Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2006, Article ID 51610, Pages 1–11
DOI 10.1155/WCN/2006/51610

Improving TCP Performance over Wireless Ad Hoc Networks with Busy Tone Assisted Scheme

Qi He,[1] Lin Cai,[2] Xuemin (Sherman) Shen,[3] and Pinhan Ho[3]

[1] *Research In Motion (RIM), Ottawa, ON, Canada, K2K 3K2*
[2] *Department of Electrical and Computer Engineering, Faculty of Engineering, University of Victoria, Victoria, BC, Canada, V8W 3P6*
[3] *Department of Electrical and Computer Engineering, Faculty of Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1*

Received 1 August 2005; Revised 29 December 2005; Accepted 29 December 2005

It is well known that transmission control protocol (TCP) performance degrades severely in IEEE 802.11-based wireless ad hoc networks. We first identify two critical issues leading to the TCP performance degradation: (1) unreliable broadcast, since broadcast frames are transmitted without the request-to-send and clear-to-send (RTS/CTS) dialog and Data/ACK handshake, so they are vulnerable to the hidden terminal problem; and (2) false link failure which occurs when a node cannot successfully transmit data temporarily due to medium contention. We then propose a scheme to use a narrow-bandwidth, out-of-band busy tone channel to make reservation for broadcast and link error detection frames only. The proposed scheme is simple and power efficient, because only the sender needs to transmit two short messages in the busy tone channel before sending broadcast or link error detection frames in the data channel. Analytical results show that the proposed scheme can dramatically reduce the collision probability of broadcast and link error detection frames. Extensive simulations with different network topologies further demonstrate that the proposed scheme can improve TCP throughput by 23% to 150%, depending on user mobility, and effectively enhance both short-term and long-term fairness among coexisting TCP flows in multihop wireless ad hoc networks.

1. INTRODUCTION

It is well known that TCP performance degrades significantly in IEEE 802.11-based multihop wireless ad hoc networks [1–5] due to the TCP instability problem and the unfairness problem. The former may cause dramatic drop of the TCP throughput to zero, while the latter may lead to substantial throughput variation of the coexisting TCP flows. These two problems are closely related to the unreliable broadcast and the false link failure.

In ad hoc networks, there is no request-to-send/clear-to-send (RTS/CTS) to reserve channels and to avoid the hidden terminal problem for broadcast frames. In addition, no Data/ACK handshake has been devised for the sender to distinguish whether the broadcast is successful or not. Since many important network management and control signaling messages are delivered by broadcast in ad hoc networks, for example, the Address Resolution Protocol (ARP) request messages and route request messages, the low success rate of broadcast transmissions may significantly downgrade the whole network functionality and efficiency.

On the other hand, when a node (sender) fails to transmit data to its next-hop receiver for a certain period of time, the node simply assumes that the link is broken. The source node (in this paper, the *sender* refers to the node who transmits or broadcasts to its *one-hop* neighbors, and the *source node* is the node who transfers data through an *end-to-end* connection) is thus notified to discover a new route to the destination based on the assumption that the link failure event is due to user mobility and node failure, and so forth. However, a link failure event can be caused not only by user mobility and node failure, but also by link-layer contention. The later case is also referred to as the false link failure, where the intermediate node that fails to relay the data will also inform the source node to discover a new route by mistakenly assuming that the link is broken. The route discovering procedure is very time consuming and imposes great overhead to the network. The procedure also relies on broadcast, and a low success rate of broadcast transmissions will prolong the route discovering procedure.

These two problems interact with TCP's congestion control mechanism (window backoff and timeout), and have

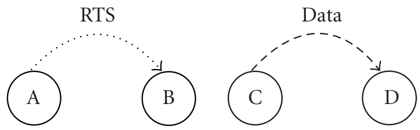


Figure 1: False link failure.

significant impact on TCP performance. Thus, to enhance TCP performance in the ad hoc networks, it is critically important to improve the success rate of broadcast transmissions and to detect and recover from false link failures promptly. Since it is impractical to use RTS/CTS for broadcast and link error detection frames, we propose to send control messages in a narrow-bandwidth, out-of-band channel, called busy tone channel, to reserve the data channel for broadcast and link error detection frames only.

Busy tone assisted schemes have been proposed in the literature for different purposes. In [6], a receiver-initiated busy tone scheme was proposed, where receivers set up busy tone during the receiving period to prevent transmission from hidden terminals. The receiver-initiated scheme is not applicable for both the broadcast frame which potentially has multiple receivers and the link error detection frame which may have no receiver at all. A dual busy tone scheme was proposed in [7], which uses two busy tone channels together with the RTS/CTS scheme to solve the hidden terminal and exposed terminal problems for unicast transmissions. This scheme requires both the sender and the receiver transmitting in two busy tone channels to protect RTS and data packets, and it is not suitable to protect the broadcast frame and the link error detection frame. Different from the previous approaches, our scheme allows the sender to send short control messages in the busy tone channel to protect broadcast and link error detection frames.

The main contributions of this paper are as follows. First, we propose a busy tone assisted broadcast and link error detection scheme with low overhead and good energy-efficiency, where only the sender needs to transmit two short messages in the busy tone channel before sending broadcast or link error detection frames in the data channel. Second, the success rates of the broadcast and link error detection frames are evaluated for both the proposed scheme and the standard IEEE 802.11 scheme. Numerical results demonstrate that the proposed scheme can effectively enhance the success rate of broadcast and link error detection frames. Third, by using NS-2, the TCP performances with and without using the proposed scheme are evaluated. Extensive simulation results with different network topologies demonstrate that the proposed scheme can improve TCP throughput by 23% to 150%, for both high-mobility and low-mobility cases, and enhance both short-term and long-term fairness among coexisting TCP flows in multihop wireless ad hoc networks.

The rest of the paper is organized as follows. Section 2 gives a brief introduction on the system model by discussing two severe problems in IEEE 802.11-based ad hoc networks and their negative impacts on TCP performance due to using a single channel. The busy tone assisted scheme is presented in Section 3, followed by its performance analysis in Section 4. In Section 5, simulation results are given to verify the performance gain of the proposed scheme. Related work is discussed in Section 6 and Section 7 concludes the paper.

2. SYSTEM MODEL

In the IEEE 802.11 standard, a wireless ad hoc network is defined as an independent basic service set (IBSS) deploying the distributed-coordination-function-(DCF-) based carrier sense multiple access with collision avoidance (CSMA/CA) medium access mechanism. CSMA/CA is effective and efficient in single-hop wireless networks such as infrastructure-based WLANs. However, it faces great challenges in the multihop scenarios. Here, we identify two problems, the unreliable broadcast problem and the false link failure problem, which result in the most serious impact on TCP performance.

