

# Adaptive Video Streaming over a Mobile Network with TCP-Friendly Rate Control

Ktawut Tappayuthpijarn<sup>1,2</sup>, Guenther Liebl<sup>1</sup>, Thomas Stockhammer<sup>1</sup> and Eckehard Steinbach<sup>2</sup>

<sup>1</sup>Nomor Research GmbH, Brecherspitzstrasse 8, 81541, Munich, Germany

<sup>2</sup>Institute for Media Technology, Technische Universität München, Arcisstrasse 21, 80333, Munich Germany

ktawut@nomor.de, liebl@nomor.de, stockhammer@nomor.de, eckehard.steinbach@tum.de

## ABSTRACT

This paper investigates the performance of TCP-Friendly Rate Control (TFRC) to control the transmission rate of scalable video streams when used in a mobile network. The streams are encoded using the Scalable Video Coding (SVC) extension of the H.264/AVC standard. Adding or removing the layers is decided based on the TFRC during varying channel conditions of the mobile network. We conduct simulations in various realistic use cases, evaluate and compare the performance with and without TFRC-based adaptation. The results show significant improvements in terms of lower loss rate, delay, required buffer size and less playback interruption.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design – *wireless communication*

## General Terms

Algorithms, Performance, Experimentation

## Keywords

TCP-Friendly Rate Control, H.264 SVC, Mobile network

## 1. INTRODUCTION

The mobile radio channel exhibits fast-changing channel characteristics. The Signal to Interference and Noise Ratio (SINR) can vary significantly as the users move to different locations due to fading, shadowing and interference. These variations, coupled with other factors such as varying traffic load and priority assignments in the cell, often significantly restrict the quality. The stringent QoS can only be maintained if the required bitrates and delays are not too challenging. Mobile networks such as HSPA, WiMAX and LTE are likely to be able to provide guaranteed throughput to the streaming users if system load is low. However, this generally comes at the expense of wasting large amount of radio resources in bad channel situation to maintain the guaranteed service quality, implying that fewer users are admitted

to the cell. On the other hand, it is difficult to meet the target qualities for all users in a heavily loaded situation, e.g. some users would have to be dropped or suffer degraded QoS.

With the availability of the SVC extension of the H.264 codec [1], [2], it is now possible to add or remove enhancement layers from the video stream so that its transmission rate can be adapted on the fly. However, what is still missing to integrate SVC in mobile networks is a mechanism to detect changing channel conditions and decide on the appropriate transmission rate such that the exact number of layers to be added or removed can be determined.

Although TFRC [3] is a rate control mechanism designed for streaming services over a typical Internet environment, there have been other related works such as [4], [5] and [6] that also introduce TFRC into the wireless domain. However, their focus is on modifying the TFRC's reaction to non-congestion related losses from the wireless channel. This is not the case for next-generation 3.5G and 4G mobile networks where there are several retransmission mechanisms in their Media Access Control (MAC) and Radio Link Control (RLC) layers to reduce losses from the wireless channel to virtually non-existence but rather increased end-to-end delay and reduction in throughput instead. This work evaluates the performance of TFRC as a rate-controlling mechanism for SVC streams in reacting to typical characteristics of mobile networks such as rapid changes of the throughput. This can be used as a basis for further improvements for TFRC in the 3.5G and 4G mobile environments in future work.

The remainder of the paper is organized as follows. Section 2 provides an overview of the SVC extension to H.264/AVC. A brief introduction on the TFRC is provided in Section 3. Section 4 covers the emulator for the LTE mobile network along with the TFRC component, the SVC client/server application, the test SVC stream and the adaptation strategy used in the simulations. Section 5 presents simulation results. Finally conclusion and future work are covered in Section 6.

## 2. SCALABLE VIDEO CODING

SVC is an extension to the H.264/AVC standard. It is classified as a layered video codec which can encode a video stream in several types and numbers of enhancement layers on top of the H.264/AVC-compatible base layer. These enhancement layers can be added or removed from the bit stream during streaming without re-encoding of the media.

There are three main types of scalability in SVC. The first type provides a spatial enhancement layer which allows the stream to be decoded in several resolutions. Switching between spatial

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IWCMC'09, June 21–24, 2009, Leipzig, Germany. Copyright © 2009 ACM 978-1-60558-569-7/09/06...\$5.00

layers can be done only at some specific points in the stream called the Instantaneous Decoder Refresh (IDR) frames. Typically this is at the beginning of each Group of Picture (GoP).

