Equation Based Congestion Control for Video Transmission over WCDMA Networks

Antonios Alexiou, Dimitrios Antonellis and Christos Bouras
Research Academic Computer Technology Institute, Greece and
Computer Engineering and Informatics Dept., Univ. of Patras, Greece
alexiua@cti.gr, antonel@ceid.upatras.gr, bouras@cti.gr

Abstract

The scheme of real time streaming video is one of the newcomers in wireless data communication, raising a number of new requirements in both telecommunication and data communication systems. This scheme is applied when the user experiences real time multimedia content. Rate control is an important issue in video streaming applications for both wired and wireless networks. A widely accepted rate control method in wired networks is equation based rate control. In this work, we focus on a mechanism for equation based congestion control for video transmission over WCDMA networks. With our mechanism, the sender explicitly adjusts its sending rate as a function of the measured rate of loss events, the round trip time and the packet size. Furthermore, we examine the performance of UMTS air interface for real time video transmission using real time protocols, through a number of experiments.

1. Introduction

As communications technology is being developed, user demand for multimedia services raises. Meanwhile, the Internet has enjoyed tremendous growth in recent years. Consequently, there is a great interest in using the IP-based networks for providing multimedia services. One of the most important areas in which the issues are being debated, is the development of standards for the Universal Mobile Telecommunications System (UMTS) [1].

Rate control is an important issue in both wired and wireless streaming applications. A widely popular rate control scheme over wired networks is the equation based rate control [2], [3], also known as TCP Friendly Rate Control (TFRC). There are basically three advantages for rate control using TFRC: first, it does not cause network instability, which means that congestion collapse is avoided. Second, it is fair to TCP flows which are the dominant source of traffic on the Internet. Third, the TFRC's rate fluctuation is lower than the TCP's rate fluctuation, making it more appropriate for streaming applications which require constant video quality [4].

An in depth overview of streaming video over variable bit rate wireless channels is presented in [8]. In [4], the authors propose a widely accepted rate control method in wired networks, which is the equation based rate control also known as TFRC. In this approach the authors use multiple TFRC connections as an end-to-end rate control solution for wireless streaming video. Another way to achieve rate control for streaming over wireless is by inserting a TFRC-aware Snoop-like module, similar to [9], into the network to do local retransmissions when packets are corrupted by wireless channel errors.

In this work, we focus on solutions for streaming video over the UMTS transport channels, which only require insignificant modifications in the streaming server, but provide a certain guaranteed Quality of Service (QoS). By using the TFRC mechanism, we monitor the network state of the UMTS and estimate the appropriate transmission rate of the video data. In parallel, we control the packet losses of the transmitted video sequence, improving the representation quality of the transmitted video in the terminal of the mobile client. Furthermore, the performance of UMTS Dedicated Channels (DCHs) for real time video transmission are examined.

This paper is structured as follows. In section 2 we provide an overview of the TFRC protocol. In Section 3, an analysis of the TFRC mechanism for UMTS is presented. Following this, Section 4 reviews the operation of the TFRC mechanism and presents how the critical parameters of the TFRC mechanism are estimated. Section 5 is dedicated to the experiments' results. Finally, some concluding remarks and planned next steps are briefly described.

Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)
1550-445X/06 $20.00 © 2006 IEEE

2. The TCP Friendly Rate Control protocol

This section presents an overview of the TFRC protocol. TFRC is not actually a fully specified end-to-end transmission protocol, but a congestion control mechanism that is designed to operate fairly with TCP traffic [2]. The main idea behind TFRC is to provide a smooth transmission rate for streaming applications. TFRC has also been designed to behave reasonably fairly when competing for the available bandwidth with concurrent TCP flows, that comprise the majority of flows in today's networks.

A widely popular model for TFRC is described by the following equation [3]:

$$T = \frac{kS}{rtt\sqrt{p}} \tag{1}$$

T represents the sending rate, S is the packet size, rtt is the round trip time, p is the end-to-end packet loss rate, and k is a constant factor between 0.7 and 1.3 [6], depending on the particular derivation of Eqn. (1).

