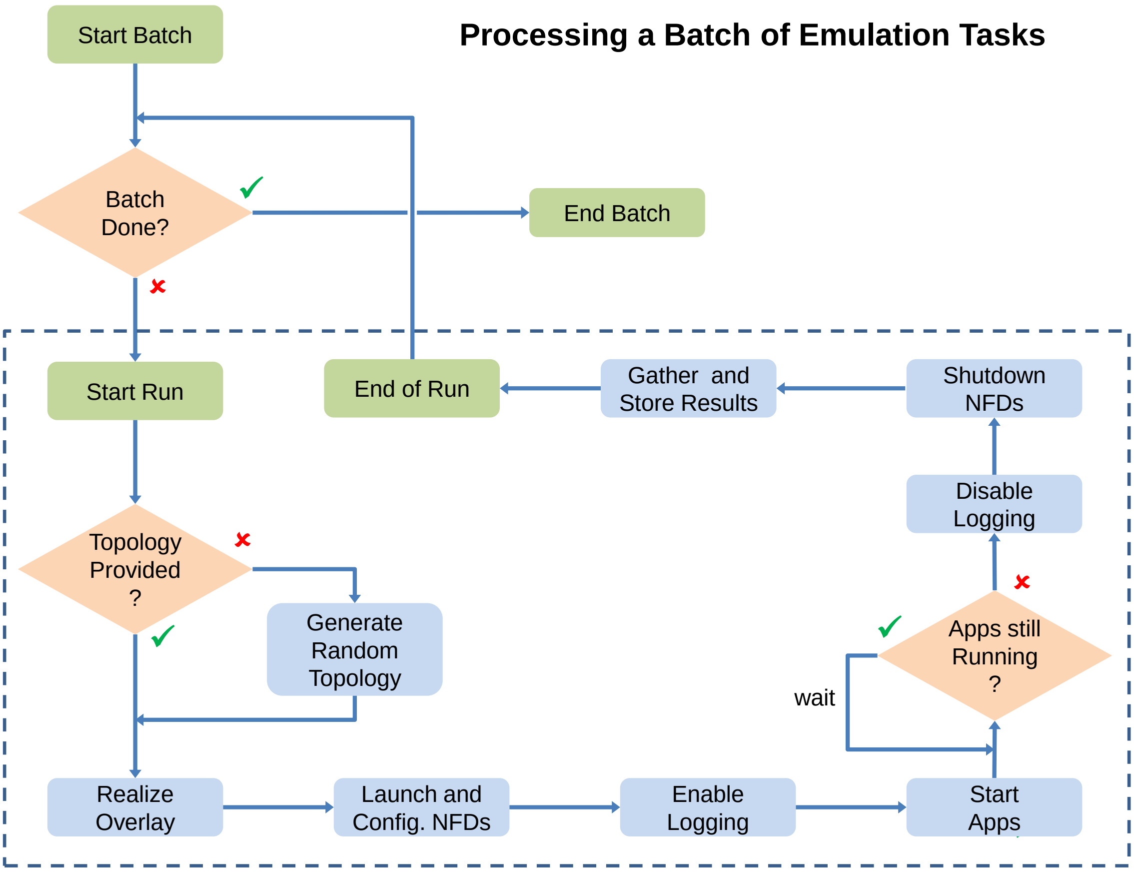


Figure 6.6: The figure shows the individual steps foreseen by the framework to process a batch of emulations tasks.

6.2.4 Conducting and Observing Experiments

Figure 6.6 shows the processing of a batch of emulations. The user needs to specify the emulation parameters in a script (a detailed guide and discussion about the parameters is available at `http://icn.itec.aau.at`). As previously mentioned, this includes the specification of the basic properties for the network topology, and the NFD configuration at first. Then, the framework foresees the following behaviour. One task after the other will be processed as illustrated in the Figure 6.6. Besides specification of the parameters for the network topology, the user specifies two kinds of *abstract* applications: *consumers* and *producers*. Which application is executed on the individual nodes has to be specified in the previously mentioned topology file (cf. Subsection 6.2.2). An exemplary topology file is presented in Listing 6.3 that describes the overlay network shown in Figure 6.2. The listing further assumes/specifies two consumers (A and C) that retrieve content from a producer

(F). After the network is deployed and the NFDs are set up, the framework enables the logging functionality on all Pis. Subsequent, all producer/consumer applications are executed (producers are started first). The gateway monitors the progress of the applications and stops the emulation once all consumer applications have finished. Then the logging functionality is stopped and the corresponding logfiles are gathered and stored on the gateway for later processing/analysis. The logfiles provide information about the Pis including CPU load, power consumption, memory usage, and may also contain application-specific data provided by the specified consumer/producer applications. Finally, the framework checks if there are remaining tasks to process and continues the batch processing or terminates.

```
#number of nodes
6
#links: a, b, bw a->b (kbps), bw a<-b (kbps), delay a->b (ms), delay b->a (ms)
A, B, 1000, 1500, 5, 3
A, D, 1000, 1500, 17, 20
B, C, 2000, 1000, 5, 4
C, D, 1500, 2000, 12, 9
C, E, 1500, 1000, 7, 2
D, F, 1000, 2000, 6, 8
E, F, 1500, 1500, 4, 3
#applications: consumer, producer
A, F
C, F
#eof
```

Listing 6.3: An example network topology file on the basis of the network shown in Figure 6.2 assuming two consumers (A and C) and one producer (F).

6.2.5 Testbed Monitoring

Testbed monitoring is coordinated by the gateway (cf. Figure 6.2). There are basically two levels of monitoring that can be accessed via the Web interface. The first level provides a very rough overview that allows monitoring the health state of the testbed. To that end, each Banana Pi uses *cron* [149] (a job scheduler for Linux) to trigger the logging of important data periodically, e.g., on an hourly basis. The logged data is provided in the JavaScript Object Notation (JSON) [150] format and is collected by the gateway using *wget* (a program to fetch data via HTTP-/FTP connections). This data includes important information about the Pis including CPU load, power consumption, memory usage, network traffic, disk usage, etc. A history of about one week is stored at the gateway providing a quick overview whether the testbed has been operating as expected. The second monitoring level is more detailed, allows near real-time observations, and is active once emulations are conducted. During an experiment, every few seconds the Pis actively push logfiles to a virtual RAM-disk on the gateway (through the MN) using the Network File System (NFS). These files are then accessible via a Web server and users may monitor the ongoing experiment in near real-time using the Web interface as illustrated in Figure 6.7. The second monitoring level can be disabled for experiments demanding highly accurate measurements of the CPU load or power consumption, since the provision of this real-time log data requires a small but noticeable resource consumption. Nevertheless, when designing and conducting experiments, the Web interface can be very helpful in finding configuration errors. Furthermore, the Web interface can also be used to set up experiments in a drag and drop-like manner. However, we suggest using this only as a starting point and conduct experiments with a significant amount of runs using the batch processing discussed in Subsection 6.2.4.

6.3 Testbed Evaluation and Comparison to Network Simulations

This section presents the design and results of an exemplary experiment conducted on the testbed. The selected experiment investigates the influence of different forwarding strategies on the data delivery performance of NDN. We use this experiment to asses the performance of the discussed forwarding strategies (cf. Subsection 4.1.3) in a physical deployment scenario. Furthermore, the obtained results are compared to the same experiment conducted using ndnSIM 2.0 [37]. This provides further insights on the challenges for effective content distribution in NDN with respect to forwarding when deployed on physical networks. Finally, we push the Pis to their limits to assess the maximum work load these devices can handle providing an indication for which kind of experiments (workload) the testbed is powerful enough.

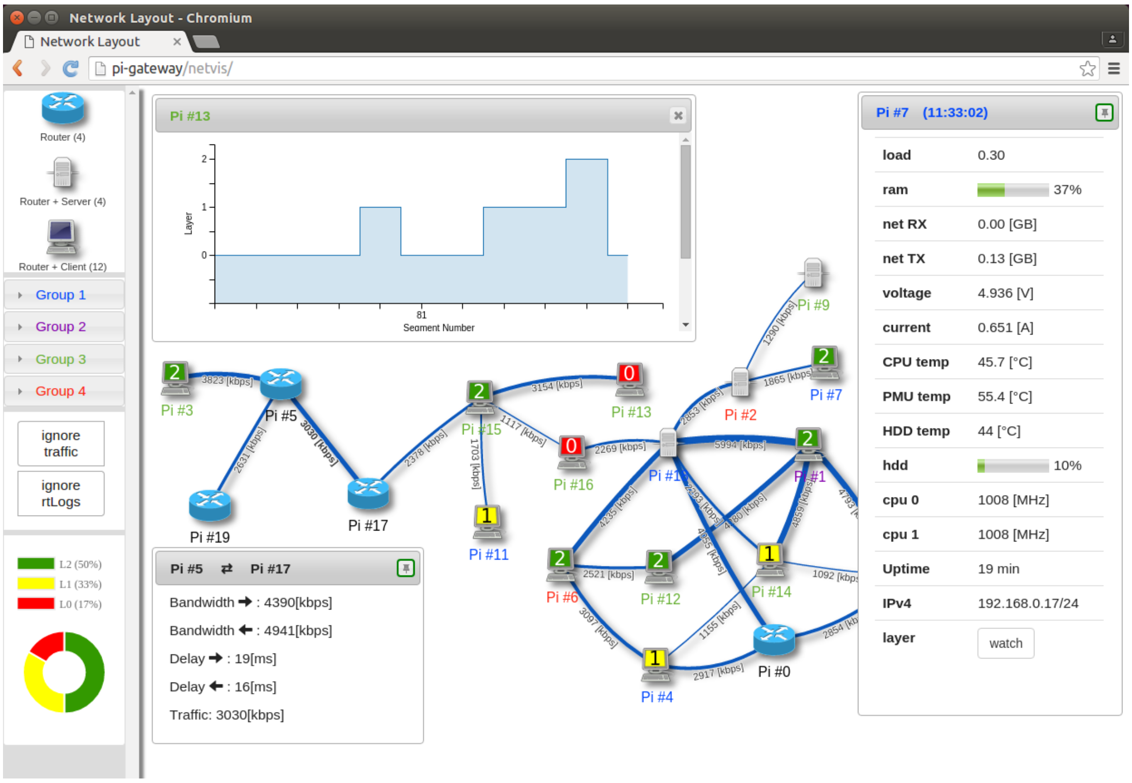


Figure 6.7: Snapshot of the Web interface which can be used to monitor ongoing emulations in near real-time or to configure experiments in a drag and drop-like manner.

