Sandra-Nadine Bergmann

Experimental Study of Multi-Path Transmission for Mobile Clients over NDN

Master Thesis

submitted in fulfilment of the requirements for the degree of
Diplom-Ingenieurin

Programme:
Master's programme Applied Informatics

Alpen-Adria-Universität Klagenfurt

Evaluator
Univ.-Prof. Dipl.-Ing. Dr. Hermann Hellwagner
Alpen-Adria-Universität Klagenfurt
Institut für Informationstechnologie

Pre-Evaluator
DI Dr. Philipp Günther Moll, B.Sc.B.Sc.
Alpen-Adria-Universität Klagenfurt
Institut für Informationstechnologie

Klagenfurt, April 2022

In case of comments or problems, please contact the Multimedia Communication Research Group (sek-hellwagner@itec.uni-klu.ac.at) or the author (sbergman@edu.aau.at).

What lies behind us and what lies before us are small matters compared to what lies within us.
Oliver Wendell Holmes

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 thesis, 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
- when passing on copies of the academic thesis (e.g. in bound, printed or digital form), I will ensure that each copy is fully consistent with the submitted digital version.

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

Sandra-Nadine Bergmann m.p. Klagenfurt, April 2022

Acknowledgements

It is a great feeling to complete my master thesis and finish my studies in Applied Informatics. Therefore, I will thank some people who have supported me a lot during my journey of writing my master thesis and also during my student time.

I would like to thank especially my advisor Univ.-Prof. Dipl.-Ing. Dr. Hermann Hellwagner for the possibility of writing my master thesis within the Multimedia Communication research group and for his great support during my work with this topic. He always supported me during the practical part of my master thesis and during writing this master thesis.

I would also like to thank Univ.-Ass. Dipl.-Ing. Philipp Moll a lot for his great support. He supported me during the practical part of my master thesis and always helped me to find good solutions as well.

Last but not least I would like to thank my family and friends. Without their help and support, it would not have been possible for me to finish my studies in Applied Informatics during my whole student time and during writing this master thesis.

Abstract

Today, the Internet uses the Internet Protocol (IP) as the network layer protocol. In the scope of this master thesis, a novel network layer protocol called Named Data Networking (NDN) is used where content is requested by its name. This is similar to when using domain names in IP. The novel network layer protocol NDN is used in this master thesis instead of the IP protocol within an NDN network. An NDN network can be simulated with the help of an NDN network simulator which is used for the practical part of this master thesis.

Different strategies are available on how packets can be sent in an NDN network when a content consumer requests data from a content producer. But not all strategies fit the current needs when mobile clients are using an NDN network. This master thesis aims to create a new strategy that works well for mobile clients in an NDN network with loss-prone wireless links. Therefore, a new strategy for sending packets is implemented. An evaluation scenario is created as well to show the performance evaluation of the new strategy. Furthermore, this new strategy is compared with already existing strategies with the help of the created evaluation scenario. The new strategy and the performance evaluation of different strategies are used within an NDN network simulator. Finally, the results from the performance evaluation are presented with the help of the simulation scenario using an NDN network.

Kurzfassung

Heutzutage wird im Internet das Internet-Protokoll (IP) als Netzwerkprotokoll genutzt. Im Rahmen dieser Masterarbeit wird ein Netzwerkprotokoll namens Named Data Networking (NDN) vorgestellt, bei dem Daten anhand vom Namen angefordert werden. Dies ähnelt der Verwendung von Domänennamen in IP. In dieser Masterarbeit wird anstelle des IP-Protokolls das neuartige Netzwerkprotokoll NDN innerhalb eines NDN-Netzwerkes verwendet. Ein NDN-Netzwerk kann mithilfe eines NDN-Netzwerksimulators simuliert werden, der für den praktischen Teil dieser Masterarbeit verwendet wird.

Es gibt verschiedene Strategien, wie Pakete in einem NDN-Netzwerk gesendet werden können, wenn ein Benutzer Daten von einem Server anfordert. Aber nicht alle Strategien passen gut zu den Anforderungen, wenn mobile Benutzer ein NDN-Netzwerk verwenden. Ziel dieser Masterarbeit ist es, eine neue Strategie zu entwickeln, die für mobile Benutzer in einem NDN-Netzwerk mit verlustbehafteten drahtlosen Verbindungen gut funktioniert. Daher wird eine neue Strategie zum Senden von Paketen implementiert. Ein Evaluierungsszenario wird ebenfalls erstellt, um die Performanz der neuen Strategie zu veranschaulichen. Des Weiteren wird diese neue Strategie mithilfe des erstellten Evaluierungsszenarios mit bereits existierenden Strategien verglichen. Die neue Strategie und die Evaluierung verschiedener Strategien werden innerhalb eines NDN-Netzwerksimulators verwendet. Abschließend werden die Ergebnisse der Evaluierung anhand von dem erstellten Simulationsszenario in einem NDN-Netzwerk vorgestellt.

Table of Contents

1 Introduction and Overview

In this master thesis, the network layer protocol *Named Data Networking* (NDN) [JSTP+09] is used for network simulations instead of the *Internet Protocol* (IP). NDN is a novel network layer protocol and has advantages over IP. When requesting content with IP, a content consumer has to know the location where requested data is stored. This means, the IP address of a content provider has to be known. When requesting content with NDN, then this content can be retrieved by the name of the content without knowing the location where the data is stored. Data in NDN has always a unique name in the whole network. In a simplified example, a packet for requesting data is sent out over NDN. This packet finds a way to the data in the network, by sending this packet from one network node to another network node, until the requested data is found. For that, stored information in tables on each node is used. Sending a packet from a content consumer to a content provider on a route in the network is called forwarding. Different strategies can be used to forward packets in a network. When the data for a content request is found somewhere in the network, then this data is sent out to the content consumer by using the reverse path from the node, where the data has been found, to the content consumer. The same path has already been used by the requesting packet.

The aim of this master thesis is designing and implementig a new forwarding strategy, which works well for mobile clients in a network. NDN is used in the network as network layer protocol, instead of using IP. Furthermore, a simulation scenario is created for the performance evaluation of the practical part of this master thesis.

This master thesis is structured as follows. Chapter 2 describes technologies and background information this master thesis is based on. At first, the network layer protocol NDN is described in detail. After this, the communication model of NDN, packet forwarding concepts of NDN, and finally, considerations regarding mobile clients in NDN are discussed.

Chapter 3 explains the aim of this master thesis with the research questions. Firstly, multi-path transmission and the effect of loss-prone wireless links and mobile clients in an NDN network are examined. Then, the research questions of this master thesis are introduced. Solutions for solving the research questions are proposed and an idea for the implementation is presented.

Chapter 4 describes the implementation of the idea. The implementation has been performed with a network simulator named *ndnSIM* [MAMZ16]. The implementation part includes the implementation of a new forwarding strategy for mobile clients, named *Mobile-Client Forwarding Strategy*. Moreover, extensions of ndnSIM components are implemented as well. These extensions are required for the implementation of the forwarding strategy, and to simulate an evaluation scenario for the performance evaluation of the newly implemented forwarding strategy.

Chapter 5 presents the performance evaluation of the Mobile-Client Forwarding Strategy. The evaluation scenario consists of a randomly generated NDN network, where mobile client nodes are listening to voicemail messages in an NDN network. The evaluation scenario and the used metrics for the performance evaluation are described in this section in detail. After this, the results of the performance evaluation are presented. At first, the results of the Mobile-Client Forwarding Strategy with different parameters are presented. These different parameters influence the forwarding decision. Furthermore, the results are evaluated, to get a preferred configuration for the newly implemented forwarding strategy. After this, the results of the Mobile-Client Forwarding Strategy with predefined parameters are compared to other forwarding strategies. This is done to show that the Mobile-Client Forwarding Strategy performs well with mobile client nodes in an NDN network, in comparison with already existing forwarding strategies.

Chapter 6 concludes with a summary and final remarks.

2 Background: Named Data Networking (NDN)

This chapter describes the used technologies and relevant background information, which are used while working on this master thesis. First, the novel network layer protocol Named Data Networking (NDN) [JSTP+09] is introduced, where the general concept and advantages over other network layer protocols, like the Internet Protocol (IP), are discussed. The communication model and packet types for communication in NDN are described in detail. Forwarding and forwarding strategies are explained, to show how packets can be forwarded in an NDN network. However, the impact of wireless networks and mobile clients on the performance of NDN is discussed. Since wireless links can be loss-prone, packet loss in NDN networks is described in detail. Finally, loss models and mobility models are explained, to simulate communication with mobile clients on loss-prone wireless links in an NDN network.

2.1 Introduction

In this section, the network layer protocol Named Data Networking (NDN) [JSTP+09] is introduced, to get an overview of this approach, which is the foundation of this master thesis and used for network simulations in the practical part of this master thesis. With NDN, the commonly known network principles are shifted from a host-centric to a data-centric paradigm.

Today, the Internet Protocol (IP) is used as the network layer protocol in the Internet. Using IP as a network layer protocol is a host-centric approach. In a simple example, a client - the content consumer - wants to retrieve a certain file - the content. In IP, the content is requested from a designated server - the content provider - and can only be requested if the IP address of this server is known. This means, before requesting content with IP, the user has to know the location where the preferred data is stored. So, data can be requested only by knowing the IP address of the content provider. Generally, the user does not know the exact IP address but a domain name or a name of a website when using search engines. An example of a domain name is *www.aau.at*. The Domain Name System (DNS) is used to translate a domain name to an IP address. Without knowing the exact IP address of specific content, data cannot be requested by a content consumer. Using IP as a network layer protocol also means that content will always be requested and delivered by a content provider. It is not possible that another node in the network can reply with data to the requested content.

Advantages of NDN compared to IP:

- Content can be addressed by its name, without knowing the location in the network where the content is stored.
- Content does not have to be stored only on a server of a content provider, but it can also be stored in caches of nodes within the NDN network.

Named Data Networking (NDN) is a novel network layer protocol and can be used instead of IP. Using NDN as a network layer protocol is a content-centric approach, improving on the host-centric approach known from IP. In NDN, the location of the content provider with the content does not have to be known, instead, content is addressed by its name. Names in NDN are structured hierarchically similar to the use of domain names. An example of a hierarchically structured content name can be */aau.at/user-a/voicemail/s0*. With the NDN approach, only the name of the requested content has to be known when requesting content. This means, the concept of NDN communication is that named data is requested instead of addressing hosts to request content.

When data is requested by a consumer using IP, only the content provider of the requested data can reply with the data. When data is sent from a content provider to a content consumer in NDN, this data can be stored in caches of nodes along the transmission path from the content provider to the content consumer. As a result, another benefit is that when the same data is requested again from the same consumer, or even from another consumer, then not only the content provider can answer with the data, but also a router node on the request path within the NDN network, when the data has been found in the cache of this node. This means, it is not necessary that the request has to be sent to the content provider when the data for the request will be found before reaching the content provider. Then another node in the NDN network can send the data to the content consumer.

In future, NDN might be a promising addition to IP in today's Internet, due to the previously mentioned advantages of NDN over IP as a network layer protocol.

2.2 Communication in NDN

In the following, requesting data in NDN is introduced. Different packet types are used for communication in NDN. The types of packets and the communication model in NDN networks are introduced in the following.

NDN uses two packet types for communication. *Interest Packets* are used to request data of preferred content, and *Data Packets* are the response to Interest Packets and deliver requested content. The types of packets are shown in figure 2.1 [JSTP+09]. A content consumer requests data in the form of Interest Packets, where each Interest Packet requests content by name. One Data Packet follows after one Interest Packet, by delivering the requested data. Data Packets are always transferred on the reverse path of Interest Packets. Both packet types contain some fields. In the following, the fields of Interest Packets and Data Packets are described.

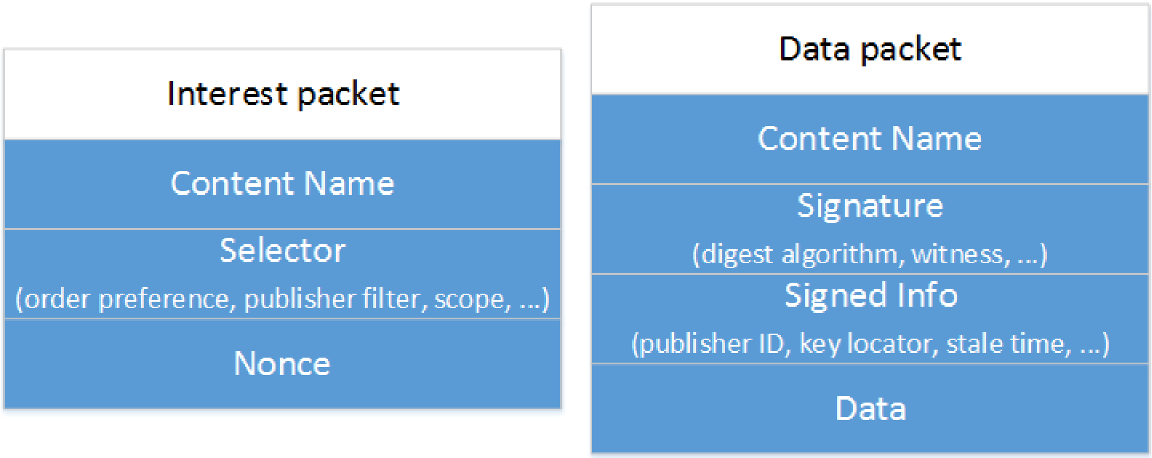

Fig. 2.1: Packet types for communication in NDN: There are two types of packets, Interest Packets which request the content, and Data Packets which answer to an Interest Packet by delivering the content (adopted from [JSTP$^+$09]).

2.2.1 Interest Packets

Interest Packets request data and contain the following fields, which are shown in figure 2.1:

- Content Name
- Selector
- Nonce

The *Content Name* represents the name of the data. Names in NDN are structured hierarchically and are used to identify and request content in NDN. In contrast to IP, the location of a content provider does not have to be known. As a result, to decide how to forward an Interest Packet with a specific *Content Name*, the longest prefix match on the *Content Name* of entries in a forwarding table is made. Therefore, each node in an NDN network holds a table with forwarding information. How forwarding an Interest Packet from a content consumer to a content provider works in detail in NDN, is described in section 2.3. An example of a *Content Name* can be */aau.at/user-a/voicemail/s0* or */aau.at/user-a/voicemail/s1*.

The *Selector* is an additional field of the Interest Packet. When two Interest Packets have the same *Content Name* and the same *Selector*, then these two Interest Packets request the same content.

The *Nonce* value is a further field of an Interest Packet. When an Interest Packet gets retransmitted, then the same content will be sent again. In this case, the initial Interest

Packet and the retransmitted Interest Packet have the same *Content Name* and the same *Selector*, but a different *Nonce* value. The *Nonce* value is a random value and with the help of this value, each Interest Packet gets unique. A more thorough description of NDN's packet types can be found in [JSTP+09] and in the NFD Developer's Guide [ASZZ+16].

2.2.2 Data Packets

A Data Packet answers to an Interest Packet with the requested data. Each Data Packet has the following fields, which are also shown in figure 2.1:

- Content Name
- Signature
- Signed Info
- Data

The *Content Name* of a Data Packet, which follows after an Interest Packet, is the same as the *Content Name* of the corresponding Interest Packet. This means, when an Interest Packet with the *Content Name /aau.at/user-a/voicemail/s0* requests data, then the Data Packet, which answers to the Interest Packet, has the same *Content Name /aau.at/user-a/voicemail/s0*. The *Content Name* uniquely identifies the content in the *Data* field of the packet.

Data Packets are signed for security reasons. Therefore, the fields *Signature* and *Signed Info* are used for the signature of the Data Packet. The most important result of signing the Data Packet is that the corresponding data remains immutable. Finally, the *Data* field contains the requested content.

A more thorough description of NDN's packet types can be found in [JSTP+09] and in the NFD Developer's Guide [ASZZ+16]. How communication in NDN works with Interest Packets and Data Packets is described in the following.

2.2.3 Data Communication Example

An example of how data can be requested by a user - the content consumer - in NDN is shown in figure 2.2, to explain how communication in NDN works.

In this example, there are two user nodes - *User A* and *User B* -, one server node *Server* - the content provider -, and an NDN network consisting of four router nodes - *Router 1*, *Router 2*, *Router 3* and *Router 4*. All nodes in this example use NDN as a network layer protocol.

At first, *User A* requests content by sending Interest Packets containing the name of the preferred data. Each Interest Packet is forwarded in the network until the data for the corresponding Interest Packet is found. For instance, on a server of a content provider that offers the data or in a cache of a node in the NDN network. Each node holds a cache and each node has several interfaces. An interface or face on a node represents a connection to another node in the network. Each node chooses a face or multiple faces, on which to forward incoming Interest Packets, depending on the installed forwarding strategy and the forwarding table stored on the current node. In section 2.3, the cache on a node, forwarding in NDN, forwarding strategies, and the forwarding table are described in more detail.

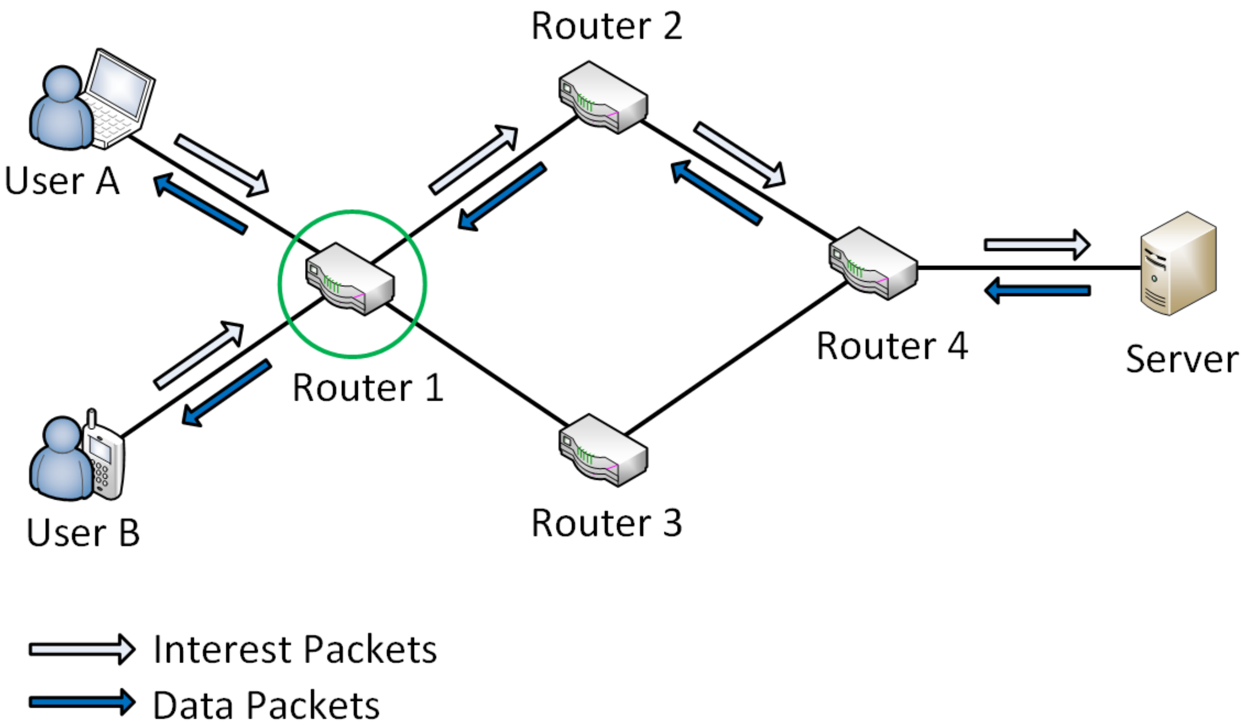

Fig. 2.2: A communication example in NDN: User A and User B - the content consumers - request the same content, by issuing Interest Packets containing the name of the requested content. The Server - the content provider - or a router node in the network answers with the corresponding Data Packets to the Interest Packets.

In this example, Interest Packets are forwarded from *User A* to *Router 1*, from *Router 1* to *Router 2*, from *Router 2* to *Router 4* and from *Router 4* to the *Server*. The *Server* replies with Data Packets to corresponding Interest Packets. Data Packets are sent on the reverse path of Interest Packets through the network until *User A* receives these Data Packets with the content. On the reverse path, the data can be stored in the caches of the router nodes, like in the caches of *Router 4*, *Router 2* and *Router 1*.

When *User B* requests the same content by sending an Interest Packet after *User A* has received the data, the Interest Packet is also forwarded through the network until the data for the corresponding Interest Packet has been found. In this case, the data can be found in the cache of *Router 1*, because the data has been stored in the cache, while delivering the data from the *Server* to *User A*. As a result, the Interest Packet does not have to be forwarded through the whole network to the *Server*, but can be satisfied from the cache of *Router 1*. This means, the Interest Packet is only forwarded from *User B* to *Router 1* and *Router 1* replies to the Interest Packet with the corresponding Data Packet.

In NDN, it is sufficient to forward Interest Packets until the data is found at a node somewhere in the NDN network. The signature field in the Data Packet ensures that the content has not been modified during storing the content in a cache on a node in the network. How forwarding in NDN works for Interest Packets is described in the following.

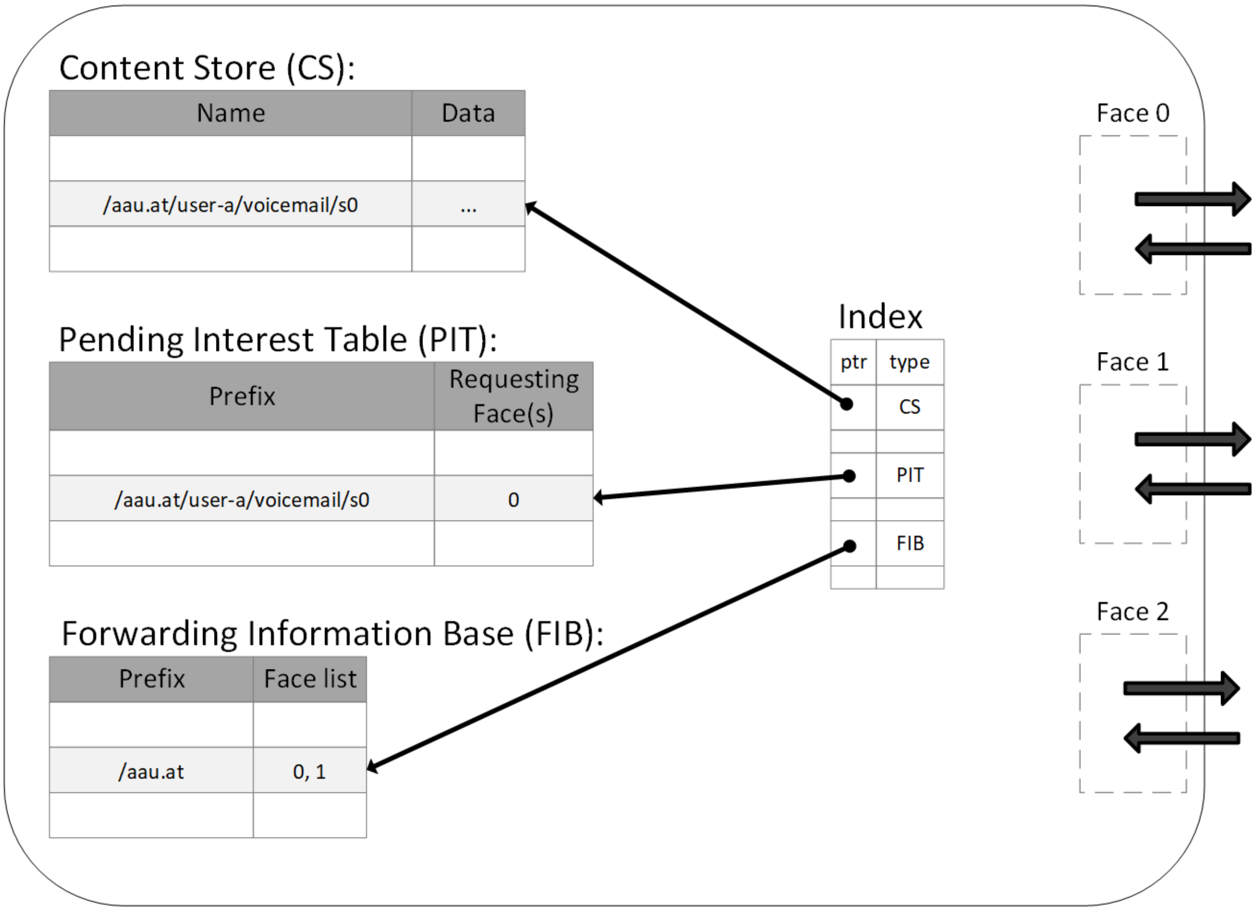

Fig. 2.3: Forwarding in NDN: An NDN node has several available faces where packets can be sent and received. Interest Packets can be forwarded to one or multiple faces. Each node maintains three tables, the Content Store, the Pending Interest Table, and the Forwarding Information Base (adopted from [JSTP+09]).

2.3 Forwarding in NDN

In the following, forwarding of Interest Packets in NDN is described. Figure 2.3 gives a short overview of the structure of NDN nodes [JSTP+09]. In NDN, each node, which receives an Interest Packet, decides where to send the Interest Packet next. Therefore, NDN nodes maintain three tables and a list of available interfaces, where packets can be forwarded and received. An interface or face on a node represents a connection to another node in the network. The three tables are the *Content Store* (CS), the *Pending Interest Table* (PIT), and the *Forwarding Information Base* (FIB).

At first, an overview of the three tables is given. After this, it is described what happens when an Interest Packet arrives on an NDN node before the Interest Packet will be forwarded. Furthermore, forwarding strategies are introduced, which can be installed on NDN nodes. The described forwarding strategies in this section are used for the performance evaluation of the practical part of this master thesis, during using an NDN network simulator, called ndnSIM [LosA17]. NdnSIM is an NS-3 [Foun17] based NDN simulator to perform network simulations within an NDN network. These forwarding strategies are available in ndnSIM through the integration of Named Data Networking Forwarding Daemon (NFD) [ASZZ+16]. Finally, it is described how NDN can be extended with a new forwarding strategy, when the behavior of the available forwarding strategies does not fit the current needs. In the scope of this master thesis, a new forwarding strategy, called Mobile-Client Forwarding Strategy, is implemented, which is described in section 4.3 in more detail.

2.3.1 Content Store

At first, the Content Store is a cache, which is located on every network node in an NDN network and where data can be stored for an unspecified time, when caching is enabled on the node in an NDN network. The structure of this cache is shown in figure 2.3. The following fields are stored in a Content Store:

- Name
- Data

The Content Store saves a tuple, where the first column *Name* is the name of the content, which can be found in Interest Packets and Data Packets within the field *Content Name*. Furthermore, the second column *Data* contains the stored content data in the cache, which represents the data from the *Data* field of a Data Packet. An example of a content name can be */aau.at/user-a/voicemail/s0*. When a Data Packet is received on an NDN node, then this node can choose to store the content data of this Data Packet in the Content Store of the current node, when caching data is enabled on this node.

2.3.2 Pending Interest Table

Furthermore, the Pending Interest Table stores previously arrived Interest Packets, which have already been forwarded to an outgoing face. The entries in the Pending Interest Table represent Interest Packets, which are waiting for the arrival of the corresponding data. The following fields are stored in a Pending Interest Table:

- Prefix

- Requesting Face(s)

The first column in the table is the *Prefix*, which represents the name of the content, which can be found in Interest Packets and Data Packets within the field *Content Name*. Furthermore, the second column *Requesting Face(s)* is a list of face IDs of an Interest Packet. This list contains the face IDs of all faces where an Interest Packet with a specific content name comes from. One Pending Interest Table entry can contain more than one face ID in the *Requesting Face(s)* list, because the same content can be requested by different users, which are then waiting for the arrival of this requested data. After receiving the requested data, the node knows where to send back the Data Packets to corresponding Interest Packets, because the requesting faces are stored in a Pending Interest Table entry. When a Data Packet has been received on this node, then the Data Packet will be sent out to all requesting faces, and the Pending Interest Table entry will be deleted after sending the Data Packet. An example of a Pending Interest Table entry is shown in figure 2.3. This example shows, when a Data Packet with content name */aau.at/user-a/voicemail/s0* arrives on the current node, then this Data Packet will be sent to Face 0.

2.3.3 Forwarding Information Base

In NDN, the forwarding information is stored in a table called Forwarding Information Base. With the help of this table and the installed forwarding strategy, the decision where to forward an incoming Interest Packet takes place. Interest Packets can be forwarded to one or multiple available faces. The following fields are stored in a Forwarding Information Base:

- Prefix
- Face list

The Forwarding Information Base consists of entries with the columns *Prefix* and *Face list*. Content names are structured hierarchically in NDN. This means, the Forwarding Information Base does not store the whole content name, but a prefix of the name. Incoming Interest Packets, where the content name starts with a specific *Prefix*, can be forwarded to one or multiple faces from the *Face list*. The structure of the Forwarding Information Base is shown in figure 2.3, as well as an example of a Forwarding Information Base entry. This example shows, that a prefix */aau.at/* is stored in a Forwarding Information Base entry. This means, when an Interest Packet with content name */aau.at/user-a/voicemail/s0* arrives on the current node, then this Interest Packet can be forwarded to Face 0 and Face 1.

2.3.4 Forwarding

In NDN, Interest Packets are forwarded through the network until the data for the corresponding Interest Packet has been found. Then, a Data Packet is replied on the reverse path of the Interest Packet. Therefore, it is described in the following, how forwarding of an Interest Packet works in an NDN network.

When a node in the network receives an incoming Interest Packet, the longest prefix match on the content name is made. At first, the node checks if the data for the Interest Packet is already stored in the local cache, called Content Store. If the data is found in the Content Store, then this data is sent out as Data Packet to the consumer, and the Interest Packet is discarded.

If the data is not found in the Content Store, then the longest prefix match with the entries of the Pending Interest Table is made. The Pending Interest Table stores incoming Interest Packets, which have already been forwarded to an outgoing face, and where the corresponding Data Packets to the Interest Packets are not yet received. If the content name of the Interest Packet is already in the Pending Interest Table, and the face of the incoming Interest Packet is not stored in the list of faces, then the face will be added to the list of faces and the Interest Packet can be discarded. If the face is already stored in the list of faces for the requested content name, then the Interest Packet can be discarded.

