Collaborative Video streaming for Mobile Devices

Paul Lindt: 6141815

Ali Saleh: 6517831

Universität Hamburg

Seminar:Mobilnetze, dienstintegrierte Netze und Echtzeitkommunikation

Contents

Chapter 1

Introduction

1.1 Motivation/Problem

There is a number of reasons to expect an extending demand for higher network throughput capabilities for mobile devices. This demand is even more urging when looking at applications working with video streaming. The following two sections look at two driving forces behind this trend.

1.1.1 Trends in mobile internet usage

The transmission of video information in realtime requires a higher throughput capability than the transmission of voice or still picture information does. Because of this fact, the growing usage of mobile video results in much greater challenges than the increased usage of other services. With mobile video being up to 50% of mobile internet traffic, as the Cisco visual networking index suggests, we need to increase the available throughput as much as possible.

1.1.2 Trends in mobile device development

The capabilities of mobile devices keep improving. On one hand the devices are capable of using new network infrastructure like LTE-Networks, which allows them to communicate at a faster rate. On the other hand, their ability to process multimedia content is improving. One of the often advertised characteristics of mobile devices is the resolution of their display. At this point devices with a resolution of 1920x1080 Pixels are available.

However the manufacturers are already promising to bring devices with a resolution of 4096x2160 Pixels. If users try to download video content at the apropriate resolution to make full use of their device over the mobile networks, they will use up a lot more of the network capabilities than they currently do.

1.1.3 Mobile network operators

One obvious solution to this problem is the ”dumb fat pipe” approach. However this is a solution only the mobile network providers can implement and it requires investment on their side, which might result in increased fees for the customer. Even if the network operator provides almost ideal coverage, a number of problems might appear when communicating over a wireless link. The changing distance to the base station and external interference as well as multi-path propagation might result in signal coverage fluctuation. Additionaly, a number of mobile network providers in Germany are currently offering monthly plans with an unlimited amount of data, which can be transmitted with the limitation of an artificialy decreased transmission speed after a certain amount of data received. For a user of such a network plan it might be tempting to circumvent the limited throughput through cooperation with other users in the same situation.

In the following we try to evaluate the concept of collaborative video streaming and its implementation, based on ColStream, the system introduced in [1].

1.2 Overview of ColStream

The following sections introduce ColStream as a possible solution to the previously described problems. They show the scenario it is assumed to be used in and how it intends to solve the problem of limited throughput over a 4G/LTE connection. The name ColStream is a contraction of the words collaborative and streaming. It was introduced in 2014 by Mingyang Zhong, Peizhao Hu, Jadwiga Indulska and Mohan J Kurnar as an approach to inprove the video streaming capabilities of one mobile device through cooperation with other devices. Its concept was implemented as an Android application.

