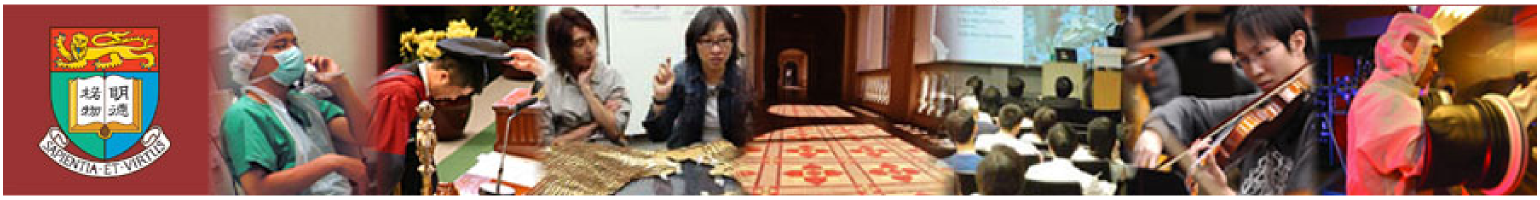


Distributed Scheduling with End-to-End Compensation in Multihop Ad Hoc Networks

Yijiang Sun, Victor O.K. Li and Ka-Cheong Leung
Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong SAR, China
Email: {yjsun, vli, kcleung}@eee.hku.hk

***Abstract*—In this paper, we investigate the problem of providing QoS to end-to-end flows in multihop ad hoc networks with channel errors through packet scheduling. Each flow is associated with some QoS requirement, which is requested and granted in the form of a desired service rate. The achieved rate is estimated at the destination and fed back to the source periodically. Both the desired rate and achieved rate of a multihop flow are piggybacked on the packets of the flow and propagated from the source node to all its downstream relaying nodes. With such information, a compensation-capable scheduling algorithm originally designed for infrastructured wireless networks can be adapted to each ad hoc node for compensating a lagging flow, i.e., a flow with the achieved rate smaller than the desired rate. We propose the feedback and propagation mechanism as an end-to-end compensation framework, which is the key contribution of this work. We use BGFS-EBA, a scheduling algorithm for infrastructured wireless networks, as an example to demonstrate how such an algorithm is adapted to ad hoc networks within the proposed framework. Our simulation results show that the proposed mechanism maintains outcome fairness and compensate flows that suffer sporadic bursty channel errors effectively.**

I. Introduction

A wireless ad hoc network consists of a number of wireless nodes communicating with each other on wireless links without infrastructure support. A multihop ad hoc network is a network in which a traffic flow needs to be relayed by one or more intermediate nodes before they reach the destination. Providing quality of service (QoS) guarantees for multihop flows is an important issue in wireless ad hoc networks. Some recent work [1], [2], [3], [4] have proposed fair packet scheduling in mobile ad hoc networks.

[1] aims to construct a transmission schedule at each node for one-hop flows so that the bandwidth allocation to these sessions is maxmin fair. It assumes that any two single-hop flows not sharing a node can transmit packets simultaneously. Tokens are used to model and achieve fairness. However, the proposed algorithm requires global information of the network, but no practical distributed implementation scheme is provided.

In [2], channel errors are taken into account. By employing a Time Division Multiple Access (TDMA)-based system, the proposed Timestamp Based Compensation Protocol (TBCP) is designed to adapt the start-time fair queueing (SFQ) scheme [5] into the ad hoc environment. Service tags are computed locally and exchanged among two-hop neighbors. Time slot allocation is computed based on these service tags. Collisions between neighboring nodes are inevitable since they do not have exactly the same information. However, TBCP does not handle collisions due to such conflicts.

