Distributed Packet Scheduling for Multihop Flows in Ad Hoc Networks

Hongqiang Zhai, Jianfeng Wang and Yuguang Fang
Department of Electrical & Computer Engineering
University of Florida
Gainesville, Florida 32611-6130
Tel: (352) 846-3043, Fax: (352) 392-0044
E-mail: zhai@ecel.ufl.edu, jfwang@ufl.edu, and fang@ece.ufl.edu

Abstract— **In wireless multihop ad hoc networks, nodes need to contend for the shared wireless channel with their neighbors, which could result in congestions and greatly decrease the end-to-end throughput due to severe packet loss. Several recent papers have indicated that the IEEE 802.11 fails to achieve the optimum schedule for this kind of contentions. In this paper, we present a framework of multihop packet scheduling to achieve maximum throughput for traffic flows in the shared channel environment. The key idea is based on the observation that in the IEEE 802.11 MAC protocol the maximum throughput for chain topology is 1/4 of the channel bandwidth and its optimum packet scheduling is to allow simultaneous transmissions at nodes which are four hops away. The proposed fully distributed scheme generalizes this optimum scheduling to any traffic flows which may encounter intra-flow contentions and inter-flow contentions. Extensive simulations indicate that our scheme could perform well and achieve high throughput at light to heavy traffic load while the performance of the original IEEE 802.11 MAC protocol greatly degrades when the traffic load becomes heavy. Moreover, our scheme also achieves much better and more stable performance in terms of delay, fairness and scalability with low and stable control overhead.**

I. Introduction

In wireless multihop ad hoc networks, nodes have to cooperate 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 by the neighborhood, i.e., the *inter-flow contentions*, 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 contentions*. This effect could result in congestions at some nodes and seriously limit the performance of ad hoc networks especially in terms of end-to-end throughput.

Li. et. al [1] has found that the IEEE 802.11 does a reasonable job of scheduling packet transmission with some traffic patterns but its performance greatly degrades under heavy load for the chain topology. In fact, we will show in later sections that 802.11 fails to achieve the optimum scheduling for all multihop flows whose end-to-end throughput even degrades to zero under heavy traffic load. This severely impacts the scalability of the ad hoc networks.

To alleviate the congestion, there are quite a few papers discussing the dynamical load balancing algorithms. Lee and Gerla presented a dynamic load-aware routing algorithm (DLAR) [2] which uses the traffic load of the intermediate nodes as the main route selection criterion. It periodically monitors the congestion status of active data sessions and dynamically reconfigures the routes that are being congested. Lee and Campbell presented a hotspot mitigation protocol (HMP) [3] where hotspots represent transient but highly congested regions. HMP balances resource consumption among neighboring nodes by suppressing new route requests and rate controlling TCP flows. These solutions focus on routing algorithms and do not fully consider the MAC layer contentions which result in different possibility of channel access at the neighboring nodes.

Ye et.al [4] presented two MAC layer enhancements, i.e., quick-exchange and fast-forward, to address self-contention in ad-hoc networks. They are similar to the packet fragmentation of the 802.11 MAC protocol which uses ACK to convey the immediate negotiation information for the next DATA/fragment transmission. Quick-exchange allows the receiver to return a DATA packet to the sender and fast-forward includes an implicit RTS to the next hop. They could save some transmission negotiation procedures, i.e., the RTS/CTS exchanges, and also reduce the contentions during the succeeding DATA transmission, but do not address the congestion problem due to the MAC layer contentions.

Li and Kanodia et al. [5] [6] presented two schemes, i.e., the distributed priority scheduling and the multi-hop coordination, to assign different priorities using different backoff window size to packets for accessing the channel. The packet with highest priority has larger probability to access the channel than neighboring nodes and hence encounters fewer contentions. Their schemes could better satisfy the end-to-end QoS target than the IEEE 802.11 by considering MAC layer contentions, but do not address the congestion problem either.