When no Pending Interest Table entry is found for the content name, then the longest prefix match with the entries in the Forwarding Information Base is made. The node has to decide how to forward the Interest Packet. The Interest Packet can be forwarded to one or multiple faces from the face list of a found prefix of the content name from the Interest Packet in the Forwarding Information Base. The face list does not contain the requesting face of the Interest Packet. After forwarding the Interest Packet, a new entry in the Pending Interest Table with the content name and the requesting face is created for this Interest Packet.

The forwarding decision, to which faces the Interest Packet should be forwarded, depends on the installed forwarding strategy on the current node. On each node, a predefined forwarding strategy can be chosen and installed. Several nodes can use different forwarding strategies. An overview of the available forwarding strategies can be found in the NFD Developer's Guide [ASZZ+16], which can be used when using NDN.

This master thesis deals with developing a new forwarding strategy in an NDN network, and evaluating the performance of the newly implemented forwarding strategy compared to already existing forwarding strategies. Therefore, two forwarding strategies are chosen from the list of available forwarding strategies, which can be used when using NDN. In the following, the chosen forwarding strategies are presented, which will be used for network simulation in the practical part of this master thesis.

2.3.5 Best-Route Forwarding Strategy

The *Best-Route Forwarding Strategy* is a default forwarding strategy, which can be used when using NDN. To use the Best-Route Forwarding Strategy, it has to be installed on preferred NDN nodes.

When an Interest Packet arrives on a node, then the longest prefix match with the content name of the Interest Packet on the Forwarding Information Base entries is made. When a matching entry is found, then the forwarding strategy chooses from the found entry one of the available faces from the face list to forward the Interest Packet. Different routes can be used between a content consumer and a content provider depending on the chosen face. The Best-Route Forwarding Strategy selects always a face from the available faces, where the selected route from a content consumer to a content provider has the lowest amount of costs. For example, when two routes exist from a content consumer to a content provider, then the route with the lowest amount of costs can be the route that needs fewer network hops. The selected face is called the best face. This forwarding strategy forwards Interest Packets without redundancy, because each Interest Packet is forwarded only to one face, which is the best face, where the route has the lowest amount of costs.

2.3.6 Multicast Forwarding Strategy

The *Multicast Forwarding Strategy* is another available forwarding strategy, which can be used when using NDN. To use the Multicast Forwarding Strategy, it has to be installed on preferred NDN nodes as well. The Multicast Forwarding Strategy forwards the incoming Interest Packets to all available faces for a matching prefix to the content name of the Interest Packet in the Forwarding Information Base. This means, all incoming Interest Packets are forwarded to multiple links redundantly with the Multicast Forwarding Strategy.

2.3.7 Developing a New Forwarding Strategy

When the behavior of the available forwarding strategies does not fit the current needs, then a new forwarding strategy can be created for NDN. Therefore, ndnSIM can also be extended with new forwarding strategies which is described in the ndnSIM documentation [LosA17]. In the scope of this master thesis, a new forwarding strategy, called Mobile-Client Forwarding Strategy, has been developed, which is described in section 4.3 in more detail.

A new forwarding strategy has to extend the *nfd::fw::Strategy* class. Furthermore, some of the following virtual methods can be overridden:

- afterReceiveInterest
- beforeSatisfyInterest
- beforeExpirePendingInterest

The method *afterReceiveInterest* is a pure virtual method. So, this method has to be overridden at least by the new forwarding strategy. This method is triggered, when the data for an incoming Interest Packet is not found in the Content Store and the Pending Interest Table shows that this Interest Packet can be forwarded. The forwarding decision is implemented in this method. The forwarding strategy can select one or multiple faces where the Interest Packet can be forwarded.

The method *beforeSatisfyInterest* is triggered before the Data Packet to a corresponding Interest Packet arrives. This means, before an Interest Packet of a Pending Interest Table entry gets satisfied through the corresponding Data Packet.

Finally, the method *beforeExpirePendingInterest* is triggered before the timer of a pending Interest Packet which waits for the corresponding Data Packet times out.

This master thesis deals with developing a new forwarding strategy for mobile clients in an NDN network, and evaluating the performance of the newly implemented forwarding strategy compared to the forwarding strategies Best-Route and Multicast. An evaluation scenario has been implemented for the performance evaluation as well, where WiFi networks are added to an NDN network, to simulate a scenario, where mobile client nodes are listening to voicemail messages in NDN. In the following, considerations of mobile clients in an NDN network will be made.

2.4 Mobile Clients in NDN

Mobile clients can connect to one of the multiple available wireless networks in a nearby environment, like WiFi networks with access points as well as cellular wireless networks like 4G or 5G with base transceiver stations from mobile network providers. Figure 2.4 shows a user, *User A*, who communicates with another user, *User B*. *User A* is near an access point of a WiFi network and a base transceiver station of a cellular wireless network. Wireless links from the mobile client to an access point or a base transceiver station can be loss-prone. This can happen, when a mobile client moves away from the connected access point or base transceiver station and consequently, the connection of the wireless link gets worse, which can lead to a higher packet loss.

In figure 2.4, the red link from *User A* to a WiFi access point represents a loss-prone wireless link. Wireless links can be loss-prone, when, for instance, a user is too far away from an access point or a base transceiver station, so that the connection gets worse due to packet loss. An advantage of the availability of multiple wireless networks is that mobile users can switch to another wireless network when the current wireless link from the connected wireless network gets worse. Therefore, when *User A* is connected to the loss-prone wireless link of the WiFi network, which is shown in figure 2.4, then *User A* can switch to another available wireless network, for instance, a 4G cellular wireless network.

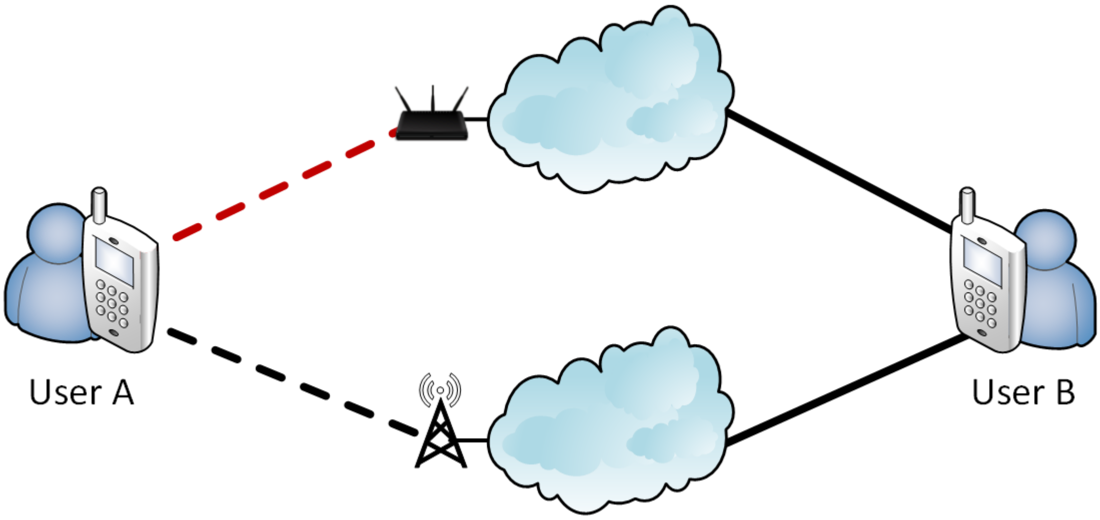

Fig. 2.4: Wireless links can be loss-prone: Wireless links from mobile clients to access points (WiFi networks) as well as wireless links from mobile clients to base transceiver stations (4G, 5G networks) can be available. These wireless links can be loss-prone.

Within the practical part of this master thesis, a network is created with ndnSIM, which uses NDN as a network layer protocol. Furthermore, network simulations are performed within this network, to compare different forwarding strategies. At first, a network with randomly generated network nodes is created for the evaluation scenario. Other nodes are additionally added to this network as well. A predefined number of WiFi networks with a specific WiFi standard, and a predefined number of access points for each WiFi network, as well as a predefined number of mobile clients, are added to the network. These mobile clients can connect to available WiFi networks. Other clients are randomly added to the randomly generated network, which are fixed clients instead of mobile clients. These fixed clients represent the communication partners for mobile clients. Furthermore, to simulate loss-prone wireless links, one of the available propagation loss models in ndnSIM can be chosen for wireless links. Finally, to simulate the movement of mobile clients, one of the available mobility models in ndnSIM can be used as well. With the use of a mobility model, mobile client nodes can move in a predefined area. The movement of a mobile client node is defined by the chosen mobility model.

In the following, packet loss in NDN is described in detail. After this, a loss model for wireless links, which is used for the evaluation scenario, is introduced as well as used mobility models for mobile client nodes. In this master thesis, an evaluation scenario with WiFi networks and mobile clients is created, where mobile clients are listening to voicemail messages, which have been created by other clients in the randomly generated network. For this evaluation scenario, only the content consumers, which are listening to voicemail messages, are mobile clients. The clients, which produce the voicemail messages for the mobile clients, are fixed nodes in the network. This is because producer mobility is hard to realize in an NDN network. Finally, the problem of producer mobility is described in detail.

2.4.1 Loss-Prone Links and Packet Loss

Wireless links can be loss-prone. In the following, packet transmission on loss-prone links in an NDN network is introduced.

Three cases can occur, when transmitting Interest Packets and Data Packets on loss-prone links between content consumer and content provider in NDN. These cases are introduced in figure 2.5.

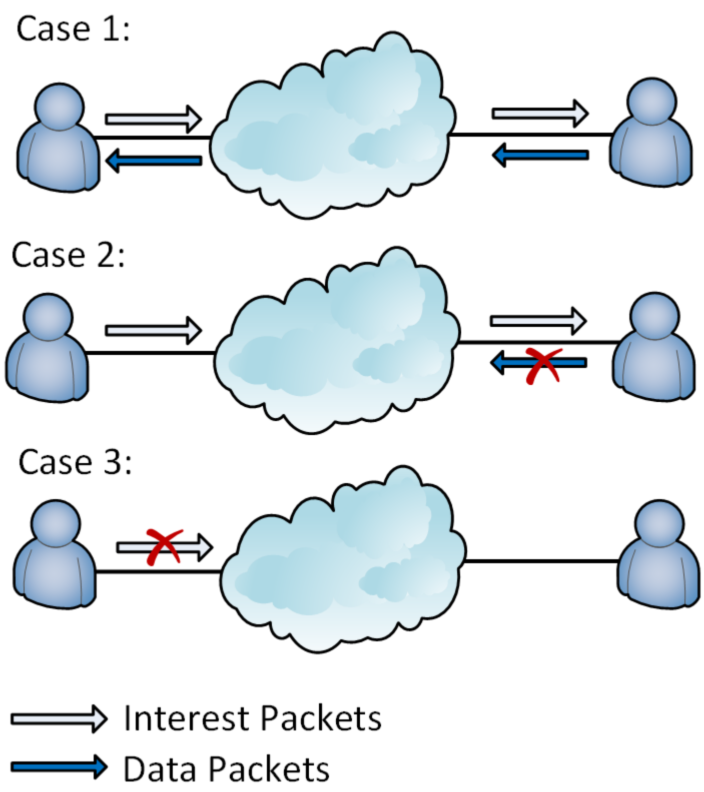

Fig. 2.5: Three cases of packet transmission on loss-prone links in NDN: In the first case the Interest Packet and the Data Packet are received. In the second case, only the Interest Packet is received and the corresponding Data Packet gets lost. Finally, in the third case, the Interest Packet gets lost and therefore, the Data Packet cannot be received as well.

The first case of transmitting packets on loss-prone links is the positive case. A consumer requests data by sending an Interest Packet and this packet is received at the content provider or any other node, where the data is stored. The content provider answers with a Data Packet to the corresponding Interest Packet and the Data Packet is also received at the content consumer.

In the second case, a consumer requests data by sending an Interest Packet and this Interest Packet is received at the content provider. The content provider answers with the corresponding Data Packet to the Interest Packet, but this Data Packet gets lost on the way to the content consumer.

Finally, the third case is that an Interest Packet is sent by a consumer, and this packet gets lost on the way to the content provider. This means, the Interest Packet is not received and the corresponding Data Packet to the Interest Packet cannot be received as well.

Generally, packet loss on loss-prone links does not occur randomly, but it follows a certain pattern. To simulate packet loss on loss-prone links, various loss models can be applied for links in an NDN network [Foun17] [StRi12]. In the following, loss models are introduced and the Log Distance Propagation Loss Model is presented, which is used for wireless links in the evaluation scenario of this master thesis.

2.4.2 Log Distance Propagation Loss Model

A number of propagation loss models are available, which can be used to model packet loss on network links. A list of available loss models for ndnSIM can be found on the NS-3 site [Foun17]. For the evaluation scenario of this master thesis, the *Log Distance Propagation Loss Model* has been applied for all wireless links, to simulate loss-prone wireless links.

With the Log Distance Propagation Loss Model, the reception power is calculated [Foun17] and the formula for the Log Distance Propagation Loss Model is shown in equation 2.1. This formula can be found on the NS-3 site [Foun17].

$$L = L_0 + 10nlog_{10}(\frac{d}{d_0}) \tag{2.1}$$

The following variables are used within the formula of the Log Distance Propagation Loss Model, where:

- **n:** is the path loss distance exponent
- $\mathbf{d_0}$**:** is the reference distance in meter (m)
- $\mathbf{L_0}$**:** is the path loss at reference distance in decibel (dB)
- **d:** is the distance in meter (m)
- **L:** is the path loss in decibel (dB)

The implementation of creating loss-prone wireless links is described in section 4.4.3 in more detail. In section 5.1.2, the used parameters for this loss model are shown as well.

Not only loss models can affect the loss rate, but also the mobility of mobile clients. When mobile clients move away from a WiFi access point or a base transceiver station of a cellular network, then the connection to the network can get worse. In the following, mobility models are introduced, which are also used for the practical part of this master thesis.

2.4.3 Mobility Models

A huge number of mobility models are available, which can be applied on mobile client nodes, to simulate a movement, and to specify movement patterns for these clients. A list of available mobility models for ndnSIM can also be found on the NS-3 site [Foun17].

For the practical part of this master thesis, the *Random Waypoint Mobility Model* and the *Gauss-Markov Mobility Model* have been implemented. The Gauss-Markov Mobility Model is used for the evaluation scenario. For each mobility model, a predefined movement area can be defined. Within this area, mobile clients can move. In the following, these mobility models are introduced.

The Random Waypoint Mobility Model [Foun17] [CaBD02] is a mobility model where a mobile client chooses a random point out of a predefined movement area, walks to this point with a random speed, and pauses for a random time after reaching the target point. After pausing, the next point is chosen out of the movement area randomly and the mobile client starts moving again to the position of the next target point at a random speed.

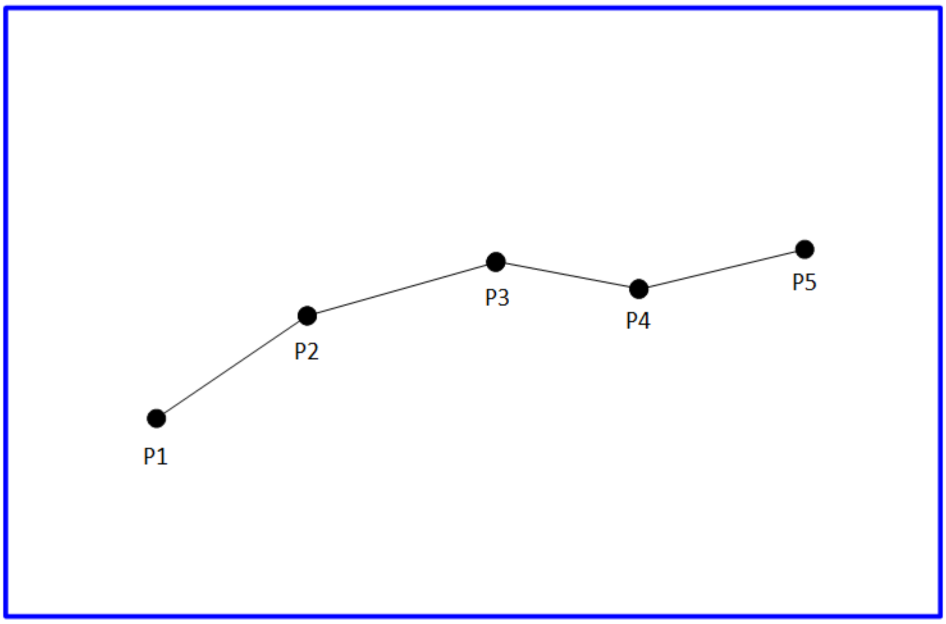

Fig. 2.6: Mobility model movement: The blue rectangle shows a movement area of a mobile client, used by a mobility model. P1, P2, P3, P4 and P5 show positions of a mobile client, where a mobile client starts at position P1 and moves to position P5.

Figure 2.6 shows in a simple example how mobile clients can move when using a mobility model. The blue rectangle shows the movement area of a mobile client, used by a mobility model. P1, P2, P3, P4 and P5 are positions of a mobile client within the movement area, where a mobile client starts at position P1 and moves to position P5.

The Random Waypoint Mobility Model movement pattern simulates how mobile clients might move. The movement of this mobility model is realistic and a good choice in comparison to other mobility models. For instance, with the Random Direction Mobility

Model, mobile clients always move until the movement-boundary is reached. In the real world it is not realistic that mobile clients only change their direction when a wall or a border is reached.

Another more realistic mobility model is the Gauss-Markov Mobility Model [Foun17] [CaBD02]. This is a mobility model where a mobile client walks in a random direction with a random speed for a predefined time as well. After this time, the client changes the direction and the speed, depending on a specific parameter alpha which indicates how random new values for direction and speed are chosen. The parameter alpha is a value between zero and one. When alpha is zero, then the values are chosen completely random. Furthermore, when alpha is one, then the values consider previous direction and speed values.

An advantage of the Gauss-Markov Mobility Model is that mobile clients move without sharp turns or sudden stops. Depending on a defined parameter alpha, the Gauss-Markov Mobility Model is not completely random, which is another advantage. Previous direction and speed values are considered as well. Therefore, it is a realistic mobility model for mobile clients. Only when the parameter alpha is set to zero, then the calculation of the new values for the direction and speed is completely random. When alpha is a value between 0 and 1 - for example 0.3 - then the calculation of the new direction, speed, and pitch angle values are not completely random and the calculation of the new values considers previous direction, speed, and pitch angle values as well.

Finally, producer mobility is introduced. Producer mobility means that a content producer, which provides content, is a mobile client. Producer mobility has disadvantages in NDN networks. These findings are shown in the following section.

2.4.4 Producer Mobility

In the evaluation scenario of this master thesis, content consumers are mobile clients and content producers are fixed clients. Consumers request voicemail messages in NDN and these voicemail messages have been created by content producers, which are fixed clients in the NDN network.

When content producers will also be mobile clients instead of fixed clients, then the following problem is present. When mobile clients as consumers request voicemail messages with Interest Packets, then it is difficult to choose a route based on the Forwarding Information Base, if content producers were mobile clients as well. The Forwarding Information Base in NDN has no information about the actual positions of content producers, when they are mobile clients. To forward Interest Packets to content producers, it is not easy to choose the right wireless link to the content producer, when multiple wireless networks are available. The forwarding engine has no information, to which wireless network the content producer is connected.

In figure 2.7, an example of the producer mobility problem with wireless NDN networks is shown. *User A* and *User B* are mobile clients. *Access Point 1* (AP 1) is the access point of the WiFi network where the content consumer is connected to. *Access Point 2* (AP 2) and *Access Point 3* (AP 3) are access points of different WiFi networks in the environment of *User B*. *User B* is near *Access Point 2*.

User A requests data from *User B* by sending Interest Packets. These Interest Packets are forwarded through the network. Each network node has to decide where to forward an Interest Packet. The Forwarding Information Base provides information about routes to specific content name prefixes for data. The route information for the data which is hosted by *User B* is also stored in the Forwarding Information Base. Network nodes only see the available routes, but not, which route is the best one or which route contains a wireless link. The Forwarding Information Base has also no information about the wireless network, to which *User B* is connected to at a specific time or at the current time. Using the route over *Access Point 3* would not be a good decision, because *User B* is far away from *Access Point 3* and *User B* is currently connected with *Access Point 2*. But the nodes within the network have no information about the actual position of the mobile client *User B*. The route over *Access Point 2* is the preferred route. But when considering mobile clients in NDN, the right route could not be chosen easily.

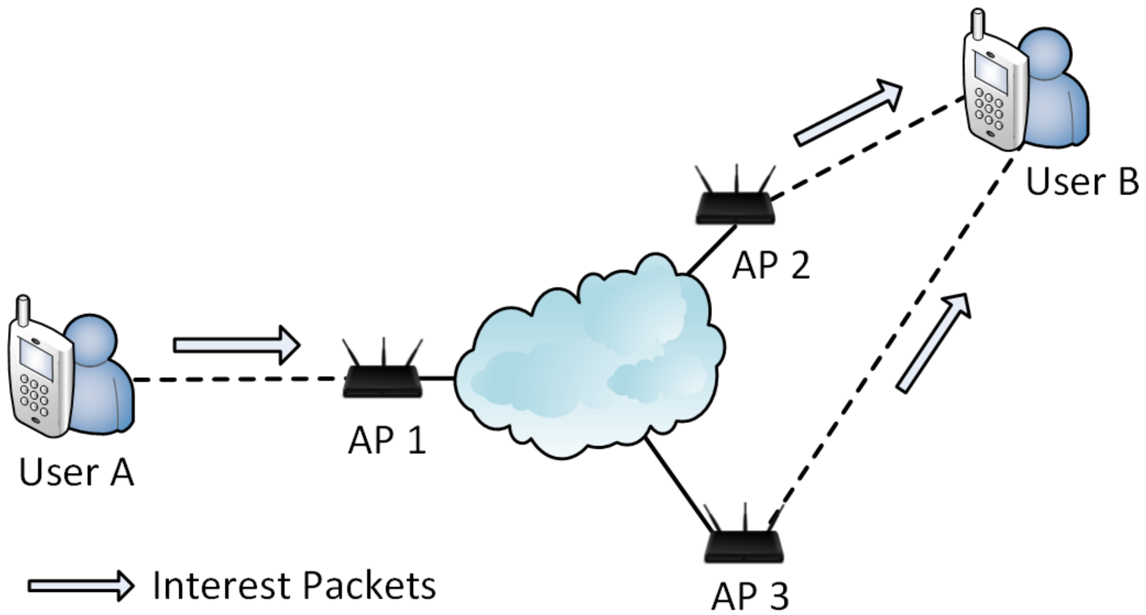

Fig. 2.7: Producer mobility: In this example, the content consumer and the content provider are mobile clients. The Forwarding Information Base has no information about the position of the content provider.

Because producer mobility is hard to realize in an NDN network, only content consumers are mobile clients in network simulations for the practical part of this master thesis [ZABZ16, CZMC+17]. For this reason, voice conferences are not considered, instead, mobile clients are listening to voicemail messages, which have been created by fixed clients. When voice conferences were realized in the evaluation scenario for this master thesis, then both communication partners were consumers and producers at the same time. Therefore, mobile clients would be also content producers and the problem of producer mobility would be present.

In the following chapter, the overall topic of this master thesis is described in detail. To reduce packet loss on loss-prone wireless links in NDN, the advantages of multiple available wireless links are introduced. These advantages can be used for network simulations with mobile clients for the practical part of this master thesis. Therefore, the term multi-path transmission is described. After this, the research questions are presented. Finally, the idea for the implementation of a new forwarding strategy for mobile clients is introduced.

3 Multi-Path Transmission for Mobile Clients

In this chapter, the main topic of this master thesis with the research questions is introduced. Firstly, short introductions to multi-path transmission and multi-path transmission for mobile clients in an NDN network are given. After this, the research questions for this master thesis are shown and described. For the research questions, the definitions of *Interest Satisfaction* and *Interest Satisfaction Ratio* are relevant. Therefore, these terms are defined as well. Finally, some considerations of how data can be requested with Interest Packets without redundancy and also with redundancy are presented. Furthermore, solution proposals of how data can be requested in a new forwarding strategy, which will be used for mobile clients in an NDN network, are given, to build a solution for the practical part of this master thesis and to answer the research questions. The implementation of this new forwarding strategy is described in chapter 4.

3.1 What is Multi-Path Transmission?

Firstly, it is explained what multi-path transmission means. Generally, when using multi-path transmission, multiple links are available at a network node and these links can be used for data transmission [LMRT+13]. This means, a network node can use one or more links for data transmission. Multi-path transmission can serve different purposes:

- Requesting data on multiple links to reduce load
- Requesting data redundantly to reduce packet loss
- Switching links when one link gets worse

At first, multi-path transmission can be requesting data over multiple available links, to reduce the transmission load on each link. Therefore, the data transmission can be split up over more than one link, to avoid an overload of links, as well as reducing the packet loss rate due to an overloaded link. With this approach, no redundant packets are produced.

Multi-path transmission can also be requesting data on multiple links redundantly. Therefore, data is requested separately and redundantly over more than one link. This causes several redundant packets. This means, multi-path transmission can be used for sending packets redundantly, to mitigate the packet loss rate due to duplicated packets.

Finally, multi-path transmission can also be switching between multiple links, when one link gets worse. Therefore, the data transmission switches to another link with a better connection, to reduce the packet loss rate as well.

For this master thesis, the advantages of multi-path transmission are used for the design and implementation of a new forwarding strategy, which is created for mobile clients in an NDN network. For this forwarding strategy, the advantages of sending redundant packets over multiple links, as well as switching between links, when one of the available links where the data is transmitted gets worse, are used. In the following, multi-path transmission for mobile clients is introduced.

3.2 Multi-Path Transmission for Mobile Clients

Mobile clients can use multi-path transmission when multiple wireless links of wireless networks are available. These wireless links can be links from mobile clients to WiFi access points of WiFi networks, as well as links from mobile clients to base transceiver stations of 4G or other cellular networks, like already shown in figure 2.4. Wireless links can be loss-prone, when the connection gets worse, for instance, when mobile clients move away from a WiFi access point or from a base transceiver station, which is already described in section 2.4.1, where loss-prone wireless links are introduced.

The aim of this master thesis is using multi-path transmission for mobile clients, to build a solution where packet loss on loss-prone wireless links in NDN networks is reduced. When packets are transmitted over one link and this link gets worse due to packet loss, then the advantages of multi-path transmission can be used. Further packets can be transmitted on multiple available links redundantly, instead of transmitting packets over the loss-prone link. But transmitting packets always redundantly causes several redundant packets. To overcome this disadvantage of multi-path transmission, a better link can be chosen for further packet transmission, where fewer packet losses occur. When a link with a better connection has been found, then further packets can be transmitted on this link and no longer redundantly.

3.3 Research Questions

There are three research questions this master thesis deals with. These research questions are described in this section. Before introducing the research questions, additional terms have to be introduced. The terms Interest Satisfaction and Interest Satisfaction Ratio are defined in the following.

3.3.1 Interest Satisfaction

A content consumer requests data with an Interest Packet from a content provider in an NDN network. A Data Packet with the corresponding data to the Interest Packet follows after the Interest Packet. The communication model of NDN has been already described in section 2.2 in detail.

Interest Satisfaction is defined as follows. An Interest Packet is satisfied through the corresponding Data Packet. This means, the Data Packet of the data requested by an Interest Packet is received by the content consumer. When the Data Packet of the data requested by an Interest Packet will not be received at the content consumer, then the Interest Packet is not satisfied, due to packet loss of the Interest Packet or the corresponding Data Packet.

This means, several Interest Packets are sent from a content consumer to a content provider. For each Interest Packet, the content provider answers with a Data Packet. When a Data Packet has been received at the content consumer with the requested data, then the corresponding Interest Packet to this Data Packet is satisfied.

3.3.2 Interest Satisfaction Ratio

The Interest Satisfaction Ratio (ISR) is a value between 0 and 100, that presents the percentage of satisfied Interest Packets. The formula to calculate the Interest Satisfaction Ratio is shown in equation 3.1.

$$ISR = \frac{Number\ of\ satisfied\ Interest\ Packets}{Number\ of\ total\ Interest\ Packets} \cdot 100 \tag{3.1}$$

An Interest Satisfaction Ratio of 100 percent means, all Interest Packets are satisfied through the corresponding Data Packets. Further, an Interest Satisfaction Ratio of 0 percent means, no Interest Packet is satisfied through the corresponding Data Packet.

As already explained, this master thesis uses the advantages of multi-path transmission, to reduce packet loss on loss-prone wireless links in NDN networks. Therefore, a new forwarding strategy for mobile clients has been implemented. This forwarding strategy decides where and how to forward incoming Interest Packets to increase the Interest Satisfaction Ratio. Moreover, the performance of the forwarding strategy has been evaluated within an evaluation scenario. The evaluation scenario consists of a predefined number of WiFi networks with mobile clients listening to voicemail messages in a randomly generated NDN network. These voicemail messages have been created by fixed clients, which are located elsewhere in the NDN network. The Interest Satisfaction Ratio is one of the metrics, which is used for the performance evaluation. In the following, the research questions for this master thesis are introduced.

3.3.3 Question 1: Increase the Interest Satisfaction Ratio

Packets can get lost, for instance, on loss-prone wireless links. An aim of this master thesis is to increase the Interest Satisfaction Ratio. This means that fewer packets get lost and a higher number of Data Packets will be received by content consumers which are requesting data at content providers. Therefore, the first research question for this master thesis is defined as follows.

"How can the Interest Satisfaction Ratio (ISR) be increased in loss-prone NDN networks?"

The Interest Satisfaction Ratio can be increased in loss-prone NDN networks, when data is requested redundantly by a content consumer when sending redundant Interest Packets. But this causes an amount of redundant Interest Packets and Data Packets. For this reason, other research questions have to be considered as well.

3.3.4 Question 2: Reduce the Number of Redundant Interest Packets

When data is always requested redundantly, then a lot of redundant Interest Packets are sent by the content consumer. This causes a higher packet rate, due to a lot of redundant packets. Therefore, another aim is to reduce the number of redundant requests of data, due to redundant Interest Packets, by addressing the following research question.

"How can the number of redundant Interest Packets be reduced in loss-prone NDN networks?"