In [3], a self-coordinated scheduling scheme is presented within the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) Medium Access Control (MAC) paradigm. The fairness model attempts to maximize spatial channel reuse while guaranteeing a basic fair share of throughput for all backlogged flows on the basis of their flow weights. Similar to [2], an SFQ-like virtual time is maintained at each node to compute timestamps for each packet. Timestamps are exchanged among neighboring nodes and used in the computation of the MAC backoff time. The proposed algorithm achieves notable improvements in both fairness and system throughput compared with the IEEE 802.11 Distributed Coordination Function (DCF). However, it focuses only on one-hop flows and no compensation mechanism is considered.

In [4], the author seeks to provide fair bandwidth allocation to multihop flows based on the principle of maximizing the network throughput without violating each flow's basic fair share. A two-phase algorithm is proposed. In the first phase, each flow's bandwidth share is calculated according to its assigned weight. In the second phase, the transmission schedule is constructed by an SFQ-like scheduler based on the calculated bandwidth shares. The algorithm addresses the fairness issue of multihop flows rather than single-hop flows. However, the fairness model does not take error-prone channels into account. No compensation mechanism is provided for end-to-end flows. In addition, the assigned bandwidth share of a flow may not be in proportion to its flow weight.

The existing algorithms have three major limitations:

1) Channel errors and collisions, which are very common in wireless networks, are typically ignored. No compensation mechanism is provided to make up the lost service.
2) The QoS requirement is given in the form of a flow weight, which by itself cannot guarantee the achieved rate.
3) The only error compensation mechanism is provided in [2]. It only applies to single-hop flows. No end-to-end compensation mechanism is provided.

To address these issues, we propose a novel framework known as End-to-end Compensation Scheduling for Ad hoc

This work is supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant No. HKU 7148/06E.

978-1-4244-2644-7/08/$25.00 ©2008 IEEE

networks (ECSA). In ECSA, the QoS requirement is given as the desired service rate. Sporadic bursty errors on the wireless channels are taken into account. The service loss due to these errors will be compensated for in an end-to-end manner.

The rest of the paper is organized as follows. Section II describes the system model and assumptions used in this paper. Section III introduces the end-to-end feedback mechanism. Section IV describes the scheduling algorithm under study, BGSA, which is adapted from an infrastructure-based wireless scheduling algorithm. Simulation results and discussion are given in Section V. We conclude in Section VI.

II. System Model and Assumptions

The following terminologies and assumptions are made:

1) We consider a multihop wireless ad hoc network. The terms "source" and "destination" refer to the sender and receiver, respectively, of a multihop flow. The terms "transmitter" and "receiver" refer to the sender and receiver, respectively, of a wireless link.
2) A node cannot transmit and receive packets simultaneously. A collision happens when a receiver is in the transmission ranges of multiple transmitters.
3) Assume that all flows are properly routed. We do not consider routing issues in this paper.
4) The MAC protocol gets immediate acknowledgment from the receiver for each successful transmission.
5) A wireless link is error-prone and channel errors are not negligible. Besides, the transmission schedule is computed at each node locally, based on some incomplete network information. Thus, collisions are inevitable.
6) Unsuccessful transmissions are due to either channel errors or packet collisions. The transmitter has no means to know the cause of an unsuccessful transmission.
7) The desired service rate of an end-to-end flow is assigned by the source node and propagated to all the downstream nodes.
8) A transport layer protocol is responsible for compensating lagging flows by sending new packets and retransmitting any lost packets.

III. An End-to-End Feedback Mechanism

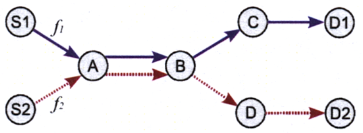


Fig. 1. Two End-to-End Flows.

Fig. 1 shows two end-to-end flows, f_1 and f_2. f_1 is from the source node S1 to the destination node D1, via nodes A, B and C. f_2 is from the source node S2 to the destination node D2, via nodes A, B and D. Let r_i be the desired rate of flow i (refer to Section III-D for details about the propagation of r_i). We use this simple example to illustrate the local (one-hop) compensation and the end-to-end (multihop) compensation.

