Limitations of Equation-based Congestion Control in Mobile Ad Hoc Networks

Kai Chen and Klara Nahrstedt
Computer Science Department
University of Illinois at Urbana-Champaign
Urbana, IL 61801, U.S.A.
Email: {kaichen,klara}@cs.uiuc.edu

***Abstract*— Equation-based congestion control has been a promising alternative to TCP for real-time multimedia streaming over the Internet. However, its behavior remains unknown in the mobile ad hoc wireless network (MANET) domain. In this paper, we study the behavior of TFRC (TCP Friendly Rate Control [1], [2]) over a wide range of MANET scenarios, in terms of throughput fairness and smoothness. Our result shows that while TFRC is able to maintain throughput smoothness in MANET, it obtains less throughput than the competing TCP flows (i.e., being conservative). We analyze several factors contributing to TFRC's conservative behavior in MANET, many of which are inherent to the MANET network. We also show that TFRC's conservative behavior cannot be completely corrected by tuning its loss event interval estimator. Our study shows the limitations of applying TFRC to the MANET domain, and reveals some fundamental difficulties in doing so.**

At the same time, our study in this paper opens up the question of how to properly perform multimedia streaming over MANET. To this end, we propose an alternative scheme (called EXACT-AA) based on router's explicit rate signaling and application's adaptation policies. We demonstrate the feasibility of our scheme using an audio streaming application over a real MANET test-bed.

***Index Terms*— Equation-based congestion control, TFRC, fairness, smoothness, mobile ad hoc networks (MANET).**

I. Introduction

Recent years have seen a stream of TCP-friendly congestion control mechanisms designed for the Internet [1]–[5]. They are driven by the need of multimedia streaming over the network, which requires smooth rate adaptation, instead of TCP's abrupt "cut-half" rate change policy. At the same time, they attempt to maintain long-term throughput fairness with other competing TCP flows in the network, i.e., their long-term throughput should approximately equal to that of a TCP flow under the same network condition. Among the class of TCP-friendly congestion control mechanisms, the TCP equation-based approach has been one of the most well-studied algorithm [1]–[3], [6]–[8]. It relies on a "TCP throughput equation" which captures the TCP throughput over a network path with certain loss rate and round-trip time (RTT). Past studies have shown that the TCP equation is able to achieve reasonable fairness with competing TCP flows under a wide range of traffic conditions in wireline networks [6], [7]. Real experiments over the Internet also suggest that it is safe to be deployed [1]. In fact, the protocol that implements the TCP-equation based approach, TFRC (TCP Friendly Rate Control), has recently become a standard RFC [2].

Now we shift our attention from Internet to a mobile ad hoc network (MANET). In MANET, each node is free to move about, creating not only fluctuating wireless link bandwidth, but also link breakage, route breakage and dynamic routing. Currently TCP remains the de facto standard for congestion control in MANET (despite its many well-known deficiencies in this environment), simply because of its wide acceptance and deployment over the Internet. With the emerging need of multimedia streaming over MANET, equation-based congestion control is likely to find its way into MANET as well, for example, by reusing the same software that has been developed for the Internet.

However, the behavior of equation-based congestion control (TFRC) is very much unknown in MANET where the degrees of network dynamics are far more diverse than those in wireline networks. For instance, wireless link's bandwidth can vary greatly in very small timescale, due to the randomness in channel contention and signal fading. Packet loss can occur due to congestion-related queuing loss, wireless-related random loss, and mobility-related routing loss. Under this environment, it is unclear whether TFRC will be able to compete fairly with TCP, and if not, what are the factors that contribute to such behavior.

In this paper, we study the behavior of TFRC in MANET. Our finding indicates that, while TFRC is able to maintain smooth rate change, its throughput is often "beaten" down by competing TCP flows to a certain degree, especially under heavy background traffic and dynamic topology conditions. To explain TFRC's conservative behavior, we analyze several factors including loss rate discrepancy, inaccuracy of loss rate prediction, and lack of auto-correlation in MANET's loss process. We also explore TFRC's response to the tuning of its loss event interval estimator, and show that its conservative behavior cannot be completely corrected. Our study shows the limitations of applying TFRC to the MANET domain, and reveals some fundamental difficulties in doing so.

Our findings in this paper also open up the question of how to properly perform multimedia streaming over MANET. To this end, we propose an alternative scheme (called EXACT-AA) based on router's explicit rate signaling and application's adaptation policies. We demonstrate the feasibility of our scheme using an audio streaming application over a real MANET test-bed.

The rest of the paper is organized as follows. In Section II we discuss background and related work of TFRC. In Section III, we study TFRC's behavior in MANET, and explain the factors that lead to such behavior in Section IV. In Section V we explore parameter tuning of TFRC. In Section VI we propose an explicit rate signaling scheme for multimedia streaming and present its test-bed experiment results. We then conclude the paper in Section VII.

II. Background and Related Work

A. Background of TFRC

TFRC is a protocol that implements equation-based congestion control. [1] In TFRC, the receiver measures the *loss event rate* (i.e., loss rate) and feeds this information to the sender. The sender uses the feedback messages to measure the RTT, and then inputs the loss rate and RTT to a TCP throughput equation to compute its acceptable transmission rate.

The core of TFRC is the TCP throughput equation, which is a slightly simplified version of the equation from [3]:

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

where X is the transmission rate in bytes/sec, s is the packet size in bytes, R is the RTT in seconds, p is the loss event rate between 0 and 1.0, t_{RTO} is the TCP re-transmission timeout value in seconds, and b is the number of packets acknowledged by a single TCP acknowledgment. This equation can be further simplified by setting $t_{RTO} = max(4R, 1.0)$, since RTO should be at least 1.0 second as recommended in RFC 2988. In practice, b is usually set to 1, to match the behavior of many TCP receiver which acknowledges every data packet it receives. Other parameters, s (packet size), p (loss event rate) and R (RTT), need to be measured.

