Distributed Flow Control and Medium Access in Multihop Ad Hoc Networks

Hongqiang Zhai, Student Member, IEEE, and Yuguang Fang, Senior Member, IEEE

Abstract—Recent studies have shown that the performance of wireless multihop ad hoc networks is very poor. In this paper, we first demonstrate that one important reason of the poor performance is the close coupling between medium contention and network congestion. Therefore, we present a framework of distributed flow control and medium access control to address both medium contention and network congestion. The proposed scheme utilizes the MAC layer control frames to efficiently conduct the network layer's flow control function and only allows the upstream nodes to forward enough packets to make it possible for the downstream nodes to fully utilize the shared channel but never introduce severe MAC collisions and network congestion. Extensive simulations illustrate that the proposed scheme well controls congestion and greatly alleviates medium collisions. It achieves up to 12 times the end-to-end throughput of IEEE 802.11, maintains a short delay and a low control overhead, and improves the fairness regardless of the hop count and the traffic load.

Index Terms—Wireless ad hoc networks, medium access control, flow control, intraflow contention, interflow contention.

1 INTRODUCTION

TITH the advancement of wireless technology, recent years have witnessed an ever-increasing popularity of wireless networks, including wireless local area networks, mobile ad hoc networks, wireless sensor networks, and wireless mesh networks. Wireless local area networks, or Wi-Fi hot spots, have been widely deployed in cities, college campus, airports, coffee bars, conference halls, hotels, and many other public places. However, communication in wireless local area networks is still limited to one-hop wireless communication between clients and access points, which restricts wireless access to a small range around access points. If mobile devices are allowed to forward packets for others, a mobile ad hoc network can be formed, and the range of wireless access to Wi-Fi hot spots can be significantly extended. This kind of wireless multihop communication is also common in wireless sensor networks that are often used in many applications such as environmental monitoring and health care. Recently, metropolitan Wi-Fi wireless mesh networks have become hot research topics because they can provide affordable outdoor wireless broadband Internet access to residents, businesses and visitors in a large area through a wireless multihop backhaul mesh structure. In this paper, we refer to all these networks that support multihop wireless communications as wireless multihop ad hoc networks and study how to improve their performance.

In wireless multihop ad hoc networks, nodes have to cooperate to forward each other's packets through the

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0260-0905. networks. Due to the contention for the shared channel, the throughput of each single node is limited not only by the channel capacity, but also by the transmissions in its neighborhood. Thus, each multihop flow encounters contentions not only from other flows which pass by the neighborhood, i.e., the *interflow contention*, but also from the transmissions of the flow itself because the transmission at each hop has to contend for the channel with the upstream and downstream nodes, i.e., the *intraflow contention*.

These two kinds of flow contentions could result in severe collisions and congestion, and significantly limit the performance of ad hoc networks. It has been shown in many papers that wireless multihop ad hoc networks perform poorly with TCP traffic as well as heavy UDP traffic ([4], [11], [20], [25], [32], [36], [41], [42], [43]). When traffic load is heavy or there are a large number of greedy users, severe MAC contention and network congestion will be observed and they will lead to dramatically decreased end-to-end throughput and increased end-to-end delay. The MAC protocol itself cannot solve congestion problems and often aggravates congestion due to contention in the shared channel. Fang and McDonald [9] studied how the throughput and delay can be affected by the path coupling, i.e., the MAC layer contention between nodes distributed along node disjoint paths, say, interflow contention. The results demonstrated the need for the control of cross-layer interactions and methodologies for cross-layer optimization.

To the best of our knowledge, there are no comprehensive studies on or good solutions to congestion control considering the MAC layer contentions and the packet scheduling of multihop traffic flows along their selected paths in the shared channel environment (more details are given in Section 5). In this paper, we present a framework of network layer flow control and MAC layer medium access to address the collisions and congestion problem due to *intraflow contention* and *interflow contention*. Based on the

H. Zhai is with the Department of Electrical and Computer Engineering, University of Florida, 481 Engineering Building, PO Box 116130, Gainesville, FL 32611-6130. E-mail: zhai@ecel.ufl.edu

Y. Fang is with the Department of Electrical and Computer Engineering, University of Florida, 435 Engineering Building, PO Box 116130, Gainesville, FL 32611-6130. E-mail: fang@ece.ufl.edu.

Manuscript received 2 Sept. 2005; revised 28 Feb. 2006; accepted 2 Mar. 2006; published online 15 Sept. 2006.



Fig. 1. Chain topology.

framework, a multihop packet scheduling algorithm is incorporated into the IEEE 802.11 Distributed Coordination Function (DCF) protocol [17].

The framework includes multiple mechanisms: fast relay, backward-pressure congestion control, receiver-initiated transmission scheduling, queue space limitation, and Round Robin scheduling. The fast relay assigns a high priority of channel access to the downstream nodes when they receive packets, which can reduce a lot of intraflow contentions. The backward-pressure congestion control gives transmission opportunity to the congested node while keeping its upstream nodes from transmissions. This can not only reduce a lot of contentions in the congested area, but also quickly eliminate the congestion. It is also a quick method to notify the source to slow the sending rate down by exploiting the RTS/CTS of the IEEE 802.11 MAC protocol. The receiver-initiated transmission scheduling scheme uses a three-way handshake to resume the blocked flow at the upstream nodes when the congestion is cleared. It is a timely and economical approach with even less control overhead than the normal four-way handshake transmission in the IEEE 802.11 protocol. The queue space limitation for each flow prevents the irresponsible application as well as the congested flows from occupying the whole queue space and leaves the resource for other responsible applications instead of the congested flows. The Round Robin scheduling is adopted in the queue management to further address the unfairness problem due to greedy sources.

