Real-time Streaming over Wireless Links: A Comparative Study

Guang Yang, Ling-Jyh Chen, Tony Sun, Mario Gerla and M. Y. Sanadidi*
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095, USA
{yangg, cclljj, tonysun, gerla, medy}@cs.ucla.edu

Abstract

Real-time streaming over wireless links is challenging. The streaming protocol must be efficient and robust to random wireless loss, fair to itself, and friendly to legacy TCP. Various solutions have been proposed in the literature, among which the Video Transport Protocol (VTP), TFRC Wireless, and MULTFRC are end-to-end representatives. In this paper we provide an in-depth comparison on the performance of VTP, TFRC Wireless, and MULTFRC in various wireless scenarios. The results show that VTP and TFRC Wireless both perform well and deliver similar performance, with VTP exhibiting greater efficiency and smoothness in presence of heavy errors. In contrast, MULTFRC performs less satisfactorily, as it experiences large rate fluctuation and slow convergence caused by the frequent changes in the number of simultaneous connections.

1 Introduction

Real-time video streaming is an important application on the Internet. Unlike conventional applications such as file transfer and email, real-time streaming requires a minimum bandwidth as well as stringent bounds on the end-to-end delay and jitter. TCP, the most widely used transport protocol, can be applied to streaming under certain conditions [7]. However, since the instantaneous sending rate of TCP changes drastically with time, buffering is needed [10], which causes a startup delay of seconds or longer. For Video-on-Demand (VoD) this delay is tolerable, but for real-time, especially interactive applications, e.g. video conferencing and online gaming, delay has to be very small [11]. Moreover, more and more portable wireless devices are now connected to the Internet. These devices are often small and inexpensive with limited computation and memory capabilities; large buffer capacity is not expected.

To address this problem, several cross layer and end-to-end rate control mechanisms have been proposed. Cross layer approaches [6] usually require modifications to both end hosts and gateways. The deployment difficulties often render such approaches not as feasible as end-to-end solutions. TCP Friendly Rate Control (TFRC) [4] is a popular end-to-end scheme that attempts to match the long-term throughput of TCP[1] while keeping a smoother sending rate. Similar to TCP, TFRC assumes every packet loss as congestion-induced and reacts by cutting the sending rate. In wireless networks where packet loss can also be due to signal attenuation, fading, scattering, interference, or mobility [9],TFRC is known to have reduced efficiency [14].

TFRC wireless [2] and MULTFRC [3] are recently proposed extensions to TFRC that target to provide more efficient support against wireless loss. Alternatively, we have also proposed an end-to-end rate control mechanism for real-time streaming in the context of Video Transport Protocol (VTP) [13]. Unlike the equation based TFRC protocols, VTP achieved robust wireless performance via a combination of rate estimation and loss discrimination techniques. In this paper, we provide an in-depth comparison on the performance of VTP, TFRC Wireless, and MULTFRC in various wireless scenarios. The results show that VTP and TFRC Wireless both perform well and deliver similar performance, with VTP exhibiting greater efficiency and smoothness in presence of heavy errors. On the other hand, VTP and TFRC Wireless both significantly outperform MULTFRC, which experiences large rate fluctuation due to the frequent changes in the number of simultaneous connections.

The rest of the paper is organized as follows. Section 2 provides a brief review of the related work. Section 3 recapitulates the three protocols that we compare: VTP, TFRC

---

*This work was supported in part by Office of Naval Research (ONR) under Grant No. N00014-01-C-0016 and NSF under Grant No. CNS-0335302.

[1] The rate equation in TFRC is based on the behavior of TCP Reno. In this paper we use "TCP" and "TCP Reno" interchangeably unless explained otherwise.

Wireless and MULTFRC. Section 4 studies and compares the performance of these protocols via Ns-2 simulations. Finally Section 5 concludes the paper.

