

# ALLEVIATING INTRA-FLOW AND INTER-FLOW CONTENTIONS FOR RELIABLE SERVICE IN MOBILE AD HOC NETWORKS

Hongqiang Zhai, Xiang Chen and Yuguang Fang  
Department of Electrical & Computer Engineering  
University of Florida, Gainesville, Florida

## ABSTRACT

*In this paper, we focus on the problems from interaction between the congestion of traffic flows and the MAC layer contentions for reliable service, say TCP traffic. We first characterize the interaction as intra-flow contention and inter-flow contention and illustrate their severe impact on the performance of MANET. Then we propose a novel distributed scheme combining both flow control and media access control to alleviate these two kinds of contentions. The key idea is to differentiate packet transmissions. More specifically, better transmission opportunities are assigned to urgent or backlogged packets, which may be those just received by the downstream nodes or those accumulated at the congested nodes. By doing so, our scheme can promptly and smoothly forward each packet to the destination without incurring explosive increase in the number of control packets at the MAC layer or excessive queueing delay. Extensive simulations in ns-2 demonstrate that our scheme can greatly reduce the MAC layer contention and collision and improve the end-to-end throughput of TCP traffic.*

## INTRODUCTION

To support reliable transport service and hence fully exploit the potential of mobile ad hoc networks (MANET), efficient congestion control is of paramount importance to make MANET viable for many applications in battlefield, disaster rescue and conventions. However, medium contention in the shared channel environment of MANET offers a great challenge to traditional TCP congestion control mechanism and make TCP traffic suffer poor performance in MANET ([1]–[7] and reference therein).

Most of the current work on TCP performance in MANET, such as [6]–[14], focus on end-to-end congestion control mechanism of TCP with or without network layer feedback. To our best knowledge, in recent studies, only [15] comprehensively discussed hop-by-hop congestion control for MANET. However, their system model did not completely

describe the characteristics of MANET. It only considered the channel sharing for those nodes with the same flows passing through, and did not consider other medium contention among nodes which are in the sensing range or interference range of each other. Their studies focused on the theoretical part and did not give a scheme based upon the widely employed 802.11 MAC protocol [16]. They concluded that hop by hop congestion control will get the same end-to-end throughput as end-to-end congestion control.

In this paper, we will show that hop-by-hop congestion control can help TCP traffic get higher throughput by greatly decreasing medium contentions based upon a well designed scheme over the 802.11 MAC protocol. Combined with our previous work in [17], which showed hop-by-hop congestion control could gracefully decrease the impact of irresponsible UDP traffic, significantly improve the maximum end-to-end throughput and reduce the end-to-end delay, our studies of TCP performance in this paper demonstrate that hop-by-hop congestion control is a necessary component to support reliable and stable service for MANET.

To well address the impact of medium contention on the performance of TCP traffic, we characterize the interaction between the congestion of traffic flows and the MAC layer contentions as intra-flow contention and inter-flow contention. Notice that, in MANET, nodes are cooperative to forward each other's packets through the networks. Due to the contention for the shared channel, the throughput of each single node is limited not only by the raw channel capacity, but also by the transmissions in its neighborhood. Thus, each multi-hop flow encounters contentions not only from other flows which pass through the neighborhood, i.e., the *inter-flow contention*, but also from the transmissions of itself because the transmission at each hop has to contend the channel with upstream and downstream nodes, i.e., the *intra-flow contention*.

These two kinds of flow contentions could result in severe collisions and congestion, and seriously limit the performance of multihop ad hoc networks [1], [6], [18]–[20]. The MAC protocol itself could not solve the congestion problem

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and under grant N000140210554.

and it often aggravates the congestion due to the contentions in the shared channel environment. Fang and McDonald [21] studied how throughput and delay can be affected by path coupling, i.e., the MAC layer contention among the nodes distributed along the node-disjoint paths, which is the inter-flow contention. The results demonstrated the need for the control and design of cross-layer interactions and optimization.