To the best of our knowledge, there are no comprehensive study and good solutions to the packet scheduling of multihop

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

traffic flows along their selected paths in the shared channel environment. In this paper, we present a framework dealing with flow control over the MAC layer and queue management to address the congestion problem due to the *intra-flow* and *inter-flow contentions*. Based on the framework, a multihop packet scheduling algorithm is incorporated into IEEE 802.11 MAC protocol. The salient feature here is to generalize the optimum packet scheduling of chain topology, which allow nodes four hops away to transmit simultaneously, to any traffic flows in general topology. It turns out that our scheme could maintain stable performance with high throughput independent of traffic status, and improve the maximum throughput by several times. At the same time, it also improves fairness among flows, and has much smaller delay and much less control overhead compared to IEEE 802.11 MAC protocol. Moreover it is scalable for large networks where there are more multihop flows with longer paths.

The rest of this paper is organized as follows. Section II details the impact of MAC layer contentions on traffic flows and the resulting problems. Section III introduces our scheme and the implementation based on the IEEE 802.11 MAC protocol. Section IV evaluates the performance of our scheme through simulation. Finally, we conclude the paper in section V.

II. Impact of MAC Layer Contentions on Traffic Flows

The *intra-flow contentions* discussed here is 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. Li et al. has observed that IEEE 802.11 fails to achieve the optimum chain scheduling [1]. 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 3 subsequent nodes, while transmission of node 2 is interfered with by five other nodes. This means that node 0, i.e., the source, could actually inject more packets into the chain than the subsequent nodes can forward. These packets are eventually dropped at the two subsequent nodes. We call this problem as *intra-flow contention problem*.

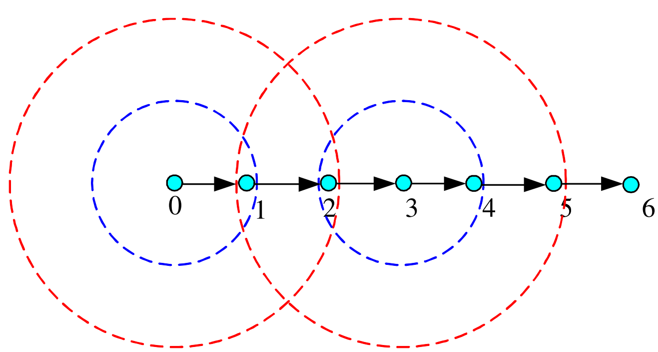

Fig. 1. Chain topology

Besides above contentions inside each multi-hop flow, the contentions between flows could also seriously decrease the delivered 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. 2 where there are two flows, one is from 0 to 6 and the other is from 7 to 12. Obviously node 3 is the most congested one. The packets will accumulate at and be dropped by node 3, 9, 2, 8 and 1. We call this problem as *inter-flow contention problem*.

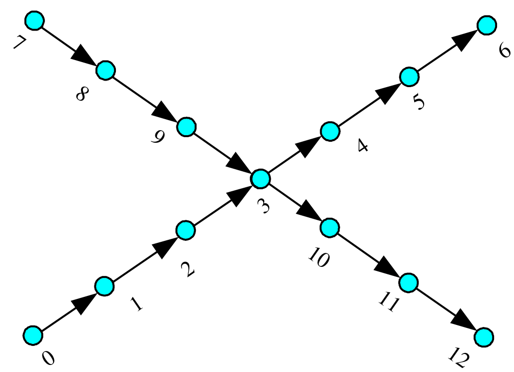

Fig. 2. Cross traffic

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

III. Distributed Packet Scheduling for Multihop Flows

A. Overview

The objective of our scheme is to achieve *O*ptimum *P*acket scheduling for *E*ach *T*raffic flow (OPET), and hence greatly increase end-to-end throughput and decrease end-to-end delay of multihop flows. By alleviating the *intra-flow contention* and *inter-flow contention problems*, our scheme OPET greatly reduces the resource wasted by those dropped packets at forwarding nodes and thus could significantly improve the end-to-end performance.

OPET includes four 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 a lot of 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 are restrained to send packets to their next hops. This efficiently reduces the MAC layer contentions due to *intra-flow contentions* and *inter-flow contentions* on those congested nodes. The third one is not to allow the source node to occupy the whole outgoing queue space, which could efficiently prevent the irresponsible applications from injecting more packets than the network could forward, and leave more queue space for other flows passing this node. The fourth one is the Round Robin scheduling for the queue management which provides fairness among flows which could in fact allocate the same bandwidth to the flows passing by the same congested node on their paths. It also addresses the intra-flow contentions by not sending more packets than the next hop could forward at a time.

