TCP Friendly Rate Adaptation for Multimedia Streaming in Mobile Ad Hoc Networks

Zhenghua Fu, Xiaoqiao Meng, Songwu Lu

Computer Science Department, University of California, Los Angeles
{zfu,xqmeng,slu}@cs.ucla.edu

Abstract. Transport protocol for supporting multimedia streaming in mobile ad hoc networks has to cope with the rich dynamics, including mobility-induced disconnection, reconnection, and high out-of-order delivery ratios; channel errors; and network congestion. In this work, we describe the design and implementation of ADTFRC, a TCP-friendly transport protocol for ad hoc networks. ADTFRC adapts wireline TFRC protocol to ad-hoc networks with improved rate adaptation behavior, the capability of application layer framing and selective retransmissions. It detects different loss behaviors based on end-to-end measurements of multiple metrics; this allows ADTFRC to more accurately gauge the network behavior and achieve higher throughput. Our ns-2 simulations show much performance improvement over standard TFRC and TCP with explicit-link-failure-notification (ELFN) support in terms of throughput, rate adaptation behavior and application level quality while still maintaining the TCP-Friendliness property.

1 Introduction

With the advent of IEEE 802.11 technology, wireless channel typically supports a data rate up to 11M bps or higher, which makes multimedia streaming feasible. Potential multimedia streaming applications for a mobile ad hoc network include multimedia instant messaging, environmental monitoring and distributed gaming.

Current research on multimedia streaming over ad hoc networks focuses on low-layer design such as service differentiation in MAC QoS-aware routing, admission control and adaptive packet scheduling. In this paper, we address transport-layer issues. We adapt the popular, slowly responsive congestion control protocol – TCP Friendly Rate Control (TFRC) – proposed for wired multimedia transport, to mobile ad hoc networks. To this end, we devise novel end-to-end mechanisms to effectively address the wireless and mobility issues.

Early proposals use UDP to carry multimedia streams [21]. Since UDP does not provide congestion control, these unresponsive multimedia flows compete unfairly with other responsive TCP flows. Using TCP in multimedia streaming transport can prevent congestion collapse, however, TCP provides 100% reliability through its retransmission mechanism, which is not necessary for loss-tolerant multimedia streaming. Moreover, TCP halves its transmission rate upon any congestion event; such dramatic oscillations in rate adaptation are deemed detrimental to applications.

Recent research on multimedia transport has focused on developing a TCP friendly protocol that does not react to any single congestion event dramatically and slowly adapts to network dynamics [16][18][19]. A noticeable proposal is TCP Friendly Rate Control (TFRC) [17][20], which theoretically characterizes a TCP friendly throughput given the RTT and congestion probability measurements, and increases/decreases the transmission rate accordingly.

There are several technical challenges for TFRC to function well in mobile ad hoc networks. Such networks exhibit a rich set of packet loss behaviors. It is well known that applying congestion control upon every packet loss in a TCP friendly protocol leads to unsatisfactory performance in ad hoc networks [1][6][7][5]. Moreover, TFRC has to handle other types of network events, including mobility-induced disconnection and re-connection, route-change-induced out-of-order delivery and error/contention-prone wireless transmissions[1]. These events require TFRC to respond differently from congestion control. For example, it makes sense to simply ignore a random packet loss due to channel errors rather than to multiplicatively decrease the current sending rate [3]; and it is more appropriate to periodically probe the network during disconnections for a prompt recovery than to slow down and exponentially increase the retransmission timer [1]. Another challenge is how to accurately detect and differentiate these events in the first place. Packet loss as the sole detector used by conventional TCP/TFRC flows cannot differentiate all these new events [8].

1.1 Rate Adaptation in Ad Hoc Networks

In order to effectively adapt the transmission rate, network congestion has to be reliably detected. In particular, among all kinds of packet losses, the congestion loss probability needs to be estimated; treating all losses as congestion loss leads to undesirable rate adaptations. Most of the literature on congestion detection for ad hoc networks endorses a network-oriented approach. In this approach, the routers implement a monitoring module and generate explicit notifications back to the sender upon various packet losses. Specifically, if mobility triggers network disconnection, an explicit link failure notification (ELFN) will be sent to the sender [1]; an explicit congestion notification (ECN) message is generated when a congestion loss occurs [24]; and an explicit loss notification (ELN) is sent to the TCP sender when the router observes a wireless channel-induced packet loss [3][4]. While these are sound techniques to improve TCP, they have several drawbacks when used for proper rate adaptation in ad hoc networks.