A. Local (One-hop) Compensation

Take node B as an example. Node B has to schedule two flows f_1 and f_2 on links B→C and B→D, respectively. To compensate the flow that is receiving service at a rate below its desired rate due to channel errors or collisions, some existing scheduling algorithms designed for the infrastructured wireless networks can be used.

Note that, in a multihop wireless ad hoc network, unsuccessful transmissions are caused not only by channel errors, but also by collisions with other flows. However, from the transmitter's point of view, it cannot tell if a transmission failure is due to channel error or due to collision. The service loss caused by transmission failures should be compensated to achieve the bandwidth guarantee.

Now suppose that due to a short period of bursty channel errors on link B→D, f_2 is lagging behind its desired service rate r_2. Since link B→C is in the good state, f_1 can always maintain a rate of r_1. If node B runs a distributed scheduling algorithm capable of making service compensation, f_2 will be compensated for the lost service when link B→D recovers from the erroneous state.

B. End-to-End (Multihop) Compensation

We still use the same example as shown in Fig. 1, but errors are supposed to happen on link D→D2 instead. Node D will try to compensate f_2 similarly as node B. We argue that the local compensation made at node B is not sufficient for a multihop flow. As the service rate of f_2 on link D→D2 is degraded to a value below f_2's desired rate r_2 and the upstream link B→D is still being served at a rate of r_2, the buffer at node D may overflow and cause packet drops at node D. In fact, any packet dropped at an intermediate relaying node adds to the service loss at the destination node D2.

With a reliable transport protocol, the source node S2 will be informed about these service losses and retransmit the lost packets. We assume that the transport protocol will also make end-to-end compensation efforts by temporarily increasing the source sending rate to a value above r_i. Such an increase requires more transmission efforts at all downstream relaying nodes. Therefore, not only the source node S2, but also all f_2's relaying nodes should be informed about the service loss of f_2 and make compensation efforts at the link layer. To this end, we propose an end-to-end feedback mechanism, which is described in the next two subsections.

C. Feedback Mechanism

In order to implement end-to-end compensation for one flow, the source node and all relaying nodes of that flow should be informed about the achieved service rate at the destination. There are two design issues here: how does the destination node estimate the achieved rate and how to propagate this information to all upstream nodes.

To estimate the achieved rate, the destination node of any flow f_i uses a measurement window with time period t_m. The destination node keeps track of all packets of flow f_i in the current measurement window by adding the packet lengths to a variable L_i, which is reset to 0 at the start of each measurement window. At the end of each measurement window, flow f_i's achieved rate s_i is calculated as $s_i = L_i/t_m$.

When s_i is updated at the destination node, it is sent back to f_i's source node to complete the feedback loop. Such a route should be available if a reliable transport protocol is used. Since the forward and reverse paths may pass through different nodes, s_i may not be propagated to all f_i's relaying nodes along the reverse path. Therefore, it should first be sent back to the source node. When the source node has received it, the updated values of s_i and r_i are then propagated to all f_i's relaying nodes.

D. Propagation of r_i and s_i

The values of r_i and s_i for each flow f_i is used to control the scheduling and compensation at each relaying node, so the reliable delivery of r_i and s_i is crucial to the operation of the algorithm. We employ a simple mechanism for the propagation. Two new fields are added for r_i and s_i in the MAC layer data packet, as shown in Fig. 2. The source node of f_i writes the values of r_i and s_i into every data packet it sends. When a relaying node receives a data packet, it will use these two values to do the scheduling and forward the packet to the next hop with these values unchanged.

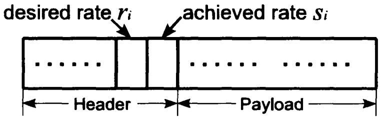


Fig. 2. Data Packet Format with Fields for r_i and s_i.