2.1. Unreliable broadcast

Broadcast is very important in carrying critical information in the network, such as the routing information, ARP message, and node advertisement message, and so forth. Unlike wired networks, it is very difficult to provide reliable broadcast in wireless ad hoc networks due to the high bit error rate, wireless medium contention, and so forth. In the IEEE 802.11 standard, since no RTS/CTS is devised for broadcast frames, comparing with unicast frames, broadcast frames are more vulnerable to the hidden terminal problem. Furthermore, unlike unicast frames, there is no Data/ACK handshake for broadcast frames. Therefore, the broadcast senders cannot tell whether the broadcast frame is correctly received by all intended receivers or not.

If broadcast fails, some important network functions may fail. For instance, a route failure may happen if either the broadcast ARP request message or the route request message fails to be delivered. Consequently, the data transmission is frozen. Therefore, how to provide reliable broadcast in wireless ad hoc networks is critical.

2.2. False link failure

Link failure in wireless ad hoc networks may be due to node mobility, power attenuation, node failure, all of which can discontinue the sender forwarding packets to the next hop, and lead to the source searching for a new route to the destination. However, if a sender fails to transmit a frame due to link contentions, the source node may unnecessarily start to search for a new route by assuming a link failure event. We call it false link failure, as illustrated below.

As shown in Figure 1, node A attempts to transmit a data frame to node B while node C is transmitting a data frame to node D. Since node C is out of the carrier sensing range of node A, node A tries to send RTS to node B. Node B cannot receive the RTS frame successfully due to the collision with the data frame sent by node C. Thereafter, node A retransmits the RTS using an exponential backoff algorithm.

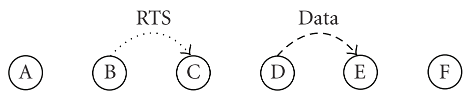


Figure 2: TCP instability problem.

However, since the transmission time of a data packet (e.g., with 1500 bytes) is usually much larger than the transmission time of the RTS (about 20 bytes), node A may fail to send RTS to node B for several times consecutively. Thus, false link failure may happen even when the traffic load is not heavy. Since all routes via this link need to be recalculated, false link failures bring significant overhead to the network.

2.3. Impacts on TCP performance

TCP faces two major problems in ad hoc networks, which are not encountered in wired networks [1–5]. The first is the throughput instability problem: the throughput of a TCP flow fluctuates severely and even frequently drops to zero. The second is the unfairness problem: when there are several TCP flows competing in the network, some flows tend to dominate the channel and the other flows are starved, even when all nodes are static. By examining the interactions between TCP and the IEEE 802.11 MAC protocol, the false link failure problem and unreliable broadcast problem play an important role for both the TCP instability problem and the unfairness problem.

(1) TCP instability problem

As shown in Figure 2, a TCP connection is established between node A and node F. When node D is transmitting a frame to node E, since the transmission is out of the sensing range of node B, node B attempts to send RTS to node C. At node C, the RTS sent by node B collides with the packet sent by node D. After retransmitting RTS seven times, node B assumes that a link failure occurs, which is a typical false link failure event.

Thereafter, the intermediate nodes discard all packets transmitted via the route and notify the source node of the route failure. The source node then broadcasts a route request message to search for a new route. Here, we consider Dynamic Source Routing (DSR) protocol as an example. Furthermore, due to the unreliable broadcast problem, the broadcast route request message and ARP request message can easily get lost due to link-layer contention. If either the route request message or ARP request message is lost, the new route cannot be established successfully.

As an on-demand routing protocol, DSR searches for a new route only when any packet is ready to be sent. On the other hand, TCP will retransmit the packet until the current transmission timeouts. TCP exponentially increases the timeout value after each retransmission. Therefore, it may take a fairly long time (several seconds) to resume the TCP transmission when a route failure happens. Until a new route is established successfully, the TCP sending rate drops to zero, leading to the TCP instability problem.

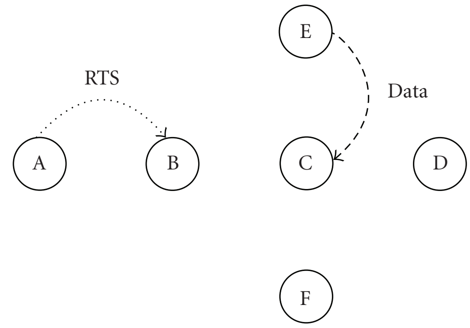


Figure 3: TCP unfairness problem.

(2) TCP unfairness problem

TCP suffers from severe unfairness problem due to many factors, such as hidden terminal and exposed terminal problems, capture effect, the adoption of the binary exponential backoff (BEB) scheme, and variation of hop lengths, and so forth. Besides, the false link failure and unreliable broadcast in the MAC layer that may cause serious impacts on the TCP throughput performance have not been fully addressed in the literature.

As illustrated in Figure 3, there are two flows competing with each other: flow 1 between node A and node D, and flow 2 between node E and node F. Due to the collision of the RTS (sent by node A) and the data frame (sent by node E), node A may trigger route failure. Due to the unreliable broadcast problem, it takes a long time for flow 1 to establish a new route and resume transmission. During this period, flow 2 completely captures the channel, which causes severe unfairness.

2.4. Why single channel is not sufficient

Since the RTS/CTS scheme is not applicable for broadcast transmission, is it possible to increase the carrier sensing range, to avoid the hidden terminal problem? To discuss this issue, we first define the transmission range, sensing range, and interference range as below [8].

The transmission range is the range (with respect to the transmitting station) within which a transmitted frame can be successfully received by receivers. The physical carrier sensing range is the range within which the signal-to-noise-ratio (SNR) is greater than or equal to the threshold by which the other stations can detect the transmission. The interference range is the range within which any station in the receiving mode can be interfered with the transmitter and thus loss of the packets. The interference range is usually larger than the transmission range, and it is a function of the distance between the sender and receiver.

To avoid the hidden terminal problem, the carrier sensing range should be set to be larger than the maximum interference range plus the transmission range. However, the sensing range is limited according to the physical sensitivity of the receiver (the receive threshold) and the sender transmission power. Increasing the transmission power will increase the maximum interference range, and thus it is not applicable. Since both unicast data frames and broadcast frames share the same channel, increasing sensing range by using more sensitive receiver may lead to more serious exposed terminal problem and may reduce network capacity.

From the above discussions, it is very difficult, if not impossible, to *efficiently* solve the problems with a single channel.

3. BUSY TONE ASSISTED SCHEME

3.1. Channel architecture

To alleviate the false link failure problem and unreliable broadcast problem, we introduce the busy tone assisted scheme. Besides the wide-bandwidth data channel, a separate narrow-bandwidth busy tone channel is used for control purpose. We assume that these two channels are completely orthogonal, and thus the interference between these two channels is negligible. With different transceivers working in different channels, each node can transmit or receive in both the data channel and the busy tone channel simultaneously.