6.3.1 Evaluation Scenario

This experiment investigates the performance of the following forwarding strategies: Broadcast [38], BestRoute [38], NCC [38], RFA [95], and SAF [108] (cf. Subsection 4.1.3). Unfortunately, we are not able to consider iNRR [107] and OMP-IF [127] for investigation, since iNRR requires an oracle (which does not exist in the real world) and OMP-IF distinguishes strictly between router and client nodes. However, due to the limited number of devices available, we have to use a single node for both roles. We have 20 BPI-R1 at hand, and therefore we decided to model a typical peer-to-peer overlay network for our experiment. We generate network topologies consisting of $n = 20$ nodes using the Erdős Rényi model [151]. The probability of creating a link between two nodes was set to $p \in \{0.10, 0.15, 0.20\}$ and the link delays $d \in [5\ ms, 20\ ms]$ are drawn from a uniform distribution (this does not include the processing delay of packets by the device). We ensure that the generated graph is connected by omitting topologies not satisfying this condition. For the bandwidth capacity of a link we consider three different settings (uniformly drawn): LowBW, i.e., from the interval $[2000\ kbps, 3000\ kbps]$; MediumBW, i.e., from $[3000\ kbps, 4000\ kbps]$; and HighBW, i.e., from [4000 $kbps$, 5000 $kbps$]. Each node maintains a 250 MB large cache using the FIFO replacement strategy. For each simulation/emulation, 4 nodes are randomly assigned as servers initially providing unique content to 12 clients (the remaining 4 nodes only act as routers). Each client requests content with a constant bitrate of approximately 2000 $kbps$ from a single server, with a Data packet size of 4 kiB. The pairing between client and server is randomized for each run. Furthermore, we added all available routes to the NFD's FIB to ensure the all strategies can make use of their multi-path capabilities. In total 40 runs are performed for each configuration simulating/emulating (ns3-ndnSIM/testbed) 30 minutes of network traffic.

6.3.2 Results Obtained from the Testbed

Figures 6.8, 6.9, and 6.10 illustrate the results of the experiment when conducted on the testbed. The *Interest satisfaction ratio*, which provides the ratio of received Data packets and generated Interests per client, is shown on the top left of individual figures. It is evident from the results that SAF outperforms its competitors in all scenarios regarding Interests satisfied, especially if resources and connectivity are low. If the bandwidth resources and the network connectvity are increased, the other algorithms start to catch up, particularly BestRoute is a strong competitor. This observation is consistent with the results obtained in Section 4.3 and allows us to conclude that SAF performs well in peer-to-peer like networks and also in classical Internet-like topologies. We will discuss the differences between the testbed results and network simulations in detail providing a direct comparison in Subsection 6.3.3.

The top right graphs in Figures 6.8, 6.9, and 6.10 depict the overall *cache hit ratio* per network node. Here most of the algorithms perform similar in scenarios with low and moderate resources and network connectivity. Only RFA performs significantly worse than the other algorithms, due to its principle of load balancing leading to a lower cache hit ratio. When network connectivity is high (cf. Figure 6.10b) one can observe that Broadcast and NCC obtain the highest cache hit ratio outperforming SAF concerning this metric. However, one has to consider that those strategies excessively replicate Interests leading to congestion. Matching Figures 6.10a and 6.10b we can see that the high cache hit ratios of Broadcast and NCC do not necessarily result in effective content distribution, particularly in scenarios with low bandwidth resources . Therefore, we conclude that solely a high cache hit ratio is not a convincing performance indicator concerning effective content distribution.

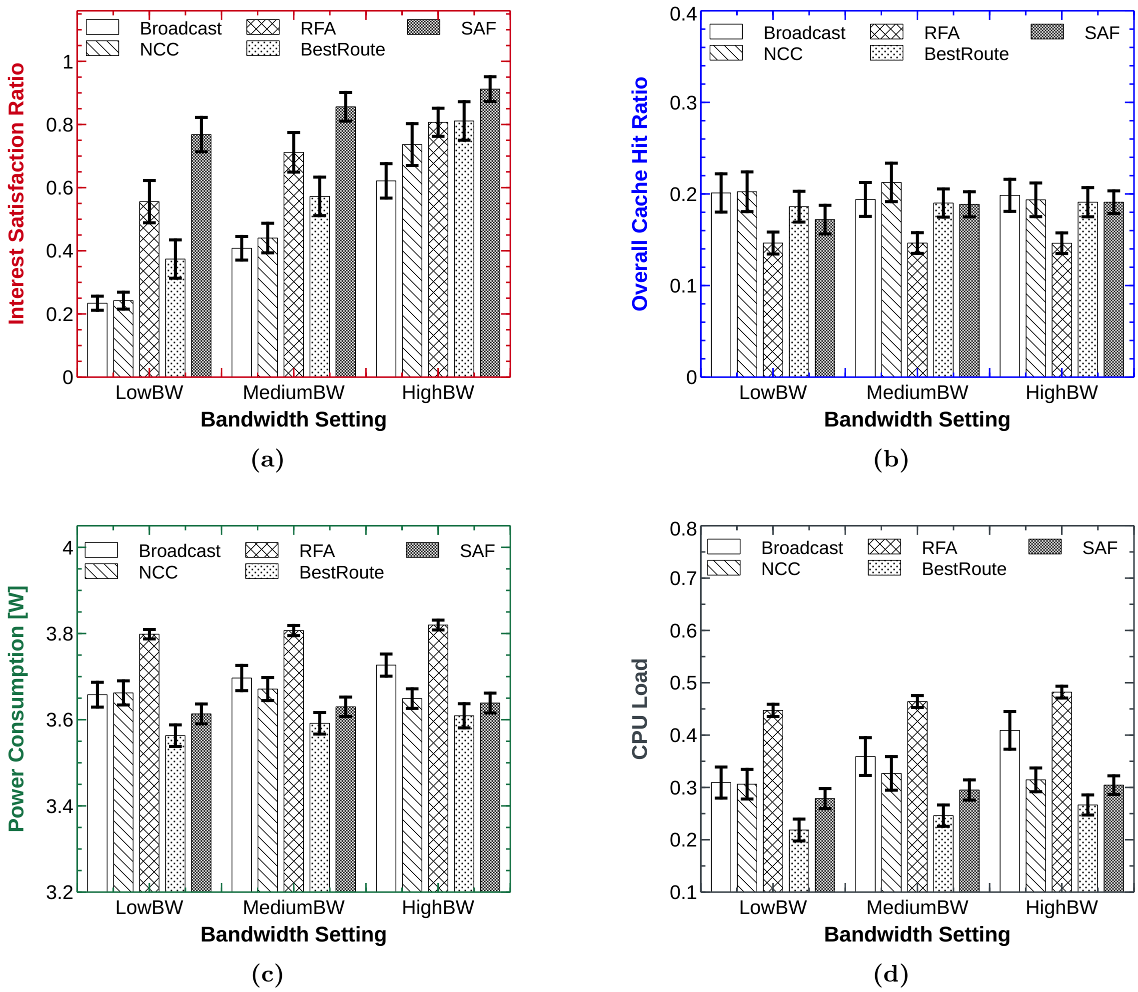


Figure 6.8: Evaluation results obtained from the testbed (link probability of **p=0.10**). 95% confidence intervals for the observed parameters are shown.

The bottom left graphs in Figures 6.8, 6.9, and 6.10 depict the average *power consumption* and the bottom right graphs depict the average *CPU load* per node. These results can only be obtained from the testbed. This makes them particularly interesting for investigating the real-world requirements of algorithms. It is evident that both, SAF and BestRoute, consume less power on the single-board computers than the other algorithms regardless of the available network resources or connectivity. For an environment with scarce energy resources, these algorithms may thus be better suited. SAF tends to use a bit more power and CPU resources than BestRoute. Considering the algorithm's design we conclude that has the following reason. Since BestRoute focuses on the single best delivery path, some devices (e.g., network nodes with low degree centrality that are not resided on the best path) may

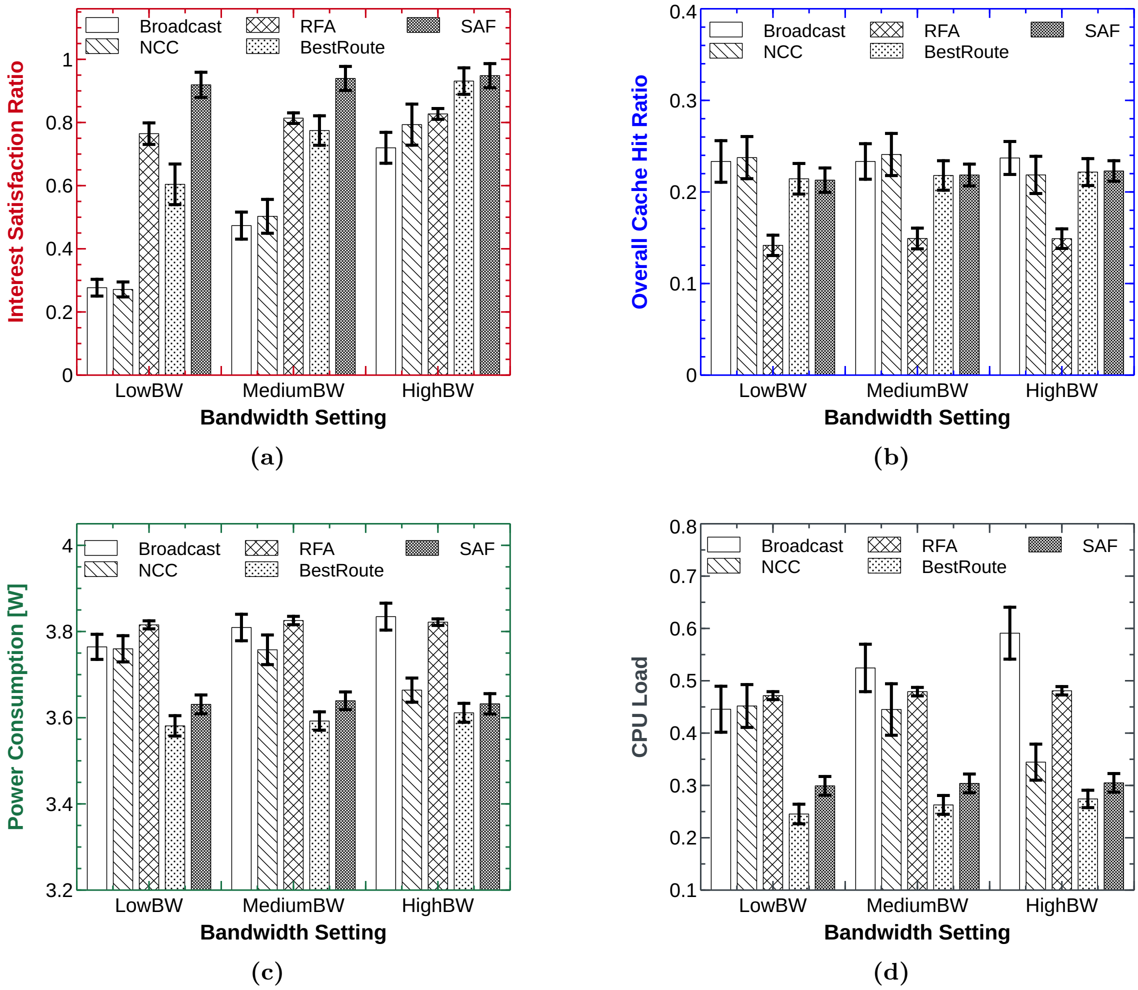


Figure 6.9: Evaluation results obtained from the testbed (link probability of **p=0.15**). 95% confidence intervals for the observed parameters are shown.

not receive any or only very little traffic allowing them to under-clock their CPU, thus saving power. SAF instead may also use these devices to retain additional throughput leading to slightly higher power consumption. Concerning RFA we can observe that it tends to use a lot of resources, particularly in very low-connected topologies. We know that RFA performs a kind of load balancing, thus permanently using all available paths. Therefore, none of the nodes are able to save power, e.g., by under-clocking their CPU. Nevertheless, in scenarios with moderate and high resources and network connectivity, the resource consumption of Broadcast and NCC drastically increases even wasting more resources than RFA. This is again rooted in their excessive replication of Interest messages that leads to heavy load on the individual nodes.