The number of redundant Interest Packets can be reduced in loss-prone NDN networks, when data will not always be requested redundantly by a content consumer, due to sending redundant Interest Packets. Redundant Interest Packets also cause redundant Data Packets. For this reason, the following research question has to be considered as well.

3.3.5 Question 3: Reduce the Number of Redundant Data Packets

Due to the redundant requests of data with redundant Interest Packets, several redundant Data Packets may be sent to the content consumer. This causes again a higher packet rate, due to a lot of redundant packets. Moreover, another aim is to reduce the number of redundant Data Packets with the following research question.

"How can the number of redundant Data Packets be reduced which is caused by requesting redundant data with redundant Interest Packets in loss-prone NDN networks?"

To build a solution where packet loss on loss-prone wireless links in NDN networks is reduced, and where the previously presented research questions are considered, considerations of sending Interest Packets redundantly and without redundancy in an NDN network are made. Therefore, ideas for a solution for the implementation of a new forwarding strategy for mobile clients are collected in the following. This new forwarding strategy is designed and implemented for the practical part of this master thesis.

3.4 Considerations of Requesting Data with Interest Packets

In the following, two possibilities of requesting data with Interest Packets are introduced. The first option is sending Interest Packets over one link without redundancy. Another option is sending Interest Packets over multiple links redundantly.

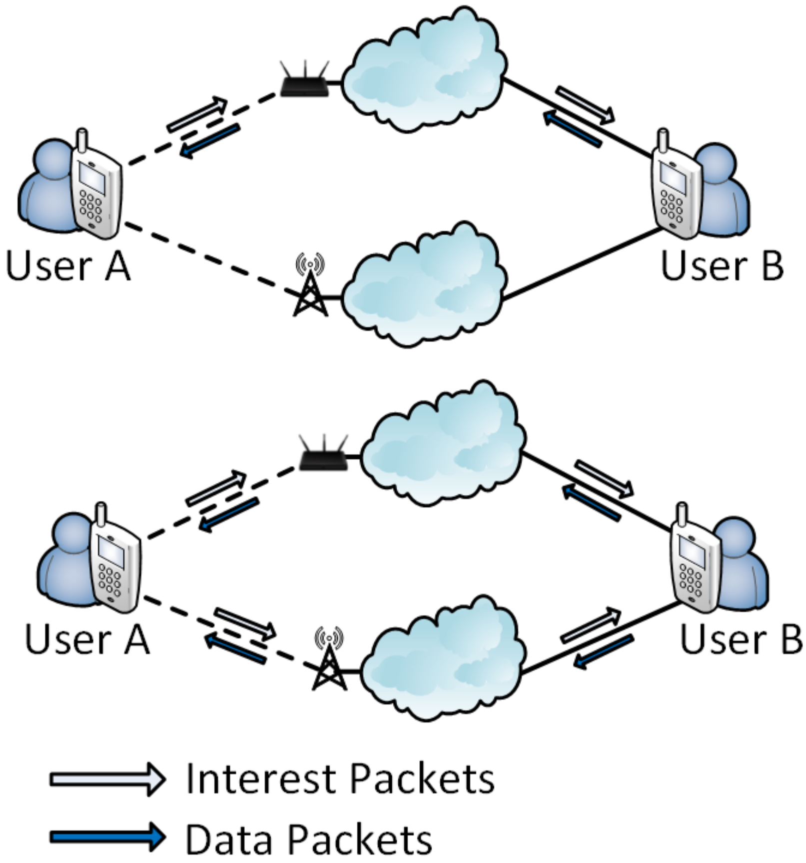

Fig. 3.1: Considerations of requesting data with Interest Packets: Sending Interest Packets over one link (upper picture) and sending redundant Interest Packets over multiple links, for instance, two links (lower picture).

Sending Interest Packets over one link without redundancy is the first described option. This means, that each Interest Packet is sent only once, and this will cause the following on loss-prone wireless links in an NDN network:

- A lower Interest Satisfaction Ratio
- A higher packet loss rate
- A lower Quality of Experience for the end-user, but also
- No redundant Interest Packets and no redundant Data Packets

Sending Interest Packets on multiple links redundantly is the second option. This will cause the following on loss-prone wireless links in an NDN network:

- A higher Interest Satisfaction Ratio
- A lower packet loss rate
- A higher Quality of Experience for the end-user, but also
- Redundant Interest Packets and Data Packets

The previously described considerations of requesting data with Interest Packets are shown in figure 3.1. On the upper picture, a content consumer *User A* requests Interest Packets over one link by the content provider *User B*. On the lower picture, *User A* sends redundant Interest Packets over multiple links to the content provider *User B*, where data is requested and received over the two available links.

In the following section, ideas for the design and implementation of the new forwarding strategy for mobile clients in NDN networks are collected, which consider the previous findings. Both possibilities of requesting data with Interest Packets with and without redundancy are considered for the design and implementation of the new forwarding strategy.

3.5 Considerations and Idea for the Implementation

There are considerations and ideas for the practical part of this master thesis, where a new forwarding strategy for mobile clients in NDN networks is designed and implemented, to answer the research questions, which have already been presented in this chapter. In the evaluation scenario for this master thesis, this forwarding strategy will be only used for mobile clients using loss-prone wireless links. All other nodes in the NDN network in the evaluation scenario are using a default forwarding strategy, called Best-Route Forwarding Strategy, which has already been introduced in section 2.3.5.

The idea for the implementation of the new forwarding strategy for mobile clients is that content consumers start sending Interest Packets redundantly for a predefined time over multiple links, in order to find the best link. In this case with multiple links, all available links at a node, except the link where the incoming Interest Packet comes from, are used. This is the same behavior as sending Interest Packets with the Multicast Forwarding Strategy, which has already been introduced in section 2.3.6.

For mobile clients, the Interest Satisfaction Ratio is calculated for a predefined time for each available link, while Interest Packets are forwarded. The best link of a mobile client is the link with the highest Interest Satisfaction Ratio during this predefined time, and the Interest Satisfaction Ratio of the best link has to be higher than a predefined Interest Satisfaction Ratio threshold.

When the best link has been found, which meets the previously defined requirements, then further Interest Packets are sent only on the best link without redundancy, in order to reduce the packet rate. If the connection of the selected link gets worse, for instance, due to a higher packet loss, redundant Interest Packets can be sent on multiple links again for a predefined time. As long as the connection stays worse, redundant Interest Packets can be sent. If the connection gets better, for instance, when fewer packet losses occur, Interest Packets need no longer be sent redundantly, and the best link can be selected again.

Mobile clients notice a bad connection, when the Interest Satisfaction Ratio of the link, where the data transmission takes place, falls under a certain predefined threshold. This means that fewer Data Packets are received by the content consumer due to packet loss. This can happen, for instance, when mobile clients move away from access points or base transceiver stations of cellular networks.

The pseudocode for the new forwarding strategy is presented in the following:

```
forwardingStrategy = multicast
timerInSec = 1
thresholdISR = 90

start forwardingDecisionTimer(timerInSec)

while true

  while forwardingDecisionTimer is running
    if (Interest Packet received)
      forward Interest Packet with forwardingStrategy
      count Interest Packet
    else if (Data Packet received)
      count Data Packet

  calculate ISR for all links
  calculate link with maxISR

  if maxISR > thresholdISR
    forwardingStrategy = unicast over link with maxISR
  else
    forwardingStrategy = multicast

  start forwardingDecisionTimer(timerInSec)
```

In the next chapter, the implementation of the practical part of this master thesis is described in detail. Therefore, the implementation of the new forwarding strategy for mobile clients in NDN networks is described, which can be used in ndnSIM. Due to the implementation of the new forwarding strategy, additional parts of ndnSIM have to be extended as well. Finally, for generating and evaluating an evaluation scenario, to be able to evaluate the performance of the newly implemented forwarding strategy, other parts and extensions of ndnSIM have to be implemented and extended as well.

4 Implementation of Multi-Path Transmission for Mobile Clients

In this section, the implementation part of this master thesis is described in detail. For the implementation of the practical part of this master thesis, an NDN network simulator called ndnSIM [LosA17] is used. With this simulator, simulation scenarios can be created, where it is possible to simulate NDN networks and NDN communication within an NDN network.

When using this simulator, a set of available forwarding strategies is provided, which can be used in created simulation scenarios. The Best-Route Forwarding Strategy and the Multicast Forwarding Strategy have been already introduced in sections 2.3.5 and 2.3.6. The set of available forwarding strategies can be extended with the implementation of a new forwarding strategy. In the scope of this master thesis, a new forwarding strategy, called Mobile-Client Forwarding Strategy, has been created. This forwarding strategy has been designed and implemented to use it only for mobile client nodes in an NDN network, and will be used in ndnSIM simulation scenarios for this master thesis. For the implementation of the new forwarding strategy, other ndnSIM classes have been extended, to be able to use the new forwarding strategy. These necessary changes are also described in this chapter.

To evaluate the performance of the new forwarding strategy, a simulation scenario with an NDN network has been created for using it with ndnSIM. The NDN nodes and the connections between these NDN nodes of the generated NDN network are created randomly. To create this randomly generated network for simulation scenarios, a network generator based on the network topology generator *BRITE* [MeMB00] is used within ndnSIM which has been implemented and extended by the ITEC Multimedia Communication group [MoPH17, PoRH17a]. This network generator has been extended again in the scope of this master thesis with several functions, to create a specific randomly generated NDN network for the evaluation scenario, which evaluates the performance of the newly implemented Mobile-Client Forwarding Strategy. The network generator is extended to be able to create mobile client nodes, WiFi networks with WiFi access point nodes, and wireless links between WiFi access point nodes and mobile client nodes. The evaluation scenario is described in the next chapter in detail, where the results of the evaluation are also shown.

Firstly, an NDN network simulator called ndnSIM is introduced in this chapter. After this, the required changes in the Pending Interest Table entry class in the *Named Data Networking Forwarding Daemon* (NFD) [ASZZ+16] submodule of ndnSIM are described, which are needed for the implementation of the new forwarding strategy. Furthermore, the development and implementation of the new forwarding strategy for mobile clients and the forwarding decision of this forwarding strategy are described in detail. After this, the implementation of the extensions of the network generator for creating a randomly generated NDN network for the evaluation scenario is described. After this, the required changes in the Forwarding Information Base entry class are described, which are needed also for the implementation of the new forwarding strategy. Then the consumer application extensions are introduced, which are important for the evaluation scenario to avoid retransmitting packets in the evaluation scenario. Finally, the implementation of the used evaluation scripts is presented, which are used to evaluate and compare different simulation scenario results and to print several result charts.

4.1 The ndnSIM Simulator

For the practical part, an NDN simulator called ndnSIM [MAMZ16] [LosA17] [AfMZ12] [MaAZ17] has to be installed on an Ubuntu distribution. The simulator is based on the NS-3 network simulator framework [ASZZ+16]. With this simulator, an NDN network with NDN nodes can be created and NDN communication can be simulated. This means that nodes use the NDN network layer protocol instead of IP.

A simulation visualizer called *NS-3 PyViz* can be used, to be able to visualize NDN network scenarios and NDN simulations in ndnSIM [Foun17]. The visualizer is very useful to show the randomly generated NDN network or when NDN nodes use mobility models, to be able to check if the position, the movement, and the movement area of the mobile client nodes are correct. It is also very useful to check on which routes Interest Packets or Data Packets are transmitted in case of an error for debugging purposes.

For the implementation of this master thesis, Ubuntu 16.04 LTS version is used and version 2.3 of ndnSIM is recommended, otherwise, minor changes may have to be made in the implementation part of this master thesis, when another version of ndnSIM will be used. An installation guide, a documentation of the simulator, and useful tutorials, on how simple simulation scenarios can be created to simulate various NDN networks, can be found at the official site of ndnSIM [LosA17] and on the site of NS-3 [ASZZ+16]. Programs like git and Python have to be installed before downloading the source code of ndnSIM with git. A detailed description of how to set up and install the required programs and how to download ndnSIM source code can be found in an installation guideline on a GitHub repository [Berg17].

The simulator is open source and so ndnSIM can be extended with various components. NdnSIM provides several implemented forwarding strategies, like Best-Route and Multicast among others. The forwarding strategies Best-Route and Multicast have already been introduced in sections 2.3.5 and 2.3.6. A list of available forwarding strategies which can be used in ndnSIM can be found on the official site of ndnSIM [ASZZ+16]. These forwarding strategies are available in ndnSIM through the integration of the Named Data Networking Forwarding Daemon (NFD) [ASZZ+16]. In the practical part of this master thesis, a new forwarding strategy has been developed and implemented.

To be able to implement a new forwarding strategy, the ndnSIM source code is required. The source code of ndnSIM consists of different parts. At first, the source code for NS-3 is required. For the implementation of the practical part of this master thesis, the following git commit hash is used for the NS-3 source code:

- git clone https://github.com/named-data-ndnSIM/ns-3-dev.git ns-3
- commit hash: 333e6b052c101625199af40107edd6e379a36119
- version: ns-3.23-dev-ndnSIM-2.1-1-g333e6b0

The next part is the Python bindings generation library, which is used for NS-3 PyViz. For the implementation of the practical part of this master thesis, the following git commit hash is used for the Python bindings generation library source code:

- git clone https://github.com/named-data-ndnSIM/pybindgen.git pybindgen
- commit hash: 4806e4f322202b8e9a11b2fd216b6bdee41de4dc

Finally, the source code for ndnSIM with the git submodules ndn-cxx library and NDN Forwarding Daemon (NFD) is also required. For the implementation of the practical part of this master thesis, the following git commit hash is used for the ndnSIM source code:

- git clone –recursive https://github.com/named-data-ndnSIM/ndnSIM.git ns-3/src /ndnSIM
- commit hash: 0970340dd68742e5433f237c0a48de35986cd597
- version: ndnSIM-2.3-2-g0970340

Moreover, the following ndn-cxx library version is used as a git submodule in ndnSIM:

- commit hash: 4692ba80cf1dcf07acbbaba8a134ea22481dd457
- version: ndn-cxx-0.5.0-24-g4692ba8

Furthermore, the following NFD version is used as a git submodule in ndnSIM:

- commit hash: 38111cde9bab698f6eaf1a9d430130c2cbb3eca4
- version: NFD-0.5.0-41-g38111cd

Finally, a scenario template can be used when creating a simulation scenario in ndnSIM for the evaluation. For the implementation of the practical part of this master thesis, the following git commit hash is used for the scenario template source code:

- git clone https://github.com/named-data-ndnSIM/scenario-template.git voicemail_scenario

- commit 7aeb1a08f17b52c7c0cb8c3f517a223b555e4783

In the following, the implementation details of the implemented forwarding strategy, the implementation details of several extensions which are needed for the newly implemented forwarding strategy and the implementation details for extensions to be able to create and evaluate the evaluation scenario, are described in detail. The GitHub repository [Berg17] provides the source code for the implementation of the practical part of this master thesis, based on the scenario template. This GitHub repository also provides an installation guideline for using the implementation of this master thesis with the evaluation scenario for the newly implemented forwarding strategy.

4.2 Pending Interest Table Entry Extensions

The Pending Interest Table is a table which is used for forwarding in NDN networks. It is located on each node in an NDN network. Interest Packets which have already been forwarded to an outgoing face and which are waiting for the arrival of the corresponding Data Packets, are stored as an entry in the Pending Interest Table. Forwarding in NDN networks is already described in detail in section 2.3 and the content of a Pending Interest Table and a Pending Interest Table entry is shown in section 2.3.2 as well.

Before implementing the new forwarding strategy, extensions in the Pending Interest Table entries have to be made, so that the functionality of the new forwarding strategy works. The new forwarding strategy has to distinguish between Data Packets that arrive on a node for the first time and Data Packets that arrive redundantly on a node. When a Data Packet arrives repeatedly, then this Data Packet is a duplicate packet due to requesting data redundantly. Therefore, to distinguish if a Data Packet arrives for the first time or arrives repeatedly, changes in the Pending Interest Table entry class have to be made, which is implemented in the following file:

- Filename: *pit-entry.hpp*
- Location of the file: *ndnSIM_2-3/voicemail_scenario/extern*
- This file replaces a file in ndnSIM: *ndnSIM_2-3/ns-3/src/ndnSIM/NFD/daemon /table/pit-entry.hpp*

These files can be found on the GitHub repository [Berg17]. The original file from the NDN Forwarding Daemon integration of ndnSIM does not have to be deleted. When a file in the extern folder of a simulation scenario has the same filename as an already existing file of ndnSIM, then the new file will be used instead of the already existing file.

The class *Entry* in this file has been extended with a boolean variable *m_satisfied*, which is initially set to *false* for each Pending Interest Table entry. This class has also been extended with the following methods for this variable:

- isSatisfied
- setSatisfied

The methods *isSatisfied* and *setSatisfied* are added as well, to get and set the value of the variable *m_satisfied* of a Pending Interest Table entry. For each arriving Data Packet, the value of *m_satisfied* is checked in the newly implemented forwarding strategy. When a Data Packet arrives for the first time, then *m_satisfied* is set to *true* in the Pending Interest Table entry. When the same Data Packet arrives repeatedly, then the node notices that the arrived Data Packet is a redundant Data Packet.

This extension of the Pending Interest Table entry with the field *m_satisfied* is important to be able to count arriving Data Packets correctly for the calculation of the Interest Satisfaction Ratio value in the newly implemented forwarding strategy. A Data Packet which previously has already arrived and has already been counted when it arrives for the first time, should not be counted a second time. Therefore, duplicate Data Packets will not be counted and considered for the calculation of the Interest Satisfaction Ratio value, which will be used for the decision of how to forward further Interest Packets. The term Interest Satisfaction Ratio has already been introduced in section 3.3.2. In the following part, the implementation and behavior of the new forwarding strategy are described in detail.

4.3 Implementation of the Mobile-Client Forwarding Strategy

In this section, the implementation of the new forwarding strategy called Mobile-Client is described. The Mobile-Client Forwarding Strategy is a client forwarding strategy where a client decides how to forward incoming Interest Packets. The forwarding strategy was designed for the use in NDN networks for mobile client nodes. When mobile client nodes move away from an access point or a base transceiver station of a wireless network, then the connection of the used wireless link can get worse and packet loss can occur due to loss-prone wireless links.

4.3.1 General Implementation

The implementation of the forwarding strategy is provided on the GitHub repository [Berg17] and the class *MobileClientStrategy* is implemented in the following files:

- Filenames: *mobile-client-strategy.hpp* and *mobile-client-strategy.cpp*
- Location of the files: *ndnSIM_2-3/voicemail_scenario/extensions/fw/*

The class *MobileClientStrategy* extends the class *Strategy*, holds a constructor method *MobileClientStrategy* and overridable methods from the class *Strategy*:

- MobileClientStrategy
- afterReceiveInterest
- beforeSatisfyInterest

The class *MobileClientStrategy* also holds other methods, which are described later in this section, when the forwarding decision is explained in detail.

The method *afterReceiveInterest* is a pure virtual and an overridable method of the class *Strategy*. Pure virtual means this method has to be implemented at least by the Mobile-Client Forwarding Strategy. If the Mobile-Client Forwarding Strategy is installed on a node, for instance, on a mobile client node, then the *afterReceiveInterest* method is called each time an Interest Packet arrives at this node or when a new Interest Packet is created and sent out by this node. Within this method, the Mobile-Client Forwarding Strategy decides how to forward the incoming Interest Packet. This means, the forwarding decision takes place within this method. This method has the following parameters:

- const Face& inFace
- const Interest& interest
- const shared_ptr<pit::Entry>& pitEntry

where the *inFace* represents the face of the incoming Interest Packet. The parameter *interest* is the Interest Packet itself, which can be forwarded afterward. Another parameter is the *pitEntry*, which represents the Pending Interest Table entry for an Interest Packet. In the *pitEntry* it is stored, for instance, from which face of the current node this Interest Packet comes from.

The method *beforeSatisfyInterest* is called each time before a Data Packet corresponding to an Interest Packet arrives at this node. This method is also an overridable method of the class *Strategy* and has the following parameters:

- const shared_ptr<pit::Entry>& pitEntry
- const Face& inFace
- const Data& data

where *pitEntry* represent the Pending Interest Table entry for the requested data, *inFace* represents the incoming face, where the Data Packet arrives and the parameter *data* contains the arrived Data Packet. In the following, the implementation of the forwarding decision in the *MobileClientStrategy* class is described in more detail.

4.3.2 Forwarding Decision

In forwarding strategies, each node can choose on which face a created Interest Packet or an incoming Interest Packet will be forwarded. The decision takes place in the method *afterReceiveInterest*. With the newly implemented forwarding strategy, called Mobile-Client, an Interest Packet can be forwarded by this node to all available faces except the face, where the Interest Packet comes from, or only to a selected face.

At the beginning, each Interest Packet is forwarded to all available faces, except the face where the Interest Packet arrives from. For this forwarding strategy, further decisions where to forward the next Interest Packets, depend on a calculated value, called the

Interest Satisfaction Ratio, which has already been introduced in section 3.3.2. In the Mobile-Client Forwarding Strategy, each node calculates the Interest Satisfaction Ratio value for each face. This helps to find out which outgoing face is the best face. This means, which wireless link has the best connection, where fewer packet losses occur. After the calculation, this forwarding strategy chooses how to forward further Interest Packets for a predefined time, based on the results from the Interest Satisfaction Ratio value calculation. So the forwarding strategy can forward further Interest Packets to a selected face or again to all available faces, except the incoming face of the Interest Packet. When only one face is selected for further forwarding, the selected face is the face with the highest Interest Satisfaction Ratio value of this node and the Interest Satisfaction Ratio value of this face also has to be higher than a predefined threshold value. After a predefined time, the Interest Satisfaction Ratio values are calculated again for the forwarding decisions on further Interest Packets.

This forwarding strategy is designed for mobile clients in an NDN network. So for this case, incoming Interest Packets are always Interest Packets which are created on mobile client nodes and which will be forwarded on wireless links to WiFi access points of WiFi networks. In the following, the implementation of the forwarding decision in the class *MobileClientStrategy* is described in more detail.

Outgoing Interest Packets and incoming Data Packets are counted, to be able to calculate the Interest Satisfaction Ratio value for each face of a node where the Mobile-Client Forwarding Strategy is installed. Therefore, each node counts for all available faces and for each prefix separately the number of outgoing Interest Packets and the number of incoming Data Packets. The used prefix is a part of the content name of Interest Packets and Data Packets.

For storing the calculated values, the following counter variables are used:

- std::map<std::string, std::map<std::string, int>> allInterestCounter
- std::map<std::string, std::map<std::string, int>> allDataCounter

The variable *allInterestCounter* counts the number of outgoing Interest Packets for each available prefix and for each face of a node separately. These values are stored in a map of maps. The index of the first map is the prefix of the requested data, where this prefix is used by requesting the Interest Packet, and this index points to another map. The index of the second map is the face ID of the face, where the Interest Packet is forwarded and this index points to the number of outgoing Interest Packets on this face. This means, the number of outgoing Interest Packets for a specific prefix and a specific face is stored in the variable *allInterestCounter*.

The *allDataCounter* variable counts the number of incoming Data Packets for each prefix and for each face the same way as outgoing Interest Packets are counted with the variable *allInterestCounter*. Both counter variables are relevant for the forwarding decision. Based on these variables, the Interest Satisfaction Ratio values for each face on a node can be calculated, which is required for the decision where to forward further Interest Packets.

The following variables are used to store the Interest Satisfaction Ratio values for each face and to compare these values with a predefined threshold value:

- std::map<std::string, int> isr
- int isrThreshold

The variable *isr* stores the last calculated Interest Satisfaction Ratio values of all available faces of this node in a map. The index of this map is the face ID, which points to the last calculated Interest Satisfaction Ratio value of a specific face. The variable *isrThreshold* represents a threshold value for the Interest Satisfaction Ratio values which defines the value which should be reached at least by the best face, after this to be able to select this face as the best face. The default value for this predefined threshold is set to 90 for the implementation of the practical part of this master thesis. In section 5.4.1 different threshold values are compared to find the value which is used for evaluating the Mobile-Client Forwarding Strategy. The *isrThreshold* is a predefined value which can influence the forwarding decision within the Mobile-Client Forwarding Strategy. If the Interest Satisfaction Ratio value of a specific face is under this predefined threshold value, then this face cannot be chosen as the best face for the forwarding decision in the forwarding strategy.

For the forwarding decision, the following methods are also available in the Mobile-Client Forwarding Strategy. These methods are introduced in the following:

- updateCounter
- onTimerTimedOut
- calculateForwardingDecision
- calculateISR
- getISRFace
- refreshCounters
- getMaxISRFaceId

To calculate the Interest Satisfaction Ratio values for all available faces for the forwarding decision, a timer is used in the forwarding strategy. Outgoing Interest Packets and incoming Data Packets are counted while this timer is running. At the beginning, the timer starts in the method *afterReceiveInterest*, when the first Interest Packet has been received or has been created on the installed node. This timer is running a predefined time, which is by default one second. When the timer expires for the first time, then the calculation of the Interest Satisfaction Ratio value for all available faces for the forwarding decision takes place. When this timer expires later again, then the recalculation of the Interest Satisfaction Ratio values takes place. The duration of the timer is a predefined value, but changing this value can influence how often the Interest Satisfaction Ratio values are recalculated.

After the timer has started with this period of time, all outgoing Interest Packets and all incoming Data Packets for each prefix and for each face are counted by the variables *allInterestCounter* and *allDataCounter*, until the timer expires after the predefined time. The method *updateCounter* updates the variable *allInterestCounter* when an Interest Packet in the *afterReceiveInterest* method is sent out to an outgoing face. The method *updateCounter* updates also the variable *allDataCounter* when an incoming Data Packet in the *beforeSatisfyInterest* method arrives. When the timer expires after the predefined time, then the method *onTimerTimedOut* is called, where the forwarding decision is calculated within the method *calculateForwardingDecision*. After calculating the forwarding decision, the timer starts again for the predefined time.

In the method *calculateForwardingDecision*, the calculation of the Interest Satisfaction Ratio value for each face takes place and the Interest Satisfaction Ratio values for each face are stored in the previously introduced variable *isr*. After this, the variables *allInterestCounter* and *allDataCounter* are cleared, to start counting again and to prepare for the recalculation of the Interest Satisfaction Ratio values.

With the method *getISRFace* in the *calculateForwardingDecision* method, the Interest Satisfaction Ratio value for each face is calculated and returned within a map which is stored in the *isr* variable. The *allInterestCounter* and *allDataCounter* stores counter values for each prefix and for each face separately. Before calculating the Interest Satisfaction Ratio value for a specific face, all Interest Packets from the same face but with a different prefix have to be summed up. The same has to be done with Data Packets from the same face and a different prefix. After this, the Interest Satisfaction Ratio values for each face can be calculated and stored in this list. The Interest Satisfaction Ratio value for a face is calculated in the method *calculateISR* using equation 3.1.

In the *afterReceiveInterest* method, the forwarding decision takes place. Initially, the Interest Satisfaction Ratio values for all faces are zero. To calculate the Interest Satisfaction Ratio values for all available faces for the forwarding decision, a timer is used in the forwarding strategy. After the timer expires for the first time, the Interest Satisfaction Ratio values are calculated and after this, the variable *isr* holds the Interest Satisfaction Rato values for all faces. When this timer expires later again, then the recalculation of the Interest Satisfaction Ratio values takes place. The face with the highest Interest Satisfaction Ratio value is chosen in the method *afterReceiveInterest*. The face ID with the highest Interest Satisfaction Ratio value is returned from the method *getMaxISRFaceId*. Then it is checked if the Interest Satisfaction Ratio value of this face is smaller or larger than the Interest Satisfaction Ratio threshold value. If the Interest Satisfaction Ratio value of this face is smaller than the predefined threshold value, then the incoming Interest Packet will be forwarded to all available faces. But if the Interest Satisfaction Ratio value of this face is larger than the predefined threshold value, then the incoming Interest Packet will be forwarded only to this face, which represents the best face of this node to the current time.

In the following, the implementation of the NDN network is described in detail. This NDN network is created for the performance evaluation of the Mobile-Client Forwarding Strategy. The network nodes in this NDN network are generated randomly.

4.4 Network Generator Extensions

To evaluate the performance of the Mobile-Client Forwarding Strategy within an evaluation scenario, a network topology generator named BRITE [MeMB00] is used. Therefore, a network generator using BRITE has been implemented and extended by the ITEC Multimedia Communication group [MoPH17, PoRH17a], where a randomly generated network can be created. This network generator has been further extended in the scope of this master thesis with specific methods, to be able to add other network components for the performance evaluation of the Mobile-Client Forwarding Strategy. For the performance evaluation, an evaluation scenario with mobile clients, which are listening to voicemail messages created by fixed clients within an NDN network is used. Therefore, fixed client nodes as well as WiFi networks with mobile client nodes are added, after the randomly generated network is created.

The implementation of the extended network generator can be found on the GitHub repository [Berg17]:

- Filenames: *ndnbritehelper.h*, *ndnbritehelper.cc*, *networkgenerator.h* and *networkgenerator.cc*
- Location of the files: *ndnSIM_2-3/voicemail_scenario/extensions/randnetworks/*

Based on the network generator source code from the ITEC Multimedia Communication group [MoPH17, PoRH17a], the extensions for the practical part of this master thesis are implemented.

To be able to use the network generator in ndnSIM, recommended changes in ndnSIM have been made as well by the ITEC Multimedia Communication group [MoPH17, PoRH17a]. Therefore, the following files have been adapted in ndnSIM and can also be found on the GitHub repository [Berg17]:

- Filenames: *ndn-link-control-helper.hpp* and *ndn-link-control-helper.cpp*
- Location of these files: *ndnSIM_2-3/voicemail_scenario/extern*
- These files replace the files in ndnSIM: *ndnSIM_2-3/ns-3/src/ndnSIM/helper/ndn-link-control-helper.hpp* and *ndnSIM_2-3/ns-3/src/ndnSIM/helper/ndn-link-control-helper.cpp*

Furthermore, a configuration file is necessary for the generation of the random network with the network generator. The used configuration file for creating the randomly generated NDN network can be found in the following folder on the GitHub repository [Berg17]:

- Filename: *brite_3_as.conf*

- Location of the files: *ndnSIM_2-3/voicemail_scenario/brite_configs/*

This BRITE configuration file for the network generator is also based on an implementation from the ITEC Multimedia Communication group [MoPH17, PoRH17a]. This configuration file has been adapted for creating the specific NDN network for this master thesis.

