Konstantin Niedermann

Network Coding based Interactive Multiview Video Streaming in ICN Networks

Bachelors's Thesis

University of Bern

CDS
Head: Prof. Dr. Torsten Braun

Supervisor: Dr. Eirina Bourtsoulatze

Bern, February 2015

Abstract

In Content Centric Networks (CCN), data transmission is based on content names instead of identifiers as in IP networks. In a CCN network users communicate with their connected neighbours to search for data. This approach can be used for mobile networks. Considering such wireless mobile networks, network coding (NC) is able to improve the throughput in the network. Since mobile Internet traffic is increasing fast, new solutions are required that would increase the overall throughput of the wireless network.

In this thesis we consider a CCN network with a free viewpoint video streaming application. We propose two network coding schemes for the delivery of video data and evaluate their performance in terms of the average video quality. Matlab simulations are used to compute the average quality of the users view. With these simulations, we can show that applying NC rules provide a better quality in the displayed views for some communication scenarios.

Contents

List of Figures

Acknowledgement

Many people have participated directly or indirectly in the development of this bachelor thesis. I am very grateful to all of them!

I would like to specially thank my supervisor Dr. Eirina Bourtsoulatze for her support and advice during the whole working process. I would like to extend my gratitude to Prof. Dr. Torsten Braun for giving me the opportunity to write this thesis in his research group. I thank my friend Samuel Hool for his help with the design of the figures in this thesis.

Last but not least, I am very thankful for the support and encouragement my family and friends have provided.

1 Introduction

The usage of the Internet has changed since the early days. Not just has the amount of data transmitted been increasing over the past years, but clients are rather interested in retrieving data - no matter at which location this data has been stored. The Internet has become a cloud for information and services and the average user is interested in specific content, independent of its location or host. However, the architecture is still the same as it was in the early years of the Internet although the number of mobile devices such as tablets and smart phones has been and still is increasing. More flexible approaches are required in order to deal with the increasing amount of information communicated in today's networks and the highly dynamic mobile environments.

Its a fact, that a bigger part of today's mobile Internet traffic is caused by video streaming. If we want to reduce the global Internet traffic without giving something up, finding a better suited Internet architecture with higher throughput will be beneficial.

Device to device (D2D) communication allows users to exchange data reducing thus the base station load. The concept of D2D communication integrates well with the Information Centric Network (ICN) architecture, since users may request the content from any user within their communication range. The biggest change to IP-Routing is that in ICN-Networks, nodes are routing content considering its name and not considering its storage location. One specific ICN architecture, which we will be using in this thesis, is Content Centric Networking (CCN). In CCN users communicate with their connected neighbours to search for data. Either the neighbours may help with data stored in their cache, or they can forward the request to other users in the network with the help of routing tables. In this setting, throughput can be improved by using Network Coding

(NC). Both approaches, CCN and NC have shown in previous works, that they can be used to improve throughput in a wireless network.

In this thesis we want to achieve a higher throughput in a mobile wireless network by using NC in a CCN Network. Therefore we propose NC algorithms and compare their results in Matlab simulations to the results in a network without NC.

2 Background

2.1 Content Centric Networking

The change of usage of the Internet since early days has motivated the development of a new Internet architecture. In contrast to Internet's usage 20-30 years ago, when it has been used to contact a specific host for some service, users nowadays are interested in getting data fast, not focusing on the location where the data comes from. The increasing demand for data is another reason for searching for a more suitable routing architecture. In contrast to current networks, which are host-centric and communication is based on IP addresses, the CCN architecture is an information-centric approach in which named data objects (NDOs) are needed. NDOs are independent of location, storage method, application program and transportation method. Therefore NDOs keep their name, and thus their identity, regardless of their location. This means, that any node holding a copy of a certain data object can supply it to any requester. In a host-to-host model, a client's request has to go through the whole network between requester and server to arrive at it's destination. Then the data has to be transmitted back along the whole path. This means, that for instance when a user wants to watch a live video on the Internet, he has to get the data from the video server, even when his room mate or neighbour is watching that video too. Since CCN supports content caching, the user will be able to get the data independently of his/her location. In the example above, the user will be able to retrieve the data from his room mate or neighbour reducing the networks latency and bandwidth cost.

In CCN networks, there are two packet types: Interests and Data. A consumer asks for content by sending its Interest over all available connections (to other consumers and basestations of the network). Each node operates on three data structures that are of Interest for our work and will be explained in the following sections. [2], [3] ,[4]

Content Store

Any node in a CCN-Network has some free cache to buffer data that has past the node. Data that has been stored in a nodes Content Store (CS) will stay saved as long as possible.

Any node receiving the Interest and having a copy of the requested data stored in its (CS) will send a data packet in response. This process consumes the Interest.

Pending Interest Table

The Pending Interest Table (PIT) is used to keep track of nterests that could not be answered immediately and have been forwarded upstream.

When a node receives an Interest that can not be answered by data from the CS, the node looks up its PIT for an entry for the same data. If there is no entry matching the requested data, this node creates a new entry in its PIT, and the Interest is routed upstream using the Forwarding Information Base (FIB). Else, if an entry for the corresponding data already exists in the PIT, the outgoing face for the requesting user is removed from the entry in the FIB, since the data can't be found at this node for sure. Furthermore the face of the new requester is added to the entry corresponding to the data in the PIT. Since an Interest in this data has already been sent upstream, the node needs to make sure that when the requested data packet arrives, a copy of that packet will be sent to the face where the new Interest arrived. Since routing of the Interest has already been done before, the node just has to remember the Interest to be able to respond later if he receives the requested data.

This process consumes the Interest. The PIT consist of the face of the requester (multiple requester possible) and the name of the data. With help of the PIT, nodes are able to remember Interests of users, and may respond to them as soon as they receive the corresponding data.

When a user receives some data object, it is forwarded on all faces in the entry for the data in his PIT.

In other words, as Interest packets are routed, they are forwarded upstream towards potential data sources, and leave a trail of 'bread crumbs' for a matching data packet to follow back to the original requester. Each entry in a PIT-entry is a 'bread crumb' for a matching data packet and is consumed (erased) as soon as it has been used to forward corresponding data.

Forwarding Information Base

The FIB is used to manage outgoing faces that lead to potential sources for the requested data. It consists of a list of outgoing faces (potential sources) of the node for each data object.

When a node receives an Interest and neither has the matching data in its CS, nor has a corresponding entry in the PIT, the node has to send forward the Interest. Therefore the FIB is being used. The face of the requester is removed from the list for the requested data object in the node's FIB-TABLE (since it is no source for the requested data for sure). Then the Interest for the requested data is forwarded to all remaining faces of the corresponding FIB-entry. Additionally a new PIT, entry is created from the Interest and its arrival face to make sure the data will be sent forward to the requesting user on arrival.

