INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2007; **20**:1315–1335
Published online 29 December 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/dac.867

Scalable rate control for video transmission over UMTS

Antonios Alexiou[1,2], Christos Bouras[1,2,*,†] and Vaggelis Igglesis[2]

[1] *Research Academic Computer Technology Institute, N Kazantzaki str, GR-26500 Patras, Greece*
[2] *Computer Engineering and Informatics Department, University of Patras, Greece*

SUMMARY

This paper proposes a mechanism for the congestion control for video transmission over Universal Mobile Telecommunications System (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. A first level of evaluation is carried out using the ns-2 simulator. A second level then validates the proposed mechanism in real world traffic scenarios by performing experiments in a commercial UMTS network.

Received 1 November 2005; Revised 1 September 2006; Accepted 1 October 2006

KEY WORDS: adaptive multimedia transmission; TFRC; UMTS; mobile multimedia

1. INTRODUCTION

As communications technology is being developed, user's demand for multimedia services is increasing. 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. 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 equation-based rate control [2,3], also known as TCP Friendly Rate Control (TFRC). 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

*Correspondence to: Christos Bouras, Research Academic Computer Technology Institute, N Kazantzaki str, GR-26500 Patras, Greece.
† E-mail: bouras@cti.gr

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 [4]. 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 [5]. 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 [6], 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 video data. Our mechanism, handles fairly all the data flows sharing the available channel resources without causing network congestion. In parallel, we control the packet loss of the transmitted video sequence, improving the representation quality of the transmitted video in the terminal of the mobile client. Through a number of simulations and experiments we validate the correctness and measure the performance and efficiency of the mechanism. A first level of evaluation is carried out using the ns-2 simulator. A second level then validates the proposed mechanism in real world traffic scenarios by performing experiments in a commercial UMTS network.

This paper is structured as follows. Section 2 is dedicated to the related work. In Section 3, we present an overview of UMTS in the packet switched domain while Section 4 presents an analytical computation of packet service time for video traffic over UMTS. In Section 5, the TFRC mechanism for the UMTS network is analysed. Following this, Sections 6 and 7 are dedicated to the simulation and experiment results. Finally, some concluding remarks and planned next steps are briefly described.

2. RELATED WORK

Reference [4] proposes the now 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. As [7] presents, TCP Reno, treats the occurrence of packet loss as a manifestation of network congestion. This assumption may not apply to networks with wireless channels, in which packet loss is often induced by noise, link error, or reasons other than network congestion. Equivalently, TCP Vegas uses queuing delay as a measure of congestion [8]. Thus, the authors in [7] propose an enhancement of the TCP Reno and TCP Vegas for the wireless networks, namely TCP Veno.

DOI: 10.1002/dac

In [9], the authors present two algorithms that formulate resource allocation in wireless networks. These procedures constitute a preliminary step towards a systematic approach to jointly design TCP congestion control algorithms, not only to improve performance, but more importantly, to make interaction more transparent. Xu *et al.* [10] study the performance characteristics of TCP New Reno, TCP SACK, TCP Veno and TCP Westwood under the wireless network conditions and they propose a new TCP scheme, called TCP New Jersey, which is capable of distinguishing wireless packet losses from congestion.

Recent work [11] provides an overview of MPEG-4 video transmission over wireless networks. A critical issue is how we can ensure the QoS of video-based applications to be maintained at an acceptable level. Another point to consider is the unreliability of the network, especially of the wireless channels because we observe packet losses resulting in a reduction of the video quality. The results demonstrate that the video quality can be substantially improved by preserving the high priority video data during the transmission.

Although these adaptive streaming techniques show noticeable benefits in streaming video over heterogeneous networks, they require significant modifications in the streaming client, the streaming server, or both.

3. OVERVIEW OF UMTS IN THE PACKET SWITCHED DOMAIN

A UMTS network is split into two main domains: the User Equipment (UE) domain and the Public Land Mobile Network (PLMN) domain. The UE domain consists of the equipment employed by the user to access the UMTS services. The PLMN domain consists of two land-based infrastructures: the Core Network (CN) and the UMTS Terrestrial Radio-Access Network (UTRAN) (Figure 1). The CN is responsible for switching/routing voice and data connections, while the UTRAN handles all radio-related functionalities. The CN is logically divided into two service domains: the Circuit-Switched (CS) service domain and the Packet-Switched (PS) service domain [1,12]. The PS portion of the CN in UMTS consists of two kinds of General Packet Radio

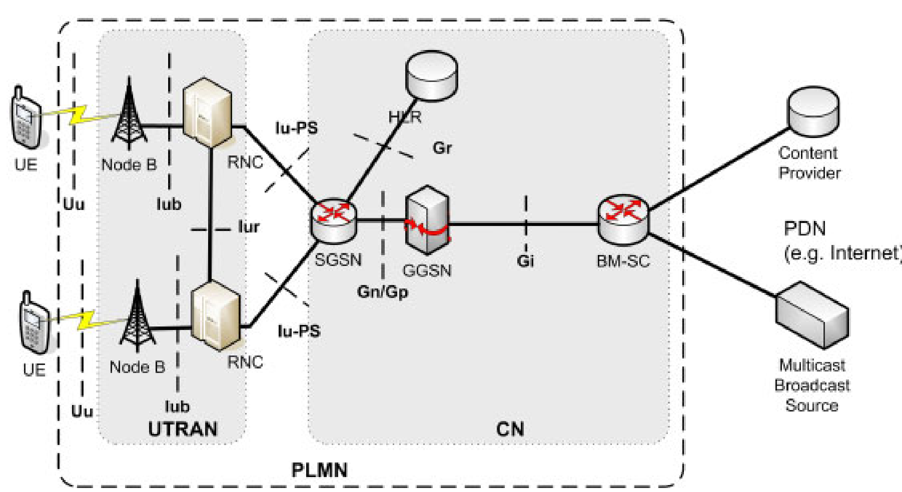

Figure 1. UMTS architecture.

DOI: 10.1002/dac

Service (GPRS) Support Nodes (GSNs), namely Gateway GSN (GGSN) and Serving GSN (SGSN) (Figure 1). SGSN is the centrepiece of the PS domain. It provides routing functionality interacts with databases (like Home Location Register (HLR)) and manages many Radio Network Controllers (RNCs). SGSN is connected to GGSN *via* the Gn interface and to RNCs *via* the Iu interface. GGSN provides the interconnection of UMTS network (through the Broadcast Multicast-Service Centre) with other Packet Data Networks (PDNs) like the Internet [1].

UTRAN consists of two kinds of nodes: the first is the RNC and the second is the Node B. Node B constitutes the base station and provides radio coverage to one or more cells (Figure 1). Node B is connected to the UE *via* the Uu interface (based on the wideband code division multiple access, W-CDMA technology) and to the RNC *via* the Iub interface. One RNC with all the connected to it Node Bs is called Radio Network Subsystem (RNS).

