In-Network Adaptation of H.264/SVC for HD Video Streaming over 802.11g Networks

Ingo Kofler, Robert Kuschnig, Hermann Hellwagner
Institute of Information Technology (ITEC)
Alpen-Adria-Universität Klagenfurt
Austria
firstname.lastname@itec.uni-klu.ac.at

ABSTRACT

In this paper, we present an approach for in-network adaptation of H.264/SVC in the context of 802.11 wireless networks. It builds upon our previous work on an adaptive RTSP/RTP proxy which allows to adapt video streams on Linux-based home router platforms. The proposed approach tackles the throughput variations that occur as a consequence of the physical rate adaptation in 802.11 equipment caused by the mobility of clients. By combining monitoring information available exclusively on the wireless router with the ability to adapt scalable video streams on-the-fly, the proposed in-network adaptation approach allows to quickly adjust the video bit rate to the current link conditions. Instead of reacting on packet loss, our approach uses an increase in queueing delay at the router to detect phases of throughput degradation. This allows a higher responsiveness compared to traditional end-to-end approaches that rely solely on RTCP feedback. The behavior of our novel approach was evaluated in several mobility scenarios in an experimental test bed. The results obtained by streaming and adapting high-definition content clearly demonstrate the feasibility and benefits of this approach.

Categories and Subject Descriptors

C.2.6 [[**Computer Communication Networks**]: Internetworking; H.4.3 [**Information System Applications**]: Communications Applications

General Terms

Design, Performance, Experimentation

Keywords

H.264/SVC, Video Streaming, In-network Adaptation

1. MOTIVATION

The advent of the scalable video coding (SVC) extension of H.264/AVC introduced new possibilities in multimedia communication. In contrast to prior approaches, H.264/SVC offers scalability along different adaptation dimensions (temporal, spatial, SNR) at a comparatively low bit rate overhead. In our previous work [5], we proposed a light-weight adaptation mechanism for performing in-network adaptation of H.264/SVC on an off-the-shelf router platform. The approach that utilizes the layered encoding of the scalable video bit stream is based on an RTSP/RTP proxy running on the Linux-based home router. It allows for stateful and signaling-aware adaptation of H.264/SVC streams and therefore meets the requirements of a media-aware network element (MANE) [8]. Our initial evaluations based on a rather modest Linksys WRT54GL router in [5] showed that the adaptation of up to four parallel standard-definition video streams is feasible on such home router platforms. Latest results obtained using more recent router platforms (TP-Link TL-WR1043ND, UBNT Router Station Pro) demonstrated that even the handling of parallel high-definition streams with a cumulative bit rate up to 40 Mbps is possible. While our previous work was rather focused on the adaptation mechanism and protocol details, this paper proposes the application of in-network adaptation to wireless streaming in an 802.11g network. In these networks, streaming high-definition content still imposes challenges in case of sub-optimal wireless link conditions.

Significant research efforts have been made during the last decade in the context of 802.11 networks and video streaming. One of the core problems, however, is the medium access scheme of 802.11 that is used almost exclusively: the Distributed Coordination Function (DCF), which is tailored for best-effort services rather than providing guarantees for real-time applications. The access to the shared medium, the lack of a central point of coordination and the unreliable wireless communication cause a multitude of different research challenges. A lot of work either based on theoretical analysis, simulation or experimental evaluation can be found in the literature that tackles single issues like throughput limitations, contention-based loss, airtime-fairness, service differentiation, etc. But still there is no means for solving all the challenges with 802.11 networking. Following this observation, our work does not claim to do so but rather provides a solution for dealing with the varying throughput in 802.11g networks caused by a single mobile client. Compared to earlier work in this field [3] we make use of H.264/SVC-based in-network adaptation performed on real home router platforms instead of MPEG-2 combined with a PC-based solution. Using H.264/SVC in the context of

802.11 was already proposed in [2], where different layers were mapped to 802.11e access categories. However, the authors of [2] proposed a non-signaling aware approach which uses a fixed mapping and relies on packet dropping by the MAC layer. This is however, not aligned with the general concept of a MANE. Our approach of performing in-network adaptation directly on the router offers two major advantages. First, it is possible to access monitoring information that is only available locally. Second, the availability of this information allows to react much faster to link degradations instead of relying on traditional end-to-end feedback like RTCP reports.