We can set the carrier sensing range in the *busy tone channel* equal to the maximum interference range plus the transmission range in the *data channel* to solve the hidden terminal problem for broadcast frames, as explained in Section 3.2. On the other hand, for broadcast, the exposed terminal problem is less severe than that for unicast or multicast since all nodes within the transmission range need to receive the broadcast frame successfully. Therefore, increasing the sensing range in the busy tone channel only will not significantly exaggerate the exposed terminal problem.

For simplicity, we assume the transmission range, the carrier sensing range, and the maximum interference range for each channel to be the same. Our scheme can be easily extended to the case that these three ranges are different for each channel.

The unicast data transmission scheme in the data channel remains mostly unchanged, which uses RTS/CTS to alleviate the hidden terminal and exposed terminal problems. In addition, all nodes should monitor the busy tone channel. Only the sender of broadcast frames or link error detection frames takes advantages of the busy tone channel to reserve the data channels, and all the other nodes should not transmit if a node has made a successful reservation.

3.2. Broadcast

In the study, time is slotted. The duration of each slot is τ, which is long enough to include the one-hop propagation delay, carrier sensing delay, processing delay, and transmit/receive turnaround time. There are two different busy tone messages used for broadcast: PILOT and broadcast notification (BN). Once receiving a PILOT in the busy tone channel, a node sets a timer TR_c which is equal to $\beta\tau$, where β is a system parameter. Before timeout, the node cannot initiate any broadcast.

To initiate a broadcast, the node sends a PILOT in the busy tone channel first, and then monitors the busy tone channel for a random delay time T_{rand}. The random delay time T_{rand} equals $N_{rand}\tau$, where N_{rand} is a random number. To give a higher priority to link error detection frames, N_{rand} is chosen as follows: $0 \leq N_{rand} \leq \alpha$ for link error detection frames as discussed in Section 3.3, and $0 \leq N_{rand} \leq \beta$ for normal broadcast frames, where α is a system parameter and $\alpha < \beta$. PILOT and random backoff T_{rand} are used to resolve collisions among competing broadcast frames. Once receiving a PILOT, any node is not allowed to initiate broadcast in order to reduce competition among broadcast frames, but it is still allowed to use the data channel for on-going data transmission.

If the busy tone channel is idle during T_{rand}, the sender sends BN in the busy tone channel to reserve the channel for the incoming broadcast frame. After broadcasting BN, the sender waits for time T_{max}, and then broadcasts the data packets in the data channel. T_{max} is set to be large enough to finish all on-going data/ACK transmissions starting before BN: $T_{max} = 3\text{SIFS} + tx(\text{RTS}) + tx(\text{CTS}) + tx(\text{MTU}) + tx(\text{ACK})$, where SIFS stands for the short interframe spacing, $tx(\text{MTU})$ is the transmission time of a maximum transmit unit, $tx(\text{RTS})$, $tx(\text{CTS})$, $tx(\text{ACK})$ represent the transmission time of RTS, CTS, and ACK, respectively.

Once receiving the BN, a node reserves the interval $[T_{max} + t, T_{max} + T_{mb} + t]$ in its local table, where t is the time when it receives BN, T_{mb} is the maximum broadcast duration time. Therefore, the node must stop data transmission before $T_{max} + t$, and it must keep silent during $[T_{max} + t, T_{max} + T_{mb} + t]$.

The steps taken by a broadcast sender are given as follows.

(1) Before broadcasting, the sender first checks whether its timer TR_c is active. If the timer is active, it should defer broadcasting until timeout.

(2) If the timer TR_c is not active, the sender should check whether or not any reserved interval in its local table overlaps with its own requested broadcasting interval. The sender's own broadcasting interval is calculated as $[\text{PIFS} + tx(\text{PILOT}) + tx(\text{BN}) + T_{max} + t, \text{PIFS} + tx(\text{PILOT}) + \beta\tau + tx(\text{BN}) + T_{max} + T_{mb} + t]$, where PIFS stands for point interframe spacing which equals SIFS + 1 slots, and t is the current time instance. If the sender's own broadcasting interval overlaps with any reserved interval in its local table, it should exponentially back off.

(3) If there is no reserved interval that overlapped with its own broadcasting interval, the sender should sense the busy tone channel for PIFS. If the busy tone channel is idle, it transmits PILOT in the busy tone channel and then waits for a random delay time T_{rand}; otherwise, it should exponentially back off.

(4) If the channel is idle after T_{rand}, the sender broadcasts BN to notify the incoming broadcast attempt to all its neighbors, and then waits for T_{max}; otherwise, it should exponentially back off.

(5) After waiting the maximum duration time T_{max}, the sender senses the data channel for PIFS. If the data channel is idle, the sender broadcasts in the data channel; otherwise, it should exponentially back off.

By reserving the channel before broadcasting, the proposed scheme reduces the collision probability of the broadcast packets. The hidden terminal problem can be eliminated because the sensing range of the busy tone channel equals the sensing range plus the transmission range of the data channel.

3.3. Link error detection

Busy tone channel is used to identify the false link failures. Instead of triggering the link failure right after retransmitting for the maximum times, the sender enters the link error detection phase and tries to identify whether it is a real link failure or not.

The procedure of link error detection is similar to the broadcast procedure with only a few differences which enable them to incorporate well. Steps (1) to (4) are the same as that of the broadcast procedure.

(5) After waiting the maximum duration time $T_{\max}$, if the data channel is sensed idle for PIFS, the sender launches a control frame SI (status inquire) in the data channel to the suspected failed node in order to probe for the status of that node; otherwise, it should exponentially back off.

(6) If the receiver receives the SI, it should reply with a control frame SR (status response) in the data channel after SIFS.

(7) If the sender receives the SR correctly, it marks the link as available, exits link error detection phase, and resumes data transmission. Otherwise, it discards the data frame, marks the link as unavailable, exits link error detection phase, and reports link failure to the source node.

3.4. Collision due to mobility

Because the sender should wait $T_{\max}$ before it broadcasts a data packet, a mobile node which is outside the sensing range of the sender may move into the two-hop neighborhood of the sender and cause collision after $T_{\max}$. However, since $T_{\max}$ is very small and the node can only move a very small distance during $T_{\max}$, the probability of collision due to user mobility is negligibly small, as illustrated in the following example.

As defined earlier, $T_{\max} = 3\text{SIFS} + tx(\text{RTS}) + tx(\text{CTS}) + tx(\text{MTU}) + tx(\text{ACK})$. If the channel bandwidth is 2 Mbps, and the maximum packet size is 1500 byte,

$$T_{\max} = 3 \cdot 10 \cdot 10^{-6} + \frac{(20+14+14+1500)8}{2 \cdot 10^{6}} + 4 \cdot 192 \cdot 10^{-6} \approx 7\,\text{ms}. \tag{1}$$

Even with a speed at 120 km/hour, the moving distance within 7 milliseconds is only about 0.2 m. Moreover, we can slightly increase the sensing range by $2VT_{\max}$ to eliminate the collision due to mobility, where V is the maximum speed of mobile nodes.