The data volume of r_i and s_i is extremely low compared with the payload of a data packet. It is easier to implement the mechanism as only link layer information is needed to do the propagation. The mechanism is also reliable since r_i and s_i are embedded into every data packet and always available to the scheduler.

The destination-to-source feedback and the propagation of the desired and achieved rates form an important mechanism for ECSA. The mechanism provides each relaying node with the global information so that each node can compensate local flows regarding the end-to-end service losses and thus help improve the end-to-end service guarantee.

IV. End-to-End Compensation Through Local Scheduling

With the end-to-end feedback mechanism proposed in the previous section, every relaying node has the information of the desired and achieved rates of all the flows that are relayed by it. End-to-end compensation should be achieved by packet scheduling based on these rates.

Since many wireless scheduling algorithms with error compensation have been proposed for infrastructured networks, they can be adopted here to meet the requirements. We take BGFS-EBA [6] as an example to show how to adapt such an algorithm to ad hoc networks and provide the end-to-end compensation to the multihop flows.

A. Adapt BGFS-EBA to Ad Hoc Networks

BGFS-EBA is an opportunistic scheduling algorithm for infrastructured wireless networks. It aims to provide bandwidth and delay guarantees for flows with an error-free link and allocate excess bandwidth among lagging flows (flows with achieved goodput smaller than its target share, typically caused by channel errors) in an equitable manner. The detailed description of BGFS-EBA algorithm is omitted due to space limitations. Interested readers can refer to [6] for details.

In order to adapt BGFS-EBA to each node in an ad hoc network, we first compare the requirements of a scheduling algorithm used in an infrastructured network, like BGFS-EBA, and one used in an ad hoc network:

- In an infrastructured wireless network, the scheduler runs at a base station. In an ad hoc network, the scheduler runs at each node.
- In an infrastructured wireless network, a base station has the ability to estimate the status of a physical link. It is not practical for an ad hoc node to have the status of each link in the network.
- In BGFS-EBA, time is divided into fixed-length time slots. In this work, we use the modified slotted ALOHA protocol as the medium access scheme in order to be consistent with BGFS-EBA.

The above discussion suggests that, to adapt BGFS-EBA to an ad hoc network, we need to:

1) Relax the requirement of knowledge of the channel status.
2) Utilize r_i and s_i to allocate the compensation efforts.
3) Use a simple MAC protocol that reports the result of each transmission and makes no retransmissions.

We call the adapted version of BGFS-EBA as Bandwidth Guaranteed Scheduling for Ad hoc networks (BGSA).

The requirement of "no automatic retransmission" on the MAC protocol is necessary for our design. For example, the retransmission mechanism of the IEEE 802.11 DCF protocol is not controlled by the scheduler. The medium access opportunities cannot be fully utilized for compensation. Thus, we use a modified version of slotted ALOHA. It is described in the next subsection.

B. Modified Slotted ALOHA Protocol

All nodes in an ad hoc network communicate over a single wireless channel. The operation of each node is synchronized at the boundary of each time slot using some existing synchronization protocols [7], [8]. When a DATA packet is passed down from the scheduler of a node, the node waits until the next time slot and sends it immediately with probability P, where P is a node specific parameter and $0 < P \leq 1$. P should be provided and adjusted by the admission control mechanism during the route establishment phase. If the packet is not transmitted in the current time slot, the node tries the next time slot with the same probability until the packet is transmitted. When a node has successfully received a DATA packet, it passes the packet to the scheduler and sends an ACK packet back to the transmitter immediately. A successful transmission consists of a DATA packet and an immediate ACK sent by the receiver. All events occur within the same time slot. The result of a transmission (either successful or unsuccessful) is reported to the scheduler.

The modified slotted ALOHA protocol is not a very efficient MAC protocol for ad hoc networks since it does not utilize carrier sensing or collision avoidance. We use it to demonstrate our end-to-end compensation framework only because of its simplicity. ECSA is open to other MAC protocols as long as they report each transmission result and do not retransmit packets automatically.