The temporal enhancement layers are encoded using hierarchical B frames. Finally, the SNR enhancement layers is simply spatial enhancement layers but with the same resolutions as the base layer for inter-layer prediction. This allows for encoding the same spatial resolution in several layers but with finer quantization parameters resulting in higher PSNR. Adding and removing the SNR enhancement layers can be done at any point in the stream.

Figure 1 shows an example of an SVC stream with 1 spatial layer, 3 temporal layers and one SNR enhancement layer. The intra-frame and inter-frame prediction dependencies for decoding are shown with arrows. For more detailed information on the SVC, interested readers are referred to, for example, [1] and [2].

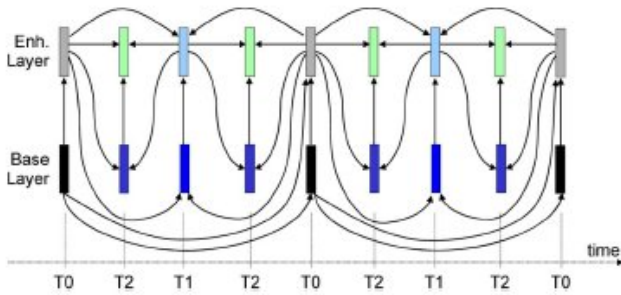


Figure 1. An example GoP structure of a SVC stream (reproduced from [2])

### 3. TCP-FRIENDLY RATE CONTROL

The TFRC was originally designed as a rate control mechanism for video streaming in an Internet environment for which the fairness with competing TCP flows in the network is of importance. The TFRC provides a sending rate within a factor of two of the sending rate a TCP flow would have under the same condition but with relatively more stable throughput which is a desirable characteristic for a streaming service.

The TFRC is a receiver-based mechanism where the receiver performs some calculation of the congestion control indicators and reports them back to the server. It relies on the underlying transport protocol such as the DCCP [7] to provide means for the exchange of control information between the server and the client. An earlier solution prior to the development of DCCP was to use a modified RTP/RTCP protocol to carry such information as shown in Figure 2.

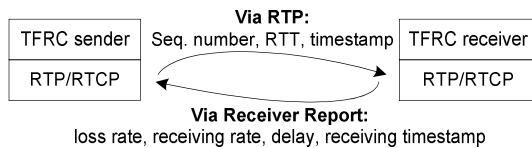


Figure 2. TFRC information exchange over RTP / RTCP

The TFRC sender uses the RTP header to carry the packet sequence number, its current estimate of the Round Trip Time (RTT) and the sending timestamp to the receiver. In return, the

receiver uses this information to calculate, amongst other parameters, its average receiving rate during the past  $RTT$  period and the loss event rate ( $p$ ). This information is reported back to the sender via the RTCP Receiver Report message.

Upon receiving a Receiver Report, the sender updates its  $RTT$  estimation and its sending rate. The algorithm used to calculate the next sending rate depends on whether the sender is still in the initial Slow Start phase or in the Congestion Avoidance phase.

In the Slow Start phase, the sender approximately tries to double its sending rate every time a Receiver Report is received in order to reach the maximum throughput the channel can support which can be detected by increasing  $RTT$  and losses. Once the first loss has been detected, the sender enters the Congestion Avoidance phase. The next sending rate  $X$  is now determined from the minimum between twice the previous receiving rate and the sending rate as calculated from the TCP throughput equation.

$$X = \min ( \text{TCP throughput}, 2 * \text{receiving rate} ) \quad (1)$$

The TCP throughput used in [3] depends on the  $RTT$ , the packet size  $s$ , the TCP retransmission timeout  $t_{RTO}$  and the  $p$  as follows.

$$x_{tcp} = \frac{s}{R \sqrt{\frac{2p}{3} + t_{RTO}} \left( 3 \sqrt{\frac{3p}{8}} \right) * p (1 + 32p^2)} \quad (2)$$

As for the receiver, it continuously monitors sequence numbers of received packets, detects if losses occur and calculates the receiving rate and the loss event rate. The receiver sends the Receiver Report back to the server approximately every  $RTT$  or as soon as a new loss event has been detected. This means the TFRC sender updates its next sending rate roughly every  $RTT$  as well.

## 4. THE TESTBED

### 4.1 The LTE Emulator

The LTE emulator is a real-time emulator that accurately models the characteristics of the LTE mobile technology [8]. This emulator is used as a basis for building up the testbed in this work. Figure 4 depicts the functional elements of the emulator. All major elements required for real-time LTE protocol simulation at the MAC, RLC, and PDCP layer have been implemented and are compliant with the respective 3GPP specifications. Note that the LTE emulator allows for having one external traffic sink and source each to connect to it. We refer to this external traffic sink as the user 1 for the rest of the paper.

