International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009
1793-8201

Equation Based Rate Control and Multiple Connections for Adaptive Video Streaming in Cellular Networks

A. Mardanian Dehkordi and V. Tabataba Vakili

***Abstract*— Rate control is an important issue for video streaming in cellular networks. In this paper, we propose an equation based rate control and multiple connections for adaptive video streaming in cellular networks. In our method the sending rate is calculated as a function of round trip time (RTT), loss event rate (p), packet size (s) and new control parameters α and β, that are able to provide flexible and smooth transmission rate and slowly responsible congestion control and also adaptability to unpredictable wireless channel conditions. On the other hand, by using one TFRC connection, the wireless bandwidth is underutilization, so we introduce a method by using more TFRC connections with new equation, it has the potential to achieve optimal performance, maximum throughput, and minimum packet loss rate. We have simulated this method in UMTS and according to results, this method in addition to network stability increases throughput with low fluctuation by varying α and β and opening appropriate number of connections.**

***Index Terms*— adaptive video streaming, cellular networks, rate control, TFRC, UMTS.**

I. Introduction

As communications technology is being developed, user's demand for multimedia services raises. One of the most promising services is the transmission of rich multimedia content, but bandwidth is a valuable and limited resource for UMTS and every wireless network. So currently, the poor video quality on the low and fluctuated bandwidth networks is making the use of streaming applications very difficult [1], [2]. Congestion control is aimed at solving this problem by adapting the streaming rate to the network conditions. A widely popular rate control scheme over wired networks is equation-based rate control also known as TCP friendly rate control (TFRC). It explicitly designed for best-effort unicast multimedia traffic, but it is not a full transport layer protocol [3]. Some applications require a fixed interval of time between packets and vary their segment size instead of their packet rate in response to congestion and according to [3] TFRC is not designed for those applications. On the other hand, by using the TFRC connection in wireless streaming applications results in underutilization of the wireless bandwidth. In this work we introduce a method that is based on new throughput equation and provides flexible and smooth transmission rate, slowly responsible congestion control and also it has the potential to fully utilize the wireless bandwidth provided number of connections. We have simulated this method in UMTS and according to results, this method in addition to network stability increases throughput with low fluctuation. This paper is structured as follows. Section 2 is a description of the related works. In Section 3 the TFRC mechanism with new equation for the UMTS is evaluated. In section 4 we propose strategy of using more connections to achieve maximum throughput. Section 5 is dedicated to the analytic and simulation results, Section 6 presents conclusion. Some ideas for future work are also outlined.

A. Mardanian Dehkordi is with Iran University of science and Technology, (corresponding author to provide phone: (+9821) 77808022-3; (e-mail: a_mardanian@vu.iust.ac.ir).