In Figure 2.1 you can see a typical example for the CCN application. Some application of a first requester on the bottom left is requesting some data. A PIT entry is created with the face of the application and the data name. The Interest message is sent to the next router ① generating entries in the requesters FIB and the routers PIT, since the router can't respond with the corresponding data. The Interest is routed further to the next router ② and from there to the source node ③ leaving entries in the FIB and PIT of the intermediate Router. The source, which has a copy of the requested

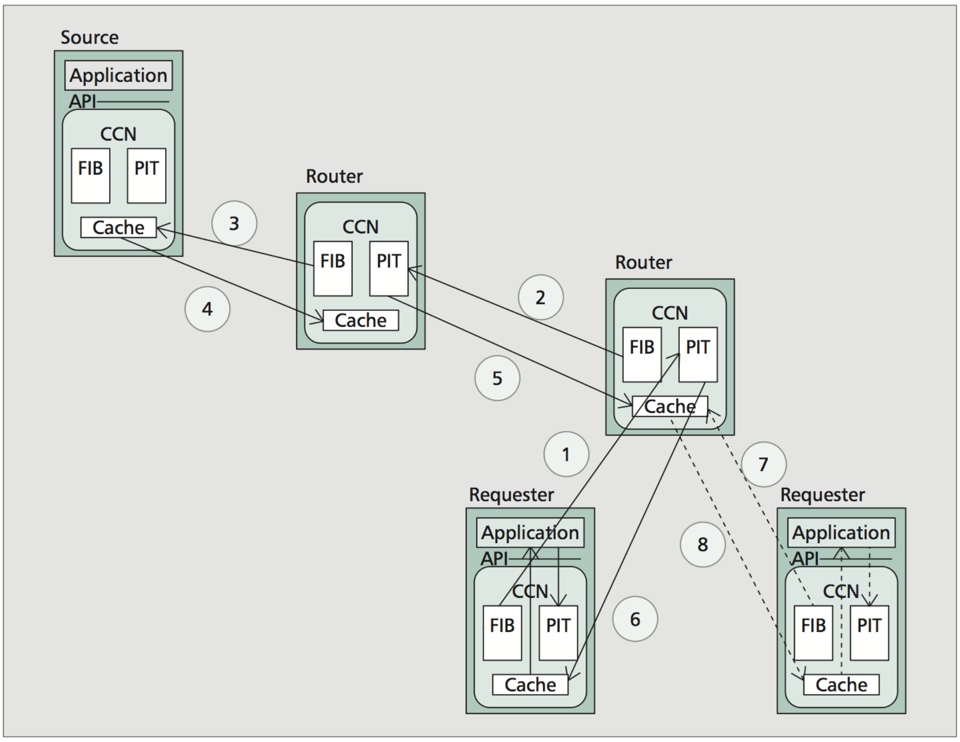

Figure 2.1: CCN communication model [1]

file in its cache (CS) may directly answer with the data object ④. Upon receiving the data each intermediate node will save the packet in its cache and forward it according to the PIT entries ⑤,⑥. The application of the requester node may now access the data from the cache. Using content caching a request for the same data by another Requester may be served without having to route the Interest towards the source and waiting for the data to be sent back. Each intermediate node between the requester and the source has a copy of the requested data object in its CS, and may answer right with the content ⑧ when some Interest for the same data arrives ⑦.

2.2 Network Coding

Network Coding (NC) is an alternative to the state-of-the-art routing. In state-of-the-art routing, each intermediate node of the network simply stores and forwards the information received. In contrast, networks using NC allow nodes to generate output data by combining and encoding multiple data objects. Sending a new packet generated by NC leads to more data diversity in the network. Using NC brings several more advantages such as increased throughput, robustness to losses and reduced delay. [5] However, when a user receives an encoded packet he needs to decode it before being able to use the data. [6]

Random Linear NC

If an encoded copy of data packets $a_1, a_2, a_3, ..., a_n$ is requested, the sender will combine these packets with a random linear combination with coefficients λ_i of a finite field F

$$A = \sum_{i=1}^{n} (\lambda_i * a_i).$$

The vector $\boldsymbol{\lambda}$= $[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n]$ is called the *global encoding vector* of A, which is sent along with A as side information in its header. The overhead, that is produced this way is negligible if packets are sufficiently large. Therefore, the size of the encoded packet is:

$$size(A) = max\{size(a_1), size(a_2), ..., size(a_i)\}$$

A receiver node collects packets and stores them in the cache. If it has n packets with linearly independent global encoding vectors, it is able to decode the packets. This can be done by Gaussian elimination.

In Figure 2.2 you can see the Butterfly Network. There are two source nodes (at the top), two destination nodes (at the bottom) and two intermediate nodes (in the center). The two destination nodes at the bottom

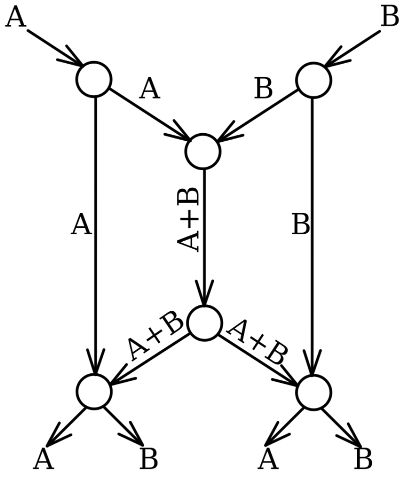

Figure 2.2: Butterfly Network

request data A and B, and the source nodes provide each one of the data elements. If only common routing protocols without NC are allowed, then the central link between the two intermediate nodes would only be able to submit data A or B in one time slot. For instance if we send data A through the middle link, the left destination node would receive data A twice, but not B. The same problem would occur at the right destination node if data B would be sent through the central link. Using NC we can send a combination (A+B) of both requested data elements through the middle link. Since both destination nodes have stored one of the combined data elements (A or B), they are able to decode the packet and may reconstruct each data element. For further information about NC we refer to [7]

2.3 Free Viewpoint Video

Free Viewpoint Video is an upcoming multimedia application. It allows users to select a displayed view by switching between different camera positions of a scene. The views are captured by an array of cameras. They record a scene of interest from different camera perspectives. A displayed view may either correspond to one of these physical cameras, or to any virtual view, synthesized by two physically acquired camera views. With the help of depth-image based rendering (DIBR) any itermediate view between two physical camera positions can be synthesized. The quality of a synthesized view u depends on the closest physical camera views received to the left and right and can be calculated as follows:

$$Q(u) = Q_{max} - D_u(v^l, v^r)$$

D_u is the distortion function of view u according to the next physical view to the left (v^l) and right (v^r)

$$D_u(v^l, v^r) = \gamma e^{\alpha_u (v^r - v^l)} (e^{\beta_u * min(v^r - u, u - v^l)} - 1)$$

α_u, β_u and λ are multiplicative coefficients to that depend on the video sequence and the distance between two physical cameras. For further details on the distortion model and for the specific meaning of each parameter we refer the reader to [8].

3 Interactive Free Viewpoint video streaming with Network Coding

3.1 Problem setup

In this chapter, we describe in detail the problem considered in this thesis. In particular, we consider an Interactive Free Viewpoint video streaming scenario in a wireless communication environment.