C. The BGSA Scheduling Algorithm

As described in Section IV-A, BGSA is an ad hoc version of BGFS-EBA. Fig. 3 shows the flow chart for the BGSA algorithm, where d_i, g_i and G_i are the same as in the original BGFS-EBA. At a certain node, there are (n-1) local flows $\{f_i\}$ under consideration, where $i = 1, 2, \cdots, n-1$. A local flow is defined as an end-to-end multihop flow that passes through the node. The scheduler gets the available bandwidth R of the node from the admission control scheme, where $R > \sum_{i=1}^{n-1} r_i$. We assume that R is estimated with the consideration of channel errors and contentions. Similar to BGFS-EBA, the algorithm consists of two phases in the virtual system and the real system, respectively. However, the link status is not taken into account. Moreover, a flow with the minimum $\frac{s_i}{r_i}$ has priority in receiving service compensation.

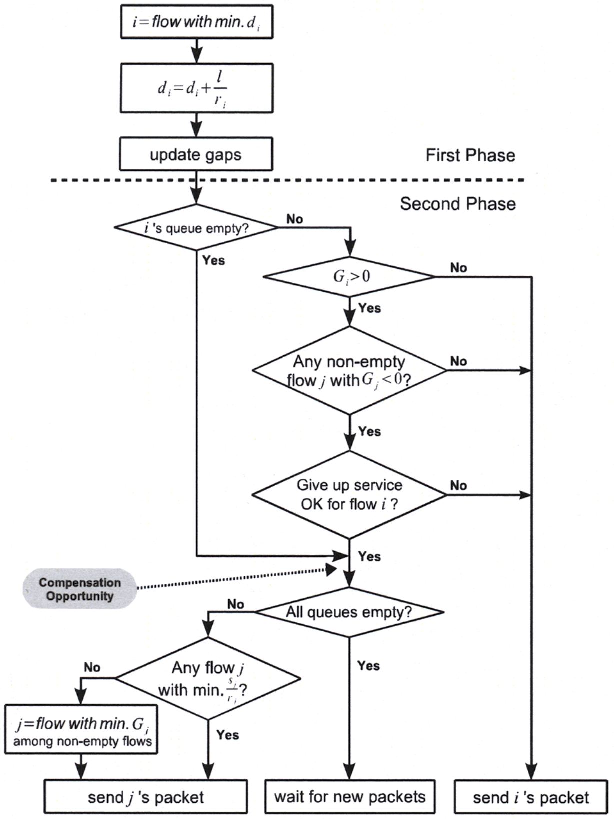


Fig. 3. Flow Chart for BGSA Algorithm.

V. Simulation Results

The simulation is performed in ns-2 [9], with the modified slotted ALOHA, BGSA and the end-to-end feedback mechanism implemented.

In order to simulate bursty channel errors, the physical model of the wireless medium is extended to support random error upon each packet reception. The probability that a packet transmission fails on a "good" link is zero. The probability that a packet transmission fails on a "bad" link can be set to a value between 0 and 1. The capacity of the wireless channel used in the simulation is 2 Mbps.

We assume a compensation-capable transport layer protocol is used to perform the end-to-end compensation. The behavior of such a transport protocol is emulated by adjusting the source sending rate in the simulation.

The simulation results of a BGSA scheduler over the modified slotted ALOHA MAC are compared with the results of the "drop tail with priority" (Queue/DropTail/PriQueue in ns-2) scheduler over the modified slotted ALOHA MAC to demonstrate the effectiveness of the proposed scheduling framework. To emulate the behavior of a MAC that invokes packet retransmissions, the DropTail scheduler is modified such that it always reschedules any failed packet with the highest priority unless it exceeds the given retransmission limit.