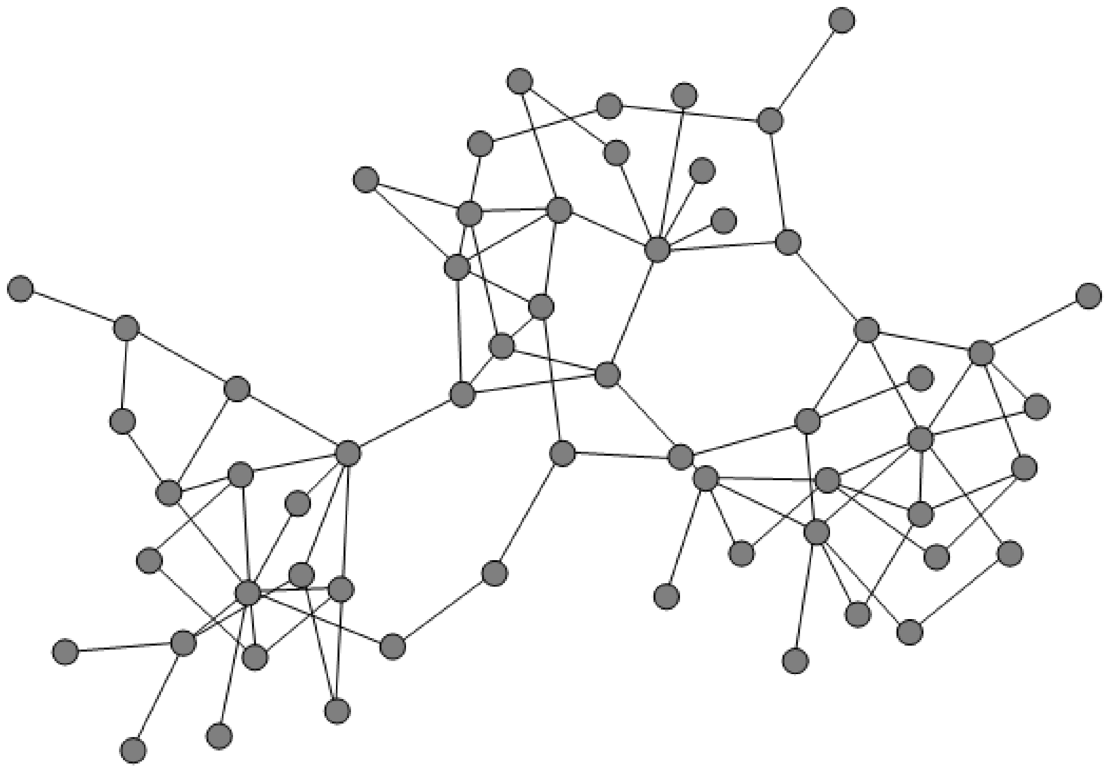

Fig. 4.1: A randomly generated network: With the network generator, a randomly generated network with three autonomous systems, 20 nodes per each autonomous system, and a predefined number of connections between nodes have been created. Each point in this figure represents a network node, and each line represents a connection between two network nodes.

When using the network generator based on BRITE, a randomly generated NDN network is created. This network consists of a predefined number of *Autonomous Systems*, where an autonomous system represents a set of network nodes. Each autonomous system consists of a predefined number of network nodes. Connections between nodes within the same autonomous system are available in each autonomous system. Connections between nodes of different autonomous systems can also exist. This randomly generated NDN network is created based on parameters specified in the BRITE configuration file. This means, the number of autonomous systems, the number of nodes per autonomous systems, the number of connections between nodes within the same autonomous system, and the number of connections between nodes of different autonomous systems are defined in the configuration file. When a new node is added to the network, then connections between the new node and already existing nodes are created. The connected nodes are chosen randomly when adding a new node. Several connections between nodes can be added additionally after creating the randomly generated network to achieve a better connectivity within the whole network. This can be done because, by default, a network with no redundant connections is created by the network generator.

In figure 4.1, an example of a randomly generated network with three autonomous systems is shown, which has been created with the network generator, and where each autonomous system consists of 20 nodes. In this example, several connections have been additionally added after generating the network to achieve a better connectivity within the whole network.

By default, the network generator generates networks with no redundant connections. Due to this fact, a predefined number of connections between network nodes can be added, after creating the randomly generated network. Therefore, the network generator provides methods where connections can be added randomly to the randomly generated network which has been created before:

- randomlyAddConnectionsBetweenTwoAS
- randomlyAddConnectionsBetweenTwoNodesPerAS

Depending on the desired connectivity between network nodes in the randomly generated network, several predefined connections can be added between two nodes of two autonomous systems or between two nodes within the same autonomous system. The method *randomlyAddConnectionsBetweenTwoAS* randomly adds connections between nodes of different autonomous systems. This means, connections between randomly chosen nodes from one autonomous system with randomly chosen nodes from another autonomous system are added. The method *randomlyAddConnectionsBetweenTwoNodesPerAS* randomly adds connections between nodes within the same autonomous system.

Both methods have the following parameters:

- int numberOfConnectionsPairs
- int minBW_kbits
- int maxBW_kbits
- minDelay_ms
- maxDelay_ms

where *numberOfConnectionsPairs* is the number of additionally added connections between nodes. Random bandwith values between *minBW_kbits* and *maxBW_kbits* as well as random delay values between *minDelay_ms* and *maxDelay_ms* are defined for creating these additional connections between nodes. The methods *randomlyAddConnectionsBetweenTwoAS* and *randomlyAddConnectionsBetweenTwoNodesPerAS* have been implemented by the ITEC Multimedia Communication group [MoPH17, PoRH17a].

So far, this randomly generated network only consists of router nodes. Consequently, the network generator has been extended in the scope of this master thesis to additionally add other network components that are needed for creating a whole WiFi network, to simulate mobile clients listening to voicemail messages which have been created by fixed clients. Therefore, fixed client nodes as well as WiFi networks with WiFi access point nodes, mobile client nodes, and wireless connections have to be added to the randomly generated NDN network.

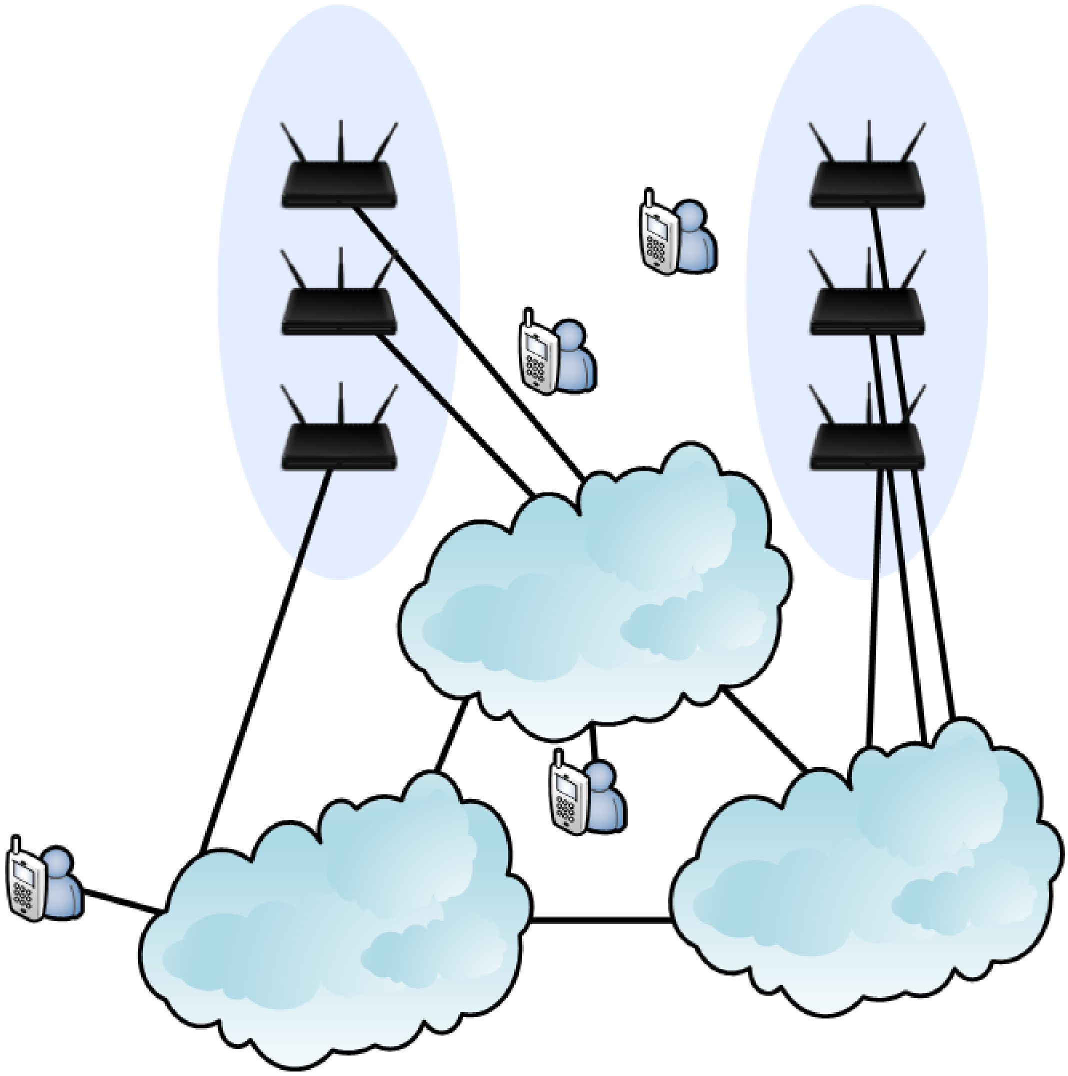

Fig. 4.2: A randomly generated network, where network components are added: Clouds represent autonomous systems. Fixed clients and access points of WiFi networks are connected to random nodes in autonomous systems, as well as some mobile clients are added between access points to the existing network.

In figure 4.2, an example network with additionally added network components to the randomly generated network is shown. The three clouds in the figure represent three autonomous systems, which have been already introduced in figure 4.1. The autonomous systems consisting of nodes are created with the network generator with predefined settings in the BRITE configuration file and by optionally adding additional connections between nodes in this randomly generated network. This network will be extended with other network components to fit the current needs for the evaluation scenario for this master thesis. Therefore, a predefined number of fixed client nodes, as well as a predefined number of access point nodes are added to the existing network. Each fixed client node and each access point node is connected to a randomly chosen router node located in a predefined autonomous system in the existing network. Each network component has a specific position. This position is also used for the visualization of the evaluation scenario in the visualizer of ndnSIM. This visualizer has already been introduced in section 4.1. The position of access point nodes is chosen so that mobile client nodes can use an area between access point nodes where mobile client nodes can move as shown in figure 4.2. These positions of the network components are chosen for the visualization of the evaluation scenario in the visualizer of ndnSIM. For simulating WiFi networks in ndnSIM, adding access point nodes to the network is necessary. Mobile client nodes are placed between the available WiFi networks. This means they are placed between the available access point nodes. Connections from each mobile client node to each access point node are added as well. Mobile client nodes can use these wireless links. These connections are not visible in figure 4.2 due to the fact that these connections are not the same as the other connections in the randomly generated network, but these connections are wireless connections.

The network generator provides methods for adding client nodes. Therefore client nodes on fixed positions and mobile client nodes are considered. The following methods are available to add client nodes to the randomly generated network:

- randomlyPlaceNodes
- placeMobileClientNodeAtPos

The method *randomlyPlaceNodes* has been implemented by the ITEC Multimedia Communication group [MoPH17, PoRH17a]. With this method, it is possible to add several fixed client nodes randomly to an existing network. These fixed client nodes are added the same way as router nodes were added before. The autonomous systems where the fixed client nodes are added are chosen randomly. Each fixed client node is connected to another randomly chosen node from a randomly chosen autonomous system.

To add not only fixed client nodes to the existing network but also mobile client nodes, the network generator has been extended in the scope of this master thesis with the method *placeMobileClientNodeAtPos*. Within this method, mobile client nodes can be added to the existing network at a predefined position. These positions of the mobile client nodes are chosen for the visualization of the evaluation scenario in the visualizer of ndnSIM. This means, mobile client nodes are added near available WiFi networks. This predefined position represents only the initial position of a mobile client. After installing a predefined mobility model on the mobile client, the mobile client will be able to walk. It is important that the chosen mobility model for the mobile client node must not be a constant mobility model. When using a constant mobility model, then nodes have always the same position and cannot move. In the case of mobile clients, a mobile client is not connected to a randomly chosen node from a randomly chosen autonomous system, but mobile client nodes will be added to a predefined area in the network without a wireless link connection to another node in the existing network. Wireless links are added later between mobile client nodes and access point nodes, after access point nodes have been added to the existing network.

The source code of the network generator has been extended in the scope of this master thesis with the following methods:

- placeAPNodesAtPositionInAS
- placeMobileClientNodeAtPos
- createWiFiLink

With the method *placeAPNodesAtPositionInAS*, access point nodes of WiFi networks can be added to an existing network. The method *placeMobileClientNodeAtPos* adds mobile client nodes to an existing network and the method *createWiFiLink* adds wireless links which connect mobile client nodes with access point nodes in an existing network. The implementation of these extensions, which is used for creating the evaluation scenario for this master thesis, is described in the following in detail.

4.4.1 Adding Access Point Nodes

To create and simulate access point nodes of WiFi networks in ndnSIM, access point nodes can be added to an existing randomly generated network, by extending the network generator with the previously mentioned method *placeAPNodesAtPositionInAS*. This method has the following parameters:

- int nodeCount
- std::string setIdentifierAP
- std::string setIdentifierRouter
- NodePlacement place
- PointToPointHelper *p2p
- int positionX
- int positionY
- int nextPositionY
- std::vector<int> ASnumbers

With this method, a predefined number of access point nodes can be added to an existing network, where these access point nodes are placed in a vertical line for the visualization of the evaluation scenario in the visualizer of ndnSIM. These access point nodes belong to the same WiFi network. The number of added access point nodes is defined by the parameter *nodeCount*.

The parameter *setIdentifierAP* contains a name for these access point nodes. The name for access point nodes is for each access point node the same, which is created within the same method call. This means, *nodeCount* number of access point nodes have the same name. An ascending number is appended at the end of the name of an access point node, to guarantee a unique name for each access point node within the network. The appended ascending number starts with zero.

Access point nodes of a WiFi network where, for instance, three access point nodes belong to this WiFi network can have the following names:

- FirstWifi_AP_0
- FirstWifi_AP_1
- FirstWifi_AP_2

An access point node is connected to a randomly chosen router node in the existing network. The parameter *setIdentifierRouter* sets names for router nodes which are connected to the created access point nodes. This parameter does not represent the name of how a router node is called in the network. The names of how router nodes are called in the network have already been set when creating these nodes by the network generator. But the *setIdentifierRouter* name represents only a name to identify router nodes which are connected to access point nodes to a later time in an evaluation scenario. These names have to be unique as well. Therefore, an ascending number is also appended at the end of the name of a router node.

For instance, the router nodes which are connected to access point nodes of a WiFi network can have the following names:

- Router_FirstWifi_AP_0
- Router_FirstWifi_AP_1
- Router_FirstWifi_AP_2

Finally, the node names *setIdentifierAP* and *setIdentifierRouter* are used for a map of nodes in the network generator, which is called *nodeContainerMap*. Only the parameter *setIdentifierAP* is also used for the name of how an access point node is called in the network.

The parameter *place* defines the type of router node. This is the node type of the randomly chosen router node which will be connected with an access point node. For instance, when the value of the parameter *place* is *ns3::ndn::NetworkGenerator::LeafNode*, then only leaf nodes are considered by selecting a random router node which will be connected with an access point node. The parameter *p2p* sets settings for links, which connect an access point node with a router node. For instance, the link delay or the data rate on the link is defined with the parameter *p2p*.

WiFi access point nodes can be placed at various positions in the randomly generated network. Therefore, the parameters *positionX* and *positionY* from the method *placeAPN-*

odesAtPositionInAS define the position where an access point node will be added to the network for the visualization of the evaluation scenario in the visualizer of ndnSIM. In the case, when the parameter *nodeCount* is one, then *positionX* and *positionY* represent the position of this access point. In the case when the parameter *nodeCount* is greater than one, then these parameters for the position represent the position of the first access point node. Further access point nodes have the same x position *positionX* but a different y position *positionY*. The y position of a further access point node is calculated by summing up the y position *positionY* from the previous access point node with the value of the parameter *nextPositionY* from the method *placeAPNodesAtPositionInAS*. An example of how access point nodes are placed in the network is shown in figure 4.3.

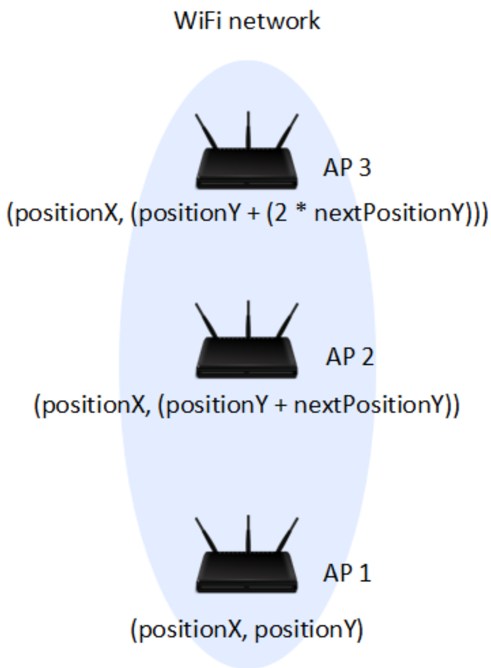

Fig. 4.3: Positions of access point nodes: This figure shows access point node positions from a WiFi network for the visualization in the visualizer of ndnSIM.

The parameter *ASnumbers* is a list of autonomous system IDs which can contain all available autonomous systems of the randomly generated network, or only specific autonomous systems of this network. This list contains autonomous system IDs, from where a randomly chosen node can be selected for connecting with an access point node. This means, a randomly chosen node can be selected also from a specific autonomous system or several autonomous systems, instead of selecting a randomly chosen node from all available autonomous systems.

In ndnSIM, WiFi networks can be created by adding wireless links between access point nodes and mobile client nodes. The settings for the WiFi network are defined by creating these wireless links. This will be described in detail after the introduction of how mobile client nodes can be added to the randomly generated network. In the following, the implementation of adding mobile client nodes to the randomly generated network is described in detail.

4.4.2 Adding Mobile Client Nodes

To create and simulate mobile client nodes in ndnSIM, mobile client nodes can be added to an existing randomly generated network by extending the network generator with the method *placeMobileClientNodeAtPos*. Mobile client nodes have a predefined name and it is recommended to place them near access point nodes of WiFi networks. This method has the following parameters:

- std::string name
- int positionX
- int positionY
- int minX
- int maxX
- int minY
- int maxY
- std::string mobilityModelId

A mobile client is created by calling the method *placeMobileClientNodeAtPos*. The parameter *name* is the name of the mobile client node in the network, and this name of the mobile client node should be unique in the whole network.

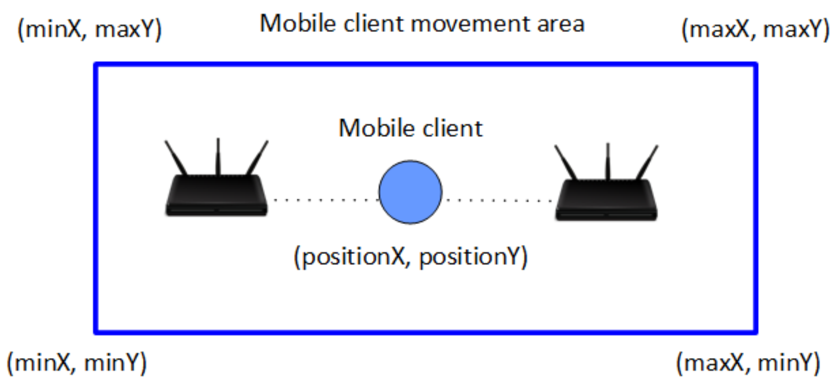

Fig. 4.4: Initial position of a mobile client node and boundaries for the movement area: This figure shows the start position of a mobile client node for the visualization in the visualizer of ndnSIM. The boundaries for the movement area are shown as blue lines.

The parameters *positionX* and *positionY* from the method *placeMobileClientNodeAtPos* represent the initial position of a mobile client node for the visualization of the evaluation scenario in the visualizer of ndnSIM. For each mobile client, a movement area with predefined boundaries is defined. This is a predefined area in which mobile client nodes can move. The parameters *minX*, *maxX*, *minY* and *maxY* are position values which define the movement area for a mobile client for the visualization of the evaluation scenario in the visualizer of ndnSIM. Within this area, mobile client nodes can walk when a mobility model is installed on the mobile client node. An example of how the movement area and a mobile client node are placed in the network is shown in figure 4.4.

For a mobile client node, a preferred mobility model can be set with the parameter *mobilityModelId*. By default, the Constant Position Mobility Model is installed on each network node, on each fixed client node, and on each access point node in the existing network. With the Constant Position Mobility Model, a node has always the same position and the node is not able to move around. To make mobile clients able to move, another mobility model can be set. In addition to the Constant Position Mobility Model, the mobility models Random Waypoint Mobility Model and Gauss-Markov Mobility Model are implemented within this method. One of these mobility models can be selected for a mobile client with the parameter *mobilityModelId*. How these mobility models work, has already been introduced in section 2.4.3.

The Random Waypoint Mobility Model is the first mobility model which has been implemented for the practical part of this master thesis. In ndnSIM, different settings can be defined for the Random Waypoint Mobility Model:

- Movement boundary
- Speed
- Pause

For each mobile client node, a movement area with predefined boundaries is set. Mobile client nodes move with a predefined *speed*, which is a random variable between a minimum and a maximum speed value, until a target point is reached. After reaching the target point, mobile client nodes *pause* for a random time, before the next randomly chosen target point is selected. Then they start moving again with a random *speed*.

The Gauss-Markov Mobility Model is another mobility model which has been implemented within this master thesis. The following settings can be defined in ndnSIM for the Gauss-Markov Mobility Model:

- Movement boundary
- Time step
- Alpha
- Mean velocity
- Mean direction
- Mean pitch

- Normal velocity
- Normal direction
- Normal pitch

For each mobile client node, a movement area with predefined boundaries is set as well. With the Gauss-Markov Mobility Model, mobile client nodes walk for a predefined *time step* in a random direction, with a random speed and a random pitch angle, before they change direction, speed, and pitch angle. The *time step* is a predefined constant. A constant *alpha* between zero and one defines the randomness of the Gauss-Markov Mobility Model. This means, when *alpha* is zero, then the calculation of the new direction, speed, and pitch angle values are completely random. Furthermore, when alpha is one, then the values always consider previous direction, speed, and pitch angle values. When alpha is a value between 0 and 1 - for example 0.3 - then the calculation of the new direction, speed, and pitch angle values are not completely random and the calculation of the new values considers previous direction, speed, and pitch angle values as well. In the following, the implementation of adding WiFi networks to the randomly generated network is described in detail.

4.4.3 Adding WiFi Networks

To create and simulate WiFi networks in ndnSIM, wireless links can be added between available nodes in the created network. These wireless links are added between access point nodes and mobile client nodes. Settings can be added for these wireless links, to define parameters for the WiFi network. For instance, the preferred WiFi standard or the SSID of the WiFi network can be defined.

The method *createWiFiLink* in the network generator creates a wireless link between a selected mobile client node and a selected access point node with the following parameters:

- Ptr<Node> client
- Ptr<Node> ap
- enum WifiPhyStandard wifiStandard
- std::string phyMode
- std::string wifiSsid
- double logDistanceExponent

The parameter *client* is a pointer to a mobile client node in the network and the parameter *ap* is a pointer to an access point node in the network. These two parameters represent the nodes, which will be connected with a wireless link, to simulate a wireless link from a WiFi network. The parameter *wifiStandard* sets the preferred WiFi standard for the WiFi network. Possible WiFi standards in ndnSIM are the 802.11a, 802.11b, or 802.11g standards. The parameter *phyMode* sets the data rate of the WiFi network and with the parameter *wifiSsid* the SSID name of the WiFi network can be set. To evaluate the performance of the new forwarding strategy with an evaluation scenario, loss-prone wireless links are required in the network. For each wireless link, a propagation loss model can be set to simulate loss-prone wireless links in the network. The Log Distance Propagation Loss Model has already been introduced in section 2.4.2. This propagation loss model is implemented on wireless links in the network. Wireless links can use the Log Distance Propagation Loss Model with a predefined parameter *logDistanceExponent*.

With the method *createWiFiLink*, a wireless link of a WiFi network is created between a selected mobile client node and a selected access point node. With this method, further wireless links for the same WiFi network can be created between nodes in the network. When a simulation scenario contains more than one WiFi network, then WiFi links of other WiFi networks can also be created with this method. To create a new wireless link of another WiFi network, a new SSID name can be set for the wireless link. This can be done with the parameter *wifiSsid*. When the SSID name does not exist yet in the network, then a wireless link with a new WiFi network is created.

Due to the implementation of the network generator extensions, additional changes have to be implemented in ndnSIM. Therefore, extensions in the Forwarding Information Base entry and in the layer 3 protocol are required. The implementation of these extensions is described in the following in detail.

4.5 Forwarding Information Base Entry and Layer 3 Protocol Extensions

In this master thesis, mobile client nodes are listening to voicemail messages of fixed client nodes. Therefore, all possible routes are calculated, to be able to forward Interest Packets from mobile client nodes to fixed client nodes within the created generated network.

In the Forwarding Information Base, all possible routes for a predefined prefix are stored on each node, but some of these routes are impractical. Examples of impractical routes are described in the following. Therefore, these impractical routes have to be deleted from the Forwarding Information Base after the network has been created. This means that those specific nodes cannot forward further Interest Packets to specific faces.

An example of an impractical route is when a mobile client node sends out an Interest Packet to an access point node. Then, the access point node can forward the Interest Packet to one of the available routes, using the Best-Route Forwarding Strategy. This means the access point node can forward the Interest Packet to the network nodes it is connected to, or to another mobile client node. It should not be possible for the access point node to forward the Interest Packet to another mobile client node. Therefore, all routes from access point nodes to all mobile client nodes have to be deleted out of the Forwarding Information Base. That means it should not be possible that a mobile client node sends an Interest Packet to an access point node and this access point node sends this Interest Packet to another mobile client node instead of sending this Interest Packet to the network.

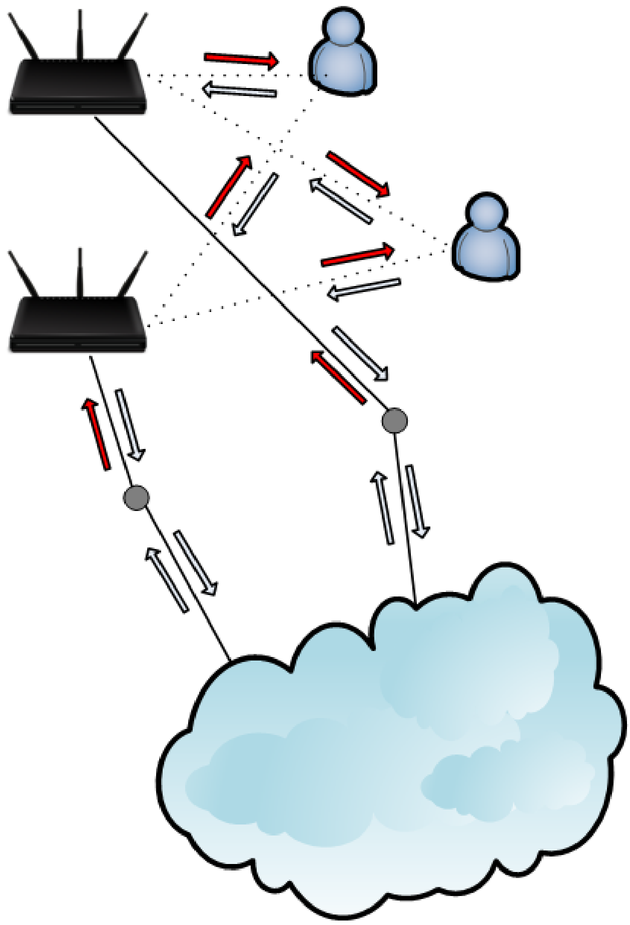

Fig. 4.5: Forwarding Information Base changes: The red marked routes are deleted from the Forwarding Information Base to avoid forwarding Interest Packets on these routes. This means entries from access point nodes to mobile client nodes and entries from network nodes to access point nodes are deleted from the Forwarding Information Base.

Another unwanted possible route is when a mobile client node sends out an Interest Packet to an access point node. Then, the access point node can forward the Interest Packet to one of the available routes as well. This means the access point node can forward the Interest Packet to the network nodes it is connected to. The routes from an access point node to a mobile client node cannot be chosen, because these impractical routes have already been deleted. After the Interest Packet is forwarded from an access point node to a network node, then the network node can choose where to forward the Interest Packet as well. This means the network node can forward the Interest Packet to another node in the network, or to an access point node. It should not be possible for the network node to forward the Interest Packet to an access point node. This is because when an access point node receives an Interest Packet from a network node, then the access point node is not able to forward the Interest Packet to a mobile client node. The Interest Packet can only be forwarded to the network node where the Interest Packet comes from, and this is why loops during forwarding an Interest Packet can occur. Therefore, the routes from network nodes to access point nodes have to be deleted as well, to avoid loops. The impractical routes which are deleted in the Forwarding Information Base are shown in figure 4.5. In this figure, we can see the red marked routes which are deleted from the Forwarding Information Base.

Methods for removing routes from a Forwarding Information Base are provided by ndnSIM, but these methods are not compatible with wireless links. This means these methods can only be used for connections between nodes where the connection is no wireless link.

To remove also routes where two nodes are connected with a wireless link, new methods have been implemented for the practical part of this master thesis. Therefore, ndnSIM has been extended within this master thesis to be able to remove routes where nodes are connected with wireless links.