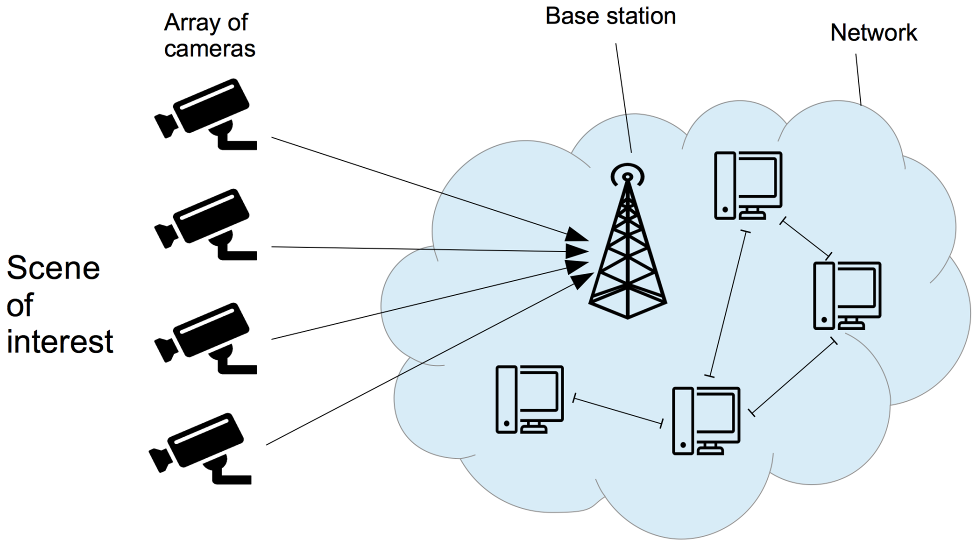

Figure 3.1: Interactive free viewpoint video streaming in a wireless network

Figure 3.1 shows the structure of an Interactive Free Viewpoint video streaming scenario. In this scenario we have an array of **N** cameras that capture the scene of interest. Each camera then transmits its data to a base station, which is located in a local wireless network and which further disseminates the video data to a set of wireless users. The network users are equipped with multimedia enabled devices and may communicate with the base station, as well as with other users within their communication range **r**. Each user independently selects a view of the scene. These views can either be physical views directly captured by one of the cameras in the array or a virtual view which has to be synthesized from views of two physical cameras. The users generate interests for data objects corresponding to their chosen view of the scene. These data objects may be obtained directly from the base station, or from any user in the network within the user's communication range **r**.

If a large communication range is chosen, the users may communicate with users which are physically further away and may therefore statistically gain access to more data. On the other hand, there is more interference between wireless communications in the local communication area of the user.

We assume a simple interference model where, given a data transmission session from user u_1 to user u_2, any transmission from another user within the range of u_2 causes interference for the link u_1 - u_2. This interference model is illustrated in Fig. 3.2 While the two nodes in the center are using their connection to transfer data or interests (blue arrow), other connections within the communication radius of the involved nodes have to comply with some restrictions. All nodes within the radius of the sender node (purple circle) may not receive any data from other users (red arrows), since the signal that would be sent would interfere with the signal of the active connections from the sender node of the data transfer of the blue arrow. On the other hand, any user within range of the receiver node (yellow circle) may not send any data (red arrows), since there would occur interference with the active (blue arrow) connection.

Following these two rules, some other connections may be possible (green arrows). For some user to user links, a transmission from one to the other may be possible, while the other way round, the connection would cause interference (green-red arrows).

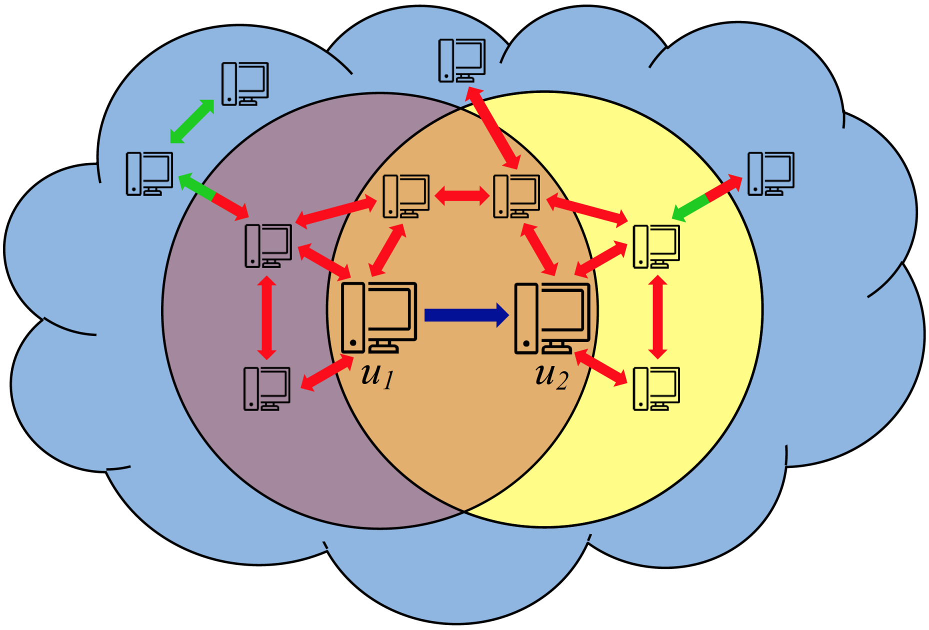

Figure 3.2: Wireless communication model

Please note that for clarity reasons, not all possible or impossible connections between users are marked.

Given this interference model, it is clear that the communication range r for each user, which depends on the power lever for each transmission, is an important parameter that influences the overall throughput of the network and therefore the quality of the views delivered to the users. If a large communication range is chosen, the users may communicate with users which are physically further away and may therefore statistically gain access to more data. On the other hand, there is more interference between wireless communications in the local communication area of the user.

3.2 Proposed Network Coding Algorithms

In order to optimize the communication efficiency within the network, we will consider the use of network coding for data delivery. In particular, our aim is to maximize the average quality of the video delivered to the users by increasing the throughput of the wireless network through the use of direct user to user (U2U) links and by taking advantage of the data diversity that is introduced by the network coding. We propose two communication protocols that use network coding and compare them to the traditional communication scenario that does not use network coding. Our solution relies on the intuition that, on average, it may be beneficial for users to send/request packets that are encoded with the help of NC instead of repeatedly sending multiple copies of the same data object to several users.

The proposed rules focus on the behaviour of the base station. Normal users in the network only forward NC-packets, but do not generate new ones. This design choice has several reasons. Since the base station has stored all known data objects, it is able to combine any number of views and send any encoded packet. Furthermore, the base station has a global knowledge about the network. It knows the network topology and every user's requested view. With the help of this knowledge, the base station can predict which user may have access to what data object in the future and will be able to decode some NC-packet. On the other hand a single user does not know much about the network. It knows its outgoing faces and has some knowledge about the data requested by the users within its communication range. However, for taking advantage of the benefits described in [5] in this wireless scenario, users need to know more about the topology and requests of other users.

There are two types of requests. One for a physical view, and one for a virtual view. In each rule the base station handles each type of request in a different way. In the following sections we will describe in detail the proposed rules.

3.2.1 Rule 1

As previously mentioned, the NC rules handle the case when the base station receives a request from a user. Upon receiving an interest, the base station needs to distinguish between requests for a physical (real camera position) and requests for a virtual view (synthesized from two physical cameras). In this rule, the base station, which distributes data from the cameras, handles incoming requests as follows. If the incoming request is for a physical view, the base station looks in the near network of the requesting user for some other user that already received some uncoded data from the base station that does not match with the requested data. If this search is successful, the base station combines the two data files to an NC-packet and sends it to the requesting user. Due to this procedure, the base station sends a new data packet in the network and thus increases its data diversity. If the search is not successful, the base station will just send an uncoded packet with the requested data. The second case, where the base station receives an interest for a virtual view, it will send a NC-packet consisting of the two neighbouring physical views. This rule is summarized in pseudo code in Algorithm 1.