To the best of our knowledge, there are no comprehensive studies and good solutions to the interaction problem between traffic congestion and MAC contention in multihop ad hoc networks. Moreover, the scalability issue of 802.11 have not been well addressed and the performance of throughput and end-to-end delay degrades severely in the multihop environment. In this paper, we present a framework of hop-by-hop flow control and medium access control to address the collisions and congestion problem due to the *intra-flow contention* and *inter-flow contention*. Based on this framework, a multihop packet scheduling algorithm is incorporated into the IEEE 802.11 MAC protocol.

The rest of this paper is organized as follows. Section II details the impacts of MAC layer contentions on traffic flows and the resulting problems. Section III describes our scheme and the implementation based on the IEEE 802.11 MAC protocol. In Section IV the performance of our scheme is evaluated for TCP traffic through extensive simulations. Finally, we conclude the paper in section V.

## IMPACTS OF MAC LAYER CONTENTIONS ON TCP TRAFFIC FLOWS

Different from the wired networks where the links are independent of each other, the wireless links may share the same channel resource. Thus the MAC layer contentions come into play when the traffic flows travel through the networks. In this section, we study the intra-flow contention and inter-flow contention problems resulting from interaction between TCP traffic and MAC layer contentions.

The *intra-flow contention* here means the MAC layer contentions for the shared channel among nodes which are in each other's interference range along the path of the same flow. Nodes in a chain experience different amount of competitions as shown in Fig. 1, where the small circle denotes a node's valid transmission range, and the large circle denotes a node's interference range. Thus, the transmission of node 0 in a 7-node chain experiences interference from three subsequent nodes, while the transmission of node 2 is interfered by five other nodes. This implies that node 0, i.e., the source, could actually inject more packets into the chain than the subsequent nodes can forward. These packets are

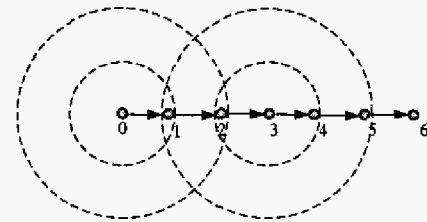


Figure 1. Chain topology

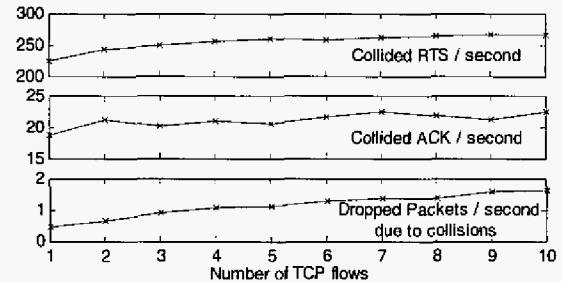


Figure 2. TCP Performance in Chain topology

eventually dropped at the two subsequent nodes. We call this problem as the *intra-flow contention* problem.

Fig.2 demonstrate that TCP traffic introduce a great number of packet collisions. Fig. 3 illustrates more detail why this could happen. Actually, as illustrated in the previous paragraph, node 3, 4, and 5 in a 9-node chain encounter more medium contention than node 1 and 2, thus packet cumulate at these nodes, and keeping them contending for channel access. This results in severe medium collision and a lot of dropped packets. Here, the simulation settings are the same with those of Section IV-A, and different number of TCP flows travel from node 1 to node 9 in a 9-node chain topology.

