Ausarbeitung Anwendungen 1 WS2010/11

Fabian Jäger

Scalable Videocodec in a Video Conference

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Topic of this paper

Scalable Videocodec in a Video Conference

Keywords

Videoconferencing, SVC, scalable video codec, bandwidth estimation, adaptive codec, iPhone, mobile video conference, Daviko, Placecam, H.264

Abstract

This work focuses on the design of an adaptive, bandwidth aware streaming strategy for real-time video data in a conferencing system called PlaceCam by Daviko [4]. The background of mobile video conference, bandwidth estimation and codec is researched and presented. The current state of the software is analyzed and we present an approach to adapt the software to mobile devices.

Contents

List of Figures

1 Introduction

Compressing video data in a videoconference is an important task to improve the efficiency. In a videoconference with multiple participants, the hardware and available bandwidth of each participant may differ. Therefore it is an important task to scale the amount and complexity of the videodata for each participant individually. This can be done with the scalable video codec (SVC), which is an extension to H.264. The ability to scale the video data is not a new feature, but rarely used. Because of smartphones which can deliver and receive videos with very good quality, but have often just a small Internet connection, the SVC could be a solution to this issue.

With these codecs it is possible to reduce the amount of data for clients with low bandwidths by dropping certain packages with additional video data that are not necessary to decode the video stream. This allows us, if we know the available bandwidth, to send each client as much data as possible without jamming the connection. This work will focus on the scaling of the video codec depending on the maximum available bandwidth. The purpose of this background research is the design of an adaptive, bandwidth aware streaming strategy for real-time video data in a conferencing system called PlaceCam by Daviko [4].

This paper is organized as follows. In section 2 we give a short overview of the research field. In section 3 we present our motivation and describe the problem in section 4. Section 5 and 6 contains background information to codecs and bandwidth estimation. In section 7 the current state of the video conferencing is described. In section 8 and 9 we present our approach and also the risks visible for our project. Section 10 is the conclusion and outlook.

2 Overview

Vidoeconferencing is not a new phenomenon, but it is still not so commonly used as audio conversation. There are reasons why it has not been spread widely. In the early beginning, special devices like ISDN videotelephones were needed which were very expensive. Videoconferencing with computers became possible with better Internet connections and growing processor power for better enconding algorithms. But we still use in most cases a normal telephone to talk to people. We think this could change with the boom of smartphones. Their processor power grows rapidly and the Internet connection are good enough to handle multimedia. In contrast to computers, smartphones were originally designed for telecommunication which is the reason why we think videoconferencing gets more important for future communication [7].

In this work the focus lies especially on mobile conferencing for more than two participants with different hardware and Internet access. For example some participants could use a normal PC while the rest uses mobile devices like smartphones with a slower connection and fewer hardware resources. Therefore normal computer must take care to not overload the smartphones while a smartphone itself must be aware of its hardware resources and available bandwidth.

Overall we consider a video conferencing software with heterogeneous devices as participants.

3 Motivation

The motivation behind this work is to find a better performance approximation for video conferences where the participants may differ. This is especially interesting since the mobile devices nowadays have the ability to join these conferences. Until now there are no well established solutions on calculating the scale of the codec depending on the bandwidth.

3.1 General Motivation

Mobile videoconferencing and Internet multimedia applications in general are a relatively new research field with much potential for researches [17] [5]. The interesting task is to increase the quality as much as possible with limited hardware resources. There are many parts which influence the qualities on a smartphone. First of all the camera on different devices deliver different picture quality and resolutions. This inflicts the performance of the codec. Also some smartphones are able to make videos in high definition by now, which may overcharge the codec when not scaled down. But the bottleneck is the bandwidth. First of all, the bandwidth is normally very small on mobile devices. Additionally the bandwidth can change rapidly depending on the amount of other network neighbours and how they use their connection. For example, a neighbour that accesses a webpage with large pictures generates a short, bursty data stream, while a user watching a video generates a long but constant data stream. The bandwidth on mobile devices differ from stationary devices mainly in the fact that they change their access point while they are moving. The result is a very unstable sustainable bandwidth which is a big issue for mobile videoconferencing. These unsolved problems are our motivation to develop an adaptive bandwidth approximation and combine the codec with the available bandwidth and hardware resources to get a stable videoconference in every situation and on every device.