2 Related Work

Maintaining a smooth, efficient and TCP-friendly sending rate is important for rate control mechanisms in real-time streaming. Bansal et al. have proposed a class of *binomial algorithms* in [1] to provide TCP-friendly congestion/rate control. Two particular algorithms, IIAD and SQRT, are shown to be well-suited to streaming applications. However, binomial algorithms share with TCP the same inefficiency problem in presence of random loss and are not directly applicable to wireless networks.

TCP Friendly Rate Control [4] is another end-to-end mechanism aiming to achieve the equivalent long-term throughput of TCP with less short-term fluctuation. It measures the packet loss rate and Round-Trip Time (RTT) of a connection, then computes the appropriate sending rate using the equation:

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \qquad (1)$$

where T, s, R, p and t_{RTO} are the upper bound of the sending rate, packet size, RTT, loss rate and Retransmit Time Out value (RTO), respectively. TFRC also suffers from random loss and achieves very low throughput in error-prone wireless networks.

There also exist solutions that require assistance from lower layers. E.g., Explicit Congestion Notification (ECN) [8] is a general method where congested gateways stamp packets with a designated bit in the packet header. ECN gives end hosts the capability of distinguishing between random and congestion loss. ECN requires assistance from intermediate gateways; the cost of implementation has been a major deployment limitation.

Rate Control Scheme (RCS), proposed in [12], also requires modifications at gateways. RCS uses low-priority dummy packets to probe the available bandwidth on the path. It is able to estimate an admissible rate and deal with errors efficiently. However, all intermediate gateways must implement a multiple-priority mechanism with a low priority level for dummy packets. This feature is currently not available on the Internet.

3 VTP, TFRC Wireless and MULTFRC

3.1 VTP

The Video Transport Protocol (VTP) [13] is an experimental protocol with a new end-to-end rate control mechanism designed specifically for real-time streaming in wireless networks. It relies on two important techniques, namely the Achieved Rate (AR) estimation and Loss Discrimination Algorithm (LDA). AR is the rate that the sender has successfully pushed through the bottleneck. It is the rate that the receiver measures, plus the fraction corresponding to packet loss at the exit of the bottleneck due to errors. For the LDA, we use an end-to-end scheme called Spike that monitors the RTT and switches between a congestion and an error state. Please refer to [13] for more information.

VTP rate control is based on the analysis of TCP instantaneous sending rate. Similar to TCP, VTP linearly probes the available bandwidth until congestion is detected. VTP does not perform multiplicative decrease, instead it reduces the sending rate to AR. Figure 1 illustrates the rate control mechanism of VTP. VTP avoids the drastic rate reductions for better smoothness, at the same time mimicking the TCP behavior and maintaining the same *average* rate. In contrast to the highly fluctuating TCP rate, VTP reduces its rate by a lesser amount but keeps this reduced rate for longer.

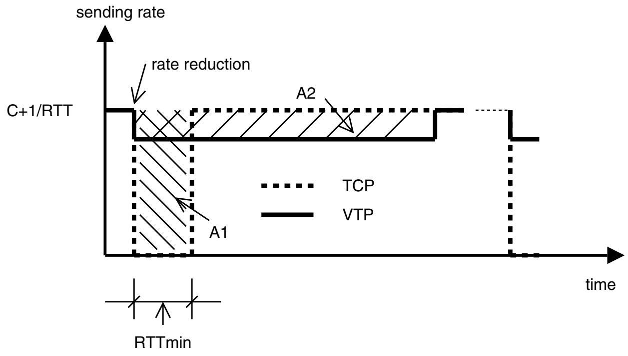

Figure 1. Comparison of the instantaneous sending rate between TCP and VTP.

Reflected in Figure 1, the two shaded areas $A1$ and $A2$ must be equal. When VTP has given up the same amount of data transmission as TCP would in the same situation, it enters congestion avoidance. VTP defines an "equivalent window", $ewnd$, as the number of packets transmitted by the sender during one RTT. $Ewnd$ increases at the pace of 1 packet/RTT and is used to compute the sending rate.

