The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks

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

Nitin Vaidya
Department of Electrical and Computer Engineering
and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, U.S.A.
Email: nhv@uiuc.edu

Abstract— **Flow control in a mobile ad hoc network (MANET) must face many new challenges such as frequent re-routing and bandwidth variation of the wireless links. TCP's implicit AIMD flow control performs poorly in this environment, because it often cannot keep up with the dynamics of the network.**

In this paper, we explore the potential utility of *explicit* flow control in the MANET domain. To this end, we propose an end-to-end rate-based flow control scheme (called EXACT), where a flow's allowed rate is explicitly conveyed from intermediate routers to the end-hosts in each data packet's special control header. As a result, EXACT reacts quickly and precisely to re-routing and bandwidth variation, which makes it especially suitable for a dynamic MANET network. We also discuss several supporting mechanisms required for such a scheme at the MAC and the transport layers. By ns-2 simulations, we show that EXACT outperforms TCP in terms of fairness and efficiency, especially in a highly dynamic MANET environment.

I. INTRODUCTION

Mobile ad hoc network (MANET) is formed by a group of mobile nodes connected by wireless links. The nodes can talk to each other by direct peer-to-peer wireless communication when they are close to each other, or by multi-hop forwarding via intermediate nodes when they are far away.

As in wireline networks, end-host in MANET must face the non-trivial problem of deciding how fast it can send packets to a destination over the network. This is the *flow control* (or congestion control) problem in networking research. Generally, a flow control scheme has two goals [1]: *efficiency* and *fairness*. Efficiency refers to the property that the aggregated traffic at the bottleneck router should match the available bandwidth of the outgoing links. Fairness refers to the property that each competing flow should get its "fair" share of the available bandwidth.

Over the Internet, TCP's AIMD (Additive Increase Multiplicative Decrease) is the predominant flow control algorithm. It belongs to the *implicit flow control* category, because it measures the network congestion state by performance measurements at the end-hosts (i.e., packet loss), without any help from the network. Due to TCP's wide acceptance and software availability over the Internet, it remains the current de facto flow control standard in MANET as well. However, recent studies have shown that TCP suffers fairness and efficiency problems in this environment (e.g., [2]–[5]). We summarize these problems as follows. First, TCP detects congestion by packet loss events, which is not a reliable congestion signal, because packet loss can be wireless related random loss and mobility related routing loss. Second, TCP's additive increase of congestion window limits its ability to acquire spare bandwidth quickly, which is important after a re-routing event. Third, since TCP only reacts to packet loss, it tends to keep the bottleneck router queue full, which may put the router at risk of dropping packets when the link bandwidth fluctuates. Fourth, TCP's window-based transmission can lead to a burst of packet transmissions when several ACKs arrive at the same time. Although there have been many efforts to enhance TCP performance in MANET (e.g., [2], [3]), the problems mentioned above are fundamental to the implicit approach taken by TCP's AIMD flow control algorithm.

Prompted by the deficiencies of TCP, we attempt to design a more effective flow control scheme in MANET. To this end, we adopt the *explicit flow control* approach, and propose an EXplicit rAte-based flow ConTrol (EXACT) scheme as our solution to the flow control problem in MANET. Here we refer to a scheme where routers provide explicit rate information to the flows. The explicit congestion information is carried in the IP header of each data packet, and is modified by intermediate routers to signal the flow's allowed data sending rate. The rate information is then returned from the receiver to the sender as feedbacks. Our scheme is in part inspired by the rate-based feedback framework of ATM's ABR (Available Bit Rate) congestion control [6] (details in Section VI), but has incorporated a number of mechanisms to fit into the new MANET environment. To the best of our knowledge, this is the first study of using explicit rate-based flow control in MANET.

The rest of this paper is organized as follows. We first outline the design rationales of EXACT in Section II, followed by a detailed description in Section III. The supporting mechanisms are discussed in Section IV, and the simulation and testbed results are reported in Section V. Section VI discusses some related work, followed by a conclusion in Section VII.

II. DESIGN RATIONALES

Since EXACT is a fundamental departure from the traditional implicit flow control approach (e.g., TCP), we describe the design rationales behind our scheme as follows.