2. LINK RATE ADAPTATION IN 802.11

In wireless networks the mobility of a station typically has a major impact on the wireless connectivity because of fading effects. A specific feature of the 802.11 standard is that it allows to switch the modulation and coding parameters to adapt the robustness of the transmission on a per-frame basis. As the selection of these parameters also influences the achievable transmission rate, this adaptive behavior also dictates the maximum achievable throughput to a single station. The variation in throughput can be very significant since the 802.11g standard offers physical rates between 1 Mbps and 54 Mbps. Furthermore, the overhead of the medium access scheme used by 802.11 devices limit the achievable throughput to values far below these nominal physical rates [4]. The adaptation of the physical rate according to the link conditions is controlled by the rate adaptation algorithm. No particular algorithm is specified in the standard which leaves the implementation up to vendors and researchers. Consequently, a variety of different algorithms [1] were proposed in the last decade. The algorithms typically use heuristics and transmissions statistics like packet loss and signal strength measurements [9] to decide which PHY rate to use for transmitting the next frame.

In the context of Linux-based devices, the *minstrel* algorithm turned out to perform very well under various conditions as investigated in recent work [10]. It uses packet loss as an indicator and maintains throughput and reliability statistics for all available physical rates. Additionally, it frequently uses probing frames to determine if it is possible to use a higher physical rate than currently in use. The minstrel algorithm is used as the default rate control algorithm on our Linux-based router platforms. In the Linux kernel, these algorithms are however separated from the actual drivers of the wireless devices. This allows to experiment with different algorithms or to implement another algorithm by providing an own kernel module. This modular concept was also utilized for our evaluation as discussed later.

One can conclude that in current 802.11 networks there is no way to guarantee a certain bandwidth to a mobile station for video streaming or other real-time services. Apart from 802.11e, which allows a basic service differentiation, the 802.11 standard lacks real QoS mechanisms for such services. This situation requires adaptive solutions on the application layer which can cope with varying link throughput. In the following, we propose an adaptive approach which adapts a scalable video stream according to the changing networking conditions caused by the mobility of a single client. We assume that the bottleneck is the wireless network and not the wired networks (Ethernet, VDSL) involved in the content delivery chain. In these wired networks the required bandwidth for high-definition content can be typically provisioned as it is already the case for VoD services in current IPTV deployments.

3. H.264/SVC IN-NETWORK ADAPTATION

In order to support the adaptation of the video stream according to the varying throughput of the 802.11 link, we extended the architecture of the RTSP/RTP proxy in [5] by an Adaptation Decision Taking and Monitoring component. Its task is to control the adaptation of the scalable video content based on the current wireless link conditions. For that purpose it makes use of the scalability information exchanged during the RTSP session setup to learn which layers (and resulting video bit rates) are contained in the video bit stream. In this work, we put the focus on spatial adaptation only. However, the concept can be easily applied to temporal or quality scalability as well.

Monitoring is based on the Linux monitoring interface offered by the *mac80211* wireless stack used by many wireless network drivers. It allows to obtain information of all packets that are transmitted or received via the wireless interface by the router. The information includes the packet's payload as well as details about the physical rate, the number of retransmissions used, etc. Obviously, this information can only be provided after the packet was successfully transmitted and acknowledged by the client. Consequently, this monitoring interface can be used to make acknowledgements at the link layer visible to the application layer. In our approach, the proxy uses this feedback to estimate how long the packet was queued in the networking stack at the router. This estimation is accomplished by keeping track of the timestamps of when a packet was sent via the socket API and when its acknowledgement was notified via the monitoring interface. The differences between both timestamps obviously also contain the serialization and propagation delays of the wireless transmission. However, it turned out that the monitored delay can be on the order of several hundred milliseconds which renders these two components rather negligible. The difference of the timestamps is therefore considered as an estimation of the queueing delay and is monitored on a per-packet basis. The obtained values are smoothed by the proxy using an exponential weighted moving average (EWMA).