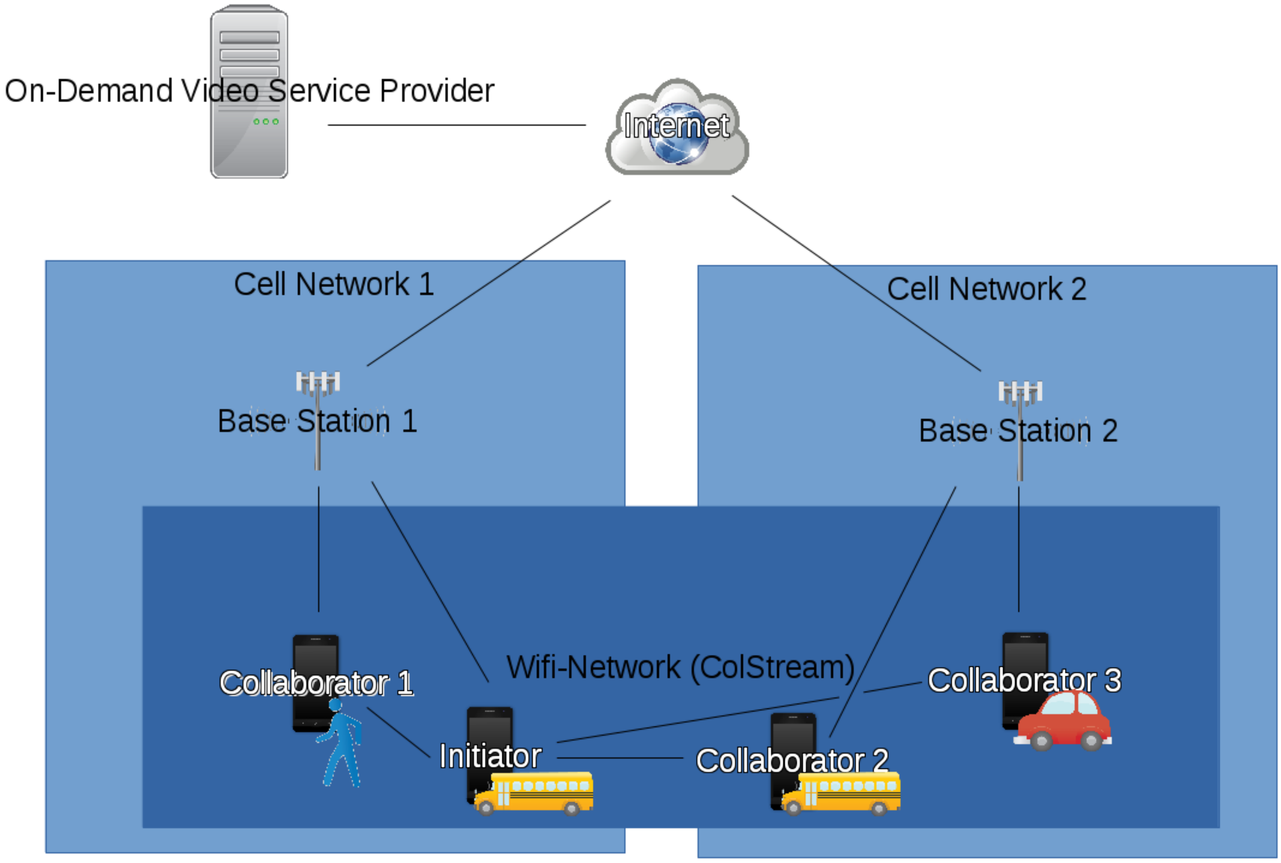


Figure 1.1: Overview of the idea of ColStream

1.2.1 Scenario

In [1] Zhong et al. describe a scenario of a person accessing a high-definition video stream through a mobile device over a 4G/LTE network. Figure 1.1 illustrates this scenario. The mobile network might experience interference or not be available at all times. To compensate for these limitations the mobile device connects over Wi-Fi to other mobile devices and negotiates with them the transmission of parts of the video stream over their 4G/LTE connection. As the location of the person is changing so is the level of experienced interference and the number and capabilities of devices surrounding it. ColStream promises to adopt to these changes and through negotiations to find the best available combilation of collaboration partners. The base for this negotiation process is provided by a virtual currency through which the video consuming participant can pay others who offer him their bandwidth.

1.3 Related Work

There are a number of concepts and implementations that try to leverage cooperation to improve user experience when streaming or downloading information.

1.3.1 Collaborative Download

There are several Peer to Peer (P2P) networks that allow collaborative download. One famous example is the BitTorrent network which can be used to distribute large files like Operating System images with limited server capacity. Even though a streaming system with questionable legality was created based on it, these systems were designed to maximize the availability of a download, not its speed [2]. When looking at streaming video to mobile devices, we assume the 3G/LTE connection to be the bottleneck. In this situation, the usage of such a P2P-system over this connection is of no advantage to us. While we could try to share information with our peers over Wi-Fi, we can not assume to have enough peers interested in the same information in range of our Wi-Fi connection.

CStream

CStream [3] appears to be a system very similar to ColStream, even in its naming, in which the C is short for collaborative. Both focus on the improvement of video streaming performance. However CStream does not focus on the mobile scenario. Instead it intends to apply this approach to a wider range of scenarios specifically including the combination of several wired connections.

COMBINE

COMBINE [4] is a system designed to improve the download speed of files by using the 3G/LTE connection of other mobile devices, which can be reached over Wi-Fi. So it is very similar to CosStream, however it was designed without the realtime demands that appear when handling a video stream. Additionaly its current implementation only works on notebooks and PCs, not on smartphones or tablets. These two classes of devices still differ in many ways based on their processing power, the time they can work without recharging, the probability to find one of those devices turned on in your Wi-Fi range. We consider this to be problematic when evaluating the behaviour of this system on what we consider to be mobile devices.

1.3.2 Incentive Systems

One of the most important parts of ColStream is its incentive system, which is necessary to motivate users to offer their unused throughput to others. Unfortunately while the negotiation process for payment is discussed in depth, it is unclear if the system has any means to motivate the user to look for pazments to earn and to offer his available throughput to do so.In the following section we list a number of systems with a simmilar problem and their incentive approach.

BitTorrent

P2P networks face a similar problem. BitTorrent tries to evaluate the effort a user is providing based on his ratio of up- and downloads in relation to the resources available to him, specifically storage space and network access. Users can be punished for having a low ratio of uploads to downloads by restricting the download speed[5].

Webservices

A number of webservices like reddit.com or duolingo.com allow the user to earn virtual points similar to the payments a user receives in ColStream. In the extreme case of reddit these points have no use besides showing others that a user has made valuable contributions.

Chapter 2

Technical Detail

2.1 Collaborative Downlaod

ColStream assigns one of two roles to its users, initiators and collaborators. A user streaming a video is an initiator. Users offering their bandwidth to others are collaborators. The initiator creates a group of potential collaborators and assigns them chunks of the video stream data to download, based on their capabilities and price demands. The figures 2.1 and 2.2 illustrate the process of group formation, respectively the actual download of video information by role.