The implementation can be found on the GitHub repository [Berg17], which is implemented in the following files:

- Filenames: *ndn-fib-helper.hpp* and *ndn-fib-helper.cpp*
- Location of the files: *ndnSIM_2-3/voicemail_scenario/extern*
- These files replace the files in ndnSIM: *ndnSIM_2-3/ns-3/src/ndnSIM/helper/ndn-fib-helper.hpp* and *ndnSIM_2-3/ns-3/src/ndnSIM/helper/ndn-fib-helper.cpp*

To solve the problem of forwarding an Interest Packet to a wrong node, methods for removing routes from the Forwarding Information Base are implemented. To be able to delete some routes out of the Forwarding Information Base, two methods are added to the class *ndn-fib-helper*:

- RemoveRoutesToMobileClients
- RemoveRoutesToAPs

The method *RemoveRoutesToMobileClients* removes routes between access point nodes and mobile client nodes. This method has the following parameters:

- Ptr<Node> node
- const Name& prefix

The parameter *node* is the node in the network on which Forwarding Information Base entries will be deleted. Routes can only be deleted for a specific prefix. Therefore, the parameter *prefix* contains the name of the prefix for which routes have to be deleted.

At first, the randomly generated network which contains network nodes is created for the evaluation scenario. Access point nodes are created after the randomly generated network has been created. When adding an access point node, a link between one random network node and the access point is added. This means, the first face of the access point node is always the node in the network to which network node the access point node is connected to. After this, mobile client nodes are created and links from mobile client nodes to access point nodes are added. This means that all faces, except the first face, has to be deleted from the Forwarding Information Base entry on access point nodes. Therefore, all faces to mobile client nodes for a specific prefix are deleted on access point nodes, by removing entries in the Forwarding Information Base in the method *RemoveRoutesToMobileClients*.

The method *RemoveRoutesToAPs* removes routes between router nodes and access point nodes. This method is similar to the method *RemoveRoutesToMobileClients* and has the following parameters:

- Ptr<Node> node
- const Name& prefix
- uint32_t numberOfRoutes

The parameter *node* is the node in the network on which Forwarding Information Base entries will be deleted. Routes can only be deleted for a specific prefix. Therefore, the parameter *prefix* contains the name of the prefix for which routes have to be deleted. Finally, the parameter *numberOfRoutes* describes the number of routes which have to be deleted on the node.

The method *RemoveRoutesToAPs* removes routes from router nodes which are connected to access point nodes. It can also be possible that a network node is connected to more than one access point node. This means, routes from a router node to all connected access point nodes are deleted.

When a router node is connected to one access point node, then the last face on this router node is the face to the access point node. This is because, at first, router nodes are created, when creating the randomly generated network. After this, access point nodes are created and connections from access point nodes to router nodes are added. Therefore, the last *numberOfRoutes* routes have to be deleted from the Forwarding Information Base of a router node, where *numberOfRoutes* is the number of access points to which the current router node is connected to.

The methods *RemoveRoutesToMobileClients* and *RemoveRoutesToAPs* need to know the available faces on a specific node on which routes can be deleted. Therefore, ndnSIM has to be extended in the scope of this master thesis, to be able to get a list of all available faces on a specific node in the network.

The implementation of getting a list of all available faces on a node which can be deleted, can be found on the GitHub repository [Berg17] and this is implemented in the following files:

- Filenames: *ndn-l3-protocol.hpp* and *ndn-l3-protocol.cpp*
- Location of the files: *ndnSIM_2-3/voicemail_scenario/extern*

- These files replace the files in ndnSIM: *ndnSIM_2-3/ns-3/src/ndnSIM/model/ndn-l3-protocol.hpp* and *ndnSIM_2-3/ns-3/src/ndnSIM/model/ndn-l3-protocol.cpp*

Therefore, the following method has been implemented in the class *ndn-l3-protocol*:

- getRemovableFaces

In the method *getRemovableFaces*, all available faces on a node are retrieved and a set of available faces for a node is returned in a vector. This means all faces for the current node where the face ID of the face is greater than 255, are returned. This is because all face IDs from faces with connections to other nodes in the network start at 256 as face ID. Other faces of a node, which are not connected to another node in the network, have a face ID smaller than 256.

In NDN, Interest Packets are used by a content consumer to request a preferred content at a content provider. Data Packets are the responses to Interest Packets, which are sent out by a content provider. Data Packets deliver the requested content. When no Data Packet to an Interest Packet has been received by the content consumer after a predefined time, then the Interest Packet is sent out again to request the content again. Sending Interest Packets again, when the Data Packet to the corresponding Interest Packet is not received, is not relevant for the evaluation scenario of this master thesis. This is because Data Packets corresponding to retransmitted Interest Packets may be arriving too late at the content consumer which is described in the following section in more detail. Therefore, further changes have to be made in a consumer application of ndnSIM. These changes are described in the following.

4.6 Consumer Application Extensions

A consumer application in ndnSIM defines how content consumers can send out Interest Packets. Therefore, ndnSIM provides different consumer applications. A list of predefined consumer applications is provided on the official site of ndnSIM [LosA17].

An example of a consumer application in ndnSIM is the *ConsumerCbr* application, where several Interest Packets can be sent out with a constant frequency. For instance, a predefined number of Interest Packets is sent out each second with the *ConsumerCbr* application. Another consumer application provided by ndnSIM is called *ConsumerBatches*, where a predefined number of Interest Packets can be sent out at a predefined time.

In this master thesis, the *ConsumerCbr* application is used for the evaluation scenario. Content consumers request data by sending Interest Packets. Each Interest Packet has a timer, called a retransmission timer. This timer is started when sending out an Interest Packet. When no Data Packet arrived for a corresponding Interest Packet after the transmission timer expires, then a new Interest Packet for requesting the preferred data will be sent out by the consumer application. But retransmissions are not relevant for the evaluation scenario for this master thesis where mobile clients are listening to voicemail messages. This is because Data Packets corresponding to retransmitted Interest Packets may be arriving too late at the content consumer. When a Data Packet arrives too late at a content consumer, then this Data Packet may not be needed anymore. Mobile clients will not stop listening to voicemail messages when some Data Packets are not received punctually. This means, when Data Packets from already listened parts of the voicemail message are received, then mobile clients will not use these Data Packets anymore.

When configuring the consumer application in a simulation scenario in ndnSIM, disabling retransmissions with an optional parameter is not possible. When using the *ConsumerCbr* application, some changes have to be made in the superclass *Consumer* to disable retransmissions.

The implementation of disabling retransmissions of Interest Packets can be found on the GitHub repository [Berg17] and this is implemented in the following file:

- Filename: *ndn-consumer.cpp*
- Location of the file: *ndnSIM_2-3/voicemail_scenario/extern*
- This file replaces the file in ndnSIM: *ndnSIM_2-3/ns-3/src/ndnSIM/apps/ndn-consumer.cpp*

Therefore, the changes have been implemented in the following method:

- OnTimeout

The fields of Interest Packets are already described in section 2.2.1 where it is described that when two Interest Packets have the same *Content Name* and the same *Selector*, then these two Interest Packets request the same content. This means retransmitted Interest Packets have the same *Selector* where a sequence number for an Interest Packet is stored. In the class *ConsumerCbr*, the method *OnTimeout* is called when a retransmission timer for a sequence number of an Interest Packet times out before the corresponding Data Packet arrives. Then, a new Interest Packet with this sequence number is sent out again. The code where retransmissions are sent out has to be disabled in the consumer application. Therefore, to be able to disable retransmissions, the following lines have to be removed in the *OnTimeout* method:

```
m_rtt->IncreaseMultiplier();
m_rtt->SentSeq(SequenceNumber32(sequenceNumber), 1);
m_retxSeqs.insert(sequenceNumber);
```

After removing these lines, further Interest Packets will not be retransmitted when the requested data has not been received before the retransmission timer of an Interest Packet expires. The evaluation scenario for the performance evaluation uses the *ConsumerCbr* consumer application. So this means that retransmissions are disabled for the evaluation scenario. This evaluation scenario is created for the performance evaluation of the practical part of this master thesis. When running simulation scenarios for the performance evaluation, output files with packet statistics are created for each simulation run. With the help of these output files, the simulation scenario results can be evaluated. For the performance evaluation, different metrics are used, which will be described in chapter

5 in more detail. Therefore, evaluation scripts are implemented, which use the output files with packet statistics to evaluate simulation scenario results. In the following, the created output files and the implementation of the evaluation scripts for the performance evaluation are described in detail.

4.7 Simulation Output and Evaluation Scripts

In ndnSIM, output files can be generated by using trace helpers, to obtain metrics out of a simulation scenario. In this section, available trace helpers in ndnSIM are introduced, which are used for the performance evaluation of this master thesis. The provided trace helpers which can be used in simulation scenarios can be found on the official site of ndnSIM [LosA17]. Finally, evaluation scripts are introduced which are implemented for this master thesis to obtain and calculate metrics from output files of performed simulation scenarios, as well as to draw charts with results from the performance evaluation. The evaluation scripts for the performance evaluation are implemented in Python. In the following, the used trace helpers are introduced.

4.7.1 Trace Files

Different trace helpers are provided by ndnSIM. The provided trace helpers can be found on the official site of ndnSIM [LosA17]. The *Rate Trace* and the *App Delay Trace* are used for generating simulation scenario output files for the evaluation scenario. These trace helpers are described in the following.

By using the *Rate Trace*, the simulation scenario creates an output file where packet statistics from the simulation scenario, like the number of packets or the number of packet bytes, are logged. This trace helper obtains data from the simulation scenario with the following columns:

- Time
- Node
- FaceId
- FaceDescr
- Type
- Packets
- Kilobytes
- PacketRaw
- KilobytesRaw

The column *Time* represents the current simulation time in seconds. The column *Node* is the name of the node in the network. This name is also present in the visualizer of ndnSIM. The column *faceId* is the ID of the face of the current node where packet statistics are logged and *FaceDescr* is a description of the current face. When *FaceDescr* starts with *netdev*, then the face of the current node is no internal face, but it is a face to another node in the network. The column *Type* specifies what is logged in the current line. For instance, outgoing Interest Packets or incoming Data Packets can be logged for the simulation scenario. The columns *Packets* and *PacketRaw* log the number of packets of the specific *Type* where *PacketRaw* is the total number of packets at a specific simulation time which is defined in the *Time* column. Finally, the columns *Kilobytes* and *KilobytesRaw* log the number of packet bytes.

Another trace helper in ndnSIM is called *App Delay Trace*. This trace helper generates also an output file where packet statistics from a simulation scenario, like the delay, the number of retransmitted Interest Packets, or the number of hops in the network are logged. This trace helper obtains data from the simulation scenario with the following columns:

- Time
- Node
- AppId
- SeqNo
- Type
- DelayS
- DelayUS
- RetxCount
- HopCount

The column *Time* represents the current simulation time in seconds when a Data Packet corresponding to an Interest Packet with the sequence number *SeqNo* is received. The column *Node* is the name of the node in the network, which is also present in the visualizer of ndnSIM. *SeqNo* contains the sequence number of the Interest Packet. The column *Type* describes the type of delay. Possible values for the type of delay are *LastDelay* and *FullDelay*. *FullDelay* means the delay between sending an Interest Packet for requesting specific data and receiving the corresponding Data Packet. When the data to an Interest Packet is not received before the retransmission timer expires, then regularly an Interest Packet is sent out again. Therefore, *LastDelay* means the delay between sending the last Interest Packet for requesting specific data and receiving the corresponding Data Packet. Before receiving the data, this last sent Interest Packet can be a retransmission of an Interest Packet. Retransmissions are disabled for our evaluation scenario. Therefore, *LastDelay* and *FullDelay* are always the same in the output files of the evaluation scenario. The column *DelayS* represents the delay in seconds from sending an Interest Packet to receiving the corresponding Data Packet and *DelayUS* is the delay expressed in microseconds. The column *RetxCount* is the number of Interest Packet retransmissions for the sequence number. Finally, *HopCount* means the number of network hops traversed while transmitting a Data Packet from a content producer or a network cache to the content consumer.

Output files of a simulation scenario are created in an output folder when using these trace helpers. To obtain metrics from these output files and to show the results of the simulation scenario, Python scripts are implemented. These Python scripts are used for the performance evaluation of the simulation scenario for this master thesis. In the following, the implemented Python scripts are described in detail.

4.7.2 Python Scripts for the Evaluation

To evaluate the results of the implementation part of this master thesis, Python scripts have been implemented. With the help of these scripts, a predefined number of simulation scenario runs can be executed. Furthermore, metrics are obtained and calculated out of simulation scenario output files of a simulation scenario run. Finally, bar charts are created with the evaluation results.

These Python scripts can be found on the GitHub repository [Berg17] in the following folder:

- Location of the files: *ndnSIM_2-3/voicemail_scenario/pythonScripts*
- Guideline for running these Python scripts: *ndnSIM_2-3/voicemail_scenario /README.pdf*

A guideline how to run these Python scripts with specific command-line arguments can also be found on the GitHub repository. The following Python scripts are used for the performance evaluation of this master thesis:

- executeSimulations.py
- calculateMetrics.py
- executeCalculateMetrics.py
- calculateIsrResults.py
- calculateResults.py
- createBarChart.py

At first, the Python script *executeSimulations.py* is described in more detail. To execute a predefined number of simulation scenario runs from an evaluation scenario in ndnSIM, the Python script *executeSimulations.py* can be used. This Python script has been implemented by the ITEC Multimedia Communication group [MoPH17, PoRH17a] and has been adapted and extended within this master thesis for the current needs. This Python script starts a predefined number of simulation scenario runs for each possible parameter configuration. These possible parameter configurations are defined within the Python script.

When executing this Python script, the following command-line arguments can be used for the execution of simulation scenarios:

- -t
- -r
- -s

where *-t* is the number of threads which are used for the execution. This means parallel execution of simulation scenario runs is possible with this argument. The command-line argument *-r* specifies the number of simulation scenario execution runs for each possible parameter configuration. Finally, the command-line argument *-s* specifies a start number of runs. This argument is used for skipping a certain number of runs, but this argument is not needed for the performance evaluation. Therefore, the argument *-s* is by default zero. This argument is useful when continuing with the execution of simulation scenario runs at a later time. For instance, when 3 runs have already been executed, then 3 runs can be skipped by using the argument *-s* and the execution will start with run 4.

Furthermore, possible parameter configurations are defined within the Python script. For the evaluation of this master thesis the following parameters can vary:

- The used forwarding strategy for mobile client nodes
- The number of access point nodes per WiFi network
- The distances between access point nodes for the visualization in the visualizer of ndnSIM

When the Mobile-Client Forwarding Strategy is chosen as forwarding strategy for mobile client nodes, then the Interest Satisfaction Ratio threshold value for this forwarding strategy can also vary. But the Interest Satisfaction Ratio threshold value is not configurable within this Python script. This means, that the Interest Satisfaction Ratio threshold value is defined as constant in the script and is not configurable with a parameter when executing this script. By default, the Interest Satisfaction Ratio threshold value is always 90. When executing simulation scenario runs by using another threshold value, then the hardcoded value for the Interest Satisfaction Ratio threshold value can be changed to a preferred value. Therefore, the constant value can be changed in the Python script before running this script. This is only needed when comparing different threshold values to find a threshold value that works well for the Mobile-Client Forwarding Strategy. When comparing the Mobile-Client Forwarding Strategy with different forwarding strategies then the constant value from the script is used.

When executing the *executeSimulations.py* Python script, an output folder is created for every possible parameter configuration defined within the Python script. The folder name of this output folder contains the used parameters. Furthermore, a predefined number of simulation scenario runs is executed for each possible parameter configuration for a simulation scenario. This means subfolders for each simulation scenario run are created.

Folder names of subfolders contain the current number of the run. The output files of specific simulation scenario runs are stored in these subfolders.

The following output files are produced for each simulation scenario run:

- *app-delays-trace.txt*
- *callLengths.txt*
- *rate-trace.txt*
- A standard output file, e.g. *t_0.stdout.txt*

The files *app-delays-trace.txt*, *callLengths.txt* and *rate-trace.txt* are created within the simulation scenario for the performance evaluation. The *callLengths.txt* file is described in chapter 5 in more detail. Furthermore, the standard output file is created while performing the current run when executing this Python script. This file holds the thread output of the simulation scenario run execution.

After this, output files from a simulation scenario can be evaluated with the Python script *calculateMetrics.py* to retrieve metrics from a simulation scenario run. When this Python script is executed in a folder where output files from a simulation scenario run are stored, then no command-line argument is needed for the execution. Otherwise, the path to a folder where output files are stored has to be passed as a command-line argument.

A number of metrics is obtained and calculated from a simulation scenario run within this Python script. The most important metrics for the performance evaluation of a simulation scenario are introduced in section 5.3. These metrics are obtained and calculated in this Python script. The Interest Satisfaction Ratio value is an example of a used metric. In the simulation scenario for the performance evaluation, each mobile client node is listening to a voicemail message produced by a fixed client node. This means, for instance, mobile client node 1 communicates with fixed client node 1, and mobile client node 2 communicates with fixed client node 2. A prefix for each communication pair is set. For instance, when mobile client node 1 listens to a voicemail message produced by fixed client node 1, then the prefix */voicemail_1* is used for Interest Packets and Data Packets during data transfer. Within the Python script, metrics are calculated for each voicemail call separately. Subsequently, the average values for all metrics are calculated from all voicemail calls, and for further usage these values are written into a new average results output file. This means, after executing this Python script, the following file is created in the output folder, where average values for all calculated metrics are stored:

- *averageValues.txt*

The script *executeCalculateMetrics.py* can be executed instead of executing the Python script *calculateMetrics.py* to obtain and calculate metrics not only for one simulation scenario run, but also for all simulation scenario runs for a specific parameter configuration. This Python script can be used to calculate metrics from output folders from all available simulation scenario runs for a possible parameter configuration at the same time. Within the script *executeCalculateMetrics.py*, the script *calculateMetrics.py* is executed for all simulation scenario output folders of a parent folder.

When executing this Python script, the following command-line arguments can be used for the execution:

- -noRuns
- -dir

where *-noRuns* is the number of simulation scenario runs which are stored in the parent folder. The command-line argument *-dir* is the name of the parent folder where all simulation scenario output files for all simulation scenario runs of a specific parameter configuration are stored. After executing this Python script, the *averageValues.txt* file is created in each output folder as well.

The Python script *calculateIsrResult.py* calculates results for the Mobile-Client Forwarding Strategy where different Interest Satisfaction Ratio threshold values are considered and compared.

When executing this Python script, the following command-line arguments have to be used for the execution:

- -dir10
- -dir20
- -dir30
- -dir40
- -dir50
- -dir60
- -dir70
- -dir80
- -dir90
- -dstDir

where *-dir10*, *-dir20*, *-dir30*, *-dir40*, *-dir50*, *-dir60*, *-dir70*, *-dir80* and *-dir90* are folder names for specific parameter configurations with simulation scenario output folders. Each folder holds all simulation scenario output files for all simulation scenario runs of a specific parameter configuration. These output folders hold output files from different Interest Satisfaction Ratio threshold values. For instance, this means the output folder *-dir10* holds output files from simulation scenario runs where the Interest Satisfaction Ratio threshold value is set to 10.

Each output folder holds subfolders for a predefined number of simulation scenario runs. The file *averageValues.txt* is located in each subfolder where a simulation scenario run is stored. In this Python script, the average values from all average files for a configuration are calculated. The confidence intervals of 95% are calculated as well.

Furthermore, the command-line argument *-dstDir* specifies the folder name where created result files will be saved. After calculation, created bar chart files are stored in this folder where a bar chart file holds 9 bars and shows the results of a specific metric. This means, one bar represents one configuration where a specific Interest Satisfaction Ratio threshold value is considered.

The Python script *calculateResults.py* calculates also metrics from simulation output files. Different forwarding strategies are compared within this Python script. When executing this Python script, the following command-line arguments have to be used for the execution:

- -bDir
- -mDir
- -cDir
- -dstDir

where *-bDir*, *-mDir* and *-cDir* are folder names of simulation scenario output folders. Each folder holds all simulation scenario output files for all simulation scenario runs of a specific parameter configuration. These output folders hold output files for different forwarding strategies. For instance, this means the output folder *-bDir* holds output files from simulation scenario runs where the Best-Route Forwarding Strategy is used, *-mDir* uses the Multicast Forwarding Strategy and *-cDir* uses the Mobile-Client Forwarding Strategy.

In this Python script, the average values from all average files of a configuration are calculated, like this is done in the Python script *calculateIsrResult.py*. The confidence intervals of 95% are calculated as well.

Furthermore, the command-line argument *-dstDir* specifies the folder name where created result files will be saved as well. After calculation, created bar chart files are stored in this folder. For instance, bar charts are created where a bar chart file holds 3 bars. This means one bar represents one configuration where a specific forwarding strategy is used.

Finally, the Python script *createBarChart.py* provides methods for creating bar charts and storing them into files. These methods are used in the Python scripts *calculateIsrResult.py* and *calculateResults.py*. The following methods are implemented in this Python script:

- createBarChartCompareOneMetric9Bars
- createBarChartCompareOneMetric3Bars
- createBarChartComparePacketsOneMetric4Bars
- createAndSaveBarChartTwoMetrics3Bars
- createAndSaveBarChartThreeMetrics3Bars

In the next chapter, the performance evaluation of the practical part of this master thesis is described in detail. For the performance evaluation of the newly implemented Mobile-Client Forwarding Strategy, an evaluation scenario has been created and implemented for the ndnSIM simulator. This evaluation scenario is introduced in the following chapter. Furthermore, used metrics for the performance evaluation are introduced as well. Finally, the results of the performance evaluation are shown.

5 Performance Evaluation of the Mobile-Client Forwarding Strategy

In this chapter, the results of the performance evaluation of the Mobile-Client Forwarding Strategy are shown. This forwarding strategy has been designed and implemented for the practical part of this master thesis. An evaluation scenario has been defined and implemented for the performance evaluation, to perform network simulations within an NDN network. This evaluation scenario has been created to use it within the ndnSIM simulator. Firstly, the evaluation scenario is described in detail. After this, the used metrics for the performance evaluation are introduced. Furthermore, the results of the Mobile-Client Forwarding Strategy are shown, compared with different Interest Satisfaction Ratio threshold values. Different Interest Satisfaction Ratio threshold values are compared, to determine an Interest Satisfaction Ratio threshold value which works well for the Mobile-Client Forwarding Strategy as a forwarding strategy for mobile client nodes. Moreover, the results of the Mobile-Client Forwarding Strategy with a predefined Interest Satisfaction Ratio threshold value are compared with the results of other forwarding strategies. Therefore, the Mobile-Client Forwarding Strategy is compared with the forwarding strategies Best-Route and Multicast. In the evaluation scenario, the compared forwarding strategies are always installed on mobile client nodes. All other nodes in the network use a default forwarding strategy. This is done to show that the Mobile-Client Forwarding Strategy works well when client nodes can walk in an NDN network. In the following, the evaluation scenario is described in detail.

5.1 Evaluation Scenario

In this section, the evaluation scenario for the performance evaluation in ndnSIM is described in detail. The results of the performance evaluation are compared with different approaches. The forwarding strategies Mobile-Client, Best-Route, and Multicast are compared as forwarding strategies for mobile client nodes. These forwarding strategies are compared to show that the Mobile-Client Forwarding Strategy has advantages over the other forwarding strategies. Therefore, an evaluation scenario named *big-mobile-voicemail* has been implemented for the performance evaluation of the practical part of this master thesis. The implementation of this evaluation scenario can be found on a GitHub repository [Berg17]:

- Filename: *big-mobile-voicemail.cpp*
- Location of the file: *ndnSIM_2-3/voicemail_scenario/scenarios*

This evaluation scenario provides the following command-line arguments:

- briteConfig
- logDir
- fwStrategy
- noAPs
- xPos
- yPos
- yPosNext
- isrT

The command-line argument *briteConfig* is the path to a configuration file. With the help of this file, a randomly generated network is created using BRITE. The command-line argument *logDir* is the path to a folder where output files from a simulation scenario are stored. A predefined forwarding strategy is used by mobile client nodes within this evaluation scenario. Therefore, the command-line argument *fwStrategy* specifies the forwarding strategy which will be installed on mobile client nodes. When the Mobile-Client Forwarding Strategy is chosen as forwarding strategy for mobile client nodes, then the command-line argument *isrT* can specify the Interest Satisfaction Ratio threshold value for this forwarding strategy.

Furthermore, two WiFi networks are used in the evaluation scenario. Each WiFi network has a predefined number of access point nodes, which is defined by *noAPs*. Two WiFi networks are chosen so that swtiching between different WiFi netwoks is possible for mobile client nodes. Therefore, command-line arguments for defining parameters for access point nodes of WiFi networks are provided as well. Each access point has a specific position in the whole network for the visualization of the evaluation scenario in the visualizer of ndnSIM. This visualizer has already been introduced in section 4.1. The position of the first access point node of the second WiFi network is defined in the command-line arguments *xPos* and *yPos*. The first access point node of the first WiFi network has the same *yPos* position, but a different *xPos* position (-*xPos*). When WiFi networks have more than one access point node, then the command-line argument *yPosNext* specifies the distance in the visualizer of ndnSIM between the previously added access point node to the next added Acces Point node of the same WiFi network. An example is shown in figure 5.1 where access point node positions from two WiFi networks for the visualization in the visualizer of ndnSIM are introduced.

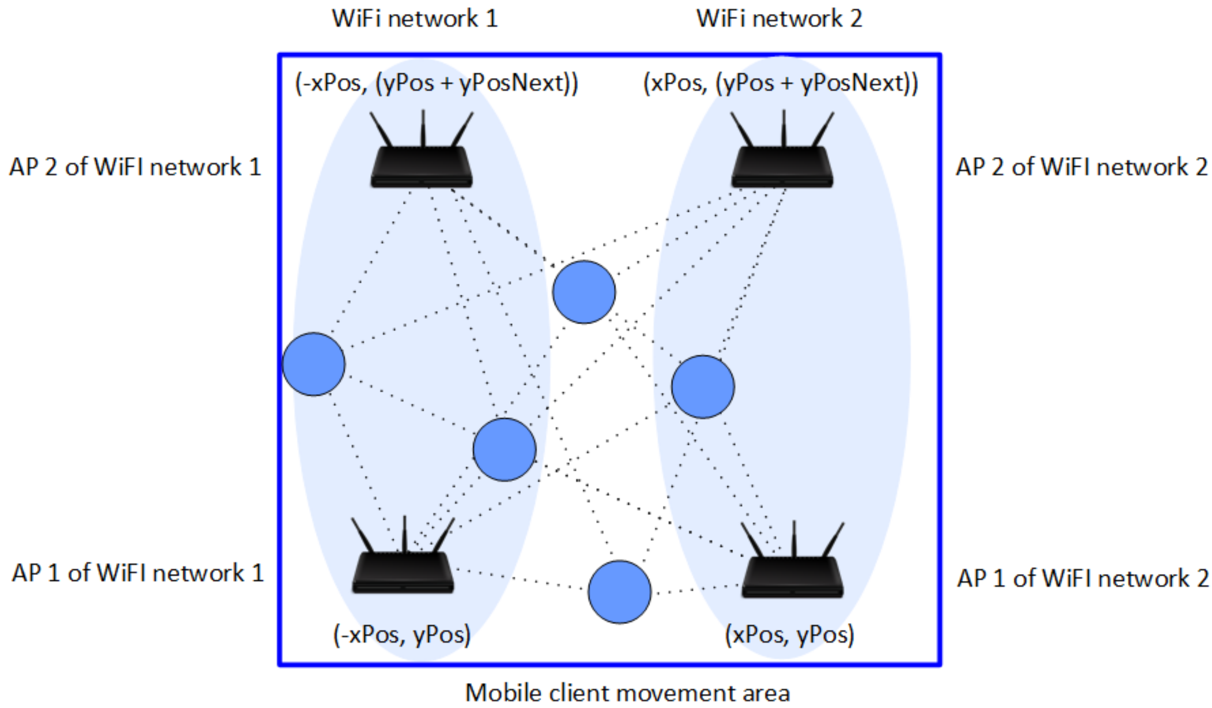

Fig. 5.1: Positions of access point nodes: This figure shows access point node positions from two WiFi networks where access point nodes are added to the network for the visualization in the visualizer of ndnSIM. Furthermore, mobile client nodes and the boundaries for the movement are also shown.

At first, an overview of the evaluation scenario is given. Therefore, general configurations for the evaluation scenario are introduced. After this, the whole network topology of this evaluation scenario is described. The used parameters for creating the whole NDN network are introduced as well. Finally, specific parameters are explained, which are used to simulate mobile client nodes listening to voicemail messages.

5.1.1 General Scenario

Generally, the evaluation scenario consists of several nodes in an NDN network. This NDN network holds network nodes and fixed client nodes, as well as access point nodes of WiFi networks. Moreover, mobile client nodes in this network can connect to existing WiFi networks.

In this evaluation scenario for the performance evaluation, it is simulated that mobile client nodes are listening to voicemail messages. These voicemail messages are created before, by fixed client nodes within the network. Furthermore, specific parameters are used to simulate a realistic evaluation scenario. This means, for instance, the size of packets in the evaluation scenario simulates a real audio codec. Moreover, mobile client nodes are using mobility models to simulate moving around while listening to voicemail messages. Finally, a loss model is added to wireless links, to simulate loss-prone wireless links. For instance, a loss-prone wireless link can occur when a mobile client node moves away from an access point node of a WiFi network. In the following, the whole network topology for the evaluation scenario is described in detail.

5.1.2 Scenario Network Topology

The network topology of the evaluation scenario is generated randomly with a network topology generator named BRITE [MeMB00]. Therefore, a network generator using BRITE has been implemented by the ITEC Multimedia Communications group [MoPH17, PoRH17a]. Furthermore, to create a randomly generated NDN network with network nodes, fixed client nodes, access point nodes for WiFi networks, and mobile client nodes, this network generator has been extended within the scope of this master thesis. This has already been described in section 4.4 in detail. In the following, the parameters in the evaluation scenario are introduced, which are used for creating the randomly generated NDN network.

At first, the evaluation scenario needs a BRITE configuration file, where used parameters are defined. This configuration file consists of different sections where each section starts with *BeginModel* and ends with *EndModel*. The field *Name* in each section defines the type of network components where parameters are defined. Therefore, nodes in the network are defined in the section where $Name = 2$ is configured and autonomous systems are defined in the section where $Name = 4$ is configured in the BRITE configuration file. The Barabasi Albert model is used for generating the random network components within this configuration file [PoRH17a]. Furthermore, three autonomous systems ($\mu = 3$) are generated randomly for the evaluation scenario. Each autonomous system consists of 20 nodes ($\nu = 20$). The number of autonomous systems is defined in the field N with $N = 3$ ($\mu = 3$) in the section where autonomous systems are configured ($Name = 4$) and the number of nodes within an autonomous system is defined in the field N with $N = 20$ ($\nu = 20$) in the section where nodes in the network are configured ($Name = 2$) in the BRITE configuration file. Nodes within an autonomous system represent network nodes within the network topology. By default, the network generator generates a network with no redundant paths. This means each newly added node has only one neighboring node. The number of neighboring nodes for a new node is also defined in the BRITE configuration file in the field m ($m = 1$).