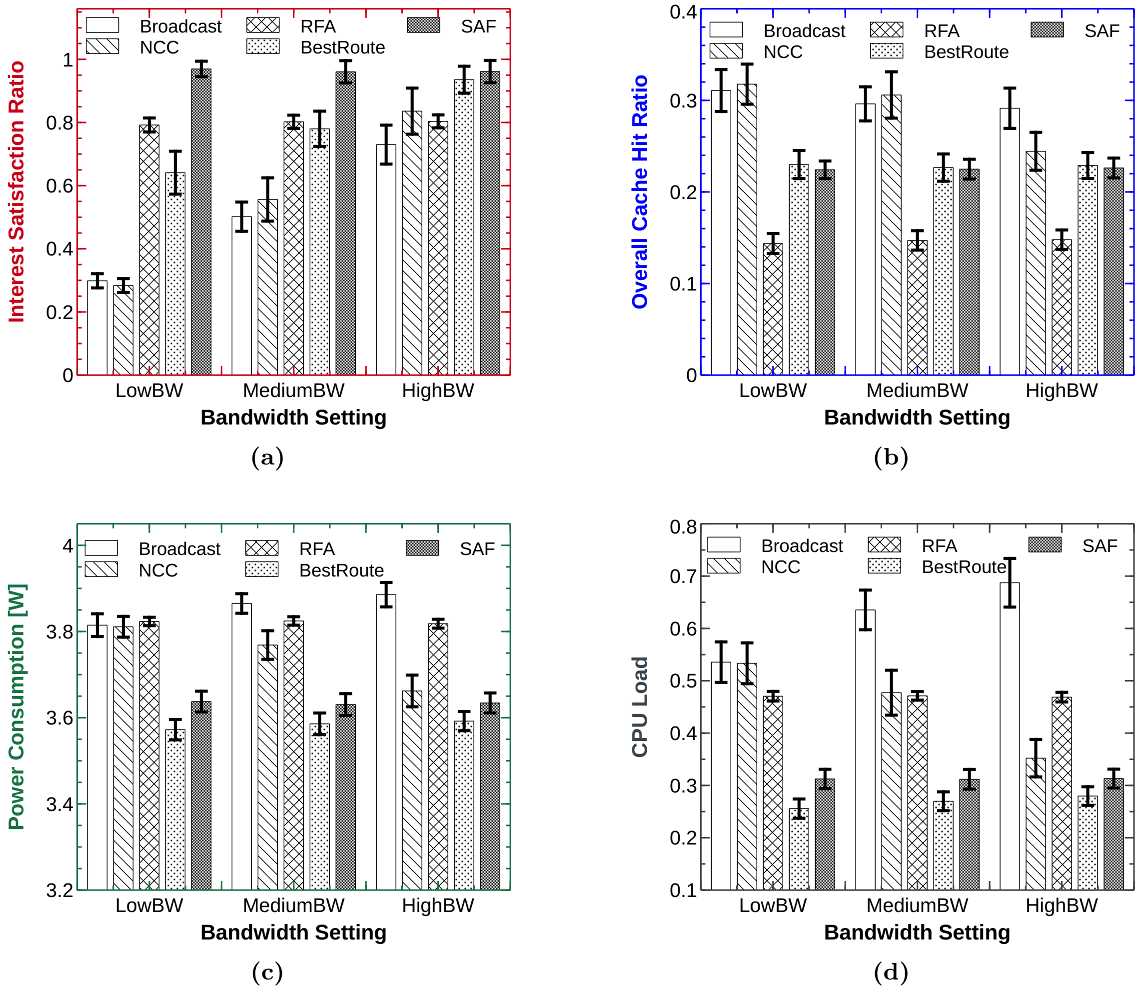
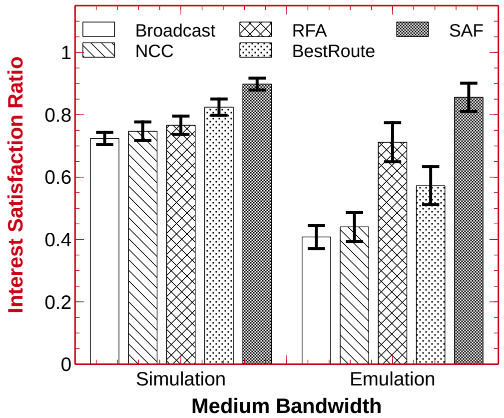


Figure 6.10: Evaluation results obtained from the testbed (link probability of **p=0.20**). 95% confidence intervals for the observed parameters are shown.

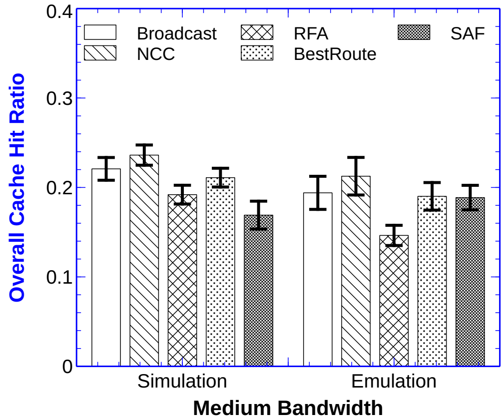
6.3.3 Comparing the Results of ndnSIM and the Testbed

We have conducted the same experiments as in the previous section using the network simulator ns-3/ndnSIM [37]. We anticipate that the observed performance of the algorithms in the simulator is better than on the testbed. We expect this because ndnSIM does not consider the overhead caused by the underlying protocols (UDP: 8 bytes, IPv4: 20 bytes, Ethernet: 18 bytes) that are required for communication on the testbed. Instead, the simulator uses a virtual channel to transmit NDN packets with nearly no overhead (2 bytes for a protocol flag).

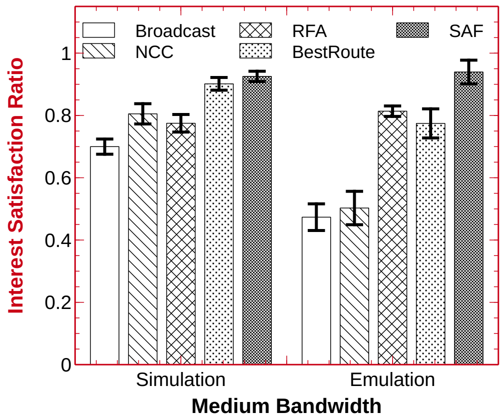
As previously mentioned, using ndnSIM we are not able to obtain figures for the power



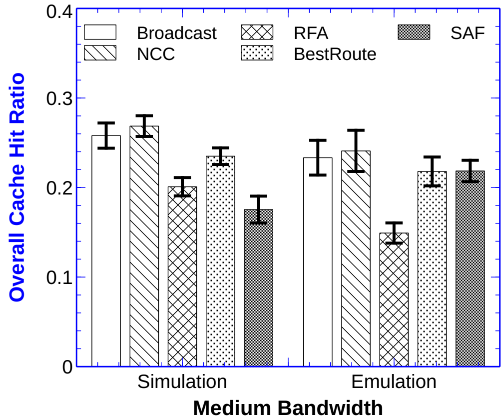
(a) link probablity $p = 0.10$



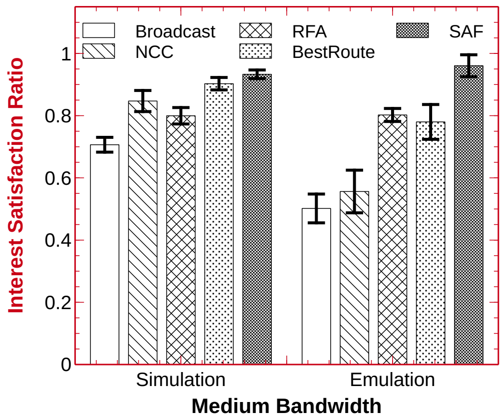
(b) link probablity $p = 0.10$



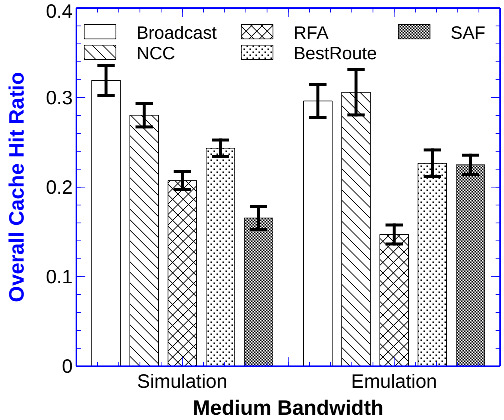
(c) link probablity $p = 0.15$



(d) link probablity $p = 0.15$



(e) link probablity $p = 0.20$



(f) link probablity $p = 0.20$

Figure 6.11: Comparison of simulations and emulations considering the scenario *MediumBW* with varying link probabilities p for the Erdős Rényi model [151].

consumption or the CPU load required by the algorithms. Therefore, Figure 6.11 depicts only the results for the Interest satisfaction ratio and the cache hit ratio. Please note that we compare only the results for the setting *MediumBW* with link probabilities $p \in \{0.10, 0.15, 0.20\}$ to ease readability; the results for the other two settings are similar. The left set of bars in Figures 6.11a – 6.11f depict the results obtained by conducting simulations, while the right set of bars depict the results obtained conducting emulations on the testbed. It is evident from Figure 6.11 that especially Broadcast and NCC suffer from a severe performance loss when executed on the testbed. This is definitely due to the larger overhead in the testbed as these strategies tend to replicate Interests. Considering that the average Interest size during simulation was about 40 bytes (according to the ndnSIM documentation the minimum size of an Interest message is 14 bytes), an additional overhead of $8 + 20 + 18 = 46$ bytes roughly doubles the size of each forwarded Interest. Also, BestRoute is suffering from a minor performance loss (especially in the case of low network connectivity), however, since this strategy does not replicate Interests the impact is substantially smaller. Interestingly, the forwarding strategies RFA and SAF do not suffer from a significant performance loss (only RFA sustains a small loss in the case of low network connectivity, cf. Figure 6.11a). This is due to their basic principles of distributing traffic among the interfaces with the lowest load without creating replicas of Interests avoiding Interest drops at congested interfaces. Regarding the cache hits illustrated in Figure 6.11, there is no significant difference observable for the algorithms Broadcast, NCC and BestRoute as their confidence intervals are overlapping. Surprisingly, SAF is able to maintain a slightly higher cache hit ratio when executed on the testbed, while for RFA exactly the opposite is true. Unfortunately, we can not provide a scientifically sound explanation for this phenomenon. However, we think that the phenomenon could be caused by the randomness of the algorithms. Both SAF and RFA heavily use random numbers. While both implementations for the simulator uses the ns-3 based *UniformRandomVariable* [86], the implementations for the physical hardware depends on the *std::random_device* of the C++ standard libraries.