A. 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. For instance, when wireless link's bandwidth varies, EXACT is able to convey such variation to the flows quickly, without requiring them to detect the variation only after packet losses. When a flow changes its route as result of mobility, EXACT is able to provide rate signal to the flow immediately along its new path, without requiring the sender to go through the additive probing phase of TCP. Therefore, the router-assisted EXACT scheme is more precise and responsive, which makes it especially suitable in a dynamic MANET environment.

B. Rate-based Transmission

In EXACT, the sender follows the rate information in the feedback packets from the receiver, and hence the packet transmission is rate-based. This alleviates the bursty transmission problem of TCP. Moreover, by using rate-based transmission, the feedback packets can now be sent less frequently if the allocated rate has not significantly changed.

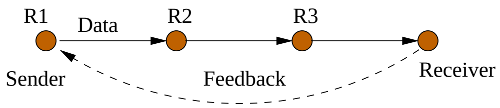
C. Feasibility in MANET

Admittedly, our 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. We believe our EXACT scheme fits precisely into its unique characteristics. First, MANET is often a small scale network for a group of mobile users, such as emergency workers. Second, unlike Internet, there is no "core" router in MANET. Flows are more evenly routed throughout the network rather than going through some hierarchical aggregation points. Another factor is that MANET traffic often displays locality, i.e., a mobile node is more likely to talk to another node physically close to itself. As a result, the number of concurrent flows going through a MANET router is likely to be relatively small. Even with the additional processing overhead, EXACT is a feasible and practical solution in MANET.

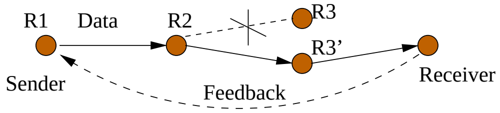
III. EXACT FRAMEWORK

A. Overview

An overview of the EXACT framework is shown in Figure 1(a), where the sender sends a continuous stream of data packets to the receiver. Each data packet carries a special IP header, called *flow control header*, which is modified by the intermediate routers to signal the flow's allowed sending rate. When the packet reaches the destination, the explicit rate information is returned to the sender in a feedback packet. As a result, any bandwidth variation along the path will be returned to the sender within one RTT. Note that the feedback packet may not travel along the same path as the data packets.



(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. 1. Overview of the EXACT flow control scheme.

In the event of re-routing (Figure 1(b)), the first data packet traveling through the new path collects the new allowed rate of the flow. As a result, the sender learns the exact sending rate after only one RTT of delay after re-routing, without having to go through the additive probing phase of TCP.

A packet's flow control header includes two fields: *ER (Explicit Rate)* and *CR (Current Rate)*. ER is the allowed sending rate of a flow. It is initially set at the sender as its maximum requested rate, and subsequently reduced by the intermediate routers to signal its allowed data rate. ER is typically set to infinity for those flows requiring the largest possible bandwidth (e.g., FTP). CR is initially set at the sender as its current sending rate, and modified by the intermediate routers to signal possible rate reduction along the path. Each router remembers the CR of the current flows in its *flow table*, in order to compute each flow's fair share of bandwidth.

Note that EXACT does *not* assume any particular underlying routing protocol in MANET, nor symmetric routes between the sender and the receiver. It is a separate flow control module that can be attached to any routing agent in MANET.

B. End-host's Behavior

We assume end-systems are cooperative. Sender's behavior is as follows:

- Sender sets the ER field as its desired maximum rate and the CR field as its current sending rate, in every data packet it sends out.
- On the reception of a feedback packet, the sender adjusts its sending rate to the rate included in the feedback.

Receiver's behavior is as follows:

- On the reception of a data packet, receiver copies the ER field of the data packet into a small feedback packet, and sends it to the sender.
- An optional *delay-ack* strategy allows the receiver to send a feedback only after receiving a number of data packets, or when the ER has significantly changed.

- Each feedback packet's ER and CR fields are both set to the ER of the incoming data packet, adjusted by the feedback packet's smaller size and its delay-ack strategy, to indicate the feedback packet flow's rates.

On start-up, the sender is allowed to send out packets using a small initial sending rate. Once the first data packet is received and acknowledged by the receiver, the sender then uses the explicit rate in the feedback packet as its sending rate.