4. PERFORMANCE ANALYSIS

The objective of the proposed scheme is to improve the success rate of broadcast and link error detection frames by reducing the collisions due to hidden terminals. Therefore, we compare the collision probability of broadcast frames of the proposed scheme with that of the legacy IEEE 802.11 MAC. The collision probability for link error detection frames can be obtained in a similar way, which is not presented here due to space limitation.

We assume that the node spatial distribution is two-dimensional Poisson distribution with λ as the average number of nodes per unit area. Therefore, the probability that i nodes appear in a circular region with radius R is

$$p(i,R) = \frac{(\lambda\pi R^2)^i e^{-\lambda\pi R^2}}{i!}. \tag{2}$$

For each time slot, the node broadcasts with probability p and keeps silent with probability $1-p$.

4.1. Collision probability of the busy tone scheme

For the proposed scheme, since there is no hidden terminal problem, the broadcast fails only if multiple nodes broadcast simultaneously within the two-hop neighborhood. Here, simultaneously means two events occurring at the same time slot. If only one node sending PILOT in one slot, all its two-hop neighbors can know the broadcast, and they should refrain from initialize broadcasts to avoid collision. The probability of only node S sending PILOT within the circular region with radius $2R$ is

$$P_1 = \sum_{i=0}^{\infty} p(1-p)^i \frac{(4\lambda\pi R^2)^i e^{-4\lambda\pi R^2}}{i!}. \tag{3}$$

After broadcasting the PILOT, node S further randomly chooses N_{rand} between 0 and β. The probability that the sender S eventually succeeds to broadcast when its PILOT has collided with PILOTs from other N nodes is

$$P' = \sum_{n=1}^{\beta-1} \frac{1}{\beta}\left(1-\frac{n}{\beta}\right)^N. \tag{4}$$

This is because S can succeed only if the chosen N_{rand} is smaller than those chosen by the other N nodes.

Therefore, the probability of node S successfully broadcasting given that its PILOT has collided with PILOTs from other nodes is

$$P_2' = \sum_{i=1}^{\infty}\left(p\sum_{N=1}^{i}\left(\frac{i!}{N!(i-N)!}p^N(1-p)^{i-N}\sum_{n=1}^{\beta-1}\frac{1}{\beta}\left(1-\frac{n}{\beta}\right)^N\right)\right) \times \frac{(4\lambda\pi R^2)^i e^{-4\lambda\pi R^2}}{i!}. \tag{5}$$

The success probability of a broadcast frame (there is no collision to this frame) is

$$\Pr\{\text{Success}\} = P_1 + P_2' \tag{6}$$

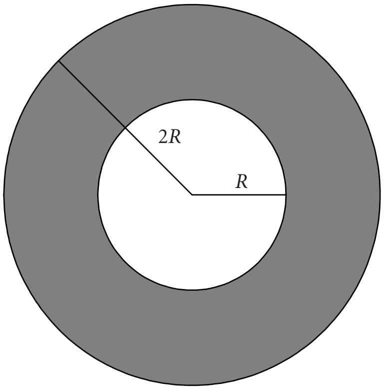


FIGURE 4: Hidden terminal area.

4.2. *Collision probability of IEEE* 802.11 *MAC*

For the IEEE 802.11 MAC, we need to consider the hidden terminal problem, which is illustrated in Figure 4.

All nodes within the carrier sensing range R can directly sense the broadcast of sender S and refrain themselves from transmission. However, all of the hidden terminals located between R and $2R$ cannot sense the broadcast and may broadcast during the period when sender S is broadcasting. Therefore, we divide the whole circular region with radius $2R$ into two areas: the carrier sense area and the hidden terminal area, as indicated in Figure 4.

The hidden terminal area is

$$A_H = 4\pi R^2 - \pi R^2 = 3\pi R^2. \tag{7}$$

The probability that only sender S broadcasts within the carrier sense area in a particular slot is

$$P_C = \sum_{i=0}^{\infty} p(1-p)^i \frac{(\lambda\pi R^2)^i e^{-\lambda\pi R^2}}{i!}. \tag{8}$$

The probability that none of the nodes broadcasts within the hidden terminal area is

$$P_H^0 = \sum_{i=0}^{\infty} (1-p)^i \frac{(3\lambda\pi R^2)^i e^{-3\lambda\pi R^2}}{i!}. \tag{9}$$

Considering the hidden terminal problem, the broadcast of sender S can be successful if (a) there is no other packet scheduled for transmission during the interval $(t-T, t+T)$ in the hidden terminal area, where t is the time instance when sender S broadcasts, T is the number of slots to transmit the broadcast packet; or (b) even if there is another transmission, none of the one-hop neighbors of S is in the transmission range of the other transmission. The probability that nobody is located in the collision region when i nodes transmit in the hidden terminal area in a vulnerable time slot is P_E^i, and the probability that i hidden terminals transmit in a vulnerable time slot is P_H^i.

With the two-dimensional Poisson distribution given the existence of a hidden terminal, the probability that the hidden terminal's distance to S equals x is $2x/3R^2$, where $R \le x \le 2R$. Therefore, $P_E^1 = \int_R^{2R} 2x/(3R^2) e^{-\lambda y} dx$, where $y = 2R^2 \arccos(x/2R) - x\sqrt{R^2 - x^2/4}$ is the intersection area of two circles with radius R, and the distance between the two centers is x.

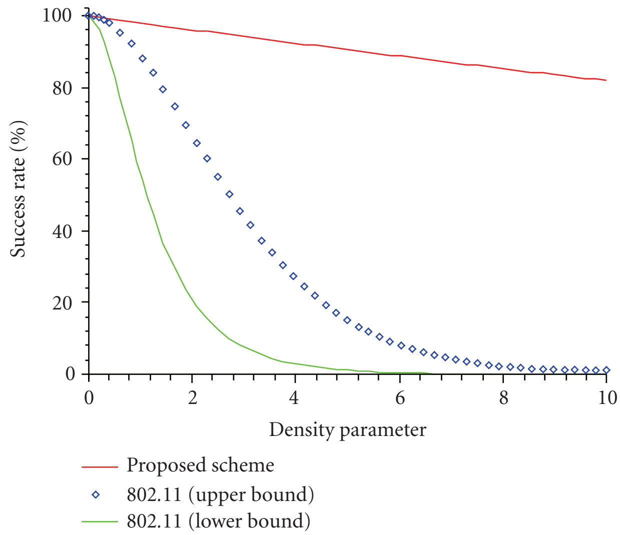


FIGURE 5: Density parameter ($\lambda\pi R^2$) versus success rate.