Algorithm 1 Receive request at base station - Rule 1

```
incoming request r at the base station from user u
if $isPhysicalViewRequest(r)$ then
    $c_1$ is the requested camera data
    if base station has sent data for $c_i$ to a user connected to u then
        $c_2 \leftarrow c_i$
        send a linear combination $\lambda_1 * c_1 + \lambda_2 * c_2$ to the user u
    else
        send $c_1$ to the user u
else                                                    ▷ virtual view request
    $c_1$ and $c_2$ is the requested camera data
    send a linear combination $\lambda_1 * c_1 + \lambda_2 * c_2$ to the user u
```

Let us consider an example for a better understanding of the rule. Let us assume we are in a network with one base station and two users that are able to communicate with each other and both want to generate some virtual view between camera positions A and B. The base station will receive a request for a virtual view from both of them, and answer with two randomly generated NC-packet, which is a linear combination of A and B. After gaining the first data object, the receiver can not display his requested view as two linearly independent NC packets are required to decode the views A and B. In a next step the users may communicate with each other and exchange the data they received from the base station, while the base station resources may be allocated to other users in the network. Given that the size of the Galois field is sufficiently large, the two NC packets will be linearly independent with high probability. Thanks to this transmission between the users, they will be able to decode the NC-packet and display the view.

In the non-NC case, where the base station will first send one data packet (A or B) to each user at random, it may be the case that both users receive the same data. If this happens, the users will have to contact the base station again for obtaining the other data object. Since the views must be delivered within some given time constraints, it may happen that the users do not have sufficient time to obtain all the necessary data in order to reconstruct the requested view. It is clear that in this case the base station resources are not used efficiently and the users cannot take advantage of the U2U communication since there is not enough data diversity in the network.

If the base station receives a request for some physical camera position B, in a non-NC scenario it would just send back data B. According to this rule however the base station searches for any known data besides B in the close environment of the receiver node and uses NC to send a combination of these two data elements (i.e. B+D if data D has been sent to some neighbour). Since the base station may have sent other data objects to neighbouring users, it knows to which data objects the receiving user has access. In a next step the receiver will create requests for data that helps to decode. This can be uncoded data B or D or any linearly independent coded data B+D. If the base station's search for known data in the close environment of the receiver was not successful, it will send data B to the user.

An advantage of this choice is to have more data diversity in the network, forcing the users to communicate with each other.

3.2.2 Rule 2

In Rule 2, again the base station needs to distinguish between the two types of interests. If it is a request for a physical view, the base station will send an uncoded packet with the requested data. In the second case, where the base station receives an interest for a virtual view, it will send a NC-packet consisting of the two neighbouring physical views. This rule is summarized in pseudo code in Algorithm 2.

Algorithm 2 Receive request at base station - Rule 2

```
incoming request $r$ at the base station from user $u$
if $isPhysicalViewRequest(r)$ then
    $c_1$ is the requested camera data
    send $c_1$ to u
else                                                    ▷ virtual view request
    $c_1$ and $c_2$ is the requested camera data
    send a linear combination $\lambda_1 * c_1 + \lambda_2 * c_2$ to the user u
```

Resuming the example above from rule 1, the base station again receives a request for some virtual view between two physical camera positions A and B. Again the base station will take advantage of NC and combine two data packets A and B to be able to send a linear combination of both data elements. The reason to do this does not differ from the motivation explained in rule 1. The receiving node has more possibilities to reconstruct and display the virtual view, since a larger number of different packets are able to help to decode the NC-packet.

Upon receiving an interest for a physical camera position B, the base station will just send an uncoded data packet containing the data for camera position B. The reason for this choice is to compare results of the case in rule 1(sending some encoded camera positions) with the results of the simulation with rule 2 (answering with a non-NC-packet). Users gaining an uncoded packet in this case, will be able to display their view and will no more send any interests to other users besides the ones they are forwarding for other users. This leads to less traffic in the network and may therefore be a better solution than in rule 1. However in networks with low data diversity, for example in a network where camera view B, C and virtual views between B and C are often requested, it may perform worse.

3.2.3 Rule 3 - no Network Coding

The two proposed NC rules are compared to the case where the base station and the rest of the network do not use any NC. When receiving an interest for a virtual view, the base station will answer with data for one of the requested physical cameras (chosen randomly). Then a new request for the second data is sent to the base station, which is handled right after having successfully submitted the first data object.

Upon receiving an interest for a physical camera position, the base station will send an uncoded data packet containing data for the requested camera position. This procedure is summarized in Algorithm 3.

Algorithm 3 Receive request at base station - no NC

```
incoming request r at the base station from user u
if isPhysicalViewRequest(r) then
    $c_1$ is the requested camera data
    send $c_1$ to u
else                                          ▷ virtual view request
    $c_1$ and $c_2$ is the requested camera data
    $c_r = rand(c_1, c_2)$
    send $c_r$ to u
```

4 Simulation

4.1 Simulation setup

In this chapter, we evaluate the performance of the algorithms proposed in Chapter 3.2 with respect to the quality of video delivered to the users. In particular we consider the scenario of Figure 3.1. We assume that the users are spread over a square area with the base station situated at the center of the square. Without loss of generality, we consider that all the distances and the communication radius are normalized with respect to the dimensions of the area. In order to evaluate the communication scenario examined in this thesis, we implemented an event driven simulator in Matlab with the following preferences.

Users have a communication radius $\mathbf{r} \in (0, \sqrt{2})$. A radius $\mathbf{r} = 0$ means that there is no communication possible with other users. With growing $\mathbf{r}$ a user may communicate with more users and thus may more easily discover information. However, the more users there are within the communication radius $\mathbf{r}$, the more interference occurs. Considering the communication radius has the maximum value $\mathbf{r} = \sqrt{2}$, on the one hand, the user can send and receive interest messages and data to and from all users in the network, but on the other hand, using connections between users causes maximal interference with other connections allowing only one connection between users to be active at a time.

At the start of the simulation at time t_0 each user chooses a view which may either be a physical or a virtual camera view. In the simulation we consider that the users select the views according to a Gaussian distribution. A simulation with Gaussian distributed views reflects a real world scenario, since often in free viewpoint video streaming scenarios, there are some more popular and some barely requested views of the scene.

Furthermore, the transmission speed of connections can separately be modified. The user to user (U2U) communication speed and the transmission speed of connections between users and the base station (U2B/ B2U) are variable parameters that we modify and study in this work. In addition the number of parallel U2B and B2U connections can also be managed.

In a CCN network, users have to forward the interest, they received. If an Interest or data packet can not be forwarded because of interference, it is scheduled and after a delay of 10 ms, submission is retried. When a retry fails, the transfer is rescheduled with a linearly increased delay.

Another parameter is the Galois field size q. The size of the Galois field is defined by 2^g, which is used for encoding and decoding packets with NC. In these simulations we set the size of the Galois field to 2^8 which is large enough to guarantee that the probability of generating two linearly dependent packets is negligible. [5].

In the simulation we consider a source rate of the cameras of 360 kbps. They generate one chunk of video data of size 360 kb, while each interest for some data has the size of 30 b.