Thus, altogether, all the above mechanisms provide a framework of distributed flow control and medium access control designed to reduce the MAC layer contentions and eliminate the congestion. Our contribution is to devise these mechanisms for the shared channel environment in multihop ad hoc networks and incorporate them into the IEEE 802.11 DCF protocol. Extensive simulation studies are carried out to validate their performance. It turns out that our scheme can maintain stable performance with high throughput independent of traffic status and improve the aggregated throughput by up to more than 12 times, especially for the multihop flows under heavy traffic load. At the same time, it also improves the fairness among flows in terms of end-to-end throughput and has much shorter delay and much lower control overhead compared to the IEEE 802.11 DCF 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 2 details the impact of MAC layer contentions on traffic flows and the resulting problems. Section 3 introduces our



Fig. 2. Cross topology.

scheme and the implementation based on the IEEE 802.11 DCF protocol. Section 4 evaluates the performance of our scheme through simulation. Related work is discussed in Section 5. Finally, we conclude the paper in Section 6.

2 IMPACT OF MAC LAYER CONTENTIONS ON TRAFFIC FLOWS

Different from wired networks where the links are independent of each other, wireless links share the same channel resource. Mobile nodes rely on the MAC layer to coordinate the channel access. The close interactions between the MAC layer and traffic flows offer great challenges to congestion control as well as medium access coordination. In this section, we characterize the interactions as intraflow and interflow contentions and discuss their impacts on the end-to-end performance of traffic flows as well as the MAC layer performance.

The intraflow contention discussed here is the MAC layer contentions for the shared channel among nodes of the same flow, which are in each other's interference range. Li et al. have observed that IEEE 802.11 fails to achieve the optimum chain scheduling [20]. 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. Unless otherwise indicated, in this paper, the maximum transmission radius is 250 m and the maximum interference radius is 550 m [20]. 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 with by five other nodes. This means that node 0, i.e., the source, could actually inject more packets into the chain than what the subsequent nodes can forward. These packets are eventually dropped at the subsequent nodes. We call this problem the *intraflow contention problem*.

In addition to the above contentions inside a multihop flow, the contentions between flows could also seriously decrease the end-to-end throughput. If two or more flows pass through the same region, the forwarding nodes of each flow encounter contentions not only from their own flow but also from other flows. Thus, the previous hops of these flows could actually inject more packets into the region than what the nodes in the region can forward. These packets are eventually dropped by the congested nodes. In Fig. 2, 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 chances to successfully transmit packets to its downstream nodes. The packets will accumulate at and be dropped by node 3, 9, 2, 8, and 1. We call this problem the *interflow contention problem*.

These two problems are very common and have unique features in multihop ad hoc networks. First, packet forwarding at each hop has to contend for channel resource with other traffic in the neighborhood. Second, interflow contention not only appears when several flows pass through the same forwarding node, but also exists when the flows' paths are close to each other such that the MAC layer only allows one transmission at a time to avoid collisions. Third, once the congestion occurs, MAC layer contentions become severe so that the MAC layer throughput decreases due to the increasing collision probability ([2], [35], [37], [38], [40]). This does not help to solve the congestion and instead results in more packets accumulating in the queue. These are the reasons why traditional congestion control schemes, such as TCP and heavy UDP traffic, have poor performance in ad hoc networks. The dropped packets due to both medium collisions and queue overflow waste a lot of scarce bandwidth and degrade end-to-end performance greatly. TCP cannot respond to congestion in time and often decreases the sending window a long time after the congestion occurs since it depends on the end-to-end feedback in acknowledgments and timeouts to conduct congestion control. And, TCP acknowledgments are delayed and even dropped not only due to the increased MAC layer collision probability, but also for the increased queue length (notice that each node only has one shared outgoing link and a corresponding queue for all outgoing packets).

Therefore, we argue that a good solution to the flow and congestion control problem in ad hoc networks must consider the MAC layer characteristics and respond quickly to the congestion. An intuitive solution to the above problems is to allow the downstream nodes and the congested ones to obtain the channel access 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.

3 OPET: OPTIMUM PACKET SCHEDULING FOR EACH TRAFFIC FLOW

3.1 Overview

The objective of our scheme is to approximate 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 solving the *intraflow contention* and *interflow contention* problems, our scheme OPET can significantly reduce the resource wasted by those dropped packets at forwarding nodes and, thus, could significantly improve the end-to-end performance.

OPET includes four major mechanisms. The first one is to assign a high priority of channel access to the current receiver. This could achieve optimum packet scheduling for chain topology and avoid severe intraflow contentions in each flow. The second one is the hop-by-hop backwardpressure scheduling. The forwarding nodes as well as the source are notified of the congestion and then are restrained from sending more packets to their next hops. This efficiently reduces the MAC layer contentions due to the *intraflow contention* and *interflow contention* on those congested nodes by keeping other nodes silent. The third one is not to allow the source node to occupy the whole outgoing queue, which could efficiently prevent the irresponsible applications from injecting more packets than the network could handle, and leave more queue space for other flows passing through this node. The last one is the Round Robin scheduling for the queue management, which further alleviates the unfairness problem between traversing flows and greedy source flows.

3.2 Address the Intraflow Contention Problem

In each multihop flow, the intermediate node on a path needs to contend for the shared channel with the previous nodes when forwarding the received packet to the next hop. One way to prevent the first few nodes on the path from injecting more packets than what the succeeding nodes can forward is to assign high channel access priority to each node when it just receives a packet. That is to say, the source node tries to hold the succeeding packets until the preceding packet is transmitted out of its interference range. This can achieve optimum scheduling for one-way traffic in the regular 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 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 cannot send packets to node 1 either when node 3 forwards a packet to node 4 because node 1 is in the interference range of node 3. When node 4 forwards a packet to node 5, node 0 has a chance to send a packet to node 1. Similar procedures are adopted by the succeeding nodes along the path. Nodes 0 and 4 can simultaneously send packets to their next hops, and a similar case happens to nodes which are four 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 achieved by the chain topology [20]. For a more random path, it is possible for more than four hops to interfere with the first hop transmission, so the maximum throughput is less than 1/4 of the channel bandwidth. OPET, however, allows the downstream nodes to access the channel with a higher priority. Succeeding packets are allowed to be forwarded when previous packets are forwarded out of the interference range of the current hop. Therefore, collisions between upstream nodes and downstream nodes are greatly reduced. In this way, maximum throughput can be approached by scheduling a transmission in each interference range. A more comprehensive study on the maximum throughput of a chain topology has been provided in [39].