2.2 Group formation

In the left half of figure 2.1 we see the behaviour of an initiator when he begins to stream a video. He starts by downloading and showing the video on its own. The initiator then analyses the bitrate at which he receives the video information. If this bitrate is enough to provide the user with an acceptable video quality, the initiator keeps streaming the video on its own and periodically rechecks the bitrate. Should at some point the bitrate be no longer acceptable, the initiator tries to form a group with collaborators. To find candidates for a group, the initiator broadcasts a "JOIN-ME" message. This message contains information on the amount of required bandwidth as well as how much the initiator is willing to pay for this bandwith. Potential collaborators in range of the initiator will receive the "JOIN-ME" request and process it as shown in the right half of figure 2.1. Each collaborator first calculates the bitrate he could offer to the initiator based on the signal strength of the 4G/LTE network. He then compares the price offered

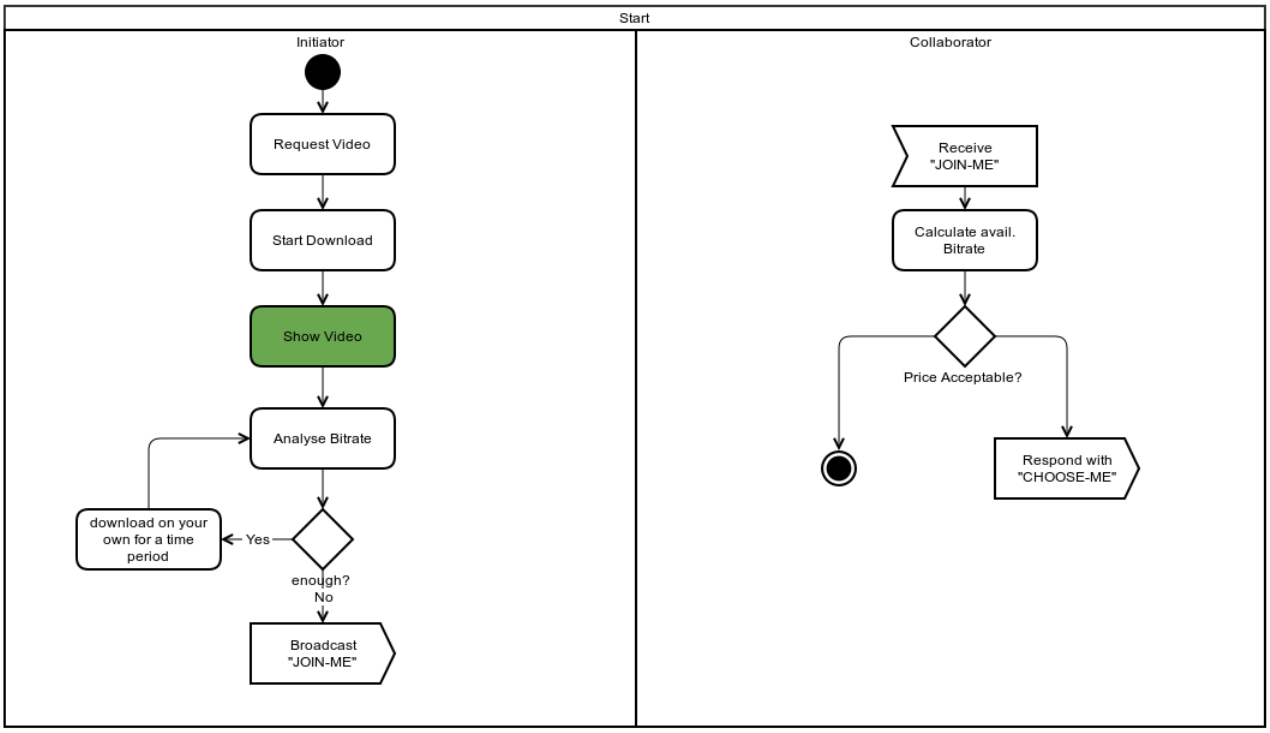


Figure 2.1: Group formation

by the initiator to his own current demand. Limited resources like battery capacity can drive the price up. Should the offered price be acceptable to the collaborator, he responds with a ”CHOOSE-ME” message to indicate his consent to join the group.

2.2.1 Bitrate estimation

The collaborators estimate the bitrate they can offer to the initiator based on their signal strentgh. To be able to do this in a realistic way, collaborators periodically measure their signal and the corresponding bitrate. To compensate for fluctuations, historic information is factored into these measurements. The specific equations are

- $signal = \alpha * signal_{new} + (1 - \alpha) * signal_{old}$
- $tp = \alpha * tp_{new} + (1 - \alpha) * tp_{old}$

with tp being throughput and α an unspecified factor.

2.3 Collaborative download

The process of collaborative streaming within the group created in the previous section is shown in figure 2.2. Based on the "CHOOSE-ME" messages received by the initiator, a pool of possible participants in the streaming is created. The initiator then periodicly selects the best suited candidate to download a chunk of the video stream and assignes him this task by sending a request to do so. The size of the chunks is dynamically calculated to fit the throughput the candidates provide, using this equation:

- $ChunkSize_i = \frac{MAXCHUNKSIZE*tp_i}{tp_{max}}$

Candidates are selected based on their price demand and offered bitrate. On receiving a download request, a collaborator downloads the assigned chunk of the stream and forwards it as a message to the initiator. When the initiator receives this message, he integrates the data into the video stream and displays it to the user. The number of collaborators available to an initiator is expected to be changing over time. The collaborators send regular "I-AM-ALIVE" messages to the initiator to indicate that they are still available. They initiator will tolerate up to three missed messages in a row before dismissing the collaborator as not longer available.

