AN EFFICIENT MECHANISM FOR ADAPTIVE MULTI-MEDIA TRANSMISSION IN 3G NETWORKS

Antonios Alexiou, Konstantinos Barounis, Christos Bouras
Research Academic Computer Technology Institute and
Computer Engineering & Informatics, University of Patras
N. Kazantzaki str. GR 26500, Patras, Greece
alexiua@cti.gr, barounis@ceid.upatras.gr, bouras@cti.gr

ABSTRACT

This paper proposes a mechanism for the congestion control for video transmission over UMTS. Our scheme is applied when the mobile user experiences real time multimedia content and adopts the theory of a widely accepted rate control method in wired networks, namely equation based rate control. In this approach, the transmission rate of the multimedia data is determined as a function of the packet loss rate, the round trip time and the packet size and the server explicitly adjusts its sending rate as a function of these parameters. Through a number of simulations and experiments we validate the correctness and measure the performance and efficiency of the mechanism. The mechanism at first level is evaluated through simulations that carried out in the ns-2 simulator. At second level, in order to validate the proposed mechanism in real world traffic scenarios we perform some experiments in a commercial UMTS network.

KEYWORDS

UMTS, Adaptive Multimedia Transmission, Streaming Video over Wireless.

1. INTRODUCTION

As communications technology is being developed, users' 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 to provide multimedia services. An important area in which the issues are being debated, is the development of standards for the Universal Mobile Telecommunications System (UMTS) (Holma and Toskala, 2003).

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, also known as TCP Friendly Rate Control (TFRC) (Floyd et al, 2000), (Floyd and Fall, 1999). There are basically three main 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 the traffic on the Internet. In case that a TCP flow co-exists in the network with a non-TCP flow, the mechanism adjusts the sending rate of the non-TCP flow so as to enhance the performance of the TCP flows. Third, the TFRC's rate fluctuation is lower than TCP, making it more appropriate for streaming applications which require constant video quality (Chen and Zachor, 2004). This means that the client does not deal with a great variety of rates of the streaming application and it is able to make more accurate estimations of the packets' arrival time.

An overview of streaming video over variable bit rate wireless channels is presented in (Stockhammer et al, 2004). Chen and Zachor (2004) 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 (Balakrishnan et al, 1996), 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 UMTS transport channels, which only require insignificant modifications in the streaming server and client, but provide a certain guaranteed Quality of Service (QoS). This could be achieved through the use of the TFRC mechanism. In our approach, we have modified the TFRC mechanism that is mainly used in wired networks, in order to support the specific characteristics and architecture of the UMTS network. With the aid of the TFRC mechanism, we monitor the network state of the UMTS and estimate the appropriate transmission rate of the multimedia data. Through a number of simulations and experiments we validate the correctness and measure the performance and efficiency of the mechanism. The mechanism at first level is evaluated through simulations that carried out in the ns-2 simulator. At second level, in order to validate the proposed mechanism in real world traffic scenarios we perform some experiments in a commercial UMTS network.

This paper is structured as follows. In Section 2, the TFRC mechanism for the UMTS network is analyzed. Following this, Section 3 is dedicated to the simulation and experiment results. Finally, some concluding remarks and planned next steps are briefly described.

2. ANALYSIS OF THE TFRC MECHANISM FOR UMTS

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 Dedicated Channel (DCH) as it is shown in Figure 1. The model is based on the UMTS system architecture. In this analysis, we use a DCH to transmit packet data.

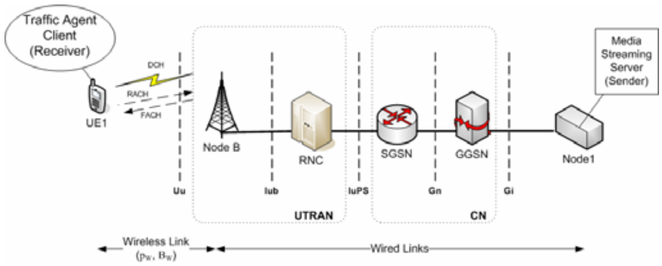


Figure 1. UMTS Architecture

The wireless link is assumed to have available bandwidth 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 could 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: $T (1 - p) < B_W (1 - p_W)$.