V. Tabataba Vakili is with Iran University of science and Technology, (e-mail: vakily@iust.ac.ir).

II. Related Works

Rate control is an important issue in both wired and wireless streaming applications. References [1] and [2], present streaming video over UMTS transport channels through the use of the modified the TFRC mechanism that is mainly used in wired networks. Reference [4] presents, TCP Reno, treats the occurrence of packet loss as a manifestation of network congestion. Equivalently, TCP Vegas uses queuing delay as a measure of congestion [5]. Thus, the authors [4], propose an enhancement of the TCP Reno and TCP Vegas for the wireless networks, namely TCP Veno. In Reference [6], two algorithms are presented that formulate resource allocation in wireless networks. In Reference [7], the performance characteristics of TCP New Reno, TCP SACK, TCP Veno and TCP Westwood under the wireless network conditions are studied and the authors propose TCP New Jersey, which is capable of distinguishing wireless packet losses from congestion. In other scheme [8], when a packet is lost, TFRC goes beyond layering abstraction and enquires the link layer about the recent signal strength. The packet loss is recognized to be due to wireless channel error if recent signal strength is low and due to congestion otherwise. Explicit Loss Notification (ELN) can also be applied to notify TCP/TFRC sender when a packet loss is caused by wireless channel errors rather than congestion [9]. In Reference [10], the authors interpret loss as a sign of congestion if one-way delays are increasing and a sign of wireless channel error otherwise. Similarly, Barman and Matta [11] proposed a loss differentiation scheme based on the assumption that the variance of round trip time is high when congestion occurs, and is low otherwise. Cen et. al. present an end-to-end based approach to facilitate streaming over wireless [12]. They combine packet inter-arrival times and relative one way delay to differentiate between packet loss caused by congestion, and that due to wireless channel errors. There are two key observations behind their approach; first, relative one way delay increases monotonically if there is congestion, second, inter-arrival time is expected to increase if there is packet loss caused by wireless channel errors. However, the high wireless error misclassification rate may result in under-utilizing the wireless bandwidth, as shown in [17]. Yang et. al. also propose a similar approach to improve video streaming performance in presence of wireless error, under the assumption that wireless link is the bottleneck [13]. Yang et.al. propose a cross-layer scheme that uses link layer information to determine whether a packet loss is caused by channel error or congestion, assuming that only the last link is wireless [14]. A similar assumption is made by Akan and Akyildiz in [15], to derive a wireless TFRC-like equation based protocol to facilitate video streaming. The disadvantage of end-to-end statistics based approaches is that congestion detection schemes based on statistics are not sufficiently accurate, and they either require cross layer information or modifications to the transport protocol stack. It is also possible to enable the routers with ECN markings capability to do rate control using ECN as the measure of congestion [16]. As packet loss no longer corresponds to congestion, ECN based rate control does not adjust sending rate upon observing a packet loss. MULTCP [17] and NetAnts [18], open multiple connections to increase throughput, MULTCP was originally used to provide differential service, and was later used to improve the performance in high bandwidth-round-trip time product networks and NetAnts achieves higher throughput by opening multiple connections to compete for bandwidth against others, but opening more connections than needed in wired Networks increases the end-to-end packet loss rate experienced by end host and there is no mechanism to control the number of connections in NetAnts. Multiple TFRC is an end to-end rate control solution for wireless video streaming [8]. The main differences between Multiple TFRC and our approach are as follows: First, in our method, we use a new throughput equation and the transmission rate is function of some other parameters such as α and β in addition to pervious mentioned parameters, which are able to provide flexible and smooth transmission rate and slowly responsible congestion control. Second, we use smoothed sending rate for new connections.

III. TFRC Mechanism with New Equation for UMTS

The equation used in TFRC reflects the throughput behaviour of a TCP flow, in this way any equation that models realistically a TCP flow could be used. In our method we use a novel rate estimation formula presented in (1). It is based on a modified version of the TCP Reno throughput equation (which is designed to compete fairly with TCP) and the sending rate (X) is calculated as a function of round trip time (RTT), loss event rate (p), packet size (s) and new control parameters (α, β) [19], which are able to provide flexible and smooth transmission rate and slowly responsible congestion control and also adaptability to unpredictable wireless channel conditions.

$$X = \frac{kS}{RTT\sqrt{\frac{2p(d-1)}{a(d+1)}} + 3RTO\sqrt{\frac{p(d-1)(d+1)}{2ad^2}}\,p(1+32p^2)} \qquad (1)$$

In equation (1), k is a constant factor between 0.7 and 1.3 [1] and RTO (retransmission timeout) is equal to $4RTT$ [3], and $d = 1/b$, so we have :

$$X = \frac{kS}{RTT\left(\sqrt{\frac{2p(d-1)}{a(d+1)}} + 12p\sqrt{\frac{p(d-1)(d+1)}{2ad^2}}(1+32p^2)\right)} \qquad (2)$$

By replacing α=1 and β=0.5 we have the following equation which is well known TFRC simplified throughput equation.

$$X = \frac{kS}{RTT\sqrt{\frac{2p}{3}} + 12p\sqrt{\frac{3p}{8}}(1+32p^2)} \qquad (3)$$

According to [20] the relationship between α and β to be TCP-friendly is:

$$a = \frac{4(1-b^2)}{3} \qquad (4)$$

This relationship offers a wide selection of possible values for α and β to achieve desired transient behaviours, such as responsiveness and reduced rate fluctuations. We use the typical scenario for streaming video over UMTS is shown in Figure1. It is include a UMTS radio cell covered by a Node B connected to a radio network controller (RNC) and user equipment (UE1) connected to a bi-directional channel (DCH) to transmit packet data 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.