3.2 Personal Motivation

In my bachelor thesis I ported the Placecam [4] videoconference software to the iPhone G3. The focus was to explore the possibility of a videoconference on a smartphones its problems. The porting was a success, but the software did not scale very well since it does not sufficiently take care of the available bandwidth. With this work we investigate what needs to be done to make the software more scalable and bandwidth-adaptive. We use smartphones as devices since they are still a very new technology with interesting research potential. We use the iPhone as development platform, because we used it already in the bachelor thesis due to its big community and very stable development tools. For future work a port to another smartphone (maybe with Android [1]) as reference device would be interesting.

Both, the mobile videoconference and smartphones, are new technologies with much potential and are a great theme for researches, which makes it very interesting for us to explore their possibilities.

4 Problem description

The aim of this project is to analyse the interaction between the bandwidth and the codec. The main problems to be solve are the following:

1. We need to determine the currently available bandwidth, which is a pretty hard task. We need a solution which is very fast (because we want a real-time multimedia streaming), stable and if possible without any data overhead. As sustained bandwidth may change rapidly we need to refresh the bandwidth information periodically.
2. H.264 has multiple options for scaling the video data. We must figure out which of them are the best in Quality of Experience for a video conference and if this decision depends on the participants or not. For example a mobile participant with UMTS connection could prefer a mode which is not optimal for a mobile participant with a WLAN link. Unfortunately the scalable video codec enhances processing requirements, so we must be careful to not overload the source of the videostream. So the first thing here is to compare the performance, the amount of extra data and the suitability for our purpose in each mode.
3. The scale of the video codec depends on the observed bandwidth, so we must find a way to adapt these two parts. This issue is not very well known at the moment. We expect to develop our own algorithm. Therefore a lot of experiments are necessary to find an approach that fits the requirements of a videoconference best.

5 Background Codec

Since raw video data is often very large with high redundancy, a codec can be used to compress it. Most of the common codecs are asymetric, which means that the compressing needs much more resources than the decompressing. In common schemes compression fixes the information rate and it is not possible to influence the quality of the stream afterwards. In this section we discuss this issue and present codecs which are a solution to this issue.

5.1 Common Codecs

The difference between video and still image compression is in timescale. Video stream attain high temporal correlations. Predictive interframe codecs like H.261, H.263, H.264, MPEG-1 or MPEG-2 try to predict a frame from the previous frame and if possible also from the following frame (for example if the video is a movie on a harddisk where all frames are available at compressing time). This prediction is done on both ends, the compressing and the decompressing side. The codec tiles each frame in evenly sized blocks which are called macroblocks. The macroblocks in two following frames are compared to each other to find a motion correlation. This motion is saved and can be used to predict the next frame.

To compress data, the predicted frame is compared to the actual picture and only the differences are transmitted along with motion vectors. For decompressing the video the predicted frame is updated with the differences.

Necessary for the prediction is at least one full frame as reference which is called Intra-Frame (I-Frame). The predicted frames are called Inter-Frame (P-Frame if it is predicted only by the previous frame and B-Frame if it is predicted by the previous and the next frame).

Normally an I-Frame is send periodical to make the decompressing more robust. Otherwise if one P/B-Frame gets lost, the whole rest of the video can not be decompressed correctly anymore. After a few Inter-Frames, another Intra-Frame is transmitted and the decompressing will use this one as new reference. This arrangement of Frames like in figure 1 where one Intra-Frame is followed by several Inter-Frames is called Group of Picture (GOP).

5.2 Scalable Video Codec

Normal codecs are not able to scale the quality of the stream. It is for example not possible to drop certain frames to reduce the amount of data, because every following frame relies on

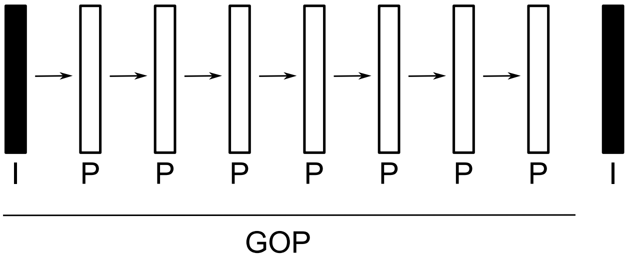

Figure 1: *Group of Pictures (GOP)*

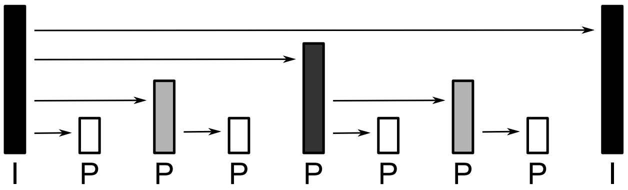

Figure 2: *GOP with temporal scalability*

it. Scalable Video Codecs extends a video stream with enhancement layers which contain additional data but are independent of other frames.