Given the above described scenario we assume the following:

- The wireless link is assumed to be the long-term bottleneck. This means that there is no congestion due to streaming traffic to the nodes GGSN, SGSN and RNC.
- 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, which means that the round trip time for a given route has the minimum value, i.e. RTT_{min}. Thus, this assumption can be restated as follows: for a given route $RTT = RTT_{min}$ if and only if $(1 - p) \leq B_W (1 - p_W)$. This in turn implies that if $T (1 - p) > B_W (1 - p_W)$ then $RTT \geq RTT_{min}$.
- The packet loss rate caused by wireless channel error (p_W) is random and varies from 0 to 0.16.

We use the TFRC model described in Eqn (1) to analyze the problem. TFRC is not actually a fully specified end to-end transmission protocol, but a congestion control mechanism that is designed to operate fairly along with TCP traffic. Generally TFRC should be deployed with some existing transport protocol such as UDP or RTP in order to present its useful properties (Floyd et al, 2000). A widely popular model for TFRC is described by the following equation (Floyd and Fall, 1999):

$$T \le \frac{kS}{RTT_{\min}\sqrt{p_1}} \tag{1}$$

T represents the sending rate, S is the packet size, RTT is the end-to-end round trip time, p is the end-to-end packet loss rate, and k is a constant factor between 0.7 and 1.3 (Mahdavi and Floyd, 1997), depending on the particular derivation of Eqn (1).

As it has already been mentioned, the average throughput measured at the receiver is $T(1-p)$, when the streaming rate is T and the overall packet loss rate is p. Chen and Zachor (2004) claim that the end-to-end packet loss rate p is a combination of p_W and $p_{node\ name}$ (node name: GGSN, SGSN, RNC, NODEB).

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

According to Eqn (2), the total packet loss rate is the sum of the packet loss rate in each node of the network (Figure 1). More specifically the packet loss rate in each node is the product of the packet loss rate in the specific node and the offered rate of the previous node. Furthermore, we use $p^{(1)}_{node\ name}$ and $p^{(2)}_{node\ name}$ to represent 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, $p_{node\ name} = p^{(1)}_{node\ name} + p^{(2)}_{node\ name}$. Given the fact that there is no self congestion at the wired part of the network since the wireless link is assumed to be the long-term bottleneck (Assumption 1), p can be re-written as:

$$p = p_1 + p_2 \tag{3}$$

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

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

p_1 is independent of packet loss caused by streaming traffic itself, and hence also independent of 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. $p^{(1)}_{NODEB}$ and thus may vary according to the streaming rate. Eqn. (4) 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 streaming traffic, i.e. $p^{(1)}_{NODEB} = 0$ and hence, $p_2 = 0$. By combining Eqn. (1) and (4), an upper bound on the streaming rate of one TFRC connection can be derived as follows:

$$T \le \frac{kS}{RTT_{\min}\sqrt{p_1}} \tag{6}$$

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, whereas the client, denoted by UE1 (Figure 1), uses the RTCP protocol in order to exchange control messages. In the following paragraphs, details about the different aspects of the mechanism are given.

The mobile user (client) in recurrent time space sends RTCP reports to the media server. These reports contain information about the current conditions of the network 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 using the feedback information (send by the mobile user) 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. It provides a small variety of them and has to approximate the calculated value choosing the sending rate from the provided transmission rates. This extends the functionality of the whole congestion control mechanism. 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 to one of the available sending rates of the media server as has already been mentioned. 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 (7).

$$T^{Smoothed} = \frac{\sum_{i=1}^{m} w_i \cdot T_{m+1-i}^{Smoothed}}{\sum_{i=1}^{m} w_i} \tag{7}$$

The value m used in calculating transmission rate determines TFRC's speed in responding to changes in the level of congestion (Handley et al, 2003). 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 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}. Thus, we have chosen to keep track of eight values according to (Handley et al, 2003). More details regarding the operation of the mechanism and the estimation of the parameters appeared in Eqn (6) are presented in (Alexiou, Antonellis and Bouras, 2006) and (Alexiou, Antonellis and Bouras, 2007).

3. SIMULATION AND EXPERIMENTAL VERIFICATION

3.1 Simulation Results

To validate the above presented analysis we carry out simulations using the ns-2 simulator. The topology for the simulations is the one shown in Figure 1. In the simulations, we use a 384 kbps-DCH in the downlink and a 128 kbps-DCH in the uplink direction. The parameter k is set to $1.5 \cdot \sqrt{2/3}$ (Floyd and Fall, 1999).