The probability that a broadcast frame is successfully delivered is

$$\Pr\{\text{Success}\} = P_C \left(P_H^0 + \sum_{i=1}^{\infty} P_H^i P_E^i \right)^{2T}. \tag{10}$$

Since the collision region is no larger than πR^2 and $P_E^1 \ge P_E^i$, P_E^i can be bounded by $P_E^1 \ge P_E^i \ge e^{-\lambda\pi R^2}$ for $i \ge 1$. The lower bound of the success probability is $P_C(P_H^0 + (1 - P_H^0)e^{-\lambda\pi R^2})^{2T}$, and the upper bound is $P_C(P_H + (1-P_H)P_E^1)^{2T}$. According to the numerical results in Section 4.3, the derived upper and lower bounds of the success probability are quite tight.

4.3. *Numerical results*

The analytical results are visualized by numerical results with the following parameters: $\beta = 4$ and $T = 30$.

Figure 5 shows the relationship between $\lambda\pi R^2$ and broadcast success rate, where the broadcast probability p is 0.005. It can be seen that the success rate decreases as the density increases because more nodes compete for a channel, leading to more collisions. The success rate with the proposed scheme decreases slowly while the success rate with the IEEE 802.11 MAC quickly drops to zero since IEEE 802.11 suffers from the hidden terminal problems especially when the node density is high.

Figure 6 shows the relationship between broadcast probability p and broadcast success rate, where $\lambda\pi R^2$ equals 5. The success rate with the proposed scheme decreases slightly when p becomes larger, while the success rate decreases exponentially with the IEEE 802.11 MAC.

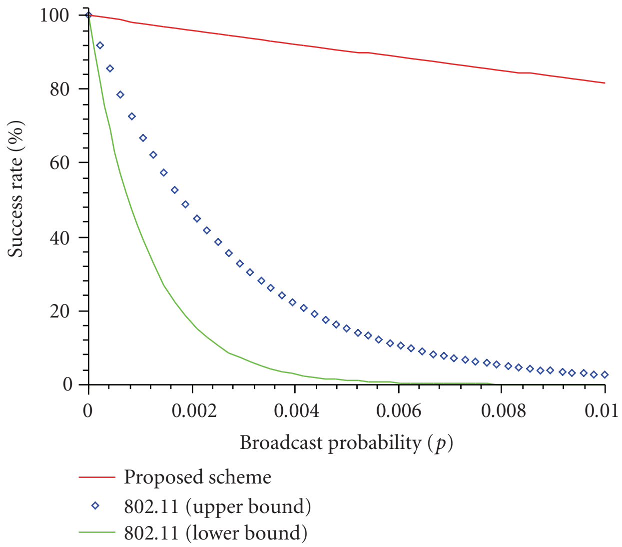


FIGURE 6: Broadcast probability versus success rate.

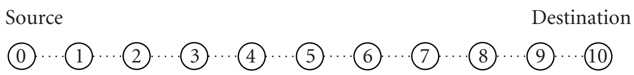


FIGURE 7: Chain topology.

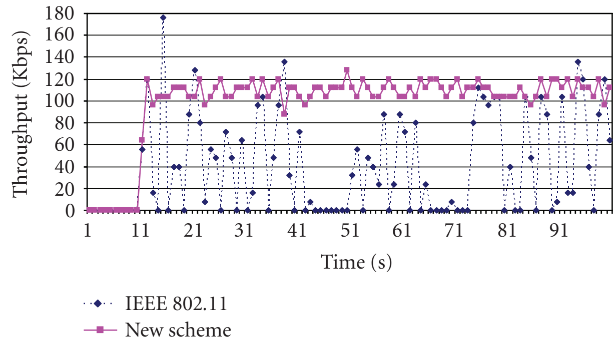


FIGURE 8: Instantaneous throughput of chain topology (10 hops).

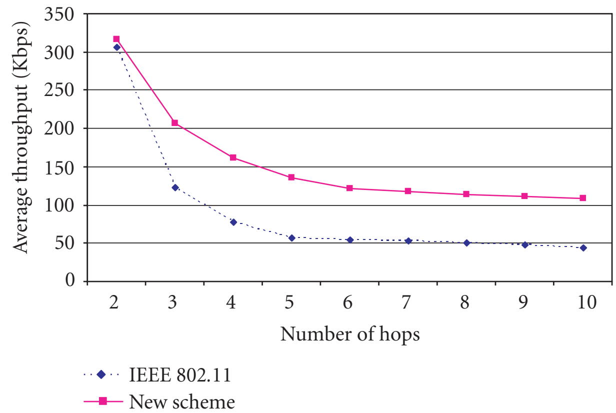


FIGURE 9: Average throughput of chain topologies with different hops.

The numerical results in Figure 6 show that the proposed scheme can dramatically enhance the success rate of delivery of broadcast and link error detection frames. On the other hand, given a fixed number of broadcast requests, p is inversely proportional to the success rate. Therefore, given the same number of broadcast requests, the lower p makes the proposed scheme well outperform the IEEE 802.11 MAC.

5. SIMULATIONS

We further evaluate the TCP performance with the proposed scheme by using the NS-2 simulator. The following parameters are used in the simulations. The sensing range of the data channel R is 250 meters, and the link-layer buffer size is 50 packets with a drop-tail first-in-first-out (FIFO) queue. DSR is taken as the routing protocol. The bandwidth of the wireless links is 1 Mbps, and the data packet size is 1000 bytes. TCP newReno [9] is used for large file transfers (infinite backlog).

5.1. Throughput

We consider three different topologies: the static chain topology, the static cross-topology, and the random mobile topology.

The static chain topology with 10 hops is illustrated in Figure 7. There is only one single flow transmitting from node 0 to node 10. The distance between the neighboring nodes is 200 m, and each simulation lasts for 100 seconds.

Ideally, since all nodes are static with only one single flow, there should be no route failure due to mobility and no competition with other TCP flows, and the throughput of TCP should be stable. However, as shown in Figure 8, the TCP throughput over IEEE 802.11 seriously fluctuates and frequently drops to zero, which shows a typical TCP instability phenomenon. On the other hand, with the proposed scheme, TCP has a fairly stable throughput, and the average throughput of TCP with the proposed scheme is 109 Kbps, which is around 250% of that with IEEE 802.11.

We further examine the average throughput of TCP over chain topologies with different hops, varying from two to ten. The results are shown in Figure 9. The TCP throughput decreases as the hop number increases because more serious link contention is caused. For the two-hop chain topology, the throughput of the proposed scheme is almost the same as IEEE 802.11 because the false link failure does not occur. As the hop number increases, the performance with the proposed scheme becomes much better than that with IEEE 802.11. The simulation results demonstrate the effectiveness of the proposed scheme to detect and recover from false link failures.