4. ANALYTICAL COMPUTATION OF PACKET SERVICE TIME FOR VIDEO TRAFFIC

In this section we present an analytical computation of the time required for any packet of a given video sequence to travel from the RNC to the mobile user (UE). The packets of the video sequence do not have a constant size. The size of the video packets which are being transmitted over the UMTS air interface is presented in Figure 2. The video sequence follows the ITU H.263 standard, in QCIF format (176×144 pixels) at the PAL frame rate of 25 frames per second [13]. In Figure 2, the y-axis presents the size of the packets in bytes, while the x-axis shows the packet sequence number. The average packet size is 758 bytes. The above-mentioned video traffic is also going to be used in the simulation that is presented in the following sections.

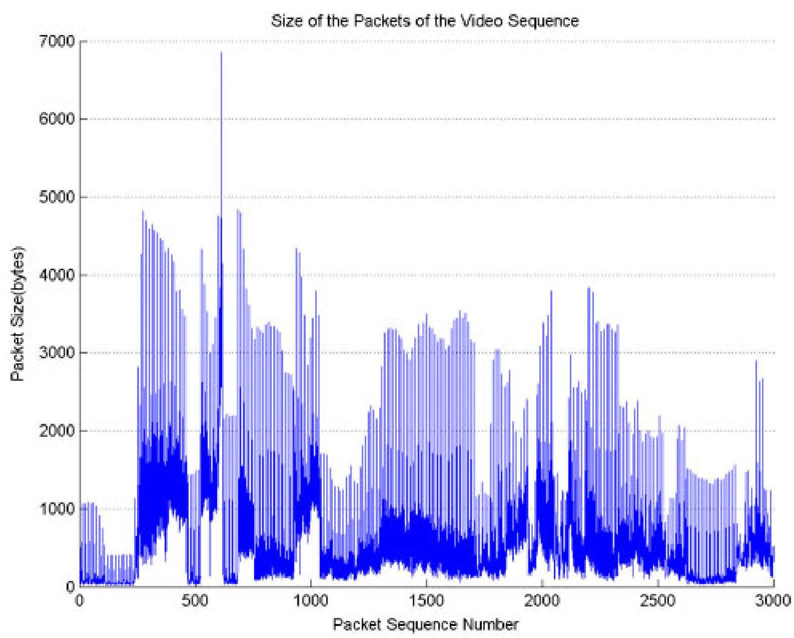

Figure 2. Video packets' size.

DOI: 10.1002/dac

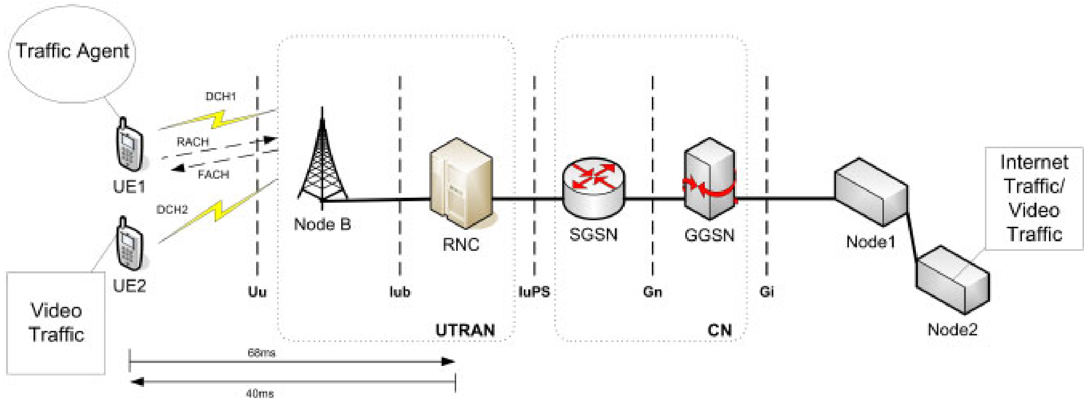

Figure 3. Typical scenario for streaming video over UMTS.

Considering a video transmission from a fixed internet node to a mobile user (Downlink) (Figure 3), as the packets leave the RNC and arrive at the Node B, they queue up in order to be broken down into smaller size packets. Every Packet Data Protocol (PDP) Service Data Unit (SDU) is segmented into multiple Radio Link Control (RLC) PDUs of fixed size. Each of these PDUs fits into a transport block in order to be transmitted over the air. The size of these transport blocks is fixed and is called Transmission Time Interval (TTI). According to [14], the size of each RLC PDU is 40 bytes.

Based on the analysis presented in [14], in Downlink, for any RLC PDU that is transmitted over the air, the RNC receives a status report of the UE 68 ms after its transmission. For the opposite direction, since generally, the uplink TTI is twice as large as in the downlink direction, the UE receives a status report 40 ms after its transmission.

In this section we determine the minimum IP packet service time for the video traffic that we use in the simulation. This time is comparable to the delay in UTRAN (time required for any packet to travel from RNC to UE). This time is calculated as follows.

The number of RLC PDUs an IP packet is segmented into is

$$N_{\mathrm{PDU}} = \frac{\mathrm{Packet\_size(bytes)}}{40\ \mathrm{bytes}} \tag{1}$$

The maximum RLC PDUs that can be transmitted within one TTI is

$$L_{\max} = 8 \tag{2}$$

Consequently, the number of TTIs required for the transmission of the packet is

$$N_{\mathrm{TTI}} = \frac{N_{\mathrm{PDU}}}{L_{\max}} \tag{3}$$

Since the TTI in downlink is 10 ms, the minimum time required for the transmission and reception of a whole packet is

$$T_{\min} = 40\ \mathrm{ms} + \lceil N_{\mathrm{TTI}} - 1 \rceil \cdot 10 \tag{4}$$

For the video traffic used in the simulation, the minimum time required for the transmission and reception of any packet of the video sequence is presented in Figure 4. The y-axis shows the packet service time in ms, while the x-axis presents the packet sequence number.

DOI: 10.1002/dac

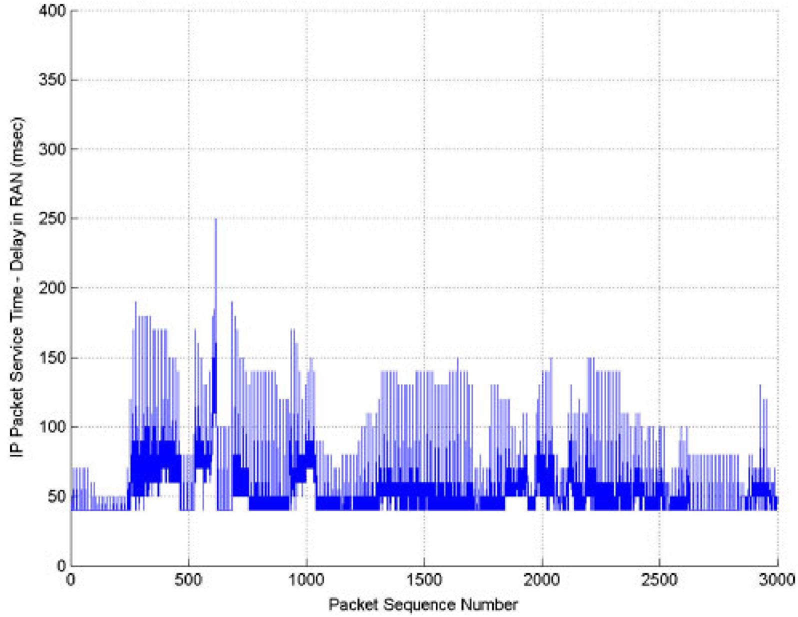

Figure 4. Time required for the transmission and reception of any packet of the video sequence.