To incorporate this procedure into the IEEE 802.11 DCF protocol, one solution is to assign higher channel access priorities to those packets which have traversed more hops. It requires that the MAC layer supports many different priority levels and needs the hop count information from





Fig. 4. The packet format of RTSM and CTSR.

Fig. 3. Optimum packet scheduling for chain topology.

the routing protocol. Currently, we opt for a simpler implementation which sets the initial value of the backoff window size of each receiver at 8 (i.e., whenever a node receives a packet, its backoff window is set to 8). When it finishes the transmission, the scheme resets its contention window size to the normal value 32 [17]. The example in Fig. 3 shows the optimum packet scheduling for the chain topology implemented by our scheme. Notice that nodes 1, 2, and 3 cannot become receivers at the same time due to the shared channel, and node 0 has low channel access priority and can rarely send the succeeding packets before the preceding packets have been forwarded to node 4.

This mechanism 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 seize the channel even with the highest priority. So, we introduce the backward-pressure scheduling to deal with the interflow contention.

3.3 Address the Interflow Contention Problem

Similar to the above mechanism, which keeps the source and upstream nodes from overloading the downstream nodes, the basic idea of backward-pressure scheduling is to keep nodes from transmitting to their already congested downstream nodes and, hence, yield the channel access to the congested nodes to clear congestion as well as to avoid severe medium contention. To avoid both severe congestion and medium collision, the mechanism detects an early sign of congestion for each flow, i.e., when the downstream node has enough packets of the flow to make full use of the channel bandwidth and, accordingly, starts corresponding procedures.

The mechanism includes a transmission blocking procedure and a transmission resuming procedure. It requires that each node monitors the number of packets of individual flow in the shared outgoing queue. Let n_i denote the number of packets of flow *i*. If n_i reaches a *backwardpressure threshold*, the transmission of flow *i* from its upstream node will be blocked and the upstream node is referred to as a *restricted node* of flow *i* in the following discussions. When the node successfully forwards some packets to its downstream node so that n_i is less than the backward-pressure threshold, it initiates the transmission resuming procedure to allow the restricted node to transmit packets of flow *i*.

Our scheme OPET sets the backward-pressure threshold as 1, which indicates the upper limit of the number of packets for each flow at each intermediate node. The smaller the value is, the less the medium contention, and 1 is large enough to be able to make full use of the channel bandwidth and is simple to implement. Notice that, in ad hoc networks, the wireless channel is shared by all the nodes in the same neighborhood. At any one time, at most one node can successfully access the channel and at most one packet can be successfully transmitted and received. Therefore, at all the nodes which are in the interference range of each other, if the total number of backlogged packets is equal to or larger than 1 at any time, the channel bandwidth will not be wasted due to idle period. For example, in a chain topology with more than three hops, 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. Considering other contending traffic in the neighborhood, this number should be smaller to minimize the medium contention as well as to make full use of the channel bandwidth. Since a fractional threshold is difficult to implement, we opt for the nearest integer 1 as the value of this threshold.

The transmission blocking 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 to the RTS when the intended receiver has reached the *backward-pressure threshold* for the corresponding flow. To uniquely identify each flow, the RTS for the multihop flows (RTSM) should include two more fields than the RTS, i.e., the source address and the flow ID. The 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. The NCTS packet has the same format as CTS except the different value in the frame type field. The format of the RTSM is shown in Fig. 4.

The transmission resuming procedure adopts the receiver-initiated transmission. It uses the three-way handshake CTS/DATA/ACK instead of the normal four-way handshake RTS/CTS/DATA/ACK because the downstream node already knows the restricted node has packets destined to it. The CTS to resume the transmission (CTSR) should include two more fields than the CTS, the source address and the flow ID, to uniquely specify the flow as shown in Fig. 4. The CTSR as well as the CTS has no information about its transmitter as that in the RTS. The two



Fig. 5. Message sequence for packet transmission.

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 paths if multipath routing is used. The procedure of transmitting a CTSR is similar to that of an RTS and allows multiple retransmissions before dropping it. Different message sequences at different situations are shown in Fig. 5.

The transmission resuming procedure also employs a complementary mechanism, i.e., resuming transmission by the upstream node itself. We notice that the mobility in ad hoc networks could result in link breakage followed by the transmission failure of the CTSR, and the CTSR may also collide several times and be dropped. The restricted node should start a timer, i.e., the *flow-delay timer*, and

TABLE 1 The Algorithms of the Backward-Pressure Scheme

=

=

=

Algorithm 1 Backward-Pressure Scheme
RecvRTSM(Packet p)
1: <i>number-of-packets</i> =CheckFlowTable(<i>p</i>)
2: if number-of-packets \geq backward-pressure-threshold
3: TransmitNCTS()
4: SetFlowTalbe(<i>p</i> , <i>block-flag</i> =1)
5: else
6: TransmitCTS()
7: end if
RecvNCTS(Packet p)
1: SetFlowTable(<i>p</i> , <i>restriction-flag</i> =1, <i>restriction-start-time</i> = now)
Algorithm 2 Resume-Transmission
ResumeTransmissionFromReceiver(Packet p)
Require: p is the packet that has been just successfully
transmitted to the downstream node by the MAC layer
1: <i>block-flag</i> =CheckFlowTable(<i>p</i>)
2: <i>number-of-packets</i> =CheckFlowTable(<i>p</i>)
3: if $block$ -flag = 1 and
4: number-of-packets < backward-pressure-threshold
5: TransmitCTSR()
6: end if
RecvCTSR(Packet p)
1: <i>data</i> =GetDataPktFromQueue(<i>p</i>)
2: if data! =NULL
3: TransmitDATA(data)
4: end
ResumeTransmissionFromTransmitter(Packet data)
Require: The retransmission timer for the restricted flow
expires at the transmitter.
data is one packet of the restricted flow in the queue
1: TransmitRTSM(Packet data)



Fig. 6. The packet scheduling when congestion occurs at node 4.

begin retransmission if its intended receiver has not sent the CTSR back in a long period, which we set to 1 second in our scheme. If the timeout value is too large, the blocked flow may be blocked for a very long time if CTSR is failed. If it is too small, the transmission of the blocked flow may be resumed earlier than the time when the downstream node eliminates the congestion. One second is a trade-off between them.