First, the failure notifications, such as ELFN, generated by intermediate nodes are sent directly to the sender; the receiver is left unaware of this message. Thus, single packet loss event can trigger two different reports from intermediate nodes and the receiver end host. Since these two reports can arrive at the sender in arbitrary order, it is difficult to combine the two observations made at different time and render a consistent image of overall network conditions. [2] Second, the failure notifications require a global deployment of monitoring modules at every node, which could be difficult to adopt in

[1] Even with link-layer re-transmissions of 802.11 MAC, packet loss still occurs due to bursty channel error or MAC-layer contentions.

[2] Even if the measurements are performed at sender side, feedbacks from both receiver and intermediate nodes are still needed, same complication arises when trying to combine the two overlapping observations.

practice because of the inherent heterogeneity of participating nodes in ad hoc environments. Nodes ranging from notebooks to hand-held devices have variant resources and consequently varying abilities to adopt full-blown network-oriented transport solutions.

Based on the above considerations, an end-to-end approach is more desirable for TCP-Friendly rate adaptation in mobile ad hoc networks since it falls naturally into the existing TFRC protocol [17]. Network events are differentiated at the receiver end host, who derives the congestion probability and paces the feedback to help sender effectively adjust its transmission rate. In addition, the end-to-end approach is much easier to be implemented and deployed in practice.

1.2 Main Contributions

The key innovation proposed in this paper is the use of multi-metric joint detection instead of single-metric detection. Because end-to-end measurement data in ad hoc networks is highly noisy, frequent false identifications and notifications could happen [8]. How to detect events in an ad hoc network in a *robust* manner using noisy measurements poses a great design challenge. With multi-metric joint detection, we exploit the degree of independence in the measurement noise of each individual metrics, so that the probability of false identification is significantly reduced by cross verification among the multiple metrics.

In addition, we applied the Application Level Framing (ALF) and Partial Reliability techniques to improve the quality of the multimedia streaming perceived by an end user. The resultant ADTFRC protocol is implemented in NS-2 simulator, and the performance of ADTFRC is extensively evaluated in mobile ad hoc networks.

The results show that, without compromising the TCP-Friendliness property, ADTFRC outperforms TFRC and TCP NewReno with ELFN support in terms of throughput, packet loss ratio and smoothness in rate adaptation behavior. With real MPEG-4 video traces, the streaming quality of ADTFRC is evaluated by *aggregated starvation time* at the user level. We demonstrate that ADTFRC significantly improves the streaming quality of TFRC and TCP NewReno with ELFN support in mobile ad hoc networks.

The remainder of the paper is organized as follows: Section 2 provides an overview of our ADTFRC design. Section 3 describes the detection design. The ADTFRC protocol implementation issues together with the ALF and partial reliability design is presented in Section 4. Performance evaluations are given in sections 5. Section 6 discusses the related work and Section 7 concludes the paper.

2 Design Goal of ADTFRC Protocol

In this section, we describe the design rationale of the ADTFRC, together with an overview of the protocol specification. Our starting point is the TFRC framework introduced in the wireline network [17]. The first question one would ask is: what is the ideal rate adaptation behavior in the mobile ad hoc networks? In another words, if the perfect knowledge of the underlying network is given, what particular network states must ADTFRC distinguish and how should the transmission rate be adapted in each of these states?

In the following we define the network states to be distinguished and the most suitable (ideal) rate adaptation policies to be applied in each situation.

CONGESTION (CONG): We define congestion in ad hoc networks as the signal that the offered load exceeds the network capacity. When congestion occurs, there will be queue building up and the network throughput is reduced due to excessive contention delays and collision losses. To deal with congestion, the transport protocol should reduce the sending rate, and react similarly to the congestion control actions of standard TFRC.

CHANNEL_ERR (CHERR): When random packet loss occurs, the receiver should not count it as congestion event; without slowing down, the sender should calculate the sending rate as normal.

ROUTE_CHANGE (RTCHG): The delivery path between the two end hosts can change from time to time, with disconnections that are too transient to result in a retransmission timeout. Depending on routing protocols, the receiver may experience a short burst of out-of-order packet delivery or packet losses. In both cases, the receiver, again, should not count it as congestion; and the sender should keep the streaming rate unchanged in the next RTT period, waiting for the receiver to feedback more measurement statistics for the new path.

DISCONNECTION (DISC): When the delivery path is disconnected for long enough to cause a retransmission timeout, instead of exponentially slowing down and backing off, the sender should freeze the current congestion window and the retransmission timer. It then performs a periodic probing so that the transmission can be resumed promptly once a new path is established. Once it is recovered, the actions of RTCHG should be followed. We notice that this probing technique is also proposed in [1][2].

So far, we have established the target network states to be identified in ADTFRC, that is, CONG, CHERR, RTCHG and DISC. How to reliably detect them using noisy end-to-end measurements poses a major challenge. In the next section, we present the multi-metric joint detection algorithm to achieve this goal. The detection algorithm is implemented at the receiver end host, who periodically updates the sender with its current network state estimation using ADTFRC feedback packets.