More specifically, the packet service time represents the delay in UTRAN, which is the time required for any packet to travel from the RNC to the UE.

Furthermore, Figure 5 presents the probability density function (pdf) of the minimum time required for the transmission and the reception of any packet of the video sequence. The x-axis presents the packet service time, while the y-axis presents the pdf. Since all IP packets do not have a constant size, the minimum time varies and at $T_{\min} = 40$ ms has a maximum value pdf $= 0.5087$. This means that the 50.87% of the packets have a minimum delay in UTRAN of 40 ms approximately. The average $T_{\min}$ is 54.39 ms. This value of the average packet service time is low. This fact signifies that the majority of the packets that reach the mobile user are characterized by low delay and therefore can easily be transmitted *via* the UMTS air interface without causing any problems in the quality of the video experienced by the mobile user.

5. ANALYSIS OF THE TFRC MECHANISM FOR UMTS

The typical scenario for streaming video over UMTS is shown in Figure 3, 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 DCH as it is shown in Figure 3. In the analysis, we use the 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 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

DOI: 10.1002/dac

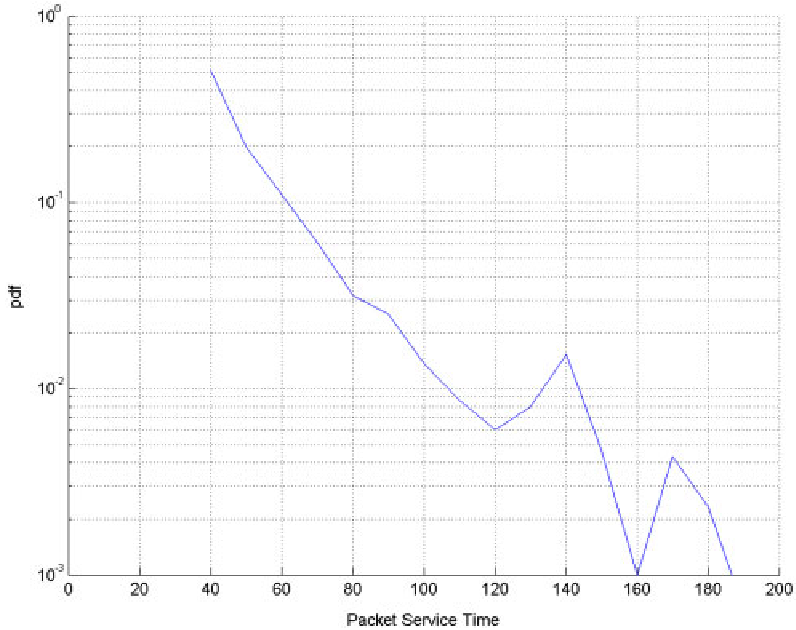

Figure 5. Pdf of the IP packet service time.

the wireless link is $B_W(1-p_W)$. There could also be packet loss caused by congestion at wired nodes denoted by $p_{\text{node name}}$ (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:

1. 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.
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, which means that the round trip time for a given route has the minimum value, i.e. $\text{RTT}_{\min}$. Thus, this assumption can be restated as follows: for a given route, $\text{RTT} = \text{RTT}_{\min}$ if and only if $T(1-p) \leqslant B_W(1-p_W)$. This in turn implies that if $T(1-p) > B_W(1-p_W)$ then $\text{RTT} \geqslant \text{RTT}_{\min}$.
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 and congestion-free.

We use the TFRC model described in Equation (5) to analyse 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 [2].

DOI: 10.1002/dac

The main idea behind TFRC is to provide a smooth transmission rate for streaming applications. The other properties of TFRC include slow response to congestion and the opportunity of not using all the available bandwidth for streaming applications. Consequently, in case of a single packet loss, TFRC does not halve its transmission rate like TCP, while, on the other hand it does not respond rapidly to the changes in available network bandwidth. TFRC has also been designed to behave 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]:

$$T = \frac{kS}{\mathrm{RTT}\sqrt{p}} \tag{5}$$

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 [15], depending on the particular derivation of Equation (5).

As it has already been mentioned, the average throughput measured at the receiver is $T \times (1-p)$, when the streaming rate is T and the overall packet loss rate is p. According to [4], the end-to-end packet loss rate p is a combination of p_{W} and $p_{\text{node name}}$ (node name: GGSN, SGSN, RNC, NODEB) and is computed as follows:

$$\begin{aligned} p = {} & p_{\mathrm{GGSN}} + (1-p_{\mathrm{GGSN}})p_{\mathrm{SGSN}} + (1-p_{\mathrm{GGSN}})(1-p_{\mathrm{SGSN}})p_{\mathrm{RNC}} \\ & + (1-p_{\mathrm{GGSN}})(1-p_{\mathrm{SGSN}})(1-p_{\mathrm{RNC}})p_{\mathrm{NODEB}} \\ & + (1-p_{\mathrm{GGSN}})(1-p_{\mathrm{SGSN}})(1-p_{\mathrm{RNC}})(1-p_{\mathrm{NODEB}})p_{\mathrm{W}} \end{aligned} \tag{6}$$

According to Equation (6), the total packet loss rate is the sum of the packet loss rate in each node of the network (Figure 3). 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 $p^{(1)}_{\text{node name}}$ and $p^{(2)}_{\text{node name}}$ 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, $p_{\text{node name}} = p^{(1)}_{\text{node name}} + p^{(2)}_{\text{node name}}$. 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

$$\begin{aligned} p = {} & p^{(2)}_{\mathrm{GGSN}} + (1-p^{(2)}_{\mathrm{GGSN}})p^{(2)}_{\mathrm{SGSN}} + (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})p^{(2)}_{\mathrm{RNC}} \\ & + (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})(1-p^{(2)}_{\mathrm{RNC}})(p^{(1)}_{\mathrm{NODEB}} + p^{(2)}_{\mathrm{NODEB}}) \\ & + (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})(1-p^{(2)}_{\mathrm{RNC}})(1-p^{(1)}_{\mathrm{NODEB}} - p^{(2)}_{\mathrm{NODEB}})p_{\mathrm{W}} \\ \Leftrightarrow & \\ p = {} & p^{(2)}_{\mathrm{GGSN}} + (1-p^{(2)}_{\mathrm{GGSN}})p^{(2)}_{\mathrm{SGSN}} + (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})p^{(2)}_{\mathrm{RNC}} \\ & + (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})(1-p^{(2)}_{\mathrm{RNC}})p^{(2)}_{\mathrm{NODEB}} \\ & + (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})(1-p^{(2)}_{\mathrm{RNC}})(1-p^{(2)}_{\mathrm{NODEB}})p_{\mathrm{W}} \\ & + (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})(1-p^{(2)}_{\mathrm{RNC}})p^{(1)}_{\mathrm{NODEB}} \\ & - (1-p^{(2)}_{\mathrm{GGSN}})(1-p^{(2)}_{\mathrm{SGSN}})(1-p^{(2)}_{\mathrm{RNC}})p^{(1)}_{\mathrm{NODEB}}p_{\mathrm{W}} \Leftrightarrow p = p_1 + p_2 \end{aligned} \tag{7}$$