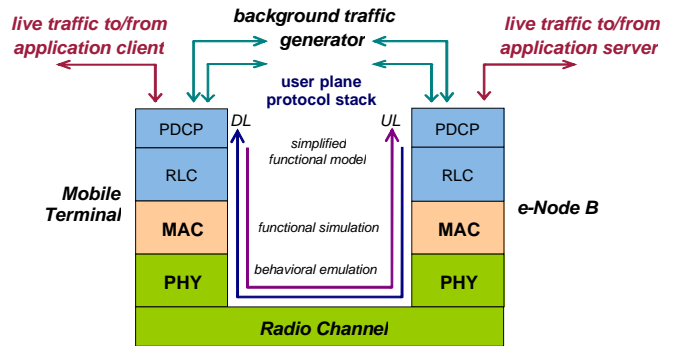


Figure 4. Functional elements of the LTE emulator

## 4.2 TFRC and the Client / Server Application

Figure 5 shows the end-to-end testbed used for the simulations. It comprises of the LTE emulator connected to 2 other computers, one represents the server or an adaptive media gateway close to the core network and another one for the user 1. Both the server and the user 1 have a TFRC entity running on top of RTP/RTCP. The core network represents the core network of the mobile operator which is simulated by adding a constant delay of 35 ms to all packets traversing it, or an additional of 70 ms to the *RTT*.

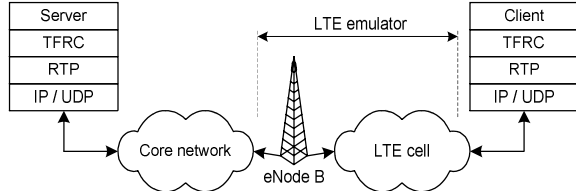


Figure 5. Overall testbed layout

The adaptation cycle at the server begins with the server adapting the rate of the next GoP to be sent according to the sending rate from the TFRC. In cases several sending rates have been given from the TFRC in the last GoP period, a weighted average sending rate  $X$  for the next GoP is calculated as

$$X = \left( X_1 + \frac{1}{2}X_2 + \frac{1}{3}X_3 + \frac{1}{4}X_4 + \dots \right) / \left( 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots \right) \quad (3)$$

where  $X_i$  is the most recent sending rate,  $X_2$  is the next older one and so on.

The adapted GoP will be put into a sending buffer in which the server can send data from this buffer exactly at the sending rate allowed by the TFRC at any given time. Once the next GoP is due to be forwarded to the sending buffer, the cycle repeats again.

In addition, the client has the capability to request retransmissions of lost packets detected at the application layer up to 4 times before considering the packets as unrecoverable and move on.

## 4.3 The Test Streams

The test video clip is made from repeating the ‘‘Crew’’ sequence resulting in a 60-second long clip. It was encoded by the JSVM reference software [9] at 4CIF resolution and 30 fps. There are 2 versions of the encoded stream, one as a H.264 AVC stream for a reference case where there is no adaptation and another one as a SVC stream when adaptation is applied to the stream.

The SVC stream has 1 SNR enhancement layer and 5 temporal layers. Removal of SNR enhancement packets can be done from the highest temporal layer to the lowest temporal layer without removing the base layer packets at all, allowing the frame rate to remain constant at 30 fps even though SNR enhancement packets are removed. This results in a total of 6 extractable sub-streams to use with bit rates ranging from 300 Kbps to 1 Mbps and varying PSNR as shown in Figure 6. Note that the R-D curve for the AVC stream which is also shown represents the resulting PSNR of the stream with different frame rates, e.g. when using the AVC stream for adaptation by reducing its frame rate instead. This demonstrates the fact that using an SVC stream with SNR enhancement layers is more preferable to an AVC stream in terms of the resulting quality if adaptation is to be applied on the stream.

