The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)

SELECTIVE RATE CONTROL FOR MEDIA-STREAMING APPLICATIONS IN WIRELESS INTERNET ENVIRONMENTS

Panagiotis Papadimitriou and Vassilis Tsaoussidis
Democritus University of Thrace, Electrical and Computer Engineering Department
Xanthi, Greece

ABSTRACT

Media-streaming applications experience limited performance and perceptible quality degradation in the presence of random wireless errors, as the underlying congestion control typically interprets packet loss as the outcome of congestion. In this context, we propose a selective rate control, namely AIAMD, which manages to differentiate congestive and non-congestive loss by utilizing *history* in its control rules. AIAMD combines the most desirable features of *Additive Increase Additive Decrease* (AIAD) and *Additive Increase Multiplicative Decrease* (AIMD) controls, reacting gently to wireless loss and more aggressively to congestion. Exploring AIAMD's potential, we identify notable gains in terms of link utilization and media delivery, without compromising intra-protocol fairness.

I. INTRODUCTION

In recent years Internet has been experiencing an increasing demand for multimedia services, typically involving audio and video delivery. Media-streaming applications yield satisfactory performance only under certain *Quality of Service* (QoS) provisions, which may vary depending on the application task and the type of media involved. Unlike bulk-data transfers, multimedia flows require a minimum and continuous bandwidth guarantee, while they are also affected by reliability factors, such as packet drops due to congestion or link errors.

Today's media-streaming applications are expected to run in physically heterogeneous environments composed of both wired and wireless components. Wireless links exhibit distinct characteristics, such as limited effective bandwidth, varying error-rates and potential handoff operations. Link-layer error control, such as *Automatic Repeat Request* (ARQ) [7], is widely used to provide reliable and efficient transmission over wireless channels. However, the presence of such mechanisms may result in dramatic bandwidth and delay variations in wireless networks, degrading the performance of multimedia applications. Considering also time-sensitive traffic where data packets bear information with a limited useful lifetime, retransmissions are often a wasted effort.

The Internet is governed by the rules of *Additive Increase Multiplicative Decrease* (AIMD) [6], which effectively contribute to its stability. Essentially, the goal of such algorithms is to prevent applications from either overloading or under-utilizing the available network resources. Although AIMD-based *Transmission Control Protocol* (TCP) provides reliable and efficient services for bulk-data transfers, several design issues render the protocol a less attractive solution for multimedia applications. More precisely, the protocol causes considerable variations in the transmission rate, due to its abrupt multiplicative decrease adjustments upon congestion. In addition, TCP occasionally introduces arbitrary delays, since it enforces reliability and in-order delivery. Furthermore, most existing TCP mechanisms, as well as TCP-friendly protocols [8, 20, 21], do not satisfy the need for universal functionality in heterogeneous wired/wireless environments, since they do not flexibly adjust the rate and pattern of the media streams to the characteristics of the end-to-end network path. In particular, TCP utilizes wireless resources inefficiently, as it invokes unnecessary congestion-oriented responses to wireless errors and operations (e.g. handoffs) [19].

Tackling TCP's inefficiency over wireless links, split connection protocols, such as *Indirect-TCP* [1], split a TCP connection into two separate connections by installing an agent at every base station (BS) in the entire wireless communication system. Apparently, split connection protocols can not be easily deployed, while they also violate TCP's end-to-end semantics. TCP-aware *Snoop* protocol [2] recovers link errors locally, using link-level buffers at the BS to cache packets traversing the wireless channel. Despite potential throughput gains, buffering incoming packets at the BS usually increases queuing delays, inducing perceptible variations in *Round Trip Times* (RTT) and disturbing fluctuations in the receiving rate [15]. Alternatively, existing end-to-end loss algorithms can be applied to decouple congestion from wireless errors, based on packet inter-arrival times [4]. However, inferring a specific behavior from inter-arrival times or packet pair may be inaccurate, due to the variation and complication of traffic patterns in the Internet.