The measurement of the loss event rate receives the most attention, because it should track smoothly in a steady loss environment, and should respond strongly to persistent loss [1]. To this end, TFRC recommends using the average *loss interval*, i.e., the number of packets between loss events, to measure the loss event rate, as follows:

$$p_n = 1/\hat{\theta}_n \quad and \quad \hat{\theta}_n = \sum_{l=1}^{L} w_l \theta_{n-l} \quad (2)$$

where $\hat{\theta}_n$ is the weighted average of the loss intervals at the n-th loss event, θ_{n-l} (l = 1 to L) is the history of the latest L loss intervals, and w_l (l = 1 to L, $\sum_{l=1}^{L} w_l = 1$) is the set of weights used in the estimation. In TFRC, the history is chosen to be $L = 8$, and their eights (w_l, l = 1 to 8) are $\frac{1}{6} \times \{1.0, 1.0, 1.0, 1.0, 0.8, 0.6, 0.4, 0.2\}$. This arrangement gives equal weights to the recent $L/2$ samples, and linearly decreasing weights after that.

B. Related Work

The TCP throughput equation (Equation (1)) was first derived by Padhye et al. [3], where a deterministic network loss process is assumed, i.e., network's loss rate is *constant*. This equation was later adopted by TFRC and has been extensively evaluated by Floyd et al. [1], which shows approximate fairness with TCP over a wide range of simulated wireline networks and over the real Internet.

Bansal et al. [6] and Yang et al. [7] studied the dynamic behavior of TFRC in a wireline network with time-varying background traffic. They found that TFRC may not always get its equitable share when the network condition changes dynamically, and it may incur higher packet loss rate than TCP due to its slow response to network congestion. In a similar study performed on TCP with different responsive parameters (i.e., GAIMD), Zhang and Tsaoussidis [9] observed that a less responsive TCP flow may lose throughput to a more responsive one, especially when the network has high transient error rates.

[1] In this paper, we use TFRC to refer to both the equation-based congestion control mechanism, as well as the protocol that implements such mechanism. Its meaning should be clear from the context.

Vijnovic and LeBoudec [8] studied the long-term behavior of an adaptive source using the TCP throughput equation. They found that if the network loss process is deterministic, the equation-based adaptive source achieves comparable long-term throughput with TCP; however, if the loss process is random, the long-term throughput guided by the TCP equation may not be TCP-friendly, due to the non-linearity of the equation. Especially, they showed that if the loss event intervals of the network are not correlated or negatively correlated, the equation-based source will under-shoot the long-term throughput of TCP, i.e., being *systematically* conservative, and the degree of conservativeness depends on the variation of the estimated loss event intervals.

III. Behavior of TFRC in MANET

In this section we study the behavior of TFRC in terms of long-term and short-term fairness and smoothness, under various static and dynamic MANET topologies and with different levels of background traffic.

A. Simulation Network and Parameters

We consider two types of MANET topologies: static and dynamic. In static topology, we consider a chain that consists of 2 to 7 stationary nodes, which provides a controlled environment where TFRC can be evaluated over a path with increasing number of hops. In dynamic topology, two scenarios are considered: a small 600×600m network with 50 nodes (where a path has 1 to 4 hops), and a larger 1500×300m network with 60 nodes (where a path has 1 to 7 hops). In both scenarios, random way-point mobility is used with maximum speed of 10 m/s and pause time of 0 seconds, and the network is not partitioned at any time. We hope to use these scenarios (6 static and 2 dynamic) to represent the spectrum of MANET topologies.

In each scenario, 10 TCP-SACK flows and 10 TFRC flows are created to compete with each other over the same path. [2] In the static chain scenarios, TCP and TFRC flows run from one end of the chain to the other. In the dynamic scenarios, a pair of nodes are randomly chosen to be the sender and receiver of the TCP and TFRC flows. Since they travel through the same path, they should encounter the same network conditions. Sharing a path also shields the potential discrepancy of route discovery for different paths. We use Dynamic Source Routing (DSR [11]) as the underlying routing protocol.

[2]There are many existing studies in enhancing TCP performance in MANET, e.g. TCP-ELFN [10]. Similar techniques may be applied to TFRC as well. In this paper, we only focus on the behaviors of unmodified TCP and TFRC flows.

Background traffic consists of non-adaptive CBR flows to create consistent but varying levels of congestion within the network. In the chain scenarios, a CBR flow is created with various data rates, from one end of the chain to the other. In the dynamic scenarios, in order to spread out the background traffic across the network, 10 CBR flows are created each between a pair of randomly selected nodes. In order to avoid stalling the TCP and TFRC flows, we have carefully selected different levels of CBR data rates for each of the simulated scenarios, such that the non-adaptive CBR traffic does not overflood the whole network.

We keep most of TFRC's default settings in the ns-2 (2.1b9a) simulator, which mostly corresponds to the parameters suggested in [1], [2]. We use the same data packet size (1000 bytes) for TCP and TFRC, so that we can also compare their throughputs by the number of data packets. Each simulation run lasts for 1000 seconds.

B. Fairness

We consider both the *long-term* and *short-term* fairness between TCP and TFRC. To evaluate long-term fairness, we obtain the *average throughput* of all the TCP (or TFRC) flows over the entire course of their simulation (1000 seconds), and then normalize TFRC's average throughput again TCP's (so that TCP's throughput is always one). Figure 1 shows the results under different simulated topologies with various levels of background traffic. Three observations can be made from the figure: 1) TFRC shows conservative behavior over all simulated scenarios; 2) TFRC is generally more conservative with heavier background traffic and in a dynamic topology; and 3) overall TFRC obtains 0.2 to 0.8 the throughput of TCP. Therefore, although TFRC can be used in situations where strict throughput fairness is not a major concern, it consistently possesses conservative behavior in MANET.

To evaluate short-term fairness, we use the *average short-term throughput* of all the TCP (or TFRC) flows, over every 10-second time interval. We choose the 10-second interval to measure short-term throughput, because the RTT over a long path (e.g. over 5 hops) in MANET may take as long as several seconds. Therefore, the 10-second interval we use is within the time-scale of several RTTs, which is the time-scale usually used in measuring short-term fairness. For each MANET scenario, we choose a mid-level background traffic rate. Figure 2 plots the average 10-second throughputs of the TFRC and TCP flows, in four representative scenarios (others are omitted for brevity). Two observations can be made from this figure: 1) the short-term throughputs of

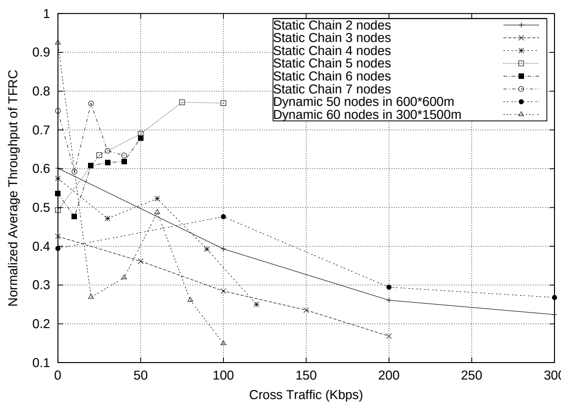


Fig. 1. Normalized long-term throughput of TFRC.