C. Router's Behavior

Router plays the central role in our EXACT scheme. A router has four major tasks: 1) keep track of current flows and their sending rates in a *flow table*; 2) measure the current bandwidth of the outgoing wireless links; 3) compute rates for the current flows; and 4) update the header of each passing data packet. Below we discuss details of these tasks.

Each router maintains a *soft-state* flow table in the format of: `<src_ip, src_port, dest_ip, dest_port, next_hop, refresh_time, current_rate>`. The first four fields are used to uniquely identify a flow. On receiving a data packet, the router updates the flow's `next_hop`, `refresh_time`, and `current_rate` to keep an up-to-date view of the flow. As mentioned earlier, the CR field of the packet's flow control header is used to update the flow's `current_rate`. A flow has to refresh itself within a certain period of time; otherwise, it will be purged from the table possibly as a result of re-routing or termination.

The core part of each router is its rate computation algorithm to allocate sending rates for the competing flows. The rate computation, performed locally, is based on the current measured bandwidths of the outgoing links, as well as the current rates of the flows going through the router. Efficiency is achieved by making sure that the flows can fully occupy the outgoing wireless links. Fairness can be achieved by allocating the bandwidth "fairly" to each flows. In this paper we adopt a special *max-min fairness* [7] as our baseline fairness criterion. In max-min fairness, flows with minimum requests are granted their requests first; the remaining bandwidth resource is then evenly divided among the higher demanding flows.

Here we propose to maintain fairness among competing flows according to their *channel time* demands to access the wireless channel, because a wireless channel's bandwidth to different neighboring nodes can be very different, due to location-dependent channel conditions. For example, a flow requesting 1Mbps rate to a neighboring node with 2Mbps actual bandwidth requires the router to dedicate 50% of channel time to the flow; while a flow with the same request to a "better" node with 4Mbps bandwidth requires only 25% of channel time. Note that here the wireless links' bandwidths are dynamically measured at the MAC layer (details in Section IV-A). To represent a flow's resource request, we normalize a flow's requested rate to its next-hop link's bandwidth as $TF_i = r_i/b_i$, where r_i is the flow's data rate (`current_rate` in the flow table), and b_i is the current bandwidth of the link. The max-min allocation is then performed on top of the requests of the flows: $TF_i, i = 1$ to N. Since each flow obtains a throughput proportional to its next-hop link's bandwidth, we call it *bandwidth-proportional max-min fair*.

The local max-min rate computation is as follows [7]: initially the available channel time is $C = 1$ and the set of flows whose demand has been satisfied is empty: $\mathcal{R} = \emptyset$; then we compute the first-level allocated resources as $AR_0 = C/N$, where N is the total number of flows, and we include all the flows with $TF_i < AR_0$ in set $\mathcal{R}$. Next compute $AR_1 = \frac{C-\sum_{i\in\mathcal{R}} TF_i}{N-\|\mathcal{R}\|}$; if for all flows $i \notin \mathcal{R}, TF_i \geq AR_1$, then stop; otherwise, include those flows with $TF_i < AR_1$ in set $\mathcal{R}$, and re-compute the next level AR_2. When the algorithm terminates at level k, the result is a resource allocation AR_k (or denoted as $\bar{AR}$), which is the *largest* request that can be fully satisfied. A request over $\bar{AR}$ can only be granted $\bar{AR}$ of resources. Since $\bar{AR}$ represents the allocation of *channel time*, it should be converted to the allocation of real data rate over link j as: $DAR_j = \bar{AR} * b_j$, where b_j is the measured bandwidth of link j. For those flows going through link j, DAR_j is the maximum data rate each flow can send.

In our scheme, a router immediately computes the rates whenever the previous computation is invalidated by any of the following reasons: 1) arrival of a new flow, 2) purge of an existing flow, 3) change of rate of an existing flow, or 4) change of link bandwidth. This allows the router to quickly react to the dynamics in MANET.

D. Flow Control Header Updates

As mentioned earlier, routers modify the flow control header of each data packet to explicitly signal a flow's allowed sending rate. On receiving a data packet, the router obtains the maximum allocated data rate DAR_j based on the packet's next-hop neighbor j, and updates the packet's flow control header as follows: $ER = min(ER, DAR_j)$; $CR = min(CR, DAR_j)$.