Posch et al. define topology bandwidth variants [PoRH17a]. For this evaluation scenario, a medium bandwidth is chosen. A medium bandwidth means links have a random link capacity between 3 and 5 Mbps, when nodes of different autonomous systems are connected. This is defined in the BRITE configuration file ($BWInterMin = 3000$ and $BWInterMax = 5000$). Furthermore, links have a random link capacity between 2 and 4 Mbps when nodes within the same autonomous system are connected. This is defined in the BRITE configuration file ($BWIntraMin = 2000$ and $BWIntraMax = 4000$) as well.

When using BRITE, only a predefined number of neighbors for each node can be defined, but it is not possible to add several random links to the created network. Therefore, several various connections are added between nodes within the same autonomous systems. Connections between nodes of different autonomous systems are added as well. Additional connections in the network are added as well, after the randomly generated network has been created. This will be done to get a better connectivity between nodes in the created network.

Posch et al. define topology connectivity variants [PoRH17a] to specify the connectivity between nodes in a network, such that additional connections are added to this network. For this evaluation scenario, a medium connectivity is chosen. A medium connectivity means, when the number of autonomous systems is μ, then μ links are added randomly between nodes of different autonomous systems. The number of autonomous systems is defined with $\mu = 3$ for the evaluation scenario. This means, three links are added to the network between nodes of different autonomous systems. Furthermore, when the number of nodes per autonomous system is ν, then $\nu/2$ links are added randomly between nodes within an autonomous system, which is performed for each autonomous system. The number of nodes per autonomous system is defined with $\nu = 20$ for the evaluation scenario. This means, 10 links are added to the network in each autonomous system.

Furthermore, a medium bandwidth is also chosen for these additionally added connections [PoRH17a]. This means, a random link capacity between 3 and 5 Mbps for connections between nodes of different autonomous systems is selected. And a random link capacity between 2 and 4 Mbps for connections between nodes within the same autonomous system is chosen. All network links have a network delay between 5 and 15 ms.

Before performing simulations with the created evaluation scenario, other network components have to be added to the network topology as well. In this evaluation scenario, a mobile client node listens to a voicemail message, created by a fixed client node. Therefore, fixed client nodes, access point nodes for WiFi networks, and mobile client nodes have to be added to the network topology as well. This is done after the additional links have been added to the network. Furthermore, wireless links are added as well to connect mobile client nodes with access point nodes. The used parameters to add these additional network components are described in the following.

At first, fixed client nodes are added to the created network topology. In this evaluation scenario, 10 fixed client nodes are added to the network. A link from each fixed client node to a randomly chosen network node in a randomly chosen autonomous system is added as well. This link has a capacity of 4 Mbps and a network delay of 5 ms.

A network example is shown in figure 5.2, where three autonomous systems are created. Each autonomous system consists of 20 network nodes. A point in this figure represents a node in the network. Additional links have also been added to this network to reach a medium connectivity in the whole network. This means, three connections between nodes of different autonomous systems are added. Furthermore, 10 connections between nodes within the same autonomous systems are added as well. This is done for each autonomous system. Finally, 10 fixed client nodes are added to this network. The blue points in this figure represent these fixed client nodes.

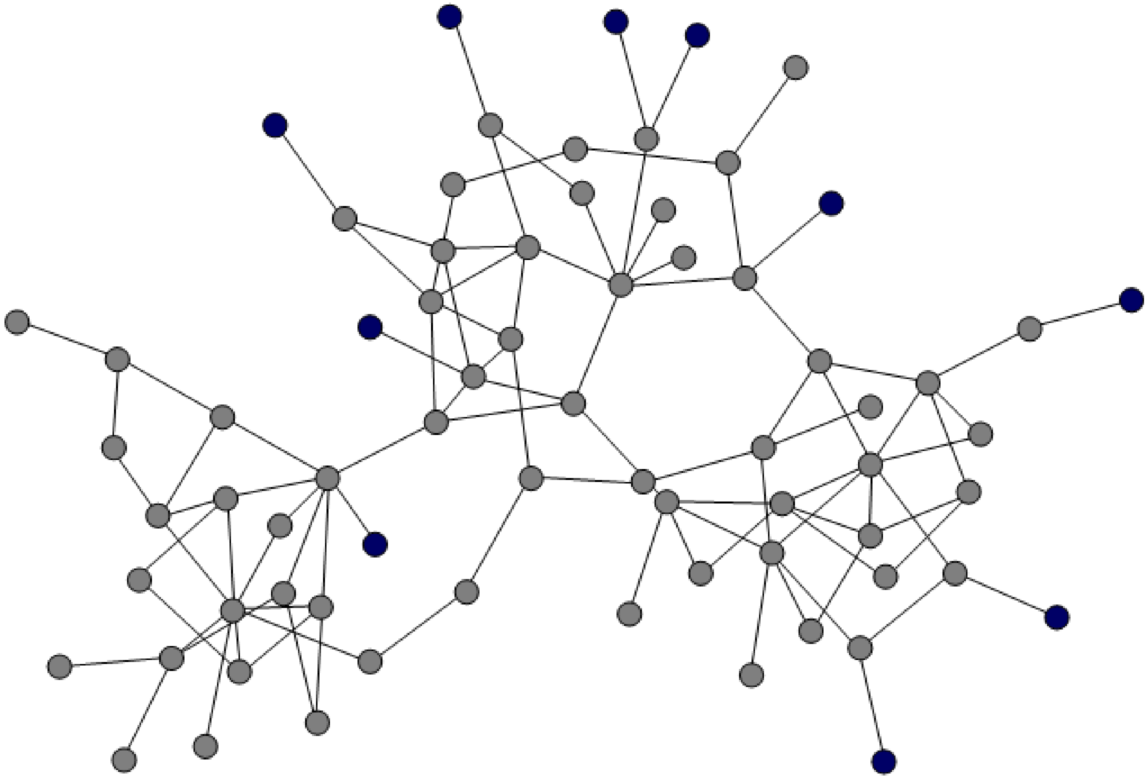

Fig. 5.2: Evaluation scenario (1/3): A randomly created network with three autonomous systems is shown. Each autonomous system consists of 20 network nodes. Additional links are added after the network creation, to reach a medium connectivity. The blue points in this figure represent fixed client nodes.

Furthermore, two WiFi networks are added to the existing network for the evaluation scenario as well. Each WiFi network consists of several WiFi access point nodes. Each WiFi network has one access point node by default. WiFi networks are always placed above the autonomous systems. These positions are used for the visualization of the evaluation scenario in the visualizer of ndnSIM. This means, WiFi networks are placed above all other network components. The first WiFi network is located on the left side and the second WiFi network is located on the right side in the evaluation scenario.

Each access point node is connected to a randomly chosen node from the existing network. Therefore, the access point nodes of the first WiFi network are connected to randomly chosen network nodes of one randomly chosen autonomous system. Furthermore, the access point nodes of the second WiFi network are connected to randomly chosen network nodes from the other two autonomous systems. These links have a capacity of 4 Mbps and a network delay of 5 ms.

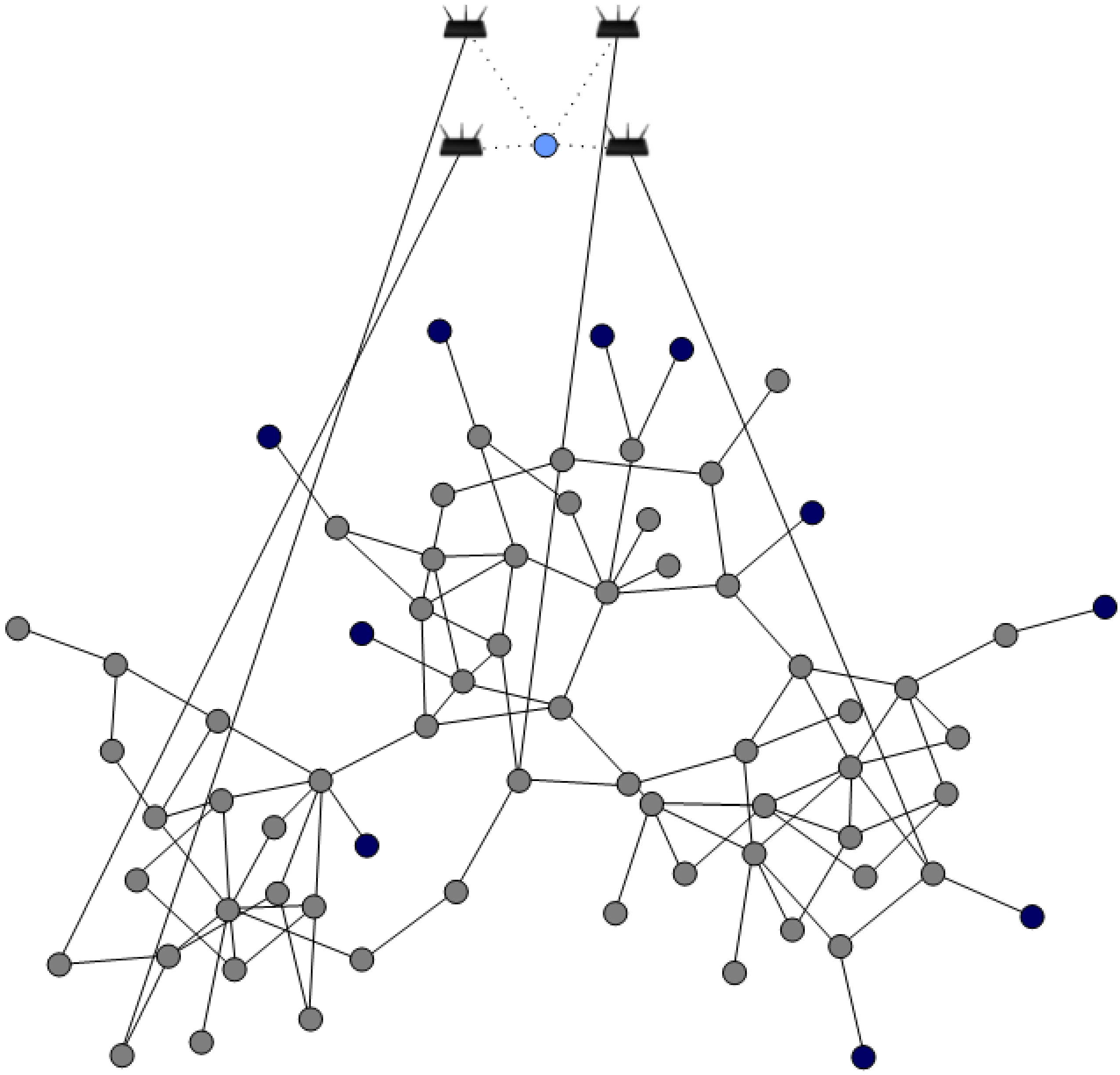

Fig. 5.3: Evaluation scenario (2/3): Additional network components are added to the randomly created network. WiFi access point nodes for WiFi networks are added as well as a mobile client node. The dashed lines represent wireless links between access point nodes and the mobile client node. The light-blue point represents the mobile client node.

Moreover, the distances between two access point nodes for the visualization in the visualizer of ndnSIM can vary. By default, the distance between two access point nodes of the same WiFi network is 40 units in the visualizer of ndnSIM. Furthermore, the distance between two access point nodes of different WiFi networks is 40 units in the visualizer of ndnSIM as well. A greater distance between access point nodes is not recommended. Mobile client nodes can walk between access point nodes. When a distance between two access point nodes is too high and a mobile client node is located between these access point nodes, then it can happen that the mobile client node is not able to connect to one of these access point nodes, depending on the position of the mobile client node.

A network example is shown in figure 5.3, where WiFi access point nodes are added to the existing network to simulate WiFi networks. The light-blue point between the access point nodes in this figure represents a mobile client node. Furthermore, the dashed lines which connect access point nodes with the mobile client node, show wireless connections. In this example, two WiFi networks are displayed, where each WiFi network holds two access point nodes. The access point nodes on the left side belong to the first WiFi network, and the access point nodes on the right side belong to the second WiFi network. The number of access point nodes of a WiFi network can vary in the evaluation scenario as well.

The position of the first access point of the first WiFi network is defined in equations 5.1 and 5.2.

$$xPosFirstWifi = -xPosWiFi \tag{5.1}$$

$$yPosFirstWifi = yPosWiFi \tag{5.2}$$

where the command-line argument *xPos* is used for *xPosWiFi* and the command-line argument *yPos* is used for *yPosWiFi*. Additional access point nodes of a WiFi network have the same x position, but a different y position. Therefore, the command-line argument *yPosNext* is used. Furthermore, the position of the first access point of the second WiFi network is defined in equations 5.3 and 5.4.

$$xPosSecondWifi = xPosWiFi \tag{5.3}$$

$$yPosFirstWifi = yPosWiFi \tag{5.4}$$

After adding the access point nodes to the network, several mobile client nodes are added to the network topology. In this scenario, 10 mobile client nodes are added to the network for the evaluation scenario. Each mobile client node is added at a specific default start position for the visualization in the visualizer of ndnSIM. This start position is the same for each mobile client node, when creating the network and before starting the simulation scenario. This start position is defined in equations 5.5 and 5.6.

$$mobileClientPosX = 0 \tag{5.5}$$

$$mobileClientPosY = yPosWiFi \tag{5.6}$$

For simulating mobility in the evaluation scenario, mobile client nodes can walk inside a predefined movement area during a simulation of the evaluation scenario. When WiFi networks of the evaluation scenario hold one access point node for each WiFi network, then the positions for the movement area for the visualization in the visualizer of ndnSIM are calculated with equations 5.7, 5.8, 5.9 and 5.10.

$$mobileClientMinX = -xPosWiFi - edge \tag{5.7}$$

$$mobileClientMaxX = xPosWiFi + edge \tag{5.8}$$

$$mobileClientMinY = -edge \tag{5.9}$$

$$mobileClientMaxY = edge \tag{5.10}$$

where the value of *edge* is 10. This predefined *edge* value from the evaluation scenario is added so that the movement area boundaries are not exactly at the access point nodes positions for the visualization in the visualizer of ndnSIM, but so that the access point nodes are also inside the movement area of the mobile client nodes. When WiFi networks of the evaluation scenario hold more than one access point node for each WiFi network, then $mobileClientMinY$ for the movement area is calculated with equation 5.11.

$$mobileClientMinY = numberOfAPs * yPosNextAP/2 - edge \tag{5.11}$$

Mobile client nodes start walking when the simulation of the evaluation scenario starts. They walk respecting a specific mobility model. The Gauss-Markov mobility model [Foun17] [CaBD02] is used for this evaluation scenario. The Random Waypoint Mobility Model is also implemented for the practical part of this master thesis but this mobility model is not used for the evaluation scenario.

After adding mobile client nodes, wireless links between mobile client nodes and access point nodes are created. Each mobile client node is connected to each access point node. A specific WiFi standard is set for these wireless links. Therefore, the WiFi standard 802.11g with a data rate of 54 Mbits/s is used in this evaluation scenario. Furthermore, wireless links between mobile client nodes and access point nodes can be loss prone as well. Therefore, a propagation loss model is used for wireless links. The Log Distance Propagation Loss Model [Foun17] is used as a loss model for wireless links in this evaluation scenario. The parameter which is used as an exponent for the Log Distance Propagation Loss Model, is set to three by default.

In figure 5.4 an example of mobile client nodes and access point nodes of WiFi networks, which are connected with wireless links, is presented. The additional network components of figure 5.3 are added to the randomly created network. These components are shown in figure 5.4 in more detail. Mobile client nodes can walk with an installed mobility model. In this figure, mobile client nodes have different positions, due to the movement of these nodes during a simulation.

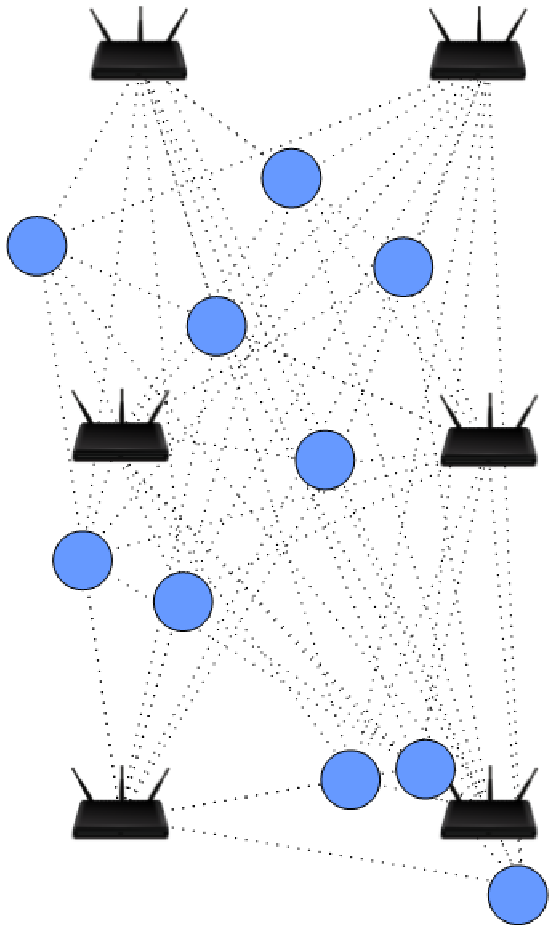

Fig. 5.4: Evaluation scenario (3/3): The added network components of figure 5.3 are shown in this figure in more detail. Mobile client nodes can change their position, due to an installed mobility model on these nodes. For the movement, a predefined area is defined where mobile client nodes can walk.

In the evaluation scenario, mobile client nodes are listening to voicemail messages. These voicemail messages are created by fixed client nodes. Therefore, different settings for the evaluation scenario and different parameters for listening to voicemail messages are specified, to simulate a realistic evaluation scenario. In the following, these settings and parameters are introduced.

5.1.3 Scenario Parameters for Listening to Voicemail Messages

In this evaluation scenario, the cache on all nodes is disabled because for this evaluation scenario there is no need to store data of received Data Packets into a cache. Data of a voicemail message will not be used at a later time again by the mobile client node. When a mobile client node would listen to the same voicemail message again at a later time, then caching data would be useful. But this is not done by mobile client nodes in the evaluation scenario. Furthermore, no other mobile client node will use the same data stored in a cache used by another mobile client node, because each mobile client node listens to its own voicemail message. Therefore, an NDN stack is created in the evaluation scenario where caching is disabled. This NDN stack adds NDN functionality with default values to existing nodes. For instance, a default forwarding strategy is installed, which can be changed later. This NDN stack is installed on all created nodes:

- Router nodes
- Access point nodes
- Fixed client nodes
- Mobile client nodes

Moreover, a global routing helper is installed on each node in the network. With the help of this global routing helper, all tables for using the NDN functionality are created. This means the Forwarding Information Base and Pending Interest Table are created on each node. After this, a forwarding strategy is installed on each network node. The Best-Route Forwarding Strategy is installed on all router nodes, access point nodes, and fixed client nodes. Furthermore, a specific forwarding strategy is installed on all mobile client nodes. Therefore, the command-line argument *fwStrategy* is used, where the used forwarding strategy for mobile client nodes is specified. Available forwarding strategies for mobile client nodes are the Best-Route Forwarding Strategy, the Multicast Forwarding Strategy, or the Mobile-Client Forwarding Strategy.

A timer is used to calculate the Interest Satisfaction Ratio values for all available faces for the forwarding decision when using the Mobile-Client Forwarding Strategy. Outgoing Interest Packets and incoming Data Packets are counted while this timer is running. This timer is set to 1 second when using the Mobile-Client Forwarding Strategy in the evaluation scenario. At the beginning, each Interest Packet is forwarded to all available faces, except the face where the Interest Packet arrives from. Further decisions on where to forward the next Interest Packets take place every second when the timer expires. The decision depends on the Interest Satisfaction Ratio which is calculated for each face to find out which outgoing face is the best face where fewer packet losses occur. So the forwarding strategy can forward further Interest Packets to a selected face or again to all available faces, except the incoming face of the Interest Packet. When only one face is selected for further forwarding, the selected face is the face with the highest Interest Satisfaction Ratio value of this node and the Interest Satisfaction Ratio value of this face also has to be higher than a predefined threshold value. The forwarding decision used in the Mobile-Client Forwarding Strategy is described in section 4.3.2 in more detail.

The simulation duration of the evaluation scenario lasts 5 minutes. During the simulation time, mobile client nodes are listening to voicemail messages created by fixed client nodes. Each mobile client node listens to one voicemail message. It is shown that phone calls last an average of 2.5 minutes [Bund11]. When phone calls last 2.5 minutes on average, then we can assume that listening to voicemail messages takes less than phone calls. Therefore, voicemail message durations for the evaluation scenario are chosen randomly between 30 seconds and one minute. This means the duration for each voicemail message is chosen randomly with a uniform distribution. During the simulation time, each mobile client node is listening to a voicemail message. Mobile client nodes do not start listening to the voicemail messages at the same time in the evaluation scenario. Therefore, start times for listening to a voicemail message are chosen randomly with a uniform distribution for each mobile client node. Finally, the time when a mobile client node is finished with listening to a voicemail message is calculated as well.

Each fixed client node is a so-called producer and generates data. Furthermore, each mobile client node is a consumer of data. Mobile client nodes send Interest Packets to fixed client nodes and fixed client nodes reply with Data Packets. This means, not a real phone call is simulated because only one side sends data. When simulating a real phone call in the evaluation scenario, then the problem of producer mobility occurs which has already been introduced in section 2.4.4. In this evaluation scenario, a mobile client node listens to a voicemail message from a fixed client node. A real voice codec is simulated for voicemail messages. Therefore, the voice codec G.711/PCM is used for voicemail messages. This voice codec requires 64 kbps bandwidth and typically data is sent all 10 ms or all 20 ms. To simulate this voice codec in this evaluation scenario, 100 Interest Packets per second are sent, where each Data Packet responding to an Interest Packet has a size of 82 bytes.

Therefore, an application helper is installed on client nodes in the evaluation scenario. The *ns3::ndn::ConsumerCbr* application is installed on mobile client nodes. Retransmissions are disabled in this consumer application because retransmitted Interest Packets are not relevant for the evaluation scenario. The frequency of sending Interest Packets is set to 100. This means, 100 Interest Packets are sent out each second by mobile client nodes. Furthermore, the *ns3::ndn::Producer* application is installed on fixed client nodes. The size of Data Packets is set to 82 bytes.

Moreover, additional settings are set for each communication pair. 10 mobile client nodes and 10 fixed client nodes exist in the NDN network of the evaluation scenario. Each mobile client node communicates with one fixed client node, where a mobile client node listens to a voicemail message produced by a fixed client node. This means, for instance, mobile client node 1 communicates with fixed client node 1, and mobile client node 2 communicates with fixed client node 2. The following settings are set for each communication pair for the application helper and the application container:

- Prefix
- Voicemail message call start time
- Voicemail message call end time

A prefix for each communication pair is set. For instance, when mobile client node 1 listens to a voicemail message produced by fixed client node 1, then the prefix */voicemail_1* is used for Interest Packets and Data Packets during data transfer. Furthermore, the voicemail call start time and the voicemail call end time are set for each producer and consumer. After this, the prefix is added to the global routing helper as well.

Moreover, all possible routes are calculated for the Forwarding Information Base. This means all possible routes are calculated for each node. Specific entries have to be removed later from the Forwarding Information Base. This has to be done in order to avoid that a mobile client node can send Interest Packets to an access point node, and from the access point node to another mobile client node. Therefore, all entries in the Forwarding Information Base from access point nodes to mobile client nodes are removed. All entries in the Forwarding Information Base from a network node to an access point node have to be deleted as well. This ensures that Interest Packets are sent out via an access point node to the network and are then forwarded to a fixed client node, without using another mobile client node as a node within the route.

Finally, trace helpers are installed on all client nodes. This means trace helpers are installed on mobile client nodes and fixed client nodes as well. Therefore, the Rate Trace and the App Delay Trace are used for the evaluation scenario. With the help of these trace helpers, output files can be created for each simulation scenario run. The paths to the output files are also set in the evaluation scenario. After this, a file is created, where the call durations of voicemail messages are stored for each communication pair.

Different forwarding strategies which are installed on mobile client nodes within an NDN network will be evaluated in this chapter. This is done to show that the newly implemented forwarding strategy, called Mobile-Client Forwarding Strategy, works well for mobile client nodes in an NDN network. Therefore, the results of the performance evaluation for the forwarding strategies Mobile-Client, Best-Route, and Multicast as mobile client forwarding strategies are compared to each other. How the performance evaluation is performed, is introduced in the following.

5.2 What to Evaluate?

At first, to evaluate the performance of the Mobile-Client Forwarding Strategy implementation, results of the evaluation scenario with different Interest Satisfaction Ratio threshold values for the Mobile-Client Forwarding Strategy are shown and compared. Different Interest Satisfaction Ratio threshold values are compared, to determine an Interest Satisfaction Ratio threshold value which works well for the Mobile-Client Forwarding Strategy as forwarding strategy for mobile client nodes.

Moreover, the results of the Mobile-Client Forwarding Strategy with a predefined Interest Satisfaction Ratio threshold value are compared with the results of other forwarding strategies, namely Best-Route and Multicast. In the evaluation scenario, the compared forwarding strategies are always installed on mobile client nodes. All other nodes in the network use a default forwarding strategy. This is done to show that the Mobile-Client Forwarding Strategy works well when client nodes can walk in an NDN network.

Finally, to evaluate the results of these forwarding strategies for the performance evaluation, metrics are obtained and calculated from evaluation scenario output files. The used metrics for the performance evaluation are introduced in the following.

5.3 Evaluation Metrics

In this section, the measured data and the used metrics are described in detail. To obtain metrics from a simulation scenario ndnSIM provides trace helpers which have already been introduced in section 4.7.1. Therefore, the Rate Trace and the App Delay Trace are used in the evaluation scenario. With the help of these trace helpers, output files are generated out of a simulation scenario run. Furthermore, metrics can be obtained and calculated out of these output files with the help of the provided Python scripts, which have already been introduced in section 4.7.2.

5.3.1 Interest Satisfaction Ratio

The Interest Satisfaction Ratio (ISR) has already been introduced and defined in section 3.3.2. The Interest Satisfaction Ratio represents the percentage of Interest Packets, which are satisfied through a Data Packet and can be calculated with equation 3.1. A detailed introduction of the Interest Satisfaction Ratio can be found in chapter 3.

This metric is calculated with the Python script *calculateMetrics.py* where metrics from output files of simulation scenario runs are retrieved. Furthermore, this metric is calculated for each voicemail message between a content consumer and a content producer. After this, the average value of this metric is calculated from all voicemail message calls from one simulation scenario run. Finally, the average Interest Satisfaction Ratio value of all simulation scenario runs is calculated.

5.3.2 Estimated Mean Opinion Score

The *Estimated Mean Opinion Score* (MOS-CQE, MOS Conversational Quality Estimated) [SMJI15] is an additional metric used for the performance evaluation. This metric is used to compare the quality of several voicemail messages.

Generally, The *Mean Opinion Score* (MOS) is a score where the quality of a conversation can be expressed with a value. This value is rated by users and is a value between 1 and 5. The higher the Mean Opinion Score value, the higher the quality for the user. But no real user ratings are considered for this performance evaluation. Therefore, the Mean

Opinion Score has to be estimated. To estimate the Mean Opinion Score, the Estimated Mean Opinion Score (MOS-CQE) is used for the performance evaluation. The Estimated Mean Opinion Score can be calculated using equation 5.12.

$$MOS - CQE = \begin{cases} 1 & \text{for } R \leq 0 \\ 1 + 0.035R + R(R-60)(100-R)7 \cdot 10^{-6} & \text{for } 0 < R < 100 \\ 4.5 & \text{for } R \geq 100 \end{cases} \tag{5.12}$$

Before calculating the Estimated Mean Opinion Score, the so-called R-value has to be calculated. This R-value is used for the Estimated Mean Opinion Score calculation and is a so called E-Model [SMJI15], where packet loss and delay are taken into account as shown in equation 5.13. Furthermore, the R-value can be mapped to user satisfaction as well. This means the R-value indicates how satisfied a user can be with the quality of a conversation. For this evaluation scenario, user satisfaction during listening to voicemail messages is estimated. A mapping from the R-value to user satisfaction is shown in table 5.1.

R-value range	Speech transmission quality category	User satisfaction
100-90	Best	Very satisfied
90-80	High	Satisfied
80-70	Medium	Some users dissatisfied
70-60	Low	Many users dissatisfied
60-50	Poor	Nearly all users dissatisfied

Tab. 5.1: A mapping from R-values to user satisfaction [SMJI15].

Possible Mean Opinion Score values are shown in table 5.2, where several Mean Opinion Score values are compared with R-values.

R-value	MOS score	User experience
90	4.3	Good
80	4.0	Good
70	3.6	Fair
60	3.1	Fair
50	2.6	Poor

Tab. 5.2: R-values vs. MOS score values [SMJI15].