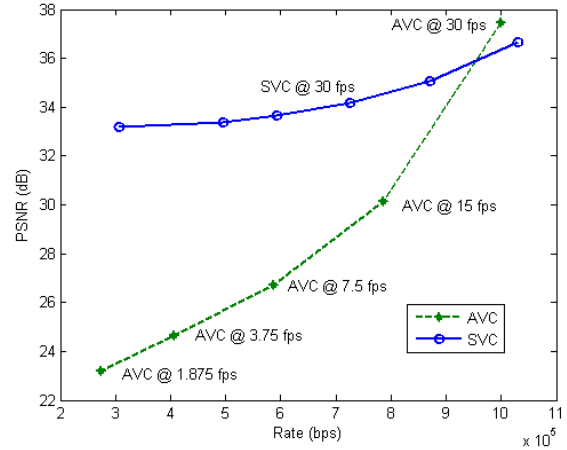


Figure 6. Rate-Distortion curve for the SVC test stream with one SNR and 5 temporal enhancement layers compared to the AVC test stream using only temporal adaptation

## 5. SIMULATIONS

### 5.1 Simulation Configurations and Scenarios

In all simulations, we set the system bandwidth of the cell to be 5 MHz with proportional fair MAC schedulers used for both downlink and uplink. There are different types of users in the cell, namely, voice users with the highest priority, video streaming users and best effort FTP users to simulate a realistic combination of cross traffic with different guaranteed QoS levels in the cell. The streaming users all have a soft delay guaranteed to be 2 seconds and the guaranteed throughput of around 1 Mbps each.

There are 3 main scenarios considered in this work. The first is a reference scenario in which all users are pedestrian users with a total of 7 streaming users in the cell, including the user 1 and no rapid channel changes. This shows that the cell is under a loaded situation but can still provide guaranteed QoS to all the users.

The second scenario is the same as the reference one but with rapid changes in reception quality for user 1. This represents the case where there are sudden changes in the channel quality such as when the user walks into a building or a tunnel.

For the last scenario, all the users move at vehicular speed, 30 km per hour. The channel does not rapidly change as in the second scenario, but gradually changes as the user moves to different areas shown in Figure 7. In this case, the channel estimation at the base station becomes more inaccurate due to increasing user’s speed which results in higher loss and retransmission rate. It also reduces the total cell’s capacity such that the number of streaming users had to be reduced to 2 in order to maintain adequate throughput to other users.

In each scenario, we ran 3 simulations for streaming without the TFRC and 3 simulations for streaming with adaptation.

### 5.2 Results and Analyses

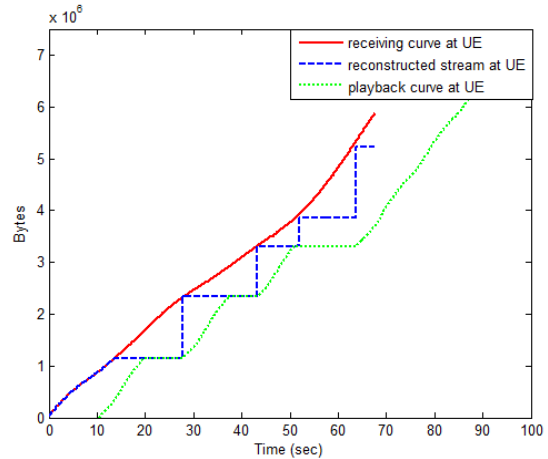
Due to space restriction, we will show selected results in details only from the 3rd scenario, streaming to fast-moving users.

From Figure 8 which shows the receiving and sending rates without adaptation, it is obvious that as the channel SINR changes, the throughput to user 1 varies. However, the sender

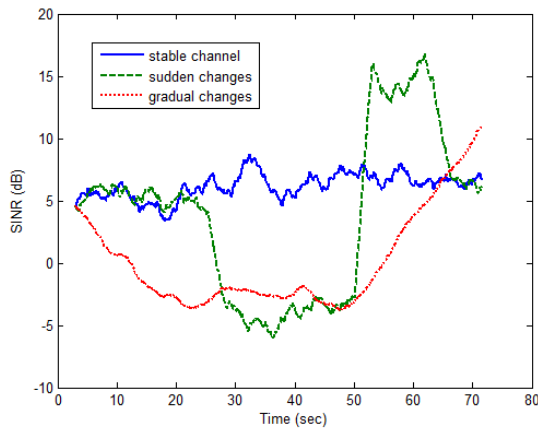
does not adapt its sending rate to the varying throughput situation but keeps the sending rate higher than what the channel can provide most of the time. This results in large delay and losses from buffer overflow at the base station. Figure 12 shows that once the delay accumulates at the base station's buffer to approximately 5 seconds, the base station starts to drop packets away. The delay afterward also remains high implying packets were continuously dropped.