The base layer behaves like a normal codec stream and every other enhancement layer depends on it. It is, like in a normal codec, not possible to drop frames of the base layer and is therefore the minimal quality video and data amount. Every enhancement layer contains additional video data to improve the quality of the video. For the best quality of the video, we need the base layer and all enhancement layers which also increases the transmitted data amount.

It is possible to add more than one enhancement layer in a hierarchical order where every sublayer depends on the layer below. For example if we have a SVC stream with 3 layer (1 base layer and 2 enhancement layer), the base layer is independent of the enhancement layers. The first enhancement layer depends on the base layer but is independent of the second enhancement layer while the second enhancement layer depends on both other layers. This means that if we want also use the second enhancement layer to get the best possible video quality we can not drop any of the frames of the first enhancement layer since the second layer depends on these packages.

5.2.1 Scalability Modes

There are several options of scaling a videostream. The most common modes are temporal, spatial and quality scalability [16].

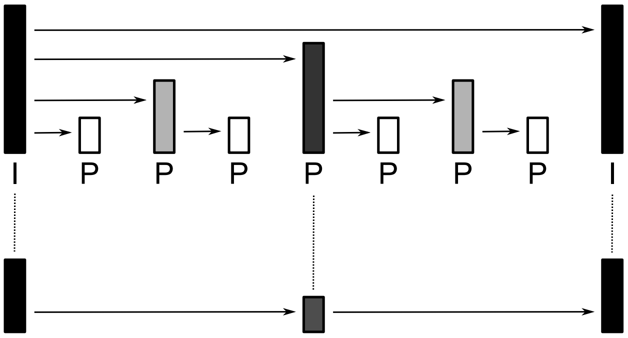

Figure 3: *GOP with spatial/quality scalability*

Temporal scalability reduces the framerate of the base layer which is shown in figure 2. The hight of the blocks indicate the level of temporal layer and therefore the amount of frames which are inter depended. In this case, the little white blocks are frames which are on the highest layer and can be dropped easily while the middle gray blocks can only be dropped if we also drop the lowest layer since they rely on them. The result is a base videostream with a low framerate and may seem jerky to the user.

Spatial scalability reduces the spatial resolution of the base layer and can be increased by the enhancement layers like in figure 3. In this case, the the temporal and spatial scalability are used but the spatial enhancement is only available for the base layer. Anyhow it is technically possible to add spatial scalability also to the other temporal layer. The result is a base videostream with a fixed size but the base videostream has a low spatial resolution which can be increased with the enhancement layers.

Quality scalability reduces the fidelity (signal-to-noise ration gets worse) while the spatio-temporal resolution stays the same. The further process is almost the same as the spatial scalability like in figure 3. The quality scalability is obtained by the factor of quantization. The quantization uses a matrix (based on experience) to filter certain frequencies of a frame. The amount of filtered frequencies influences the quality of the frame. The result is a base videostream generated by a rough quantization matrix which filters high frequencies (low-pass). The enhancement layers use a more precise quantization matrix which allows more frequencies to increase the quality of the video.

Temporal and quality scalability are already supported by the H.264 and SVC adds just a few enhancement information, while spatial scalability is provided by SVC only.

A few more modes exist which are rarely used but could be interesting for a videoconference. Modes like region-of-interest (ROI) and object-based scalability concentrate on a single region or object on the videostream and use the enhancement layers to improve the quality of it [8]. This mode could be of interest for a videoconference where the background often stays the same and only the head of the participant is in motion.

5.2.2 Mixing of Scalability Modes

The scalability modes are not exclusive so they can be mixed to get better compression rates. Also it is possible to use scalable coding in the enhancement layers from other scalability modes. We could for example use temporal scalability in the enhancement layers of the quality scalability mode. Every additional scalability makes a videostream more adaptive - at the cost of a lower encoding performance. So the hardware is an important factor for the level of scalability.