Besides above contentions inside a multi-hop flow, the contentions between flows could also seriously decrease the network throughput. If two or more flows pass through the same region, the forwarding nodes of each flow encounter contentions not only from its own flow but also from other flows. Thus the previous hops of these flows could actually inject more packets into the region than the nodes in the region can forward. These packets are eventually dropped by the congested nodes. As shown in Fig. 4, where there are two flows, one is from 0 to 6 and the other is from 7 to 12. Obviously node 3 encounters the most frequent contentions and has few chance to successfully transmit packets to its

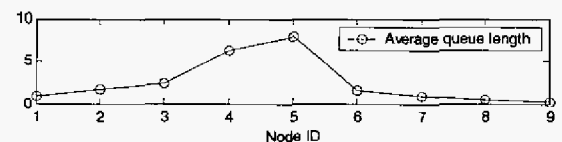


Figure 3. Chain topology with 6 TCP flows

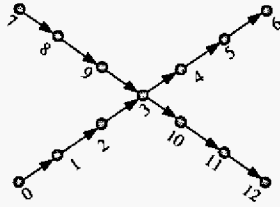


Figure 4. Cross traffic

downstream nodes. The packets will accumulate at and be dropped by node 3, 9, 2, 8 and 1. We call this problem as the *inter-flow contention* problem.

In the shared channel environments in multihop ad hoc networks, these two kinds of contentions are widespread and result in congestion at some nodes, where packets continuously accumulate, which then aggravates the contentions and finally results in packet dropping. This not only greatly decreases the end-to-end throughput but also increases the end-to-end delay due to the long queueing delay.

The intuitive solution to the above problems is to allow the downstream nodes and the congested ones to transmit packets while keeping others silent, and hence smoothly forward each packet to the destination without encountering severe collisions or excessive delay at the forwarding nodes. This motivates us to develop our scheme presented in the next section.

## OPET: OPTIMUM PACKET SCHEDULING FOR EACH TRAFFIC FLOW

### A. Overview

The objective of our scheme is to approach the Optimum Packet scheduling for Each Traffic flow (OPET). Optimum here means that our scheme can achieve optimum packet scheduling for each single traffic flow, which is obtained from the optimal scheduling for chain topology. By reducing the *intra-flow contention* and *inter-flow contention* problems, our scheme OPET can significantly decrease the overhead wasted by those dropped packets at forwarding nodes and hence could significantly improve the end-to-end performance.

OPET consists of two key mechanisms. The first one is to assign high priority of channel access to the current receiver. This could achieve optimum packet scheduling for chain topology and avoid severe intra-flow contentions in each flow. The second one is the hop-by-hop backward-pressure scheduling. The forwarding nodes as well as the source are notified of the congestion and then restrained to send packets to their next hops. This can efficiently reduce the MAC layer overheads due to the *intra-flow contention* and

*inter-flow contention* on those congested nodes. In what follows, we will present the detailed descriptions about these mechanisms.

### B. Rule 1: Assigning High Priority of the Channel Access to the Receiver

In each multi-hop flow, the intermediate node on the path needs to contend for the shared channel with the upstream nodes when forwarding the received packet to the next hop. One way to avoid the first few nodes on the path to inject more packets than the succeeding nodes can forward is to assign high priority of channel access to each node when it receives a packet. This can achieve better scheduling for the chain topology.

For example, in Fig. 1, node 1 has the highest priority when it receives one packet from node 0 and then forwards the packet to node 2. Node 2 immediately forwards the received packet from node 1 and forwards it to node 3. It is the same for node 3 which immediately forwards the received packet to node 4. Because node 0 can sense the transmissions of node 1 and 2, it will not interfere with these two nodes. Node 0 could not send packets to node 1 either when node 3 forwards packet to 4 because node 1 is in the interference range of node 3. When node 4 forwards packet to 5, node 0 could have chance to send a packet to node 1. The similar procedures are adopted by the succeeding nodes along the path. Node 0 and 4 could simultaneously send packets to their next hops, and similar case happens to the nodes which are 4 hops away from each other along the path. Thus, the procedure could utilize 1/4 of the channel bandwidth, the maximum throughput which can be approached by the chain topology [18].

To incorporate this procedure into the IEEE 802.11 MAC protocol, our scheme OPET sets the initial value of the backoff window size of each receiver at 8. When it finishes the transmission, the scheme resets its contention window size to the normal value 32 [16]. The example in Fig. 5 shows the optimum packet scheduling for the chain topology implemented by our scheme. To simplify the illustration how our scheme work, we use chain topology in this figure and following ones, which is conceptually same with any random multihop path in the mobile ad hoc networks.