4.2 Evaluation

In this chapter we discuss the results of simulations for various scenarios. At the beginning of the simulation every user tries to send requests to the base station and to connected users. Depending on interference in wireless communications and limited number of parallel communication channels with the base station some requests may be sent immediately and others may be scheduled for a later time instant. The base station has stored the chunks for video data of every camera for the first second. These packets have to be delivered to the users within one second, otherwise the data is useless. Therefore, after one second, the simulation is stopped and the average quality of the users views is computed. The same scenario is run three times - once for each NC-rule and once in a network without

NC. Then the return value - the average quality of the users view can be compared.

4.2.1 Toy network

First, we consider a scenario with four users with four fixed requests for virtual views and fixed communication radius. In particular, there are two pairs of two users in the network. The two users in each pair can communicate with each other and have no connection to the other pair. Additionally, both users in each pair are interested in reconstructing a virtual view between the same physical views and, thus request the same physical views.

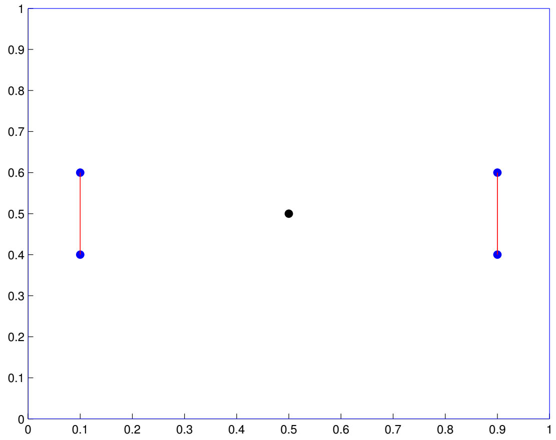

Figure 4.1: Toy network with four users

Figure 4.1 illustrates the topology of this scenario. Blue points represent users in the network, while the red lines stand for possible connections between users. The black point in the middle stands for the base station in this example network. We want to analyse the average quality of the views delivered to the users for the network coding algorithms presented in Section 3.2.

At the start of the simulation, each user will try to send an interest for its requested view to the base station and to his neighbour. Depending on the activated NC rule, the base station handles the incoming requests as described in Section 3.2. The users behave with respect to CCN as explained in Section 2.1.

Influence of U2U link speed

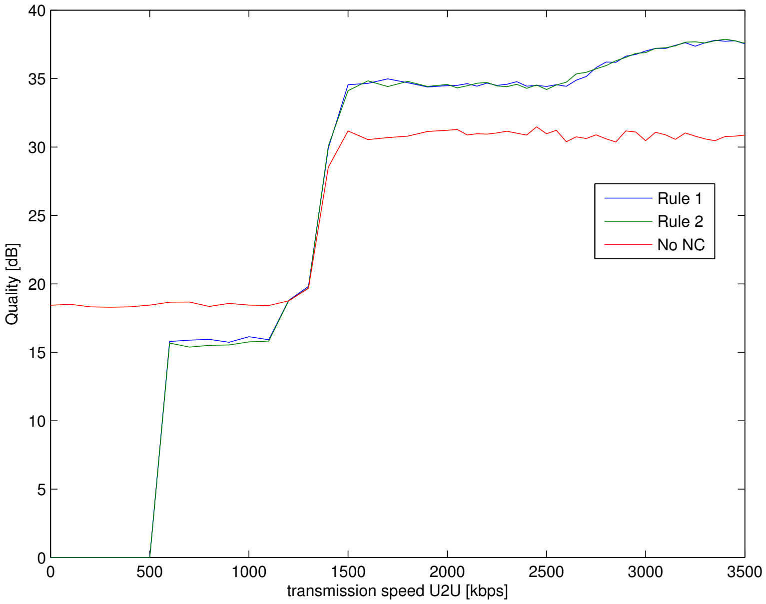

Figure 4.2: Average video quality versus the transmission speed of the U2U communication links in the toy network of figure 4.1

A plot over the transmission speed of U2U links with fixed B2U links' speed is shown in Figure 4.2. In this scenario, the maximum number of parallel connections with the base station is set to two and their transmission speed is set to 1000 kbps. We are focussing on the first second after the start of the simulation. In this case of the toy network, this means that the base station may send 4 packets of data with size of 360 kb within the first second. The results of this scenario can be seen in figure 4.2.

We can observe two different behaviours of the examined rules for three different rate regions.

In the first region, for a low U2U-transmission speed (< 560 kbps), the average quality of users views after one second of simulation with an NC rule is 0 dB. This means that no user is able to display any view. When the base station receives a request for a virtual view, it answers with a coded packet. With the fixed B2U-transmission speed of 1000 kbps, the base station will be able to send one packet to each user within the first second. The users receiving a coded packet however, try to get some data for decoding from another user in its communication range, but since the U2U-transmission speed is too low, this transmission will not be successful within the first second of simulation. Every user has received one coded packet for his requested virtual view, but was not able to get a packet that helps to decode in time.

In the simulation without NC, we see another behaviour. Upon receiving a request for some virtual view, the base station sends uncoded data for one of the adjacent physical cameras. Upon receiving this packet, a user immediately sends another interest for data of the second neighbouring physical camera to the base station. Often the channel to the base station that has been used before is still free and can be used again, resulting in a second transmission of the base station to the same user. In this simulation with a B2U speed of 1000 kbps, two users will receive each two packets before the end of the simulation after one second and thus will be able to reconstruct the virtual view, while the other two users did not receive any data. This can be seen in Figure 4.2.

The second region is between a transmission speed of 560 and 1200 kbps. The first increase of the quality for simulations with NC rules occurs when the remaining time after the first data transmission from the base station ($t_{remain} = 1s - \frac{360kb}{1000kbps} = 0.64s$) is long enough for a user to send the received data to its neighbour at a U2U-transmission speed over ($ts_{U2U} = \frac{360kb}{t_{remain}} = 562.5kbps$). The user that received this packet from a neighbour, may now decode the NC-packets with help of Gaussian elimination and display the virtual view. The value for the average quality raises above 15 dB but can not yet reach the value of the simulation without NC. This comes from the random backoff-time of the starting requests. Although the backoff-time is low enough compared with transmission-times to have no significant impact on the performance, it sets the order of incoming requests at the base station at the start of the simulation. Therefore, the destinations for the data transmissions from the base station are chosen randomly. This leads in 33% of the cases to the situation, where the base station, in a first interval sends packets to two connected users u_1 and u_2. The interval t_{remain} is not large enough for both data transmissions from u_1 to u_2 and from u_2 to u_1, resulting in a situation with only one user with a view to display.

The third region, and the second big increase of the quality for simulations with NC rules can be seen above a U2U speed of 1285 kbps. This comes from the fact that after the base station sent packets in the second interval (after $0.72s = 2 * \frac{360kb}{1000kbps}$) there is enough time to send the received data further to the connected user. ($t_{remain} = 1s - 0.72s = 0.28s > \frac{360kb}{ts_{U2U}}$ for $ts_{U2U} > 1285kbps$). This leads to an average of about 35 dB.

In the no-NC case however, this increase after a U2U speed of 1285 kbps takes also place but not to the same extent. The reason for the increase is the same as in the NC case. The reason for the lower increase value is that if the base station sends data first to two neighbours(in 33% of the cases), the other two will never get the data in time, since the first two intervals of data sending by the base station will take 0.72s. This leads for the non-NC case to an average quality of about 31.6 dB.