The averaged queueing delay is subsequently used for controlling the in-network adaptation of the video bit stream. In contrast to traditional adaptation approaches that rely on packet loss as feedback to trigger adaptation, using the queueing delay allows to react much faster to varying throughput. Considering the queuing delay to control video adaptation is proposed among others in [7]. The proxy can detect a throughput degradation by a sharp increase of the queueing delay and decrease the video bit rate immediately. This prevents or at least reduces packet loss *a priori*, in contrast to relying on packet loss to trigger the adaptation *a posteriori*. The Adaptation Decision Taking and Monitoring component follows a rather simple but effective control mechanism to vary the video bit rate depending on the encountered queueing delay. If the queueing delay exceeds a certain threshold, the proxy determines the link's throughput during the last 200 ms. This comparatively short interval was chosen based on the observations in [9] which indicate that it does not make sense to incorporate packet history older than 150 ms to 250 ms due to the fast changing wireless conditions. Based on this current throughput, the proxy decides which spatial layers depending on its video bit rate requirements can be currently served. This decision also considers overhead introduced by the packetization and transport as well as a share of remaining bandwidth to drain the queue at the router again. The rationale is to immediately reduce the video bit rate to prevent packet loss.

If, on the other hand, the proxy encounters a very low queueing delay during a configurable interval, this is considered as an indicator for low link utilization. In this case, the current capacity of the wireless link might allow to stream the content in a higher spatial resolution. The proxy however, does not immediately switch to a higher layer but first performs a capacity estimation. As proposed in a recent paper [6], a packet pair technique can be used to estimate the capacity of the wireless link. In contrast to the approach in [6], our approach does not need to transmit explicit probing packet pairs. Instead, it uses the collected timestamps of the packets transmitted during the last second. In our experiments, it turned out that in the case of HD content many packets are sent back-to-back by the proxy and can be used for a packet pair approach. Even in the case of non HD content, the proxy could simply generate packet pairs by transmitting the same RTP packet twice without influencing the video streaming itself. Another advantage of our approach is that the packet dispersion must not be measured by the client. Since the proxy is aware of the link layer acknowledgements, it can easily calculate the packet dispersion on the router. Consequently, our approach does not need any additional interaction or support by the client. Based on the capacity estimation, the proxy decides to switch up to a higher spatial layer, depending on whether the estimated capacity is sufficient for the increased video bit rate or not.

In steady-state operation, the proxy typically encounters an average queueing delay between two thresholds and remains in serving the current spatial layer. At the beginning of the streaming session, the proxy follows a rather conservative approach and starts with serving the base layer only. For our evaluations, the thresholds were determined empirically from results of previous experiments. The threshold to switch down was set to 150 ms, while the capacity estimation and possible switch up was triggered after the queueing delay was permanently below 50 ms for a whole second. In the following, we provide an experimental evaluation of our proposed approach for in-network adaptation. Again, in this evaluation we focus on the adaptive wireless streaming to a single mobile client without considering any cross traffic on the network.

4. EVALUATION METHODOLOGY

Evaluating the performance of wireless systems under mobility aspects is a difficult task. In order to obtain reproducible results, we chose the following novel approach consisting of two distinct steps.