In the backward-pressure scheduling scheme, each node needs to maintain a table, i.e., flow-table, to record the information of the flows which currently have packets in the outgoing queue. A table item is created when a flow has the first packet in the outgoing queue and will be deleted when all the packets of the flow have been forwarded to the downstream 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 item includes the source-address, flow-ID, number-ofpackets in the queue, restriction-flag, restriction-start-time, upstream-node-address, and block-flag. The restriction-flag es whether the node is not allowed to forward the of this flow to the downstream node and the ion-start-time indicates when the restriction starts. ock-flag indicates whether the transmission of the m node is blocked. The algorithm for the backwarde scheme is shown in Table 1.

One simple example to illustrate how our scheme works s shown in Fig. 6 and Fig. 7. When congestion occurs at node 4 and node 4 could not forward packet 0 to its downstream node 5 as shown in Fig. 6, the flow along the chain will accumulate one packet at each node from node 1 o node 4 and then prevent the nodes 0, 1, 2, and 3 from contending for the channel to reduce the contention to the congested node 4. After eliminating the congestion at node 4, the transmission will be resumed by the congested node as shown in Fig. 7. Notice that in a random topology,



Fig. 7. The packet scheduling after eliminating the congestion at node 4.

the congestion can result from the interference or contention from any crossing and/or neighboring flows such that the considered node cannot capture the channel in time. OPET can efficiently force the upstream nodes of these flows to yield the channel access opportunity to the congested nodes, which then can quickly forward the backlogged packets and, hence, eliminate the congestion.

It is important to note that the control overhead of the backward-pressure scheduling is very small. The information of backward-pressure is carried by the original message sequences RTS/CTS in IEEE 802.11 and the blocked flow is resumed by a three-way handshake procedure with less overhead than the original four-way handshake. Moreover, our scheme only maintains several small entries for each active flow which has at least one packet at the considered node. 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 of much smaller order than in the wired networks; thus, the scalability problem should not be a major concern in our scheme.

3.4 Address the Unfairness between Traversing Flows and Greedy/Congested Source Flows

When 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 chances 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 four hops away. Considering that the channel is shared by other traffic flows in a random topology, the achievable throughput is equal to or less than that when there is only a single flow. Therefore, in a general topology, c + h/4 is large enough to saturate a path if the source is greedy and vacates more queue space for traversing flows than that when there is no such source self-constraint scheme. This threshold is applied to UDP flows and is optional for TCP flows. Notice that TCP can only inject packets up to the receiver's advertised window size into the queue. Furthermore, Chen et al. [6] have discovered in their simulation that TCP's congestion window size should be less than kN when considering transmission interference at the MAC layer, where 1/8 < k < 1/4 and N is the number of round-trip hops. So, c + h/4 should work for TCP flows if we set the congestion window limit less than the upper bound kN.

Flow-based Round Robin scheduling is adopted in our scheme for the queue management. It aims to further address the unfairness problem resulting from greedy sources, especially those of one-hop flows. When a greedy source node is also a forwarding node of other flows, it may continuously transmit multiple packets generated from its own applications if using FIFO scheduling because it could have as many packets as what these applications can inject into the queue. The upstream nodes of traversing flows need to contend for the channel with this node to squeeze the packets into the limited queue space, most of which may have been occupied by the packets generated at this node. The Round Robin scheme can efficiently allow the traversing traffic to pass through and, hence, avoid the starvation problem. Notice that if the flow that the head-of-queue packets belong to are blocked by its downstream node, the node should attempt to transmit packets of those flows which are not blocked and may have better path characteristics to avoid a head-of-queue blocking problem. If variable sizes of packets are used in the network, the Deficit Round Robin (DRR) [27] or Surplus Round Robin (SRR) [1] could be used. Different fair queueing schemes within the proposed framework will be evaluated in our future work.

Another option to solve the unfairness problem due to a greedy source is to allocate a separate queue space for the packets originated at the considered node. Only when the amount of data of each source flow in the shared outgoing queue is smaller than a certain threshold, i.e., backwardpressure threshold in the proposed scheme, can the packets in the separate queue be passed to the shared outgoing queue. Apparently, this method requires additional queue space.

4 PERFORMANCE EVALUATION

We now evaluate the performance of our scheme OPET and compare it with the IEEE 802.11 MAC protocol. The simulation tool is one of the widely used network simulation tools—ns-2. We use precomputed shortest path and there is no routing overhead if otherwise indicated. The channel bandwidth is 2 Mbps and the payload size of each DATA packet is 1,000 bytes. The transmission range is 250 meters and the sensing range is 550 meters.

In our simulations, the following several important performance metrics are evaluated:

- Aggregate end-to-end throughput—The amount of data delivered to the destinations per second.
- Average end-to-end delay—The average end-to-end delay of all packets which reach the destinations.
- *Data transmission efficiency*—The ratio of the sum of traveled hop counts of those DATA packets that successfully arrive at destinations to the total transmission times of all DATA packets at all nodes. This metric reflects the resource wasted by the collided DATA packets and those DATA packets which are discarded due to overflow of queue at the intermediate nodes of the path.
- Normalized control overhead—The ratio of the number of all kinds of control packets including RTS(M), (N)CTS, CTSR, and ACK to the sum of hop counts passed by those successfully delivered DATA packets.
- *Fairness index*—The commonly used fairness index for all flows x_i(1 ≤ i ≤ n), i.e.,

$$f = \left(\sum_{i=1}^n x_i\right)^2 / (n \cdot \sum_{i=1}^n x_i^2),$$



Fig. 8. End-to-end throughput in the 9-node chain topology (Fig. 3) and cross topology (Fig. 2).

where x_i denotes the end-to-end throughput of the *i*th flow.