As a result, ER carries the minimum (i.e., bottleneck) allowed sending rate of the routers along the path. The current rate CR is also reduced along the path in order to deliver the upstream bottleneck to downstream routers as soon as possible. This updated CR field is kept in the router's flow table as the flow's current rate.

IV. SUPPORTING MECHANISMS OF EXACT

A. MAC Layer: Dynamic Bandwidth Measurement

In order to perform the rate computation, a router must have knowledge of the current "achievable" bandwidth of the wireless links. Therefore, a dynamic bandwidth measurement mechanism must be in place at the MAC layer.

As an example, we consider a measurement method under the popular IEEE 802.11 DCF MAC layer, which depends on CSMA/CA to coordinate packet transmission using the RTS-CTS-DATA-ACK packet sequence (Figure 2). The throughput of each transmitted packet can be computed as: $TP = \frac{S}{t_r - t_s}$, where S is the size of the packet, t_s is the time-stamp when the packet is ready to be sent at the MAC layer, and t_r is the time-stamp when an ACK is received [8], [9]. Note that the time

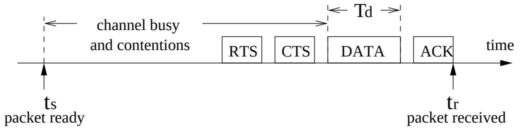


Fig. 2. IEEE 802.11 unicast packet transmission sequence.

interval $t_r - t_s$ includes the channel busy and contention time. The average throughput of the recent packets destined to a neighboring node is used to estimate the achievable bandwidth to that node. We keep separate bandwidth estimates to different neighboring nodes because the channel conditions may be very different. Using ns-2 simulations, we have verified that the dynamic measurement method is feasible and robust [9].

B. *Transport Layer: Safety Counter*

In EXACT, the sender adjusts its sending rate in response to the feedbacks from the receiver. To prevent the sender from over-flooding the network when all the feedback packets are suddenly lost, for example, due to wireless transmission outage or route disruption, a *safety* mechanism must be in place. To this end, we implement a *safety counter* at the sender to limit the number of outstanding packets that have not been acknowledged by the receiver. Note that this is not to be confused with TCP's transmission window. The safety counter does not imply any kind of reliability control, nor any re-transmission of lost packets. It is used to limit the amount of damage the sender can cause to the network when all the feedback packets are lost.

Clearly, the safety counter cannot be too small to limit a flow's sending rate which otherwise would be permitted by EXACT. In [10], we have shown and proved that in MANET, the *bandwidth-delay product* of a path, which is a measurement of the maximum packet carrying capacity of the path, cannot exceed the *round-trip hop-count* of the path. Therefore, in EXACT we use the round-trip hop-count of the network path between the sender and the receiver as the size of the safety counter.

C. *Transport Layer: Route Probing*

Route failure and re-routing are common in MANET. When the network path is unavailable, the sender would send out up to the safety counter number of packets, and enter a *probing state* in which it periodically sends out probing packets at slow speed to see whether a new path is available. When a path becomes available, the receiver should receive the probing packet, and then send a feedback to the sender. On receiving such a feedback, the sender exits the probing state and proceeds with normal packet transmission using the explicit rate of the new path returned by the receiver.

D. *Transport Layer: Reliability Control*

EXACT is a rate-based flow control scheme. It does not implement reliable data transmission by itself. As an *optional* mechanism, reliability can be added on top of the EXACT scheme as an independent mechanism. That is, flow control and reliability control are *de-coupled.* [1] In this paper we choose SACK (Selective ACKnowledgment) as the reliability mechanism because it allows the sender to selectively re-transmit only those missing packets, which should reduce the number of unnecessary re-transmissions due to out-of-order packet delivery in MANET.

V. EVALUATION OF EXACT

In this section, we compare EXACT with TCP's implicit AIMD flow control using the ns-2 (v2.1b8a) simulator. Although EXACT can be implemented with any underlying routing protocol, we choose DSR (Dynamic Source Routing) due to its simplicity. We also report experiments of running EXACT in a MANET testbed using Linux laptops.