First, we collected traces of the wireless transmission conditions for four representative mobility scenarios. The traces were obtained in an office environment at the university. The floor plan is shown in Figure 1. The position and movement of the mobile client is represented by the numbers 1 – 4 in the plan. The wireless router, denoted as access point (AP), was positioned at the corridor, while the mobile receiver was positioned at, and moved to, different places. The first two scenarios do not include movement, but represent the best and worst case where the receiver is located near (scenario 1) and very far (scenario 2) from the access point. The other two scenarios (scenarios 3 and 4) cover the movement at pedestrian speed away from the access point into two different directions as indicated by the arrows. The traces were obtained by transmitting UDP packets at a constant rate of 20 Mbps to the receiver and measuring per packet the signal strength and physical rate at the receiver, among other parameters. All of the traces consist of data collected over a time of 50 seconds and represent the behavior of the minstrel algorithm used by the router platform. Figures 2 and 3 show the traces for scenarios 2 and 3, respectively. Although evaluated, scenario 1 and 4 are not explicitly shown in this paper due to space constraints. However, scenario 1 can be summarized as a best-case scenario characterized by perfect reception conditions and the usage of the highest physical rate of 54 Mbps for more than 99 percent of the packets. In scenario 2, the receiver encounters a quite low signal strength of approx. -80 dBm, resulting in the selection of a modest physical rate of 11 Mbps for a vast majority of the packets. After 40 seconds a further degradation of the signal strength can be observed, which causes the minstrel algorithm to use even lower physical rates for some seconds. Scenario 3 consists of a movement away from the access point within the first 30 seconds while remaining at the end position for further 20 seconds until the end of the experiment. The signal strength measured for each frame shows a steady degradation during the actual movement of the receiver. However, the impact on the physical rate used by the minstrel algorithm does not show such steady degradation. Instead, the majority of frames is transmitted at the highest rate (54 Mbps) until second 25, followed by a transition phase of less than 5 seconds where the physical rate drops to 11 Mbps and even lower. After that, the physical rate settles at 11 Mbps although the minstrel algorithm continuously probes higher rates. The trace obtained in scenario 4 (not shown here) indicates a consistent behavior characterized by a smooth degradation of the signal strength and a sudden drop of the physical rate.

In the second step, the distribution of the physical rates used within a given time interval was determined based on the traces. The obtained distributions were integrated into our Linux kernel module *phyrateemu* that implements the interface for rate control algorithms. The wireless driver was configured to use the *phyrateemu* module instead of the minstrel algorithm. This means that instead of choosing the physical rates based on the current conditions, the physical rates were selected according to the obtained traces. Consequently, this allowed us to replay the mobility patterns