In the simulation study, our scheme will be referred to as the Optimum Packet scheduling for Each Traffic flow (OPET), and the IEEE 802.11 protocol with RTS/CTS and without the packet scheduling algorithm will be referred to as the Basic scheme.

4.1 Simple Scenarios

We first investigate how well our scheme works in the simple scenarios, i.e., the 9-node chain topology with one-way traffic and two-way traffic shown in Fig. 3 and the cross traffic scenario shown in Fig. 2.

Fig. 8 shows that our scheme improves the throughput by 55, 120, and 33 percent compared to the IEEE 802.11 in these three scenarios under heavy traffic load, respectively. We observe in Fig. 9 that our scheme maintains a small and stable end-to-end delay at all traffic status while the end-to-end delay increases dramatically with increasing traffic load in the IEEE 802.11 protocol. The reason is straightforward because our scheme reduces a lot of MAC layer contentions, i.e., the intraflow contention and the interflow contention, and removes the excessive queueing delay at the forwarding nodes.



Fig. 9. End-to-end delay in the 9-node chain topology (Fig. 3) and cross topology (Fig. 2).



Fig. 10. Aggregate end-to-end throughput in the random topology.

4.2 Random Topology

In these simulations, 60 nodes are randomly placed in a $1,000m \times 1,000m$ area. The source of each flow randomly selects one node as the destination, which is at least *m* hops away from the source. In our studies, we choose m = 1 or 3. There are a total of 30 flows with the same CBR/UDP traffic in the network. All results are averaged over 30 random simulations with 300 simulated seconds each.

We observe from Fig. 10 that, when the minimum number of hops for each flow increases, the aggregated endto-end throughput of both protocols decreases. This is reasonable because packets of multihop flows with longer paths 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 percent under heavy traffic. This is because OPET reduces a lot of channel contentions due to the intraflow contention and interflow contention and there are many fewer accumulated packets which are eventually dropped by the forwarding nodes. The reason that the Basic scheme can maintain certain throughput under heavy traffic is that the IEEE 802.11 MAC protocol gives preference to those one-hop or two-hop flows which have no or many fewer 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 throughputs as shown in Fig. 11, which shows one random example of throughput distribution among flows under heavy traffic and also shows the improved fairness in OPET.

If source-destination pairs of all flows are at least three hops away, OPET can still maintain high end-to-end throughput at heavy traffic load while the Basic scheme almost drops to zero end-to-end throughput. In the Basic scheme, the *intraflow contention* can allow the sources of multihop flows to inject more packets into the network than the network can forward. The *interflow contention* makes the situation worse. It is not surprising in the Basic scheme that, the longer path the flow has, the lower the end-to-end throughput it can achieve. By reducing the *intraflow* and *interflow contention*, our scheme always maintains the high end-to-end throughput for all flows at any traffic load and

Fig. 11. One random example to illustrate throughput distribution among flows in the random topology.

15 Flow ID (total 30 flows)

10

5

 \Diamond OPET

 \odot Basic

20

of hops

25

30

the improvement is more than 12 times under heavy traffic comparing to IEEE 802.11 protocol.

Fig. 12 shows that OPET has much smaller end-to-end delay than the Basic scheme. Also, for multihop flows, our scheme provides stable end-to-end delay in spite of high traffic load, while, in the Basic scheme, the end-to-end delay rapidly increases with the offered load. This is because OPET reduces a lot of accumulated packets in the outgoing queue at each node and, thus, greatly reduces the queueing delay. In addition, OPET reduces the contentions from the intraflow and interflow contention, which can also decrease the delay at the MAC layer to access the channel. It also verifies that, in OPET, there is no severe congestion which can result in excessive queueing delay at the forwarded nodes.

Fig. 13 shows that OPET achieves better transmission efficiency of DATA packets as high as about 90 percent, while the Basic scheme has much lower value, i.e., even less than 5 percent for multihop flows. This metric indicates that the Basic scheme discards a lot of packets that the sources send out, which have not reached the intended destinations. This implies that these packets waste a lot of wireless bandwidth and consume significant power. OPET greatly reduces this kind of waste and utilizes the resource to achieve higher end-to-end throughput.

In OPET, the transmission efficiency of DATA packets is still less than 1. This is because OPET is still running on the



contention-based MAC protocol, i.e., the IEEE 802.11 MAC protocol. There exists a hidden terminal problem which results in DATA packet collisions.

8

10

Fig. 14 shows that OPET could maintain small and stable normalized control overhead. This verifies that OPET can reduce a lot of collisions at the MAC layer and, hence, save a lot of unsuccessful RTS/CTS negotiations and DATA transmissions. The Basic scheme has a much higher control overhead, which rapidly increases with the offered load for multihop flows. This implies that 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.

Fig. 15 shows that OPET improves the fairness index by up to 100 percent compared to the Basic scheme. As in the random example shown in Fig. 11, the Basic scheme only takes care of one or two-hop flows while starving all other multihop flows. It is not fair to multihop flows with large hop counts. OPET gives much more bandwidth to multihop flows with large hop counts than the Basic scheme. The fairness index is still much less than 1 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.



Fig. 12. Average end-to-end delay in the random topology.







End-to-end throughput (# of pkts)

10⁵

10⁴

10

10²

10¹

10⁰

0

Fig. 14. Normalized control overhead in the random topology.



Fig. 15. Fairness index in the random topology.