Considering TCP's limitations and the impending threat of unresponsive UDP traffic, media delivery over heterogeneous networks requires rate control, which (i) enables the desired smoothness for media-streaming applications, (ii) avoids significant damage during congestion, and (iii) maintains friendliness with coexisting TCP flows. The two latter requirements are anticipated by most existing congestion control schemes; however, the former one can be only attained by a smooth transmission control that also reacts gracefully to wireless errors.

Following these observations, we propose a selective rate control, namely AIAMD, which combines the most desirable features of AIMD and *Additive Increase Additive Decrease* (AIAD) in order to adapt the sending rate to the characteristics of the end-to-end network path. In contrast to AIAD and AIMD algorithms, AIAMD utilizes *history* of measured rates in order to distinguish between congestion and error-induced loss. We note that history information was very rarely used in linear congestion controls (e.g. [11]). Exploiting history, AIAMD reacts gracefully to error-induced loss in order to keep the sending rate variation to minimum, but responds quickly to the onset of congestion. AIAMD can be easily adapted and incorporated into existing transport protocols. Rate-based protocols [8, 14, 16], in particular, generate a smooth data flow by spreading the data transmission across a time interval, avoiding the burstiness occasionally induced by window-based mechanisms. In this context, we incorporate AIAMD into *Scalable Streaming Video Protocol* (SSVP) [14] and evaluate its efficiency in terms of link utilization, fairness and media delivery.

We organize the remainder of the paper, as follows. The following section summarizes related work. In Section 3, we elaborate on the selective rate control. Section 4 includes conclusive performance studies based on simulations. Finally, in the last section we highlight our conclusions.

II. Related Work

Numerous proposals have been presented in order to improve transport-layer efficiency over wireless links [2, 10]. Most related research efforts focus on bulk-data transmission, and are usually pronounced as enhanced TCP versions. *TCP Probing* [18] grafts a probing cycle and an *Immediate Recovery Strategy* into standard TCP in order to control effectively the throughput/overhead trade-off. *Freeze-TCP* [9] distinguishes handoffs from congestion using the advertised window. *WTCP* [17] implements a rate-based congestion control replacing entirely the ACK-clocking mechanism. *TCP Westwood* [13] is a sender-side-only modification of TCP Reno congestion control that incorporates a recovery mechanism avoiding the blind halving of the sending rate of TCP Reno after packet loss and achieving high link utilization in the presence of wireless errors. Authors in [3] propose *TIBET* (*Time Intervals based Bandwidth Estimation Technique*), a new bandwidth estimation scheme implemented within TCP congestion control, which enhances TCP performance over wireless links.

The literature also includes several proposals for efficient rate control for media-streaming applications in the Internet. Since TCP is rarely chosen to transport multimedia traffic over the Internet, *TCP-friendly* protocols [8, 20, 21] constitute an elegant framework for multimedia applications. *TCP-friendly Rate Control* (TFRC) [8] is a representative TCP-friendly protocol, which adjusts its transmission rate in response to the level of congestion, as estimated based on the calculated loss rate. Multiple packet drops in the same RTT are considered as a single loss event by TFRC and hence, the protocol follows a more gentle congestion control strategy. However, TFRC's throughput model is quite sensitive to parameters (i.e. packet loss, RTT), which are often difficult to measure efficiently and to predict accurately. *MULTFRC* [5] is a recent extension to TFRC for wireless networks, establishing multiple TFRC connections on the same path, when a single connection is not able to utilize the wireless resources efficiently.

GAIMD [21] is a TCP-friendly protocol that generalizes AIMD congestion control by parameterizing the additive increase rate α and multiplicative decrease ratio β. Based on experiments, authors in [21] propose an adjustment of β = 0.875 as an appropriate smooth decrease ratio, and a moderated increase value α = 0.31 to achieve TCP friendliness.

Rate Adaptation Protocol (RAP) [16] is a rate-based protocol which employs an AIMD algorithm for the transmission of real-time streams. The sending rate is continuously adjusted by RAP in a TCP-friendly fashion, using feedback from the receiver. However, since RAP employs TCP's congestion control parameters (i.e. 1, 0.5), it causes short-term rate oscillations, primarily due to the multiplicative decrease.