3 Detection via Multiple Metrics

This section describes the end-to-end metrics used in the detection algorithm, and how we use them to jointly detect the four network states. The achieved accuracy by our approach is also evaluated through simulations at the end of this section.

3.1 Devising End-to-End Metrics

End-to-end measurement is widely used in transport protocols. In TCP, the round trip time (RTT) is maintained by the sender to calculate the retransmission timeout. Previous work uses delay related metrics to measure the congestion level of the network. For example, [2] and [8] use inter packet arrival delay, and [9] uses RTT to estimate the expected throughput. A challenge in ad hoc networks is that packet delay is not only influenced by network queue length, but also susceptible to other conditions such as random packet loss, routing path oscillations, MAC layer contention, etc. These conditions make such measurement highly noisy. Rather than pursuing any single metric that is robust to all dynamics of the network, we devise four end-to-end metrics that tend to be influenced by different conditions so that the noise independence among them can be exploited by multi-metric joint identification.

Inter-packet delay difference IDD Metric *IDD* measures the delay difference between consecutive packets. It reflects the congestion level along the forwarding delivery path by directly sampling the transient queue size variations among the intermediate nodes. However, in an ad hoc network, there are still a number of situations in which *IDD* values might give an incorrect estimation of congestion. For example, *IDD* can be influenced by non-congestion conditions like mobility induced out-of-order packet delivery. We therefore introduce an additional metric STT in the following.

Short-term throughput STT Compared with IDD, STT is also intended for network congestion identification. However, it provides observation over a time interval T, and is less sensitive to short term out-of-order packet delivery than IDD. Therefore, STT is more robust to transient route changes, which can be very frequent in a mobile ad hoc network. However, using STT alone to detect network congestion can be susceptible to measurement noise introduced by bursty channel error, network disconnections or altering source rate. In the following, we combine STT and IDD to jointly detect the network congestion. Aside from the above two delay related metrics, we also consider the following two metrics for non-congestion state identification.

Packet out-of-order delivery ratio POR A packet is counted as being out-of-order if it arrives after a packet that was sent later than it (by the same sender). The receiver records a maximum sending time for all the received packets from the connection, denoted by T_{maxtx}. Every received packet that has a sending time-stamp less than T_{maxtx} is added into POR. POR is intended to indicate a route change event. During the route switching period, multiple delivery paths exist. Packets along the new path may catch up, and those along the old path are then delivered out-of-order.

Packet loss ratio PLR At each time interval $[t, t+T]$, we compute this metric as the number of missing packets in the current receiving window. POR can be used to measure the intensity of channel error.

In this section, we describe network states that are important in improving TFRC performance in an ad hoc network, and metrics that can be measured end-to-end. In the next section, we study how to identify these states using the four metrics discussed above.

3.2 Detecting The Network States

For all simulation results shown in this section, the default settings are as follows unless otherwise specified. We use the NS-2 simulator with CMU wireless extension modules. 30 wireless nodes roam freely in a $400m \times 800m$ topology following a *random waypoint* mobility pattern, in which the pause time is zero so that each node is constantly moving. The wireless link bandwidth is 2M bps, and IEEE 802.11 and Dynamic Source Routing (DSR, see [1]) are used as MAC and routing layer protocols respectively. One TFRC flow created with packet size of 1000 bytes. To introduce congestion, three competing UDP/CBR flows, each with source rate 180K bps, are created within the time intervals of [50,250],[100,200] and [130,170] respectively. Each UDP flow transmits at 180Kbps. The simulations last for 300 seconds.

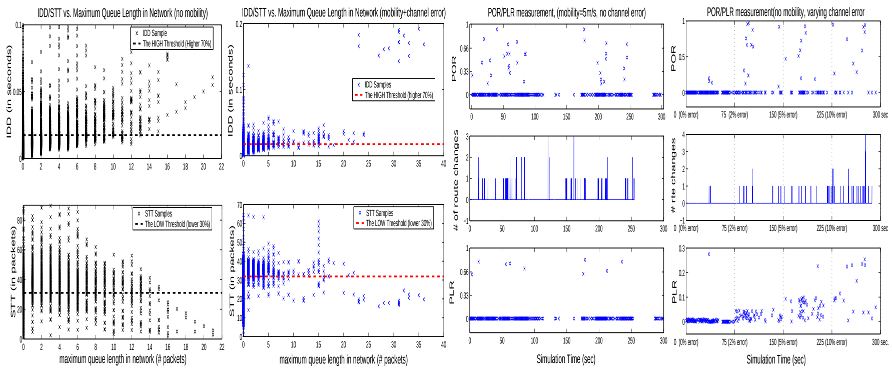

Fig. 1. *End To End Metric Measurement. Left two: the IDD and STT measurement w.r.t. instantaneous maximum queue occupation of all wireless nodes in the network. First figure shows simulation with static nodes; second figure shows simulation with mobile nodes (5m/s). Right two: POR and PLR measurements w.r.t. the number of route changes. The third figure shows simulation with mobile nodes (5m/s), no channel error. The fourth one shows simulation with static nodes, progressively increasing channel error (0%,2%,5%,10%) during the entire run.*