In this scenario there is no difference in the behaviour of NC- rule one and NC- rule two. The difference between these rules can only be seen in a situation where a user requests some physical camera. If the curves do not overlap 100%, it is because of randomness of the simulation.

Influence of B2U link speed

In a next step, we fix the U2U link speed and analyse the influence of changing the B2U link speed on the average quality of displayed views of the users. The rest parameters are set as in the scenario before. The results are presented in Figure 4.3.

At a very low B2U transmission speed, no data can be transmitted to users, thus the average quality is 0, no matter which rule is active. When the transmission speed is increased over 440 kbps, the users may receive data from the base station and send it forward within one second ($\frac{360kb}{440kbps} + \frac{360kb}{2000kbps} = 0.998s < 1s$).

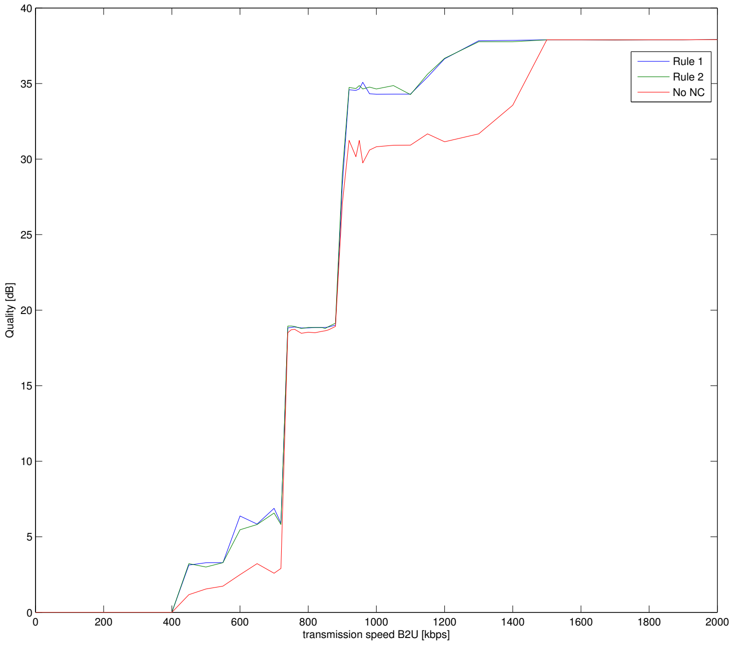

Figure 4.3: Average video quality versus the transmission speed of the B2U communication links in the toy network of Figure 4.1

In some cases this is enough for a user to receive the requested data to be able to display the requested view. With raising B2U link speed the probability for this scenario raises in both cases of the NC rules, and for the case without network coding. The reason for the higher raise in the curve of the NC rules comes from the higher probability that a neighbouring user has data that helps to decode. In the scenario without NC, the probability is lower. The data which a user needs to generate the virtual view may be not accessible at a neighbour.

The first significant rise of the quality occurs at transmission speed of the base station to users of 720 kbps. An average quality of 18 dB means, that in average, two users receive data to generate and display the virtual view within one second, and the other two do not. In all three simulations, with and without NC, two users will receive all their requested data, since two transmissions of data from the base station to the users are possible $(2 * \frac{360kb}{720kbps} = 1s)$. However, the remaining time is not long enough for users to exchange data. So only the first submitted data packet from the base station can be forwarded.

This changes above a B2U link speed of $879kbps$. The remaining time after the second data transmission of the base station $(t_{remain} = 1s - 2 * \frac{360kb}{879kbps} = 0.18s)$ is large enough for one more U2U data transmission. The influence of this can be seen in Figure 4.3. The average quality of the users view rises above $30dB$ in all cases, in both simulations with active NC-rules and in the simulation without NC. However, in the scenario without NC, the value for the quality does not rise as in the other scenarios. The reason for the lower increase in the non-NC case, is that if the base station sends data first to two neighbours (in 33% of the cases), the other two will not receive the data in time. In the NC scenario, the base station sends only one packet to each user and will thus have enough time to serve the other pair with NC packets.

The rise at a B2U link speed over $1125kbps$ comes from the fact that the remaining time after two data transmissions from the base station is enough to have two data transmissions from user to user $(t_{remain} =$

$1s - 2 * \frac{360kb}{1125kbps} = 0.36s$). This leads to a maximum average quality of users view of $38.2dB$.

For the simulation without NC, the B2U link speed has to be above $1440kbps$ to also reach the maximum value. In the worst case in this scenario, the base station sends first data to users of the same pair, and then to the other pair. Until every user receives all data he needs to compute and display the virtual view, the base station needs to transmit 2 data packets to each user. Therefore the B2U link speed has to be above $1440kbps = 4 * \frac{360kb}{1s}$

4.2.2 Toy network with random view requests

Influence of U2U link speed

In a next simulation we assume an array of three cameras and let the users randomly request one of the available views, which can be a physical view or any virtual view between two physical views. The requests of the users follow a Gaussian distribution. This means that some views and camera data are more requested in the network. The other parameters of the simulations have the same values as in the previous simulations. There is the same network topology as in Figure 4.1 and the number of parallel base station connections to users is set to two. Again we want to analyse the average quality of the views delivered to the users for the network coding algorithms presented in Section 3.2.

In Figure 4.4, we see the impact on the average quality of the users view of a changing U2U link speed. The B2U link speed is set to 1000 kbps like in the example above.

For a lower U2U transmission speed, the curve for the results of the simulations without NC is significantly higher than the two for the results withNC rules. A similar result we got from Plot 4.2. However, the average quality of the users view differ a lot from the ones in the scenario before. The explanation for this can be found in the distribution of requests. Since the requests for views of the users are Gaussian distributed, there are not only virtual views, but also a significant part of interests for a physical view. Therefore, the users need to receive only one data element from the base station and they will be able to display the view.

Since with these settings, the base station is able to submit up to four packets within the time slot of one second, in the best case scenario the users do not need to exchange data at all. However the explanation for the significantly higher average quality in the simulation without NC in Figure 4.4 is the same as for the higher value in Figure 4.2.

In the NC simulations, the base station sends just one packet to each user, then, with a delay, the users will send new requests to the base station, which in this scenario with a B2U link speed of $1000 kbps$ will not be able to respond in time. On the other hand, in the scenario without NC, the base station will focus on submitting data to a user until he is able to compute and display the view. Like this, also some users with requests for a virtual view will be able to display their view after one second.