DOI: 10.1002/dac

where

$$\begin{aligned} p_1 = & p^{(2)}_{\mathrm{GGSN}} + (1 - p^{(2)}_{\mathrm{GGSN}})p^{(2)}_{\mathrm{SGSN}} + (1 - p^{(2)}_{\mathrm{GGSN}})(1 - p^{(2)}_{\mathrm{SGSN}})p^{(2)}_{\mathrm{RNC}} \\ & + (1 - p^{(2)}_{\mathrm{GGSN}})(1 - p^{(2)}_{\mathrm{SGSN}})(1 - p^{(2)}_{\mathrm{RNC}})p^{(2)}_{\mathrm{NODEB}} \\ & + (1 - p^{(2)}_{\mathrm{GGSN}})(1 - p^{(2)}_{\mathrm{SGSN}})(1 - p^{(2)}_{\mathrm{RNC}})(1 - p^{(2)}_{\mathrm{NODEB}})p_{\mathrm{W}} \end{aligned} \tag{8}$$

and

$$p_2 = (1 - p^{(2)}_{\mathrm{GGSN}})(1 - p^{(2)}_{\mathrm{SGSN}})(1 - p^{(2)}_{\mathrm{RNC}})(1 - p_{\mathrm{W}})p^{(1)}_{\mathrm{NODEB}} \tag{9}$$

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. $p^{(1)}_{\mathrm{NODEB}}$ and thus may vary according to the streaming rate. Equation (7) 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. $p^{(1)}_{\mathrm{NODEB}} = 0$ and hence, $p_2 = 0$. By combining Equations (5) and (7), an upper bound T_{b}, on the streaming rate of one TFRC connection can be derived as follows:

$$T \leqslant \frac{kS}{\mathrm{RTT}_{\min}\sqrt{p_1}} \tag{10}$$

The communication between the sender and the receiver is based on RTP/RTCP sessions and the sender, denoted by Node 1 (Figure 3), uses the RTP protocol to transmit the video stream, whereas the client, denoted by UE1 (Figure 3), uses the RTCP protocol in order to exchange control messages. In the following paragraphs, details about the different aspects of 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 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 extracts the feedback information from the RTCP report and passes it through an appropriate filter. The use of filter is essential for the operation of the mechanism in order to avoid wrong estimations of the network conditions.

In the sender side, the media server using the feedback information 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 Equation (10) 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.

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 Equation (11).

$$T^{\text{Smoothed}} = \frac{\sum_{i=1}^{m} w_i \cdot T_{m+1-i}^{\text{Smoothed}}}{\sum_{i=1}^{m} w_i} \tag{11}$$

The value m used in calculating transmission rate determines TFRC's speed in responding to changes in the level of congestion [16]. 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 [16]. More details regarding the operation of the mechanism and the estimation of the parameters appearing in Equation (10) are presented in [17].

6. SIMULATIONS

This section reviews the main features of the simulation model that has been implemented by using the ns-2 simulator [18]. In the following we assume a media streaming system set-up consisting of a media streaming server, a transport channel and a streaming client. The above structure is depicted in Figure 3.

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 DCH, as it is shown in Figure 3. In the simulation, we use the DCH to transmit packet data. DCH is a bi-directional channel and is reserved only for a single user. The common channels are the FACH in the downlink and the RACH in the uplink as it is shown in Figure 3. In the simulation, we use a 384 kbit-DCH in the downlink and a 64 kbit-DCH in the uplink direction. The TTIs are 10 and 20 ms in the down- and uplink direction, respectively.

In order to understand the behaviour of UMTS for real time streaming video and measure the performance of the proposed rate control scheme we present two alternative scenarios:

1. Transmission of video traffic with RTP.
2. Transmission of adaptive video and TCP traffic in the same transport channel.

6.1. *Transmission of video traffic with RTP*

In this scenario, a video application is transmitted through a UMTS DCH. The video application is the same with the one used in Section 4. The duration of the experiment is 200 s. Figure 6 presents the end-to-end packet delay for the video transmission over dedicated channels. The x-axis gives the packet sequence number while y-axis shows the end-to-end packet delay in ms. In the simulation performed, the average end-to-end delay is 121.89 ms.

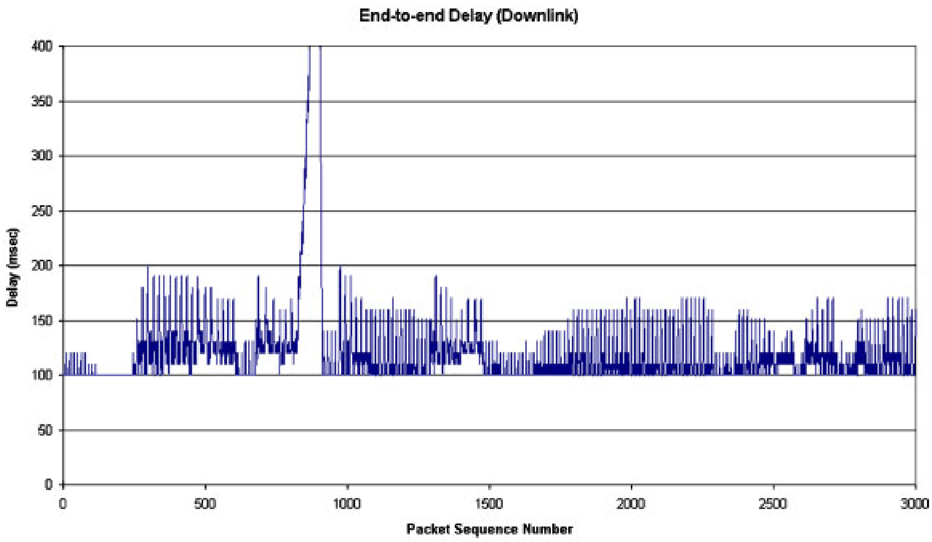

Figure 6. End-to-end delay for video transmission.