Detecting Congestion To study the relationship between network congestion and IDD/STT, we simulate both static and mobile scenarios. A TFRC flow and three competing UDP/CBR flows are introduced in each simulation and the network is expected to become congested as it becomes overloaded. The first two figures of Fig. 1 show the simulation results. The first is for the static case without channel errors, and the second is for the mobile case with node mobility speed of 5m/s and 5% channel error. In both figures, we plot the measured IDD/STT values with respect to the instantaneous maximum buffer occupation of all nodes in the network, which reflects the network congestion level at the sampling time instance. [3]

In Fig 1, we observe that when the maximum network queue size exceeds half of the buffer capacity (25 packets), IDD is clearly *high* and STT is clearly *low*. We formalize this observation by defining a value to be HIGH or LOW respectively if it is within the top or bottom 30% of all samples [4]. However, when the network queue size is small (non-congestion case), both IDD and STT vary from LOW to HIGH, with the majority of IDD samples being not HIGH and STT samples not LOW. In the left two figures of Fig. 1, when node mobility is present, the two metrics become much more noisy in non-congestion state (i.e., small network queue).

In the single metric-based detection using either IDD or STT, the noise reduces the accuracy significantly when the network is not congested, especially in scenarios with mobility and channel errors. However, in the proposed joint detection approach, we can

[3] The maximum buffer size for each node is 50 packets in our simulations.

[4] This threshold was determined empirically from simulation results and real testbed measurements [15].

use both metrics to *verify* each other to improve the accuracy. Specifically, we identify a congestion state when both IDD is HIGH and STT is LOW, and non-congestion state if otherwise. The following shows why the multi-metric approach has better detection accuracy than the single metric approach.

When the network is congested, let P_1 and P_2 be the probabilities that IDD is HIGH and STT is LOW respectively. The single metric accuracy is $acc_{idd}(cong) = P_1$ and $acc_{stt}(cong) = P_2$. For the multiple metric case, $acc_{multi}(cong) = P_1 \cdot P_2$. Since the simulations show that $P_1 \simeq P_2 \simeq 1$ (see left two figures of Fig. 1), these three accuracies are roughly equal in congestion state. On the other hand, when the network is not congested, let P_1' and P_2' be the probabilities that IDD is still HIGH and STT is still LOW. Similarly, we have $acc_{idd}(non\_cong) = 1 - P_1'$, $acc_{stt}(non\_cong) = 1 - P_2'$ and $acc_{multi}(non\_cong) = 1 - P_1' \cdot P_2'$. Since each noise probability, $0 < P_1', P_2' < 1$, is non-negligible, multiple metrics thus achieve higher accuracy. Combining these two cases, multi-metric identification improves the accuracy in non-congestion states while maintaining a comparable level of accuracy in the congestion state. Therefore, it achieves better identification performance over a variety of network conditions.

The key insight here is that in the non-congestion state, IDD and STT are influenced differently by various network conditions, such as route change and channel error; while in congestion state, they are both dominated by prolonged queuing delay. Thus, the two noise probability P_1' and P_2' become largely independent. Effective verification across multi-metrics is possible as long as these conditions do not co-exist during the measurement time interval. Although this joint identification technique cannot achieve perfect accuracy, it does increase the accuracy significantly as we show in Section 3.3

Detecting Non-Congestion States If the network state is not congestion, we next seek to detect whether it is RTCHG or CHERR. The third figure of Fig 1 shows data from a simulation run with node mobility speed being 5m/s. A single TFRC flow is created without any competing flows that might cause network congestion. We plot POR and PLR sample values together with the number of route changes in the forwarding path over time. A clear correlation is seen between route change events and bursts of high POR measurement. During the changing period, packets arrive at the receiver from multiple paths and consequently may lose their ordering. Although not all route changes result in out-of-order packet delivery, we only count those observable changes, which would have an impact upon TFRC. POR can be used to identify RTCHG state and PLR can be used for CHERR state.

Moreover, since there is no congestion or channel errors in these simulations, PLR remains stable with a few significant outliers. These anomalies correspond to situations in which packets along the old path are excessively delayed or lost.