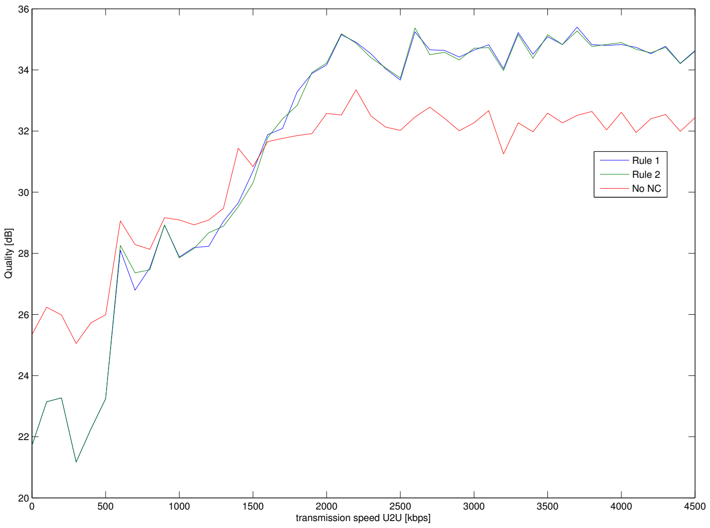

Figure 4.4: Average video quality versus the transmission speed of the U2U communication links in the toy network of Figure 4.1 with random view requests

The increases in quality over the whole figure can not be explained in the same detail as in chapter 4.2.1, since there is some randomness in the initialisation of each simulation. To gain more reliable results and a smoother curve, the simulation run 300 times for the same U2U link speed value, generating an average value for the quality of the users view. By averaging over many randomly generated runs, we are able to explain more likely the behaviour of the curve at some values. For example the significant rise at a U2U link speed over 500 kbps, which can be attributed to the fact, that the remaining time after the first session of data transmissions from the base station is long enough for a user to send the received data to its neighbour. This ability leads especially in the scenario with active NC-rules to a large improvement of quality. As already mentioned, users in a scenario without an active NC- rule are more dependent on a higher U2U link speed, since they receive just one packet from the base station in this time slot and need to get the missing information from their neighbours.

Over a value of 1500 kbps the curves of rule 1 and rule 2 intersect with the curve of the rule without NC. From this point on a simulation with NC rules give better results than the one without NC. This comes from the fact that after the base station sent packets in the second interval there is enough time to send the received data further to the connected user. Although still using the ”toy scenario topology”, the assumption, that the NC rules gives better solution at a high U2U link speed gains support by the plot of figure 4.4.

There are no big differences visible for the NC rules. In this fixed topology, the only difference may occur if two neighbours request different physical views. If one of them received already the data from the base station and the second one sends an interest to the base station, the generated packet will differ in rule 1 and rule 2. While the base station would create an NC packet in rule 1, it would send an uncoded packet in rule 2. Hence in rule 1, one more decoding transmission would be required, while in rule 2 both users have the data to display. From the results in Figure 4.4, however, we can not state this difference. The small differences between the rules are more likely to be the results of the randomness in each simulation.

Influence of B2U link speed

In this scenario, once again the topology of Figure 4.1 is used with randomly generated requests with help of a Gaussian distribution. This time, we fix the U2U link speed to 2000 kbps and vary the B2U link speed. Again, we compare the NC rules with the third rule without NC. The results are displayed in figure 4.5.

In simulations with low B2U link speed, the time limit of one second is not sufficient for users to receive data from the base station, thus the average quality of users view is 0 dB.

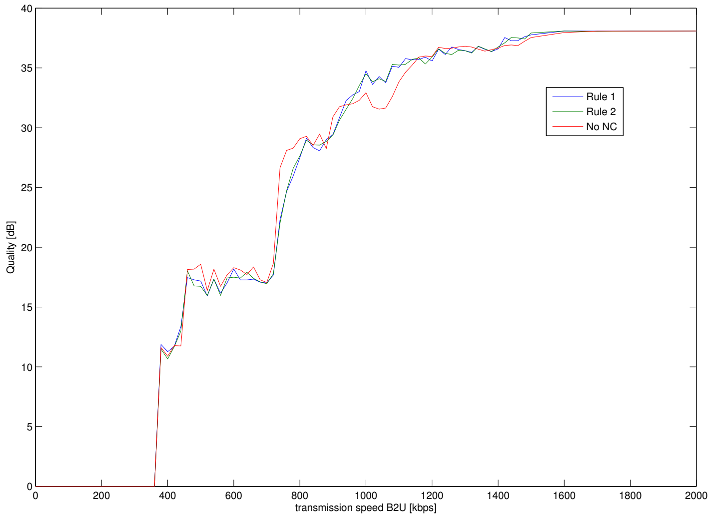

Figure 4.5: Average video quality versus the transmission speed of the B2U communication links in the toy network of Figure 4.1 with random view requests

A first increase of quality can be seen after a B2U link speed of over 360 kbps. Now some data from the base station will be received by the users in time. The users who are requesting a physical view may now display the view. Users with requests for virtual views may at the earliest be able to display a view at a B2U link speed above 440 kbps. Now the time slot of one second is long enough to receive a packet and then forward it to a neighbour. $(\frac{360kb}{440kbps} + \frac{360kb}{2000kbps} = 0.82s + 0.18s = 1s)$

There is a third significant increase of the quality in the figure. After a value of 720 kbps for the B2U link speed, the base station is able to send two packets, one after the other in one second. During the further procedure of a rising B2U link speed the values in each of the three simulations converge to the max value of the quality 38.2 dB.

4.2.3 Randomly generated network

In this section we consider a network with 20 users and communication radius $r = 0.4$, uniformly distributed over a unit square with the base station located at the center. The users each request either a physical view captured by an array of eight cameras or a virtual view in between two physical camera positions. We consider a base station with five parallel links with users, and a B2U link speed of 1000 kbps. Again we want to analyse the average quality of the views delivered to the users for the network coding algorithms presented in Section 3.2.

The results are shown in Figure 4.6. We can state a significant difference between the quality in simulations with and without NC. For all evaluated values for the U2U link speed, the resulting quality is higher in the scenario without NC. There is a big variability in the results especially for the NC rules. However, the plot provides an overview and a tendency for the performance of the proposed network coding algorithms.

This performance of the NC rules could not be improved with more runs in the simulation for random networks. However, the results of the toy topology show that there are potentially some benefits from NC. An optimization and some design changes are needed for the proposed algorithms to have benefits from NC in the random network. Some ideas for an improvement are presented in Chapter 5.2.

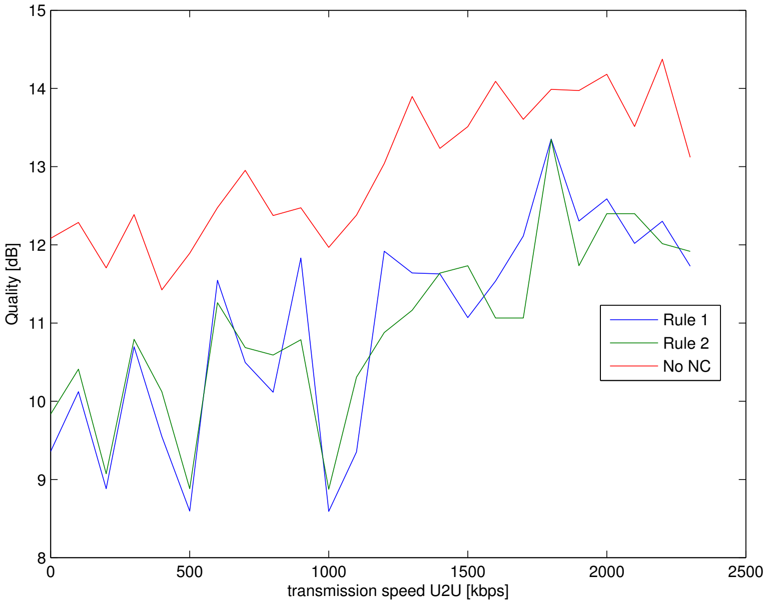

Figure 4.6: Influence of network coding rules on the average users quality at a rising U2U link speed in a network with 20 randomly distributed users

4.2.4 Communication radius r

Let us revive the discussion about the communication radius **r** mentioned in Chapter 3.1. In a wireless communication scenario there is always interference between multiple device to device communications, but as explained in Chapter 3.1, the interference can be reduced by changing the value for **r**. Since a small radius for user to user communication will make it impossible for users to communicate with each other, and a maximum radius will maximize interference and only allow one pair of users to communicate at a time, we expect, that there is an optimum value for **r**.