4.3 Random Topology with Mobility

In the simulations, 60 nodes are randomly placed in a $1,000m \times 1,000m$ area. All nodes are randomly moving in the rectangular grid with a randomly chosen speed (uniformly distributed between 0 - 10m/s). There is a total of 30 flows with the same CBR/UDP traffic. The source of each flow randomly selects one node as the destination. The routing scheme is AODV [25]. All results are averaged over 30 random simulations with 300 seconds simulated time each.

The purpose of considering the mobility is only to illustrate that our scheme can work well in the mobile scenarios with an on-demand routing scheme. In fact, we find in the extensive simulations that mobility does not change the results much. Therefore, we only show the aggregated end-to-end throughput in Fig. 16, which shows that OPET has about 50 percent higher throughput than the Basic scheme. All other performance metrics are also similar with the scenario where the source and destination are randomly selected without hop count limitation in the static topology.

We also notice that mobility decreases the throughput. This is because the route may be unavailable during certain periods due to mobility although each source has a route to its destination at the start time. In addition, the extensive simulations also indicate that mobility increases the end-toend delay because the route searching and repairing time comes into play.



Fig. 17. Collisions of TCP traffic in chain topology.

4.4 Simulation Results for TCP Traffic

We first investigate how well our scheme performs in the 9-node chain topology with a different number of TCP flows. Fig. 17 shows that our scheme OPET can reduce the packet collision by about 40 percent for both RTS and ACK frames and the number of dropped TCP packets is also reduced by about 80 percent. This verifies that the hop-byhop congestion control can effectively reduce a lot of medium contention and collision. Fig. 18 demonstrates that OPET can improve the aggregate throughput of TCP flows by about 5 percent and the fairness is even better than the Basic scheme. Fig. 19 illustrates that the TCP source node can detect the congestion status by simply observing its queue length if OPET is used and may accordingly change the sending rate to obtain better performance.

Now, we examine the TCP performance in a larger network with grid topology, where interflow contention is a common phenomenon. One hundred nodes are arranged in a 10×10 grid with nodes separated by 200 meters. Sixteen TCP flows with eight horizontal ones from the second to the ninth rows and eight vertical ones from the second to the ninth columns run for 300 seconds in the



Fig. 16. Simulation results for the random topology with mobility.



Fig. 18. Throughput and fairness of TCP traffic in chain topology.



Fig. 19. Queue length for TCP traffic in chain topology.

simulation. Compared with the Basic scheme, OPET improves the end-to-end throughput from 547Kbps to 603Kbps by 10 percent and reduces the collided RTS from 1,015pkt/s to 802pkt/s by 21 percent.

These results show that OPET can also improve TCP performance, although TCP flows tend to generate burstier traffic. This is because OPET can reduce lots of packet droppings due to MAC collisions as well as queue overflow and, hence, the TCP source conducts many fewer retransmissions and also experiences much less oscillation in the sending window size. The optimization of the interaction between TCP and OPET should provide better support for TCP traffic and will be studied in future work.

4.5 Notes on the Relative Benefits of the Four Techniques

The first mechanism, which assigns higher channel access priority to the downstream nodes when they receive packets, works very well for chain topology with one-way traffic. It gives an efficient solution to the intraflow contention problem. However, when interflow contention comes into play in a more general topology, the MAC layer contention is still severe if only the first mechanism is used. In these scenarios, the combination with backward-pressure scheduling greatly alleviates both intraflow and interflow contentions and contributes to the performance gain in endto-end throughput and delay. Despite the greatly improved aggregate performance, fairness is not improved much and starvation is still a severe problem for multihop flows, especially when some source nodes are also working as forwarding nodes. Therefore, we introduce the source selfconstraint scheme and the Round Robin scheme into the framework. The former allocates certain queue space for traversing packets to alleviate the possibility of being dropped due to queue overflow. The latter allows that the traversing flows obtain relatively fair throughputs compared with the flows generated at this node that may often occupy most of the queue space and, hence, may have more chances to be transmitted. More extensive simulation results to illustrate these relative benefits are omitted here due to the page limit.

5 RELATED WORKS AND DISCUSSION

Recently, many schemes have been presented to alleviate the MAC layer collisions. The authors in [12], [28] proposed receiver-initiated transmission schemes, which work well when the intended receiver knows exactly the traffic load information. Wang and Garcia-Luna-Aceves [30] proposed a hybrid channel access scheme, which combines both sender-initiated and receiver-initiated collision avoidance handshakes. Their scheme could alleviate the fairness problem in some cases without sacrificing much throughput and simplicity, but cannot trigger the desired receiverinitiated collision avoidance handshake in some scenarios due to the lack of flow contention information. Berger et al. [3], [33] presented two MAC layer enhancements, i.e., quick-exchange and fast-forward, to address self-contention in ad hoc networks. The former allows the receiver to return a DATA packet to the sender and the latter includes an implicit RTS to the next hop. They could save some transmission negotiation procedures, i.e., the RTS/CTS exchanges.

In the last few years, several studies ([18], [22]) have been reported for the distributed packet scheduling which considers the MAC layer collisions in the multihop ad hoc networks. The proposed schemes use different backoff window size to assign different priorities for packets to access the channel. Luo et al. [22] constructed the flow contention graph to achieve better fairness among one-hop flows between different node pairs. Kanodia et al. [18] applied EDF (Early Deadline First) criteria to obtain smaller end-to-end delay than the original IEEE 802.11, although congestion is not fully addressed and the delay still increases dramatically with the increasing offered load.

Traditional end-to-end congestion control, TCP, has been shown to be inefficient in ad hoc networks in many recent papers ([6], [7], [10], [11], [13], [14], [32], and references therein). Most of the current work to improve TCP performance, such as [5], [6], [8], [11], [16], [21], [24], [29], [31], focuses on the end-to-end congestion control mechanism of TCP with or without network layer feedback. The proposed schemes did not fully address the impact of MAC layer performance and still suffered from the severe MAC laver contentions. Gupta et al. [15] used a back-pressure concept to provide a fair channel access to TCP flows under heavy UDP traffic with an implementation of a virtual, globally accessible array that dynamically records the queue lengths for each flow at each node in the network. Monks et al. [24] conducted simulations to illustrate the limitations of TCP-ELFN [16] and discussed the pros and cons of end-to-end control and hopby-hop control. They argue that the advantages of hop-byhop control may outweigh its drawbacks.