3.2 TFRC Wireless

Cen et al. have also added an end-to-end LDA to TFRC to improve its efficiency in presence of wireless errors [2]. The original purpose of [2] was to compare and evaluate different LDAs in various wireless topologies, using TFRC as a test protocol. TFRC Wireless then became a valid protocol of its own.

The operation of TFRC Wireless is straightforward. In the original TFRC Equation (1), p includes both congestion and random loss as TFRC does not distinguish between them. Lack of loss discrimination makes the performance of TFRC drop sharply when random loss is frequent. Equipped with the LDA, TFRC Wireless recognizes a congestion-induced loss and, in such case, operates exactly the same way as the original TFRC mechanism. If the loss is induced by random errors, it is discounted in the calculation of p. Thus wireless loss, when properly detected, has no impact on the TFRC Wireless rate control.

3.3 MULTFRC

Chen et al. have proposed in [3] to create multiple simultaneous TFRC connections on the same path when a single connection cannot fully utilize the bandwidth. This extension, called MULTFRC, is strictly end-to-end and requires no modifications to the Internet infrastructure. On the other hand, MULTFRC requires more resource to manage multiple connections. The sender must also split the data across multiple connections and the receiver must then reassemble the data chunks. This adds overhead to the scheme.

Similar to many other schemes, MULTFRC monitors the RTT to decide whether to change the number of TFRC connections. If the current RTT is relatively large, the bottleneck link is already well-utilized, thus the number of connections must decrease to avoid congestion. Otherwise the bottleneck is under-utilized; the number of connections should increase to improve bandwidth utilization.

4 Performance Comparison

In this section, we compare VTP, TFRC Wireless and MULTFRC[2] via Ns-2 simulations and evaluate their properties of efficiency, fairness and friendliness, in both error-free and error-prone scenarios. We also investigate the impact of queue buffer size on the performance of these protocols.

4.1 Simulation Setup

The topology in Figure 2, representing a mixed wired-cum-wireless scenario, is used for simulations in this section. The Internet segment is abstracted as a set of error-free links. The wireless segment, e.g. a WLAN, is abstracted as a shared error-prone link and also the bottleneck in the system. Note that for simplicity we do not simulate PHY and MAC overhead of the wireless segment and assume a full 11 Mbps capacity. The bottleneck queue is drop-tail with a buffer size of 99 packets, equal to the pipe size.

The foreground traffic is directed from the Internet servers (left) to the wireless clients (right). To better study the behavior of target protocols, no cross traffic is included in the simulations, though we will briefly discuss this issue at the end of the paper.

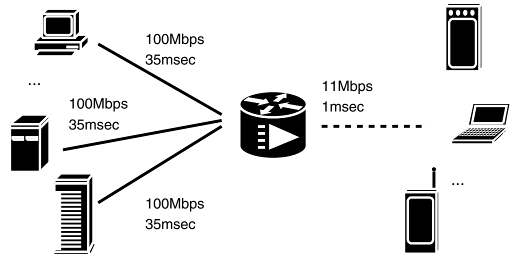

Figure 2. Simulation setup.

4.2 Efficiency against Errors

First we perform solo tests to assess the efficiency of these protocols against random errors. In each run, a single connection of VTP, TFRC Wireless or MULTFRC is set up from a server to a client. Different error rates are tested at the wireless link. The original TFRC is also included in the tests as a reference.

Figure 3 compares the sending rates of the four protocols with packet error rates of 0%, 1% and 5%, respectively. In the 0% error case, VTP and TFRC Wireless are efficient and smooth; their behavior is difficult to distinguish from the original TFRC. MULTFRC, on the other hand, has a much more fluctuating rate. We will explain this shortly.