A. *Simulation: One-Hop Scenario*

In this set of simulations, all mobile nodes are within one-hop of transmission range to each other inside a 170m by 170m space. The nodes use the "random way-point" mobility model to move around with maximum speed of 20m/s and pause time of 0s. The total number of nodes is 10. The packet size is 1000 bytes.

1) Baseline Behavior: We use two flows, one from node 0 to 1 and the other from node 0 to 2, to demonstrate the basic behavior of the EXACT scheme without any reliability mechanism. The two flows share the same bottleneck router at node 0, and compete with each other in accessing the channel bandwidth. They start with the following sequence: 1) at time 0s, the first flow starts and demands a large bandwidth (in its ER field); 2) at time 50s, the second flow starts also with large bandwidth demand; 3) at time 100s, the first flow reduces its demand to a very small value of 40,000 bytes/s, which gives away rest of the bandwidth to the second flow; 4) at time 150s, the first flow resumes its large bandwidth demand to get back its share of the bandwidth; 5) at time 200s, the second flow stops, leaving all the bandwidth to the first flow.

Three observations can be made from Figure 3. First, max-min fairness is achieved between the two flows by the rate computation algorithm. During time periods 50-100s and 150-200s, each of the two flows obtains nearly the same throughput because bandwidths of the two links are very close. Second, MAC layer link utilization at the bottleneck router is kept close to 1, which shows the efficiency of the scheme. At the same time, the router's queue length is short and stable (not shown here), hence there is no packet queuing loss. Third, a flow can quickly and precisely obtain its share of bandwidth when extra bandwidth is available, without additive probing. These results show that EXACT behaves as we have designed.

2) Comparison with TCP: Now we compare EXACT enhanced with SACK reliability control, against TCP-Reno and TCP-SACK (which are also reliable).

We create two EXACT flows: one from node 0 to 1 and the other from 0 to 2. They share the same bottleneck router

---

[1] Note that although flow control and reliability control are two independent mechanisms, their feedback information are sent back to the sender in the *same* feedback packet, in order to save bandwidth.

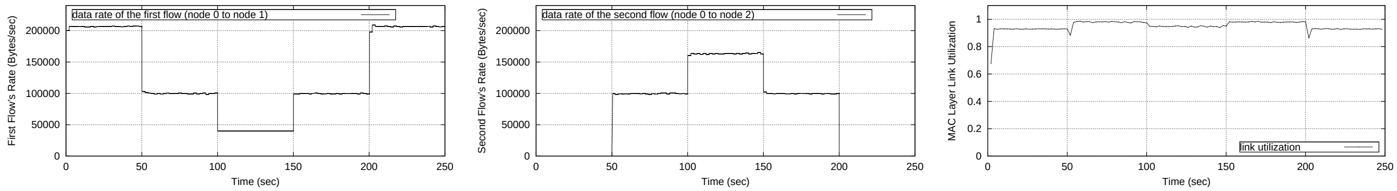


Fig. 3. Two EXACT flows in one-hop scenario. (a. Data rate of first flow; b. Data rate of second flow; c. Mac layer utilization at node 0.)

at node 0, and request large bandwidth. According to max-min fairness, they should obtain nearly the same throughput, because bandwidths of the two links are very close. Figure 4 shows this result, i.e., the sequence number plots of the two flows overlap precisely with each other. As comparison, the results of two TCP-Reno and TCP-SACK flows in the same setting show that they cannot guarantee perfect fairness even in this simple one-hop scenario (Figure 4). Moreover, (not shown here), the queue length at the bottleneck router under EXACT is kept minimum and stable at all times; while under TCP-Reno and TCP-SACK it frequently exceeds the queuing limit, which leads to packet loss. As a result, the total number of reliably transmitted packets under EXACT is 2.4% more than TCP-Reno and 2.5% more than TCP-SACK. We will show that in a multi-hop scenario, this gap is much larger.

B. Simulation: Multi-Hop Scenario

In this set of simulation, we create a MANET with 30 nodes moving in a 1500m by 300m space, using the “random way-point” mobility model with maximum speed of 20m/s and pause time of 10 seconds. This creates a moderately fast moving scenario, and forces the nodes to use long routes in the rectangular area. As a result, re-routing and link bandwidth variation are common.