In the simulation shown in the fourth figure of Fig 1, nodes are stationary and four channel error rates (0%, 2%, 5% and 10%) are introduced into four identical time intervals (75 seconds). In this case, packet loss is proportional to the channel error rate and the PLR gradually increases as the channel error rate increases. Note that a high channel error rate can also create route change in the network that will in turn result in bursts of high POR measurements. The routing layer interprets any MAC-layer transmission failures (in this case, channel error) as a sign of a broken link and consequently seek to repair/re-establish the delivery path, which may cause route changes.

In conclusion, a burst of high POR sample values is a good indication of a route change and a high PLR is a good indication of a high rate of channel error. It should be noted that the network may be both in a state of high channel error and route change, which can be identified by high values in both PLR and POR.

We next consider disconnection. Disconnection happens when packet delivery is interrupted for non-congestion reasons for long enough to trigger a retransmission timeout at the sender. Multiple network conditions can trigger such a timeout at the sender including frequent route changes, heavy channel error, and network partition after mobility. If the timeout is triggered by congestion, then previous state feedback should reflect the transient queue build up period by increasing IDD and STT measurement at the receiver; if not, the timeout was due to non-congestion conditions in the network. Therefore, a DISC state is identified at the sender if the current state estimation is non-congestion when retransmission timeout is triggered.

	IDD and STT	POR	PLR
CONG	(High, Low)	*	*
RTCHG	NOT (High, Low)	High	*
CHERR	NOT (High, Low)	*	High
DISC	(*, ≈ 0)	*	*
NORMAL	default		

Table 1. *Metrics patterns in 5 network states. High: top 30% values; Low: bottom 30% values; '*': do not care*

Table 2 summaries the metrics patterns in five different network states. They are the identification rules used in our ADTFRC. We show later in this section that such an identification method, combined with a simple sample classification technique, achieves an accuracy above 80% on average in all simulations scenarios.

3.3 Detection Accuracy

We now study the accuracy of congestion identification using the RSD technique. In particular, we compare the single-metric (using only IDD or STT) and multiple-metric (using both) approaches. We run two sets of simulations under non-congested and congested cases (Figure 2). In the first non-congested case, a single TFRC flow is created within the topology. In the second congested case, two competing UDP flows are created as before. In both cases, 1% random channel error is introduced and the mobility speed varies from 0 to 20 m/s. We repeat simulations 50 times at each speed to reduce the impact of random topology factors.

During the simulation, upon each packet loss, we compare the identified network state and the actual network state to determine the accuracy of detection[5]. In particular, if a packet is lost due to network congestion, but the algorithm gives non-congestion

[5] The real network state is obtained by a global monitor implemented in NS-2 simulator. See [5] for implementation details.

estimation, we count it as an incompatible error because this error in detection (and only this one) causes ADTFRC to be more aggressive than a TCP-friendly flow and consequently TCP-incompatible.

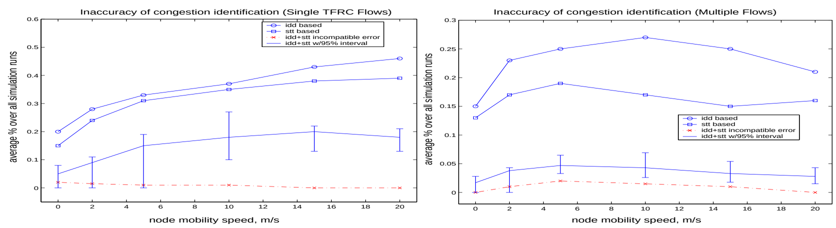

Fig. 2. *Identification Accuracy. Left: Percent of inaccurate identifications in a non-congested case, Right: Inaccuracy ratio in a congested case*

Figure 2 shows the percentage of inaccurate identification in both cases. In the single TFRC flow case (the left figure), mobility and channel error are the dominant reasons for packet loss. The increase in mobility speed reduces the accuracy of the single-metric identification quickly. However, from the simulations, the multi-metric approach results in only 10% to 30% inaccurate identification. This is achieved by the cross verification between IDD and STT measurements to eliminate false congestion alarms. Meanwhile, the incompatible error remains less than 2%.

In the multi-flow cases (the right figure), congestion happens more frequently. For multi-metric identification, more than 95% accuracy is observed in all simulations with less than 2% incompatible errors. For the single metric approach, accuracy is only about 70% to 80%.

In summary, we have demonstrated that multiple metrics combined with RSD is a feasible approach to detect network events by end-to-end measurements only.

4 ADTFRC Protocol Design and Implementation

We now incorporate the design of Sections 2 and 3 in our ADTFRC protocol to improve the performance of TFRC in ad hoc networks.

4.1 Adaptive Rate Adaptation

ADTFRC seeks to maintain backward compatibility with conventional TFRC. It uses identical connection establishment and connection teardown processes. It estimates the RTT and derives the sending rate in the same way with TFRC. To improve the performance of TFRC in ad hoc networks, ADTFRC makes several extensions at both the sender side and receiver side.