In the second scenario, we consider a static cross-topology with 12 nodes as shown in Figure 10. In this scenario, all nodes are static and placed in two lines with the

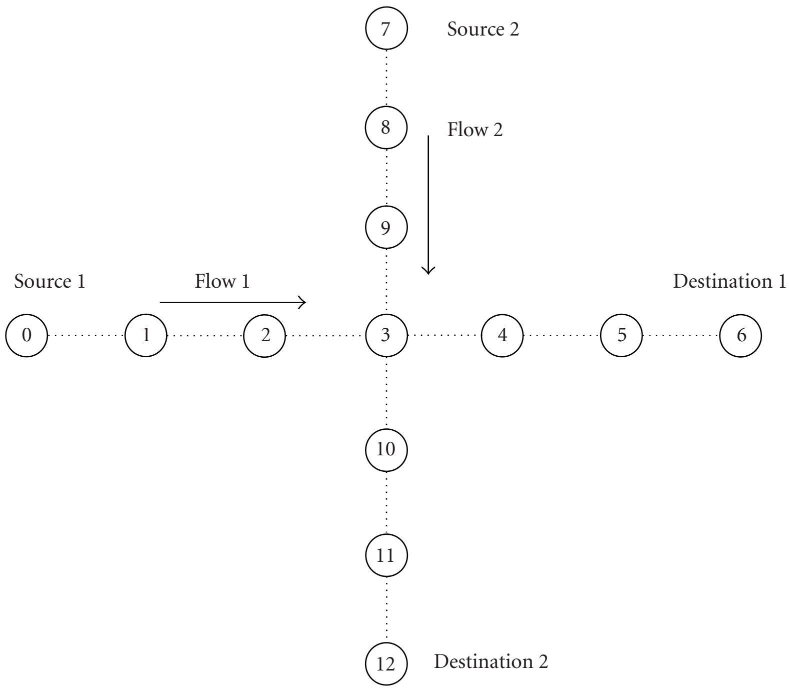


Figure 10: Cross-topology with 12 nodes.

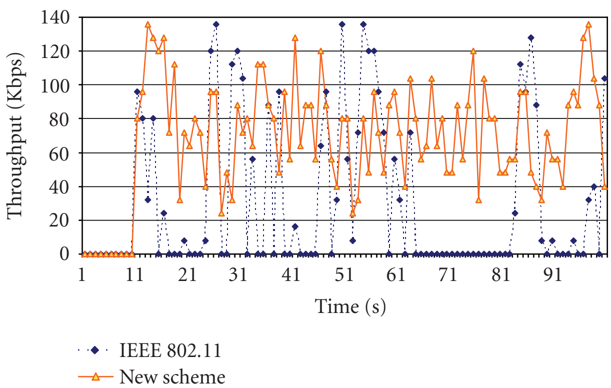


Figure 11: Instantaneous throughput of flow-1 in cross-topology.

distance between neighboring nodes 200 m. Two TCP flows are transmitted: flow 1 is from node 0 to node 6 starting at 10 seconds, while flow 2 goes from node 7 to node 12 starting at 15 seconds. Each simulation lasts for 100 seconds.

As shown in Figures 11 and 12, both TCP flows' throughputs with the proposed scheme are more stable than those with IEEE 802.11, and the throughputs do not drop to zero. The aggregate throughput of the two flows with the proposed scheme is around 200% of that with IEEE 802.11.

With the random mobile topology, the random waypoint mobility model [10] is deployed. There are 50 nodes moving randomly in a rectangular area of $1500 \times 1500\,\text{m}^2$ with the maximum speed of 10 m/s and the mean pause time of 2 seconds. At most, three TCP flows coexist with arbitrary source and destination pairs. Each simulation lasts for 150 seconds. To ensure fair comparison, in each experiment, the same topology and mobile pattern with the same set of source and destination pairs are used for both MAC schemes.

With the proposed scheme, the average aggregate throughput can achieve around 23% improvement, which is less than that with the static topologies. This is because in random mobile topology, link failures due to node mobility are frequent and network partitions are common, which also affect TCP performance.

5.2. Fairness

We examine fairness in a strict sense: only the throughput of TCP flows traversing a similar path and facing similar contentions is compared.

With the cross-topology, the two competing flows are symmetric. Therefore, ideally, they should achieve the same throughput. However, as shown in Figures 11 and 12 with IEEE 802.11, one flow tends to dominate the channel for a while and the other starves during that period. With the proposed scheme, none of the TCP flows starves during the whole simulation period and their average throughputs are approximately the same.

To evaluate the fairness issue, we adopt the Jain's fairness index (FI) [11], which is defined as follows:

$$FI = \frac{\left(\sum_{i=1}^{N} T_i\right)^2}{N\sum_{i=1}^{N} T_i^2}, \tag{11}$$

where T_i is the throughput of TCP connection i, and N is the

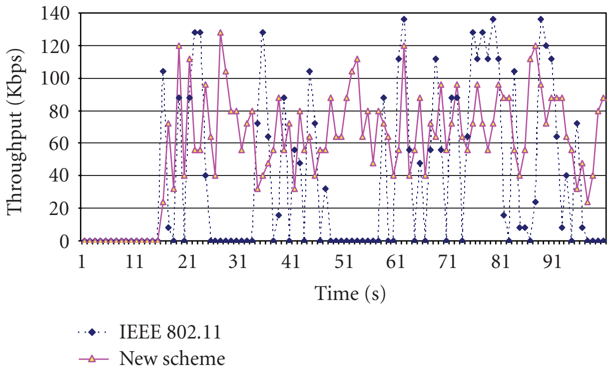


FIGURE 12: Instantaneous throughput of flow-2 in cross-topology.

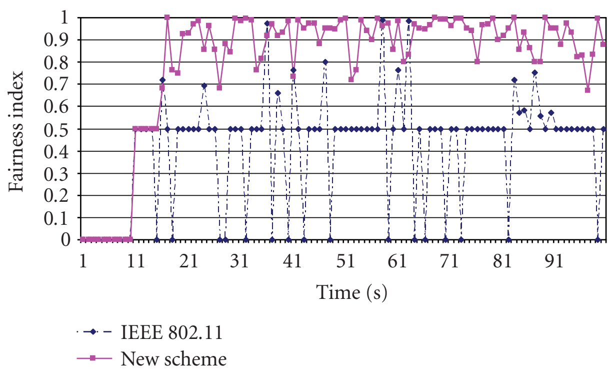


FIGURE 13: instantaneous fairness index in cross-topology (two flows).

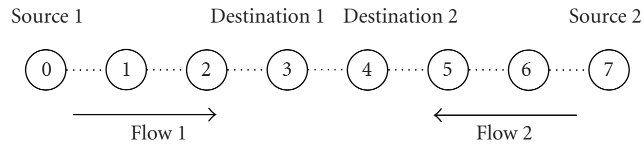


FIGURE 14: Chain topology with two competing flows.