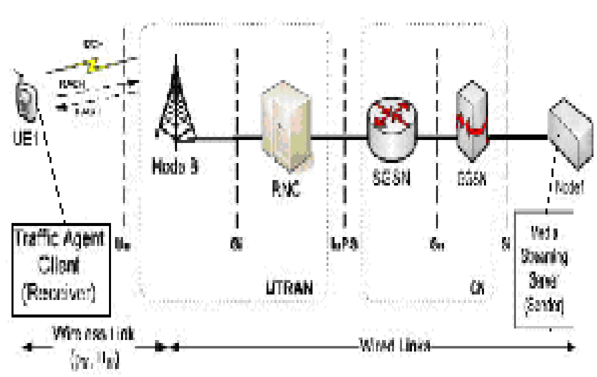

Fig. 1.Typical scenario for streaming video over UMTS

It is assumed that the wireless link has available bandwidth *Bw*, and packet loss rate *Pw*, caused by wireless channel error. So the maximum throughput that could be achieved in the wireless link is Bw(1-*Pw*). There could also be packet loss caused by congestion at wired nodes (GGSN, SGSN, RNC, NodeB). This means that the throughput is *X*(1-*P*) and *P* is end-to-end packet loss rate observed by the receiver. Under the above assumptions, the wireless channel as underutilized if $X(1-P) \le Bw(1-Pw)$. We use the TFRC model described in (2) to analyze the problem. According to [1], *P* is computed as follows:

$$P = P_{GGSN} + (1-P_{GGSN}) P_{SGSN} + (1-P_{GGSN})(1-P_{SGSN}) \quad (5)$$
$$= P_{RNC} + (1-P_{GGSN})(1-P_{SGSN})(1-P_{RNC}) P_{Node\,B} +$$
$$(1-P_{GGSN})(1-P_{SGSN})(1-P_{RNC})(1-P_{Node\,B}) P_w$$

We also assume the following:

- There is no congestion due to streaming traffic to the nodes GGSN, SGSN, RNC and Node B.
- There is no queuing delay caused at Node B, so the *RTT* has the minimum value.
- The backward route is congestion-free.
- Downlink bit rate *Bw* is 384 (kbps) and Uplink bit rate is 128 (kbps).
- Downlink *TTI* is 10(ms) and Uplink *TTI* is 20(ms).
- Node1 to GGSN bandwidth is 10(Mbps) and average delay is 15(ms).
- GGSN to SGSN bandwidth is 622(Mbps) and average delay is 10(ms).
- SGSN to RNC bandwidth is 622(Mbps) and average delay is 1 (ms).
- RNC to NodeB bandwidth is 622(Mbps) and average delay is 15(ms).
- RTCP report rate every 1s, Packet size *S* is 800(bytes) and *Pw* is varies from 0 to 0.16.
- $0 < b < 1, d = 1/b$, and according to TCP Friendly behavior $0 < a < 1.33$.

The communication between the sender and the receiver is based on RTP/RTCP sessions. The sender (server), use the RTP protocol to transmit the video stream and the mobile user (receiver) in recurrent time space sends RTCP reports to the sender. These reports contain information about the current conditions of the wireless link during the transmission of video between the server and the mobile user. The server using the feedback information and estimates the appropriate rate of the streaming video to avoid network congestion. From (2) and (6) the server is responsible for adjusting the sending rate with the calculated value and estimates the smoothed transmission rate**.**

$$X^{Smoothed} = \frac{\sum_{i=1}^{m} w_i . X^{Smoothed}_{m+1-i}}{\sum_{i=1}^{m} w_i} \quad (6)$$

So it is essential to keep a history of the previous calculated values for the transmission rate. We have chosen to keep track of eight values according to [1] Thus, we use *m*=8 and the values for weights $wi \in \{1,1,1,1,0.8,0.6,0.4,0.2\}$. The 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 and for prevent a single spurious packet loss having an excessive effect on the packet loss estimation, by using (7) the server smoothes the values of packet loss rate.