6.3.4 Testbed Limitations

To assess the capabilities of the BPI-R1 we conducted another experiment measuring the CPU usage of the NFD with respect to the processed packets. We observe a single node that has to forward/process a varying number of Interest and Data packets. Figure 6.12

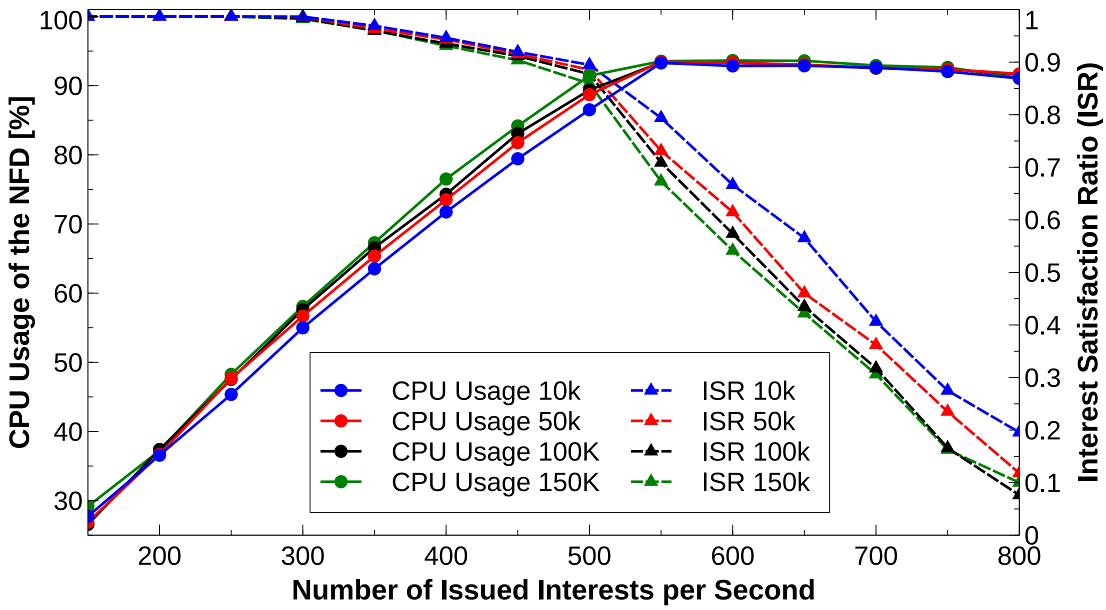


Figure 6.12: The packet processing capabilities of a BPI-R1 with respect to varying cache sizes (10k-150k entries). Measurements were started with full caches.

illustrates the result of this experiment, where we varied the number of issued Interests from 150 to 800 packets per second (one satisfied Interest results in two processed packets, i.e., an additional Data packet). Furthermore, we conducted this experiment with different cache sizes. The employed cache sizes are: 10k, 50k, 100k and 150k entries. We start the individual runs always with full caches to force extensive entry replacement to assess its influence on the CPU load. Figure 6.12 shows that the CPU load of the NFD increases proportionally with the number of processed Interests. Also larger cache sizes increase the CPU load, as the replacement of entries becomes more expensive. With 500 issued Interests the BPI-R1 reaches its capacity limits leading to drastic packet loss if more packets have to be processed. Although the BPI-R1 maintains a dual-core CPU, the NFD [38] is not capable of using more than one core limiting the Pi's capabilities. One can translate this result in a possible data rate a BPI-R1 may handle using the NFD on a single CPU core. Assuming a Data packet size of 8 kiB (maximum currently supported by the NFD), a Pi may handle at most a data rate of $500 \cdot 8 \cdot 8192 = 32768$ *kbps*. If researchers require more than 500 forwarded Interest/Data pairs per second and/or a higher data rate for their experiments, there are two options: *i*) Use more powerful devices. We expect that there will be a successor for BPI-R1 available shortly, with similar hardware as the recently released BPI-M3 (octa-core CPU with 8x1.8GHz). *ii*) Update the NFD code base to make use of multiple CPU cores.

Please note that for all our experiments in this chapter the digital signing of packets as foreseen by [15] is disabled. Using public-key cryptography [20] for real-time packet signing without dedicated cryptographic hardware is not feasible even with powerful desktop CPUs.

6.4 DAS-based Multimedia Streaming Using the Testbed

This section presents an evaluation using Dynamic Adaptive Streaming (DAS)-based multimedia streaming on the testbed. For that end we have implemented a DAS consumer and server application based on NDN technology. The applications are available at `http://icn.itec.aau.at` and are licensed under the MIT license. The consumer application has been implemented following the same principles as the applications used for the performance evaluation of DAS-based streaming in NDN presented in Section 3.2.3. The consumer supports two client-based adaptation strategies, which are equivalent to the rate- and buffer-based approaches presented in Subsection 3.2.1.b. First, Subsection 6.4.1 discusses the evaluation scenario, which is quite different to the scenario investigated in Subsection 3.2.3. Again this is due to the limited number of single board computers at hand. Subsection 6.4.2 presents and discusses the results obtained from the experiment.

6.4.1 Evaluation Scenario

For this scenario we model a peer-to-peer like DAS streaming scenario. The number of employed nodes is $n = 20$. We use the Erdős Rényi model [151] to generate a random topology for each emulation run. The edge probability for the Erdős Rényi model was set to $p = 0.15$. In total 25 runs per setting are performed. For each run 12 nodes where selected as consumers and 4 nodes as content provider. The remaining nodes do not actively participate in the streaming scenario, however, forward traffic like ordinary routers. The link capacities for each edge are drawn from a uniform distribution $\mathcal{U}_c(3000kbps, 4000kbps)$. The link delays for each hop are drawn from $\mathcal{U}_d(1ms, 5ms)$ (this does not include the processing delay of packets by the device). Clients consume SVC-based content taken from [104], which has already been used and discussed in Subsection 3.2.3. Therefore, we omit a detailed discussion about the dataset here. We assume a uniform content popularity for this scenario. The experiment investigates the DAS-based multimedia-streaming performance using NDN-based communication on physical devices considering different Interest forwarding strategies (Broadcast [38], BestRoute [38], NCC [38], RFA [95], SAF [108]) and their interplay with the two client-side rate- and buffer-based adaptation algorithms presented in Subsection 3.2.1.b.

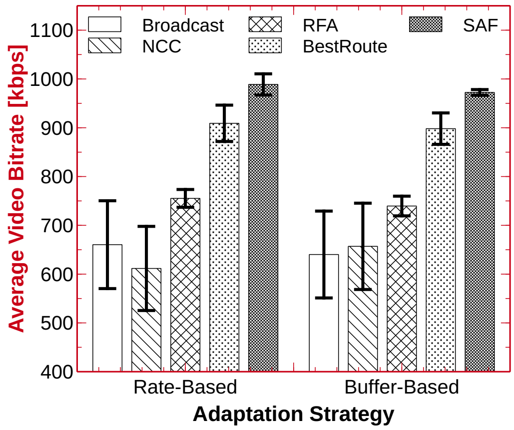
6.4.2 Results

Figure 6.13 shows the result for the multimedia streaming scenario conducted on the testbed. Again SAF is able to outperform all other forwarding strategies in terms of average video bitrate (cf. Figure 6.13a). This result is consistent with Section 3.2.3. However, the results in Section 3.2.3 showed that the average video bitrate is slightly higher if a buffer-based adaptation strategy is employed on the client, which can not be observed in the testbed results. Here, we can not observe a significant difference between both adaptation strategies concerning the average video bitrate for this scenario. We assume that this is due to employed processing delay of the individual nodes in conjunction of the employed SVC-encoded dataset. The individual segments (layers) are very small and as both client-based adaptation strategies request them in a strictly consecutive manner, this impedes content distribution efficiency.

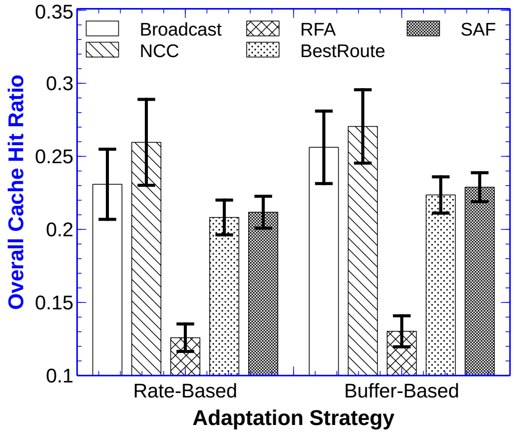
Regarding the cache hit ratio, the results are similar as in the previous Section 6.3. Again Broadcast and NCC are able to outperform the other algorithms, and especially RFA performs badly. However, as already mentioned in the previous section, the cache hit ratio has to be considered in conjunction with the satisfied Interests or as in this case with the average video bitrate. A lower video bitrate suggests that many clients retrieve the base layer only. Therefore, cache hits are more likely because less content traverses the network.

Figure 6.13c shows the number of representation switches per client. This result is again consistent with the observations in Section 3.2.3. Using a rate-based adaptation logic, clients are not able to predict the correct download bit-rate due to network-inherent caching, multi-path transmission and varying segment sizes (cf. Figure 5.2). This leads to extremely large number of representation switches impairing user satisfaction. Considering a user satisfaction model, e.g., the one presented in Section 5.3.2.b, makes this fact obvious. A buffer-based adaptation logic instead guarantees a stable playback with significantly fewer representation switches having a positive effect on the user satisfaction.

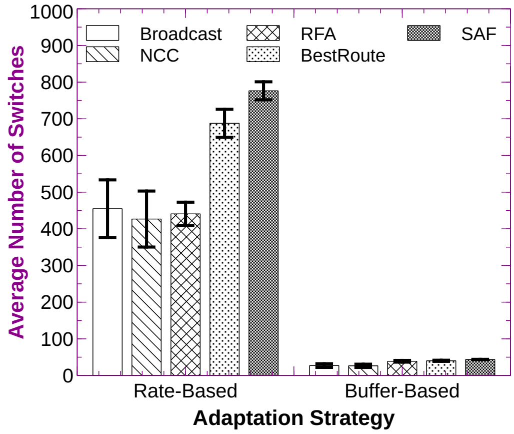
The employed client-based adaptation strategy has no significant influence on the power consumption of the devices nor on the CPU load. However, the forwarding strategies do, as discussed in Subsection 6.3.2. Again SAF and BestRoute are able to obtain the lowest amount of resource consumption. Using Broadcast and NCC the devices require more than 0.1 watt more energy on average for this scenario.