### C. Rule 2: Backward-Pressure Scheduling

If one flow encounters congestion, it should decrease its sending rate to alleviate the contention for the shared channel. Therefore other flows in the neighborhood could obtain more channel bandwidth to transmit their packets to

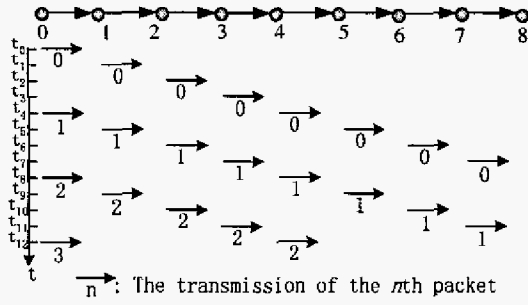


Figure 5. Optimum packet scheduling for chain topology.

achieve higher utilization efficiency of the limited channel resource.

Besides lowering the sending rate of the source, it is necessary to prevent the node, referred to as the *restricted node* in the following discussions, from transmitting packets to its next hop if the latter has already had many packets from the same flow. A multi-hop flow may have already passed through some congested regions. Even if the node has the highest priority to access the channel, it could be blocked by the contentions or other transmissions in the congested region and could not initiate transmission because it senses the channel busy. This will give chances to the preceding hops of the flow to access the channel and continuously forward packets to the blocked node if they are allowed to do so as in the IEEE 802.11 protocol. These packets are eventually dropped by the blocked node and could aggravate the congestion with a lot of unsuccessful (re)transmission attempts in the congested region.

Our scheme is called the *backward-pressure scheduling* because the restriction of transmissions at the restricted node should be passed on to its upstream nodes hop-by-hop until it reaches the source of the flow. The restricted node will accumulate packets in its queue up to the *backward-pressure threshold* which finally causes it to notify the upstream nodes not to transmit more packets to it. When the source of the flow receives this notification, it knows that there is congestion on the path of the flow, and accordingly reduces its sending rate to avoid more accumulated packets dropped at the intermediate nodes on the path.

There must be some ways to resume the transmissions at the restricted nodes when the blocked one can access the channel and the congested region becomes less congested or cleared. There are two methods: one is to retry the transmission at appropriate time at the restricted node, and another is to notify it by its intended receiver. The first method is easy to implement in most of the current protocols without much overhead. The second one requires the protocol to have some receiver-initiated transmission mechanisms but is more accurate and timely to resume the

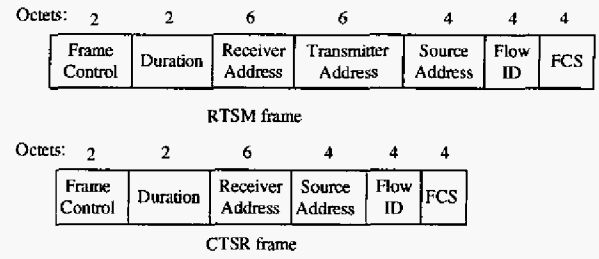


Figure 6. The packet format of RTSM and CTSR

transmissions of the blocked flow.

Our scheme OPET sets the *backward-pressure threshold* as one, which indicates the upper limit of number of packets for each flow at each intermediate node. As discussed before, the optimum chain throughput in the IEEE 802.11 MAC protocol is 1/4 of the chain bandwidth and therefore the optimum threshold for the backward-pressure objective is 1/4, which is similar in operations for any single path. Since 1/4 is difficult to implement in the actual protocol, we select the nearest integer 1 as the value of this threshold.