$$P^{Smoothed} = \frac{\sum_{i=1}^{m} w_i . P^{Smoothed}_{m+1-i}}{\sum_{i=1}^{m} w_i} \quad (7)$$

We assume the delay time of server to client is denoted by *Dsc* and the delay time of client to server is denoted by *Dsc*, so *RTT*=*Dsc*+*Dcs.* The client uses the timestamp of the received RTP packet and the local time which it is received in order to estimates the *Dsc*. We don't have the same delay into both directions for UMTS, so *Dsc*≠*Dcs* and *RTT*≠2*Dsc*, thus we need to estimate the accurate RTT. By using θ parameter we can say $RTT_{accurate} = (1+q)Dsc$. The sender estimates the RTT_{sample} every time it receives a receiver report from the client by using the following equation:

$$RTT_{sample} = (rxTime - txTime) - delayTime \quad (8)$$

In equation (8), *rxTime* is time the feedback packet was received by the sender, *txTime* is transmission time of last received data packet at the receiver and *delayTime* is time at receiver between reception of last packet and transmission of feedback packet, so we can estimate the parameter θ using the following equation:

$$q = \frac{RTT_{sample} - Dsc}{Dsc} \quad (9)$$

IV. Strategy of Using More Connections to Achieve Maximum Throughput

Equation (5) shows that *Pw* is a lower bound for *P* and if the wireless bandwidth is underutilized $(X \le Bw)$, then if we ignore the packet loss rate at the GGSN, SGSN, RNC and, NodeB caused by cross congestion, then $RTT = RTT_{Min}$ and $P = P_w$, so from (2) after we adjust α and β parameters, an upper bound on the sending rate of one TFRC connection (X_{ub}), can be derived as follows:

$$X_{ub} = \frac{kS}{RTT_{Min}\left(\sqrt{\frac{2p_w(d-1)}{a(d+1)}} + 12p_w\sqrt{\frac{p_w(d-1)(d+1)}{2ad^2}}(1+32p_w^2)\right)} \quad (10)$$

It is obvious that $X_{ub} \ge X$ and for one connection throughput is equal to $X_{ub}(1-Pw)$.

If $X_{ub} \le Bw$, then the wireless link is underutilized and If we use multiple simultaneous connections for a given streaming application, the total throughput is expected to increase with the number of connections until it reaches the hard limit of $Bw(1\text{-}Pw)$.

Let us now consider the case with two TFRC connections from Node1 to UE1 in Figure 1. In this case the aggregate throughput upper bound for both of them is $2X_{ub}(1-Pw)$, because of Pw for each of the two TFRC connections remain unchanged**.**

A similar analysis can be use for three or more TFRC connections, except that the wireless channel is no longer underutilized, $RTT \geq RTT_{Min}$ and $P \geq Pw$. In this case the wireless link is still fully utilized and $X(1\text{-}P) = Bw(1\text{-}Pw)$, but RTT is no longer at the minimum value. When full wireless channel utilization occurs, the maximum throughput in wireless link is equal to $N_{onc}.X_{ub}(1-Pw)$ and N_{onc} is the optimal number of connections so we can say:

$$Bw(1-Pw) = N_{onc}.X_{ub}(1-Pw) \tag{11}$$

In equation (11), N_{onc} is an integer value and so we can use (12) for the optimal number of connections.

$$N_{onc} = \left\lfloor \frac{Bw}{X_{ub}} \right\rfloor \tag{12}$$

If we open more than N_{onc} connections (N'), and we denoted the round trip time by RTT' and end-to-end packet loss rate by P', so from (2) we have:

$$X' = \frac{kS}{RTT'\left(\sqrt{\frac{2p'(d-1)}{a(d+1)}} + 12p'\sqrt{\frac{p'(d-1)(d+1)}{2ad^2}}(1+32p'^2)\right)} \tag{13}$$

So we can say:

$$N' = \frac{Bw(1-Pw)}{X'(1-P')} \tag{14}$$

Since $N' > N_{onc}$, if we compare the above equation with (11), we find out opening more than N_{onc} connections results in larger RTT and higher packet loss rate.