The R-value can be calculated using equation 5.13 [SMJI15]. Therefore the one-way delay I_d and the effective equipment impairment factor $I_{e\text{-}eff}$ have to be calculated before.

$$R = 93.2 - I_{\text{d}} - I_{\text{e-eff}} \tag{5.13}$$

The first parameter, which is needed to calculate the R-value, represents the one-way delay I_d and can be calculated using equation 5.14. Furthermore, the function H(x) of this equation is defined in equation 5.15 [SMJI15].

$$I_{\text{d}} = 0.024d + 0.11(d - 177.3)H(d - 177.3) \tag{5.14}$$

$$where = \begin{cases} H(x) = 0 & \text{if } x < 0 \\ H(x) = 1 & \text{if } x \geq 0 \end{cases} \tag{5.15}$$

To calculate the delay d, equation 5.16 is used [SMJI15]. The delay d is a one-way delay between endpoint nodes and is calculated by summing up the network delay d_n, the codec delay d_c and the jitter buffer delay d_b. The network delay is the delay between sending a packet and receiving the same packet. This delay is caused by the network. The codec delay is the delay which is caused by encoding or decoding a packet and is 0.25 by default. The jitter buffer delay is 50 by default.

$$d = d_{\text{n}} + d_{\text{c}} + d_{\text{b}} \tag{5.16}$$

The second parameter which is needed to calculate the R-value, is the effective equipment impairment factor $I_{e\text{-eff}}$. To calculate the $I_{e\text{-eff}}$, equation 5.17 is used [SMJI15]. I_e represents the equipment impairment factor. The value of I_e is zero when no loss happens. B_{pl} is the packet-loss robustness factor and is 34 by default. Furthermore, the values for I_e and B_{pl} are codec specific. The variable P_{pl} represents the average packet-loss rate in percent, which is the error rate multiplied by 100. Furthermore, the burst ratio BurstR is a variable for the packet loss as well.

$$I_{\text{e-eff}} = I_{\text{e}} + (95 - I_{\text{e}}) \cdot \frac{P_{\text{pl}}}{\frac{P_{\text{pl}}}{BurstR} + B_{\text{pl}}} \tag{5.17}$$

The burst ratio can be calculated using equation 5.18 [SMJI15]. When packets get lost randomly, then the burst ratio is 1. Packet loss is rarely random, but bursty. When the packet loss is bursty, then the burst ratio is greater than 1. The variable p is the probability that a packet gets not lost and the next packet gets lost. The variable q is the probability that a packet gets lost and the next packet gets not lost.

$$BurstR = \frac{1}{p + q} = \frac{P_{\text{pl}}/100}{p} = \frac{1 - P_{\text{pl}}/100}{q} \tag{5.18}$$

The Estimated Mean Opinion Score can be calculated with the help of the above-described equations, to calculate the estimated quality of experience for the user, while listening to voicemail messages in the evaluation scenario. The Estimated Mean Opinion Score is a value between 1 and 4.35. This metric is also calculated with the Python script *calculateMetrics.py*, as already described in section 5.3.1.

5.3.3 Data Packet Error Rate

The *Data Packet Error Rate* is an additional metric which is used for the performance evaluation. This metric can be calculated using equation 5.19 and is a value between zero and one. If the Data Packet Error Rate is zero, then no packet gets lost. Furthermore, if the Data Packet Error Rate is one, then all packets get lost.

$$Packet\ Error\ Rate = \frac{Number\ of\ Interest\ Packets - Number\ of\ Data\ Packets}{Number\ of\ Interest\ Packets} \tag{5.19}$$

This metric is also calculated with the Python script *calculateMetrics.py*, as already described in section 5.3.1.

5.3.4 Packet Success Percentage

Furthermore, the *Packet Success Percentage* is an additional metric for the performance evaluation, to show the successfully transferred packets in percent. The Packet Success Percentage can be calculated using equation 5.20.

$$Packet\ Success\ Percentage = \frac{Number\ of\ received\ Packets}{Number\ of\ expected\ Packets} \cdot 100 \tag{5.20}$$

This metric is also calculated with the Python script *calculateMetrics.py*, as already described in section 5.3.1. Different Packet Success Percentages are calculated in this Python script:

- Interest Packet Success Percentage
- Data Packet Success Percentage

5.3.5 Delay

Moreover, the average delay is calculated for the performance evaluation as well. Therefore, the average network delay $\mathrm{d_n}$ in milliseconds from sending a packet to receiving the same packet is calculated. This metric is also calculated with the Python script *calculateMetrics.py*, as already described in section 5.3.1.

5.3.6 Duplicate Packets

The number of duplicate packets is an additional metric, which is used for the performance evaluation. The number of duplicate packets is the number of packets which are received more than one time. This metric is calculated using equation 5.21. This metric is also calculated with the Python script *calculateMetrics.py*, as already described in section 5.3.1.

$$\begin{split} Duplicate\ Packets = Number\ of\ all\ received\ Data\ Packets\ - \\ Number\ of\ different\ received\ Data\ Packets \end{split} \tag{5.21}$$

5.3.7 Missed Packets

Furthermore, missed packets are an additional metric for the performance evaluation. Data Packets which do not arrive are called missed packets. The number of missed packets can be calculated using equation 5.22. This metric is also calculated with the Python script *calculateMetrics.py*, as already described in section 5.3.1.

$$\begin{split} Missed\ Packets = Number\ of\ expected\ Interest\ or\ Data\ Packets\ - \\ Number\ of\ received\ Data\ Packets \end{split} \tag{5.22}$$

5.3.8 Hop Count

Finally, the hop count is calculated for the performance evaluation as well. The hop count is the number of hops a packet traverses in the network. The network hops are counted from sending this packet from the content producer to receiving the packet at the content consumer. This metric is also calculated with the Python script *calculateMetrics.py*, as already described in section 5.3.1.

The results of the performance evaluation are presented in the following. Therefore, metrics are calculated out of simulation scenario output files. Firstly, the results of the Mobile-Client Forwarding Strategy with different Interest Satisfaction Ratio threshold values are presented and compared, to find out which Interest Satisfaction Ratio threshold value works fine for the Mobile-Client Forwarding Strategy.

5.4 Results of the Mobile-Client Forwarding Strategy

In this section, the results of the performance evaluation for the Mobile-Client Forwarding Strategy with different Interest Satisfaction Ratio threshold values are presented. The Interest Satisfaction Ratio threshold is a predefined value in the Mobile-Client Forwarding Strategy, which influences the forwarding decision. Results of different Interest Satisfaction Ratio threshold values are compared to each other, to find an Interest Satisfaction Ratio threshold value where the Mobile-Client Forwarding Strategy performs well. One of these Interest Satisfaction Ratio threshold values, which performs well for mobile client nodes, is chosen for further performance evaluations, where results of the Mobile-Client Forwarding Strategy are compared with results of the Best-Route Forwarding Strategy and the Multicast Forwarding Strategy. The results of this performance evaluation are presented in section 5.5.

To find an Interest Satisfaction Ratio threshold value, where the Mobile-Client Forwarding Strategy performs well as a forwarding strategy for mobile client nodes, the following Interest Satisfaction Ratio threshold values are evaluated with the evaluation scenario: 10, 20, 30, 40, 50, 60, 70, 80 and 90 %.

Therefore, several metrics are obtained and calculated out of output files from simulation scenario runs for the performance evaluation. Average values for several metrics are calculated from 100 different simulation scenario runs with the evaluation scenario, where a confidence interval of 95% is considered as well. After this, bar charts with the results are created, to show the results of the performance evaluation.

Different parameters can vary within the evaluation scenario. Therefore, the following parameters are used in the evaluation scenario for the performance evaluation of the Mobile-Client Forwarding Strategy with different Interest Satisfaction Ratio threshold values:

- Three access point nodes for each WiFi network
- A distance of 40 units in the visualizer of ndnSIM between access point nodes of the same WiFi network
- A distance of 40 units in the visualizer of ndnSIM between access point nodes of different WiFi networks

Mobile client nodes can walk in a predefined movement area. When a mobile client node is too far away from an access point node, then the connection between this access point node and the mobile client node gets lost and the mobile client node can connect to another access point node in the nearby environment. Therefore, distances between all access point nodes are chosen in such a way, that mobile client nodes do not lose the connection to all access point nodes at the same time. When a higher distance between access point nodes is chosen, then a mobile client node can be too far away from the available access point nodes and the connection to all access point nodes gets lost. Therefore, higher distances than 40 units in the visualizer of ndnSIM are not chosen for this evaluation scenario. Several metrics are evaluated and compared when using these settings. The results of the performance evaluation are presented in the following.

5.4.1 Interest Satisfaction Ratio Results

At first, the results for the Interest Satisfaction Ratio for the Mobile-Client Forwarding Strategy with different Interest Satisfaction Ratio threshold values are shown in figure 5.5.

These results are calculated the following way. The Interest Satisfaction Ratio values are calculated for each voicemail message between a content consumer and a content producer. Furthermore, the average Interest Satisfaction Ratio values for all voicemail messages from one simulation scenario run are calculated. Finally, the average Interest Satisfaction Ratio values for all simulation scenario runs are calculated.

The result bar chart in figure 5.5 shows, that the Interest Satisfaction Ratio threshold value of 90 % achieves the highest Interest Satisfaction Ratio value. Furthermore, it is shown in this bar chart, the higher the Interest Satisfaction Ratio threshold value in the Mobile-Client Forwarding Strategy is chosen, the higher the Interest Satisfaction Ratio result.

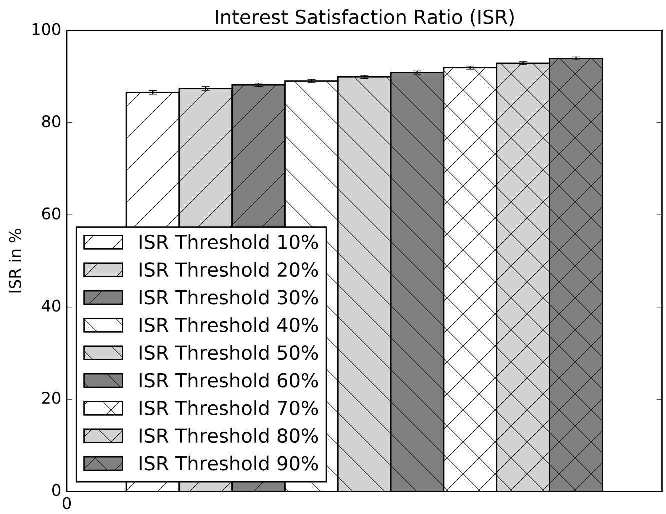

Fig. 5.5: Interest Satisfaction Ratio results: The results are shown in this figure where the Interest Satisfaction Ratio threshold value of 90 % achieves the highest Interest Satisfaction Ratio value.

5.4.2 Estimated Mean Opinion Score Results

The results for the Estimated Mean Opinion Score (MOS-CQE) for the Mobile-Client Forwarding Strategy with different Interest Satisfaction Ratio threshold values are shown in figure 5.6.

Furthermore, the highest achievable value for the MOS-CQE value is drawn as a red line in this figure. A MOS-CQE value between 4.35 and 3.6 means a good user experience. Therefore, a dashed line is drawn at the MOS-CQE value of 3.6, to show the boundary between good and fair user experience. Moreover, a MOS-CQE value between 3.6 and 2.6 means a fair user experience. Therefore, a dashed line is drawn at the MOS-CQE value of 2.6, to show the boundary between fair and poor user experience as well. Each MOS-CQE value under 2.6 means a poor user experience.

These MOS-CQE results are calculated the same way as already described for the results of the Interest Satisfaction Ratio values. The MOS-CQE values are calculated for each voicemail message between a content consumer and a content producer. Furthermore, the average MOS-CQE values for all voicemail messages from one simulation scenario run are calculated. Finally, the average MOS-CQE values for all simulation scenario runs are calculated.

The result bar chart in figure 5.6 shows that the Interest Satisfaction Ratio threshold values of 90 % and 80 % achieve a good user experience. Furthermore, the other Interest Satisfaction Ratio threshold values achieve a fair user experience.

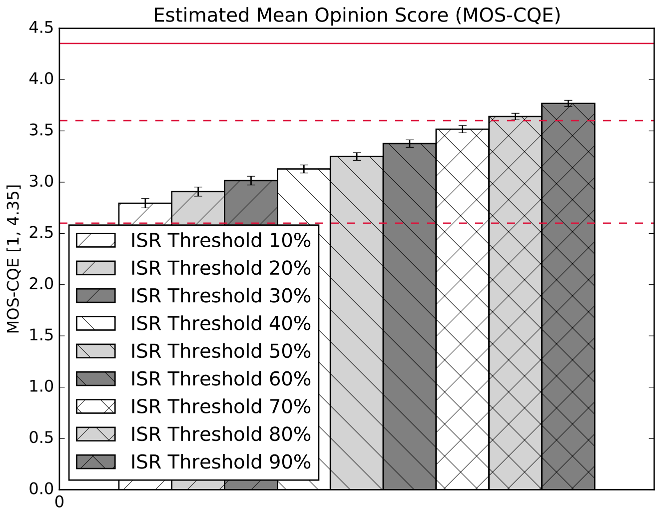

Fig. 5.6: MOS-CQE results: The results are shown in this figure where the Interest Satisfaction Ratio threshold values of 90 % and 80 % achieve the highest MOS-CQE values with a good user experience.

5.4.3 Data Packet Error Rate Results

The results for the Data Packet Error Rate for the Mobile-Client Forwarding Strategy with different Interest Satisfaction Ratio threshold values are shown in figure 5.7.

These Data Packet Error Rate results are calculated the same way as already described for the results of the Interest Satisfaction Ratio values in section 5.4.1. The result bar chart in figure 5.7 shows that the higher the Interest Satisfaction Ratio threshold value in the Mobile-Client Forwarding Strategy, the lower the Data Packet Error Rate result. Moreover, this means more Data Packets are received at mobile client nodes when a high Interest Satisfaction Ratio threshold value is chosen.

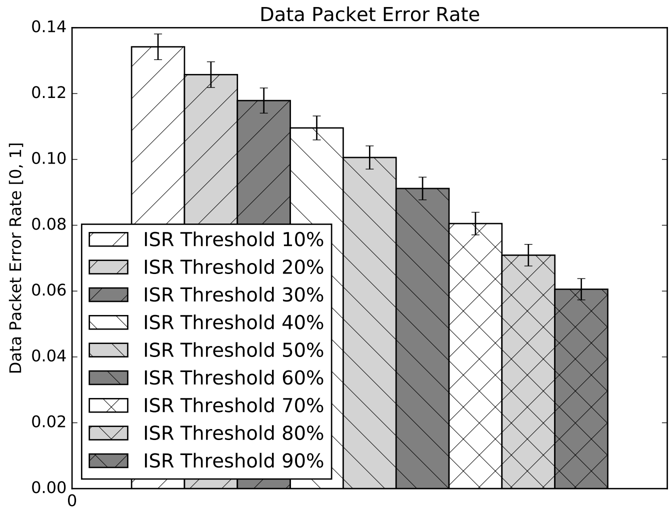

Fig. 5.7: Data Packet Error Rate results: The results show that the higher the Interest Satisfaction Ratio threshold value, the lower the Data Packet Error Rate result.

5.4.4 Packet Success Percentage Results

The results for the Packet Success Percentage of Interest Packets and Data Packets are shown in the following figures. Therefore, sent and received Interest Packets or Data Packets are counted for calculating Interest Packet Success Percentages or the Data Packet Success Percentages. For instance, sent Interest Packets are Interest Packets that are sent by a content consumer, and received Interest Packets are Interest Packets that are received at a content producer. The same can also be calculated for Data Packets.

The results for the Packet Success Percentage of sent Interest Packets are shown in figure 5.8. Furthermore, the results for the Packet Success Percentage of received Interest Packets are shown in figure 5.9 and finally, the results for the Packet Success Percentage of sent Data Packets are shown in figure 5.10.

These Packet Success Percentage results are calculated the same way as already described for the results of the Interest Satisfaction Ratio values in section 5.4.1. The result bar charts in figures 5.8, 5.9 and 5.10 show, that the higher the Interest Satisfaction Ratio threshold value in the Mobile-Client Forwarding Strategy, the higher the Packet Success Percentage result. Moreover, this means more Data Packets are received at mobile client nodes when a high Interest Satisfaction Ratio threshold value is chosen. These results also show that more Interest Packets are sent than received. This is because of packet loss when using a loss model.

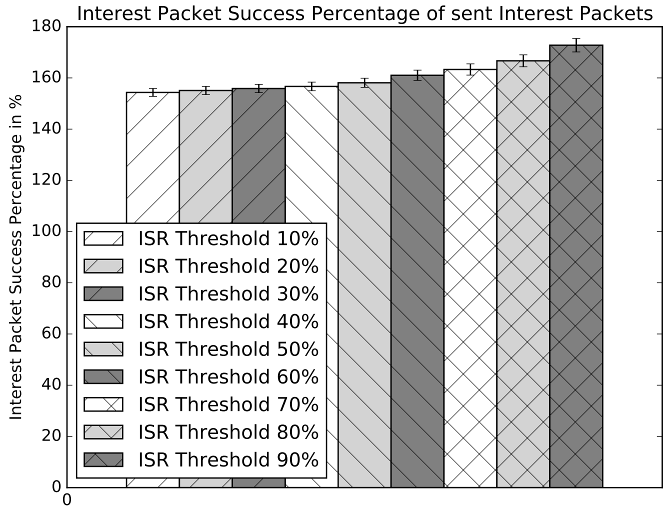

Fig. 5.8: Interest Packet Success Percentage results of sent Interest Packets: The results show that the higher the Interest Satisfaction Ratio threshold value, the higher the Interest Packet Success Percentage result of sent Interest Packets.

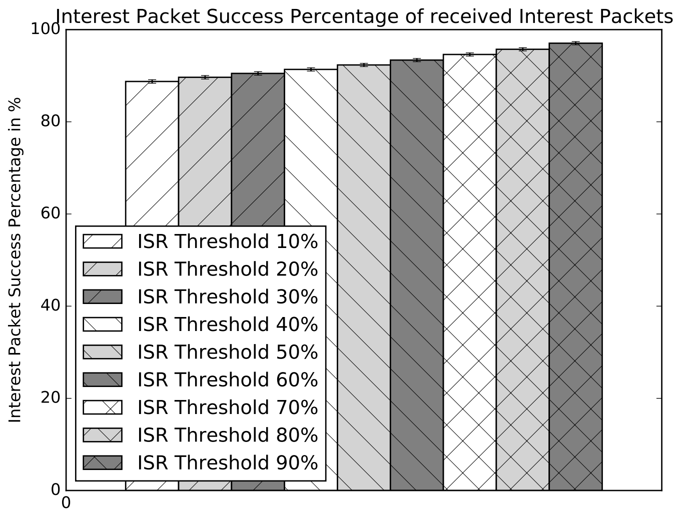

Fig. 5.9: Interest Packet Success Percentage results of received Interest Packets: The results show that the higher the Interest Satisfaction Ratio threshold value, the higher the Interest Packet Success Percentage result of received Interest Packets.

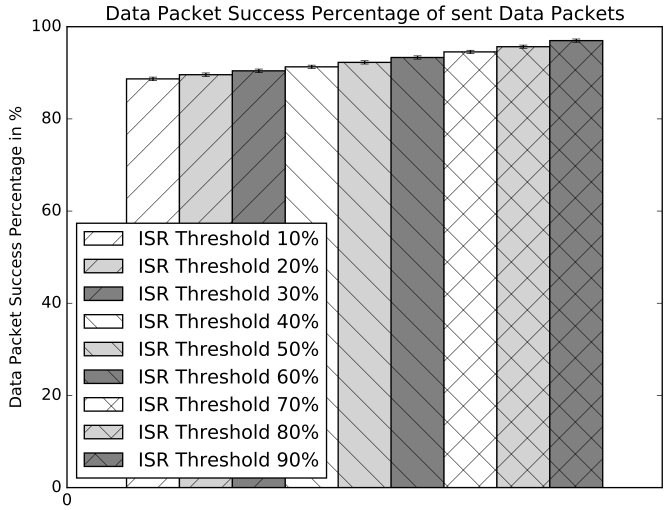

Fig. 5.10: Data Packet Success Percentage results of sent Data Packets: The results show that the higher the Interest Satisfaction Ratio threshold value, the higher the Data Packet Success Percentage result of sent Data Packets.

5.4.5 Other Metrics

Additional metrics like the delay, duplicate packets, missed packets and the hop count are implemented for the practical part of this master thesis. Results for these metrics are calculated for the evaluation scenario as well. But these results are not shown in this section, because these results are not so interesting when comparing different Interest Satisfaction Ratio threshold values for the Mobile-Client Forwarding Strategy. The metrics duplicate packets and missed packets show a number of packets. Metrics with a ratio or a percentage value are more meaningful when comparing the results of these metrics with different Interest Satisfaction Ratio threshold values for the Mobile-Client Forwarding Strategy. A percentage value or a rate value as result is more meaningful than a number of packets as result, because when only a number of packets are presented as result, then the maximum number of packets always has to be known as well. Therefore presenting results with rate values or percentage values is clearer.

5.5 Results Compared with Different Forwarding Strategies

In this section, the performance evaluation of the Mobile-Client Forwarding Strategy compared with other forwarding strategies is shown. The compared forwarding strategies are the Best-Route Forwarding Strategy and the Multicast Forwarding Strategy.

In section 5.4, results for different Interest Satisfaction Ratio threshold values are compared to each other, to find an Interest Satisfaction Ratio threshold value where the Mobile-Client Forwarding Strategy performs well as a forwarding strategy for mobile client nodes within the created evaluation scenario. These results show that an Interest Satisfaction Ratio threshold value of 90 % performs well for the Mobile-Client Forwarding Strategy. A disadvantage is that more packets are produced with this threshold value. This is shown by the results of the Packet Success Percentage of sent Interest Packets in section 5.4.4. But the Interest Satisfaction Ratio threshold value of 90 % shows good results for the metrics Interest Satisfaction Ratio, Estimated Mean Opinion Score (MOS-CQE) and the Data Packet Error Rate. For instance, only the threshold values of 90 % and 80 % achieve a good user experience for the users listening to voicemail messages in the evaluation scenario.

Due to these facts, the Mobile-Client Forwarding Strategy is compared to other forwarding strategies when using an Interest Satisfaction Ratio threshold value of 90 % for the Mobile-Client Forwarding Strategy.

In this section, the results of the performance evaluation for the forwarding strategies Best-Route, Multicast and Mobile-Client are presented. Therefore, several metrics are obtained and calculated out of output files from simulation scenario runs from the evaluation scenario as well. Average values for several metrics are calculated from 100 different simulation scenario runs with the evaluation scenario where a confidence interval of 95% is considered as well. After this, bar charts with the results are created, to show the results of the performance evaluation.

Different parameters can vary within the evaluation scenario. Therefore, the following parameters are used in the evaluation scenario for the performance evaluation of the Mobile-Client Forwarding Strategy compared with different forwarding strategies:

- One or three access point nodes for each WiFi network
- A distance of 40 units in the visualizer of ndnSIM between access point nodes from the same WiFi network
- A distance of 30 or 40 units in the visualizer of ndnSIM between access point nodes of different WiFi networks

In section 5.4, distances between access point nodes from the same WiFi network and distances between access point nodes of different WiFi networks in the simulation scenario are chosen to 40 units in the visualizer of ndnSIM. In this section, the distances between access point nodes of different WiFi networks can vary for the performance evaluation where different forwarding strategies are compared. Distances of 30 and 40 units in the visualizer of ndnSIM between access point nodes of different WiFi networks are compared. These distances vary to better show differences in the results between different forwarding strategies. Several metrics are evaluated and compared when using these settings. The results of the performance evaluation are presented in the following.

5.5.1 Interest Satisfaction Ratio Results

The Interest Satisfaction Ratio results for the forwarding strategies Best-Route, Multicast and Mobile-Client are shown in the following with different parameters for the evaluation scenario.

At first, the results with one access point node per WiFi network are shown. Therefore, the distances between two access point nodes of different WiFi networks are 30 units in the visualizer of ndnSIM. These results are shown in figure 5.11.

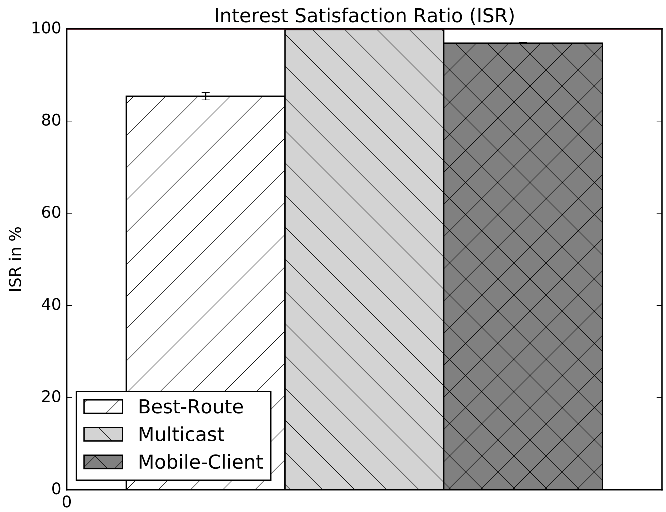

Fig. 5.11: Interest Satisfaction Ratio results (1/4): One access point node per WiFi network and a distance of 30 units between access point nodes of different WiFi networks.

Furthermore, figure 5.12 shows the results of one access point node per WiFi network and the distances between two access point nodes of different WiFi networks are 40 units in the visualizer of ndnSIM.

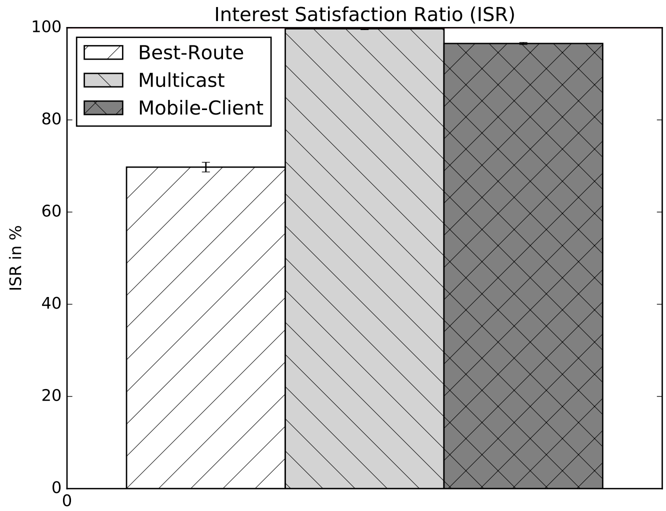

Fig. 5.12: Interest Satisfaction Ratio results (2/4): One access point node per WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

Furthermore, figure 5.13 shows the results of three access point nodes per WiFi network. The distances between access point nodes of the same WiFi network are 40 units in the visualizer of ndnSIM and the distances between access point nodes of different WiFi networks are 30 units in the visualizer of ndnSIM.

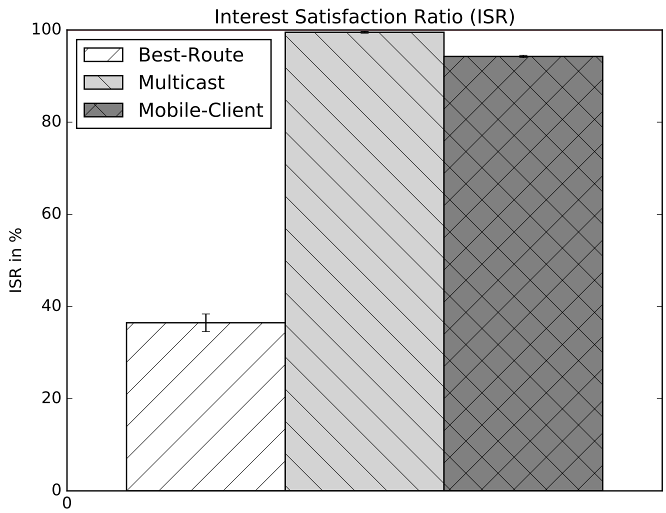

Fig. 5.13: Interest Satisfaction Ratio results (3/4): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network and a distance of 30 units between access point nodes of different WiFi networks.

Finally, figure 5.14 shows the results of three access point nodes per WiFi network. The distances between access point nodes of the same WiFi network are 40 units in the visualizer of ndnSIM and the distances between access point nodes of different WiFi networks are 40 units in the visualizer of ndnSIM as well.

Fig. 5.14: Interest Satisfaction Ratio results (4/4): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

These results are calculated the following way for each forwarding strategy. The Interest Satisfaction Ratio values are calculated for each voicemail message between a content consumer and a content producer. Furthermore, the average Interest Satisfaction Ratio values for all voicemail messages from one simulation scenario run are calculated. Finally, the average Interest Satisfaction Ratio values for all simulation scenario runs are calculated.

The result bar charts in figures 5.11, 5.12, 5.13 and 5.14 show that the Interest Satisfaction Ratio is the best for the Multicast Forwarding Strategy. Furthermore, the Interest Satisfaction Ratio for the Mobile-Client Forwarding Strategy achieves good results. These results are almost as good as the results of the Multicast Forwarding Strategy.

Finally, the Interest Satisfaction Ratio of the Best-Route Forwarding Strategy achieves the worst results. These results show that the Best-Route Forwarding Strategy cannot handle mobility of mobile client nodes well. The Best-Route Forwarding Strategy uses always the best route. When a mobile client node moves away from the access point node, where the best route is defined, then the Best-Route Forwarding Strategy reacts very slowly. This means, the higher the number of access point nodes of a WiFi network and the higher the distances between access point nodes, the lower the Interest Satisfaction Ratio of the Best-Route Forwarding Strategy.

5.5.2 Estimated Mean Opinion Score Results

The Estimated Mean Opinion Score (MOS-CQE) results for the forwarding strategies Best-Route, Multicast and Mobile-Client are shown in the following with different parameters for the evaluation scenario. Therefore, the same parameters as already described in section 5.5.1 are used.

These MOS-CQE results are calculated the same way as already described for the results of the Interest Satisfaction Ratio values in section 5.5.1. The results for the MOS-CQE are shown in figures 5.15, 5.16, 5.17 and 5.18. The red lines in these result charts are already described in section 5.4.2.

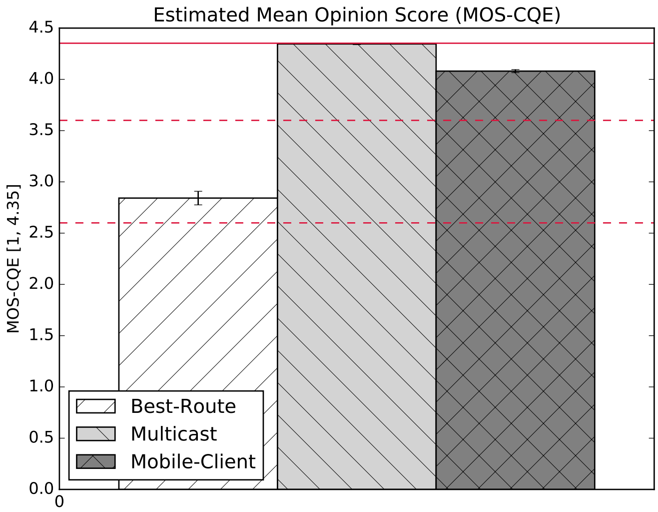

Fig. 5.15: MOS-CQE results (1/4): One access point node per WiFi network and a distance of 30 units between access point nodes of different WiFi networks.