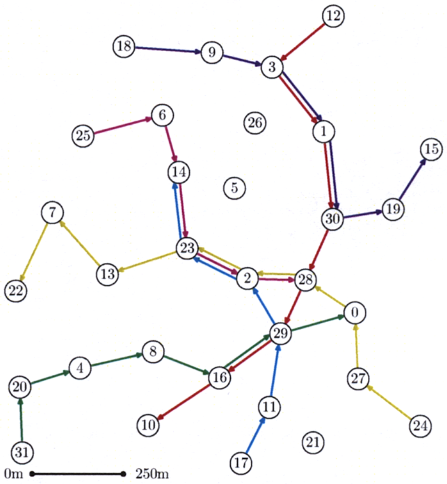


Fig. 4. Network Topology for Simulation.

The network topology used in the simulation is shown in Fig. 4. 32 nodes are placed in a 1250m×1250m flat grid. Six multihop CBR (constant bit rate) flows are created. Source routing is employed. The routes and the offered loads (OL) of these flows are listed in Table I.

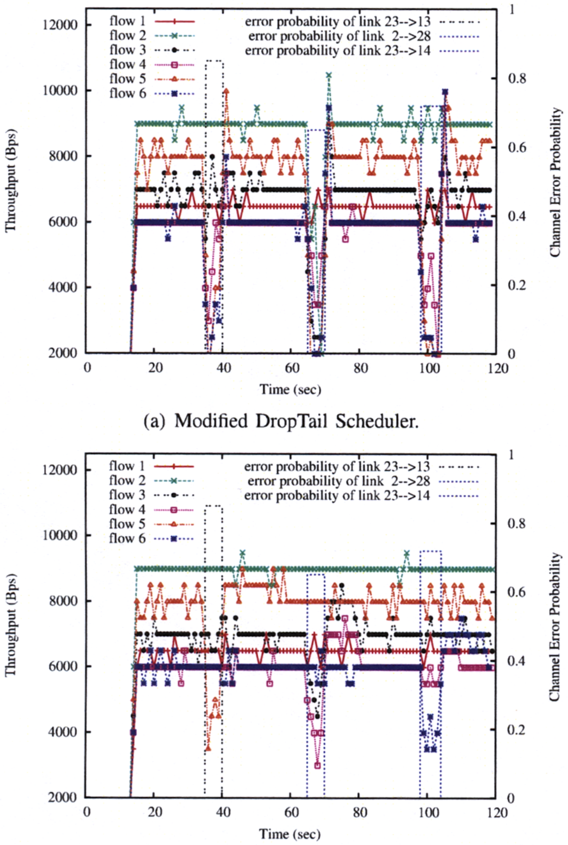
TABLE I
Routes and Offered Loads of End-to-end Flows.

Flow	Src	Dst	OL (KBps)	Route
f_1	31	0	6.5	30→20→4→8→16→29→0
f_2	18	15	9.0	18→9→3→1→30→19→15
f_3	12	10	7.0	12→3→1→30→28→29→16→10
f_4	25	28	6.0	25→6→14→23→2→28
f_5	24	22	8.0	24→27→0→28→2→23→13→7→22
f_6	17	14	6.0	17→11→29→2→23→14

Table II lists the three periods of time when a wireless link becomes "bad" during the simulation. The packet error probability of a "bad" wireless link is set to a high value during the channel error period.

TABLE II
CHANNEL ERROR PERIODS.

Time Period	Wireless Link	Error Probability
[35s, 40s]	23 → 13	0.85
[65s, 70s]	2 → 28	0.65
[98s, 104s]	23 → 14	0.7



(a) Modified DropTail Scheduler.

(b) BGSA Scheduler.

Fig. 5. Time-Throughput Curves for Various Schedulers.