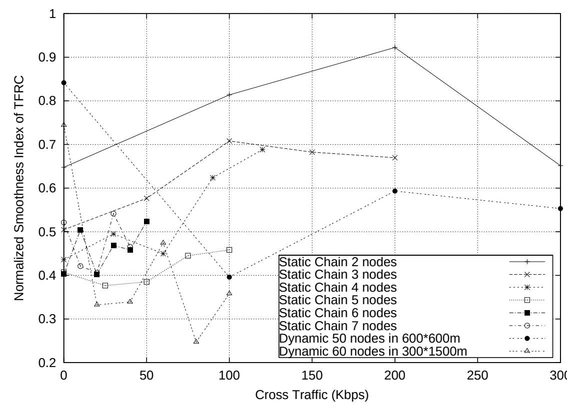


Fig. 3. Normalized smoothness index of TFRC.

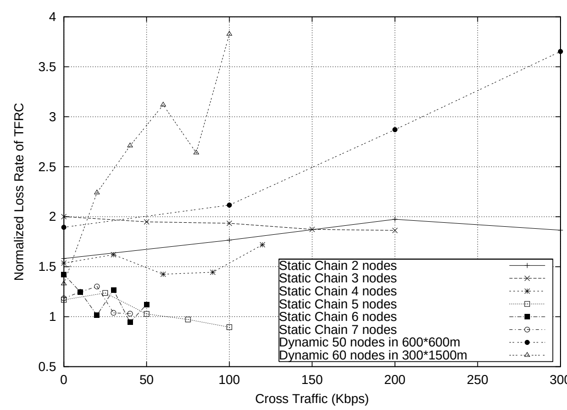


Fig. 4. Normalized loss rate of TFRC.

TCP and TFRC are very fluctuating, and in most cases, TFRC has less throughput than TCP; 2) TCP is more aggressive to increase its throughput when bandwidth becomes available, and more agile to reduce throughput when bandwidth becomes scarce. For instance, in Figure 2(d), TCP grows much faster than TFRC at time 20s, 100s, 210s, 320s and 900s, and it slows down quicker at time 90s, 200s, 450s and 960s. This slow response behavior of TFRC is same as in wireline networks.

C. Smoothness of rate change

Smooth rate change is an important feature of TFRC. Here we use a flow's *throughput-change ratio* between two consecutive time windows to measure its smoothness, as: $S_{i+1} = |r_{i+1} - r_i|/r_i$, where r_i is the average throughput over the i-th interval for that flow (each interval is 10 seconds). It can also be interpreted as a flow's throughput fluctuation over two consecutive time intervals. A flow's *smoothness index* is then defined as the average throughput-change ratio during its lifetime, as: $\bar{S} = (\sum_{i=1}^{N} S_i)/N$, where N the total number of time intervals during the simulation. [3] A smaller smoothness index indicates smoother throughput change for a flow.

Figure 3 shows the average smoothness index of all the TFRC flows, normalized against that of the TCP flows (so that the TCP flows' smoothness index is one). It shows that TFRC is able to maintain its smooth rate change over a wide range of MANET scenarios, and in most cases, TFRC's throughput fluctuates only 0.3 to 0.7 as much as TCP's.

In sum, TFRC consistently shows conservative behavior over both long-term and short-term, while it is able to maintain throughput smoothness, under a wide range of MANET topologies with varying levels of background traffic.

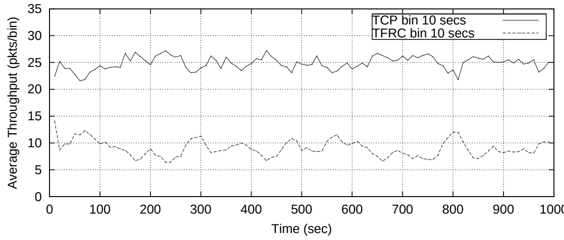
[3]Unlike using the coefficient of variation of a flow's short-term throughputs, for instance, in [7], our definition of the smoothness index captures the time serial of rate changes, whereas the coefficient of variation metric considers the short-term throughputs only as a set of samples without any relation in the time domain.

IV. Factors Contributing to TFRC's Conservative Behavior

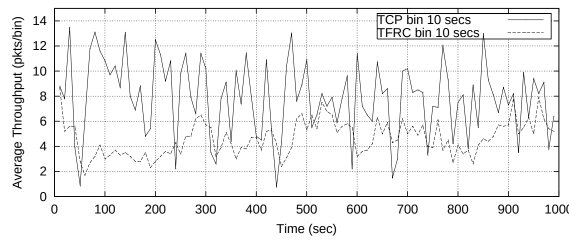
In this section, we study the factors that contribute to TFRC's conservative behavior in MANET.

A. TFRC may experience higher loss rate than TCP

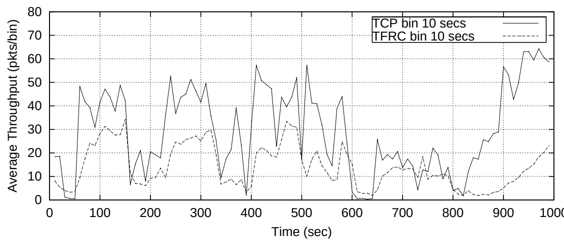
Under dynamic network conditions, loss rate experienced by TFRC flows may be higher than that by TCP flows, due to TFRC's slow response to network congestion. As a result, the larger loss rate experienced by a TFRC flow may drive down its throughput based on the TCP equation. Figure 4 shows the *average loss rate* experienced by all TFRC flows over the entire course of each simulation, normalized against that of the TCP flows. It shows that TFRC's loss rate is much larger than TCP's, especially in a dynamic network topology with heavy background traffic.