As the error rate increases to 1% and 5%, VTP and TFRC Wireless are still capable of keeping the bandwidth utilization high, though the startup delays are slightly longer. This is largely due to the fact that the LDA needs more training. MULTFRC works comparably under the 1% error case; under 5% errors its sending rate increases to the available bandwidth very slowly, with more fluctuation as well. All three protocols, however, demonstrate better efficiency than the original TFRC. This is not a surprise since all of them are equipped with tools to handle wireless loss.

To better understand the behavior of MULTFRC, we plot the number of simultaneous TFRC connections in one MULTFRC "super-connection". Intuitively, the higher the error rate the more TFRC connections MULTFRC should create. This is confirmed in Figure 4.

Figure 4 also answers a couple of questions. First, in the 0% error case, the number of TFRC connections changes back and forth frequently between 1 and 2. Since creating or destroying a TFRC connection immediately increases or

---

[2] TFRC Wireless is implemented in Ns-2.1b8a, VTP and MULTFRC Ns-2 are implemented in Ns-2.26. All are obtained from original authors, respectively.

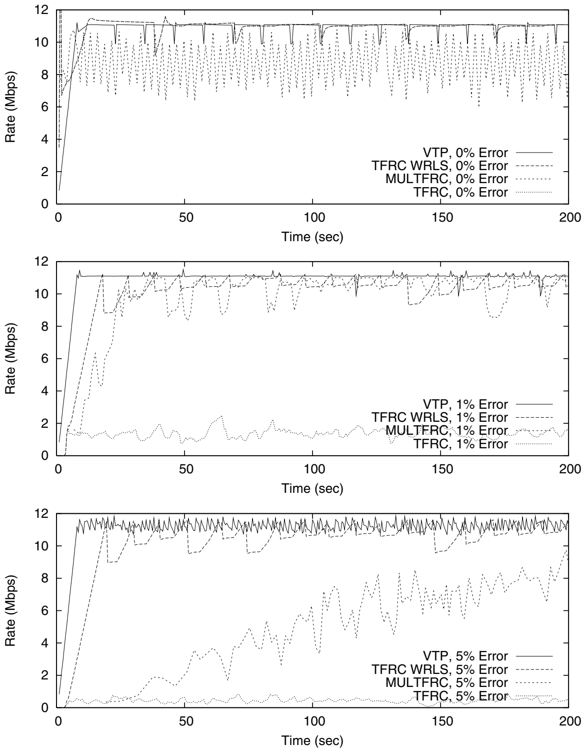

Figure 3. Rate comparison of VTP, TFRC Wireless and MULTFRC under different error rates.

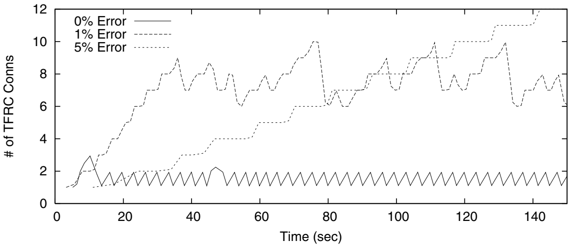

Figure 4. Number of TFRC connections in MULTFRC under different error rates.

decreases the overall instantaneous rate, the overall MULTFRC rate fluctuates accordingly. Secondly, in the 5% error case, the number of TFRC connections continues to grow during the 150 seconds of simulation time as shown. This is because the high error rate forces the throughput of each TFRC connection to be very low, so a large number of connections are needed for MULTFRC to utilize the bandwidth efficiently. However, the IIAD algorithm in MULTFRC prevents abrupt increase in the number of connections. In other words, MULTFRC needs a long time to converge to the steady state. This explains why the rate of MULTFRC keeps increasing during the simulation. We have observed that the number of connections in MULTFRC does eventually converge when the simulation time is longer, but the slow convergence speed is undesirable for many streaming applications of short to medium durations.

4.3 Intra-protocol Fairness

We then compare the *fairness* property of VTP, TFRC Wireless and MULTFRC. By fairness we refer to the relation between connections of the same protocol. We run two VTP, TFRC Wireless or MULTFRC connections simultaneously in this set of simulations. The wireless link (open medium) is shared by the connections.