Hop-by-hop congestion control has been studied in wired networks especially in ATM networks ([19], [23]), but these schemes cannot be directly applied in ad hoc networks due to the completely different MAC and physical layers. To our best knowledge, in recent studies, only [34] comprehensively discussed hop-by-hop congestion control for ad hoc networks. The authors formulated an optimization problem and studied the end-to-end throughput under both hop-byhop congestion control and the end-to-end congestion control. Their model 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. Compared to congestion control schemes for wired networks, we can regard CTSR packets in OPET as a kind of credit like those in the credit-based flow control scheme [19] and NCTS packets can be regarded as a kind of hop-by-hop source Quench although OPET does not require the cooperation of transport and application layers [26].

To the best of our knowledge, there are no comprehensive studies to effectively address the *intraflow contention* and *interflow contention* problems in multihop mobile ad hoc networks, which result in serious problems, such as "explosion" of control packets, severe collisions of data packets, poor throughput and fairness, excessively long end-to-end delay, congestion, and poor scalability. Thus, all the prior works only contribute to the improvement of one or two of these performance metrics while sacrificing other metrics more or less. By tackling these two key problems with a novel cross-layer design, our scheme could improve all these metrics for both UDP and TCP traffic, which is a significant departure from most recent works.

6 CONCLUSIONS

In this paper, we first discuss the causes of poor performance of the IEEE 802.11 MAC protocol, i.e., the intraflow contention and interflow contention, in multihop ad hoc networks. In order to reduce these two kinds of contentions, we have proposed a framework of distributed flow control and media access based on which one multihop packet scheduling algorithm, i.e., OPET, is proposed for IEEE 802.11 multihop ad hoc networks. Extensive simulations verify that our scheme OPET greatly reduces excessive collisions at the MAC layer, quickly eliminates the congestion, and has a much better multihop packet scheduling than the IEEE 802.11 MAC protocol. Thus, it could achieve stable and high throughput and short end-to-end delay independent of traffic load, while the 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, many fewer dropped DATA packets, and stabler control overhead. 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.

ACKNOWLEDGMENTS

This work was supported in part by the US Office of Naval Research under Young Investigator Award N000140210464 and by the US National Science Foundation under Faculty Early Career Development Award ANI-0093241 and under grant DBI-0529012.

REFERENCES

- [1] H. Adiseshu, G. Parulkar, and G. Varghese, "A Reliable and Scalable Striping Protocol," *Proc. ACM Sigcomm Conf.*, Aug. 1996.
- [2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE J. Selected Areas in Comm., vol. 18, pp. 535-537, Mar. 2000.
- [3] D. Berger, Z. Ye, P. Sinha, S.V. Krishnamurthy, M. Faloutsos, and S.K. Tripathi, "TCP Friendly Medium Access Control for Ad-Hoc Wireless Networks: Alleviating Self Contention," *Proc. IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems*, Oct. 2004.

- [4] 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 Conf.*, Oct. 1998.
- [5] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback-Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks," *IEEE Personal Comm.*, vol. 8, pp. 34-39, Feb. 2001.
- [6] K. Chen, Y. Xue, and K. Nahrstedt, "On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks," Proc. IEEE Int'l Conf. Comm., May 2003.
- [7] X. Chen, H. Zhai, J. Wang, and Y. Fang, "TCP Performance over Mobile Ad Hoc Networks," *Canadian J. Electrical and Computer Eng.*, vol. 29, pp. 129-134, Jan./Apr. 2004.
- [8] T.D. Dyer and R.V. Boppana, "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks," Proc. ACM Mobihoc Conf., Oct. 2001.
- [9] 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. IEEE Int'l Performance, Computing, and Comm. Conf., Apr. 2002.
- [10] Z. Fu, X. Meng, and S. Lu, "How Bad TCP Can Perform in Mobile Ad-Hoc Networks," Proc. IEEE Symp. Computers and Comm., July 2002.
- [11] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," Proc. IEEE INFOCOM Conf., Mar. 2003.
- [12] J.J. Garcia-Luna-Aceves and A. Tzamaloukas, "Reversing the Collision-Avoidance Handshake in Wireless Networks," Proc. ACM/IEEE MobiCom Conf., Aug. 1999.
- [13] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over Wireless Multihop Protocols: Simulation and Experiments," *Proc. IEEE Int'l Conf. Comm.*, June 1999.
- [14] M. Gerla, K. Tang, and R. Bagrodia, "TCP Performance in Wireless Multihop Networks," Proc. IEEE Workshop Mobile Computing Systems and Applications, Feb. 1999.
- [15] V. Gupta, S.V. Krishnamurthy, and M. Faloutsos, "Improving the Performance of TCP in the Presence of Interacting UDP Flows in Ad Hoc Networks," *IFIP Networking*, May 2004.
- [16] G. Holland and N.H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," Proc. ACM MobiCom Conf., Aug. 1999.
- [17] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11: 1999(E), Aug. 1999.
- [18] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed MultiHop Scheduling and Medium Access with Delay and Throughput Constraints," *Proc. ACM MobiCom Conf.*, July 2001.
- [19] H.T. Kung, T. Blackwell, and A. Chapman, "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing," *Proc. ACM Sigcomm Conf.*, Sept. 1994.
- [20] J. Li, C. Blake, D. Couto, H. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Network," Proc. ACM MobiCom Conf., July 2001.
- [21] J. Liu and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks," IEEE J. Selected Areas in Comm., vol. 19, pp. 1300-1315, July 2001.
- [22] H. Luo and S. Lu, "A New Model for Packet Scheduling in Multihop Wireless Networks," Proc. ACM MobiCom Conf., Aug. 2000.
- [23] P.P. Mishra and H. Kanakia, "A Hop by Hop Rate-Based Congestion Control Scheme," Proc. ACM Sigcomm Conf., Aug. 1992.
- [24] J.P. Monks, P. Sinha, and V. Bharghavan, "Limitations of TCP-ELFN for Ad Hoc Networks," Proc. Int'l Workshop Mobile Multimedia Comm., Oct. 2000.
- [25] 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 Comm.*, vol. 8, pp. 16-28, Feb. 2001.
- [26] J. Postel, "Internet Control Message Protocol," IETF RFC 792, Sept. 1981
- [27] M. Shreedhar and G. Varghese, "Efficient Fair Queuing Using Deficit Round-Robin," *IEEE/ACM Trans. Networking*, vol. 4, pp. 375-385, June 1996.
- [28] F. Talucci and M. Gerla, "MACA-BI (MACA by Invitation): A Receiver Oriented Access Protocol for Wireless Multihop Networks," Proc. IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm., Sept. 1997.