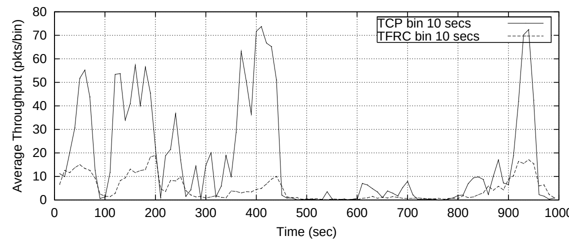
(a) Static 3-node Chain with background 50Kbps



(b) Static 7-node Chain with background 10Kbps



(c) Dynamic 600×600 Network with background 100Kbps



(d) Dynamic 1500×300 Network with background 20Kbps

Fig. 2. Short-term throughput of TCP and TFRC.

B. *TFRC's loss rate estimator is highly inaccurate*

TFRC relies on the recent history of loss event intervals to estimate the current loss event interval (equivalently the loss rate), using a weighted average estimator as in Equation (2). However, the estimation may not be accurate, due to many random factors in the network's loss process. We define a *prediction-error ratio* metric as: at the end of the i-th loss event interval, $E_i = |\theta_{predicted} - \theta_{true}|/\theta_{true}$, where $\theta_{predicted}$ is the predicted value for this interval, and θ_{true} is the true value. We then average the prediction-error ratios for each loss event interval, during a TFRC flow's lifetime, as: $\bar{E} = (\sum_i^N E_i)/N$, where N is the number of loss event intervals.

Figure 5 shows the average prediction-error ratio of all the TFRC flows. The result can be roughly divided into three groups: a) short-chain (2 to 4 nodes) with 70% to 80% error; b) long-chain (5 to 7 nodes) with 80% to 90% error; and c) dynamic network scenarios have highly varying and sometimes very high (i.e. over 100%) error ratio. Overall, this suggests that TFRC's loss event interval prediction is *highly* unreliable in MANET, and that the prediction is worse over a longer path or in a more dynamic topology.

TFRC's inaccuracy in predicting loss event interval can be attributed to a number of reasons: 1) highly varying packet losses due to dynamic wireless link bandwidth; and 2) some packet losses are wireless-medium or route-disruption related, and hence highly random and unpredictable. For instance, Figure 6 shows the measured 10-second averaged link bandwidth from node 1 to 2 in the 5-node chain scenario with 50Kbps background traffic (using the bandwidth measurement method in [12], [13]). Unlike wireline networks where a physical link's bandwidth is constant, in MANET, a wireless link's effective bandwidth is time-varying, depending on channel contention and signal fading. This MAC layer property is clearly unique in MANET, and we believe it is a fundamental difficulty in doing loss

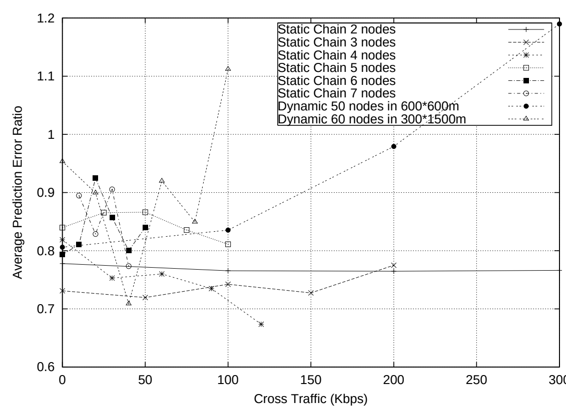


Fig. 5. TFRC's prediction error of loss event intervals.

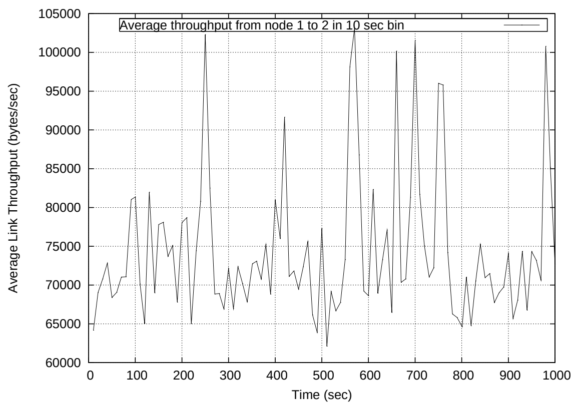


Fig. 6. Measured link bandwidth in 10-second interval.

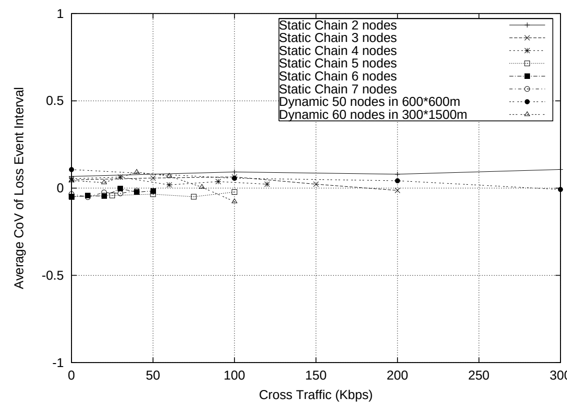


Fig. 7. Average $cov[\theta, \hat{\theta}]$ experienced by TFRC.

rate estimation in such network.

C. MANET's loss process shows little auto-correlation

To further understand the difficulty for TFRC to estimate the current loss event interval, we study the *covariance* (cov) of the estimated loss event interval ($\hat{\theta}$) and its true value (θ), experienced by a TFRC flow. [4] Since the loss event interval is estimated based on the weighted average of the past L intervals (as in Equation (2)), the covariance of the estimated interval and its true value can be computed as:

$$cov[\theta_n, \hat{\theta}_n] = \sum_{l=1}^{L} w_l cov[\theta_n, \theta_{n-l}], \quad (3)$$

where w_l is the same set of weights as in Equation (2). In other words, $cov[\theta, \hat{\theta}]$ depends only on the spectral property of the auto-covariance (with lags from 1 to L) of the loss event intervals. The loss event intervals should possess significant auto-correlation in order for TFRC to have an accurate prediction; otherwise it is impossible to do so no matter how the weights are chosen.