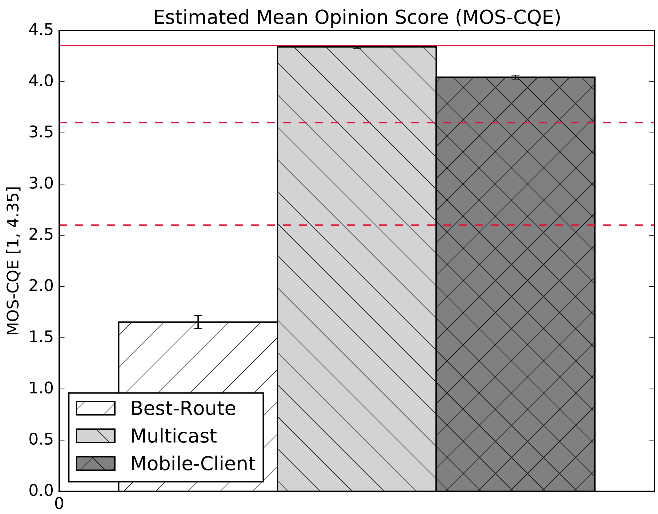

Fig. 5.16: MOS-CQE results (2/4): One access point node per WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

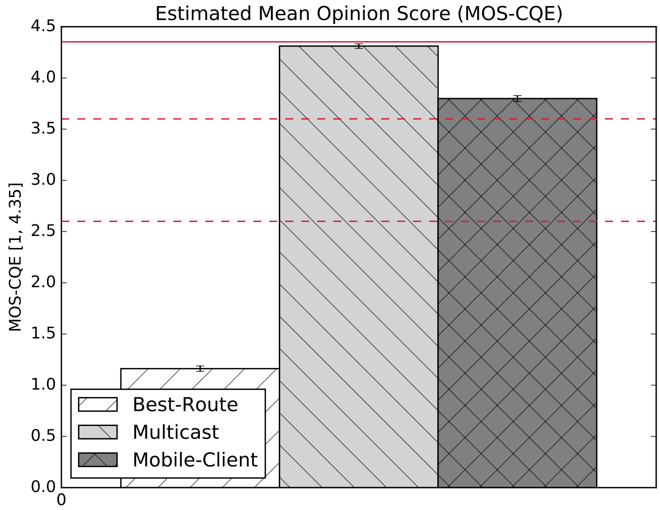

Fig. 5.17: MOS-CQE results (3/4): Three access point node per WiFi network, a distance of 40 units between access point nodes of the same WiFi network and a distance of 30 units between access point nodes of different WiFi networks.

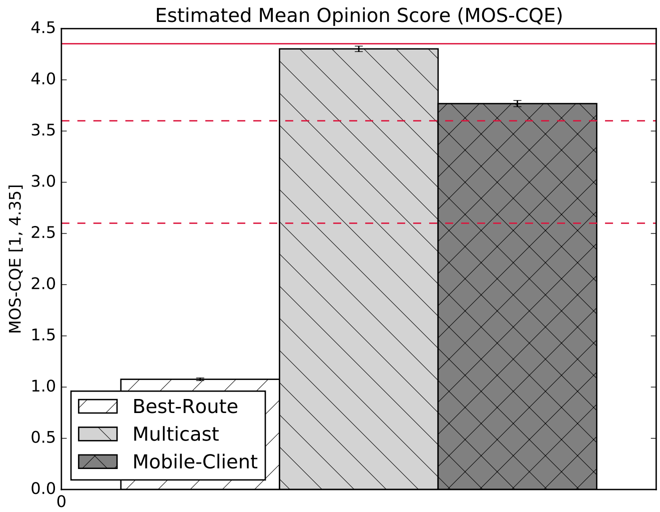

Fig. 5.18: MOS-CQE results (4/4): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

The result bar charts in figures 5.15, 5.16, 5.17 and 5.18 show that the MOS-CQE values for the Multicast Forwarding Strategy achieve the best results, where a good user experience is reached. Furthermore, the MOS-CQE values for the Mobile-Client Forwarding Strategy achieve also good results, where a good user experience is reached as well.

Finally, the MOS-CQE values for the Best-Route Forwarding Strategy achieve the worst results. These results show as well that the Best-Route Forwarding Strategy cannot handle mobility of mobile client nodes well like already described in section 5.5.1. The

Best-Route Forwarding Strategy achieves a fair user experience in figure 5.15, but a poor user experience in figures 5.16, 5.17 and 5.18.

5.5.3 Data Packet Error Rate Results

The Data Packet Error Rate results for the forwarding strategies Best-Route, Multicast and Mobile-Client are shown in the following with different parameters for the evaluation scenario. Therefore, the same parameters as already described in section 5.5.1 are used.

These Data Packet Error Rate results are calculated the same way as already described for the results of the Interest Satisfaction Ratio values in section 5.5.1. The results for the Data Packet Error Rate are shown in figures 5.19, 5.20, 5.21 and 5.22.

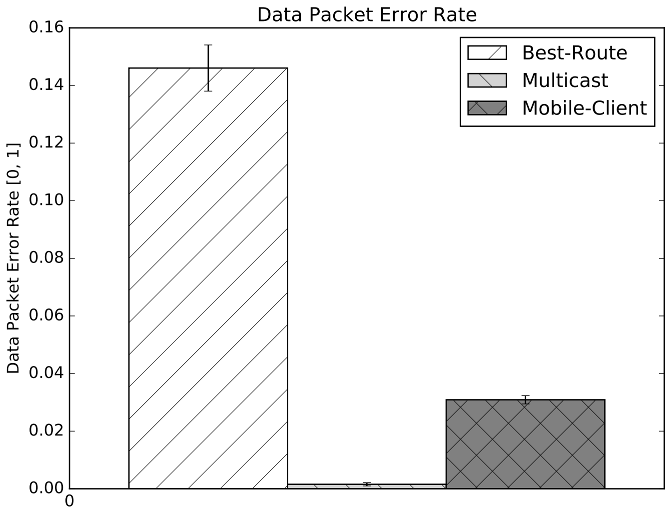

Fig. 5.19: Data Packet Error Rate results (1/4): One access point per WiFi network and a distance of 30 units between access point nodes of different WiFi networks.

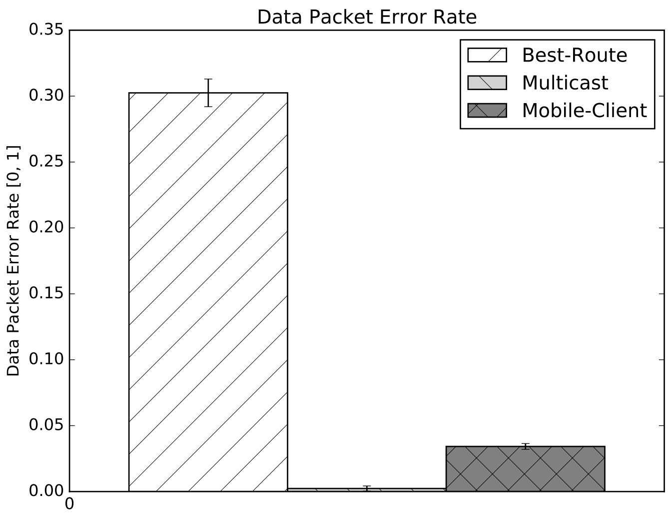

Fig. 5.20: Data Packet Error Rate results (2/4): One access point node per WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

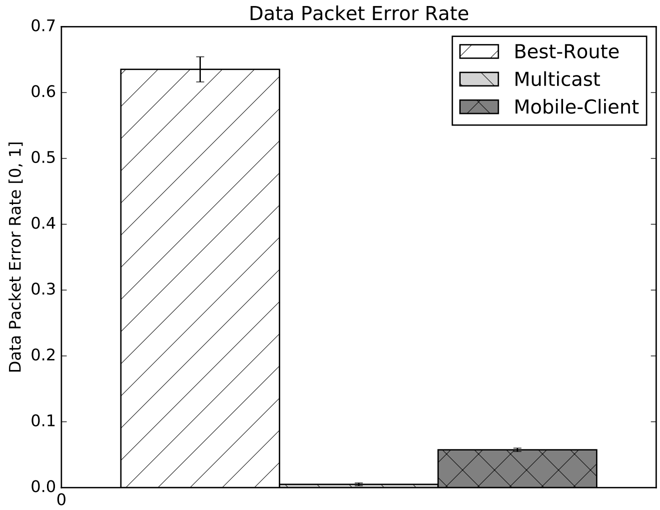

Fig. 5.21: Data Packet Error Rate results (3/4): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network, and a distance of 30 units between access point nodes of different WiFi networks.

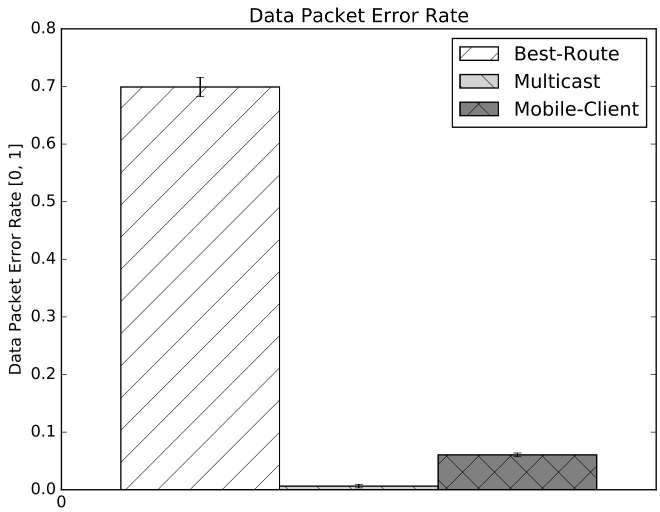

Fig. 5.22: Data Packet Error Rate results (4/4): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network, and a distance of 40 units between access point nodes of different WiFi networks.

The result bar charts in figures 5.19, 5.20, 5.21 and 5.22 show that the Data Packet Error Rate for the Multicast Forwarding Strategy achieves the best results. This means fewer packets get lost. Furthermore, the Data Packet Error Rate for the Mobile-Client Forwarding Strategy achieves almost as good results as the Multicast Forwarding Strategy.

Finally, the Data Packet Error Rate for the Best-Route Forwarding Strategy achieves the worst results. These results show that the Best-Route Forwarding Strategy has the highest amount of packet loss and that the Best-Route Forwarding Strategy cannot handle mobility of mobile client nodes well like already described in section 5.5.1.

5.5.4 Packet Success Percentage Results

The Packet Success Percentage results of Interest Packets and Data Packets for the forwarding strategies Best-Route, Multicast and Mobile-Client are shown in the following with different parameters for the evaluation scenario. Therefore, the same parameters as in section 5.5.1 are used.

The results for the Packet Success Percentage of sent Interest Packets are shown in figures 5.23 and 5.24. Furthermore, the results for the Packet Success Percentage of received Interest Packets are shown in figures 5.25 and 5.26. Finally, the results for the Packet Success Percentage of sent Data Packets are shown in figures 5.27 and 5.28.

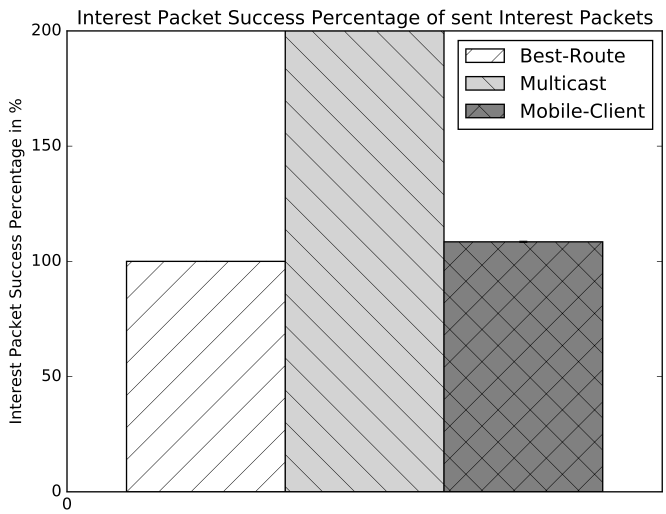

Fig. 5.23: Interest Packet Success Percentage results of sent Interest Packets (1/2): One access point node per WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

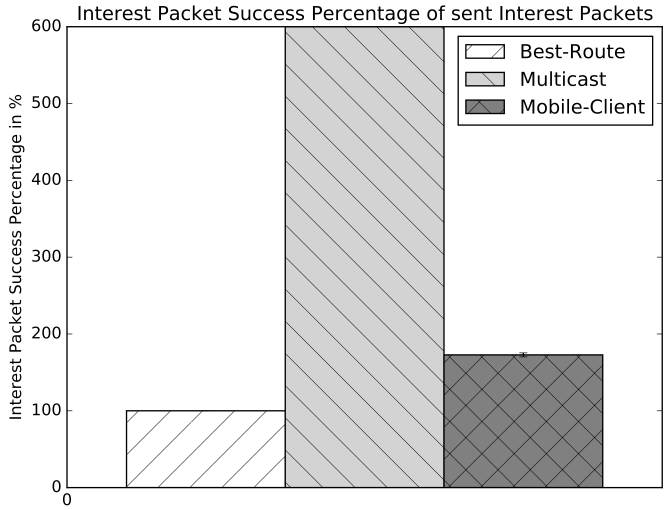

Fig. 5.24: Interest Packet Success Percentage results of sent Interest Packets (2/2): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network, and a distance of 40 units between access point nodes of different WiFi networks.

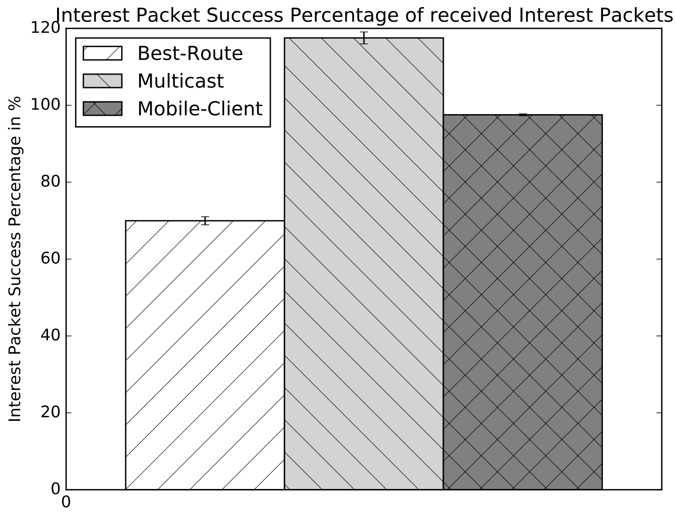

Fig. 5.25: Interest Packet Success Percentage results of received Interest Packets (1/2): One access point node per WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

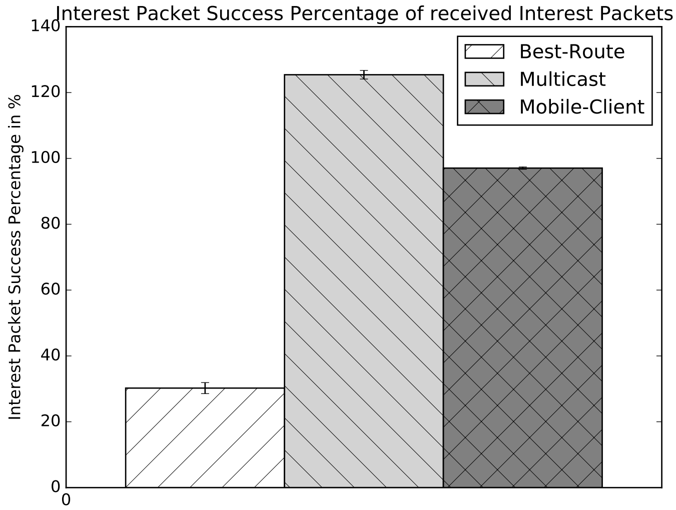

Fig. 5.26: Interest Packet Success Percentage results of received Interest Packets (2/2): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network, and a distance of 40 units between access point nodes of different WiFi networks.

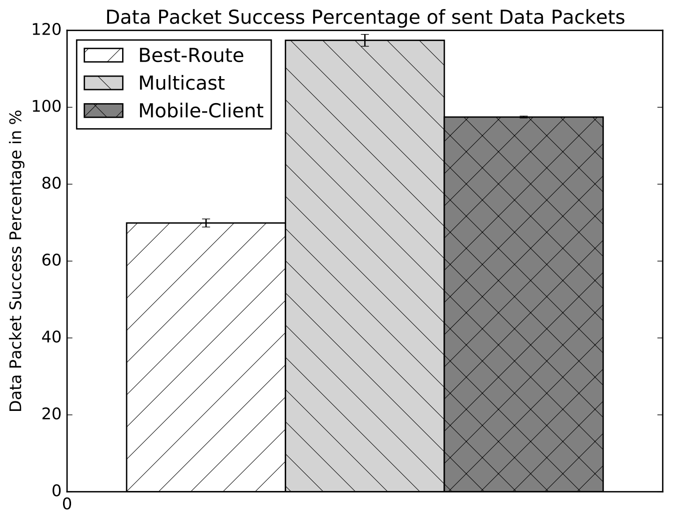

Fig. 5.27: Data Packet Success Percentage results of sent Data Packets (1/2): One access point node per WiFi network and a distance of 40 units between access point nodes of different WiFi networks.

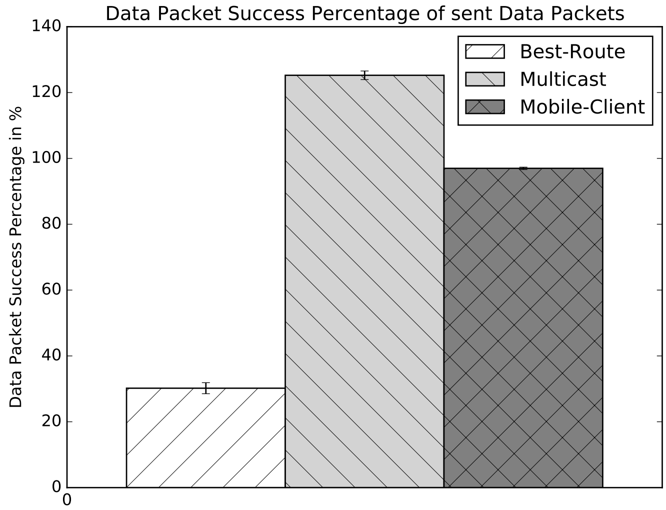

Fig. 5.28: Data Packet Success Percentage results of sent Data Packets (2/2): Three access point nodes per WiFi network, a distance of 40 units between access point nodes of the same WiFi network, and a distance of 40 units between access point nodes of different WiFi networks.

These Packet Success Percentage results are calculated the same way as already described for the results of the Interest Satisfaction Ratio values in section 5.5.1. The result bar charts in figures 5.23, 5.24, 5.25, 5.26, 5.27 and 5.28 show that the Packet Success Percentage values for the Multicast Forwarding Strategy achieve the highest Packet Success Percentage results. The Packet Success Percentage values for the Multicast Forwarding Strategy are higher than 100 % in each figure. This means more packets are sent and received where a higher load on the network is produced due to duplicate packets.

Furthermore, the Packet Success Percentage values for the Mobile-Client Forwarding Strategy achieve good results. The Packet Success Percentage values achieve nearly 100 % in each figure. This means the Mobile-Client Forwarding Strategy achieves a good result and produces a lower load on the network compared to the Multicast Forwarding Strategy.

Finally, the Packet Success Percentage values for the Best-Route Forwarding Strategy achieve the worst results. These results show, that the Best-Route Forwarding Strategy has the highest amount of packet loss and that the Best-Route Forwarding Strategy cannot handle mobility of mobile client nodes well like already described in section 5.5.1.

The Interest Packet Success Percentage of the Multicast Forwarding Strategy of sent Interest Packets is almost 200% in figure 5.23. This is because one access point node per WiFi network exists and 2 WiFi networks are used in the evaluation scenario. This means two access point nodes exist and the Multicast Forwarding Strategy can send Interest Packets to all available two links. Furthermore, the Interest Packet Success Percentage of the Multicast Forwarding Strategy of sent Interest Packets is almost 600 % in figure 5.24. This is because three access point nodes per WiFi network exist and 2 WiFi networks are used in the evaluation scenario. This means 6 access point nodes exist and the Multicast Forwarding Strategy can send Interest Packets to all available 6 links. These results also show that more Interest Packets are sent than received. This is because of packet loss when using a loss model.

5.5.5 Other Metrics

Additional metrics like the delay, duplicate packets, missed packets and the hop count are implemented for the practical part of this master thesis. Results for these metrics are calculated for the evaluation scenario as well. But these results are not shown in this section like already described in section 5.4.5.

5.6 Discussion

Finally, the results of the performance evaluation in this chapter show that the Mobile-Client Forwarding Strategy performs well as a forwarding strategy for mobile client nodes within the created evaluation scenario when comparing the results of different metrics with different forwarding strategies.

The idea for this master thesis is to design a new forwarding strategy, which performs well for mobile client nodes while the research questions presented in section 3.3 are taken into account. An aim of this master thesis is to increase the Interest Satisfaction Ratio in loss-prone NDN networks which is defined in research question 1 in section 3.3.3. The Interest Satisfaction Ratio can be increased when data is requested redundantly by a content consumer when sending redundant Interest Packets. But this causes an amount of redundant Interest Packets and Data Packets. For this reason, other research questions have to be considered as well. The results of the Interest Satisfaction Ratio metric are presented in section 5.5.1. These results show that the Mobile-Client Forwarding Strategy performs well compared with the other forwarding strategies. The Interest Satisfaction Ratio results for the Multicast Forwarding Strategy show similar results.

Another aim of this master thesis is to reduce the number of redundant Interest Packets and Data Packets which are defined in research questions 2 and 3 in sections 3.3.4 and 3.3.5. The number of redundant Interest Packets can be reduced when data will not always be requested redundantly by a content consumer, due to sending redundant Interest Packets. Redundant Interest Packets also cause redundant Data Packets. The results of the Packet Success Percentage metric are presented in section 5.5.4. The Mobile-Client Forwarding Strategy does not always send redundant Interest Packets, but only when needed to reduce packet loss. These results show that the Mobile-Client Forwarding Strategy performs well compared with the other forwarding strategies as well. The Multicast Forwarding Strategy always sends redundant Interest Packets and this causes a higher load on the network due to duplicate packets. The next chapter concludes with a summary.

6 Summary

This chapter concludes with a summary and final remarks. This master thesis aims to design and implement a new forwarding strategy, called Mobile-Client Forwarding Strategy, which works well for mobile clients in an NDN network. NDN is used in the network as a network layer protocol, instead of using IP. Furthermore, a simulation scenario is created for the performance evaluation of different forwarding strategies within the practical part of this master thesis. Therefore, the network simulator ndnSIM is used.

Chapter 1 introduces the topic of this master thesis. Chapter 2 describes the technologies and background information this master thesis is based on where basic concepts of NDN are introduced. Therefore, the communication in NDN, forwarding in NDN and mobile clients in an NDN network are introduced in more detail. Chapter 3 explains the aim of this master thesis with the research questions and an idea for designing a new forwarding strategy. Chapter 4 describes the implementation of the practical part of this master thesis, where a new forwarding strategy called Mobile-Client Forwarding Strategy is designed and implemented. Furthermore, additional extensions for the Mobile-Client Forwarding Strategy and the evaluation scenario are implemented as well. Finally, the implemented evaluation scripts for the performance evaluation are introduced. Chapter 5 presents the performance evaluation of the Mobile-Client Forwarding Strategy compared with the forwarding strategies Multicast and Best-Route. These results show, that the Mobile-Client Forwarding Strategy performs well compared to other forwarding strategies.

Bibliography

[AfMZ12] A. Afanasyev, I. Moiseenko, L. Zhang: ndnSIM: NDN simulator for NS-3. Tech. Rep. NDN-0005, Revision 2, University of California, Los Angeles (2012).

[AmCM17] M. Amadeo, C. Campolo, A. Molinaro: A Novel Hybrid Forwarding Strategy for Content Delivery in Wireless Information-Centric Networks. In: *Computer Communications*, vol. 109 (2017), pages 104 – 116.

[ASZZ+16] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, I. Moiseenko, Y. Yu, W. Shang, Y. Li, S. Mastorakis, Y. Huang, J. P. Abraham, E. Newberry, S. DiBenedetto, C. Fan, C. Papadopoulos, D. Pesavento, G. Grassi, G. Pau, H. Zhang, T. Song, H. Yuan, H. B. Abraham, P. Crowley, S. O. Amin, V. Lehman, M. Chowdhury, L. Wang: NFD Developer's Guide. Tech. Rep. NDN-0021, Revision 7, University of California, Los Angeles (2016).

[Berg17] S.-N. Bergmann: Github repository. https://github.com/sandra-nadine-b (2017).

[Bund11] Bundesnetzagentur: Dauer Handy-Telefonate. http://www.bundesnetzagentur.de/SharedDocs/Pressemitteilungen/DE/2011/110728DauerHandygespraeche.html (2011).

[CaBD02] T. Camp, J. Boleng, V. Davies: A Survey of Mobility Models for Ad Hoc Network Research. In: *Wireless Communications and Mobile Computing*, vol. 2, no. 5 (2002), pages 483 – 502.

[CZMC+17] A. Compagno, X. Zeng, L. Muscariello, G. Carofiglio, J. Augé: Secure Producer Mobility in Information-Centric Network. In: *ICN '17 Proceedings of the 4th ACM Conference on Information-Centric Networking, ACM New York* (2017), pages 163 – 169.

[Foun17] N. S. Foundation: NS-3. https://www.nsnam.org/ (2017).

[GSKS+11] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, J. Wilcox: Information-Centric Networking: Seeing the Forest for the Trees. In: *HotNets-X Proceedings of the 10th ACM Workshop on Hot Topics in Networks, ACM New York* (2011).

[JSBP+09] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, R. L. Braynard: VoCCN: Voice-over Content-Centric Networks. In: *ReArch '09 Proceedings of the 2009 Workshop on Re-architecting the Internet, ACM New York* (2009), pages 1 – 6.

[JSTP+09] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard: Networking Named Content. In: *CoNEXT '09 Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, ACM New York* (2009), pages 1 – 12.

[KCKR13] K. Kim, S. Choi, S. Kim, B. Roh: A Push-Enabling Scheme for Live Streaming System in Content-Centric Networking. In: *CoNEXT Student Workshop '13 Proceedings of the 2013 Workshop on Student Workshop, ACM New York* (2013), pages 49 – 52.

[LMRT+13] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, H. Hellwagner: Adaptive Streaming over Content Centric Networks in Mobile Networks using Multiple Links. In: *2013 IEEE International Conference on Communications Workshops (ICC), IEEE* (2013).

[LMTH14] S. Lederer, C. Mueller, C. Timmerer, H. Hellwagner: Adaptive Multimedia Streaming in Information-Centric Networks. In: *IEEE Network, IEEE*, vol. 28, no. 6 (2014), pages 91 – 96.

[LosA17] U. of California in Los Angeles: ndnSIM documentation - NS-3 based Named Data Networking (NDN) simulator. http://ndnsim.net/2.4/ (2017).

[MaAZ17] S. Mastorakis, A. Afanasyev, L. Zhang: On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation. In: *ACM SIGCOMM Computer Communication Review, ACM New York*, vol. 47, no. 3 (2017), pages 19 – 33.

[MAMZ16] S. Mastorakis, A. Afanasyev, I. Moiseenko, L. Zhang: ndnSIM 2: An updated NDN simulator for NS-3. Tech. Rep. NDN-0028, Revision 2, University of California, Los Angeles (2016).

[MeMB00] A. Medina, I. Matta, J. Byers: BRITE: A Flexible Generator of Internet Topologies. Tech. Rep. www.cs.bu.edu/brite/, Boston University (2000).

[MoPH17] P. Moll, D. Posch, H. Hellwagner: Investigation of Push-based Traffic for Conversational Services in Named Data Networking. In: *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE* (2017).

[PoRH17a] D. Posch, B. Rainer, H. Hellwagner: SAF: Stochastic Adaptive Forwarding in Named Data Networking. In: *IEEE/ACM Transactions on Networking, IEEE*, vol. 25, no. 2 (2017), pages 1089 – 1102.

[PoRH17b] D. Posch, B. Rainer, H. Hellwagner: Towards a Context-Aware Forwarding Plane in Named Data Networking supporting QoS. In: *ACM SIGCOMM Computer Communication Review, ACM New York*, vol. 47, no. 1 (2017), pages 4 – 14.

[Posc16] D. Posch: Towards Effective Multimedia Dissemination in Information-Centric Networks. Dissertation, Alpen-Adria-Universität Klagenfurt, Fakultät für Technische Wissenschaften (2016).

[QCTD16] X. Qiao, J. Chen, W. Tan, S. Dustdar: Service Provisioning in Content-Centric Networking: Challenges, Opportunities, and Promising Directions. In: *IEEE Internet Computing, IEEE*, vol. 20, no. 2 (2016), pages 26 – 33.

[ScKr15] K. M. Schneider, U. R. Krieger: Beyond Network Selection: Exploiting Access Network Heterogeneity with Named Data Networking. In: *ACM-ICN '15 Proceedings of the 2nd ACM Conference on Information-Centric Networking, ACM New York* (2015), pages 137 – 146.

[SMJI15] L. Sun, I.-H. Mkwawa, E. Jammeh, E. Ifeachor: Guide to Voice and Video over IP: For Fixed and Mobile Networks. Springer Publishing Company (2015).

[StRi12] M. Stoffers, G. Riley: Comparing the ns-3 Propagation Models. In: *2012 IEEE 20th International Symposium on Modeling Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE* (2012).

[TsXy11] C. Tsilopoulos, G. Xylomenos: Supporting Diverse Traffic Types in Information Centric Networks. In: *ICN '11 Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, ACM New York* (2011), pages 13 – 18.

[ZABZ16] Y. Zhang, A. Afanasyev, J. Burke, L. Zhang: A Survey of Mobility Support in Named Data Networking. In: *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE* (2016).