The equation describes TFRC's sending rate as a function of the measured packet loss rate, the round-trip time and the used packet size. More specifically, a potential congestion in the nodes of the path will cause an increment in the packet loss rate and in the round trip time according to the current packet size. Understanding this fluctuation, it is easy to determine the new transmission rate so as to avoid congestion and packet losses. Generally, TFRC's congestion control consists of the following mechanisms:

1. The receiver measures the packet loss event rate and feeds this information back to the sender.
2. The sender uses these feedback messages to calculate the round-trip time (RTT) of the packets.
3. The loss event rate and the RTT are then fed into the TRFC rate calculation equation in order to find out the correct data sending rate.

3. Analysis of TFRC mechanism for UMTS

Figure 1 illustrates the architecture of the packet switched domain of UMTS. The UMTS network consists of the User Equipment (UE), the UMTS Terrestrial Radio Access Network (UTRAN) and the Core Network (CN). UTRAN consists of Node B and Radio Network Controller (RNC). The Core Network is comprised from two basic nodes: Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). GGSN provides internetworking with external packet switched networks such as IP networks via the Gi interface. SGSN is connected to RNC via the IuPS interface. UE is connected to UTRAN over the UMTS radio interface Uu [13].

The typical scenario for streaming video over UMTS is shown in Figure 1, where the server is denoted by Node1 and the receiver by UE1. The addressed scenario comprises a UMTS radio cell covered by a Node B connected to an RNC. The simulation model consists of a UE connected to a DCH as it is shown in Figure 1. The model is based on the UMTS system architecture. In this simulation, we use a DCH to transmit packet data. DCH is a bi-directional channel and is reserved only for a single user. The common channels are the Forward Access Channel (FACH) in the downlink and the Random Access Channel (RACH) in the uplink.

The wireless link is assumed to have downlink bit rate B_W, and packet loss rate p_w, caused by wireless channel error. This implies that the maximum throughput that could be achieved in the wireless link is $B_W (1 - p_W)$. There can also be packet loss caused by congestion at wired nodes denoted by $p_{node\ name}$ (node name: GGSN, SGSN, RNC, Node B). The end-to-end packet loss rate observed by the receiver is denoted as p. The streaming rate is denoted by T. This means that the streaming throughput is $T (1 - p)$. Under the above assumptions we characterize the wireless channel as underutilized if $T (1 - p) < B_W (1 - p_W)$.

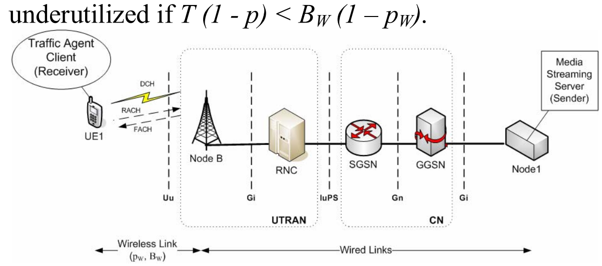

Figure 1. Streaming video over UMTS

Given the above scenario we assume the following:

1. The wireless link is a long-term bottleneck.
2. There is no congestion at Node B due to the streaming application if and only if the wireless bandwidth is underutilized, i.e. $T (1 - p) < B_W (1 - p_W)$. This also implies that no queuing delay is caused at node B. Thus, the round trip time for a given route has the minimum value. Additionally, this assumption can be restated as follows: for a given route, $rtt = RTTmin$ if and only if $T (1 - p) \le B_W (1 - p_W)$. This in turn implies that if $T (1 - p) > B_W (1 - p_W)$ then $rtt \ge RTTmin$.
3. The packet loss rate caused by wireless channel error (p_W) is random and varies from 0 to 0.16.
4. The backward route is error-free.