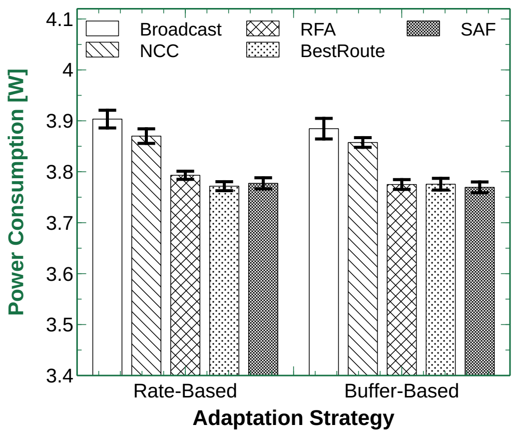
(a)



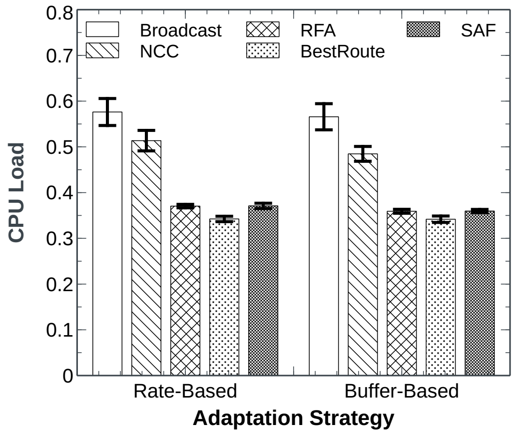
(b)



(c)



(d)



(e)

Figure 6.13: Evaluation results obtained from the testbed for the multimedia streaming scenario. 95% confidence intervals for the observed parameters are shown.

6.5 Conclusion and Outlook for the Testbed

This chapter presented a framework that enables researchers to readily deploy an NDN testbed on low-budget single-board computers, so-called Banana Pis. All required tools/applications were introduced and an open source contribution provides disk images, scripts and guidelines enabling researchers to set up their own testbed. The necessary materials are provided for download at `http:/icn.itec.aau.at`. We presented two evaluations that showed the potential benefit of the testbed that allows conducting experiments on physical devices. The conducted evaluations confirm our results obtained by network simulations presented in Chapters 3 and 4. Especially, the good performance of SAF in both scenarios (classical content distribution and on-demand multimedia streaming) could be approved. However, a direct comparison of network emulations and simulations showed that the abstraction introduced by simulations hides important details (overheads, resource usage, etc.). This is an important point that should be considered pursuing a real deployment of ICN/NDN technology.

For the future we expect that the testbed will help ICN/NDN researchers assess the performance of their proposals in more detail. Conducting experiments on physical networks and devices will provide researchers with deeper insights into their algorithms, reveal further challenges and potentially support the progress in the research area of data-oriented communication. The testbed can be perfectly used for experiments investigating the hot research topics in NDN. For example, new proposals for caching, forwarding, routing and naming can be investigated considering aspects like power consumption, memory consumption and CPU load under realistic conditions. Particularly, the investigation of the testbed limitations showed that the current NDN-based implementation is not very efficient. To achieve high data rates the software implementation needs improvement or hardware-based implementations will be necessary. This is of particular importance, especially, when considering that the Banana Pi Router maintains hardware components with similar performance as today's home routers.

CHAPTER 7

Findings, Conclusions and Future Work

"Time is what keeps everything from happening at once."

— Ray Cummings, 1887* - 1957†

In this thesis we discussed the potential benefits of Information-Centric Networking (ICN) and showed how this concept can be improved to support effective multimedia dissemination in future networks. Initially, we recapitulated the proposed mechanisms that are promoted as foundation for efficient content delivery. Related work fosters high expectations in both industry and academic research. In contrast, this thesis challenges the expectations with the objective to reveal performance gaps and weak spots. Furthermore, we analyse the identified issues and provide solutions to open ICN-related challenges. In the following we summarize the major contributions and findings of this work.

Taking Named Data Networking (NDN) as an ICN representative, we investigated the theoretical and practical content delivery performance of this technology. The results indicate significant performance gaps: Based on the use case of pull-based Dynamic Adaptive Streaming (DAS), we conducted a detailed performance investigation concerning multimedia content delivery in NDN. We developed an analytical model to assess the theoretical streaming performance of DAS-consumers in an NDN-based network, considering multipath transport and network-inherent caching. The comparison of the obtained results concerning the analytical model and the practically achievable performance showed that there is a significant gap. Furthermore, we observed that classical client-based adaptation approaches using rate-based measurements are not suitable for DAS-based streaming in NDN. However, we also observed that buffer-based approaches are. This is due to the fact that client applications are not able to correctly estimate the available resources due to multi-path transport and in-network caching, leading to highly fluctuating observations. Nevertheless, considering all results, we can conclude that ICN technology performs significantly better than today's IP-based networks concerning content delivery efficiency. However, the theoretical analysis suggests that the ICN/NDN concept provides more potential than can be realized by today's technology and implementations.

We identify shortcomings and weaknesses in NDN, in particular concerning existing Interest forwarding strategies, and provide a novel probability-based solution: Another contribution of this thesis is the identification of open problems that cause the previously mentioned performance gap. This thesis focuses on the area of forwarding in NDN, although, there is also potential for improving ICN-related areas such as routing and caching. We identified that existing forwarding strategies are not capable of utilizing the full range of opportunities that are provided by NDN's architecture. NDN's stateful forwarding plane sustains more potential. In particular, this includes the expectations on effective multi-path forwarding, autonomous error recovery (e.g., from link failures), and content and context-aware data transmission. Most of the existing strategies are content-agnostic and are focused on narrow objectives unable to benefit from NDN's flexible forwarding plane. Therefore, we designed and implemented a novel probabilistic forwarding strategy named Stochastic Adaptive Forwarding (SAF) for NDN. This strategy is capable of providing multi-path transport and is able to locate nearby cached content replicas unknown to the routing plane. Moreover, SAF is able to deal with invalid and incomplete routing information, which allows independent and fast recovery from link failures. Furthermore, this strategy can be fed with context information supplied by the network operator. We showed that this strategy is able to outperform existing forwarding strategies in multimedia streaming and in classical data distribution scenarios in Internet-like topologies as well as in unstructured peer-to-peer like network overlays.

We consider context information within NDN's forwarding plane enhancing content distribution with respect to specific consumer/application demands: A further contribution of this thesis is the consideration of context information in the forwarding plane. We showed that this idea provides significant opportunities to enhance the decisions that are made in the forwarding plane. Therefore, we adopt SAF's design for considering additional context information (e.g., application requirements) in the forwarding plane. We investigated a scenario that employs applications with different demands (VoIP, video streaming, data transfer) and developed a model that allowed us to feed the requirements into SAF's adaptation engine by ordering the importance of contents/applications. We conducted network experiments and measured the relevant Quality of Service (QoS) parameters for the individual applications. The measurements were used as input for existing user satisfaction models. With respect to the employed Quality of Experience (QoE) metrics, our results suggest that context awareness in the forwarding plane enhances consumer satisfaction significantly.

We provide a framework for a low-cost NDN-based testbed enabling researchers to conduct experiments on physical hardware. Further, we show that investigations on physical hardware reveals further challenges: The final contribution of this thesis is the provision of an NDN-based testbed. We have noticed that ICN/NDN-based research is mainly evaluated by network simulations and analytical methods. While both approaches are valid and necessary, experiments on physical hardware are important since investigations in purely synthetic environments may hide important challenges and shortcomings. Therefore, we designed and implemented a testbed framework that allows researchers to readily deploy an NDN-based network to conduct network emulations. The testbed allows researchers to investigate their proposed algorithms under realistic conditions, also providing opportunities to measure parameters that are not observable in simulations, e.g., power consumption. The testbed is based on low-cost single-board computers, so-called Banana Pi Routers. We conducted experiments on the testbed assessing its capabilities and constraints. We showed that the testbed hardware is powerful enough to conduct significant performance investigations concerning most research areas in the domain of ICN. We have used the deployed testbed at our institute and investigated the performance of SAF and its competitors on the testbed concerning classical data delivery and multimedia streaming. Furthermore, we have compared the emulation results to those obtained from network simulations. We could show that although network simulations hide some important details, the general observations are definitely reasonable and their overall trend matches with the results obtained by network emulations.

Considering all results and findings obtained in this thesis, we predict a bright future for ICN technology. The conducted experiments and investigations emphasize the significant advantages of content-oriented networking over today's Internet infrastructure, particularly for content distribution at a large scale. Nevertheless, this technology definitely has a long way to go before a global deployment can be considered. The transition may take years, starting with small steps, such as using ICN technology for content distribution via overlay networks. However, in the end, the benefits may trigger a global change as it happened with the transition from circuit to packet switching networks.

Future Work

As briefly indicated in the individual chapters of this thesis, a vast amount of future work remains before ICN/NDN can provide humanity with effective multimedia distribution. In the following we outline important challenges and questions for further research in this area. Concerning content distribution the introduction of a distinguished *strategy layer* as foreseen in NDN holds enormous potential. Research that aims at increasing content distribution efficiency in future networks should focus on this concept. While topics of interest definitely include areas such as *in-network caching* and *routing*, this thesis contributed to the yet widely unexplored area of *adaptive multi-path forwarding*. In this area we see tremendous potential in increasing data transmission efficiency. One interesting challenge is the search for nearby content replicas independent from the routing plane. Here, existing work either relies on *oracles* [107], or uses simple (e.g., broadcast-like) mechanisms flooding the network with requests. Although, our approach SAF uses only traffic that would be dropped anyway for searching replicas, it uniformly probes on all available faces to search for new paths to content replicas. We suggest that future work should investigate more sophisticated mechanisms. For instance, as inspired by [135], an entropy-based approach may be one opportunity to further enhance this field.

The concept of the strategy layer is based on the availability of content and context information in the network. This may include various information (e.g., content characteristics, QoS requirements, content popularity) that has to be gathered, represented and delivered to the individual nodes. Most context and/or context-aware approaches in the strategy layer exploit ICN's principle of naming individual content items and insert the necessary content as part of the employed name. While this is a simple approach, a dedicated and specified mechanism of carrying and representing this information is necessary. We suggest that future work should investigate this topic in detail. In particular, researchers should explore new avenues utilizing context information in the network guiding content distribution in future communication infrastructures.

Another yet unsolved challenge comes apparent when investigating real-time communication, e.g., VoIP. Many ICN approaches such as NDN and CCN are strictly based on a pull-based communication model providing yet no opportunities to simply push content to consumers. This leads to two problems. First, content distribution efficiency may be reduced as every single data chunk has to be requested by the consumer introducing unnecessary overhead. Second, delay sensitive applications may suffer from an unnecessary doubling of the delivery delay. Current solutions include ideas such as persistent (a.k.a. long-term) Interests [56, 152] and the requesting of yet not existing content [140]. The Interests proposed by [140] stay alive at the providers and are satisfied at the moment the data has been generated. However, the idea of persistent Interests impedes the principal idea of adaptive multi-path forwarding, while the pre-requesting of content is not always possible (e.g., unknown data generation rate). We suggest that a solution to this challenge should enhance real-time communication such as video conferencing in ICN significantly. Furthermore, at this point we want to draw the reader's attention to the yet slightly neglected area of online gaming (i.e., massively multiplayer online games) concerning ICN research.

Probably, the most difficult challenge is the deployment of ICN technology in existing network infrastructures. This is not possible without significant support and investment from big players in the telecommunication industry. While the NDN community does its best to provide first steps including the Networking Forwarding Daemon (NFD) [38] enabling NDN-based communication on physical hardware, the implementation lacks of performance and efficiency to support communication at regular line speed. Therefore, it is rather suitable for research than for real deployment scenarios at a large scale. Future work should supply evidence that ICN technology is ready to be adopted by industry such that sufficiently financial support is dedicated to hardware-based implementations. Furthermore, more effort should be invested in standardization of ICN technology so that industry and academia may join their forces more easily.