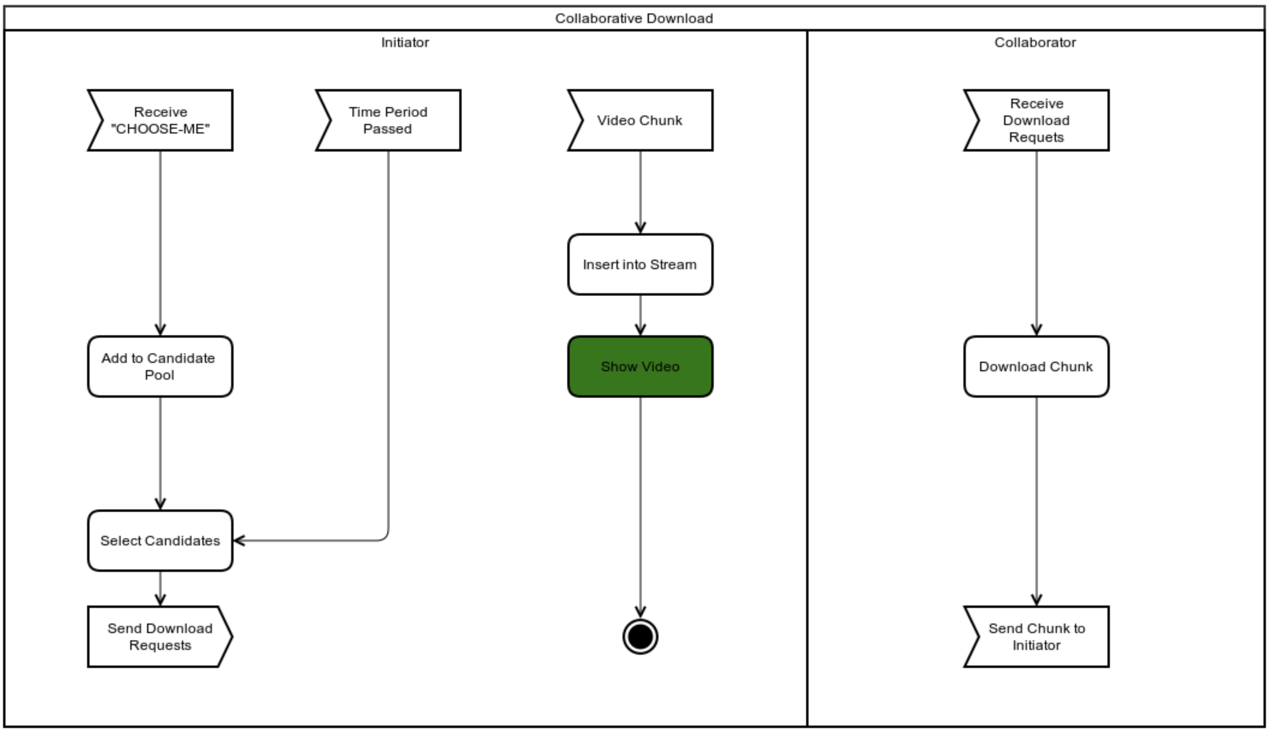


Figure 2.2: Collaborative Download

2.3.1 Problem handling

The work is distributed between collaborators through assignment of chunks of the video. However once these chunks are assigned, a number of problems might appear that will make it impossible for the collaborator to complete the download on time. For example the user might experience interference which will drop his network throughput below the expected value. In this situation ColStream collaborators notify the initiator of the problem and forward the already downloaded information to him, so he has to reassign the remaining part only. Another problem is the loss of Wi-Fi connection to one of the collaborators. The initiator can notice this loss through missing "I-AM-ALIVE" messages or if the collaborator takes longer then expected to provide a chunk. If the collaborator does not respond to additional messages, the initiator will download this collaborators chunk on his own to compensate for the missing data.

Chapter 3

Evaluation

To show the effectiveness of ColStream, Zhong et al. provide in [1] measurements they have collected in experiments. The scenarios for these experiments focus on its features, and one is a comparison with the COMBINE tool. The first is the increase of throughput by means of utilizing the network connection of several devices. The second is the adaptation to a changing number of available collaborators. We discuss these measurements in the following chapter. In the following four sections we discuss the experiments and results presented in [1] and point out aspects we consider problematic. The names of the respective sections follow the titles used in the paper.

3.1 Experiment A: "basic tests"

3.1.1 Measurement Setup

In these experiments ColStream runs on emulated android devices. One initiator and up to ten collaborators are emulated on a computer with an Intel Core i5-3210M processor and 6 GB of RAM. The paper claims that the throughput available to the individual emulated devices was fluctuated to make their behaviour more realistic, however the exact process of fluctuation is not disclosed. The initiators stream one of three videos from Youtube. These videos were randomly selected and differ in size. The time it takes for them to download the video was measured. The measurements were repeated with varying number of collaborators.