We use the TFRC model described in Eqn (1) to analyze the problem. When the streaming rate is T and the overall packet loss rate is p, the average throughput measured at the receiver is $T (1 - p)$. According to [4], the end-to-end packet loss rate p is a combination of p_W

and $p_{node\ name}$ (node name: GGSN, SGSN, RNC, NODEB) and is calculated as follows:

$$\begin{aligned} p &= p_{GGSN} + (1-p_{GGSN})\,p_{SGSN} + (1-p_{GGSN})(1-p_{SGSN})\,p_{RNC} + \\ &+ (1-p_{GGSN})(1-p_{SGSN})(1-p_{RNC})\,p_{NODEB} + \\ &+ (1-p_{GGSN})(1-p_{SGSN})(1-p_{RNC})(1-p_{NODEB})\,p_W \end{aligned} \quad (2)$$

Let $p^{(1)}_{node\ name}$ and $p^{(2)}_{node\ name}$ be the packet loss rate at the specific node caused by streaming traffic itself, i.e. self congestion, and by other traffic flows, i.e. cross congestion, respectively. Thus, $p_{node\ name} = p^{(1)}_{node\ name} + p^{(2)}_{node\ name}$. Since the wireless link is assumed to be the long-term bottleneck, we can estimate that there is no self congestion at the wired subsystem of the network (Assumption 1) and p can be rewritten as:

$$\begin{aligned} p &= p^{(2)}_{GGSN} + \left(1-p^{(2)}_{GGSN}\right)p^{(2)}_{SGSN} + \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)p^{(2)}_{RNC} + \\ &+ \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)\left(1-p^{(2)}_{RNC}\right)p^{(2)}_{NODEB} + \\ &+ \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)\left(1-p^{(2)}_{RNC}\right)\left(1-p^{(2)}_{NODEB}\right)p_W \\ &+ \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)\left(1-p^{(2)}_{RNC}\right)p^{(1)}_{NODEB} - \\ &+ \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)\left(1-p^{(2)}_{RNC}\right)p^{(1)}_{NODEB}\,p_W \Leftrightarrow p = p_1 + p_2 \end{aligned} \quad (3)$$

where:

$$\begin{aligned} p_1 &= p^{(2)}_{GGSN} + \left(1-p^{(2)}_{GGSN}\right)p^{(2)}_{SGSN} + \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)p^{(2)}_{RNC} + \\ &+ \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)\left(1-p^{(2)}_{RNC}\right)p^{(2)}_{NODEB} \\ &+ \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)\left(1-p^{(2)}_{RNC}\right)\left(1-p^{(2)}_{NODEB}\right)p_W, \end{aligned} \quad (4)$$

$$p_2 = \left(1-p^{(2)}_{GGSN}\right)\left(1-p^{(2)}_{SGSN}\right)\left(1-p^{(2)}_{RNC}\right)(1-p_W)\,p^{(1)}_{NODEB} \quad (5)$$

p_1 is independent of packet loss caused by streaming traffic itself and it is also independent of the streaming rate T. Furthermore, p_1 combines congestion due to non streaming flows and wireless channel error in one quantity. Therefore, it can be interpreted as equivalent wireless channel packet loss rate with no congestion due to other traffic flows on the wired part of the UMTS network. On the other hand, p_2 depends on packet loss due to self congestion, i.e. $p^{(1)}_{NODEB}$. Thus, it may vary according to the streaming rate. Eqn. (3) shows that p_1 is a lower bound for p, and that the bound is reached if and only if there is no congestion due to the streaming traffic, i.e. $p^{(1)}_{NODEB} = 0$, and hence, $p_2 = 0$. By combining Eqn. (1) and (3), an upper bound T_b, on the streaming rate of one TFRC connection can be derived as follows:

$$T \le \frac{kS}{RTT_{\min}\sqrt{p_1}} \equiv T_b \quad (6)$$

If there is no congestion due to the streaming traffic, i.e. $p^{(1)}_{NODEB} = 0$, and hence, no queuing delay caused by the streaming traffic, we get $rtt = RTTmin$, $p_2 = 0$, $p = p_1$, which entails $T = T_b$ in Eqn.(6). In this case, the throughput is $T_b\ (1 - p_1)$, which is the upper bound of throughput using one TFRC connection for the scenario shown in Figure 1. In [4] the authors define the wireless link to be under-utilized if the overall end-to-end throughput is less than $B_W\ (1 - p_W)$. Consequently, the sufficient and necessary condition for one TFRC connection to under-utilize the wireless link is:

$$T_b(1-p_1) < B_W(1-p_W) \quad (7)$$

4. Operation of UMTS TFRC mechanism

The communication between the sender and the receiver is based on RTP/RTCP sessions and the sender, denoted by Node 1 (Figure 1), uses the RTP protocol to transmit the video stream. The client, denoted by UE1 (Figure 1), uses the RTCP protocol in order to exchange control messages.