In [13] we have shown that VTP maintains good fairness between flows when the error rate is 0%, 1% or 5%. Therefore we are more focused on TFRC Wireless and MULTFRC in this section and compare the results to VTP. We first look at the case with 0% errors. In this case, the LDA in TFRC Wireless is accurate [2]; TFRC Wireless behaves virtually the same as the original TFRC and should maintain the same fairness property. Our simulation confirms this as shown in Figure 5. MULTFRC, on the other hand, still exhibits more fluctuation. Again this is due to the frequent changes in the number of simultaneous TFRC connections. Throughput of one MULTFRC connection may be temporarily much higher than the other.

We skip the 1% error case for conciseness and move to the more informative 5% error case as shown in Figure 6. TFRC Wireless is still able to maintain the fairness, but its efficiency drops. Compared to the results in [13], this indicates that although VTP and TFRC Wireless both keep good fairness, VTP achieves better efficiency when the error rate is high. MULTFRC again needs to create a large number of connections before high utilization is achieved; the "ramp up" speed is much slower.

4.4 Inter-protocol Opportunistic Friendliness to TCP

Now we evaluate the *opportunistic friendliness* of VTP, TFRC Wireless and MULTFRC to legacy TCP. Simply speaking, a new protocol *NP* is opportunistically friendly to TCP if TCP connections obtain no less throughput when coexisting with *NP*, compared to the throughput that they would achieve if all connections were TCP.

In each of the simulation runs, two connections are set up: one is VTP, TFRC Wireless or MULTFRC; the other is TCP. We also run two TCP connections as this is required to evaluate opportunistic friendliness. For conciseness, we summarize the results of the 0% and 1% error cases before moving to the more interesting 5% error case. With 0%

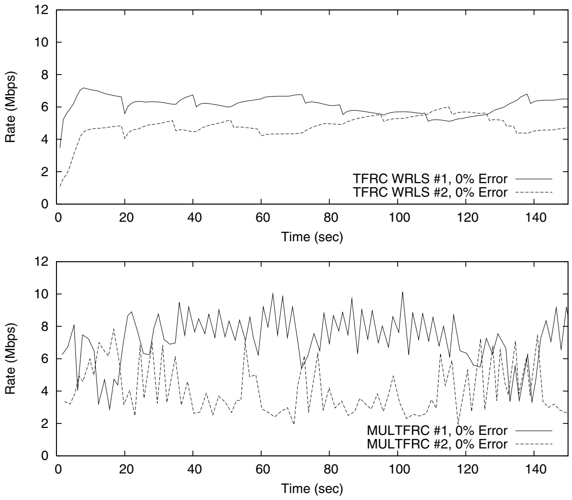

Figure 5. Fairness between two TFRC Wireless or MULTFRC connections, 0% errors.

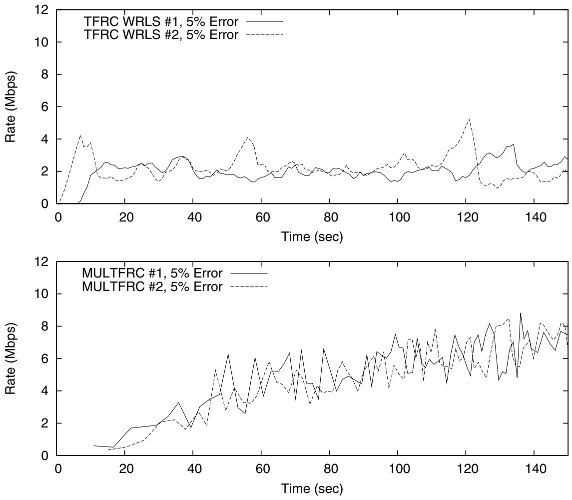

Figure 6. Fairness between two TFRC Wireless or MULTFRC connections, 5% errors.

errors, all streaming protocols approximately equally share the bandwidth with the TCP connection, although MULTFRC exhibits more fluctuation as before. With 1% errors, since TCP efficiency drops, three streaming protocols are able to utilize the residual bandwidth, a reflection of the opportunistic friendliness.