The backward-pressure scheduling procedure takes advantage of the RTS/CTS exchange in the IEEE 802.11 MAC protocol to restrict the transmission from the upstream nodes. A negative CTS (NCTS) should respond the RTS when the intended receiver has reached the *backward-pressure threshold* for this flow. To uniquely identify each flow, RTS for the multi-hop flows (RTSM) should include two more fields than RTS, i.e., the source address and the flow ID. RTS for the last hop transmission is not necessary to include these two fields, because its intended receiver is the destination of the flow which should not limit its preceding hop from sending packets to itself. The NCTS packet has the same format as CTS except the different value in the frame type field. The format of RTSM is shown in Fig. 6.

In the backward-pressure scheduling scheme, each node needs to maintain a table, i.e., the *flow-table*, to record the information of the flows which currently have packets in the outgoing queue. When the node receives a packet, it creates or updates the corresponding entry for the flow that this packet belongs to in the flow-table. On the contrary, when each packet of one flow is transmitted to the next hop, the corresponding entry in the flow-table is updated or deleted if the flow has already no packet at this node. Thus the maximum size of the table is the queue size if all packets in the queue belong to different flows and the queue is full. The flow information of each table entry includes the *source-address*, *flow-ID*, *number-of-packets* in the queue, *restriction-flag*, and *restriction-start-time*. The restriction-flag indicates whether the node is not allowed to forward

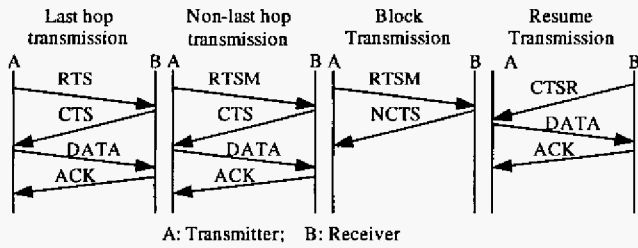


Figure 7. Message sequence for packet transmission

the packet of this flow to the downstream node and the restriction-start-time indicates when the restriction starts.

Our scheme OPET adopts the receiver-initiated transmission mechanism to resume the *restricted* transmission. It uses three-way handshake CTS/ DATA/ ACK instead of the normal four-way handshake RTS/ CTS/ DATA/ ACK, because the downstream node has already known that the restricted node has packets destined to it. The CTS to resume the transmission (CTSR) should include two more fields than CTS, the source address and the flow ID, to uniquely specify the flow as shown in Fig. 6. CTSR as well as CTS has no information about its transmitter as that in RTS. The two fields, i.e., the source address and the flow ID, are used to uniquely specify the next hop that the flow should pass through, hence we assign different flow IDs to the flows from the same application but with different path if multipath routing is used. The procedure of transmitting CTSR is similar to that of RTS and allows multiple retransmissions before dropping it. Different message sequences at different situations are shown in Fig. 7.

To use the receiver-initiated transmission mechanism, we must consider that the mobility in ad hoc networks could result in link breakage followed by the transmission failure of CTSR. And CTSR may be also collided for several times and be dropped. The blocked node should drop CTSR after multiple retransmissions like the mechanism for RTS transmission. The restricted node should start a timer and begin retransmission if its intended receiver has not sent CTSR back in a long period, which we set one second in our study of the proposed scheme.

Each node also needs to maintain a table, i.e., the *block-table*, to record the blocked flow information for the purpose of resuming transmission. When the node transmits a NCTS frame to the upstream node, it creates or updates the corresponding entry in the block-table. When it successfully resume the transmission by sending a CTSR frame to the upstream node, or the CTSR transmissions have failed for 7 times (same as the RTS retransmission times limit), it deletes the corresponding entry in the block-table. The maximum table size is less than or equals to the queue

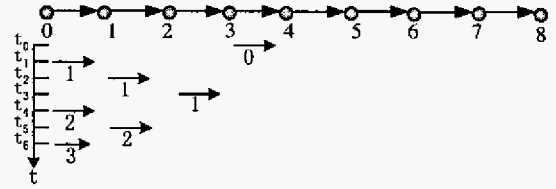


Figure 8. The packet scheduling when congestion occurs.