V. Analytic and Simulation Results

To validate the above analysis, we carry out experiments using MATLAB and NS-2 simulator [21].

We determine how each of the parameters affects the overall throughput; we show that by opening appropriate number of connections the throughput will be increased. Figure 2, evaluates the effect varying α in the interval 0.0−1.33 has on throughput for different values of $b \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$.

Next, the effect of varying the β between 0.0−1.0 for discrete values of $a \in \{0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.33\}$ is evaluated. The result of this analysis is illustrated in Figure 3 shows that the throughput grows exponentially when $0<\beta<1$. Infinite throughput is experienced as β converges on 1. This makes $\beta>0.8$ unsuitable for bit-rate tuning of multimedia streams due to its instability. The linearity between 0.0 − 0.8 makes β suitable as a tuning parameter. Figure 2 also shows the TCP-Friendly points that we use in our simulation to change the sending rate.

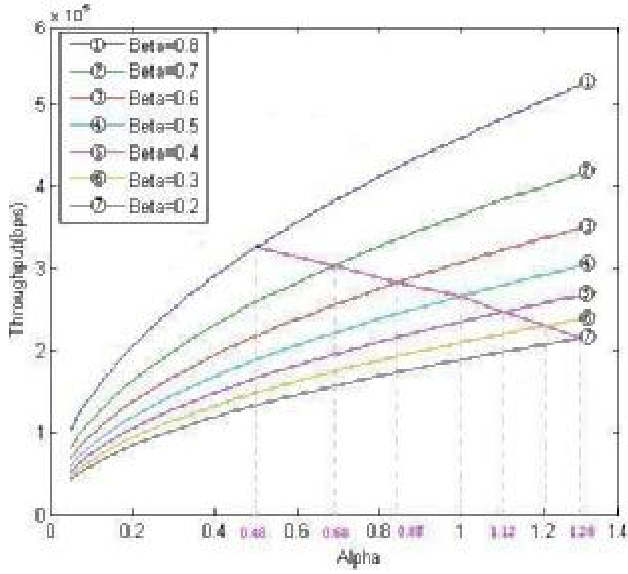

Fig. 2.The effect of varying α parameters on throughput for various values of β

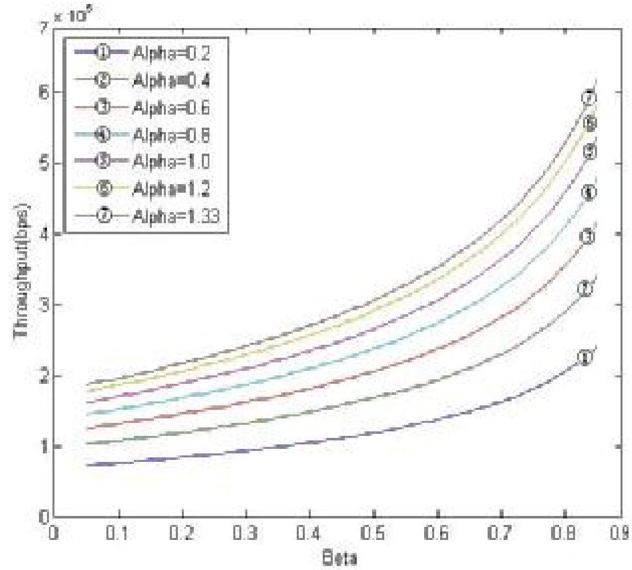

Fig. 3.The effect of varying β parameters on throughput for various values of α

We have simulated for 200 seconds and the server initially transmits the video with 256 kbps. We initially use α=1 and β=0.5 to adjust the transmission rate of the video. When the overall sending rate is increased, we observe increased packet losses due to congestion, so we measured this packet loss rate according to (5) and (7). Then we estimate the new transmission rate according to (2) by varying parameters. The server estimates the accurate RTT and calculates smoothed transmission rate by using the most recent values of the calculated sending rate according to (6). When the server observes an increase in the packet loss rate, it decreases the sending rate of the video to avoid network instability. The video throughput in wireless link is showed in Figure 4, with the average value of 206.6 kbps.