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 previous 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 succeeding nodes can forward is to assign high priority of channel access to each node when it receives a packet. This can achieve optimum scheduling for 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 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 next hop, and similar case happens to nodes which are 4 hops away from each other along the path. Thus the procedure could utilize 1/4 of the channel bandwidth, maximum throughput which can be approached by the chain topology [1].

To incorporate this procedure into the IEEE 802.11 MAC protocol, our scheme OPET sets the initial value of the contention window size of each receiver at a much smaller value 4. When it finishes the transmission, the scheme resets its contention window to the normal value 32.

Rule 1 only considers the interference in a single flow. If the next hop of the current receiver is busy or interfered by other transmission, the receiver cannot access channel even with the highest priority to access the channel. So we introduce the backward-pressure scheduling to deal with the inter-flow contentions.

C. Rule 2: Backward-Pressure Scheduling

If one flow encounters congestions, it should slow its sending rate to reduce its contention for the shared channel. Therefore other flows in the neighborhood could obtain more channel bandwidth to transmit their packets to achieve higher utilization efficiency of the limited channel resource.

Besides reducing 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 many packets of the same flow. A multi-hop flow may pass 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 chance to the previous hops of the flow to access the channel and continuously forward packets to the blocked node. These packets are eventually dropped by the blocked node and could aggravate the congestion in the congested region.

Our scheme is called backward-pressure scheduling because the restriction of transmission at the restricted node should be passed to its upstream node 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 node 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 of the path.

There must be some ways to resume the transmission at the restricted nodes when the blocked one can access the channel and the congested region becomes less congested or idle. 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-initiating transmission mechanisms but it is more accurate and timely to resume the transmissions of the blocked flow.

The backward-pressure scheduling should be implemented for each flow instead of for each source or each upstream node. The per-source based backward-pressure scheduling reduces not only the sending rate of the congested one but also that of all other flows from the same source. The per-node based backward-pressure scheduling prevents each node along the path of the congested flow from transmitting any packets to its next hop. Both of them are undesirable in the shared channel environment because it is unfair to other flows which may have different paths and decrease the utilization efficiency of the channel bandwidth.

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. It is similar for any single random path. Since 1/4 is difficult to be implemented 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. 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, 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 previous hop from sending packets to itself.

To resume the transmission at each restricted node, our scheme OPET adopts the receiver initiating transmission mechanism. It uses three-way handshake CTS/DATA/ACK instead of the normal four-way handshake RTS/CTS/DATA/ACK, because it already knows the intended sender has packets to it. The CTS to resume the transmission (CTSC) should include two fields, the source address and the flow ID, to uniquely specify the flow. The procedure of transmitting CTCS is similar to that of RTS and allows multiple retransmissions before dropping it.

To use the receiver initiating transmission mechanism, we must notice that the mobility in ad hoc networks could result in link breakage followed by the transmission failure of CTSC. And CTSC may be also collided for several times and be dropped. The routing layer may have some ways other than the detection of the transmission failures of DATA packets at MAC layer to know whether its neighbor is in its transmission range or not, such as the hello messages in AODV [7]. The node could keep the failed CTSC in record if the link is not broken and retry it after finishing transmission of another packet in the queue. And the restricted node should remove the packets in the queue if the routing layer indicates its next hop is broken. If there is no such hello mechanism in routing algorithm, the blocked node should drop CTSC after multiple retransmissions. The restricted node should start a timer and begin retransmission if its intended receiver has not sent CTSC back in a long period, which we set as one second in our simulation.

D. Rule 3: Source Self-Constraint Scheme

Adopting the backward-pressure scheduling, the packets can only be accumulated at the source node. The application at the source should slow its sending rate if the number of its packets reaches the *source-flow threshold* in the outgoing queue. If it fails to do so, the queue should drop the succeeding packets from it. This could prevent the congested flow from occupying the whole queue space, thus other flows could always have chance to utilize the queue space and transmit packets.

Our scheme OPET sets the *source-flow threshold* as the smallest integer greater than $c+h/4$, where h is the hop count for each flow. The quantity c indicates the maximum burst of the packets that the queue can tolerate for the flow. $h/4$ comes from the optimum scheduling of the chain topology which allows simultaneous transmission at nodes which are 4 hops away. This threshold is applied to both UDP and TCP flows. For TCP flows, Chen et al. [8] has discovered in their simulation that TCP's congestion window should be less than kN if considering transmission interference at the MAC layer, where $1/8 < k < 1/4$, and N is round-trip hops. So $c + h/4$ should work for TCP flows if they set their congestion window limit less than the upper bound kN.