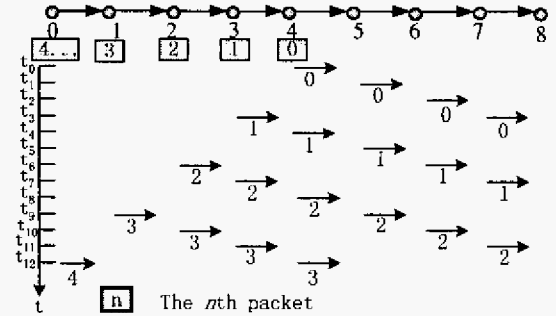


Figure 9. The packet scheduling to resolve the congestion.

size because the number of the flows which have packets at this node is less than or equals to the queue size, and the node will immediately start resuming the blocked flow at the upstream node after it transmit a packet of this flow to the downstream node. The flow information of each table item contains the *source-address*, *flow-ID*, *upstream-node-address*, and *block-flag*.

One simple example to illustrate how our scheme works is shown in Fig. 8 and Fig. 9. The congestion can result from the interference or contention from any crossing flow such that node 4 can not grab the channel in time. For example, when node 4 has congestion and it could not forward packet 0 to its downstream node 5 as shown in Fig. 8, the flow along the chain will accumulate one packet at each node from node 1 to node 4 and then prevent the nodes 0, 1, 2 and 3 from contending for the channel in order to reduce the contention at the congested node 4. Similarly the upstream nodes of all other crossing flows yield the transmission opportunity to the congested node. Thus node 4 can quickly forward the backlogged packets and hence the congestion is eliminated. Then the transmission will be resumed by the congested node as shown in Fig. 9.

It is important to note that the control overhead of the backward-pressure scheduling is relatively low. The information of backward-pressure is carried by the original message sequence RTS/CTS in IEEE 802.11. And the blocked flows are resumed by a three-way handshake procedure with less overhead than the original four-way handshake. Moreover, our scheme only maintains several short entries for each active flow which has at least one

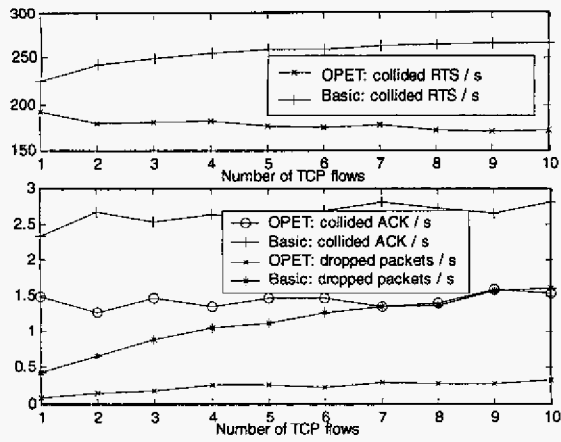


Figure 10. Collisions of TCP traffic in chain topology

packet queueing up at the considered node. We observe that in a mobile ad hoc network, the number of active flows per node is restricted by the limited bandwidth and processing capability, and hence is much smaller than that in the wired networks, thus the scalability problem should not be a major concern in our scheme.

## PERFORMANCE EVALUATION

We now evaluate the performance of our scheme OPET and compare it with the IEEE 802.11 scheme. The simulation tool is one of the widely used network simulation tools – *ns-2*. We use pre-computed shortest path and there is no routing overhead. The propagation model is two-ray ground model and the channel bandwidth is 2 Mbps. The transmission range is about 250 meters, and the sensing range is about 550 meters.

In what follows, our scheme will be referred to as the Optimum Packet Scheduling for Each Flow (OPET), and the IEEE 802.11 protocol without the packet scheduling algorithm will be referred to as the Basic scheme.

### A. Chain Topology

We first investigate how well our scheme performs in the 9-node chain topology with different number of TCP flows. The nodes are separated by 200 meters in the chain. The TCP flows starts from the beginning of the chain, i.e., node 1, to the end of the chain, i.e., node 9.