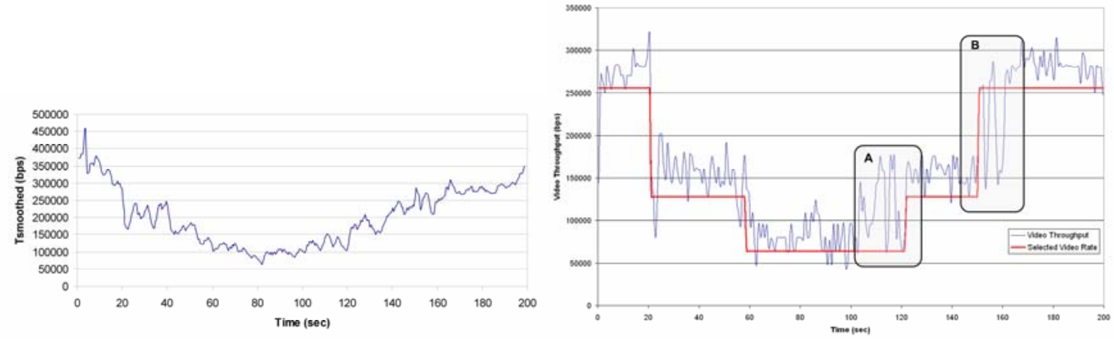
For the transmission of the video data we use RTP. The feedback information is sent via RTCP. Additionally, we use three TCP flows that have variable sending rate so as to observe the response of our mechanism. More specifically, the overall sending rate of the TCP flows has initially a low value (10 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 simulation is 200 seconds. The video sequence is encoded to ITU-H.263 (Fitzek and Reisslein, 2001). The video traces we use, are taken from (Fitzek and Reisslein, 2001) and we consider that the media server can provide the video to the mobile user in three bit rates - 64, 128 and 256 kbps. The RTCP report rate is 1 sec.

In our simulations 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 2a depicts the estimated transmission rate of the video sequence and it is calculated according to Eqn (3). 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 2a. 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 2a). 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 in the wireless link is depicted in Figure 2b. 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 2b, 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.

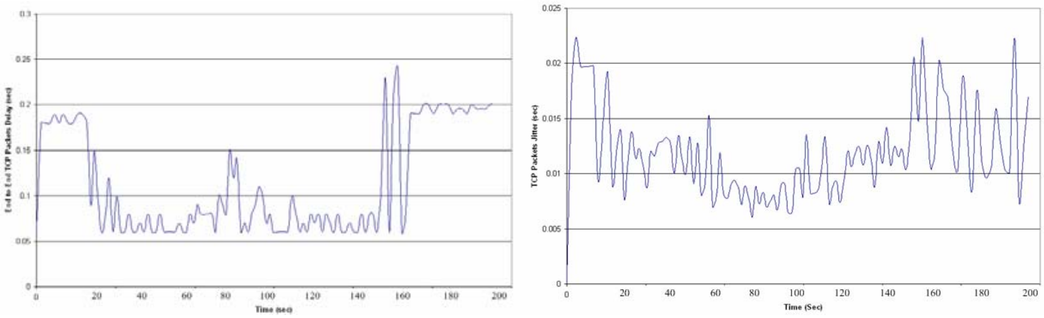


(a) (b)

Figure 2. Calculated video transmission rate (a) and video throughput in wireless link (b)

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 3a. 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 3a, 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.

The above functionality of the TFRC mechanism is also proved by the calculation of the delay jitter of the TCP packets depicted in Figure 3b. 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 3b) and restricted (Figure 3a). 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.



(a) (b)

Figure 3. End-to-end delay (a) and delay jitter (b) of the TCP packets

3.2 Experiment Results

In order to measure the performance of the TFRC mechanism in a commercial UMTS network we perform a number of experiments. The basic idea is to examine the performance of the streaming video under various network conditions. The equipment we use in the experiments are a real-time streaming server that supports RTSP (implemented on an Intel Pentium IV PC), a Release 99 3G data card and a Pentium Mobile laptop.

Regarding the real time streaming server, video content of 500 seconds duration encoded at 80 and 200 kbps is used during the streaming session while the packet size is 180 bytes. The server is programmed to examine the RTT information and the packet loss rate conveyed by RTCP packets sent by the mobile user. Then, with the help of the TFRC mechanism, an average estimation of the transmission rate is calculated. For the experiments, the parameter k is set to 0.7 and the RTCP report rate is set to 1 sec.