E. Rule 4: Round Robin scheduling

Our scheme OPET adopts the flow-based Round Robin scheduling in the queue management. In the optimum scheduling, each multihop flow at the source node and the forwarding nodes should wait the next few hops to finish forwarding the received packets before transmitting new ones to avoid intra-flow contentions. That is to say, they do not need to transmit the next packet of the same flow immediately. Thus it could begin to deal with the packets of other flows. For the single hop flow, the source node may continuously transmit multiple packets generated from the same application if using FIFO scheduling because the source node could have as many packets as the application injects into the queue and there is no back pressure packet, i.e., NCTS, for the single hop flow. This is unfair to other flows which pass through this node and they obtain much less throughput than the single hop flow. Round Robin could alleviate this unfairness by allowing flows to access channel one by one.

IV. 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.

In the simulations, 60 nodes are randomly placed in a 1000m x 1000m area. The source of each flow randomly selects one node as the destination, which is at least the minimum hops away, i.e., 1 or 3 hops. There are total 30 flows with the same CBR/UDP traffic in the network. The size of each DATA packet is 1000 bytes. All results are averaged over 30 random simulations lasting 300 seconds of simulated time each.

In our simulations, three important performance metrics are evaluated.

Aggregated end-to-end throughput – The sum of data packets delivered to the destinations.

Average End-to-end delay – The average end-to-end delay of all packets which reach the destinations.

Normalized control overhead – The ratio of the number of all kinds of control packets including RTS, CTS, NCTS, CTSC and ACK to the sum of hop count passed by those successfully delivered packets.

Fairness index – The commonly used fairness index for all flows, i.e.,

$$f_i = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} x_i^2} \quad (1)$$

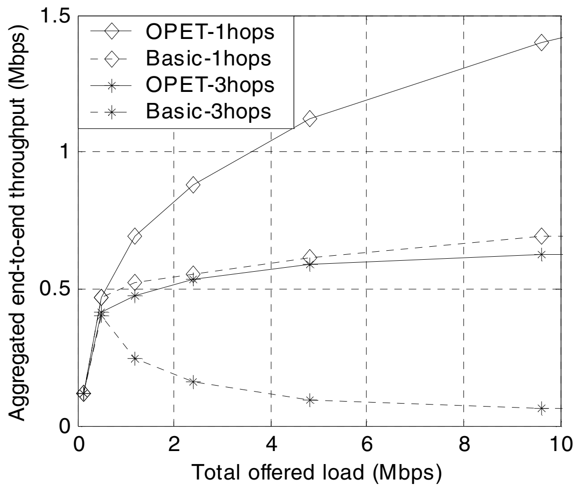

Fig. 3. Aggregated end-to-end throughput

where x_i denotes the end-to-end throughput of the ith flow.

In the following figures, 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. Aggregated Throughput and Fairness

We observe from Fig. 3 that when the minimum hops for each flow increase, the aggregated end-to-end throughput of both protocols decreases. This is reasonable because packets of multihop flows with longer path have to pass more links and thus consume more resource for the same arriving traffic.

For the random traffic without hop count limitation, our scheme OPET could improve the end-to-end throughput by 100% under heavy traffic. This is because that OPET reduces a lot of channel contentions due to the *intra-flow contentions* and *inter-flow contentions*, and there are much less accumulated packets which are finally dropped by the forwarding nodes. The reason that Basic scheme could maintain certain throughput under heavy traffic is that IEEE 802.11 MAC protocol gives preference to those one or two-hop flows which has no or much less contentions from hidden terminals. These flows could capture the whole bandwidth under heavy traffic which contributes to the aggregated end-to-end throughput. However, other flows with longer paths are starved with zero throughput as shown in Fig. 4, which also shows improved fairness in OPET.

If source-destination pairs of all flows are at least 3 hops away, OPET could still maintain maximum end-to-end throughput at heavy traffic status while Basic scheme only obtain almost zero end-to-end throughput. In Basic scheme, the *intra-flow contentions* could allow the sources of multihop flows to inject more packets into the network than the network can forward. The *inter-flow contentions* makes the situation worse. It's not surprising in Basic scheme that the longer path the flow has, the lower the end-to-end throughput it can achieve. By greatly reducing the *intra-flow and inter-flow contentions*, our scheme could always maintain the high end-to-end throughput for all flows at any traffic status.