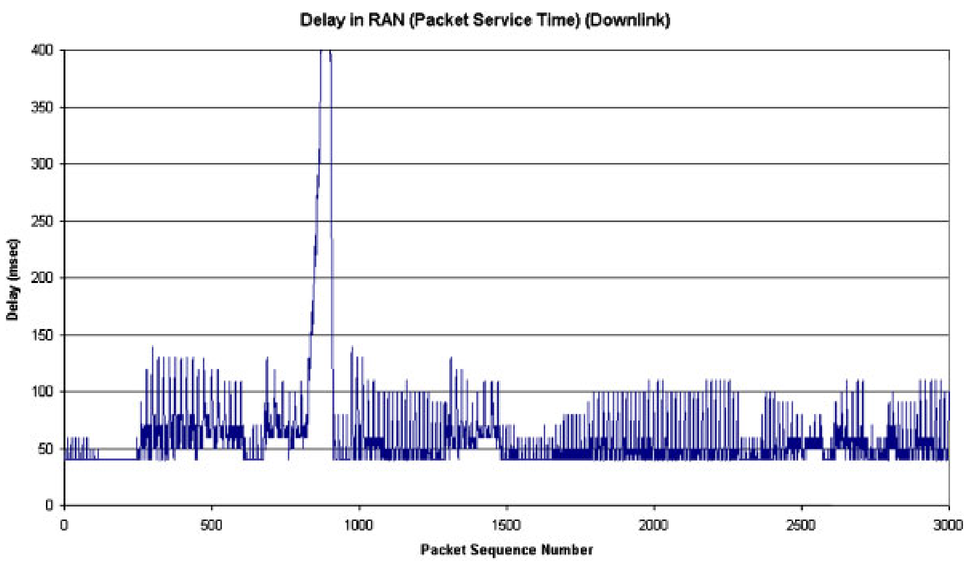

Figure 7. UTRAN delay for the video transmission.

Furthermore, Figure 7 presents the delay in UTRAN. In other words, the latter represents the packet service time that was computed, in an analytical way, in Section 4. The average packet service of the simulation has a value of 61.7 ms while the average packet service time that computed in Section 4 is 54.39 ms. As shown in Figures 6 and 7, around the packet sequence number 800 we can see a delay spike. This is due to the fact that these packets have very large

DOI: 10.1002/dac

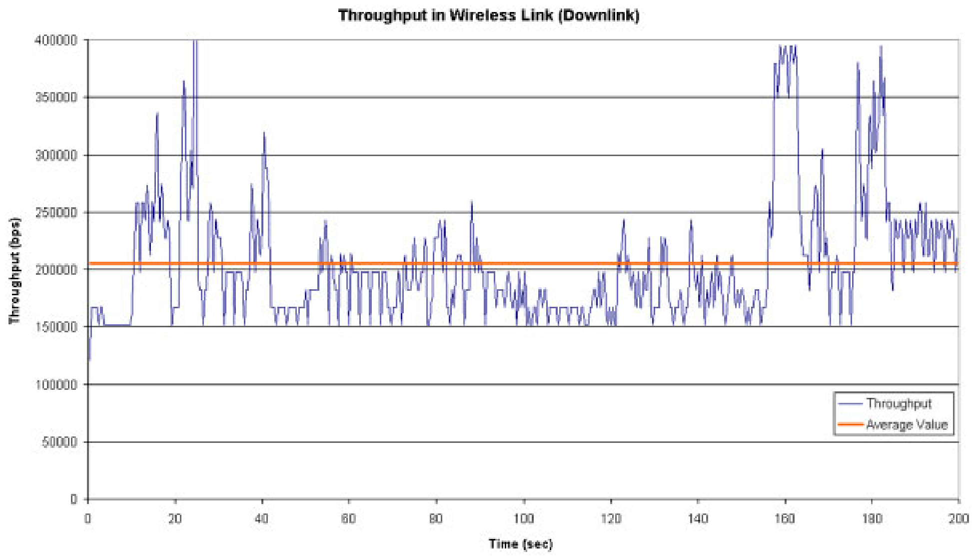

Figure 8. Throughput in wireless link.

size compared to the others and therefore, a great number of TTIs are required for the transmission of the packets.

The throughput in wireless link is depicted in Figure 8. The y-axis presents the throughput in bps while x-axis represents the duration of the simulation. The dark line represents the average throughput in the wireless link. The average throughput has a value of 205 kbps while the downlink bit rate of the dedicated channel is 384 kbps.

Figure 9 illustrates a comparison of pdfs of the packet service time for both the analytical computation (presented in Section 4) and the simulation results. The lighter line represents the experiment results, while the darker one represents the analytical results. As it is depicted in Figure 9, the simulation results are very close to the analytical results. Furthermore, Figure 9 indicates that the majority of the packets achieve a service time lower than 80 ms and only a small number of packets seem to achieve a service time higher than 100 ms. This signifies that the majority of the packets that reach the mobile user are characterized by low delay and consequently, the quality of the video sequence that the user will see in his terminal is satisfactory in comparison to the video sequence originally sent by the media server.

6.2. Transmission of adaptive video and TCP traffic

In this scenario we transmit video and TCP traffic in the same transport channel. 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

DOI: 10.1002/dac

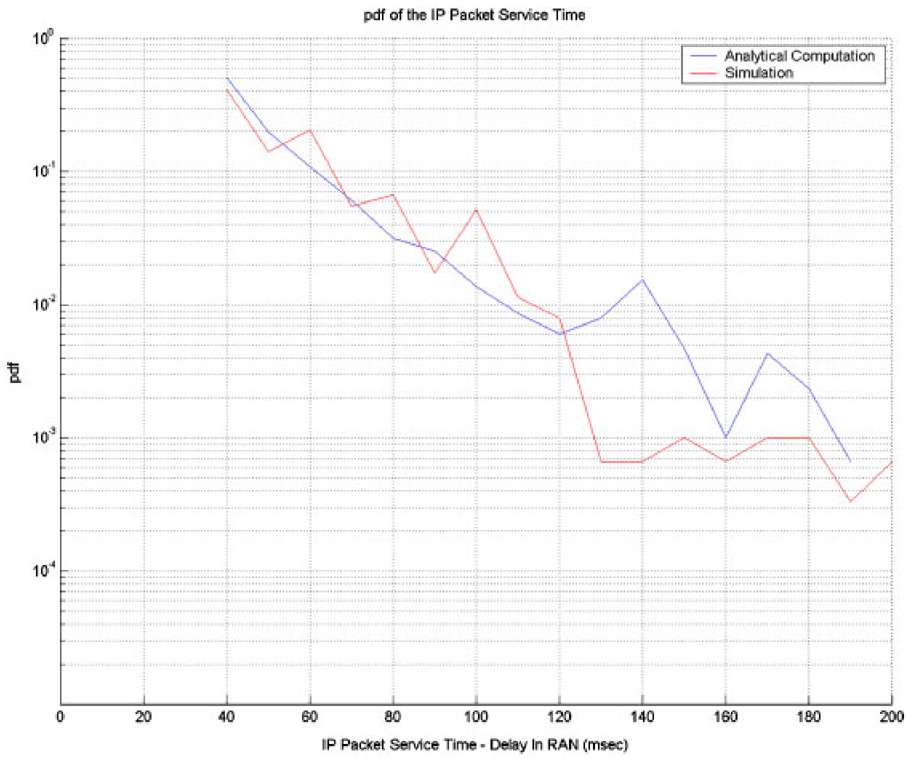

Figure 9. Comparison between analytical computation and simulation results.

simulation is 200 s. The video sequence is encoded to ITU-H.263 [13] 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 parameter k is set to $1.5 \cdot \sqrt{2/3}$ [3].