1) Comparison with TCP: In this experiment we create two EXACT flows enhanced with SACK, both from node 0 to 1. This ensures that the two flows travel along the same path and share exactly the same bottleneck router. As a result, they should expect the same sending rate at all times. This is evident in Figure 5 where the sequence number plots of the two flows overlap precisely with each other. As comparison, two TCP-Reno and TCP-SACK flows in the same setting cannot achieve perfect fairness (Figure 5). At the same time, the total number of reliably transmitted packets under EXACT is 12% more than TCP-Reno and 8% more than TCP-SACK, which demonstrates the efficiency of our scheme.

2) Under Different Degrees of Mobility: To further evaluate the efficiency of explicit rate-based flow control, we compare EXACT with TCP-Reno and TCP-SACK under different mobility patterns. The nodes move around using 20m/s maximum speed as before, but with different pause times (0s, 5s, 10s, 15s, and 20s) to create different levels of network dynamics. For each scenario, we average the total number of reliably transmitted packets over 10 runs for each flow control scheme. The results in Figure 6 show that under all mobility scenarios, EXACT overall outperforms TCP-Reno and TCP-SACK by 42% and 36% more packets, respectively. This demonstrates the efficiency and effectiveness of the EXACT scheme in a dynamic MANET environment.

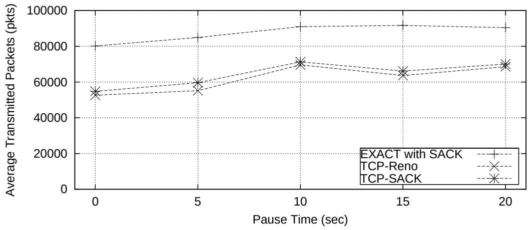


Fig. 6. Comparison of EXACT with TCP under different mobility patterns.

C. Testbed: Computational Overhead

We have implemented EXACT in a small testbed with four Linux laptops arranged in a multi-hop topology. The EXACT module is implemented at the user level using Java. The MAC layer bandwidth measurement is implemented by modifying Lucent IEEE 802.11b pcmcia card’s driver (“wvlan_cs”), and the driver exports the measured bandwidth to the EXACT module. Here we want to comment on the overhead of EXACT in our experiments. Running on a relatively slow Pentium II 266Hz laptop and with 10 concurrent flows with aggregate traffic of 640 kbps, EXACT occupies only 4% of the CPU on average. The modified MAC layer driver occupies less than 15% of CPU. Therefore, EXACT is well within the computing power of today’s mobile devices.

VI. RELATED WORK

In this section, we discuss a number of related work in applying explicit flow control to various network environments. The first scheme is Internet’s Explicit Congestion Notification (ECN [11]), where a router marks a bit in a passing packet’s IP header when incipient congestion is detected. The ECN-bit carries binary information indicating *whether* there is congestion, but not by *how much* the congestion is. In contrast, EXACT provides precise rate signals to the end hosts, without the need for an AIMD-style algorithm at the end-hosts.

The second scheme is ATM Forum’s rate-based flow control for the ABR (Available Bit Rate) service [6]. The goal of ABR flow control is to fully utilize the bandwidth left over from higher priority traffic. In this scheme, explicit rate

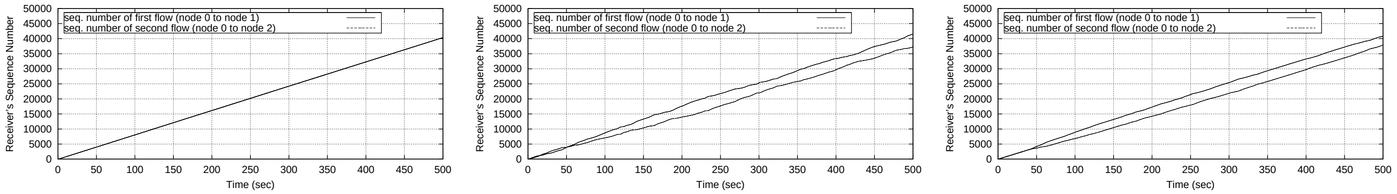


Fig. 4. Sequence number comparison with TCP in one-hop scenario. (a. EXACT with SACK; b. TCP-Reno; c. TCP-SACK)