Bibliography

[1] Sandvine, "The Global Internet Phenomena Report(s) 2011-2016," Available: https://www.sandvine.com/resources/resource-library.html, Accessed: 2016-07.

[2] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2014-2019," 2015, Accessed: 2016-05. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[3] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A Brief History of the Internet," *SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 22–31, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1145/1629607.1629613

[4] J. Postel, "Internet Protocol," RFC 791 (INTERNET STANDARD), Internet Engineering Task Force, Sep. 1981. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2010.05.010

[6] EC FIArch Group, "Fundamental Limitations of Current Internet and Path to Future Internet," March 2011, Accessed: 2016-05. [Online]. Available: http://www.future-internet.eu/publications

[7] The Internet Corporation for Assigned Names and Numbers (ICANN), "Available Pool of Unallocated IPv4 Internet Addresses Now Completely Emptied," 2011, Accessed: 2016-05. [Online]. Available: http://www.icann.org/en/news/releases/release-03feb11-en.pdf

[8] G. Pallis and A. Vakali, "Insight and Perspectives for Content Delivery Networks," *Communications of the ACM*, vol. 49, no. 1, pp. 101–106, Jan. 2006. [Online]. Available: http://dx.doi.org/10.1145/1107458.1107462

[9] M. Handley, "Why the Internet only just works," *BT Technology Journal*, vol. 24, no. 3, pp. 119–129, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10550-006-0084-z

[10] National Science Foundation (NSF), "Future Internet Architecture Project," 2010, Accessed: 2016-06. [Online]. Available: http://www.nets-fia.net/

[11] European Future Internet Assembly (FIA), "European Future Internet Portal," 2008, Accessed: 2016-06. [Online]. Available: http://www.future-internet.eu/

[12] European Commeission, "Future Internet Research and Experimentation (FIRE)," 2010, Accessed: 2016-06. [Online]. Available: https://www.ict-fire.eu/fire/

[13] D. Schwerdel, B. Reuther, T. Zinner, P. Müller, and P. Tran-Gia, "Future Internet Research and Experimentation: The G-Lab Approach," *Computer Networks*, vol. 61, pp.

102 – 117, 2014, special issue on Future Internet Testbeds – Part I. [Online]. Available: http://dx.doi.org/10.1016/j.bjp.2013.12.023

[14] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-Centric Networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, July 2012. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2012.6231276

[15] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1145/2656877.2656887

[16] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric Networking: Seeing the Forest for the Trees," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, ser. HotNets-X. New York, NY, USA: ACM, 2011, pp. 1:1–1:6. [Online]. Available: http://dx.doi.org/10.1145/2070562.2070563

[17] C. Ghali, G. Tsudik, and E. Uzun, "Network-Layer Trust in Named-Data Networking," *SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 12–19, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1145/2677046.2677049

[18] D. Smetters and V. Jacobson, "Securing Network Content," Citeseer, Technical Report, 2009, Accessed: 2016-07. [Online]. Available: http://www.parc.com/content/attachments/securing-network-content-tr.pdf

[19] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Publish-Subscribe Internetworking Security Aspects," in *Trustworthy Internet*, L. Salgarelli, G. Bianchi, and N. Blefari-Melazzi, Eds. Springer Milan, 2011, pp. 3–15. [Online]. Available: http://dx.doi.org/10.1007/978-88-470-1818-1_1

[20] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography.* CRC Press, 1996.

[21] D. R. Cheriton and M. Gritter, "TRIAD: A Scalable Deployable NAT-based Internet Architecture," 2000, Accessed: 2016-07. [Online]. Available: http://gregorio.stanford.edu/triad/

[22] M. Gritter and D. R. Cheriton, "An Architecture for Content Routing Support in the Internet," *USENIX Symposium on Internet Technologies and Systems*, 2001, Accessed: 2016-07. [Online]. Available: http://gregorio.stanford.edu/triad/

[23] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-oriented (and Beyond) Network Architecture," *SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 181–192, Aug. 2007. [Online]. Available: http://dx.doi.org/10.1145/1282427.1282402

[24] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," RFC 4423 (Informational), Internet Engineering Task Force, May 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4423.txt

[25] D. Mazières, M. Kaminsky, M. F. Kaashoek, and E. Witchel, "Separating Key Management from File System Security," *SIGOPS Operating Systems Review*, vol. 33, no. 5, pp. 124–139, Dec. 1999. [Online]. Available: http://dx.doi.org/10.1145/319344.319160

[26] D. Lagutin, K. Visala, and S. Tarkoma, *Towards the Future Internet*. Amsterdam, Netherlands: IOS Press, 2010, ch. Publish/Subscribe for Internet: PSIRP Perspective, pp. 75–84. [Online]. Available: http://dx.doi.org/10.3233/978-1-60750-539-6-75

[27] N. Fotiou, P. Nikander, D. Trossen, and G. Polyzos, "Developing Information Networking Further: From PSIRP to PURSUIT," in *Broadband Communications, Networks, and Systems*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, I. Tomkos, C. Bouras, G. Ellinas, P. Demestichas, and P. Sinha, Eds. Springer Berlin Heidelberg, 2012, vol. 66, pp. 1–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30376-0_1

[28] The Internet Corporation for Assigned Names and Numbers (ICANN), "Blackhawk Publish/Subscribe Prototype v0.3 for FreeBSD," 2010, Accessed: 2016-07. [Online]. Available: http://www.fp7-pursuit.eu

[29] C. Dannewitz, "Netinf: An Information-Centric Design for the Future Internet," in *Proceedings 3rd GI/ITG KuVS Workshop on The Future Internet*, 2009.

[30] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of Information (NetInf) - An Information-Centric Networking Architecture," *Computer Communications*, vol. 36, no. 7, pp. 721 – 735, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2013.01.009

[31] Ericsson AB (Project Lead) et al., "The FP7 4WARD Project," 2008, Accessed: 2016-07. [Online]. Available: http://www.4ward-project.eu/

[32] SAIL Project Partners, "Scalable and Adaptive Internet Solutions (SAIL)," 2010, Accessed: 2016-07. [Online]. Available: http://www.sail-project.eu/

[33] The Internet Corporation for Assigned Names and Numbers (ICANN), "Open Network of Information (OpenNetInf) Prototype," 2011, Accessed: 2016-07. [Online]. Available: http://www.netinf.org/open-source/

[34] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://dx.doi.org/10.1145/1658939.1658941

[35] Palo Alto Research Center, "CCNx Binary Release," 2015, Accessed: 2016-07. [Online]. Available: https://www.ccnx.org/

[36] V. Jacobson, J. Burke, L. Zhang, B. Zhang, K. Claffy, C. Papadopoulos, T. Abdelzaher, L. Wang, A. Halderman, and P. Crowley, "Named Data Networking Next Phase (NDN-NP) Project May 2014-April 2015 Annual Report," 2015, (Latest Progress Report of the Project) Accessed: 2016-07. [Online]. Available: http://named-data.net/wp-content/uploads/2015/06/ndn-ar2015.pdf

[37] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new Version of the NDN Simulator for NS-3," University of California, Los Angeles, Tech. Report NDN-0028, 2015, Accessed: 2015-11. [Online]. Available: http://named-data.net/publications/techreports/ndn-0028-1-ndnsim-v2/

[38] A. Afanasyev, J. Shi, B. Zhang, L. Zhang *et al.*, "NFD Developer's Guide," University of California, Los Angeles, Technical Report NDN-0021, 2014, Accessed: 2015-11. [Online]. Available: http://named-data.net/publications/techreports/nfd-developer-guide/

[39] Z. Zhu, C. Bian, A. Afanasyev, V. Jacobson, and L. Zhang, "Chronos: Serverless Multi-User Chat Over NDN," University of California, Los Angeles, Technical Report NDN-0008, 2012, Accessed: 2015-11. [Online]. Available: http://named-data.net/publications/techreports/trchronos/

[40] Z. Zhenkai and A. Afanasyev, "Let's ChronoSync: Decentralized Dataset State Synchronization in Named Data Networking," in *21st IEEE International Conference on Network Protocols (ICNP)*, Oct 2013, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1109/ICNP.2013.6733578

[41] I. Moiseenko and L. Zhang, "Consumer-producer API for Named Data Networking," in *Proceedings of the 1st International Conference on Information-centric Networking*, ser. ICN '14. New York, NY, USA: ACM, 2014, pp. 177–178. [Online]. Available: http://dx.doi.org/10.1145/2660129.2660158

[42] W. Shang, J. Thompson, M. Cherkaoui, J. Burkey, and L. Zhang, "NDN.JS: A Javascript Client Library for Named Data Networking," in *IEEE Conference on Computer Communications Workshops*, April 2013, pp. 399–404. [Online]. Available: http://dx.doi.org/10.1109/INFCOMW.2013.6970726

[43] J. Thompson and J. Burke, "NDN Common Client Libraries," University of California, Los Angeles, Technical Report NDN-0024, 2014, Accessed: 2015-11. [Online]. Available: http://named-data.net/publications/techreports/ccltechreport/

[44] NDN Project, "NDN Packet Format Specification 0.2-alpha Documentation," 2015, Accessed: 2016-07. [Online]. Available: http://named-data.net/doc/ndn-tlv/

[45] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive Forwarding in Named Data Networking," *SIGCOMM Compututer Communication. Review*, vol. 42, no. 3, pp. 62–67, Jun. 2012. [Online]. Available: http://dx.doi.org/10.1145/2317307.2317319

[46] H. Yuan, T. Song, and P. Crowley, "Scalable NDN Forwarding: Concepts, Issues and Principles," in *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*. IEEE, July 2012, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/ICCCN.2012.6289305

[47] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," *Computer Communications*, vol. 36, no. 7, pp. 779 – 791, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2013.01.005

[48] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "On the Role of Routing in Named Data Networking," in *Proceedings of the 1st International Conference on Information-centric*

Networking, ser. ICN '14. New York, NY, USA: ACM, 2014, pp. 27–36. [Online]. Available: http://dx.doi.org/10.1145/2660129.2660140

[49] L. Wang, M. Hoque, C. Yi, A. Alyyan, and B. Zhang, "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," University of Memphis and University of Arizona, Technical Report, 2012, Accessed: 2016-07. [Online]. Available: http://named-data.net/wp-content/uploads/TROSPFN.pdf

[50] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-Data Link State Routing Protocol," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13. New York, NY, USA: ACM, 2013, pp. 15–20. [Online]. Available: http://dx.doi.org/10.1145/2491224.2491231

[51] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in Named Data Networking," in *Proceedings of 22nd International Conference on Computer Communications and Networks (ICCCN)*, July 2013, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/ICCCN.2013.6614127

[52] J. Burke, P. Gasti, N. Nathan, and G. Tsudik, "Securing Instrumented Environments over Content-Centric Networking: The Case of Lighting Control and NDN," in *Proceedings of the 32nd Conference on Computer Communications Workshops*. IEEE, April 2013, pp. 394–398. [Online]. Available: http://dx.doi.org/10.1109/INFCOMW.2013.6970725

[53] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in Content-oriented Networking: Threats and Countermeasures," *SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 25–33, Jul. 2013. [Online]. Available: http://dx.doi.org/10.1145/2500098.2500102