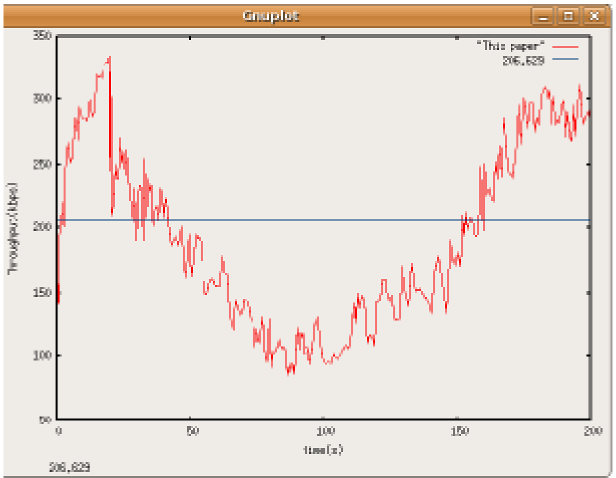

Fig. 4.The video throughput in wireless link

In other experiment we open more connections from Node1 to UE1 without changing α and β parameters and we could increase the average value of throughput by opening connection 2, but *Pw* and *RTT* are increased, after that we open another connection. It seems the total of the throughput is less than 2 connections but the packet loss rate and *RTT* are increased. The total throughput, packet loss rate and round trip time are measured (see Table 1). Clearly, the optimal number of connections is 2. Specifically, the loss rate is slightly higher for 3 connections than for 2, while the throughput is more or less the same for 2 and 3 connections.

Table1- Results for opening more connections

Number of Connections	Throughput(kbps)	*RTT*(ms)	*P*
one	206.6	140.2	0.016
two	162.3+153.9=316.2	270.5	0.035
three	107.3+104.7+103.1=315.1	282.4	0.05

In Figure 5 we show the optimal number of connections as a function of wireless link packet loss rate (*Pw*). We need to open more connections to compete for higher *Pw*.

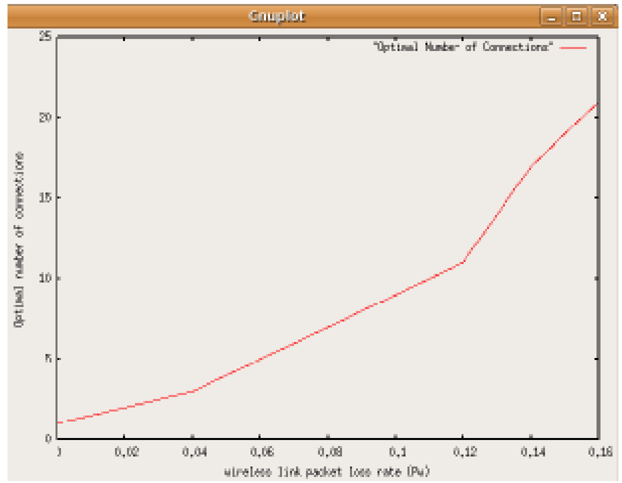

Fig. 5.The optimal number of connections

In General, This method in addition to network stability increases throughput with low fluctuation and achieves desired transient behaviors.

VI. Conclusion

Rate control is an important issue in video streaming applications. In this work, we focus on a mechanism for equation based congestion control for video transmission over UMTS networks. All previous works on this subject were based on the simplified TCP Reno equation in which sending rate is determined as a function of the packet size, the round trip time and the packet loss rate. In order to change transmission rate, packet size or distance between to sequential transmission must be modified. By changing packet size, TFRC's characteristic will change. In this research another throughput equation is used and transmission rate is function of some other parameters such as α and β in addition to pervious mentioned parameters. In our method by using α and β transmission rate will be more smoothed in congested conditions and is related to loss rate. We demonstrated that higher throughput could be achieved and it provides smoother throughput variations. Although our method increased throughput by opening appropriate number of connections and made network more stable using rate control, and also guarantees good quality in delivered video but we need more system resources for opening more connections. As a future work by using large receiver side buffering and automatic configuration of α and β we can improve quality of service. Another suggestion for future work is fairness in bandwidth allocation according to user requirements and considering the stability issues and examining the performance when both the number of connections and the sending rate of each connection are changing dynamically.