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 10 depicts the estimated transmission rate of the video sequence and it is calculated according to Equation (11). 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 10. 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

DOI: 10.1002/dac

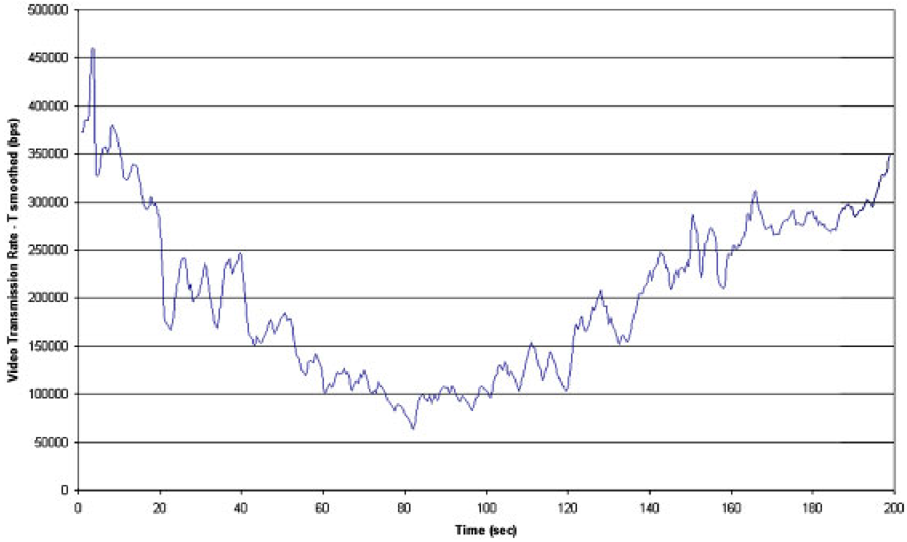

Figure 10. Calculated video transmission rate (T^{Smoothed}).

transmission rate (Figure 10). Additionally, 100 s after the beginning of the simulation, the smoothed transmission rate of the video increases. This is explained by the fact that in 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 11. The y-axis presents the throughput in bps, while the x-axis represents the duration of the simulation. Additionally, a darker line is used to demonstrate 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 s 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 behaviour. Consequently, for the throughput of the video in the wireless link the specific interval is around the value of 128 kbps. The rest of the 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 11, 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.

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 12. 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 12, the end-to-end delay of the TCP

DOI: 10.1002/dac

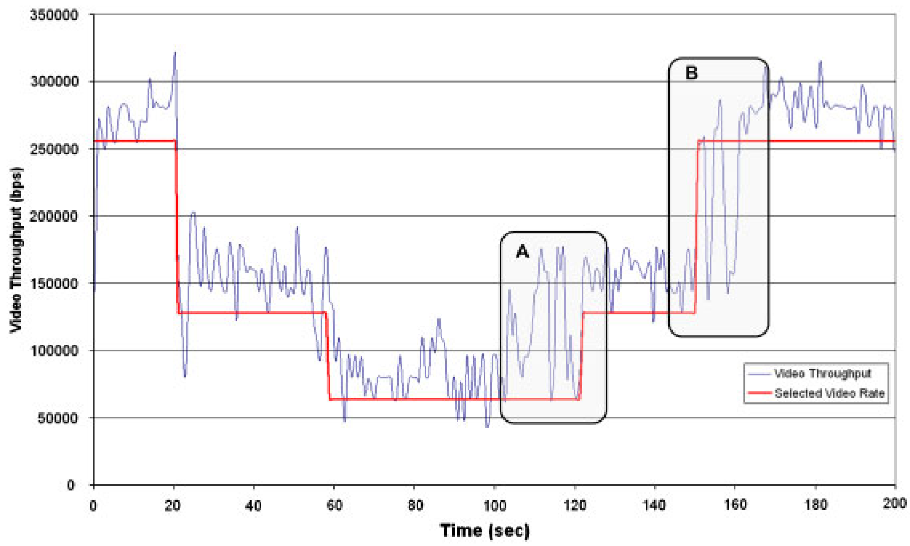

Figure 11. Video throughput in wireless link.

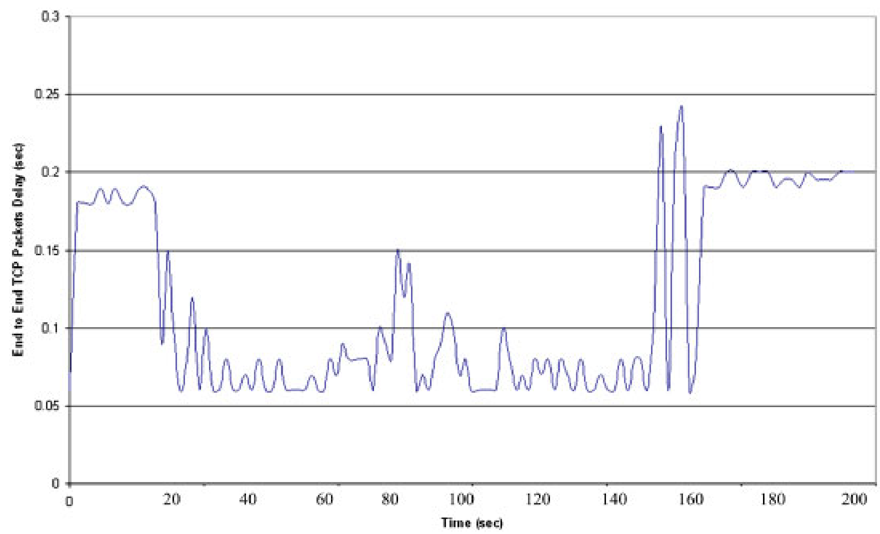

Figure 12. End to end delay of the TCP packets.

packets has a low value in the interval between 20 and 150 s, 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