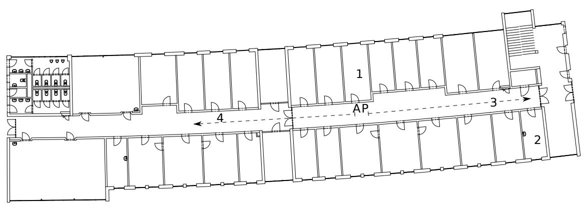

Figure 1: Floor plan showing the different scenarios

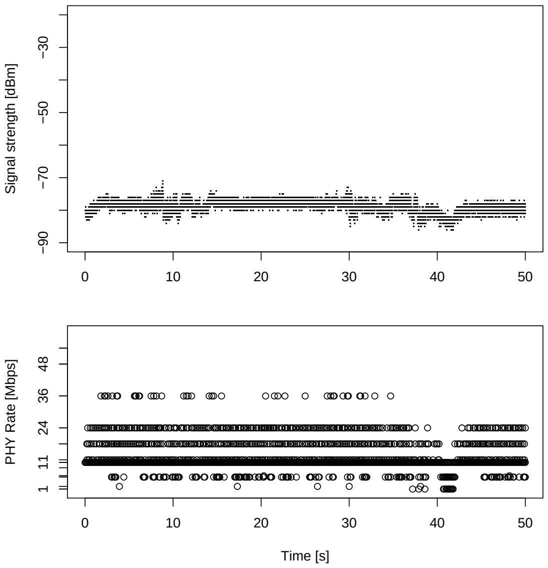

Figure 2: Signal strength and PHY rate in scenario 2

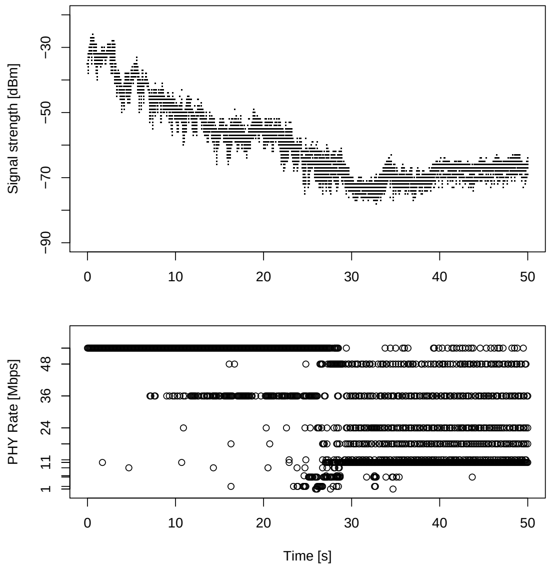

Figure 3: Signal strength and PHY rate in scenario 3

for the actual evaluations and to investigate the impact of adaptation in a reproducible manner.

The experimental setup for the evaluation consisted of a desktop PC acting as the streaming server, the home router running the proxy application and a laptop PC representing the client. The desktop PC was connected to the router via Gigabit Ethernet, while the client was connected using 802.11g. The client requested the scalable video stream using the openRTSP command line tool of the live555 library[1]. At the server side, the scalable video content was streamed using Apple's Darwin Streaming Server (DSS)[2]. The TP-Link TL-WR1043ND router platform was used for this particular evaluation.

The evaluation was performed using different high-definition video sequences. Again, only the results for a single sequence (tractor) can be discussed below due to space constraints. The sequence was encoded using JSVM reference software version 9.19.8. The base layer represents the content in 1024x576 resolution at 50 fps. The two spatial enhancement layers provide the possibility to increase the resolution to 1280x720 and 1920x1080, respectively. The sequences were encoded to achieve a constant bit rate for all of the three possible resolutions. In the case of the tractor sequence, the base layer results in a bit rate of 5 Mbps, while the additional spatial layers lead to a total bit rate of 9.8 (720p) and 15 Mbps (1080p). The media content was served in an infinite loop until the client explicitly closed the streaming session at the end of the experiment. The experiments were repeated at least three times to ensure consistent results.

5. EVALUATION RESULTS

In the following, the results of our experimental evaluation are presented. The metrics used for the evaluation are the throughput and the packet loss monitored at the client. Additionally, the average queueing delay at the router was measured during the experiment. We refrained from using video quality metrics like PSNR as they are not really representative when performing spatial or even temporal adaptation. Additionally, this metric is also influenced by the encoding settings as well as by the error concealment mechanisms employed at the decoder, e.g., in case of lost frames. Therefore, we rather focused on the networking point of view and argue that the video consumption at the client should not be negatively influenced by high packet loss and/or packets arriving too late at the client.

Again, the results of scenario 1 and 4 are not explicitly shown due to lack of space but are briefly discussed in the text. In scenario 1, the mobile client is located near the access point and enjoys good wireless connectivity, which means that the vast majority of packets are transmitted at the highest PHY rate of 54 Mbps. Consequently, the video stream can be served in the highest resolution (1080p) since the capacity of the link is sufficient to cope with the video bit rate of 15 Mbps. In fact, no adaptation would be necessary. However, because of the conservative strategy followed by the proxy only the base layer of the video stream is served at the beginning of the session. Since the encountered queueing delay is below the threshold during the first second of streaming, the proxy immediately performs the capacity estimation and ultimately switches to the highest resolution (and bit rate) after the first second of streaming. As a consequence of the high link capacity, the queueing delay remains at a low level which causes the proxy to maintain its steady-state operation and continues serving the highest resolution.