References

[1] Antonios G. Alexiou, Dimitrios Antonellis, Christos Bouras: "Adaptive and reliable video transmission over UMTS for enhanced performance", *Int. J. Communication* Systems 20(1): 65-81 (2007).

[2] Antonios G. Alexiou, Christos Bouras, Vaggelis Igglesis, Scalable rate control for video transmission over UMTS. International Journal of Communication Systems archive. Volume 20, Issue 12 (Dec. 2007).

[3] M. Handley, S. Floyd, J. Padhye, and J. Widmer, TCP Friendly Rate Control (TFRC): Protocol Specification, IETF, RFC5348 (Sep. 2008).

[4] 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.

[5] Choe H, Low SH. Stabilized Vegas. IEEE INFOCOM. The Conference on Computer Communications, 2003;22 (1):2290–2300.

[6] Chen L, Low SH, Doyle JC. Joint congestion control and media access control design for ad hoc wireless networks. IEEE INFOCOM, 2005.

[7] Xu K, Tian Y, Ansari N. Improving TCP performance in integrated wireless communications networks. Computer Networks, Science Direct 2005; 47(2):219–237.

[8] Minghua Chen, and Avideh Zakhor, Multiple TFRC Connections Based Rate Control for Wireless Networks, (2006). IEEE Transactions on Multimedia. 8 (5), pp. 1045-1062.

[9] H. Balakrishnan and R. Katz, Explicit loss notification and wireless web performance, in Proc. of IEEE Globecom Internet Mini-Conference, Nov. 1998.

[10] T. eun Kim, S. Lu, and V. Bharghavan, Improving congestion control performance through loss differentiation, in Proc. ICPP Workshop, 1999, pp. 140–145.

[11] D. Barman, I. Matta: "Effectiveness of loss labeling in improving TCP performance in wired/wireless networks", in Proc. of the 10th ICNP, Washington, DC, USA, 2002, p.p. 2–11.

[12] S. Cen, P. Cosman, and G. Voelker, "End-to-end differentiation of congestion and wireless losses", IEEE/ACM Trans. Networking, vol. 11, no. 5, pp. 703–717, 2003.

[13] G. Yang, M. Gerla, and M. Y. Sanadidi, "Adaptive video streaming in presence of wireless errors," in Proc. ACM MMNS, San Diego, USA, Jan. 2004.

[14] F. Yang, Q. Zhang, W. Zhu, and Y. Q. Zhang, "End-to-end TCP-Friendly streaming protocol and bit allocation for scalable video over mobile wireless internet," in Proc. IEEE INFOCOM, Hongkong, China, 2004.

[15] B. Akan and I. F. Akyildiz, "The analytical rate control scheme for real-time traffic in wireless networks," IEEE/ACM Trans. Networking, vol. 12, no. 4, pp. 634–644, 2004.

[16] S. Floyd, "Tcp and explicit congestion notification," ACM Computer Communication Review, pp. 10–23, Oct. 1994.

[17] J.Crowcroft and P. Oechslin, "Differentiated end to end internet services using a weighted proportional fair sharing tcp," ACM Computer Communication Review, vol. 28, no. 3, July 1998.

[18] Netants. Fast download manager. [Online]. Available: http://www.netants.com.

[19] A. Kuzmanovic, E. W. Knightly, "A Performance vs. Trust Perspective in the Design of End-Point. Congestion Control Protocols," 12th IEEE International Conference on Network Protocols (ICNP04), 2004.

[20] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control," in Proc. 8th IEEE International Conference on Network Protocols (ICNP), Osaka, Japan, November 2000.

[21] Network simulation version 2. [Online]. Available: http://www.isi.edu/nsnam/ns/.