6 Background Bandwidth estimation

Bandwidth estimation is a huge and tricky task. The main problem is the lack of Router information on the Internet. Normally we have an end-to-end connection, where we do not know anything about the infrastructure. We do not know the max. bandwidth, the available bandwidth and where the location of the bottleneck is.

Over the years several tools have been developed which mainly use two different techniques to estimate the available bandwidth.

6.1 Probe Gap Model

The Probe Gap Model (PGM) sends a pair of packets with a predefined gap in between. At the end of the path, the gap is measured again and compared to the initial gap. Every delaying of the second packet, for example if the network is too slow or a packet from the competing traffic arrives between the packet pair, will lead to an increased gap. This is shown in figure 4.

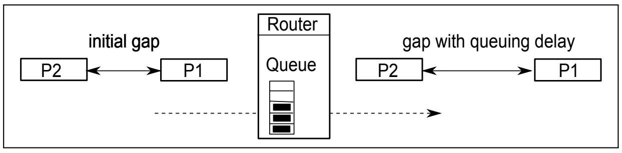

Figure 4: *PGM measures the gap at the beginning and at the end of a path*

The difference between the initial gap and the gap at the end of the path is called dispersion. Every time a queuing delay occurs, the dispersion increases. PGM can be used to measure the capacity and the available bandwidth. If we use it to measure the capacity, we try to reduce the impact of the competing traffic by using a small gap and determine the capacity of the path with the dispersion. If we try to estimate the available bandwidth we use the dispersion to measure the competing traffic on the tight link and subtract it from the capacity of the bottleneck. The result is the available bandwidth. The problem of this method is the assumption that the tight link occurs at the bottleneck. This might be the case in most of the scenarios, but is not always the case.

6.2 Probe Rate Model

The Probe Rate Model (PRM) uses a self-induced congestion on the path. The rate of the probing packets increases slowly and is measured again at the end of the path. If the rate gets higher then the available bandwidth, the rate is slowed down by the tight link, which is noticeable at the end of the path.

Figure 5 shows the one way delay which is measured by the receiver while the probing rate (R) increases. The one way delay does not change as long as the probing rate is lower than the available bandwidth (A). If the probing rate is higher than the available bandwidth we can measure the increasing one way delay.

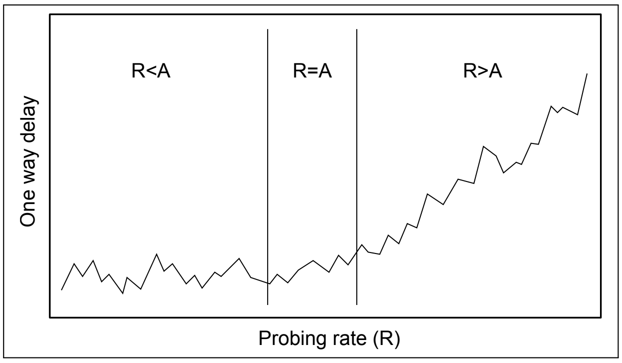

Figure 5: *Results from PRM*

The task is to identify the area where the probing rate is equal to the available bandwidth.

6.3 TCP and flooding

Normally the majority of the competing traffic on an Internet path is TCP traffic. For our project it is an important issue to know how much of the competing traffic is TCP traffic and how much is UDP, because they behave very differently if the path is jammed. TCP is flow controlled and reduces the transfer rate while UDP would not react to it at all.

If we start a videoconference on a path which is fully blocked by a TCP connection, we would measure no available bandwidth. If we start the UDP stream anyway, we would get

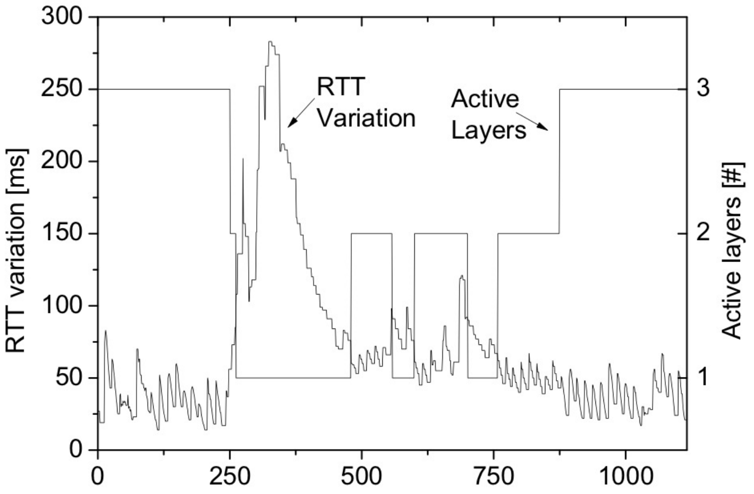

Figure 6: *SVC layer adaptation influenced by the jitter of 1000 testpackages [6]*