After the average estimation of the transmission rate, the server characterizes the network on three possible conditions:

- Condition congestion. The jitter and the round trip time have an increased value due to congestion in the network. In this case the transmission rate of the multimedia data should be reduced.
- Condition load. In this case everything works properly, as the jitter and the round trip time have regular values for a video stream. Thus, the transmission quality is satisfactory and no change happens in the video transmission rate.
- Condition unload. The jitter and the round trip time have low values the server streams the low rate video. In this case the transmission rate of the multimedia data could be increased.

In the experiment we follow the following scenario: the mobile user establishes an RTP session with the server. The streaming process starts with the high rate video (200 kbps), and the server receives the RTCP packets sent by the mobile client. In order to notice how the server adapts to the network conditions, we overload the wireless link by downloading (in the client side) a file during the streaming session. The purpose is to see how the RTT, the jitter and the suggested average transmission rate are affected. The server is responsible for making the proper switching in order to continue the streaming and to improve the network parameters affected before.

During the experiments the user uses an ftp program for the downloading of: a) a 3 MB file, from time 130 sec to 260 sec and b) a 2 MB file, from time 360 sec to 440 sec. It has to be mentioned that as the experiments went on, there was no packet loss in the users' side.

In Figure 4 we can see the RTT of the packets measured by the server. Despite some variations at the beginning of the streaming, we can see that 140 seconds after the beginning of the experiment, where the user starts downloading the first file, the RTT increases because of network congestion, until the time of 170 sec. The server in this time interval responses to this network condition with a decrease in the transmission rate (from 200 kbps to 80 kbps) of the streaming video in order the mobile user to continue served satisfactory. Until the time 260 sec the user experiences two simultaneous applications in his mobile terminal: a video streaming application and an ftp downloading. In the latter time interval the network is characterized by the server as loaded and no change happens in the transmission rate of the multimedia data.

At the time of 280 sec the value of RTT has a very low value because of the fact that the user has stopped downloading the file and the wireless link has enough available bandwidth to support the streaming application. The network is characterized by the server as unloaded and thus the server increases the rate of the multimedia data from 80 kbps to 200 kbps. The RTT does not seem to be affected by this change as there is not any congestion on the link, until the user decides to start downloading the second file at the time of 360 sec. In this time instance, RTT starts increasing again and thus the server switches to the low rate video (the network is characterized as congested by the server). Finally, the user ftp downloading stops at the second 440, and the server increases the transmission rate of the video as RTT has fallen again to regular values. At this point it has to be mentioned that the average value of the RTT is 1.69 seconds.

Figure 4. Round Trip Time

Figure 5 presents the average estimated value of the streaming rate calculated on the server. We can see that the TFRC mechanism operates efficiently during the streaming session, and its responses to the varying network conditions are immediate. To this point it has to be mentioned that the suggested average transmission rate it should not be confused with the server's transmission rate which is the rate that the server streams the video. The suggested transmission rate is an upper bound for the allowed server's transmission rate in order the server to overcome possible congestion periods in the network. In particular, as it is depicted in Figure 5, as the streaming starts, the user is able to take advantage of the whole bandwidth that the wireless link can provide him. That is why the TFRC mechanism calculates rates over 300 kbps. When the user starts downloading the first file, at the time of 130 sec, the suggested rate does not change immediately. This is because the suggested rate is an average result of previous calculated values of the average transmission 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 the next 10 seconds the suggested rate decreases continuously. Thus, the server transmits in a lower streaming rate (from 200 kbps to 80 kbps) in order to avoid congestion in the network. When the user stops downloading the first file (at time 260 sec) the rate increases until it reaches the value of 270 kbps at the time of 290 sec. Then the server, according to the suggested rate, switches to the high rate video. During the period of the second user downloading, the suggested rate has similar behavior.