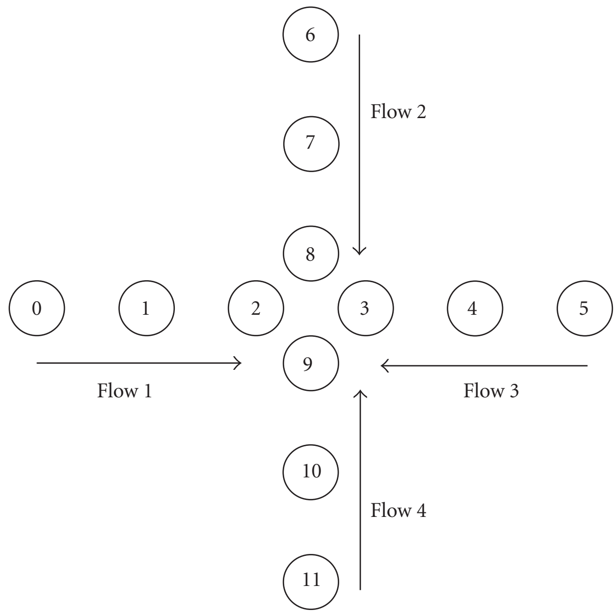


FIGURE 15: Cross-topology with four competing flows.

total number of connections. *FI* takes the value between 1 (best) and $1/N$ (worst).

The instantaneous fairness indexes of TCP flows with the proposed scheme and with IEEE 802.11 are shown in Figure 13. It can be seen that the proposed scheme achieves better short-term and long-term fairness than IEEE 802.11. With IEEE 802.11, the fairness problem is mainly due to extensive link contention, which generates extensive false link failures and causes a specific TCP flow being starved.

We further evaluate TCP fairness with two other topologies. The first one is an 8-hop chain topology with two competing flows transmitting in opposite directions, as shown in Figure 14. With the chain topology, node 3 and node 4 are in the same interference area, and node 2 and node 5 cannot sense the transmission of each other. Therefore, the transmission from node 2 to node 3 and the transmission from node 5 to node 4 may interfere with each other and cause false link failures.

The second topology is a cross-topology with 12 nodes, as shown in Figure 15. The distance between neighboring nodes is 200 m except the distance between nodes 2, 3, 8, and 9, which are in the same contention area with a radius of 100 m. Totally, four TCP flows compete with each other with the source destination pairs as (0, 2), (5, 3), (6, 8), and (11, 9), respectively. With the cross-topology, potentially more link contention exists and more severe unfairness problem could be introduced.

As shown in Figures 16 and 17, the proposed scheme can detect and recover false link failure quickly and achieve higher *FI* than IEEE 802.11 in both topologies.

6. RELATED WORK

The TCP instability problem in 802.11-based wireless ad hoc networks was first reported in [1], and it was further investigated in [2]. The TCP unfairness problem in multihop wireless ad hoc networks was reported in [4], and it was further investigated in [5]. However, the important role of unreliable broadcast on TCP fairness problem has not been addressed until the recent research in [12].

In [2], it was observed that increasing retransmission limits in the MAC layer will result in significant improvement in TCP performance. However, as indicated by the authors, increasing retransmission limit may lead to longer link breakage detection latency especially in a mobile environment. Moreover, packets from other flows sharing the same queue may be delayed as well.

In [13], a solution to alleviate this problem by modifying the 802.11 backoff algorithm was proposed with the assumption that CTS loss is always due to collision. Thus, the

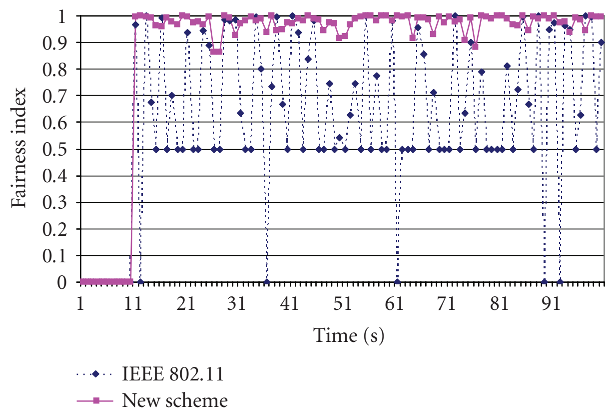


FIGURE 16: Instantaneous fairness index of chain topology (two flows).

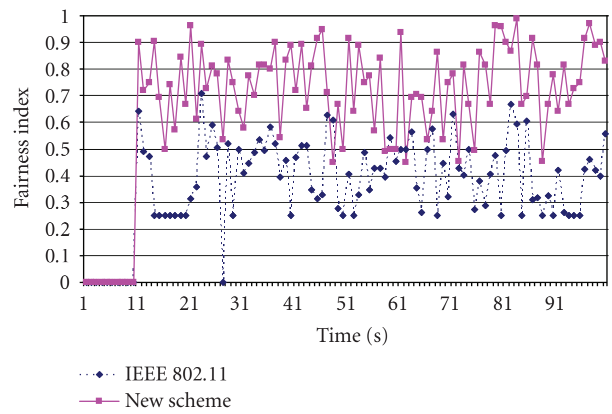


FIGURE 17: Instantaneous fairness index of cross-topology (four flows).

sender retransmits RTS after backoff for a relatively long time comparing with that defined in the IEEE 802.11 standard. However, this assumption is not always true, and thus this solution may not work well in a mobile environment.

Since the throughput instability problem is a common problem suffered by various wireless ad hoc routing protocols, such as DSR, AODV, DSDV, and so forth, a solution of continuously using the previous route until a new route is established was proposed in [3]. However, it cannot solve the fairness problem for a new setup flow.

A multichannel scheme was developed in [12] to reduce collisions in the MAC layer with split channels. However, splitchannels schemes need to divide the whole bandwidth among different channels, which may result in lower throughput. How to divide the bandwidth elegantly among these channels needs further investigations.

In [6], a receiver-initiated busy tone scheme was proposed. In [7, 14], dual busy tone channels are used to alleviate the hidden terminal problem for unicast data transmission, and the false link failure problem is unsolved. Here, we use a sender-initiated busy tone channel to protect broadcast and link error detection frames.

In [1, 2, 5, 15], it was noticed that a small congestion window limit (CWL) can achieve better TCP performance, and usually the CWL should be set around 1 or 2 packets. The recent work of [16] further demonstrated that the upper bound of CWL is approximately 1/5 of round-trip hop-count. An adaptive CWL setting algorithm was also proposed. Reference [17] found that TCP always increases its window much larger than the optimal size and leads to throughput degradation. The authors proposed using Link RED and adaptive pacing schemes to help TCP window being stabilized around the optimal size, by dropping extra packets and increasing the MAC layer backoff time. The end-to-end enhancement, RED queue management, and adaptive pacing schemes are orthogonal to our proposed scheme, and they can be deployed together to enhance TCP performance in ad hoc networks.