Fig. 10 shows that our scheme OPET can reduce the packet collision by about 40% for both RTS and ACK frames. And the number of dropped TCP packets is also reduced by about 80%. This verifies that the hop-by-hop congestion control can effectively reduce a lot of medium contention and collision.

Fig. 11 demonstrates that OPET can improve the aggregate throughput of TCP flows by about 5%. And the fairness is

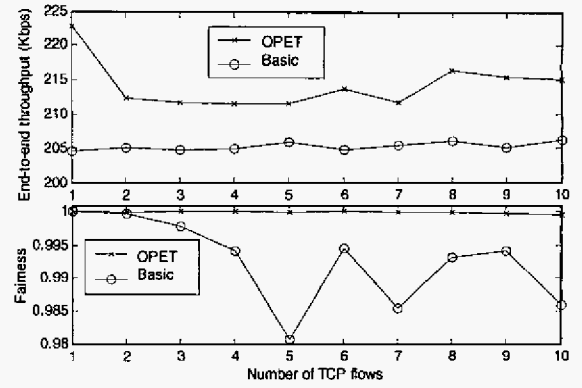


Figure 11. Throughput and fairness in chain topology

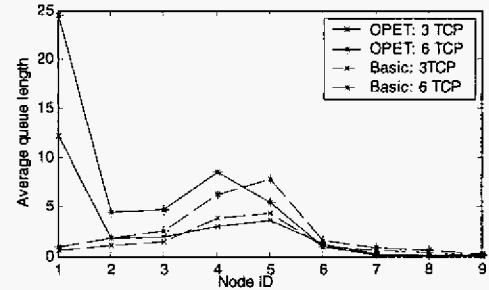


Figure 12. Queue length for TCP traffic in chain topology

even better than the Basic scheme. Here, the fairness index is calculated by the Jain's index, i.e.,  $f = (\sum_{i=1}^n x_i)^2 / (n \cdot \sum_{i=1}^n x_i^2)$ , where  $x_i$  denotes the end-to-end throughput of the  $i$ th flow.

Fig. 12 illustrates the necessity for TCP source to reduce the sending rate when it observes that the outgoing queue builds up. This demonstrates that the hop-by-hop congestion control could notify the source of the congestion status by a very simple and easily measured metric, i.e., queue length at the source node.

### B. Grid Topology

In this subsection, we will investigate the performance of OPET in a larger network with grid topology, where inter-flow contention is a common phenomenon. The grid topology is shown in Fig. 13, where there are total 100 nodes, and one-hop distance is set as 200 meters. 16 TCP flows with 8 horizontal ones and 8 vertical ones, as shown in Fig. 13, run for 300 seconds in the simulation.

TABLE I. Simulation Results for Grid Topology

	OPET	Basic
Aggregate end-to-end throughput (Kbps)	603	547
Collided RTS / second	802	1015

Table I shows the simulation result which demonstrates

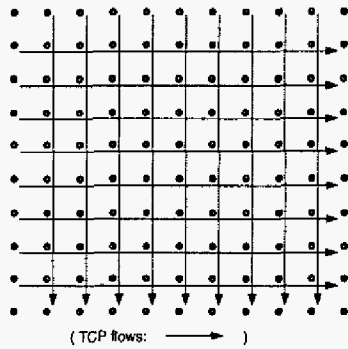


Figure 13. Grid Topology with 16 TCP flows

that OPET improves the aggregate end-to-end throughput by about 10% with about 26% less collided RTS packets. This further verifies that OPET can perform much better than the Basic scheme when the intra-flow and inter-flow contention coexist.

In summary, our scheme could reduce a lot of medium contention in MANET and hence improve the aggregate end-to-end throughput of TCP traffic. Energy efficiency is also improved by delivering higher throughput with less packet collisions.