bandwidth since the TCP stream would reduce the transferrate. We could increase the UDP rate until the TCP stream is completely suppressed. To avoid this flooding, we have the idea to estimate the capacity of the path and limit the used bandwidth if competing traffic exists. An important question is how much the bandwidth should be limited.

7 Current State

The Session Initiation Protocol (SIP) [14] is used for establishing an conference session, which is very common for an application like this. This is very helpful for us since we can share information about the codec by the Session Description Protocol (SDP) [9].

The software uses RTP [15] for the multimedia streaming and RTCP [10] is used to determine information about the bandwidth. RTCP offers parameters like delay, jitter and round-trip-time. Instead of sending extra measurement packets, we can use the inter-arrival jitter offered by RTCP to determine any changes of delays on routers and switches. This idea is basically inspired by SLoPs, which is an PRM approach and is well tested on other bandwidth estimation software [11]. In contrast to SLoPs our approach uses the delay to determine the bandwidth.

At the moment, the videoconference software uses a very simple mechanism to determine videoscaling. If the jitter increases over a short sequence of packages (e.g. 10) the codec is scaled down by one layer. On the other hand the upscaling is only done if the jitter decreases over a long sequence of packages (e.g. 50) to ensure that the connection is stable. This behaviour is shown in figure 6 which is the result of a test with 1000 packages [6].

The H.264-compliant codec uses the temporal scaling mode and it is not clear if any other modes are available and how stable they work.

8 Approach

The software uses already an SLoPs-inspired algorithm but without any extra probing packages to reduce the probing overhead and also the jitter (2nd derivation of the transmission time) is used instead of the delay (1st derivation of the transmission time) to determine information about the network. The approach to use the jitter as indicator follows the idea that the delay increases continuously when not enough bandwidth is available [13]. This is an approach we like to keep in general but with a much more sensitive interaction with the codec.

In our opinion the temporal scalability is not the best choice for a videoconference since a constant framerate with changing quality is more comfortable to watch than a changing framerate with constant quality. We must compare the modes in matters of performance, package size and sense for the user. We think the the spatial or quality mode is more suitable for a videoconference than the temporal scalability in Quality of Experience. The performance improvement of spatial and quality scalability is +10% rate increase compared to the standard H.264 without scalability [2][3]. Since we can mix different modes we could use them all but must decide which enhancement packages to drop first.

9 Risks

The risks are sectioned in two parts. The first part analizes the non-technical risks like complexity of the project and effectiveness. We try to make a guess how long each part of the project will take and why it could fail. Second the technical risks are analized which mainly depend on the existing videoconference where the scalability shall be implemented. In this section we list the risks of the network measuring for the bandwidth estimation. Also the available codec is inspected and we check in what way the bandwidth estimation and the codec can be combined and where it could fail.

9.1 General risks

- Like in almost all projects the available time is an important factor for the success. We try to split the problem in smaller pieces to get a better overview what needs to be done and how long it will take. We will start with the bandwidth estimation since we think it is the most important component of our project and every other component depends on it. The codec modus, the best usage of the smartphone hardware and the algorithm for the interaction between available bandwidth and the codec will follow.

- The bandwidth estimation could be too imprecise or too slow which would result in useless information for the scalable codec. Since the whole project depends on this information, it is very important to make sure that it will work reliable.
- It is still possible that after the implementation the software does not obtain better results than the old and simple approach. In this case our project would be completely useless. We doubt that this could happen since the old approach is very inaccurate, but it is still a risk.
- 90% of the traffic on the Internet is TCP traffic [12]. Since our videoconference software uses UDP, it is possible to push back the TCP traffic to get more bandwidth for our application. If we are not careful with this, we could easily push back the TCP so far that other applications can be disturbed. This is an issue we must analyse to find a good configuration for all applications on the network.