3.1.2 Results

Figure 3.1: Performance over number of collaborators [1]

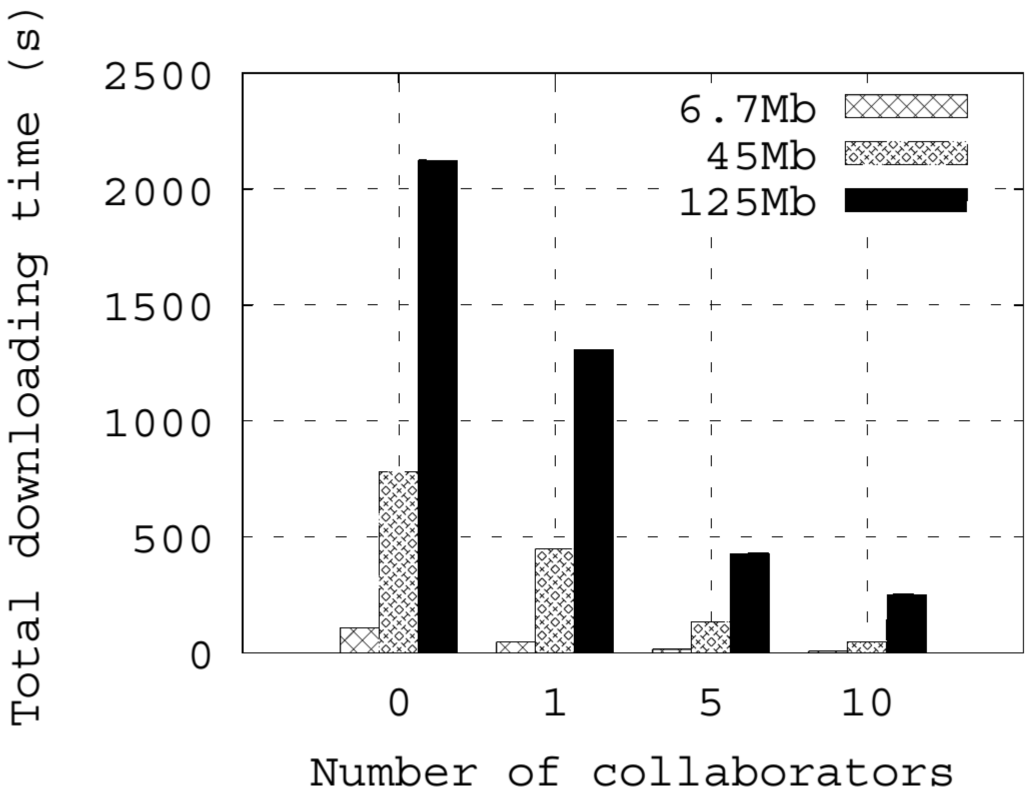


Figure 3.1 shows the results as measured by Zhong et al. as a bar graph. The values are grouped by the number of collaborators used. Within each group, the time it took for this configuration to download each of the three videos is shown in seconds by the height of the corresponding bar. The size of the corresponding video for the bar is indicated by the pattern applied to it. This experiment shows that ColStream is able to reduce the time it takes to stream a video through employment of additional collaborators. In all three filesizes, the use of one collaborator already reduces the time by about a third. The improvement appears to increase with the size of the video to download.

3.1.3 Criticism

There are several aspects that raise the question if these results can be applied to a real world scenario. The paper provides very little information on the simulated network available to the emulated devices. One of the key motivations for the creation of ColStream is the compensation for fluctuation of throughput. Without knowing the strentgh of fluctuation applied to the throughput available to the streaming devices, these measurements can not be properly evaluated. Additionally these fluctuations seem to have been applied on the simulated 3G/LTE connection only, not on the Wi-Fi connection between the individual devices. The measurements were not repeated with different streaming platforms, files, networks or in any other variation. While these measurements do show as intended that ColStream could work, they do not show that it will work in any reliable way.

3.2 Experiment B: "Dynamic adaptation"

This experiment intends to show that ColStream is able to react to a changing group of collaborator. To do so, the streaming behaviour in reaction to collaborators joining and leaving the groups as well as price changes are demonstrated.

3.2.1 Measurement Setup

Unfortunately the measurement setup for this experiment is not explicitly stated. We assume the basic setup to be the same as in Experiment A, with emulated devices running on a notebook and experiencing a simulated fluctuation of network throughput. Three different situations are simulated. Number one is a new collaborator joining the group and demanding a lower price then the previously available collaborators. Number Two is a collaborator leaving the group. Number three is the loss of connection to one of collaborators.