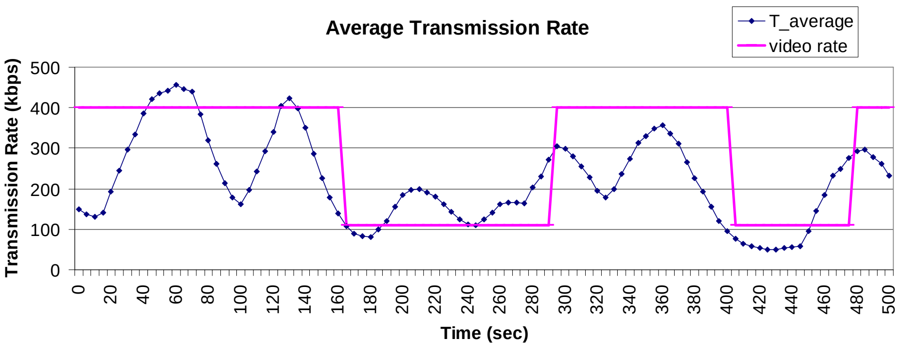


Figure 5. Suggested Average Transmission Rate

Finally, Figure 6 presents the interarrival jitter as it is calculated by the server. Obviously, the jitter is low at the beginning of the stream. During the network congestion because of the user's downloading, the jitter increases until the server reduces the sending rate of the video. The fact that the jitter is low during the downloading process means that the TFRC mechanism operates efficiently and the server is able to adapt any time to the current network conditions. As a result, the streaming application is served satisfactory and the only change that it is observed by the mobile user is the low video quality at the user's player when the network is congested. During the experiment, the average calculated jitter value is 159.9 milliseconds.

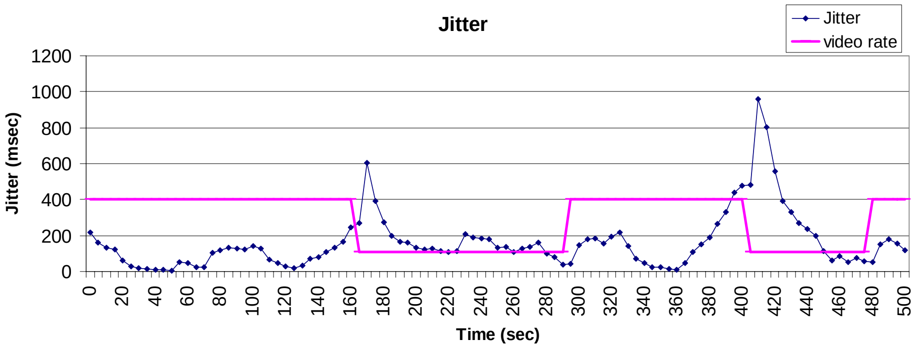


Figure 6. Jitter calculation

4. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an analysis of the TCP Friendly Rate Control mechanism for UMTS. With the aid of the TFRC mechanism, we have monitored the network state of the UMTS and estimated the appropriate transmission rate of the multimedia data. Through a number of simulations and experiments we concluded that the TFRC mechanism performs efficiently in mixed traffic conditions. The step that follows this work is to evaluate the performance of adaptive video transmission using the High Speed Downlink Packet Access (HSDPA) technology.

REFERENCES

Alexiou A. et al, 2007. Adaptive and reliable video transmission over UMTS for enhanced performance, *International Journal Of Communication Systems*, John Wiley & Sons, Vol. 20,, Issue 1, pp. 65-81.

Alexiou A. et al, 2006. Equation Based Congestion Control for Video Transmission over WCDMA Networks. *Proceedings of IEEE AINA 2006*. Vienna, Austria, pp .445-450.

Balakrishnan, H. et al, 1996. A comparison of mechanisms for improving TCP performance over wireless links, *ACM SIGCOMM 1996*. California, USA, pp. 256 – 269.

Chen, M. and Zachor, A., 2004. Rate Control for Streaming Video over Wireless. *Proceedings of IEEE INFOCOM 2004*. Hong Kong, China.

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

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

Floyd, S. et al, 2000. Equation-Based Congestion Control for Unicast Applications. *Proceedings of ACM SIGCOMM 2000*. Stockholm, Sweden, pp. 43 – 56.

Handley, M. et al, 2003. TCP Friendly Rate Control (TFRC). *RFC 3448*.

Holma, H. and Toskala, A., 2003. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. John Wiley & Sons, West Sussex, England.

Mahdavi, J. and Floyd, S., 1997. TCP-Friendly unicast rate-based flow control. Available at http://www.psc.edu/networking/papers/tcp_friendly.html.

Stockhammer, T. et al, 2004. Streaming Video Over Variable Bit-Rate Wireless Channels. *IEEE Trabsactions on Multimedia*, Vol. 6, No. 2, pp. 268-277.