We compute the covariance of $\hat{\theta}$ and θ using auto-covariance of θ with lags $l = 1$ to 8. Figure 7 shows the result of $cov[\theta, \hat{\theta}]$ normalized into range [-1,1] (same as its cor) in our simulated scenarios. The small $cov[\theta, \hat{\theta}]$ in Figure 7 suggests that $\hat{\theta}$ is a bad estimator for θ, which helps to explain the large prediction error we have seen earlier. The auto-covariance of θ, not shown here, is also very small. That means MANET's loss process possesses little auto-correlation for its loss event intervals. Furthermore, as mentioned in Section II, the

[4]Recall that covariance of two random variables is defined by $cov[X, Y] = E[XY] - E[X]E[Y]$. Their statistical correlation $cor(X, Y)$ is defined by $cor(X, Y) = \frac{cov(X,Y)}{\sigma_x \sigma_y}$ where σ_x and σ_y are the standard deviations of X and Y, respectively. Note that $-1 \leq cor(X, Y) \leq 1$.

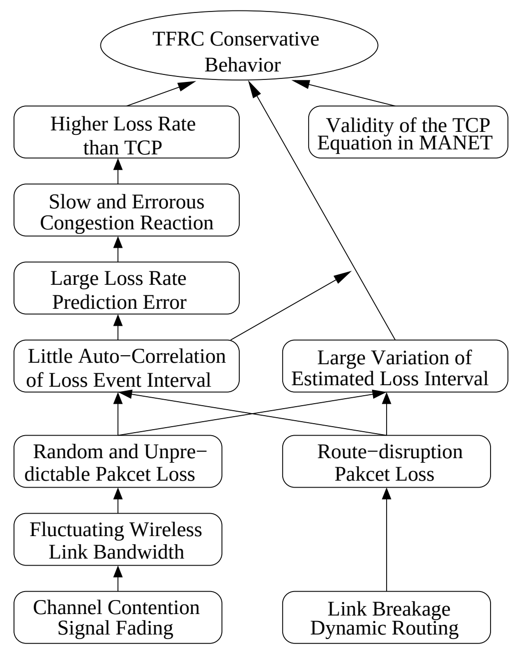


Fig. 8. Relation of various factors contributing to TFRC's conservative behavior in MANET.

lack of correlation between $\hat{\theta}$ and θ also contributes to the long-term "conservative" behavior of a TFRC-based source [8]. The lack of auto-correlation of MANET's loss event intervals shows another difficulty in applying TFRC into MANET.

We summarize the various factors contributing to TFRC's conservative behavior in Figure 8.

V. Parameter Tuning of TFRC's Loss Event Interval Estimator

By default, TFRC uses the recent 8 history intervals ($L = 8$ in Equation (2)) to estimate the current loss event interval. In this section we explore TFRC's fairness and smoothness behaviors by tuning parameter L in the equation.

The number of history samples determines not only the *responsiveness* of a TFRC flow, but also the *variation* of the estimated loss event intervals. Intuitively, including more history samples (larger L) makes TFRC *less* responsive to network condition, hence leading to better smoothness behavior. However, the effect on the long-term fairness between TFRC and TCP is less certain, because it must be determined by two counter-active factors. On one hand, when TFRC is less responsive to network condition, its loss rate may *increase*, making TFRC more conservative. On the other hand, using more history samples makes the variation of the estimated intervals *lower*, which in turn drives the TFRC control less conservative (according to Claim 1 in [8]). Therefore, the long-term fairness should be the combined effect of these two counter-active factors.

We pick two of our earlier MANET scenarios to explore TFRC's response to parameter tuning of L. Figure 9 shows the result of a 4-node static chain with 30Kbps background traffic. Two observations are evident. First, TFRC's rate change is smoother with the increase of history samples. Second, TFRC is more conservative when L is small (2 to 4), and it remains roughly unchanged when L becomes larger (8 to 128). Now let's look at the two factors that drive the long-term fairness of TFRC: 1) loss rate experienced by TFRC, and 2) the coefficient of variation of the estimated loss intervals. [5] Figure 9(c) shows that the loss rate only slightly increases when L increases from 2 to 128 (because the auto-correlation of the loss event intervals is small), while the coefficient of variation of the estimated loss intervals decreases, significantly with $L = 2$ to 16, and moderately with $L = 32$ to 128. This shows the trade-off between these two factors, which underscores the dilemma in tuning parameter L to improve TFRC's long-term fairness behavior. The result of the dynamic 600×600m scenario is similar and omitted for brevity.

Therefore, TFRC's conservative behavior cannot be completely corrected by tuning the number of history samples (L) in its loss event interval estimator. Based on our simulated scenarios, and considering the fairness, smoothness and responsiveness metrics, we conclude that using 8 to 16 samples appears to be an appropriate choice.

VI. Multimedia Streaming in MANET

So far we have uncovered the limitations of TFRC in MANET. Our findings open up the question of how to properly perform multimedia streaming over MANET. In this section, we propose an alternative scheme for multimedia streaming in this network.

To avoid the inherent difficulty of measuring loss rate at end systems, we adopt the *explicit* flow control approach, and design an EXplicit rAte-based flow ConTrol (EXACT) scheme as our solution to the flow control problem in MANET [6]. In [15], we have described the design of EXACT, and show that it outperforms implicit flow control approaches especially in a dynamic and mobile MANET environment. In this section, we study how to enhance EXACT with additional mechanisms to support multimedia streaming over MANET. We call the enhanced scheme EXACT-AA (EXACT with Application Adaptations).

A. Design Rationales

Since EXACT (and EXACT-AA) is a fundamental departure from the traditional implicit flow control approach, we outline its design rationales as follows.

1) Router Assisted Flow Control: In EXACT, router explicitly gives rate signals to the flows that are currently passing it. Since routers are the central places where congestion happens, they are in a better position to detect and react to such condition. Therefore, the router-assisted EXACT scheme is more precise and responsive, which makes it especially suitable in a dynamic MANET environment.