3.2.2 Results

The figures 3.2, 3.3 and 3.4 show the accumulated downloaded amount of data by collaborator over time. In all three figures the collaborators behave as expected. In figure

Figure 3.2: Download ammunt over time by collaborator with user joining [1]

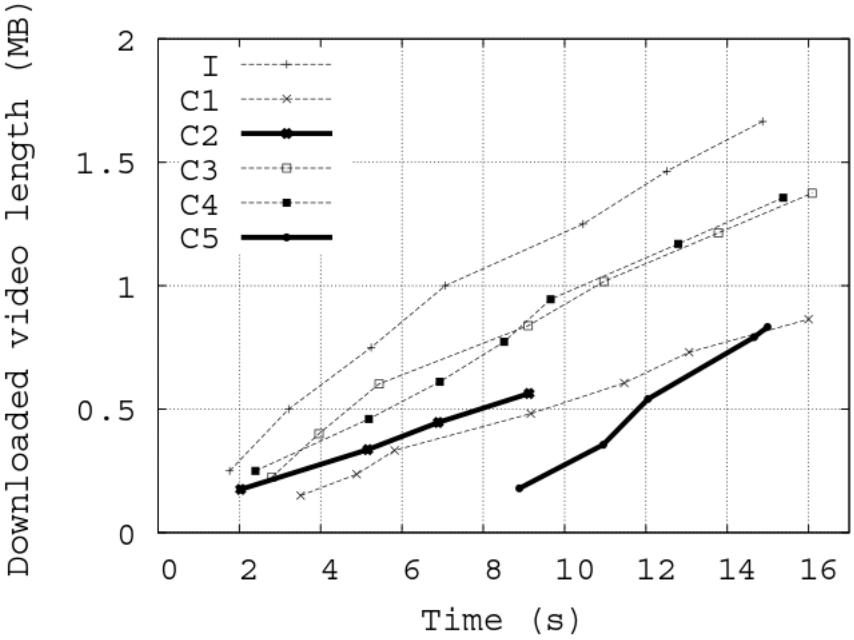
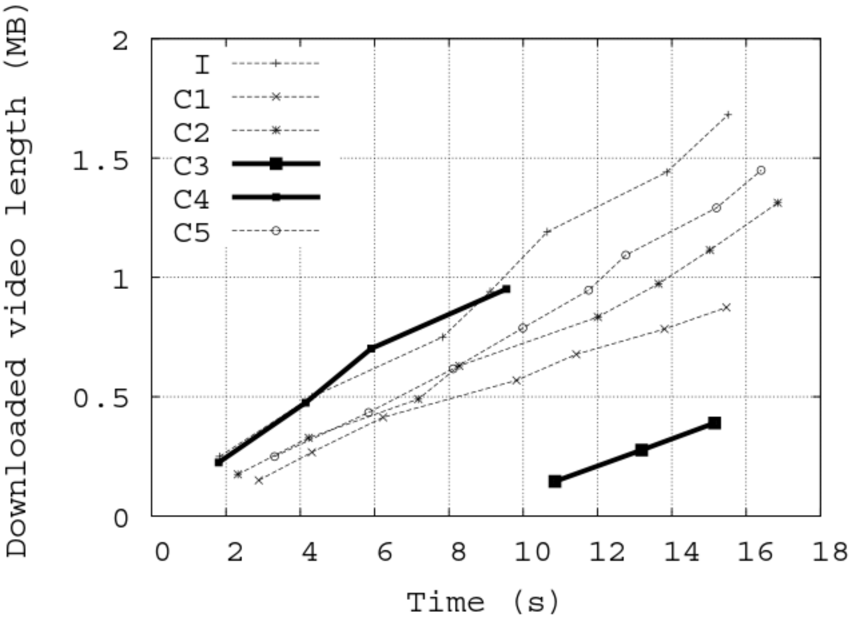
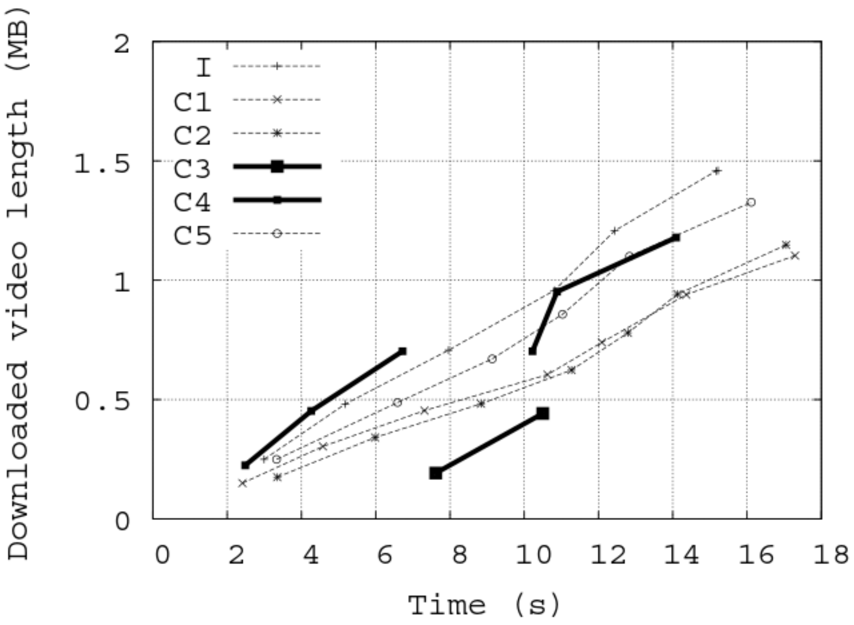


Figure 3.3: Download amount over time by collaborator with user leaving [1]



3.2 we see a new user (C5), demanding a lower price join after 9 seconds. As a result the user C2 is no longer assigned new parts to download. In figure 3.3 we see a user (C4) orderly leave and another user (C3) being selected as his replacement. Figure 3.4

Figure 3.4: Download amount over time by collaborator with connection loss to user [1]



shows that colstream is able to handle a connection loss (at around 7 seconds) and its reestablishment (shortly after 10 seconds) to a collaborator(C4).