Datagram Congestion Control Protocol (DCCP) [12] is a new transport protocol that provides a congestion-controlled flow of unreliable datagrams. DCCP is intended for delay-sensitive applications which have relaxed packet loss requirements. The protocol provides the application with a choice of congestion control mechanisms via *Congestion Control IDs* (CCIDs), which explicitly name standardized congestion control mechanisms (i.e. TCP-like and TFRC).

III. Selective Rate Control for Wireless Networks

We propose a selective AIAD/AIMD (AIAMD) control with history, beyond the conventional approach of purely additive increase and multiplicative decrease. According to the model in [6], we consider n users sharing a single bottleneck link. If during time slot t, the i^{th} user's load is $x_i(t)$, then the total load at the bottleneck resource would be $x(t)=\sum_{i=1}^{n} x_i(t)$. The network provides the users with a binary feedback y(t), which indicates whether the total load x(t-1) after the previous adjustment exceeds an optimal value X_{goal}:

$$y(t)=\begin{cases} 1 & \text{if } x(t-1) > X_{goal} \\ 0 & \text{if } x(t-1) \le X_{goal} \end{cases} \quad (1)$$

Linear congestion control algorithms are governed by the following update function:

$$x_i(t)=\begin{cases} \alpha_I + \beta_I\, x_i(t-1) & \text{if } y(t)=0 \\ \alpha_D + \beta_D\, x_i(t-1) & \text{if } y(t)=1 \end{cases} \quad (2)$$

In the case of AIMD (β_I = 1, α_D = 0), user i responds to binary feedback y(t), as follows:

$$x_i(t)=\begin{cases} \alpha_I + x_i(t-1) & \text{if } y(t)=0 \\ \beta_D\, x_i(t-1) & \text{if } y(t)=1 \end{cases} \quad (3)$$

where $\alpha_I > 0$ and $0 < \beta_D < 1$. Following AIAD ($\beta_I = \beta_D = 1$), the corresponding response for user i is:

$$x_i(t)=\begin{cases} \alpha_I + x_i(t-1) & \text{if } y(t)=0 \\ \alpha_D + x_i(t-1) & \text{if } y(t)=1 \end{cases} \quad (4)$$

where $\alpha_I > 0$ and $\alpha_D < 0$. We note that neither of these controls utilizes history information; the increase and decrease rules depend solely on current load and parameters α_I, α_D, β_I and β_D.

According to AIAD, in the absence of packet loss the rate is gracefully increased to probe for additional bandwidth;

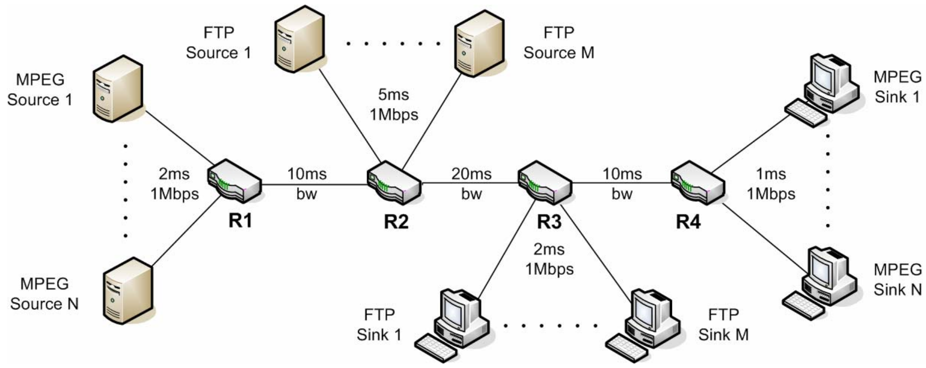


Figure 1: Simulation Topology

otherwise, the transmission rate is gently decreased in order to alleviate congestion. The graceful rate adjustments of AIAD overcome a number of problems related with AIMD rate control. More precisely, AIAD is less susceptible to transient loss, results in higher link utilization and does not induce significant fluctuations in the transmission rate. However, AIAD's responsiveness is poor upon sudden congestion, due to its additive decrease policy. In this case, AIMD responds more aggressively in order to confine packet loss. The proposed rate control combines the strengths of AIAD and AIMD, reacting gently to wireless loss and more aggressively to congestion, adapting effectively to the dynamics of the network.