Upon each packet arrival at the receiver, besides the normal operations, values for the four previously discussed metrics are calculated and network states are estimated. In ADTFRC, the congestion probability is calculated based on the outcomes of our multi-metric identification instead of the packet loss events. The receiver then passes this congestion frequency measurement together with state estimations, i.e., CONG, CHERR and RTCHG, to the sender in every feedback packet. Besides the regular feedback of each RTT, the receiver generates *Urgent* state update packet as soon as a congestion event is detected and feedback to the sender immediately. The sender maintains the most current state report received, and proceeds with normal TFRC operations until either of the following two events happen: the reception of feedback packet, or the re-transmission time out. A modified TFRC state diagram is shown in Figure 3 for the sender.

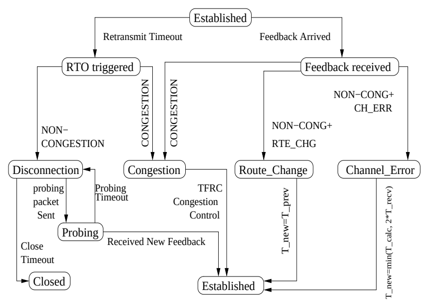

Fig. 3. *ADTFRC state diagram for sender in* NS-2 *implementation*

A feedback report or retransmission timeout triggers ADTFRC to take different control actions according to the current network state estimation. In particular, a probing state is introduced to explicitly handle network disconnection. When a non-congestion induced retransmission timeout occurs at the sender, ADTFRC freezes its current transmission state and enters a probing state. The sender leaves the probing state when a new acknowledgement is received or the probing is timed out[6]. The ADTFRC connection is closed after multiple probing attempts fail.

Pseudo-codes that illustrate the actions taken at both the sender and receiver are avaialbe in our technical report [15].

5 Performance Evaluation

This section evaluates the performance of ADTFRC through extensive *NS-2* simulations in terms of its throughput, rate adaptation behavior as well as the application level quality perceived by the end user.

5.1 Throughput Improvement

In the throughput evaluation part, we compare it to TFRC and TCP with ELFN [1] support. Instead of using end-to-end measurements, TCP ELFN collects link state information directly from the network and is expected to be more accurate. It is used as a reference system; a throughput close to ELFN indicates the effectiveness of ADTFRC.

Figure 4 shows the single flow throughput of ADTFRC, TFRC and TCP-NewReno with ELFN support. The simulation parameters for TFRC flows are set as described in

[6] A similar probing mechanism was proposed by [1]

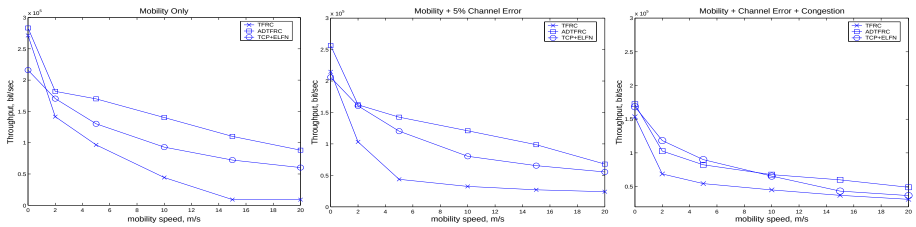

Fig. 4. *Performance Improvement of ADTFRC. From left to right:1)mobility only, 2)mobility+5% channel error, 3) mobility+5% channel error + 3 competing UDP/CBR flows*

section three, and for TCP ELFN flow, we set the packet size to be 1000 bytes and maximum window size to be 8 packets. In all three cases, ADTFRC provides significantly better throughput than TFRC. When nodes are mobile, ADTFRC achieves a throughput improvement from 100% to 800% over TFRC. Furthermore, it is surprising to see that ADTFRC out-performs TCP+ELFN even in a static network where the mobility speed is zero. The reason is because the ACK packet traffic on the reverse path is much heavier in TCP+ELFN than in ADTFRC. Due to the broadcast nature of the wireless link, such ACK flows contend for the channel access with forwarding data flows, introducing additional delay in RTT and resulting in throughput decrease.

5.2 Rate Adaptation

We further measure the rate oscillations experienced at the receiver for each of these three protocols. To effectively support best-effort multimedia streaming, dramatic rate variations are highly undesirable.