The mobile user in recurrent time space sends RTCP reports to the media server. These reports contain information about the current conditions of the wireless link during the transmission of the multimedia data between the server and the mobile user. The feedback information contains the following parameters:

- Packet loss rate. The receiver calculates the packet loss rate during the reception of sender data, based on RTP packets sequence numbers.
- Timestamp of every packet arrived at the mobile user. This parameter is used by the server for the RTT calculation of every packet.

The media server uses the feedback information and estimates the appropriate rate of the streaming video so as to avoid network congestion. The appropriate transmission rate of the video sequence is calculated from Eqn (6) and the media server is responsible for adjusting the sending rate with the calculated value. Obviously, the media server does not have the opportunity to transmit the video in all the calculated sending rates. However, it provides a small variety of them and has to approximate the calculated value choosing the sending rate from the provided transmission rates.

More specifically, the sender does not have to change the transmission rate every time it calculates a new one with a slight difference from the previous value. Consequently, it changes the transmission rate of the multimedia data only if the calculated value is differentiated a lot from the previous rate. In this approach, the number of the changes in the sending rate is restricted and the mobile user does not deal with a continually different transmission rate.

In order to implement the above assumptions, it is essential to keep a history of the previous calculated values for the transmission rate. Having this information, the media server can estimate the smoothed transmission rate, using the m most recent values of the calculated sending rate from the Eqn (8).

$$T^{Smoothed} = \sum_{i=1}^{m} w_i \cdot T^{Smoothed}_{m+1-i} \Big/ \sum_{i=1}^{m} w_i \quad (8)$$

The weights w_i are appropriately chosen so that the most recent calculated sending rates receive the same high weights, while the weights gradually decrease to 0 for the older calculated values. In our simulations we use m = 8 and the following values for the weights w_i: {1, 1, 1, 1, 0.8, 0.6, 0.4, 0.2}.

4.1. Packet loss rate estimation

The mobile user measures the packet loss rate p_1 based on RTP packets sequence numbers. The packet loss rate that is computed in the client is in accordance with the Eqn (3). This information is sent to the media server via an RTCP report. In order to prevent a single spurious packet loss, having an excessive effect on the packet loss estimation, the server smoothes the values of packet loss rate using the filter of the Eqn (9), which computes the weighted average of the m most recent loss rate values [7].

$$p_1^{Smoothed} = \sum_{i=1}^{m} w_i \cdot p_{1,m+1-i}^{Smoothed} / \sum_{i=1}^{m} w_i \qquad (9)$$

The value of $p_1^{Smoothed}$ is then used by Eqn (6) for the estimation of the transmission rate of the multimedia data. The weights w_i are chosen as in the transmission rate estimation, which means that only the eight most recent instances are kept and they take the following values {1, 1, 1, 1, 0.8, 0.6, 0.4, 0.2}.

4.2. Round trip time estimation

When the client receives an RTP packet from the server, it uses the following algorithm in order to estimate the RTT. If we assume that the media server and the mobile client have synchronized clocks, the client can use the timestamp of the RTP packet ($t_{timestamp}$) and the local time that receives the packet (t_{client}) so as to estimate the one way delay (T_{oneway}) between the server and the client:

$$t_{oneway} = t_{client} - t_{timestamp} \qquad (10)$$

If the path between the sender and the receiver was symmetric and the path had the same delay into both directions, the RTT between the sender and the receiver would be twice the value of the t_{oneway}:

$$RTT = 2 \cdot t_{oneway} \qquad (11)$$

Until now, we have made two assumptions: (1) the sender and the receiver have synchronized clocks and (2) the path between the sender and the receiver is symmetric. In order to have accurate RTT estimations we ought to take the above assumptions into account. For this reason, we use a parameter α and we can write the Eqn (11) as:

$$R\hat{T}T = (1+\alpha) t_{oneway} \qquad (12)$$

Parameter α is used in order to smooth the estimation of the RTT due to the potential unsynchronized clocks between the server and the client and due to the potential asymmetry of the path between the sender and the receiver.