We present the 5% error case in Figure 7. TCP is extremely inefficient under such a condition, so the streaming protocol should take over the bandwidth. Both VTP and TFRC Wireless perform well in this case. VTP demonstrates a bit more high-frequency fluctuation, while TFRC Wireless has a larger variation in the rate. MULTFRC is unable to utilize the bandwidth quickly since it has to wait until a large number of TFRC connections are created.

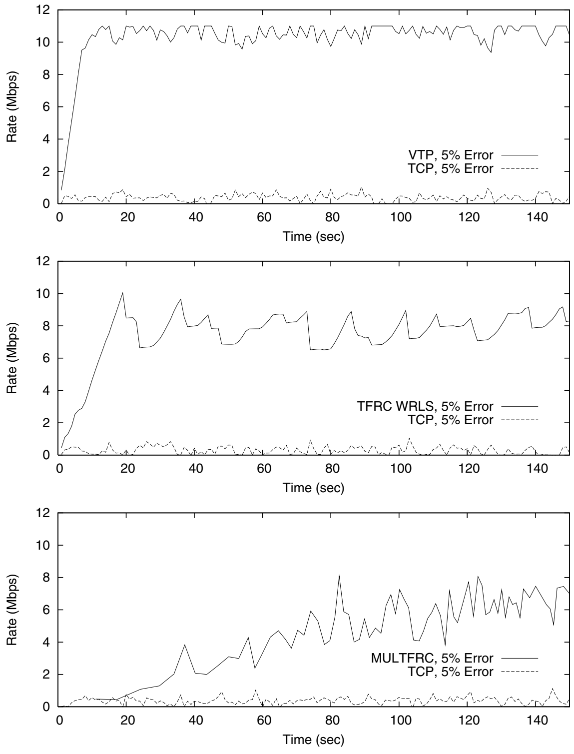

Figure 7. Opportunistic friendliness, 5% errors: VTP/TCP (top), TFRC Wireless/TCP (middle), MULTFRC/TCP (bottom).

4.5 Impact of Buffer Size

So far, all simulations assume the queue buffer size is equal to the pipe size. This does not always hold in a real network, though. To assess the performance of VTP, TFRC Wireless and MULTFRC with various buffer sizes, we repeat the simulations on opportunistic friendliness but leave the buffer size as a variable.

We test three different buffer sizes, i.e., 1/3, 1x and 3x of the pipe size, representing the small, medium and large buffer cases, respectively. In our simulations, we have found that the stability and performance of MULTFRC are very sensitive to the buffer size. Its parameters in the Ns-2 implementation are fine-turned; changing the buffer size requires to empirically find another set of appropriate values. Thus we postpone showing the results on MULTFRC until a more stable version is available.

	1/3 pipe size	pipe size	3x pipe size
VTP	6.20	5.84	4.81
TCP	3.85	4.51	4.84
TFRC WRLS	4.74	5.40	3.96
TCP	5.30	5.10	6.20

Table 1. Impact of buffer size, with 0% errors. Numbers (except ratios) are average rates in Mbps.

Table 1 shows the average sending rates of VTP/TCP or TFRC Wireless/TCP when no errors are present. VTP tends to be more aggressive than TCP when the buffer size is small; as the buffer grows, aggressiveness of VTP diminishes. This agrees with our previous study that small buffer favors AR-based schemes while large buffer favors TCP [5].

We have also performed simulations with varied buffer sizes where errors are imposed at the wireless link. In this case, the buffer size has marginal impact on the performance, since TCP loses efficiency by itself. Both protocols maintain opportunistic friendliness to legacy TCP by picking up the bandwidth that would be left unused.