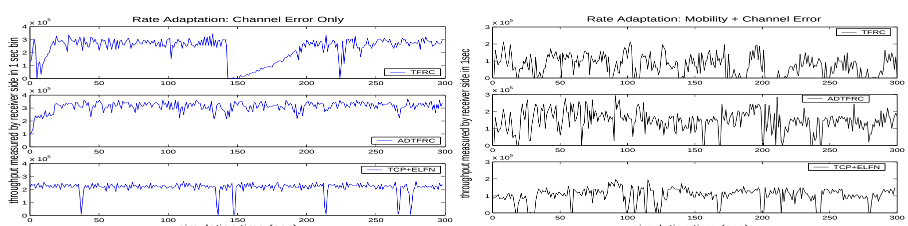

Fig. 5. *Smoothness of Rate Adaptation. From left to right:1)Single flow in static ad hoc network, 2% random channel error. 2) Single flow with node mobility 5m/s, 2% random channel error.*

In Figure 5 we show the throughput fluctuations at the receiver side in two environment settings. The left figure is for the simulation in static network, with 2% random channel error. It shows that the ADTFRC (the middle one) maintains a more stable transport rate.

When mobility and channel error are both introduced as shown in the right one of Figure 5, the rate variation of ADTFRC becomes much larger. However due to a probing mechanism of ADTFRC, the transmission interruptions are much smaller than the TFRC flow. For TCP+ELFN flow, although its disconnection period is also short, due to its aggressive bandwidth probing mechanism, it again encounters more frequent interruptions than ADTFRC.

5.3 Application Layer Quality

To evaluate the quality improvement perceived by an end host, we use an application layer metric, *client starvation time* that is defined in [28]. This metric characterizes the situation when the client experience a freezing motion or frame skip during the streaming. We use the StarWar video trace [27] encoded with 70Kbps average source rate in MPEG-4 format. The application layer framing and partial reliability is enabled at the two end hosts. The implementation details are again referred to the technical report [15].

	Single Flow				Aggregated for Two Flows			
	0 m/s	2 m/s	5 m/s	10 m/s	0 m/s	2 m/s	5 m/s	10 m/s
TFRC	14	69	102	178	31	308	1083	4032
ADTFRC	2	15	42	58	4	137	216	367
TCP-ELFN	6	17	39	67	14	187	831	6873

Table 2. *The (aggregated) client starvation time (in seconds) for single MPEG-4 streaming and two simultaneous flows (each flow has average source rate of about 70K bps). Each numbers presented are averaged from 10 repeated runs to filter out randomness of topology and node mobility. The video trace lasts for 1 hours, with receiver playback buffer set to be 96K bytes.*

Observe from Table 5.3, the streaming quality measured by client starvation time is greatly improved by ADTFRC. Compared with standard TFRC or TCP+ELFN, the improvement is especially significant in scenarios when multiple streaming flows run in a mobile network. A detailed trace analysis is available in our technical report [15].

6 Related Work

There are two types of approaches in detecting network congestion in the Internet. One is based on end-to-end measurement and the other on feedback from intermediate gateways in the network. In mobile ad hoc network, Most of the proposals, [1][6][26][24], adopt a network infrastructure oriented approach.

In general, the network assisted approach provides a more direct monitoring of congestion, while in the end-to-end measurement approach, the congestion has to be inferred from metric observation. However, end-to-end measurement maintains the end-to-end semantics of TFRC and provide a convenient implementation that does not need infrastructure support. In this paper, we explore the feasibility of end-to-end based congestion detection in mobile ad hoc network using *multi-metric joint identification*.

Recent studies on the end-to-end measurement based congestion detection are presented in [8][25]. [8] used a single metric based detection such as inter-arrival delay , throughput or packet losses. But they indicated a negative result simply because of too much noise in the end-to-end measurement, especially when node mobility and channel errors are both present. [25] uses packet out-of-order (OOO) to differentiate packet losses due to route changes and network congestion. However, this is based on the assumption that the packet losses can only be contributed by route changes or congestion. In this paper, we tries to provide a more general solution to reliably detect multiple network conditions such as channel errors and network disconnections using end-to-end measurements.

7 Conclusion

Multimedia streaming in a mobile ad hoc network is a challenging task. Not only the network is highly dynamic but also the intermediate nodes are heterogeneous, having varying capacity and battery power. In this paper, we explore an end to end approach and designed a TCP-friendly transport protocol, ADTFRC, to improve the performance of rate adaptation of TFRC in such dynamic network. In particular, ADTFRC uses multiple metrics to jointly detect network states in the presence of measurement noises, so that the sending rate can be adjusted accordingly. In addition, the streaming quality perceived by the end user is further enhanced by the application layer framing and partial reliability capability of ADTFRC. Simulations in NS-2 show that ADTFRC is able to significantly improves the transport performance of real time video streams in a TCP friendly way.