[54] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik, "Cache Privacy in Named-Data Networking," in *Proceedings of the 33rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, July 2013, pp. 41–51. [Online]. Available: http://dx.doi.org/10.1109/ICDCS.2013.12

[55] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in Content-oriented Architectures," in *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking*, ser. ICN '11. New York, NY, USA: ACM, 2011, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1145/2018584.2018586

[56] C. Tsilopoulos and G. Xylomenos, "Supporting Diverse Traffic Types in Information Centric Networks," in *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '11. New York, NY, USA: ACM, 2011, pp. 13–18. [Online]. Available: http://dx.doi.org/10.1145/2018584.2018588

[57] J. Postel, "Transmission Control Protocol," RFC 793 (Internet Standard), Internet Engineering Task Force, Sep. 1981, updated by RFCs 1122, 3168, 6093, 6528. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[58] A.-V. T. W. Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889 (Proposed Standard), Internet Engineering Task Force, Jan. 1996, obsoleted by RFC 3550. [Online]. Available: http://www.ietf.org/rfc/rfc1889.txt

[59] J. Postel, "User Datagram Protocol," RFC 768 (INTERNET STANDARD), Internet Engineering Task Force, Aug. 1980. [Online]. Available: http://www.ietf.org/rfc/rfc768.txt

[60] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)," RFC 4585 (Proposed Standard), Internet Engineering Task Force, Jul. 2006, updated by RFC 5506. [Online]. Available: http://www.ietf.org/rfc/rfc4585.txt

[61] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004. [Online]. Available: http://dx.doi.org/10.1109/TIP.2003.819861

[62] R. Pantos and W. May, "HTTP Live Streaming," RFC Draft Version 16, Apple Inc., Apr. 2015. [Online]. Available: https://tools.ietf.org/html/draft-pantos-http-live-streaming-16

[63] A. Zambelli, "IIS Smooth Streaming Technical Overview," *Microsoft Corporation*, 2009, Accessed: 2015-11. [Online]. Available: http://tinyurl.com/MS-Smooth-Streaming

[64] Adobe Systems, "Adobe HTTP Dynamic Streaming," accessed: 2015-10. [Online]. Available: http://www.adobe.com/products/hds-dynamic-streaming.html

[65] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011. [Online]. Available: http://dx.doi.org/10.1109/MMUL.2011.71

[66] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ser. MMSys '11. New York, USA: ACM, 2011, pp. 133–144. [Online]. Available: http://dx.doi.org/10.1145/1943552.1943572

[67] ISO/IEC 32009-1:2014, "Information Technology – Dynamic Adaptive Streaming over HTTP (DASH) – Part 1: Media Presentation Description and Segment Formats," accessed: 2015-10. [Online]. Available: http://www.iso.org/

[68] S. Lederer, C. Müller, B. Rainer, C. Timmerer, and H. Hellwagner, "An Experimental Analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, USA, July 2013. [Online]. Available: http://dx.doi.org/10.1109/ICME.2013.6607500

[69] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner, "Adaptive Multimedia Streaming in Information-Centric Networks," *IEEE Network*, vol. 28, no. 6, pp. 91–96, Nov 2014. [Online]. Available: http://dx.doi.org/10.1109/MNET.2014.6963810

[70] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, and A. Bragagnini, "Offloading Cellular Networks with Information-Centric Networking: The Case of Video Streaming," *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, vol. 0, pp. 1–3, 2012. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2012.6263734

[71] S. Lederer, C. Müller, B. Rainer, C. Timmerer, and H. Hellwagner, "Adaptive Streaming over Content Centric Networks in Mobile Networks using Multiple Links," in *Proceedings of the IEEE ICC - Workshop on Immersive and Interactive Multimedia Communications over the Future Internet*, Budapest, 2013, pp. 677–681. [Online]. Available: http://dx.doi.org/10.1109/ICCW.2013.6649319

[72] D. Kulinski and J. Burke, "NDN Video: Live and Prerecorded Streaming over NDN," University of California, Los Angeles, Technical Report NDN-0010, 2012, Accessed: 2015-11. [Online]. Available: http://named-data.net/publications/techreports/trstreaming/

[73] L. Wang, I. Moiseenko, and L. Zhang, "NDNlive and NDNtube: Live and Prerecorded Video Streaming over NDN," University of California, Los Angeles, Technical Report NDN-0031, 2015, Accessed: 2015-11. [Online]. Available: http://named-data.net/publications/techreports/ndn-0031-1-ndnlive-ndntube/

[74] Y. Jin and Y. Wen, "PAINT: Partial In-Network Transcoding for Adaptive Streaming in Information Centric Network," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, May 2014, pp. 208–217.

[75] D. Posch, C. Kreuzberger, B. Rainer, and H. Hellwagner, "Using In-Network Adaptation to Tackle Inefficiencies Caused by DASH in Information-Centric Networks," in *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, ser. VideoNext '14. New York, NY, USA: ACM, 2014, pp. 25–30. [Online]. Available: http://dx.doi.org/10.1145/2676652.2676653

[76] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sept 2007. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2007.905532

[77] Z. Liu and Y. Wei, "Hop-by-Hop Adaptive Video Streaming in Content Centric Network," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/ICC.2016.7511585

[78] L. Fleischer, "Approximating Fractional Multicommodity Flow Independent of the Number of Commodities," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 24–31. [Online]. Available: http://dx.doi.org/10.1109/SFFCS.1999.814573

[79] C. Westphal, S. Lederer, D. Posch, C. Timmerer, A. Azgin, W. Liu, C. Mueller, A. Detti, D. Corujo, J. Wang, M. Montpetit, and N. Murray, "Adaptive Video Streaming over Information-Centric Networking (ICN)," RFC 7933 (Informational), Internet Engineering Task Force, Aug. 2016. [Online]. Available: http://www.ietf.org/rfc/rfc7933.txt

[80] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A Survey on Content-oriented Networking for Efficient Content Delivery," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 121–127, March 2011. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2011.5723809

[81] C. Tsilopoulos, G. Xylomenos, and G. Polyzos, "Are Information-Centric Networks Video-Ready?" in *20th International Packet Video Workshop (PV)*, Dec 2013, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/PV.2013.6691438

[82] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic In-network Caching for Information-centric Networks," in *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ser. ICN '12. New York, NY, USA: ACM, 2012, pp. 55–60. [Online]. Available: http://dx.doi.org/10.1145/2342488.2342501

[83] J. Stanik, "A Conversation with Van Jacobson," *Queue*, vol. 7, no. 1, pp. 9:8–9:16, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1145/1508211.1508215

[84] B. Manoj and A. H. Baker, "Communication Challenges in Emergency Response," *Commun. ACM*, vol. 50, no. 3, pp. 51–53, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1145/1226736.1226765

[85] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582 (Proposed Standard), Internet Engineering Task Force, Apr. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6582.txt

[86] NS-3 Consortium, "ns-3: A Discerete-Event Network Simulator," 2011, accessed: 2016-03. [Online]. Available: http://www.nsnam.org/

[87] G. Zhang, Y. Li, and T. Lin, "Caching in Information Centric Networking: A Survey," *Computer Networks*, vol. 57, no. 16, pp. 3128 – 3141, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2013.07.007

[88] M. Zhang, H. Luo, and H. Zhang, "A Survey of Caching Mechanisms in Information-Centric Networking," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015. [Online]. Available: http://dx.doi.org/10.1109/COMST.2015.2420097

[89] M. Dräxler and H. Karl, "Efficiency of On-Path and Off-Path Caching Strategies in Information Centric Networks," in *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, Nov 2012, pp. 581–587. [Online]. Available: http://dx.doi.org/10.1109/GreenCom.2012.82

[90] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "Less for More" in Information-Centric Networks," in *NETWORKING 2012*, ser. Lecture Notes in Computer Science. Springer, 2012, vol. 7289, pp. 27–40. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30045-5_3

[91] A. Ioannou and S. Weber, "A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–41, 2016. [Online]. Available: http://dx.doi.org/10.1109/COMST.2016.2565541

[92] S. Podlipnig and L. Böszörmenyi, "A Survey of Web Cache Replacement Strategies," *ACM Computing Surveys*, vol. 35, no. 4, pp. 374–398, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1145/954339.954341

[93] D. Rossi and G. Rossini, "Caching Performance of Content-Centric Networks under Multi-Path Routing (and more)," Telecom ParisTech, Technical Report, 2011, accessed: 2016-03. [Online]. Available: http://perso.telecom-paristech.fr/~drossi/paper/rossi11ccn-techrep1.pdf

[94] S. Barre, C. Paasch, and O. Bonaventure, "MultiPath TCP: From Theory to Practice," in *NETWORKING 2011*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6640, pp. 444–457. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20757-0_35

[95] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang, "Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks," in *Proceedings of 21st IEEE International Conference on Network Protocols*, Oct 2013, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1109/ICNP.2013.6733576

[96] G. Pavlou, N. Wang, W. K. Chai, and I. Psaras, "Internet-Scale Content Mediation in Information-Centric Networks," *Annals of Telecommunications*, vol. 68, no. 3, pp. 167–177, 2013. [Online]. Available: http://dx.doi.org/10.1007/s12243-012-0333-8

[97] G. Kamel, N. Wang, V. Vassilakis, Z. Sun, P. Navaratnam, C. Wang, L. Dong, and R. Tafazolli, "CAINE: A Context-Aware Information-Centric Network Ecosystem," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 176–183, August 2015. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2015.7180525

[98] "A Context-Adaptive Content Ecosystem Under Uncertainty (CONCERT)," 2014, accessed: 2016-07. [Online]. Available: http://www.concert-project.org/

[99] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *MultiMedia, IEEE*, vol. 18, no. 4, pp. 62–67, April 2011. [Online]. Available: http://dx.doi.org/10.1109/MMUL.2011.71

[100] S. Lederer, C. Müller, B. Rainer, C. Timmerer, and H. Hellwagner, "Adaptive Streaming over Content-Centric Networks in Mobile Networks Using Multiple Links," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC)*, 2013, pp. 677–681. [Online]. Available: http://dx.doi.org/10.1109/ICCW.2013.6649319

[101] Y. Liu, J. Geurts, J.-C. Point, S. Lederer, B. Rainer, C. Müller, C. Timmerer, and H. Hellwagner, "Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis," in *Proceedings of the IEEE International Conference on Communications*, 2013, pp. 3629–3633. [Online]. Available: http://dx.doi.org/10.1109/ICC.2013.6655116

[102] L. Fleischer, "Approximating Fractional Multicommodity Flow Independent of the Number of Commodities," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 24–31. [Online]. Available: http://dx.doi.org/10.1109/SFFCS.1999.814573

[103] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 Advanced Video Coding Standard and its Applications," *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–143, Aug 2006. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2006.1678121

[104] C. Kreuzberger, D. Posch, and H. Hellwagner, "A Scalable Video Coding Dataset and Toolchain for Dynamic Adaptive Streaming over HTTP," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15. New York, NY, USA: ACM, 2015, pp. 213–218. [Online]. Available: http://dx.doi.org/10.1145/2713168.2713193

[105] B. Rainer, S. Lederer, C. Müller, and C. Timmerer, "A Seamless Web Integration of Adaptive HTTP Streaming," in *Proceedings of the 20th European Signal Processing Conference*, Aug 2012, pp. 1519–1523.