In order to estimate the value of parameter α, the client needs an effective estimation of RTT, which can be acquired with the use of RTCP reports as follows: the server sends an RTCP report to the client containing information about the timestamp of the RTCP packet generated at the server. When the client receives the RTCP packet from the server, it sends an RTCP report, indicating the timestamp of the last received sender report (t_{LSR}) and the delay between receiving the last sender report and sending the receiver report (t_{DLSR}). As a result of the above, the server can make an effective RTT measurement for the path between it and the client by using the following equation (where A is the time when the server receives the client report).

$$R\hat{T}T = A - t_{DLSR} - t_{LSR} \qquad (13)$$

The media server estimates an appropriate value for the RTT every time it receives a receiver report from the client and includes this RTT measurement in the next RTP packet. Since, the client has received an effective RTT measurement from the server, it estimates an appropriate value for the parameter α using the following equation:

$$\alpha = \frac{R\hat{T}T}{t_{oneway}} - 1 \qquad (14)$$

5. Experiments

To validate the above presented analysis we carry out experiments using the NS-2 simulator [10]. The topology for the NS-2 simulations is the one shown in Figure 1. In the simulations, we use a 384Kbit-DCH in the downlink and a 128Kbit-DCH in the uplink direction. The parameter k is set to $1.5 \cdot \sqrt{2/3}$ [3].

For the transmission of the video data we use RTP. The feedback information is sent via RTCP. RTCP uses sender reports and receiver reports, which contain useful statistical information like the total transmitted packets, the packet loss rate and the delay jitter during the transmission of the data. Additionally, we use three TCP flows that have variable sending rate so as to observe the response of our mechanism. More specifically, the overall sending rate of the TCP flows has initially a low value (10 kbps). Then, it increases gradually until it reaches the maximum value of 100 kbps in the middle of the simulation. Finally, we adjust the sending rate of the TCP flows and the overall sending rate so as to decrease gradually until the value of 10 kbps. The four flows (the 3 TCP flows and the video) coexist in the same transport channel and the duration of the experiments is 200 seconds.

As far as the video sequence is concerned, it is encoded to ITU-H.263 [12]. The video traces we use, are taken from [11] and we consider that the media server can provide the video to the mobile user in three available bit rates - 64, 128 and 256 kbps.

In our experiments we consider that the media server initially transmits the video with 256 kbps bit rate. During the simulation, as we described above, we change the transmission rate of the TCP flows. When the overall sending rate is increased, we observe increased packet losses due to congestion. Measuring this packet loss rate, we can estimate the congestion and adjust the transmission rate of the video.

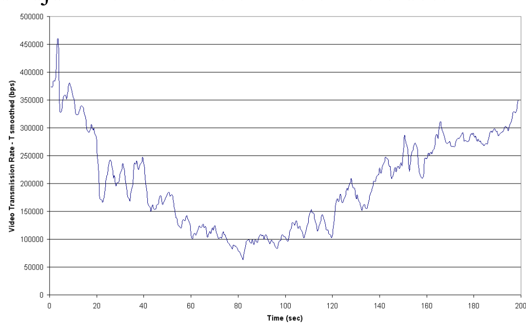

Figure 2. Calculated video transmission rate

Figure 2 depicts the estimated transmission rate of the video sequence and it is calculated according to Eqn (6). The y-axis presents the estimated transmission rate, while the x-axis represents time. According to the TFRC mechanism, the media server estimates the new transmission rates every time that the path profile changes, so as to overcome the variations in the path loss rate, as well as to serve efficiently the TCP flows. This means that when the media server transmits the video with the greater bit rate and observes an increase in the packet loss rate, it has to decrease the sending rate of the video sequence in order to 1) avoid network collapse, 2) decrease the packet loss rate and 3) continue serving the TCP flows. This explains the initial decrement in the calculated transmission rate of the video that is depicted in Figure 2.