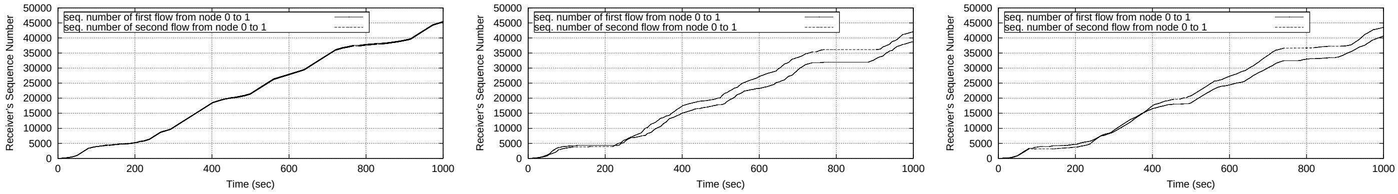


Fig. 5. Sequence number comparison with TCP in multi-hop scenario. (a. EXACT with SACK; b. TCP-Reno; c. TCP-SACK)

control information is conveyed from intermediate switches to the destination using special cells, called RM (Resource Management) cells, and the RM cells are returned to the sender via the same path. ATM's ABR provides a *generic* framework for rate-based feedback flow control, which is the framework adopted by the EXACT scheme. However, as we have shown earlier, in order to adopt this framework, a number of special mechanisms have to be designed in the special MANET environment. Therefore, our work can be considered as a pilot study in applying explicit rate-based flow control to the MANET domain.

The recently proposed ATP protocol [5] is a reliable transport protocol based on the feedback of *maximum packet delay* from the routers. Based on these feedbacks, the end host *infers* its share of the bandwidth at the bottleneck router. Although ATP gives improved throughput than TCP, it cannot guarantee perfect fairness between the competing flows. In contract, EXACT achieves both efficiency *and* fairness by explicitly allocating bandwidth to the competing flows. The clear advantage of ATP is its stateless implementation, however, we have shown that stateful implementation of EXACT is a practical solution in a small scale MANET.

VII. CONCLUSION

We present an explicit rate-based flow control scheme (called EXACT) for the MANET network. In EXACT, routers explicitly notify each flow its allowed data rate, and hence the flows are able to react quickly and precisely to bandwidth variation and re-routing events. Our simulation result shows that, EXACT outperforms TCP's AIMD in terms of fairness and efficiency, especially in a highly dynamic MANET environment. Our testbed experiment also confirms that the stateful implementation of EXACT is well within the computing power of today's mobile devices. Therefore, EXACT is an effective and practical flow control solution for the MANET domain.

ACKNOWLEDGMENT

This work was supported by the DoD Multi-disciplinary 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. Keshav, *An Engineering Approach to Computer Networking*. Reading, MA: Addison-Wesley, 1997.
[2] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," in *Proc. ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, USA, Aug. 1999.
[3] T. Dyer and R. Boppana, "A comparison of tcp performance over three routing protocols for mobile ad hoc networks," in *Proc. ACM Symposium of Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, Long Beach, California, USA, Oct. 2001.
[4] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on tcp throughput and loss," in *Proc. IEEE Infocom 2003*, San Francisco, California, USA, Apr. 2003.
[5] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar, "Atp: A reliable transport protocol for ad-hoc networks," in *Proc. of the 4th ACM Symposium on Mobile Ad Hoc Network and Computing (MobiHoc 2003)*, Annapolis, Maryland, U.S.A., June 2003.
[6] F. Bonomi and K. Fendick, "The rate-based flow control framework for the available bit rate atm service," *IEEE Network*, March-April 1995.
[7] D. Bertsekas and R. Gallager, *Data Networks (2nd Ed.)*. Prentice-Hall, 1992.
[8] 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.
[9] 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.
[10] K. Chen, Y. Xue, and K. Nahrstedt, "On setting tcp's congestion window limit in mobile ad hoc networks," in *Proc. IEEE Intl. Conf. on Communications (ICC 2003)*, Anchorage, Alaska, USA, May 2003.
[11] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ecn) to ip," RFC 3168, Sept. 2001.