In the simulation we fixed values for the length of the camera array to 8, the number of users to 30 and the number of parallel user to base station connections to 6. Furthermore, the connection speed of the user to user links is set to 2kbps and the connection speed for each of the connections with the base station is set to 1.5 kbps. The range of the value for **r** on the x axis is set from 0.1 to0.7 and on the y axis you can see the average quality for the displayed views of all users in dB.

The results of the simulation for this set-up are plotted in Figure 4.7. The blue and green lines show the results for the simulation with activated Network Coding (NC) rule 1 and 2 respectively.

The plot shows the expected results. At a low value for r, the resulting quality for the views are low too. With an increased radius, the quality for the users view are higher reaching a maximum at a communication radius of 0.2. If we increase the radius even more, the resulting quality decreases. We can spot this behaviour for all of the three compared schemes. Although each point in the plot is an average value over 20 simulations, we can state a big variation. Although we can infer from Figure 4.7 that there is a optimum radius, the resulting quality from various inputs for the communication radius depends on the chosen and fixed parameters for this simulation. If we for instance change the number of users to 40, the resulting quality of the users view for each value for the communication radius will change.

To have accurate results and a more detailed plot, we propose a higher sampling rate, a larger number of simulation rounds and to have a scenario with less variations.

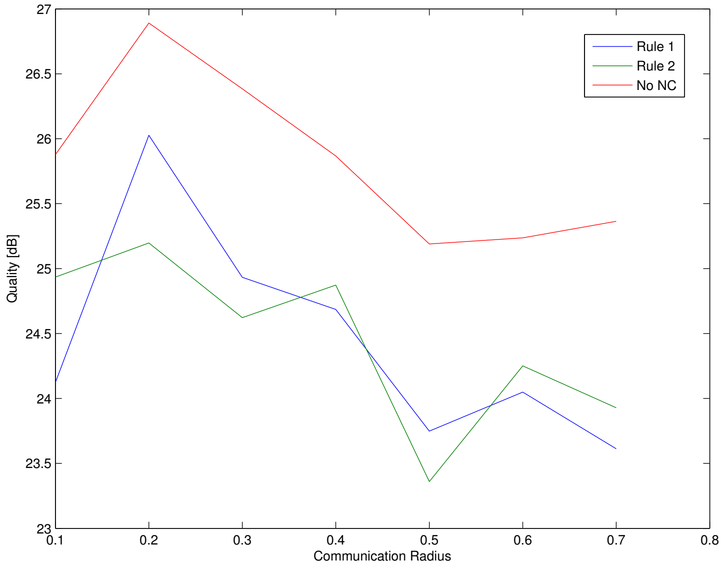

Figure 4.7: Plot over communication radius in a fixed example scenario for different Network Coding rules

If we have a larger number of simulations, the randomness of each round will not be that dramatic, because the average will level off at some point. As part of this work a significant larger number of simulations was not feasible, because of time, that it takes to run one simulation. In a future work, it is suggested to analyse this aspect in more detail.

The other proposed variation tackles the randomness of the implemented algorithm. For example if we additionally fix the position of each user, we could have a significant plot for this concrete situation. Another attribute to fix would be the requested view for each of the users. Instead of using the Gaussian distribution for determining the users requested view, we could have a fixed number of users for each view. With this approach we may have a higher explanatory power for this fixed situation, although we loose some significance to be able to say something about general wireless communication scenarios.

5 Conclusions

5.1 Conclusion

The evaluation of the different simulations has shown that the proposed network coding rules are advantageous in some situations. The toy network helped to understand the performance of the proposed network coding rules with respect to the speed of the user to user and base station to user links. In cases of a high U2U link speed, simulations with NC rules provide a higher throughput and a better result in terms of the quality of the user's view than in the same CCN toy network without NC. This conclusion can be expanded to the toy network with randomly distributed requests. Again the throughput in wireless networks with the proposed NC rules is higher than in one without NC rules. These results depend on the chosen fixed values and the topology settings. We've seen, that in networks with randomly generated topologies, the proposed NC rules do not have to perform better, even at a higher U2U link speed. However, these outcomes should not result in a general conclusion, that the NC rules offer less throughput in a randomly generated network. To be able to make a conclusion in this part, a more extensive investigation is required with other fixed values for the number of users, their communication radius and transmission speeds.

Since we have seen that the U2U communication is important for the NC algorithms to be beneficial, we have evaluated the influence of the radius on the average quality of the users views. Intuitively there must be some optimum value for a user's communication radius. This optimum is situated somewhere where there are several U2U connections possible without interference. The simulation supports this assumption. However, this simulation has a fixed number of users, which certainly has an influence on the resulting optimal communication radius.

5.2 Future Work

While working on the thesis we discovered some settings and ideas that could lead to a more beneficial use of NC and a higher throughput. Unfortunately, testing these ideas would go beyond the scope of this thesis. Nevertheless, we want to share these ideas of improvement with the reader.

In some situations in a simulation, a user requests some virtual view (between camera A and B). If a neighbour of this user has stored some NC packet of a linear combination of A+B, he will send this packet to the requester. In the next step the user will request again for some data (A, or B, or a combination A+B) to decode the just received packet. The same neighbour will then answer again with the same packet since he does not "remember" he already sent this exact copy to the requester. Since these packets are obviously linearly dependent, this transmission is totally redundant. The solution we propose is to add memory for each user, so that he knows on which face he sent which packets. The memory will not only help to solve this problem, but also will create new opportunities for NC rules for communication between users.

The next point we want to suggest for improvements tackles the simulation with big networks. Since sending an interest takes not much time, they are forwarded fast in the network. The PIT of every user grows and suddenly holds information for a bunch of data elements that the user forwards requests for other users. The base station and other users do not know which users to prefer. It may happen that some user receives data to forward to the user that originally generated the interest, but that user could have received the data directly. As a solution to this problem we suggest adding some sort of priority in the PIT to the faces, for knowing for which user the data is more important. This can be a counter for the number of hops, for knowing the covered distance of the interest, or some information if the requesting user itself has gains in receiving this data.

Bibliography

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, July 2012.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. L. Braynard, "Networking named content," in *In CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[3] N. Golrezaei, A. Molisch, A. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *Communications Magazine, IEEE*, vol. 51, no. 4, pp. 142–149, April 2013.

[4] N. Golrezaei, A. Dimakis, and A. Molisch, "Device-to-device collaboration through distributed storage," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 2397–2402.

[5] P. Chou and Y. Wu, "Network coding for the internet and wireless networks," *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 77–85, Sept 2007.

[6] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 3, pp. 497–510, June 2008.

[7] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *Information Theory, IEEE Transactions on*, vol. 56, no. 9, pp. 4539–4551, Sept 2010.

[8] L. Toni, N. Thomos, and P. Frossard, "Interactive free viewpoint video streaming using prioritized network coding," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, Sept 2013, pp. 446–451.