DOI: 10.1002/dac

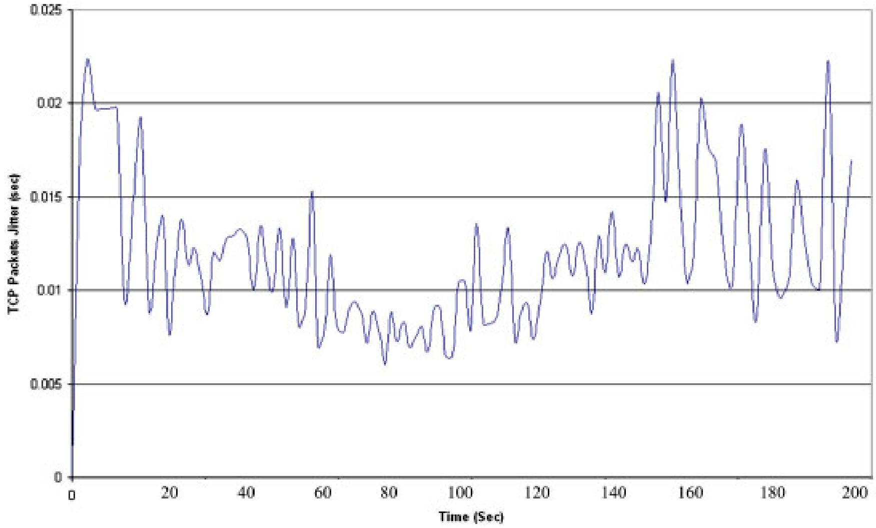

Figure 13. Delay jitter of the TCP packets.

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 13. 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 13) and restricted (Figure 12). 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.

7. EXPERIMENTAL VERIFICATION

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. In the experiments we use the following equipment:

- a real-time streaming server that supports RTSP, implemented on an Intel Pentium IV Desktop Computer with CPU speed at 2.86 GHz and 512 MB available RAM;
- a Release 99 3G data card;
- a laptop PC Pentium Mobile at 1.8 GHz.

Regarding the real time streaming server, video content of 500 s 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.

DOI: 10.1002/dac

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*: Despite the fact that 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 used an ftp program for the download of:

- A 3 MB file, from time 130 to 260 s
- A 2 MB file, from time 360 to 440 s

It has to be mentioned that as the experiments went on, there was no packet loss in the user's side.

In Figure 14 we can see the round trip time measured by server. Despite some variations at the beginning of the streaming, we can see that 140 s 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 s. The server in this time interval responds to this network condition with a decrease in the transmission rate (from 200 to 80 kbps) of the

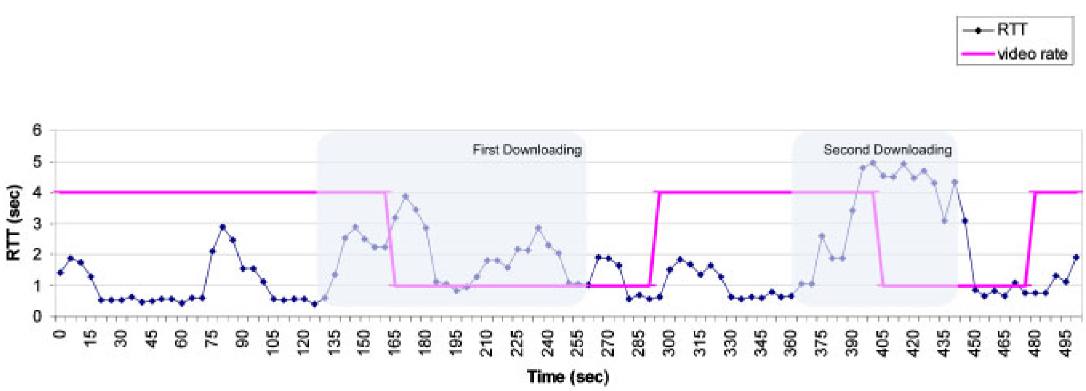

Figure 14. Round trip time.

DOI: 10.1002/dac

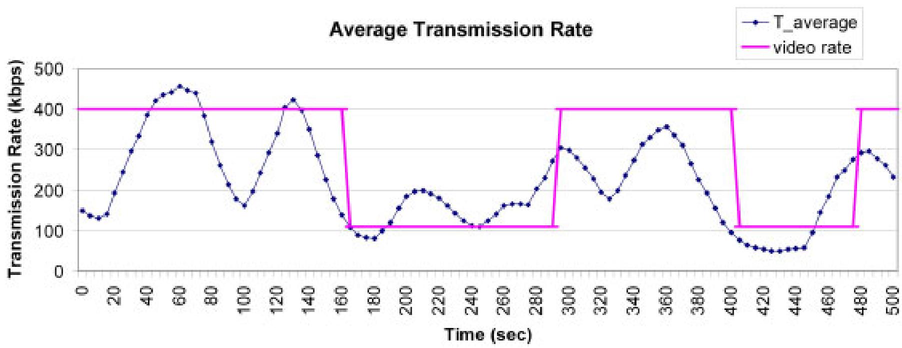

Figure 15. Calculated video transmission rate (T^{Smoothed}).

streaming video so that the mobile user can continue to be served satisfactorily. Until the time 260 s 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 s 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 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 s. In this time instance, RTT starts increasing the second file and thus the server switches to the low rate video (the network is characterized as congested by the server). Finally the user 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 s.

Figure 15 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. On this matter we should point out that the suggested average transmission rate 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 for the server to overcome the possible conjestion periods appearing in the network. In particular, as depicted in Figure 15, 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, at the time of 130 s, the suggested rate does not adapt quickly. 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 s the suggested rate decreases continuously. Thus, the server changes the streaming transmission rate to a lower value (from 200 to 80 kbps) in order to avoid congestion. When the user stops downloading the first file (at time 260 s) the rate increases until it reaches the value of 270 kbps at the time of 290 s. 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 behaves as before.

DOI: 10.1002/dac

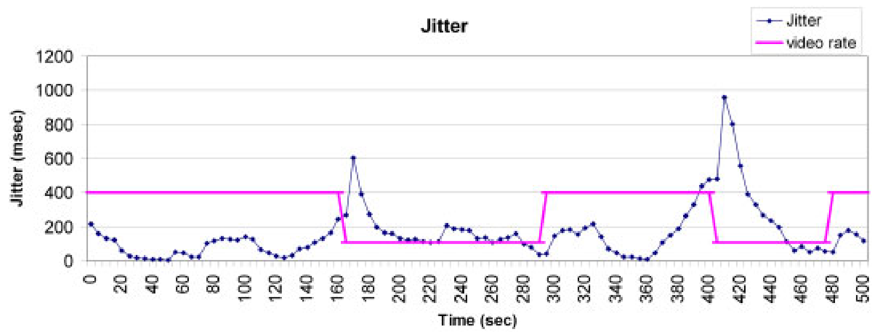

Figure 16. Jitter calculation.

Finally in Figure 16, we can see the interarrival jitter as it is calculated by the server. As expected, the jitter is low at the beginning of the experiment. During the network congestion because of the user's downloading, the jitter increases until the server reduces the sending video rate. The fact that the jitter is low during the downloading process means that the TFRC mechanism operates efficiently and that the server is able to adapt to the network changes. As a result, the streaming application continues to go on and the only change that is observed 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 ms.