AIAMD is able to differentiate congestive and non-congestive loss, by maintaining a history of measured rates throughout the connection. Observations of the network dynamics and event losses are frequently assumed within a time period of an *epoch*. We consider an epoch as the time period between two observed loss events (i.e. during an epoch the transmission rate evolves uninterrupted). The measured rates at the end of each epoch are useful, since they compose a good predictor for the congestion state for the following epochs. AIAMD needs to acquire the measured rate from the transport protocol, i.e. either a sender-side estimation (e.g. TCP Westwood) or a direct measurement at the receiving host (e.g. SSVP).

We define the state variable R, which is the measured rate at the end of each epoch. AIAMD keeps track of the profile of R by maintaining two variables: the moving average of the measured rate $\overline{R}$ and the average deviation of the measured rate $\overline{D}$. We also use a constant n which is experimentally set to 1.5. Generally, n can be adjusted in order to modify the transient behavior of the control. In the event of packet loss, a rate within the $\overline{R}-n\overline{D}$ bound indicates a wireless loss. In this case, AIAMD invokes an additive decrease in the transmission rate. On the other hand, if the measured rate falls below $\overline{R}-n\overline{D}$, the control scheme infers congestion and subsequently triggers a multiplicative decrease. With respect to equations (1)-(4), the transmission rate is adjusted based on the following algorithm:

$$R(t)=\begin{cases}\alpha_I + R(t-1) & \text{if } y(t)=0\\ \alpha_D + R(t-1) & \text{if } y(t)=1 \text{ and } R(t-1)\geq \overline{R}-n\overline{D}\\ \beta_D\, R(t-1) & \text{if } y(t)=1 \text{ and } R(t-1)< \overline{R}-n\overline{D}\end{cases} \quad (5)$$

We adopt the conventional AIAD/AIMD parameters: α_I = 1, a_D = -1, and β_D = 0.5. AIAMD preserves AIAD's property of gentle variations in the transmission rate for wireless losses, enabling the desired smoothness for media-streaming applications. At the same time, the selective control reacts more aggressively in response to the reduction of network resources or the advent of new connections.

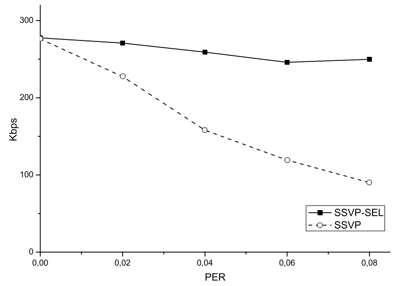
We incorporate AIAMD into SSVP [14] in order to assess its efficiency. SSVP is a congestion control scheme which operates on top of UDP and is optimized for video streaming applications. SSVP employs an AIMD-based rate control with α_I = 0.31 and β_D = 0.875. The transmission rate is controlled by properly adjusting the inter-packet-gap (IPG). If no congestion is sensed, IPG is reduced additively; otherwise, it is increased multiplicatively. The recipient uses control packets to send feedback of reception statistics to the sender. Hence, AIAMD is able to maintain the variables $\overline{R}$ and $\overline{D}$, and subsequently determine the appropriate error-recovery strategy.

IV. PERFORMANCE EVALUATION

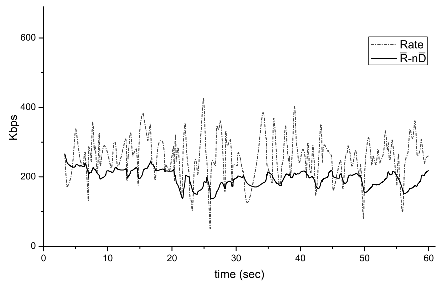
A. Experimental Environment

The evaluation plan was implemented on the *NS-2* network simulator. The simulations were conducted on a network topology that includes multiple bottlenecks, cross traffic, and wireless links (Fig. 1). Cross traffic includes FTP flows over TCP Reno. The capacity of the bottleneck links *bw* is configured depending on the experiment performed. NS-2 error models were inserted into the access links to the MPEG sink nodes. We used the Bernoulli model in order to simulate the errors on both directions of the link traffic. We set the packet size to 1000 bytes for all system flows and the maximum congestion window to 64 KB for all TCP connections. We used drop-tail routers with buffer size adjusted in accordance with the *bandwidth-delay* product. The duration of each simulation is 60 sec. All results are collected after 2 sec to avoid the skew introduced by the startup effect.