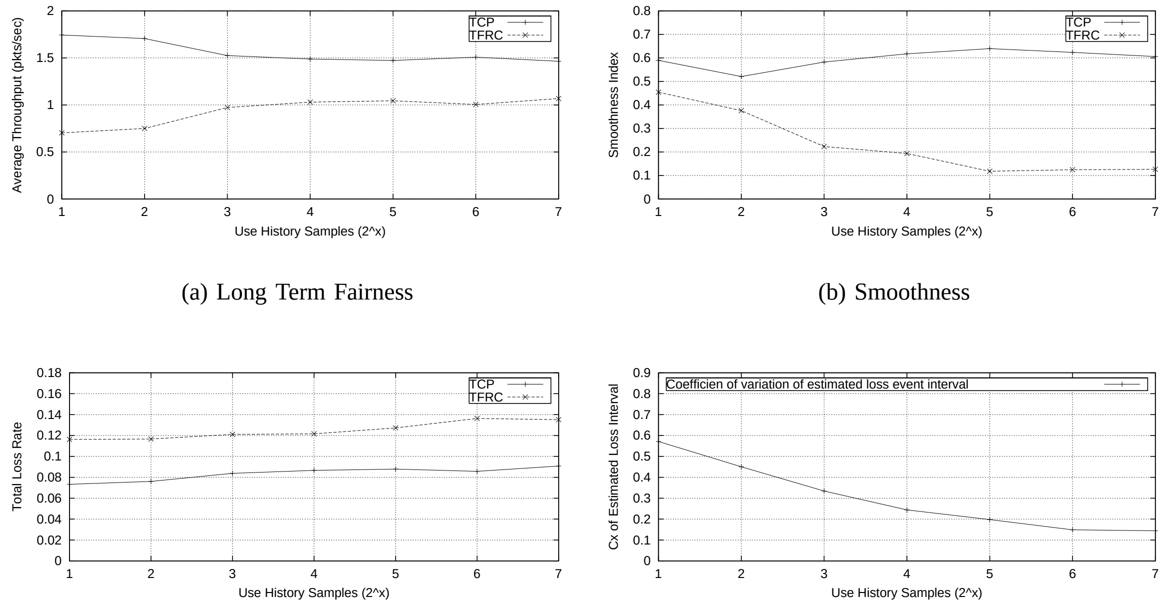
2) Application Specific Adaptation Policy: In order to support multimedia streaming on top of the EXACT flow control scheme, we divide a multimedia flow's rate adaptation into two parts. First, the flow receives explicit rate signal from the routers, which serves as the "upper-bound" that the flow cannot surpass. Second, the flow can adjust its sending rate within the upper-bound according to its own adaptation requirements such as smooth rate change. Therefore, the adaptation policy captures the requirements of multimedia streaming over the network. It provides more flexibility since the application can specify the adaptation policy for itself, and can change the policy on-the-fly depending on user's preferences. For example, a user may choose different policies depending on the media content being played out.

B. EXACT: Explicit Flow Control for MANET

An overview of the EXACT flow control scheme is shown in Figure 10. The sender sends a stream of data packets to the receiver. Each data packet carries a special

[5] Recall that the coefficient of variation of a random variable X is denoted by C_x and defined by $C_x = \frac{\sigma_x}{E[X]}$ where σ_x is the standard deviation of X and $E[X]$ is the mean of X [14].

[6] We use the terms "flow control" and "congestion control" interchangeably in this paper.



(a) Long Term Fairness

(b) Smoothness

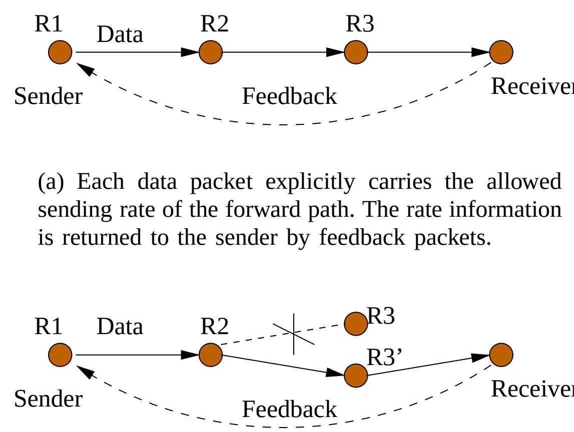
(c) Loss Rate

(d) Coefficient of Variation of Estimated Loss Event Intervals

Fig. 9. Effect of tuning the number of history samples (L) in the static 4-node chain scenario.

IP header, called *flow control header*, which is modified by the intermediate routers to signal the flow's allowed sending rate based on the current available bandwidth at the router. When the packet reaches destination, its flow control header carries the bottleneck rate for the flow, and such rate information is returned to the sender in a feedback packet. In the event of re-routing, the first data packet traveling through the new path collects the new allowed rate of the flow, and returns that to the sender after just one round-trip time of delay.

In EXACT, router bears the responsibility of computing and marking the explicit rates for the flows, i.e., the router allocates its bandwidth resources to the competing flows passing that router. As a result, *fairness* among the competing flows can be guaranteed, and by accurately allocating the total available bandwidth resources, *efficiency* can also be achieved. In [15], we have shown that EXACT outperforms TCP in terms of fairness and efficiency in a dynamic mobile MANET environment.



(a) Each data packet explicitly carries the allowed sending rate of the forward path. The rate information is returned to the sender by feedback packets.

(b) After re-routing, the allowed sending rate of the new path is immediately "learned" by the data packets going through the new path.

Fig. 10. Overview of the EXACT flow control scheme.

C. EXACT-AA: Application Adaptation Policies

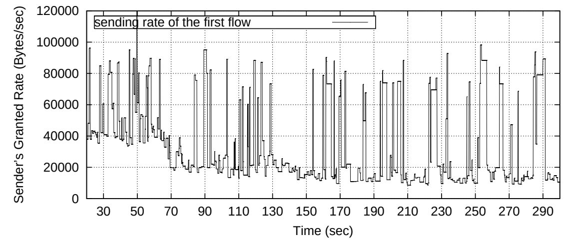
We enhance EXACT with application adaptation policies (EXACT-AA) in order to support multimedia streaming over MANET. There are many choices of adaptation policies. As an example, we propose a policy called Delayed Increase Immediate Decrease (DIID), where a flow can increase its rate only when the rate signal from the routers has increased *and* sustained over a certain period of time, and the flow has to reduce its rate immediately when the rate signal decreases in order to conform to the underlying flow control scheme. The intuition of this policy is that, many rate increases in MANET are temporary and short-lived, due to wireless channel contention and interference. Therefore, the DIID policy avoids temporary spikes in bandwidth allocation, and captures only those that are sustainable. It has similar effects of a low-pass filter. Other adaptation policies are also possible.