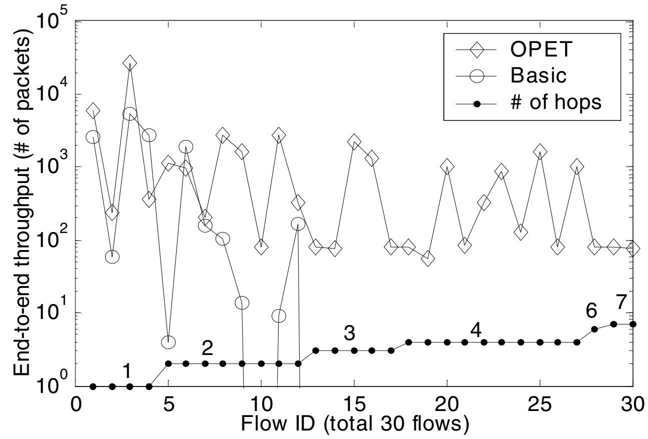

Fig. 4. One random example to illustrate throughput distribution among flows

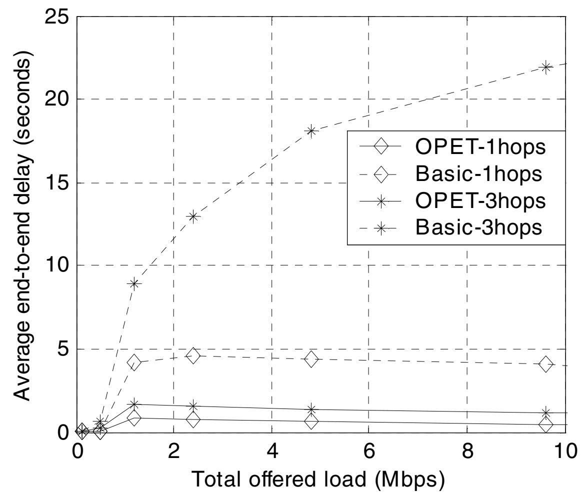

Fig. 5. Average end-to-end delay

B. End-to-End Delay

Fig. 5 shows that OPET has much smaller end-to-end delay than the Basic scheme. Also, for multihop flows, our scheme provide stable end-to-end delay in spite of the traffic status, while in the Basic scheme, the end-to-end delay rapidly increases along with the offered load. This is because that OPET reduces a lot of accumulated packets in the outgoing queue at each node and thus it greatly reduces the queueing delay. In addition, OPET reduces the contentions from the *intra-flow and inter-flow contentions*, which could also decrease the delay at the MAC layer to access the channel.

C. Normalized Control Overhead

Fig. 6 shows that OPET could maintain small and stable normalized control overhead. The Basic scheme has much higher control overhead which rapidly increases with the