In order to simulate MPEG traffic, we developed an *MPEG-4 Traffic Generator*. The traffic generated closely matches the statistical characteristics of an original MPEG-4 video trace. We used three separate *TES* models for modeling I, P and B frames respectively. The resulting MPEG-4 stream

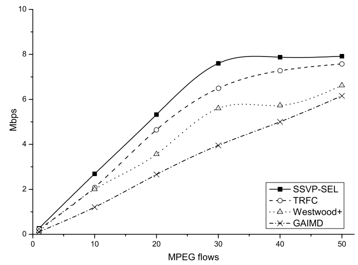


(a) Throughput vs. PER

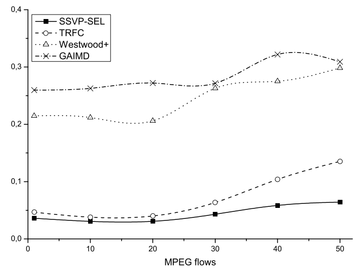


(b) Sending Rate and $\overline{R}-n\overline{D}$ (PER 0.04)

Figure 2: Performance of single SSVP-SEL flow

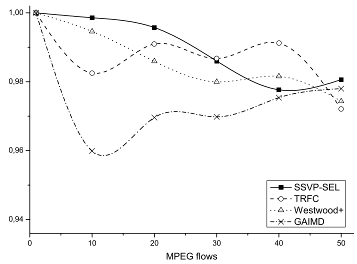


(a) Throughput of MPEG flows

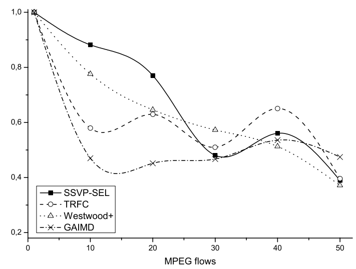


(b) Delayed Packets Rate

Figure 3: Performance with high link-multiplexing



(a) Fairness Index



(b) Worst-Case Fairness

Figure 4: Fairness

is generated by interleaving data obtained by the three models.

We hereby refer to the performance metrics supported by our simulation model. *Throughput* is used to measure the efficiency in link utilization. Long-term fairness is measured by *Fairness Index*, derived from the formula given in [6], and defined as:

$$\text{Fairness Index} = \frac{(\sum_{i=1}^{n} \text{Throughput}_i)^2}{n(\sum_{i=1}^{n} \text{Throughput}_i{}^2)}$$

where $Throughput_i$ is the throughput of the i^{th} flow and n is the total number of flows. As a supplementary fairness metric, we use *Worst-Case Fairness*, defined as $\min_{1 \le i \le n}(\text{throughput}_i)/\max_{1 \le i \le n}(\text{throughput}_i)$, in order to conduct a *worst-case* analysis and provide a tight bound on fairness. Both fairness metrics range in [0, 1] with 1 representing the absolute fairness.

In order to quantify the performance on video delivery, we monitor packet inter-arrival times and eventually distinguish the packets that can be effectively used by the client application (i.e. without causing interruptions) from delayed packets according to a configurable packet inter-arrival threshold. The proportion of delayed packets is denoted as *Delayed Packets Rate*. In accordance with media streaming requirements, we adjusted the packet inter-arrival threshold at 75 ms.

B. Results and Discussion

In this section, we demonstrate conclusive performance studies based on selected simulation results. First, we assess the efficiency of the selective rate control for a diverse range of packet error rates (*PER*). In this context, we simulated a single MPEG flow of (i) SSVP (1, 0.5) and (ii) SSVP with selective control (SSVP-SEL), competing with two FTP flows over TCP Reno. The bottleneck capacity *bw* is set to 1 Mbps in the simulation topology. Note that although SSVP originally employs AIMD with $\alpha_I = 0.31$ and $\beta_D = 0.875$, we chose $\alpha_I = 1$ and $\beta_D = 0.5$ for the simulated SSVP flow in order to obtain comparable results between SSVP and SSVP-SEL (whose AIMD control also has $\alpha_I = 1$ and $\beta_D = 0.5$).