Figure 9 shows the accumulated bytes received at user 1, number of bytes in the successfully reconstructed stream and the playback curve in the same scenario without adaptation. As the losses occur, the client has to wait for retransmitted packets and cannot further reconstruct the stream. This can be seen from the flat parts of the reconstructed stream curve which results in playback interruptions.

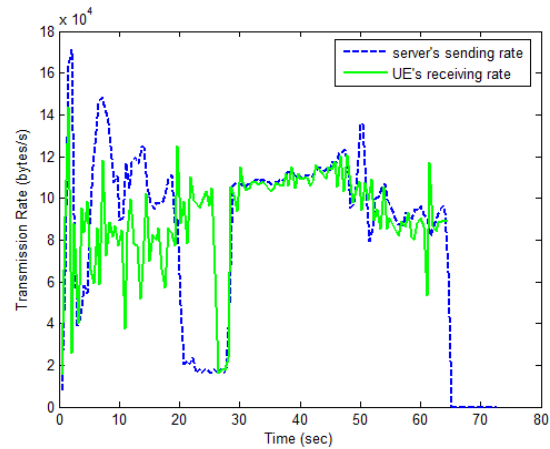
In addition, this also demands that the buffer size at the user has to be large to store packets that are not able to be playback while the user is waiting for the retransmission of the missing packets.



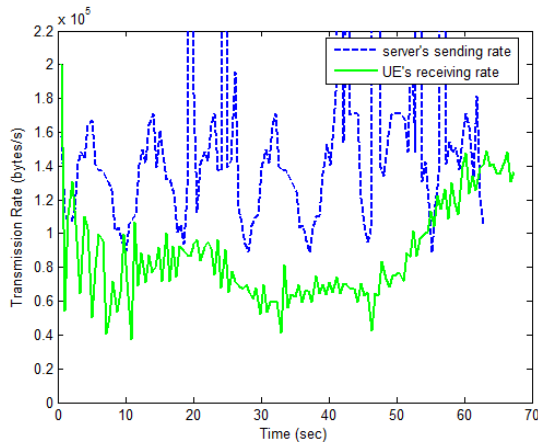
**Figure 9. Total bytes received, reconstructed stream and the playback curve at user 1 for the 3rd scenario and no adaptation**



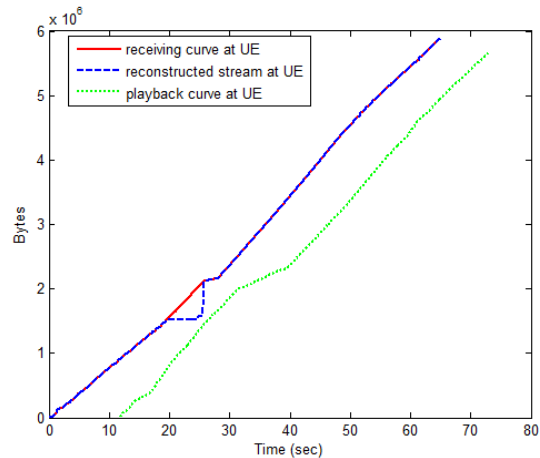
**Figure 7. Examples of channel SINR for all 3 scenarios**



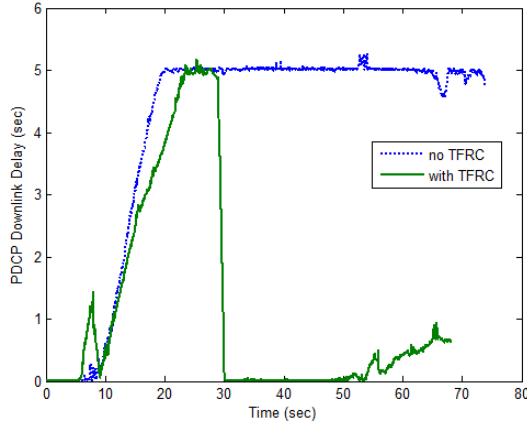
**Figure 10. Sending rate at the server and receiving rate at user 1 in the 3rd scenario with adaptation**



**Figure 8. Server's sending rate and user 1's receiving rate for the 3rd scenario and no adaptation**



**Figure 11. Accumulated bytes received at user 1, reconstructed stream and the playback curve in the 3rd scenario and with adaptation**



**Figure 12. End-to-end delay with and without the adaptation**

When the adaptation is performed, the sending rate from the server adapts to the varying throughput as shown in Figure 10. The receiving curve, the reconstructed stream and the playback curve are presented in Figure 11 which shows no playback interruption at all. In Figure 10 while the server is in the Slow Start phase from the beginning to the 20th second, the sending rate does not match well to the available throughput since the TFRC is trying to increase the rate until it encounters the first loss. Once the delay in Figure 12 reaches the 5-second limit resulting in losses from buffer overflow at the base station, the TFRC enters the Congestion Avoidance phase resulting in a better match between sending rate and available throughput afterward.