Scenario 2 can be characterized by bad wireless connectivity, which causes a majority of the packets to be transmitted at a PHY rate of 11 Mbps. Figure 4 illustrates the impact of these conditions on the video streaming if no adaptation would be performed and the video would be transmitted at the highest spatial resolution. The achieved throughput is around 6 Mbps, which is approximately the throughput that

[1]http://www.live555.com/liveMedia
[2]http://dss.macosforge.org

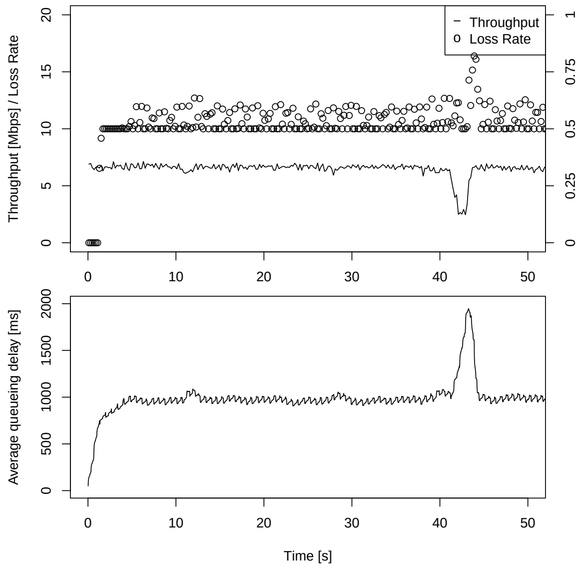

Figure 4: Scenario 2 without adaptation

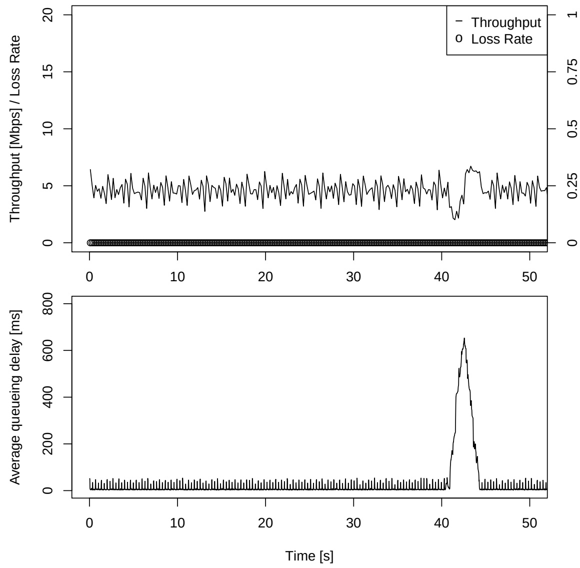

Figure 5: Scenario 2 with adaptation

can be achieved in an 802.11g network when using a physical rate of 11 Mbps and packet sizes of around 1500 bytes. As the video stream at the highest resolution requires 15 Mbps, the queue at the router quickly builds up and consequently the average queueing delay at the router quickly increases up to 1 second. If the maximum size of the queue is exceeded, the RTP packets are getting dropped at the router. This can be observed by the client as a loss rate of more than 50 percent. During the short interval around second 40 where the throughput is even lower, the queueing delay as well as the packet loss further increases.

The impact of our proposed adaptation mechanism in scenario 2 is illustrated in Figure 5. Again, the proxy starts to serve the base layer of the video (approx. 5 Mbps) at the beginning of the streaming session which can obviously be served under these link conditions. Although the average queueing delay is at a comparatively low level, the proxy does not switch to a higher resolution since the capacity estimation indicates that the capacity is insufficient to serve a higher layer. Therefore, the proxy continues to transmit only the base layer in standard-definition (576p) to the client. As it already serves the base layer only, there is also no room for further adapting the video during the short decrease of throughput around second 40. Instead, the throughput shortly degrades and an increase of the average queueing delay can be observed. However, as this degradation only takes place over a short period, the queue at the router can handle this fluctuation and no packet loss occurs. The average queueing delay, however, exceeds 600 ms during that time. Obviously, the encoding parameters of the video dictate a lower bound on the bit rate and therefore limit the operating range of approaches that rely on these scalability features.