Fig. 2a illustrates the throughput gains achieved by SSVP-SEL for PER ranging from 0 to 0.08. The protocol exhibits minimal sensitivity to the link errors across the last-hop wireless channel, exploiting AIAMD which reacts gently to error-induced loss. The selective control responds remarkably well to error rates as high as 0.08, utilizing a high fraction of the available bandwidth. Fig. 2b provides an insight to the operation of the selective control for PER = 0.04, showing the variation in SSVP-SEL's sending rate and in the $\overline{R}-n\overline{D}$ bound. On the oncurrence of packet loss, the instant value of $\overline{R}-n\overline{D}$ is compared to the currently measured rate, determining the appropriate recovery strategy. According to Fig. 2b, most packet drops are accurately interpreted as error-induced, since the measured rate often exceeds the $\overline{R}-n\overline{D}$ bound. Congestion events are less frequent, as the SSVP flow competes with FTP cross traffic.

On the contrary, the SSVP flow relying on AIMD control experiences considerable throughput degradation, which is evident throughout the PER range. SSVP's AIMD control invokes abrupt rate reductions, which may be dominant and destructive in terms of bandwidth utilization, in the presence of frequent and consecutive link errors. For example, with PER exceeding 0.05, the throughput rates achieved by SSVP are less than 50% of SSVP-SEL's corresponding throughput.

We also simulated a diverse range of MPEG flows (1-50) of (i) SSVP-SEL, (ii) TFRC, (iii) TCP Westwood+, and (iv) GAIMD, all competing with 10 FTP connections over TCP Reno. TCP Westwood+ is a recent version of Westwood [13] with a refined bandwidth estimation scheme. We do not provide results with MULTFRC [5], since it adjusts the number of connections arbitrarily, and we could not obtain results comparable with the other protocols. We set the bottleneck capacity to 10 Mbps and PER is adjusted at 0.02. We measured *Throughput*, *Fairness Index* and *Worst-Case Fairness* for the MPEG flows, and we additionally demonstrate statistics of delayed packets that compose an influencing factor for the perceptual quality (Figs. 3-4).

According to Fig. 3a, SSVP-SEL yields efficient bandwidth utilization outperforming the rest of the protocols, regardless of the level of link-multiplexing. Relying on AIAMD, SSVP-SEL is less susceptible to random wireless loss. Only congestion-induced loss enforces the protocol to reduce its transmission rate significantly in order to adapt to the prevailing network conditions. TFRC exhibits inferior throughput performance, since the protocol invokes wasteful congestion-oriented responses to all link errors. Likewise, GAIMD can not detect the nature of the error. In addition, its small increase rate does not allow a recovery after error-induced loss, and hence the sending rate remains diminished. Despite the improvements over the initial version of Westwood, Westwood+'s algorithm still does not obtain accurate estimates in heterogeneous environments, failing to achieve adequate utilization of the available bandwidth, especially at high contention.

In terms of media delivery, packet errors occasionally induce interruptions in the sending rate and the perceptual quality inevitably deteriorates. Fig. 3b illustrates that SSVP-SEL alleviates most of the impairments due to error-induced loss, maintaining an uninterrupted and smooth sending rate. TFRC's random downward adjustments occasionally induce oscillations in the sending rate, and subsequently perceptible delay variation for high link-multiplexing. Besides Westwood+'s tendency to overestimate the available bandwidth, the protocol slows down the transmission in response to the link errors. Consequently, the resulting transmission gaps induce interruptions in the stream playback, with the effect of jitter becoming evident to the end-user. GAIMD's performance on video delivery may as well frustrate the end-user. In dynamic environments with wireless errors, the protocol's congestion-oriented responses to all types of errors abolish the potential gains from a gentle decrease ratio (that could favor smoothness in a static and error-free network).