8. FUTURE WORK

The step that follows this work is to evaluate the performance of adaptive video transmission over high speed downlink packet access transmissions (HSDPA). HSDPA supports the introduction of high bit rate data services and will increase network capacity, while minimizing operators' investment. The introduction of shared channels for different users will guarantee that channel resources are used efficiently in the packet domain and will be less expensive for users than the use of dedicated ones. Furthermore, shared channels give the opportunity to perform multicast in UMTS, which could result in bandwidth savings. In UMTS, bandwidth is a limited resource and the available radio resources can support only a handful high data-rate user simultaneously over a single cell. UMTS is therefore likely to have a mechanism for multicast packet forwarding.

Finally among the envisioned steps could be the study of the congestion control in the multicast scheme in UMTS. Thus, in cases that the mechanism realizes a possible congestion in a majority of the members of a multicast group, the transmission rate of the data could be reduced.

9. CONCLUSIONS

In this work, we focus on solutions for adaptive video transmission over UMTS transport channels. 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 video data. Through a number of simulations and experiments we validated the correctness and measured the performance and efficiency of the mechanism. A first level of evaluation is carried out using the ns-2 simulator. A second level then validates the proposed mechanism in real world traffic scenarios by performing experiments in a commercial UMTS network. Both simulations and experiments show that our mechanism handles fairly all the data flows sharing the available channel resources without causing network congestion.

REFERENCES

1. Holma H, Toskala A. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. Wiley: New York, 2003.
2. Floyd S, Handley M, Padhye J, Widmer J. Equation-based congestion control for unicast applications. *Proceedings of ACM SIGCOMM 2000*, Stockholm, Sweden, 2000; 43–56.
3. Floyd S, Fall K. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking* 1999.
4. Chen M, Zachor A. Rate control for streaming video over wireless. *Proceedings of the IEEE INFOCOM 2004*, Hong Kong, 2004; 32–41.
5. Stockhammer T, Jenkac H, Kuhn G. Streaming video over variable bit-rate wireless channels. *IEEE Transactions on Multimedia* 2004; **6**(2):268–277.
6. Balakrishnan H, Padmanabhan V, Seshan S, Katz R. A comparison of mechanisms for improving TCP performance over wireless links. *ACM SIGCOMM 1996*, CA, U.S.A., 1996; 256–269.
7. Fu CP, Liew SC. TCP veno: TCP enhancement for transmission over wireless access networks. *IEEE Journal on Selected Areas in Communications* 2003; **21**(2):216–228.
8. Choe H, Low SH. Stabilized vegas. *IEEE INFOCOM—The Conference on Computer Communications* 2003; **22**(1):2290–2300.
9. Chen L, Low SH, Doyle JC. Joint congestion control and media access control design for ad hoc wireless networks. *IEEE INFOCOM*, Miami, U.S.A., March 2005.
10. Xu K, Tian Y, Ansari N. Improving TCP performance in integrated wireless communications networks. *Computer Networks*, *Science Direct* 2005; **47**(2):219–237.
11. Zhao J, Kok C, Ahmad I. MPEG-4 video transmission over wireless networks: a link level performance study. *Wireless Networks*. Kluwer Academic Publishers: Dordrecht, 2004; 133–146.
12. 3GPP TS 23.060 V7.0.0. *Technical Specification Group Services and System Aspects*; *General Packet Radio Service* (*GPRS*); *Service Description*; *Stage 2* (*Release 7*), 2006.
13. Fitzek F, Reisslein M. MPEG-4 and H.263 video traces for network performance evaluation. *IEEE Network* 2001; **15**(6):40–54.
14. 3GPP. *Technical Specification Group (TSG) RAN*; *Delay Budget within the Access Stratum* (*Release 4*), *TR 25.853 V4.0.0 (2001–2003)*.
15. Mahdavi J, Floyd S. TCP-friendly unicast rate-based flow control. Available at http://www.psc.edu/networking/papers/tcp_friendly.html (January 1997).
16. Handley M, Floyd S, Padhye J, Widmer J. TCP friendly rate control (TFRC). *RFC 3448*, January 2003.
17. Alexiou A, Antonellis D, Bouras C. Adaptive and reliable video transmission over UMTS for enhanced performance. *International Journal of Communication Systems* 2006, in press.
18. The NS-2 Simulator. Available at http://www.isi.edu/nsnam/ns

AUTHORS' BIOGRAPHIES

Antonios Alexiou obtained his Diploma from the Department of Electrical and Computer Engineering of the Aristotle University of Thessaloniki (Greece) and his Master Degree from the Computer Engineering and Informatics Department of Patras University. He is currently a PhD Candidate of the Department of Computer Engineer and Informatics of Patras University. Furthermore he is working as R&D Computer Engineer at the Research Unit 6 of the Research Academic Computer Technology Institute in Patras (Greece). His research interests include Mobile Telecommunications Networks, Multicast Routing, User Mobility in Cellular Networks, Congestion Control and Quality of Service, Mobile and Wireless Ad-hoc Networks. He has published 4 papers in Journals and 16 papers in well-known refereed conferences.

DOI: 10.1002/dac

Christos Bouras obtained his Diploma and PhD from the Computer Science and Engineering Department of Patras University (Greece). He is currently an Associate Professor in the above department. Also he is a scientific advisor of Research Unit 6 in Research Academic Computer Technology Institute (CTI), Patras, Greece. His research interests include Analysis of Performance of Networking and Computer Systems, Computer Networks and Protocols, Telematics and New Services, QoS and Pricing for Networks and Services, e-learning, Networked Virtual Environments and WWW Issues. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published 200 papers in various well-known refereed conferences and journals. He is a co-author of 7 books in Greek. He has been a PC member and referee in various international journals and conferences. He has participated in R&D projects such as RACE, ESPRIT, TELEMATICS, EDUCATIONAL MULTIMEDIA, ISPO, EMPLOYMENT, ADAPT, STRIDE, EUROFORM, IST, GROWTH and others. Also he is member of, experts in the Greek Research and Technology Network (GRNET), Advisory Committee Member to the World Wide Web Consortium (W3C), IEEE Learning Technology Task Force, IEEE Technical Community for Services Computing WG 3.3 Research on Education Applications of Information Technologies and W 6.4 Internet Applications Engineering of IFIP, Task Force for Broadband Access in Greece, ACM, IEEE, EDEN, AACE and New York Academy of Sciences.

Vaggelis Igglesis obtained his Diploma and his Master's Degree from the Computer Engineering and Informatics Department of the University of Patras (Greece). His Master Thesis 'Data Transmission in 3G Mobile Networks' has been honored by the Ericsson Awards of Excellence in Telecommunications. He is currently working for European Dynamics SA, at the International Business Development Department. He has also worked as a Research & Development Engineer at the Research Unit 6 of the Research Academic Computer Technology Institute in Patras (Greece). His research interests include Mobile Telecommunications Networks, Telematics and Internet Technologies. He has published 2 papers in Journals and 8 papers in well-known refereed conferences.

DOI: 10.1002/dac