5 Discussion and Conclusion

In this paper we have compared the performance of three streaming protocols, namely VTP, TFRC Wireless and MULTFRC. These protocols are designed to provide efficient and smooth rate control for real-time streaming in wireless networks. VTP, an experimental protocol equipped with a new end-to-end rate control mechanism, excels in all tested error-free and error-prone scenarios. TFRC Wireless, a TFRC extension with the capability of distinguishing between congestion and wireless loss, achieves performance comparable to VTP except in a few cases where it is a bit less efficient. MULTFRC extends the original TFRC by creating multiple TFRC connections to achieve better efficiency. However, frequent increases/decreases in the number of simultaneous connections trigger a fluctuating behavior, and its convergence speed to the “fair share” is slow. Moreover, the Ns-2 implementation of MULTFRC is too sensitive to the network parameters and exhibits instability in certain parameter ranges.

To summarize, VTP and TFRC Wireless are two promising schemes for real-time wireless streaming. Similarities in the performance of these protocols should be no surprise, since both VTP and TFRC Wireless try to mimic the TCP behavior in zero random loss conditions. Moreover, VTP and TFRC Wireless are equipped with similar LDAs which make them equally insensitive to random loss. Therefore, although VTP and TFRC Wireless started from different aspects, they share the same idea of mimicking TCP while ignoring wireless loss, which leads to close performance.

In the future, we plan to compare the two winning protocols in more detail, especially with respect to implementation complexity and robustness to different traffic patterns and mixes.

References

[1] D. Bansal and H. Balakrishnan. Binomial congestion control algorithms. In *Proceedings of the IEEE Infocom*, Anchorage, AK, Apr. 2001. IEEE.

[2] S. Cen, P. Cosman, and G. Voelker. End-to-end differentiation of congestion and wireless losses. *IEEE/ACM Transactions on Networking*, 11(5):703–717, Oct. 2003.

[3] M. Chen and A. Zakhor. Rate control for streaming video over wireless. In *Proceedings of the IEEE Infocom*, Hong Kong, China, Mar. 2004. IEEE.

[4] S. Floyd. Congestion control principles. *RFC 2914*, Sept. 2000.

[5] M. Gerla, B. K. F. Ng, M. Y. Sanadidi, M. Valla, and R. Wang. Tcp westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs. *Computer Communications*, 27(1):41–58, Jan. 2004.

[6] M. Kazantzidis. *Adaptive Wireless Multimedia*. PhD thesis, University of California, Los Angeles, Los Angeles, CA, Apr. 2002.

[7] P. Mehra and A. Zakhor. Tcp-based video streaming using receiver-driven bandwidth sharing. In *Proceedings of the International Packet Video Workshop*, Nantes, France, Apr. 2003.

[8] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ecn) to ip. *RFC 3168*, Sept. 2001.

[9] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, 1996.

[10] R. Rejaie, M. Handley, and D. Estrin. Quality adaptation for congestion controlled video playback over the internet. In *Proceedings of the ACM SIGCOMM*, Cambridge, MA, Aug. 1999. ACM.

[11] M. Sun and A. Reibman. *Compressed Video over Networks*. Marcel Dekker, Inc., New York, NY, 2001.

[12] J. Tang, G. Morabito, I. Akyildiz, and M. Johnson. Rcs: A rate control scheme for real-time traffic in networks with high bandwidth-delay products and high bit error rates. In *Proceedings of the IEEE Infocom*, Anchorage, AK, Apr. 2001. IEEE.

[13] G. Yang, M. Gerla, and M. Y. Sanadidi. Adaptive video streaming in presence of wireless errors. In *Proceedings of the IPIF/IEEE MMNS*, San Diego, CA, Oct. 2004. Springer-Verlag.

[14] Y. Yang, M. Kim, and S. Lam. Transient behaviors of tcp-friendly congestion control protocols. In *Proceedings of the IEEE Infocom*, Anchorage, AK, Apr. 2001. IEEE.