- [29] F. Wang and Y. Zhang, "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-of-Order Detection and Response," *Proc. ACM Mobilinoc Conf.*, June 2002.
- [30] Y. Wang and J.J. Garcia-Luna-Aceves, "A Hybrid Collision Avoidance Scheme for Ad Hoc Networks," Wireless Networks, vol. 10, pp. 439-436, July 2004.
- [31] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED," Proc. ACM MobiCom Conf., Sept. 2003.
- [32] S. Xu and T. Safadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?" *IEEE Comm. Magazine*, vol. 39, pp. 130-137, June 2001.
- [33] Z. Ye, D. Berger, P. Sinha, S. Krishnamurthy, M. Faloutsos, and S.K. Tripathi, "Alleviating MAC Layer Self-Contention in Ad-Hoc Networks," Proc. ACM MobiCom Poster, Sept. 2003.
- [34] Y. Yi and S. Shakkottai, "Hop-by-Hop Congestion Control over a Wireless MultiHop Network," Proc. IEEE INFOCOM Conf., Mar. 2004.
- [35] H. Zhai, X. Chen, and Y. Fang, "A Call Admission and Rate Control Scheme for Multimedia Support over IEEE 802.11 Wireless LANs," Wireless Networks, vol. 12, no. 4, pp. 451-463, 2006.
- [36] H. Zhai, X. Chen, and Y. Fang, "Alleviating Intraflow and Interflow Contentions for Reliable Service in Mobile Ad Hoc Networks," *Proc. IEEE Milcom Conf.*, Nov. 2004.
- [37] H. Zhai, X. Chen, and Y. Fang, "How Well Can the IEEE 802.11 Wireless LAN Support Quality of Service?" IEEE Trans. Wireless Comm., vol. 4, no. 6, pp. 3084-3094, Nov. 2005.
- [38] H. Zhai and Y. Fang, "Performance of Wireless LANs Based on IEEE 802.11 MAC Protocols," Proc. IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm., Sept. 2003.
- [39] H. Zhai and Y. Fang, "Physical Carrier Sensing and Spatial Reuse in Multirate and Multihop Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM Conf.*, Apr. 2006.
 [40] H. Zhai, Y. Kwon, and Y. Fang, "Performance Analysis of IEEE
- [40] H. Zhai, Y. Kwon, and Y. Fang, "Performance Analysis of IEEE 802.11 MAC Protocols in Wireless LANs," Wireless Comm. and Mobile Computing, vol. 4, pp. 917-931, Dec. 2004.
- [41] H. Zhai, J. Wang, X. Chen, and Y. Fang, "Medium Access Control in Mobile Ad Hoc Networks: Challenges and Solutions," Wireless Comm. and Mobile Computing, vol. 6, pp. 151-170, Mar. 2006.
- [42] H. Zhai, J. Wang, and Y. Fang, "Distributed Packet Scheduling for Multihop Flows in Ad Hoc Networks," Proc. IEEE Wireless Comm. and Networking Conf., Mar. 2004.
- [43] H. Zhai, J. Wang, and Y. Fang, "DUCHA: A Dual-Channel MAC Protocol for Mobile Ad Hoc Networks," *IEEE Trans. Wireless Comm.*, to appear.



Hongqiang Zhai received the BE and ME degrees in electrical engineering from Tsinghua University, Beijing, China, in July 1999 and January 2002, respectively. He worked as a research intern in Bell Labs Research China from June 2001 to December 2001 and in Microsoft Research Asia from January 2002 to July 2002. Currently, he is pursuing the PhD degree in the Department of Electrical and Computer Engineering, University of Florida.

He is a student member of the IEEE and the ACM.



Yuguang Fang received the PhD degree in systems and control engineering from Case Western Reserve University in January 1994 and the PhD degree in electrical engineering from Boston University in May 1997. From June 1997 to July 1998, he was a visiting assistant professor in the Department of Electrical Engineering at the University of Texas at Dallas. From July 1998 to May 2000, he was an assistant professor in the Department of Elec-

trical and Computer Engineering at the New Jersey Institute of Technology. In May 2000, he joined the Department of Electrical and Computer Engineering at the University of Florida, where he got an early promotion with tenure in August 2003 and became a full professor in August 2005. He has published more than 180 papers in refereed professional journals and conferences. He received the US National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He is currently serving as an editor for many journals, including the IEEE Transactions on Communications, the IEEE Transactions on Wireless Communications, the IEEE Transactions on Mobile Computing, and the ACM Wireless Networks. He is also actively participating in many professional conference organizations, including serving as technical program vice chair for the IEEE International Conference on Computer Communication (INFOCOM '05), cochair for the Global Internet and Next Generation Networks Symposium at IEEE Globecom '04, and program vice chair for the 2000 IEEE Wireless Communications and Networking Conference (WCNC '00). He is a senior member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.