3.2.3 Criticism

While these measurements do suggest that ColStream could work as promised, unfortunately the influence of the collaborators leaving and joining on the quality of a videostream for the user is not made clear.

3.3 Experiment C: "Performance tests: COMBINE agains ColStream"

This experiment compares ColStream and COMBINE, which is also discussed in Section 4 and intends to show that ColStream is better prepared to handle the specific challenges of distributed video streaming. To compensate for possible packet loss, chunks are assigned to multiple collaborators, to be sure to receive the information in time. This experiment looks at the amount of data that needs to be redownloaded by both systems, in reaction to random failures of collaborators.

3.3.1 Measurement Setup

Again very little is said to the setup of the experiment besides the use of five collaborators. While we can assume ColStream to be set up in a similar way as in the previous experiments, no such assumptions can be made about COMBINE.

3.3.2 Results

Figure 3.5: Redownload handling by ColStream and COMBINE [1]

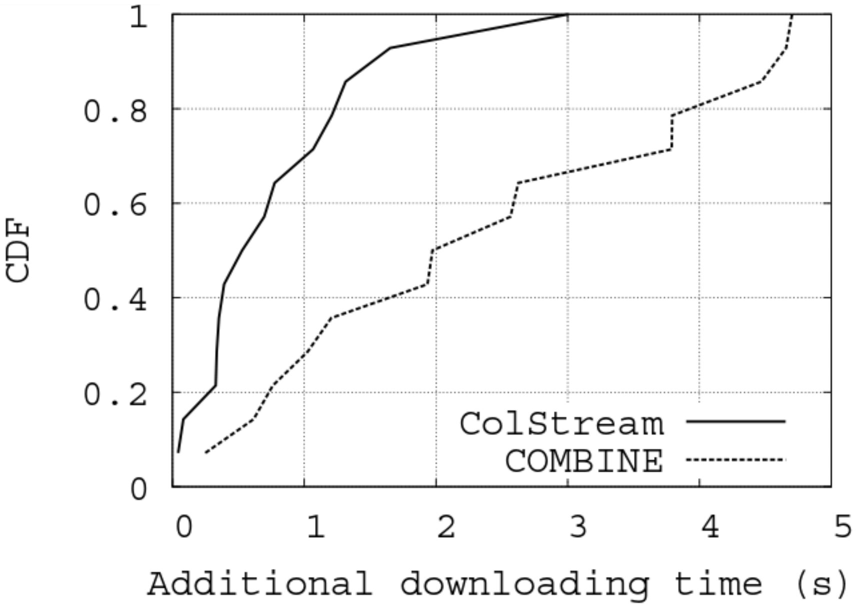


Figure 3.5 shows the performance of ColStream and COMBINE with growing additional downloading time on the x axis and their scores in a not exactly defined cumulative distribution function based benchmark on the y axis. ColStream is scoring higher at all measured values.

3.3.3 Criticism

Unfortunately it is in no way clear how reliable these benchmark tests are.

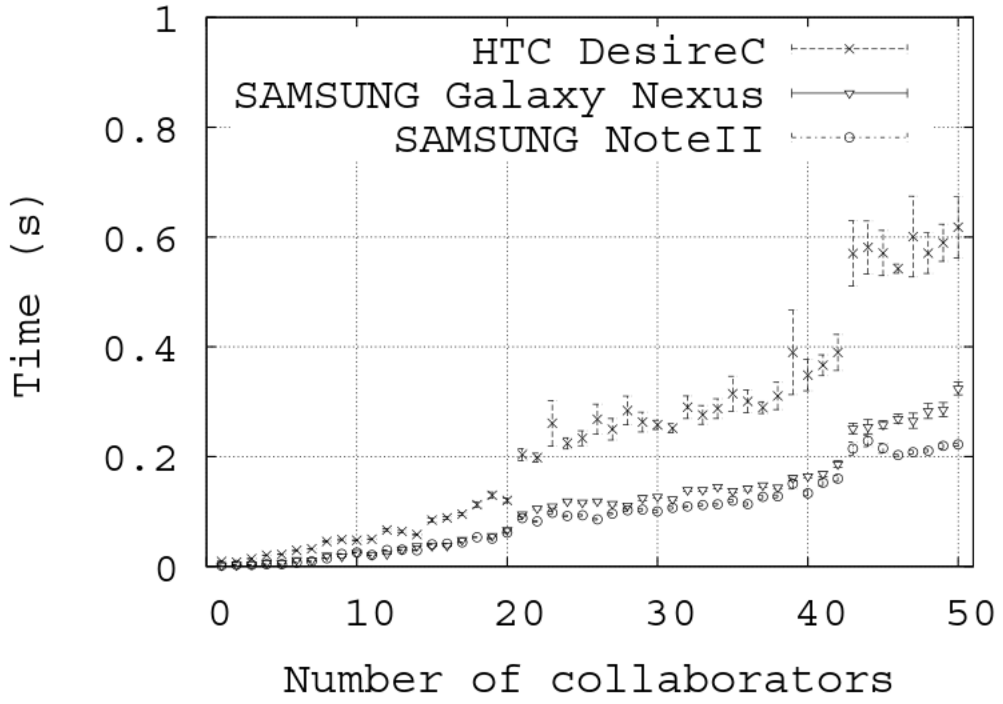
3.4 Experiment D: "Feasibility study of the optimisation algorithm"

This experiment measures the time it takes for ColStream to adjust to changing collaborator groups. This measurement allows to evaluate the amount of overhead generated by the group management and to decide if an initiator can adjust his peer selection fast enough to properly leverage changes in the group.