From an adaptation point of view, the scenarios 3 and 4 are the most interesting ones. In both scenarios, the sudden drop of the physical rate also leads to a quick degradation of the video throughput, as shown in Figure 6 representing scenario 3. If no adaptation is performed, a significant increase of queueing delay takes place within a few seconds, followed by a sudden increase of the packet loss. It should be noted that the observed increase in the average queueing delay precedes the packet loss by less than 2 seconds. This finding also confirms the decision to use the queueing delay rather than the encountered packet loss as the triggering event for adaptation decision-taking.

The impact of applying our approach to scenario 3 is illustrated in Figure 7. Again, the streaming starts with the delivery of the base layer and a switch to the highest spatial layer after 1 second due to sufficient capacity. During the short transition phase in which the physical rate rapidly decreases, the queueing delay exceeds the threshold of 150 ms and triggers decreasing the video bit rate. As the new adaptation parameters are applied immediately, the switch to a lower layer happens instantly. As a consequence of this quick response by the proxy, the average queueing delay can be kept below 400 ms. After the link conditions have stabilized and the majority of packets are sent at a physical rate of 11 Mbps, the proxy remains at serving the base layer with a bit rate of approximately 5 Mbps. Consistent results were obtained for scenario 4 where the physical rate degrades in a similar way due to the mobility of the client.

In summary, the proposed control mechanism to adjust the video bit rate according to the available monitoring data works very satisfactory. In all of the four evaluated scenarios, it succeeds in delivering the appropriate quality that is feasible with the current network conditions. The different evaluation runs for scenarios 3 and 4 have shown that no or only a few (not shown in this figure) packets are lost during this period. This enables a smooth playback of the received video at the mobile client without encountering severe service disruption for longer periods.

6. CONCLUSIONS

In this paper, we present an application of our previous work on in-network adaptation of H.264/SVC to wireless networks. Compared to traditional video coding, H.264/SVC allows for computationally cheap video adaptation compared