9.2 Technical risks

- The estimation of the available bandwidth is a big risk since everything else relies on it. Therefore it must be a good and stable bandwidth estimation with very small overhead. Since the software uses RTP [15] we use RTCP [10] to get information about the network path. Also the reaction time is very important for us, since we must react to changes as fast as possible especially when UDP bursts occur and we must scale down.
- We must determine a good algorithm to react fast but not too sensible to bandwidth changes. If we react too fast to little bandwidth changes, constant quality changes could be disturbing for the viewer of the stream. The approach is to react fast to a jammed path but slow to a free path with enough available bandwidth. We increase the quality only if we are sure the bandwidth will be stable over a long time.
- Another issue could be the codec which was developed by the Daviko GmbH. Besides the temporal scalability it is not clear which scalability modes are supported by the Daviko codec and how reliable they are. We think the temporal scalability is not the best choice for a videoconference and would prefer the spatial or the quality scalability.
- For our project we use the iPhone 4G as hardware. We choose this hardware because we already used it for a port of Placecam to mobile devices before, therefore we know that the software will run on this device. But it is still a risk that this device may not

be the optimal choice for our project. Our experience from the porting shows, that the iPhone is not always as open as we wish it would be. The most problems made the access to single frames from the camera which was not allowed by the API and needed a lot of hacks to get working.

10 Outlook and Conclusion

In this paper we showed the problems of videoconferencing in general and especially on mobile devices. The biggest problem is the link capacity. We discussed approaches to overcome this issue and analysed the problems of each part of our solution.

On a long run, the aim is to develop a software which is aware of the hardware resources and the connection to run on all kinds of devices. Therefore the software needs knowledge about the hardware and the Internet connection and also algorithm to scale the video codec. The software is optimized on normal computers at the moment, which we like to change. At the end we hope that we have a stable mobile videoconference which is available in the Apple App-Store.

On the short run it is important to develop algorithms which scale the codec dependent on the bandwidth information. This is not very well researched at the moment and it is hard to find any technical literature to this issue. Therefore we must develop our own algorithm and must test them on different connections. This could take some time to find the optimal parameters for a mobile videoconference.

References

[1] Android. http://www.android.com/.

[2] Conversational applications with quality scalability. http://ip.hhi.de/imagecom_G1/savce/MPEG-Verification-Test/SBC1.htm.

[3] Conversational applications with spatial scalability. http://ip.hhi.de/imagecom_G1/savce/MPEG-Verification-Test/SBC2.htm.

[4] Placecam. http://www.placecam.com.

[5] *End-to-end available bandwidth estimation and time measurement adjustment for multimedia QOS*, volume 3, 2003.

[6] Hans L. Cycon, Valeri George, Gabriel Hege, Detlev Marpe, Mark Palkow, Thomas C. Schmidt, and Matthias Wählisch. Adaptive temporal scalability of h.264-compliant video conferencing in heterogeneous mobile environments, 2010.

[7] Hans L. Cycon, Thomas C. Schmidt, Gabriel Hege, Matthias Wählisch, Detlev Marpe, and Mark Palkow. Peer-to-Peer Videoconferencing with H.264 Software Codec for Mobiles. In Ramesh Jain and Mohan Kumar, editors, *WoWMoM08 – The 9th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks – Workshop on Mobile Video Delivery (MoViD)*, pages 1–6, Piscataway, NJ, USA, June 2008. IEEE, IEEE Press.

[8] Lino Ferreira, Luis Cruz, and Pedro Assunção. H.264/svc roi encoding with spatial scalability. *SigMap*, 2007.

[9] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. RFC 4566 (Proposed Standard), July 2006.

[10] C. Huitema. Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP). RFC 3605 (Proposed Standard), October 2003.

[11] M. Jain and C. Dovrolis. End–to–End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. In *Proc. ACM SIGCOMM'02*, pages 295–308, New York, NY, USA, 2002. ACM.

[12] Sean McCreary and kc claffy. Trends in wide area ip traffic patterns - a view from ames internet exchange. *ITC Specialist Seminar, Monterey, CA*, 2000.

[13] Ravi Prasad, Constantinos Dovrolis, Margaret Murray, and Kimberly C. Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network*, 17(6):27–35, November–December 2003.

[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630.

[15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFC 5506.

[16] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, September 2007.

[17] T. Tunali and K. Anar. Adaptive available bandwidth estimation for internet video streaming. *Signal Processing: Image Communication*, 21(3):217–234, March 2006.