[106] C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, May 2013, pp. 1318–1323.

[107] G. Rossini and D. Rossi, "Coupling Caching and Forwarding: Benefits, Analysis, and Implementation," in *Proceedings of the 1st International Conference on Information-Centric Networking*, ser. ICN '14. New York, NY, USA: ACM, 2014, pp. 127–136. [Online]. Available: http://dx.doi.org/10.1145/2660129.2660153

[108] D. Posch, B. Rainer, and H. Hellwagner, "SAF: Stochastic Adaptive Forwarding in Named Data Networking," in *arXiv.org*, 2015, pp. 1–10.

[109] N. Karmarkar, "A New Polynomial-time Algorithm for Linear Programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, Dec. 1984. [Online]. Available: http://dx.doi.org/10.1007/BF02579150

[110] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms.* MIT press, 2009.

[111] Z. Bojthe, L. Meszaros, B. Seregi, R. Hornig and A. Varga, "INET Framework," Accessed: 2016-07. [Online]. Available: http://inet.omnetpp.org/

[112] S. Lederer, C. Müller, and C. Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, ser. MMSys '12. ACM, 2012, pp. 89–94. [Online]. Available: http://dx.doi.org/10.1145/2155555.2155570

[113] C. Müller, S. Lederer, and C. Timmerer, "A Proxy Effect Analyis and Fair Adaptation Algorithm for Multiple Competing DASH Clients," in *Visual Communications and Image Processing (VCIP), 2012 IEEE*, Nov 2012, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/VCIP.2012.6410799

[114] R. Grandl, K. Su, and C. Westphal, "On the Interaction of Adaptive Video Streaming with Content-Centric Networking," in *20th International Packet Video Workshop*, 2013. [Online]. Available: http://dx.doi.org/10.1109/PV.2013.6691451

[115] R. Albert and A.-L. Barabási, "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics*, vol. 74, pp. 47–97, Jan 2002. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.74.47

[116] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," *SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 175–187, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1145/347057.347428

[117] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The Impact of Internet Policy and Topology on Delayed Routing Convergence," in *Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies*, 2001, pp. 537–546. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2001.916775

[118] S. Kini, S. Ramasubramanian, A. Kvalbein, and A. Hansen, "Fast Recovery from Dual Link Failures in IP Networks," in *Proceedings of the 28th IEEE INFOCOM*, April 2009, pp. 1368–1376. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2009.5062052

[119] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong, "VIP: A Framework for Joint Dynamic Forwarding and Caching in Named Data Networking," in *Proceedings of the 1st International Conference on Information-Centric Networking*. ACM, 2014, pp. 117–126. [Online]. Available: http://dx.doi.org/10.1145/2660129.2660151

[120] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, IETF, 2002. [Online]. Available: https://tools.ietf.org/html/rfc3272

[121] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN Simulator for NS-3," University of California, Los Angeles, Technical Report NDN-0005, 2012.

[122] A. Tanenbaum and D. Wetherall, *Computer Networks (Fifth Edition)*. Pearson, 2010.

[123] E. J. Rosensweig, J. Kurose, and D. Towsley, "Approximate Models for General Cache Networks," in *Proceedings of the 29th IEEE INFOCOM*, March 2010, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2010.5461936

[124] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnSim: An Highly Scalable CCN Simulator," in *Proceedings of IEEE International Conference on Communications*, June 2013, pp. 2309–2314. [Online]. Available: http://dx.doi.org/10.1109/ICC.2013.6654874

[125] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini, "INFORM: A Dynamic Interest Forwarding Mechanism for ICN," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, 2013, pp. 9–14. [Online]. Available: http://dx.doi.org/10.1145/2491224.2491227

[126] H. Qian, R. Ravindran, G.-Q. Wang, and D. Medhi, "Probability-based Adaptive Forwarding Strategy in Named Data Networking," in *IFIP/IEEE International Symposium on Integrated Network Management*, 2013, pp. 1094–1101.

[127] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg, "An On-demand Multi-Path Interest Forwarding Strategy for Content Retrievals in CCN," in *Proceedings of IEEE Network Operations and Management Symposium*, 2014, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/NOMS.2014.6838389

[128] M. Brokate and G. Kersting, *Maß und Integral*. Springer Basel AG, 2010. [Online]. Available: http://dx.doi.org/10.1007/978-3-0346-0646-2

[129] J. Elstrodt, *Maß-und Integrationstheorie*. Springer-Verlag Berlin, 2006.

[130] A. Medina, I. Matta, and J. Byers, "BRITE: A Flexible Generator of Internet Topologies," Boston University, Technical Report, 2000, Accessed: 2016-07. [Online]. Available: http://www.cs.bu.edu/brite/

[131] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-law Relationships of the Internet Topology," *SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 251–262, Aug. 1999. [Online]. Available: http://dx.doi.org/10.1145/316194.316229

[132] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Optimal Cache Allocation for Content-Centric Networking," in *Proceedings of the 21st International Conference on Network Protocols (ICNP)*. IEEE, Oct 2013, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1109/ICNP.2013.6733577

[133] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in Network Traffic: Findings and Implications," *SIGMETRICS Performance Evaluation Rev.*, vol. 37, no. 1, pp. 37–48, 2009. [Online]. Available: http://dx.doi.org/10.1145/2492101.1555355

[134] X. Cheng, C. Dale, and J. Liu, "Statistics and Social Network of YouTube Videos," in *Proceedings of the 16th Internationl Workshop on Quality of Service*, 2008, pp. 229–238. [Online]. Available: http://dx.doi.org/10.1109/IWQOS.2008.32

[135] K. Lei, J. Wang, and J. Yuan, "An Entropy-Based Probabilistic Forwarding Strategy in Named Data Networking," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 5665–5671. [Online]. Available: http://dx.doi.org/10.1109/ICC.2015.7249225

[136] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474 (Proposed Standard), Internet Engineering Task Force, Dec. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2474.txt

[137] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475 (Informational), Internet Engineering Task Force, Dec. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt

[138] L. Sun, I.-H. Mkwawa, E. Jammeh, and E. Ifeachor, *Guide to Voice and Video over IP*. Springer, 2013.

[139] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *ACM SIGCOMM Computer Comm. Review*, vol. 45, no. 4, pp. 325–338, Aug. 2015. [Online]. Available: http://dx.doi.org/10.1145/2829988.2787486

[140] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice over Content-Centric Networks," in *Proc. of the 2009 Workshop on Re-architecting the Internet*. ACM, 2009. [Online]. Available: http://dx.doi.org/10.1145/1658978.1658980

[141] International Telecommunication Union, "Pulse Code Modulation (PCM) of Voice Frequencies," *ITU-T Recommendation G.711*, 1990, Accessed: 2016-07. [Online]. Available: http://itu.int/rec/T-REC-G.711-198811-I

[142] ——, "The E-Model: A Computational Model for the Use in Transmission Planning," *ITU-T Recommendation G.107*, 2015, Accessed: 2016-07. [Online]. Available: http://www.itu.int/rec/T-REC-G.107-201506-I/en

[143] ——, “Transmission Impairments due to Speech Processing,” *ITU-T Recommendation G.113*, 2007, Accessed: 2016-07. [Online]. Available: http://www.itu.int/rec/T-REC-G.113-200711-I/en

[144] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, “A Survey of Information-centric Networking Research,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1024–1049, February 2014. [Online]. Available: http://dx.doi.org/10.1109/SURV.2013.070813.00063

[145] Named Data Networking Consortium, “The NDN Testbed,” Accessed: 2016-07. [Online]. Available: http://named-data.net/ndn-testbed/

[146] G. N. Prudy, *Linux iptables Pocket Reference*. O’Reilly Media, Inc., 2004.

[147] B. Hubert *et al.*, “Linux Advanced Routing and Traffic Control HOWTO v.1.0.1,” 2012, Accessed: 2016-07. [Online]. Available: http://www.lartc.org/

[148] S. Hemminger, “Network Emulation with NetEm,” in *Linux Conference Australia*, Apr. 2005, p. 9, Accessed: 2016-07. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.1687&rep=rep1&type=pdf

[149] M. S. Keller, “Take Command: Cron: Job Scheduler,” *Linux J.*, vol. 1999, no. 65es, Sep. 1999, Accessed: 2016-07. [Online]. Available: http://dl.acm.org/citation.cfm?id=327966.327981

[150] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format,” RFC 7159 (Proposed Standard), Internet Engineering Task Force, Mar. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7159.txt

[151] P. Erdős and A. Rényi, “On Random Graphs I,” *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[152] K. Matsuzono and H. Asaeda, “NMRTS: Content Name-based Mobile Real-time Streaming,” *IEEE Communications Magazine*, vol. 54, no. 8, pp. 92–98, August 2016. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2016.7537182

List of Abbreviations

ARPANET	Advanced Research Projects Agency Network
AS	Autonomous System
AVC	Advanced Video Coding
BGP	Border Gateway Protocol
BPI-R1	Banana Pi Router
CCE	Cache Everything Everywhere
CCN	Content-Centric Networking
CDN	Content Delivery Network
CI	Confidence Interval
CPU	Central Processing Unit
CS	Content Store
DAS	Dynamic Adaptive Streaming
DASH	Dynamic Adaptive Streaming over HTTP
EN	Emulation Network
FI	Future Internet
FIB	Forwarding Information Base
FIFO	First in – First Out
FTP	File Transfer Protocol
FWT	Forwarding Table
GPL	General Public License
HTB	Hierarchical Token Bucket

HTTP	Hypertext Transfer Protocol
ICN	Information-Centric Networking
ILP	Integer Linear Program
iNRR	ideal Nearest Replica Routing
IP	Internet Protocol
ISP	Internet Service Provider
JSON	JavaScript Object Notation
LAN	Local Area Network
LP	Linear Program
LRU	Least Recently Used
MCFP	Multi-Commodity Flow Problem
MN	Management Network
MOS	Mean Opinion Score
MPD	Media Presentation Description
MPTCP	Multipath Transmission Control Protocol
NACK	Negative Acknowledge
NCC	Interest Forwarding Strategy for PARC's Implementation of CCNx
NFD	Networking Forwarding Daemon
NFS	Network File System
NONCE	Unique Number / Random Bit Pattern
OMP-IF	On-demand Multi-Path Interest Forwarding
PC	Personal Computer
PIT	Pending Interest Table
PLC	Packet Loss Concealment

QoE	Quality of Experience
QoS	Quality of Service
RAM	Random-Access Memory
RFA	Request Forwarding Algorithm
RIB	Routing Information Base
RTCP	Real-Time Control Protocol
RTP	Real-Time Transport Protocol
RTT	Round Trip Time
SAF	Stochastic Adaptive Forwarding
SAF-CAA	Stochastic Adaptive Forwarding – Context-Aware Adaptation
SATA	Serial AT Attachment
SLA	Service Level Agreement
SNR	Signal-to-Noise Ratio
SSD	Solid-State-Drive
SVC	Scalable Video Coding
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TLC	Type-Length-Value
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
VoIP	Voice over IP
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	Extensible Markup Language