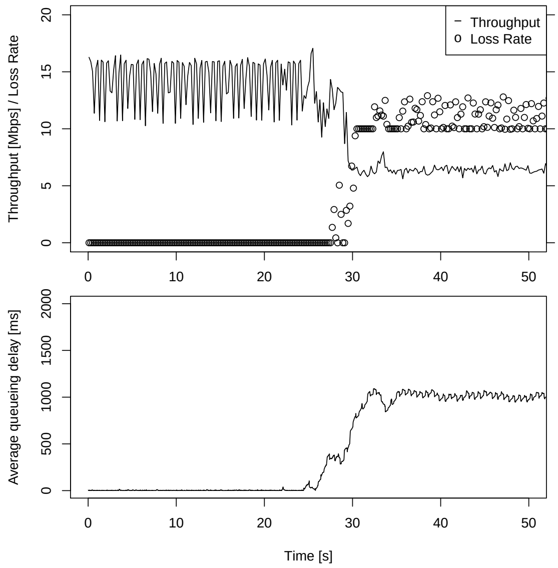

Figure 6: Scenario 3 without adaptation

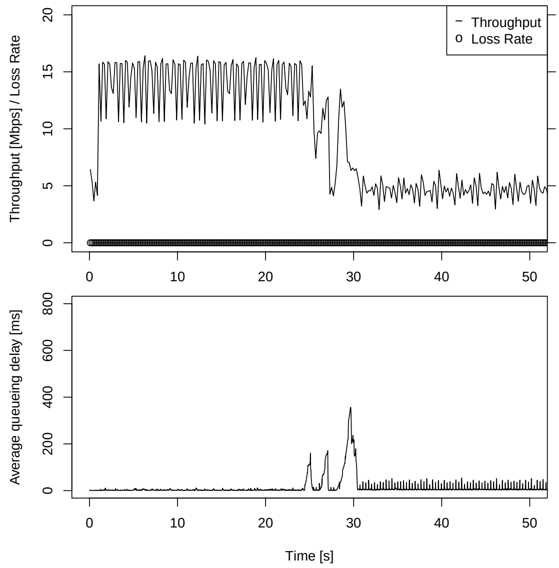

Figure 7: Scenario 3 with adaptation

to traditional video coding. This makes the in-network adaptation of high-definition streams on router platforms possible at all. Our proposed approach uses monitoring information that is available locally on the router to adjust the video bit rate according to the varying link throughput. In our work we focus on throughput changes caused by the mobility of a single client in combination with the multi-rate operation of 802.11. The adaptation is performed by an application-layer proxy-based approach which can be characterized as stateful and signaling-aware. The advantage of our approach is to use monitoring information, more particularly the queueing delay, on the router to control the adaptation. In contrast to control mechanisms that use packet loss as feedback, our approach detects changing link throughputs earlier and prevents or at least reduces packet loss. This information is obtained via a monitoring interface that allows applications to be notified of link layer acknowledgements. The same mechanism is used further to estimate the link capacity using a packet pair approach without requiring any support by the client. The proposed approach was successfully evaluated in the context of adapting different high-definition video streams in different mobility scenarios. Although the evaluation in this paper was based on spatial adaptation only, our concept can be easily applied to temporal or quality scalability as well. The main advantages of in-network adaptation compared to traditional end-to-end approaches are the monitoring information and its responsiveness. This means that it can make use of monitoring information that is only available locally, like the actual queueing delay, and react much faster as compared to end-to-end approaches that might use RTCP only. In the case of 802.11 networks where the physical rate changes quite fast, the typical RTCP feedback provided in intervals on the order of 5 seconds is obviously far too slow.

7. REFERENCES

[1] S. Biaz and S. Wu. Rate Adaptation Algorithms for IEEE 802.11 Networks: A Survey and Comparison. In *Proceedings of the ISCC'08*, pages 130–136, July 2008.

[2] H.-L. Chen, P.-C. Lee, and S.-H. Hu. Improving Scalable Video Transmission over IEEE 802.11e through a Cross-Layer Architecture. In *Proceedings of the IWCMC'08*, pages 241–246, Aug. 2008.

[3] I. Djama and T. Ahmed. A Cross-Layer Interworking of DVB-T and WLAN for Mobile IPTV Service Delivery. *IEEE Transactions on Broadcasting*, 53(1):382–390, Mar. 2007.

[4] J. Jun, P. Peddabachagari, and M. Sichitiu. Theoretical Maximum Throughput of IEEE 802.11 and its Applications. In *Proceedings of the NCA'03*, pages 249–256, May 2003.

[5] I. Kofler, M. Prangl, R. Kuschnig, and H. Hellwagner. An H.264/SVC-based Adaptation Proxy on a WiFi Router. In *Proceedings of the NOSSDAV'08*, pages 63–68. ACM, May 2008.

[6] M. Li, M. Claypool, and R. Kinicki. WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks. In *Proceedings of the LCN'08*, pages 374–381, Oct. 2008.

[7] P. van Beek and M. U. Demircin. Delay-Constrainted Rate Adaptation for Robust Video Transmission over Home Networks. In *Proceedings of the ICIP'05*, Sept. 2005.

[8] Y.-K. Wang, M. M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger. System and Transport Interface of SVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1149–1163, Sept. 2007.

[9] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *Proceedings of the MobiCom'06*, pages 146–157, Sept. 2006.

[10] W. Yin, K. Bialkowski, J. Indulska, and P. Hu. Evaluations of MadWifi MAC layer rate control mechanisms. In *Proceedings of the IWQoS'10*, June 2010.