Over the long-term, the DIID policy is conservative, i.e., it has less throughput than those who closely follow router's rate signals. Since the router's rate signal is time-varying, there is an *inherent* trade-off between smoothness and fairness in our scheme, which is similar to the observations of TFRC in Section III. However, in our scheme, the trade-off is controllable by each application. Therefore, it provides a tunable knob where users can *pro-actively* adjust their preferences of this trade-off.

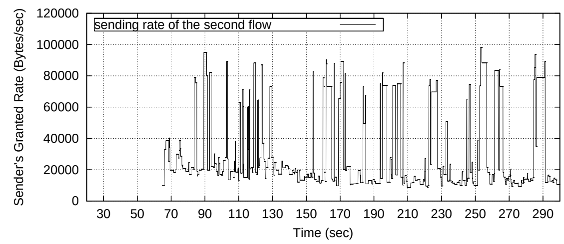
D. Test-bed Experiments of EXACT-AA

We show the results of an audio streaming application using the EXACT-AA scheme over a MANET test-bed. Our test-bed consists of four Redhat Linux laptops equipped with Lucent WaveLAN 802.11b cards in ad hoc mode. The laptops are configured with fixed-routing tables such that they form a 3-hop chain topology, and they are moved around in an office building with channel interferences from a nearby wireless LAN. We implement two types of flows: 1) greedy UDP flow which sends out data according to the rate signal from the routers; and 2) audio streaming flow which sends out audio data based on the rate signal from the routers and with a 5-second DIID adaptation policy, i.e., the audio flow will increase its rate only when the rate increase has sustained for 5 seconds.

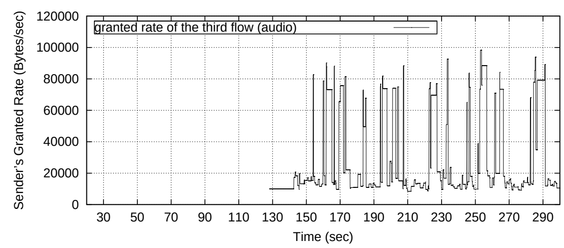
Three flows are created with staggered starting times. Two UDP flows start at time 0s and 65s, respectively. An audio flow starts later around time 130s. Figure 11 shows router's rate signals for the three flows. It is evident that their allowed sending rates are *highly* dynamic, and that many rate increases are temporary and short-lived.



(a) First Greedy UDP Flow



(b) Second Greedy UDP Flow



(c) Audio Streaming Flow

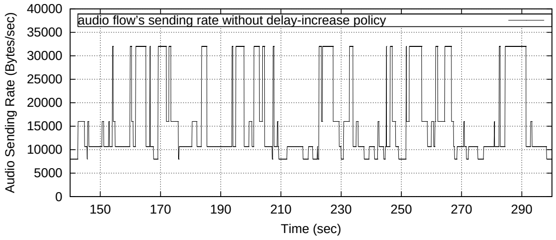
Fig. 11. Explicit rate signal from the routers.

We show the audio streaming flow's rate signal with and without the DIID policy in Figure 12. It shows that without the DIID policy, the audio flow has to adjust its media quality (i.e. the audio sampling rate from the microphone) frequently. After applying the DIID policy, the rate change events are greatly reduced, and the flow's smoothness is significantly improved. Our EXACT-AA scheme provides a tunable knob for the application to perform its own adaptations. Such *informed* adaptation is possible because of the explicit rate signals provided by the underlying EXACT flow control scheme.

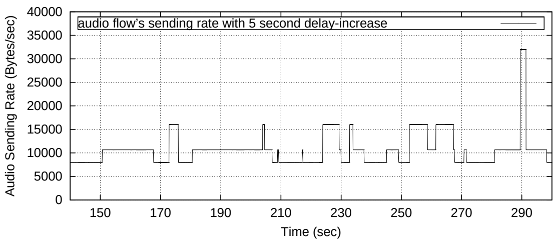
E. Feasibility in MANET

Admittedly, the EXACT (and EXACT-AA) scheme incurs additional complexity and overhead at the routers, such as computing rate allocation for the flows. Therefore, our scheme is *not* targeted for the large scale Internet (where core routers have to process huge number of concurrent flows), but rather as a solution for the special MANET environment. Since MANET is often a small scale network and there is no "core" router in the network, the number of concurrent flows going through a MANET router is likely to be relatively small.

We conduct stress tests on the test-bed in order to gauge router's overhead in running the EXACT scheme. We create 10 concurrent UDP flows over the 3-hop path, and use a slow speed laptop (Pentium II 266Hz with 224 KB memory) as one of the intermediate routers. The result shows that, even with a Java implementation, the Pentium II laptop's CPU occupancy is less than 4%. That means EXACT is well within the computing power of today's mobile devices.



(a) Without DIID Policy.



(b) With 5-second DIID Policy.

Fig. 12. Audio streaming flow's adaptation behavior.

F. Discussion of Multimedia Streaming in MANET

There are two general approaches of multimedia streaming over MANET. The first approach adopts *soft-state reservation* to protect multimedia traffic from best-effort traffic, e.g., INSIGNIA [16], SWAN [17] and dRSVP [18]. INSIGNIA is a resource signaling protocol to support end-to-end adaptive services, such as multimedia flows with a base layer and an enhanced layer. The bandwidth reservation status of the two layers is carried with each data packet, so that any change in router's reservation may be conveyed to the sender quickly. SWAN is a scheme to support the delivery of real-time traffic. Before sending out a real-time flow, the sender must probe the path to see how much bandwidth is left at the intermediate routers to accommodate additional real-time traffic. The current aggregate real-time traffic at a router, therefore, constitutes a reservation state at that router. dRSVP is a scheme to provide end-to-end bandwidth guarantee for a flow, where the flow's bandwidth request is specified as a range, instead of a scalar value. Routers along the path may choose to reserve bandwidth for the flow within that range in order to increase its chance of successful reservation in a dynamic network.