7. CONCLUSIONS

By examining the interactions between TCP and the IEEE 802.11 MAC protocol, we have identified that the false link failure and the unreliable broadcast have significant negative impacts on TCP performance in terms of throughput stability and fairness. We have proposed a sender-initiated busy tone scheme to alleviate these impacts. The proposed scheme can improve TCP performance without any modification on the TCP protocol. Both analytical and simulation results have demonstrated that the proposed scheme can effectively improve the reliability of broadcast frames and promptly detect and recover from false link failures. Consequently, higher TCP throughput and better fairness can be achieved.

REFERENCES

[1] S. Xu and T. Saadawi, “Revealing and solving the TCP instability problem in 802.11 based multi-hop mobile ad hoc networks,” in *Proceedings of 54th IEEE Vehicular Technology Conference (VTC ’01)*, vol. 1, pp. 257–261, Atlantic City, NJ, USA, October 2001.

[2] R. Jiang, V. Gupta, and C. V. Ravishankar, “Interactions between TCP and the IEEE 802.11 MAC protocol,” in *Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX ’03)*, vol. 1, pp. 273–282, Washington, DC, USA, April 2003.

[3] P. C. Ng and S. C. Liew, “Re-routing instability in IEEE 802.11 multi-hop ad-hoc networks,” in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN ’04)*, pp. 602–609, Tamba, Fla, USA, November 2004.

[4] S. Xu and T. Saadawi, “Revealing TCP unfairness behavior in 802.11 based wireless multi-hop networks,” in *Proceedings of the 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’01)*, vol. 2, pp. E-83–E-87, San Diego, Calif, USA, September-October 2001.

[5] K. Xu, S. Bae, S. Lee, and M. Gerla, “TCP behavior across multihop wireless networks and the wired internet,” in *Proceedings*

of the 5th International Workshop on Wireless Mobile Multimedia (WOWMOM '02), pp. 41–48, Atlanta, Ga, USA, September 2002.

[6] C. Wu and V. O. K. Li, "Receiver-initiated busy-tone multiple access in packet radio networks," in *Proceedings of the ACM Workshop on Frontiers in Computer Communications Technology (SIGCOM '87)*, pp. 336–342, Stowe, Vt, USA, 1987.

[7] Z. J. Haas and J. Deng, "Dual busy tone multiple access (DBTMA)—a multiple access control scheme for ad hoc networks," *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 975–985, 2002.

[8] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "Wi-fi in ad hoc mode: a measurement study," in *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom '04)*, pp. 145–154, Orlando, Fla, USA, March 2004.

[9] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," RFC 2582, April 1999.

[10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, chapter 5, pp. 153–181, Kluwer Academic, Norwell, Mass, USA, 1996.

[11] R. K. Jain, *The Art of Computer Systems Performance Analysis*, John Wiley & Sons, New York, NY, USA, 1991.

[12] T. Kuang and C. L. Williamson, "A bidirectional multi-channel MAC protocol for improving TCP performance on multihop wireless ad hoc networks," in *Proceedings of the 7th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM '04)*, pp. 301–310, Venice, Italy, October 2004.

[13] K. Kanth, S. Ansari, and M. H. Melikri, "Performance enhancement of TCP on multihop ad hoc wireless networks," in *Proceedings of IEEE International Conference on Personal Wireless Communications (ICPWC '02)*, pp. 90–94, New Delhi, India, December 2002.

[14] Z. J. Haas, J. Deng, and S. Tabrizi, "Collision-free medium access control scheme for ad-hoc networks," in *Proceedings of IEEE Military Communications Conference (MILCOM '99)*, vol. 1, pp. 276–280, Atlantic City, NJ, USA, October-November 1999.

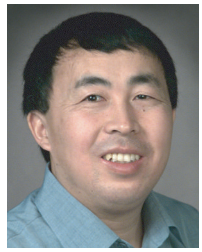
[15] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," in *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 41–50, New Orleans, La, USA, February 1999.

[16] K. Chen, Y. Xue, and K. Nahrstedt, "On setting TCP's congestion window limit in mobile ad hoc networks," in *Proceedings of IEEE International Conference on Communications (ICC '03)*, vol. 2, pp. 1080–1084, Anchorage, Alaska, USA, May 2003.

[17] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP performance," *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 209–221, 2005.

Qi He received his M.A.Sc. degree in electrical and computer engineering from the University of Waterloo in 2005. He is currently with Research In Motion (RIM), where he is a Protocol Specialist. His research interests include MANET and sensor networks, WiMAX, and WiMesh.

Lin Cai received her M.A.Sc. and Ph.D. degrees (with outstanding achievement in Graduate Studies Award) in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since July 2005, she has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Victoria, Victoria, Canada. Her research interests span several areas in wireless communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic over wireless mobile, and ad hoc and sensor networks. She serves as the Associate Editor for EURASIP Journal on Wireless Communications and Networking, and International Journal of Sensor Networks (IJSNet).



Xuemin (Sherman) Shen has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, since October 1993, where he is a Professor and the Associate Chair for Graduate Studies. His research focuses on mobility and resource management in interconnected wireless/wireline networks, UWB wireless communications systems, wireless security, and ad hoc and sensor networks. He is a coauthor of two books, and has published more than 200 papers and book chapters in wireless communications and networks, control, and filtering. He was Technical Cochair for the IEEE Globecom '03, ISPAN '04, QShine '05, IEEE Broadnets '05, and WirelessCom '05, and is the Special Track Chair of the 2005 IFIP Networking Conference. He serves as Associate Editor for IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, Computer Networks, ACM/Wireless Networks, Wireless Communications and Mobile Computing (Wiley), and International Journal of Computer and Applications. He has also served as Guest Editor for IEEE JSAC, IEEE Wireless Communications, and IEEE Communications Magazine. He received the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, for demonstrated excellence of scientific and academic contributions, and the Distinguished Performance Award in 2002 from the Faculty of Engineering, University of Waterloo, for outstanding contributions in teaching, scholarship, and service.

Pinhan Ho received his B.S. and M.S. degree from the Electrical and Computer Engineering department in the National Taiwan University in 1993 and 1995. He started his Ph.D. study in the year 2000 at Queen's University, Kingston, Canada, focusing on optical communications systems, survivable networking, and QoS routing problems. He finished his Ph.D. in 2002, and joined the Electrical and Computer Engineering Department at the University of Waterloo, Waterloo, Canada, as an Assistant Professor at the same year. He is the first author of more than 40 refereed technical papers and book chapters, and the coauthor of a book on optical networking and survivability. He is the recipient of Early Researcher Award (Premier Research Excellence Award) in 2005, the Best Paper Award in SPECTS '02 and ICC '05 Optical Networking Symposium, and the Outstanding Paper Award in HPSR '02.