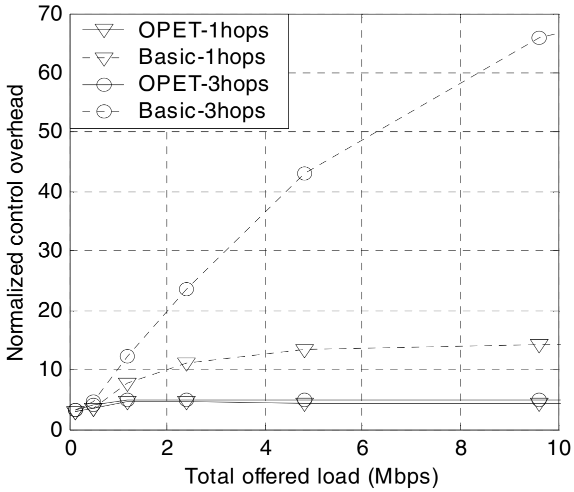

Fig. 6. Normalized control overhead

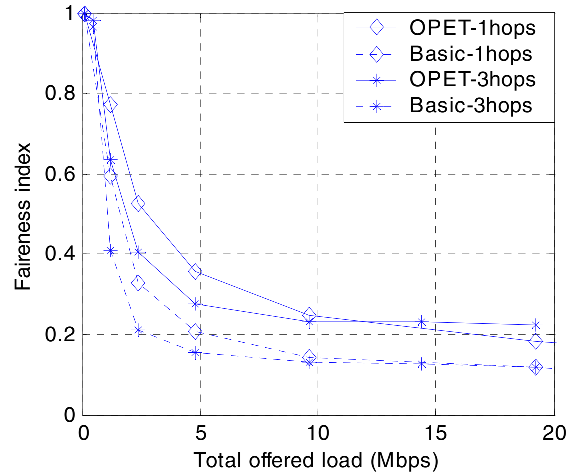

Fig. 7. Fairness index

offered load for multihop flows. That is to say, the Basic scheme is not appropriate for multihop ad hoc networks while OPET is a good choice for the multihop flows in the shared wireless channel environment and is scalable for larger networks where there are more multihop flows with longer paths.

D. Fairness Index

Fig. 7 shows that OPET improves the fairness index by up to 100% compared to the Basic scheme. In addition, we observe in the extended simulations that the multihop flows could always obtain certain throughput in OPET while they starve under heavy traffic load in the basic scheme. As shown in Fig. 4, the Basic scheme only take care of one or two hops flows while starving all other multihop flows. It's unfair to multihop flows. OPET gives much more bandwidth to multihop flows by Round Robin scheduling and source self-constraint scheme than the Basic scheme. The fairness index is still much less than one in our scheme because the traffic distribution is unbalanced in the random scenarios and the flows with shorter paths still have advantages over the flows with longer paths.

V. Conclusions

In this paper, we first present our finding that the poor performance of the IEEE 802.11 is attributed to the *intra-flow contentions* and *inter-flow contentions* in multihop ad hoc networks. In order to reduce these two kinds of contentions, we have proposed a frame work of flow control for the shared channel environment. Our scheme consist of the following four mechanisms. Assigning the highest priority of accessing the channel to the current receiver could achieve optimum packet scheduling for chain topology and greatly reduce the intra-flow contentions. Hop-by-hop backward-pressure scheduling deals with the congestion due to contentions from other traffic flows as well as from previous and succeeding hops in the same flow. Imposing constraint on the self-generated flows in the queue space provides opportunity to other flows to utilize the node. Flow-based Round Robin queue management improves the fairness among flows in addition to reducing the contention to the next hops. To evaluate the performance of the framework, we incorporate these mechanisms into IEEE 802.11 MAC protocol and the queue operations at each node.

Extensive simulations indicate that our scheme OPET could always achieve stable and high throughput and small end-to-end delay independent of traffic status, while IEEE 802.11 MAC protocol performs very poorly in terms of these two metrics for multihop flows. In addition, compared to the IEEE 802.11 MAC protocol, OPET has better fairness, much less and stabler control overhead almost independent of traffic load. Thus, OPET provides a very stable link layer and is scalable for large networks where there are many multihop flows with long paths without incurring explosion of control packets under heavy load.

References

[1] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee and R. Morris, "Capacity of Ad Hoc Wireless Network," in *Proc. of ACM MobiCom 2001*, July 2001.

[2] S.J. Lee and M. Gerla, "Dynamic Load-Aware Routing in Ad hoc Networks," in *Proc. of IEEE ICC 2001*, June 2001.

[3] S.B. Lee, and A.T. Campbell, "HMP: Hotspot Mitigation Protocol for Mobile Ad hoc Networks," in *11th IEEE/IFIP International Workshop on Quality of Service (IWQOS 2003)*, June, 2003.

[4] Z. Ye, D, Berger, P. Sinha, S. Krishnamurthy, M. Faloutsos, and S.K. Tripathi, "Alleviating MAC Layer Self-Contention in Ad-hoc Networks," in *Proc. of ACM MOBICOM 2003 Poster*, September, 2003.

[5] C. Li and E. Knightly, "Coordinated Multihop Scheduling: A Framework for End-to-End Services," *IEEE/ACM Transactions on Networking*, 10(6):776-789, December 2002.

[6] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints," in *Proc. of ACM MobiCom 2001*, July 2001.

[7] 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.

[8] K. Chen, Y. Xue, and K. Nahrstedt, "On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks," in *Proc. of IEEE ICC 2003*, May, 2003.