In general, the TFRC reduces the losses, delay, buffer requirement both at the user and at the base station and average interruption time significantly as seen in Table 1. The PSNR and SSIM [10] of the adapted streams are slightly lower than the original stream in scenario 1 due to removal of enhancement layers. However, being able to play the lower quality stream would be more desirable to the user than to have frequent interruptions. For no TFRC cases, the received streams are corrupted with losses such that sometimes the JSVM decoder can only decode the video partially and might repeat the frames to conceal errors or simply crash. Therefore, PSNR and SSIM were not calculated for such cases.

**Table 1. Simulation results for all scenarios**

KPI	Scenario 1	Scenario 2		Scenario 3	
		No TFRC	With TFRC	No TFRC	With TFRC
Throughput (Mbps)	1.00	0.94	0.90	0.78	0.78
Loss (%)	0.00	1.46	0.06	28.15	0.60
Delay (sec)	0.24	2.65	0.49	4.27	1.45
Interruption (sec)	0.00	11.24	1.85	15.83	0.99
Buffer size (Bytes)	715.94	852.91	681.30	871.61	706.64
PSNR (dB)	37.44	n.a.	35.61	n.a.	34.88
SSIM	0.882	n.a.	0.842	n.a.	0.820

## 6. CONCLUSION

In this work, we have shown that video streaming in mobile networks can greatly benefit from using the adaptation capability of the SVC codec in conjunction with TFRC. The bit rate of the stream can be dynamically adapted to the changing channel conditions which greatly improves all performance indicators such as interruption time, loss rate, delay and buffer requirements. This also implies that more users could be admitted to the cell and it would still be able to guarantee certain service qualities. This is especially true in loaded situation where there are not enough radio resources to combat bad reception quality in order to maintain guaranteed throughput to some users.

However, since the TFRC was not designed for a mobile environment, we expect that it can be further optimized. For example, there is no need for the server to be “friendly” to other TCP flows as the MAC scheduler would take care of the fairness in resource assignments to different flows already. We also plan to investigate the possibility of using cross-layer information from the several users to improve the adaptation algorithm and cover the joint adaptation case in the future.

## 7. ACKNOWLEDGEMENT

This work has been support by the European Commission under the contract number FP7-ICT-214063, project SEA.

## 8. REFERENCES

- [1] T. Schierl, T. Stockhammer, and T. Wiegand, “Mobile Video Transmission Using Scalable Video Coding”, IEEE Transaction on Curcuits and Systems for Video Technology, vol. 17, pp. 1204-1217, September 2007.
- [2] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard”, IEEE Transaction on Curcuits and Systems for Video Technology, vol. 17, pp. 1103-1120, September 2007.
- [3] M. Handley, S. Floyd, J. Padhye, and J. Widmer, “TCP Friendly Rate Control (TFRC): Protocol Specification”, IETF, 2003, RFC 3448.
- [4] B. Görkemli, M. O. Sunay, and A. M. Tekalp, “Video Streaming over Wireless DCCP”, IEEE ICIP 2008, San Diego, U.S.A., Oct. 2008.
- [5] M. Chen, and A. Zakhor, “Rate Control for Streaming Video over Wireless”, IEEE WIREL COMMUN, Aug 2005.
- [6] Y. Fu, R. Hu, G. Tian, and Z. Wang, “TCP-Friendly Rate Control for Streaming Service over 3G Network”, WiCOM 2006, pp. 1-4, Sept. 2006.
- [7] E. Kohler, M. Handley, and S. Floyd, “Datagram Congestion Control Protocol (DCCP)”, IETF, 2006, RFC 4340.
- [8] Nomor Research, “LTE HSPA WiMAX Application Tester”, <http://www.nomor.de/home/solutions-and-products/products/application-tester>
- [9] Fraunhofer HHI, “SVC Reference Software (JSVM Software)”, [http://ip.hhi.de/imagecom\\_G1/savce/downloads](http://ip.hhi.de/imagecom_G1/savce/downloads)
- [10] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity”, IEEE T IMAGE PROCESS, vol. 13, no. 4, pp. 600-612, Apr. 2004.