Fig. 4 demonstrates that fairness for SSVP-SEL is not compromised, if assessed either by the traditional *Fairness Index* or the *Worst-Case Fairness*. The AIMD-based responses during congestion, as well as the high increase rate (i.e. $\alpha_I = 1$) enforce the competing SSVP-SEL flows to converge fast to the fairness point.

V. Conclusions

We have proposed a selective rate control with history information to enhance media delivery in the presence of random wireless errors. AIAMD enables a sophisticated loss-recovery strategy combining the most desirable features of AIAD and AIMD, i.e. a graceful variation in the transmission rate and sensitivity to the onset of sudden congestion. Incorporating AIAMD into SSVP, we showed that the proposed mechanism is less susceptible to packet errors, utilizing wireless resources efficiently and achieving remarkable performance on media delivery.

References

[1] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", in *Proc. 15th IDCS*, Vancouver, Canada, May 1995.

[2] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", *ACM/IEEE Transactions on Networking*, 5(6), 1997, pp. 756-769.

[3] A. Capone, L. Fratta, and F. Martignon, "Bandwidth Estimation Schemes for TCP over Wireless Networks", *IEEE Transactions on Mobile Computing*, 3(2), 2004, pp. 129-143.

[4] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end Differentiation of Congestion and Wireless Losses", *IEEE/ACM Transactions on Networking*, 11(5), October 2003, pp. 703-717.

[5] M. Chen and A. Zakhor, "Rate Control for Streaming over Wireless", in *Proc. IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[6] D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks", *Journal of Computer Networks and ISDN*, 17(1), June 1989, pp. 1-14.

[7] A. Chockalingam, M. Zorzi, and V. Tralli, "Wireless TCP performance with link layer FEC/ARQ", in *Proc. IEEE ICC*, Vancouver, Canada, June 1999.

[8] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, August 2000.

[9] T. Goff, J. Moronski, D. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end Enhancement Mechanism for Mobile Environments", in *Proc. IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000.

[10] F. Hu and N. K. Sharma, "Enhancing Wireless Internet Performance", *IEEE Communications Surveys and Tutorials*, 4(1), 2002, pp. 2-15.

[11] S. Jin, L. Guo, I. Matta, and A. Bestavros, "A Spectrum of TCP-friendly Window-based Congestion Control Algorithms", IEEE/ACM Transactions on Networking, 11(3), June 2003, pp. 341-355.

[12] E. Kohler, M. Handley, and S. Floyd, Designing "DCCP: Congestion control without reliability", in *Proc. ACM SIGCOMM 2006*, Piza, Italy, September 2006.

[13] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", in *Proc. ACM MOBICOM '01*, Rome, Italy, July 2001.

[14] P. Papadimitriou and V. Tsaoussidis, "End-to-end Congestion Management for Real-Time Streaming Video over the Internet", in *Proc. 49th IEEE GLOBECOM*, San Francisco, USA, November 2006.

[15] P. Papadimitriou and V. Tsaoussidis, "Performance Evaluation of Real-Time Transport with Link-Layer Retransmissions in Wired/Wireless Networks", *Journal of Mobile Multimedia*, 2(4), December 2006, pp. 327-343.

[16] R. Rejaie, M. Handley, and D. Estrin, "RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet", in *Proc. IEEE INFOCOM 1999*, New York, USA, March 1999.

[17] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks", in *Proc. ACM MOBICOM '99*, Seattle, USA, August 1999.

[18] V. Tsaoussidis and H. Badr, "TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains", in *Proc. 8th IEEE ICNP*, Osaka, Japan, November 2000.

[19] V. Tsaoussidis and I. Matta, "Open issues on TCP for Mobile Computing", *Journal of Wireless Communications and Mobile Computing*, 2(1), February 2002, pp. 3-20.

[20] Y. R. Yang, M. S. Kim, and S. S. Lam, "Transient Behaviors of TCP-friendly Congestion Control Protocols", in *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA, April 2001.

[21] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control", in *Proc. 8th IEEE ICNP*, Osaka, Japan, November 2000.