The second approach does not involve any explicit or implicit resource reservation for multimedia traffic. All the flows are treated equally by the routers. Examples of this category include our EXACT-AA scheme and the utility-based adaptation approach [19] designed for infrastructure wireless networks. The utility-based adaptation approach allocates bandwidth for the flows based on their declared utility functions, and equalizes each flow's utility to resolve resource contentions. Each application may decide how much bandwidth to consume within the allocated rate.

Although the reservation-based approaches offer more protection for multimedia traffic, it is unclear how reservation can take place in MANET where the nodes interact with each other in a spontaneous fashion, without any service level agreements among them. Research in this direction must be enhanced with an incentive or pricing mechanism for the reserved resources.

VII. CONCLUSION

We study the behavior of TFRC equation-based congestion control and multimedia streaming in MANET. Using ns-2 simulations, we show that while TFRC is able to maintain smoother throughput than TCP, it obtains less throughput (0.2 to 0.8) than the competing TCP flows (i.e., being conservative). We analyze several factors contributing to TFRC's conservative behavior, including loss rate discrepancy, inaccuracy of loss rate prediction, and lack of auto-correlation in MANET's loss process, many of which are inherent to the MANET network. We also explore the effect of tuning TFRC's loss event interval estimator, and show that its conservative behavior cannot be completely correct. Our study reveals the limitations of applying TFRC to the MANET domain, and shows that it can be used only when strict throughput fairness is not a major concern.

To address the open problem of multimedia streaming in MANET, we propose an alternative scheme (called EXACT-AA) based on router's explicit rate signaling and application's adaptation policies. We demonstrate the feasibility of our scheme using an audio streaming application over a real MANET test-bed.

ACKNOWLEDGMENT

This work was supported by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant N00014-00-1-0564, and the NSF EIA 99-72884EQ grant. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

REFERENCES

[1] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[2] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "Tcp friendly rate control (tfrc): Protocol specification," RFC 3448, Jan. 2003.

[3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling tcp throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM'98*, Vancouver, Canada, Sept. 1998.

[4] D. Bansal and H. Balakrishnan, "Binomial congestion control algorithms," in *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA, Apr. 2001.

[5] Y. Yang and S. Lam, "General aimd congestion control," in *Proc. IEEE Intl. Conf. on Network Protocols (ICNP 2000)*, Osaka, Japan, Nov. 2000.

[6] B. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker, "Dynamic behavior of slowly-responsive congestion control algorithms," in *Proc. ACM SIGCOMM 2001*, San Diego, California, USA, Aug. 2001.

[7] Y. Yang, M. Kim, and S. S. Lam, "Transient behaviors of tcp-friendly congestion control protocols," in *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA, Apr. 2001.

[8] M. Vojnovic and J.-Y. LeBoudec, "On the long-run behavior of equation-based rate control," in *Proc. ACM SIGCOMM 2002*, Pittsburgh, Pennsylvania, USA, Aug. 2002.

[9] C. Zhang and V. Tsaoussidis, "The interrelation of tcp responsiveness and smoothness in heterogeneous networks," in *Proc. 7th IEEE Symposium on Computers and Communications (ISCC 2002)*, Taormina, Italy, July 2002.

[10] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," in *Proc. ACM/IEEE MobiCom'99*, Seattle, Washington, USA, Aug. 1999.

[11] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1996.

[12] M. Kazantzidis, M. Gerla, and S. Lee, "Permissible throughput network feedback for adaptive multimedia in aodv manets," in *Proc. IEEE Intl. Conf. on Communications (ICC 2001)*, Helsinki, Finland, June 2001.

[13] S. H. Shah, K. Chen, and K. Nahrstedt, "Dynamic bandwidth management for single-hop ad hoc wireless networks," in *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, Dallas-Fort Worth, Texas, USA, Mar. 2003.

[14] K. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[15] K. Chen, K. Nahrstedt, and N. Vaidya, "The utility of explicit rate-based flow control in mobile ad hoc networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Atlanta, Georgia, USA, Mar. 2004.

[16] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. Campbell, "Insignia: An ip-based quality of service framework for mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 60, no. 4, pp. 374–406, 2000.

[17] G.-S. Ahn, A. Campbell, A. Veres, and L.-H. Sun, "Swan: Service differentiation in stateless wireless ad hoc networks," in *IEEE INFOCOM 2002*, New York, NY, U.S.A., June 2002.

[18] M. Mirhakkak, N. Schult, and D. Thomson, "Dynamic bandwidth management and adaptive applications for a variable bandwidth wireless environment," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 1984–1997, 2001.

[19] R. R.-F. Liao and A. T. Campbell, "A utility-based approach for quantitative adaptation in wireless packet networks," *Kluwer Wireless Networks*, vol. 7, pp. 541–557, 2001.

Kai Chen Kai Chen is currently a Ph.D. candidate in the Department of Computer Science at the University of Illinois at Urbana-Champaign. His research interests include mobile ad hoc networks, wireless networking, transport layer protocols, quality of service, and pervasive computing. He received his B.Engr. degree from Tsinghua University, Beijing, China in 1995, and M.S. degree from the University of Delaware, Newark, Delaware, U.S.A. in 1998. From 1998 to 2000, he worked as a research programmer at the National Center for Supercomputing Applications (NCSA) at Urbana, Illinois, U.S.A.

Klara Nahrstedt Klara Nahrstedt is an associate professor at the University of Illinois at Urbana-Champaign, Computer Science Department. Her research interests are directed towards multimedia middleware systems, quality of service(QoS), QoS routing, QoS-aware resource management in distributed multimedia systems, and multimedia security. She is the coauthor of the widely used multimedia book 'Multimedia: Computing, Communications and Applications' published by Prentice Hall, the recipient of the Early NSF Career Award, the Junior Xerox Award, and the IEEE Communication Society Leonard Abraham Award for Research Achievements. She is the editor-in-chief of ACM/Springer Multimedia Systems Journal, and the Ralph and Catherine Fisher Associate Professor. Klara Nahrstedt received her BA in mathematics from Humboldt University, Berlin, in 1984, and M.Sc. degree in numerical analysis from the same university in 1985. She was a research scientist in the Institute for Informatik in Berlin until 1990. In 1995 she received her PhD from the University of Pennsylvania in the department of Computer and Information Science.