References

1. G. Holland and N. Vaidya, “Analysis of TCP performance over mobile ad hoc networks,” *MOBICOM’99*.
2. P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan, “WTCP: A reliable transport protocol for wireless wide-area networks,” *MOBICOM’99*.
3. H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, “Improving TCP/IP performance over wireless networks,” *MOBICOM’95*.
4. H. Balakrishnan, and R. Katz, “Explicit loss notification and wireless web performances,” *Globecom’98*.
5. Zhenghua Fu, Xiaoqiao Meng, Songwu Lu ”How bad TCP can perform in wireless ad hoc network” IEEE ISCC (IEEE Symposium on Computers and Communications) 2002, Italy, July 2002.
6. J. Monks, P. Sinha and V. Bharghavan, “Limitations of TCP-ELFN for ad hoc networks,” *MOMUC’00*.
7. M. Gerla, K. Tang, and R. Bagrodia, “TCP performance in wireless multihop networks,” *WM-CSA’99*.
8. S. Biaz and N.H. Vaidya, ”Distinguishing congestion losses from wireless transmission losses” *IEEE 7th Int. Conf. on Computer Communications and Networks*, October 1998.
9. L. Brakmo, S. O’Malley, and L. Peterson ”TCP Vegas: New techniques for congestion detection and avoidance” *ACM SIGCOMM 1994*
10. V. Jacobson and M.J.Karels, ”Congestion Avoidance and Control” *ACM Sigcomm 1988*
11. M.Allman, V.Paxson, and W. Stevens, ”TCP Congestion Control” *RFC 2581* April 1999.
12. S. Biaz, N. Vaidya, ”Discriminating congestion losses from wireless losses using inter-arrival times at the receiver” *IEEE ASSET* 1999
13. A. Bakre and B. Badrinath, ”I-TCP:indirect TCP for mobile hosts”, *Proc. 15th International Conf. on Distributed Computing Systems (ICDCS)* May 1995

14. A. Bakre and B.Badrinath ”Implementation and performance evaluation of Indirect TCP” *IEEE Trans. Computers*, Vol. 46, March 1997
15. Z.Fu, X.Meng, S.Lu, ”TCP Friendly Rate Adaptation for Multimedia Streaming in Mobile Ad Hoc Networks”, *Technical Report of Computer Science Department, UCLA, 2003*.
16. S.Floyd, M.Handley, J.Padhye and J.Widmer, ”Equation-Based Congestion Control for Unicast Applications” In *Proceedings of ACM Sigcomm 2000*
17. M.Handley, S.Floyd, J.Padhye and J.Widmer, ”TCP Friendly Rate Control (TFRC): Protocol Specification”, *RFC 3448, January 2003*.
18. D.Bansal and H.Balakrishnan, ”Binomial Congestion Control Algorithms.” In *Proceedings of IEEE Infocom, 2001*.
19. I.Rhee, V.Ozdemir and Y.Yi, ”TEAR: TCP Emulation at Receivers-Flow Control for Multimedia Streaming.”, *Technical Report, NCSU, April 2000*.
20. D.Bansal, H.Baladrishnan, S.Floyd and S.Shenker, ”Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms”, In *Proceedings of ACM Sigcomm 2001*
21. H.Schulzrinne, S.Casner, R.Frederick, V.Jacobson, ”RTP: A Transport Protocol for Real-Time Applications”, *RFC 1889, 1996*
22. S.Raman, H.Balakrishnan, M.Srinivasan, ”ITP: An Image Transport Protocol for the Internet”, *IEEE/ACM Trans. on Networking*, June, 2002.
23. M.Feamster and H.Balakrishnan, ”Packet Loss Recovery for Streaming Video”, *12th International Packet Video Workshop*, Pittsburgh, PA, April 2002
24. J.Liu and S.Singh. ”ATCP:TCP for mobile ad hoc networks”, *IEEE Journal on Selected Areas in Communiations, 19(7):1300–1315* July 2001
25. F.Wang and Y.Zhang, ”Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response”, *In Proceedings of the third ACM international symposium on Mobile ad hoc networking and computing*, 2002, Lausanne, Switzerland.
26. K.Chandran, S.Raghunathan, S.Venkatesan and R.Prakash. ”A feedback-based scheme for improving TCP performance in ad hoc wireless networks”, *IEEE Personal Communications Magazine*, 8(1):34–39,Feb. 2001.
27. F.Fitzek and M.Reisslein, ”MPEG-4 and H.263 Video Traces for Network Performance Evaluation”, *IEEE Network*, Vol.15 No.6, November 2001. Video traces available from the website http://www-tkn.ee.tu-berlin.de/f̃itzek/TRACE/pub.html
28. F.Fitzek, M.Reisslein, ”A Prefetching Protocol for Continuous Media Streaming in Wireless Environments”, *IEEE Journal on Selected Areas in Communications (Special Issue on Mobility and Resource Management in Next Generation Wireless Systems)*, Vol.19, No.6 Pages 2015-2028, October 2001.