In the approximately half of the simulation time, we observe the maximum packet loss rate due to the increased overall sending rate of the TCP flows, which results to the minimum estimated transmission rate (Figure 2). Additionally, 100 seconds after the beginning of the simulation, the smoothed transmission rate of the video increases. This is explained by the fact that the current period the sending rate of the TCP flows decreases with a respective decrement in the packet loss rate. Overall, we understand that the calculated transmission rate of the video does not perform weird transitions, but it changes smoothly and accordingly to the parameters of the network.

The throughput of the video in the wireless link is depicted in Figure 3. The y-axis presents the throughput in bps, while the x-axis represents the duration of the simulation. Additionally, a red line is used in order to be demonstrated the corresponding transmission rate of the video. As it is shown, the media server initially uses the video with bit rate 256 kbps and 20 secs after the beginning of the simulation, the TFRC mechanism calculates the smoothed transmission rate to be under the value of 256 kbps. This means that the media server has to change the bit rate of the video to 128 kbps in order to avoid congestion problems and maintain a TCP-friendly behavior. Consequently, the throughput of the video in the wireless link the specific interval is around the value of 128 kbps. The rest transitions among the provided bit rates of video are obvious and result from the corresponding values of the smoothed transmission rate. As we observe in Figure 3, there are multiple transitions between the transmission rate of the video in the regions A and B. This occurs because the calculated smoothed sending rate is differentiated very little from the two specific values and the media server every time refreshes the appropriate transmission rate.

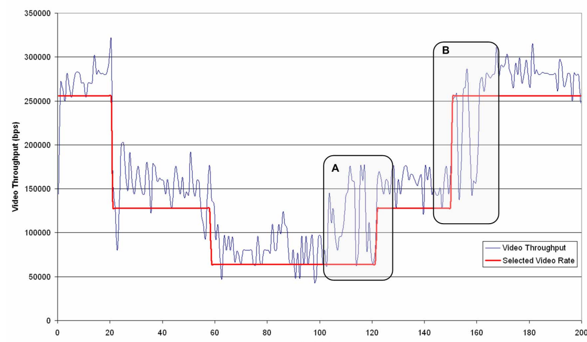

Figure 3. Video throughput in wireless link

The next interesting parameter that presents the functionality of the TFRC mechanism is the end-to-end delay of the TCP packets depicted in Figure 4. More specifically, we observe that when the media server transmits the video with 256 kbps bit rate, the end-to-end delay of the TCP packets is increased and our mechanism has to restrict this. Thus, as we described above, the media server decreases the sending rate of the video to 128 kbps and the end-to-end delay of the TCP packets reduces respectively. According to Figure 4, the end-to-end delay of the TCP packets has a low value in the interval between 20 secs and 150 secs, except the period that the transmission rate of the TCP flows is increased and we observe congestion in the path. This occurs because a significant packet loss rate results to a greater number of retransmitted TCP packets which in turn entails an increased traffic in the network. Additionally, when the media server uses the 64 kbps bit rate to transmit the video to the mobile user and we observe that the packet loss rate decreases, our mechanism decides to increase the transmission rate of the video so as to increase the total throughput.

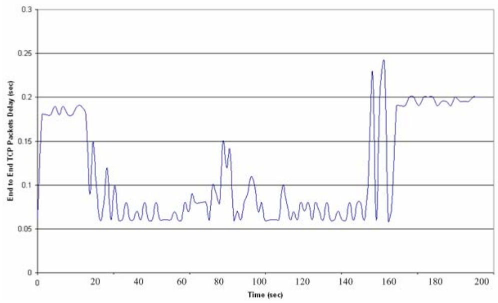

Figure 4. End-to-end delay of the TCP packets

The above functionality of the TFRC mechanism is also proved by the calculation of the delay jitter of the TCP packets depicted in Figure 5. In other words, from the time that the media server reduces the sending rate of the video until it decides to increase it to the value of 256 kbps, the delay of the TCP packets is smooth (Figure 5) and restricted (Figure 4). This means that the mobile user does not deal with different delays in the TCP flows and it is able to make more accurate estimations of the TCP packets' arrival time.