Fig. 5(a) shows the throughput curves for these six flows under the modified DropTail scheduling algorithm. During the error period of link 23 → 13, f_5's throughput drops since it uses this link. However, although flows f_4 and f_6 do not use link 23 → 13, they still receive less service under the modified DropTail scheduler. The reason is that, when link 23 → 13 is in error, f_5's packets relayed by node 23 to node 13 tend to fail with a high probability and must be retransmitted repeatedly at node 23 under the modified DropTail algorithm. On the one hand, as node 23 keeps retransmitting these failed packets for f_5, the transmission opportunities for f_1 and f_6 are reduced at node 23. On the other hand, since node 23 becomes busier sending packets and a node cannot receive while sending, node 23 has less opportunity to receive packets for all three flows.

Fig. 5(b) shows the throughput curves of BGSA. First, f_4 and f_6's throughputs are not affected by the bursty errors on link 23 → 13. It is because BGSA has the ability to schedule packets with service compensation. When a packet transmission for f_5 at node 23 fails, the scheduler can choose not to retransmit it immediately but schedule a packet for f_4 or f_6 instead, since it may have a more urgent deadline. Moreover, after link 23 → 13 recovers from error, f_5 gets compensated for the service loss during the error period. f_4 and f_6 are not affected since BGSA only uses the excess bandwidth to compensate.

Similar results can be observed from Fig. 5 for the other two error periods.

VI. Conclusion and Future Work

In this work, we propose a novel end-to-end compensation framework, ECSA. It provides error compensation for multihop flows in an end-to-end manner through local compensation. The framework consists of the end-to-end information feedback and the propagation mechanisms. With ECSA, a scheduling algorithm originally designed for an infrastructured wireless network can be converted to its ad hoc counterpart with minimal modifications. We use BGFS-EBA as an example to demonstrate how it can be converted into BGSA, a distributed scheduling algorithm for an ad hoc network.

The simulation results show that, with the ECSA framework and the converted BGSA scheduling algorithm, the throughputs of end-to-end data flows in multihop ad hoc networks are notably improved in the presence of channel errors. Flows that do not pass through error-prone wireless links will not suffer serious service degradations. Flows that pass through error-prone wireless links will receive service compensation after the links recover.

Possible future work includes devising a more accurate and efficient admission control scheme, a more efficient local scheduling algorithm, and studying the interaction between ECSA and the upper layer protocol which is responsible for service compensation.

References

[1] L. Tassiulas and S. Sarkar, "Maxmin Fair Scheduling in Wireless Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 163–173, Jan. 2005.

[2] H.-L. Chao and W. Liao, "Fair Scheduling in Mobile Ad Hoc Networks With Channel Errors," *IEEE Trans. on Wireless Communications*, vol. 4, no. 3, pp. 1254–1263, May 2005.

[3] H. Luo, J. Cheng, and S. Lu, "Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks," *IEEE Trans. on Mobile Computing*, vol. 3, no. 1, pp. 86–98, Jan. 2004.

[4] B. Li, "End-to-End Fair Bandwidth Allocation in Multi-Hop Wireless Ad Hoc Networks," in *Proc. IEEE ICDCS 2005*, Jun. 2005, pp. 471–480.

[5] P. Goyal, H. M. Vin, and H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," *IEEE/ACM Trans. on Networking*, vol. 5, pp. 690–704, Oct. 1997.

[6] Y. Cao, K.-C. Leung, and V. O. K. Li, "Bandwidth-Guaranteed Fair Scheduling with Effective Excess Bandwidth Allocation for Wireless Networks," *IEEE Trans. on Wireless Comm.*, vol. 7, no. 6, Jun. 2008.

[7] L. Huang and T.-H. Lai, "On the Scalability of IEEE 802.11 Ad Hoc Networks," in *Proc. MobiHoc 2002*, Jun. 2002, pp. 173–182.

[8] T.-H. Lai and D. Zhou, "Efficient and Scalable IEEE 802.11 Ad-Hoc Mode Timing Synchronization Function," in *Proc. AINA 2003*, Xi'an, China, Mar. 2003, pp. 318– 323.

[9] The Network Simulator. [Online]. Available: http://nsnam.isi.edu/nsnam/