## CONCLUSIONS

In this paper, we focus on the interaction between medium contention and traffic congestion for TCP traffic in MANET. We first present our observation that the poor performance of the IEEE 802.11 is attributed to the *intra-flow contention* and *inter-flow contention* in multihop ad hoc networks. In order to reduce these two kinds of contentions, we have built a framework for distributed hop-by-hop flow control and media access control, based on which a multihop packet scheduling algorithm, i.e., OPET, is proposed for the 802.11 based multihop wireless ad hoc networks.

Extensive simulations verify that our scheme OPET could greatly mitigate collisions at the MAC layer and has a much better multihop packet scheduling than the IEEE 802.11 protocol. Thus it can always achieve stable and high throughput with much less packet collisions and higher energy efficiency.

Combined with our previous studies for UDP traffic in [17], the studies in this paper demonstrate that hop-by-hop congestion control with careful design over the IEEE 802.11 MAC protocol can not only significantly alleviate the impact of irresponsible UDP flows, but also well support the reliable service over TCP flows by significantly improving the end-to-end throughput and reducing the medium collisions.

## REFERENCES

- [1] S. Xu and T. Safadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?" *IEEE Communications Magazine*, pp. 130-137, June 2001.
- [2] X. Chen, H. Zhai, J. Wang and Y. Fang. TCP performance over mobile ad hoc networks. *Canadian Journal of Electrical and Computer Engineering*, Vol. 29, No. 1/2, p129-134, January/April 2004.
- [3] Z. Fu, X. Meng, and S. Lu. How Bad TCP can Perform in Mobile Ad-Hoc Networks. *IEEE Symposium on Computers and Communications*, July 2002.
- [4] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang. TCP over Wireless Multihop Protocols: Simulation and Experiments. *Proc. IEEE ICC'99*, Jun. 1999.
- [5] M. Gerla, K. Tang, and R. Bagrodia. TCP Performance in Wireless Multihop Networks. *Proc. IEEE WMCSA'99*, Feb. 1999.
- [6] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," *Proc. IEEE INFOCOMM'2003*, March 2003.
- [7] K. Chen, Y. Xue, and K. Nahrstedt. On setting TCP's congestion window limit in mobile ad hoc networks. *Proc. IEEE ICC 2003*, May, 2003.
- [8] 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*, 8 (1):34-39, February 2001.
- [9] G. Holland and N. H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Proc. ACM MOBICOM'99*, Aug. 1999.
- [10] J. P. Monks, P. Sinha and V. Bharghavan, "Limitations of TCP-ELFN for ad hoc networks," *MOMUC 2000*.
- [11] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE JSAC*. Vol.19 No.7, July 2001.
- [12] T. D. Dyer and R. V. Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks," *Proc. ACM Mobihoc*, Oct. 2001.
- [13] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response," *Proc. ACM MobiHoc*, Jun 2002.
- [14] K. Xu, M. Gerla, L. Qi, and Y. Shu, Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED. *Proc. ACM MobiCom*, Sep. 2003.
- [15] Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. *Proc. IEEE INFOCOM*, 2004.
- [16] *IEEE standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, ISO/IEC 8802-11:1999(E), Aug. 1999.
- [17] H. Zhai, J. Wang, and Y. Fang. Distributed Packet Scheduling for Multihop Flows in Ad Hoc Networks. *Proc. IEEE WCNC'04*, March, 2004.
- [18] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee and R. Morris, "Capacity of Ad Hoc Wireless Network," *Proc. ACM MobiCom 2001*, July 2001.
- [19] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu, and J. Jetcheva, "A performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols," *Proc. ACM MobiCom'98*, Oct. 1998.
- [20] C. Perkins, E.M. Royer, S.R. Das, and M.K. Marina, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *IEEE Personal Communications*, pp. 16-28, Feb. 2001.
- [21] Y. Fang and A.B. McDonald, "Cross-layer performance effects of path coupling in wireless ad hoc networks: power and throughput implications of IEEE 802.11 MAC," *Proc. 21st IEEE International Performance, Computing, and Communications Conference*, April 2002.