3.4.1 Measurement Setup

This experiment was executed on real devices with a range of hardware capabilities, specifically an HTC Desire C, Samsung Galaxy Nexus and a Samsung Note II. It was repeated for varying group sizes ranging from one to fifty participants. The time it took the initiator to adjust its peer selection is measured. Unfortunately it is not made clear if a video was streamed during this experiment.

Figure 3.6: Group optimisation over group size [1]



3.4.2 Results

Figure 3.6 shows the group adjustment time on the y axis over group size on the x axis for the three devices. A clear trend of the adjustment time growing with the group size is visible. While the values for the two Samsung devices are close together, the Desire C takes longer to adjust. This fits the much lower hardware characteristecs of the HTC device compared to the Galaxy Nexus. The Note II is the only device with a multi core processor. Since the peer selection algorithm is in no way prepared to make use of this, it is not surprizing that the measured values for the Galaxy Nexus and the Note II are close by. In all measurements, the values are low enough to assume that the group adjustment does not cause significant overhead.

3.4.3 Criticism

The previous section already looks at the hardware resources of the mobile devices to explain differences in measurements. However we think that the very similar values measured for the Samsung Note II and the Galaxy Nexus suggest that the measurements were executed without an actual video being streamed. We are not sure if these values would be of any significance to evaluate ColStream's ability to adjust to changes in collaborator groups.

Chapter 4

Criticism of the concept and implementation

In the previous chapter we have already pointed out a number of concerns with the performance [1] promises for ColStream. However we think the concept of collaborative streaming in general and its implementation in particular raise a number of questions. As already discussed in Section , a number of application rely on collaborative download. However the applications downloading a single file to all participants share an obvious common goal. This is not the case for ColStream and as the authors point out, they intend to compensate for this fact using the incentive system. However the concept of a price and compensation is only discussed on the subject of price negotiations. It is unclear how this virtual price would be paid and in what way the users are incentivized to aggregate payments. While it can be argued that this goes beyond the scope of ColStream as a proof of concept implementation and computer science in general, we consider the incentive system to be one of its most important elements. As the authors themselves argue, it sets ColStream aside other systems. As also stated in Section 4, one of the key features of BitTorrent is its upload to download rate based incentive concept. It would be interesting to make these values part of the incentive algorithm. The distribution of a collaborative download of a single file over a number of participants makes it unnecessary to hide the content of the downloaded file. This is no longer true if the collaborators purely assist in the download. This situation can pose a threat to the initiator as well as to the collaborators. The initiator obviously has to share information about the video stream he is requesting. By analyzing the price he is willing to offer and his demand for additional throughput, collaborators might deduct additional information about the initiator. The initator of course can make similar assumptions by identifieng the threshold acceptable price of collaborators. An additional problem for the collaborators is the fact that they are downloading content for another entity. If they don't evaluate the requested data, they might be used to download illegal or at least unlicenced content. If they do evaluate the requests or content, the initiator loses a lot of privacy. A rather technical criticism is that a number of developer resources claim the simultaneous usage of Wi-Fi and and 3G/LTE on an Android device to be extremely limited if not impossible. While Zhong et al. state to have streamed video on real devices, the paper shows them only used for the evaluation of the video chunk selection and assignment algorithm. Unfortunately we were not able to find neither the sourcecode nor executable files of any kind that would allow us to recreate the measurements discussed here or extend them though our own experiments. So while the authors claim ColStream to be a good solution, there is no way for us to verify this claim. This appears to be a common problem in this area of research, since we ran into similar problems with other projects. The result provided in [1] suggest that collaborative video streaming is at least technologically possible.

Bibliography

[1] M. Zhong, P. Hu, J. Indulska, and M. Kumar, Colstream: Collaborative streaming of on-demand videos for mobile devices, in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*, pp. 1–7, 2014.

[2] D. Qiu and R. Srikant, Modeling and performance analysis of bittorrent-like peer-to-peer networks, in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pp. 367–378, New York, NY, USA, 2004, ACM.

[3] T. Seenivasan and M. Claypool, Multimedia Tools and Applications **70**, 379 (2014).

[4] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath, Combine: Leveraging the power of wireless peers through collaborative downloading, in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pp. 286–298, New York, NY, USA, 2007, ACM.

[5] B. Cohen, Incentives build robustness in bittorrent, in *Workshop on Economics of Peer-to-Peer systems* Vol. 6, pp. 68–72, 2003.

List of Figures