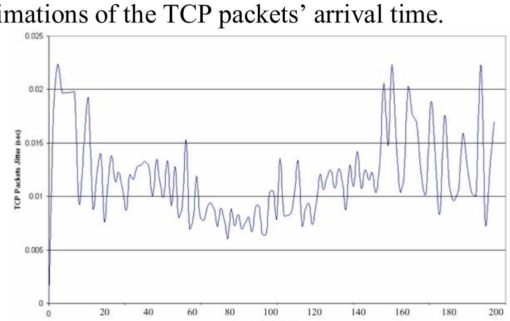

Figure 5. Delay jitter of the TCP packets

Finally, it has to be mentioned that in our experiments the packet delay of the RTCP reports that the mobile client sends to the media server has an average value of 61 msecs.

6. Conclusions and future work

In this paper we have presented an analysis of the TCP Friendly Rate Control mechanism for UMTS. Through a number of experiments we have concluded that the TFRC mechanism performs efficiently in mixed traffic conditions. The three goals of our rate control could be stated as follows. First, it does not cause any network instability, i.e. congestion collapse. Second, TFRC is assumed to be TCP Friendly and third, it leads to optimal network performance.

The step that follows this work is to evaluate the performance of adaptive video transmission using the High Speed Downlink Packet Access (HSDPA) technology. Furthermore, among the envisioned steps is the integration of a multicast mechanism in the system, which could result to bandwidth savings improving the overall performance of the UMTS network.

7. References

[1] H. Holma, and A. Toskala, "WCDMA for UMTS: Radio Access for Third Generation Mobile Communications", *John Wiley & Sons*, 2003.

[2] S. Floyd, M. Handley, J. Padhye and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", *Proc. ACM SIGCOMM 2000*, Aug. 2000, pp. 43 – 56.

[3] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", *IEEE/ACM Transactions on Networking*, Aug.1999.

[4] M.Chen and A. Zachor, "Rate Control for Streaming Video over Wireless", *Proc. IEEE INFOCOM 2004*.

[5] D. Wong and V. Varma, "Supporting Real-Time IP Multicast Services in UMTS", *IEEE Communications Magazine*, November 2003, pp. 148-155.

[6] J. Mahdavi and S. Floyd, "TCP-Friendly unicast rate-based flow control", Technical note sent to end2end-interest mailing list, http://www.psc.edu/networking/papers/tcp friendly.html, Jan. 1997.

[7] L. Vicisiano, L. Rizzo and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer" , *IEEE INFOCOM*, March 1998, pp. 996–1003.

[8] T. Stockhammer, H. Jenkac and G. Kuhn, "Streaming Video Over Variable Bit-Rate Wireless Channels", *IEEE TRANSACTIONS ON MULTIMEDIA*, VOL. 6, NO. 2, APRIL 2004, pp. 268-277.

[9] H. Balakrishnan, V. Padmanabhan, S. Seshan, R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links", *ACM SIGCOMM'96*, pp. 256 – 269.

[10] The NS-2 Simulator. Available at http://www.isi.edu/nsnam/ns.

[11] F. Fitzek, and M. Reisslein, "MPEG-4 and H.263 Video Traces for Network Performance Evaluation", *IEEE Network*, Nov. - Dec. 2001, pp. 40-54.

[12] C. Iskander and P. Mathiopoulos, "Online Smoothing of VBR H.263 Video for the CDMA200 and IS-95B Uplinks", *IEEE Transactions on Multimedia*, Vol. 6, No. 4, August 2004, pp. 647-658.

[13] 3GPP, Network Architecture, TS 23.002, March 2000.