Courtesy Piggybacking: Supporting Differentiated Services in Multihop Mobile Ad Hoc Networks

Wei Liu, *Student Member*, *IEEE*, Xiang Chen, *Student Member*, *IEEE*, Yuguang Fang, *Senior Member*, *IEEE*, and John M. Shea, *Member*, *IEEE*

Abstract—Due to the salient characteristics such as the time-varying and error-prone wireless links, the dynamic and limited bandwidth, the time-varying traffic pattern and user locations, and the energy constraints, it is a challenging task to efficiently support heterogeneous traffic with different quality of service (QoS) requirements in multihop mobile ad hoc networks. In the last few years, many channel-dependent mechanisms are proposed to address this issue based on the cross-layer design philosophy. However, a lot of problems remain before more efficient solutions are found. One of the problems is how to alleviate the conflict between throughput and fairness for different prioritized traffic, especially how to avoid the bandwidth starvation problem for low-priority traffic when the high-priority traffic load is very high. In this paper, we propose a novel scheme named Courtesy Piggybacking to address this problem. With the recognition of interlayer coupling, our Courtesy Piggybacking scheme exploits the channel dynamics and stochastic traffic features to alleviate the conflict. The basic idea is to let the high-priority traffic help the low-priority traffic by sharing unused residual bandwidth with courtesy. Another noteworthy feature of the proposed scheme is its implementation simplicity: The scheme is easy to implement and is applicable in networks using either reservation-based or contention-based MAC protocols.

Index Terms—Differentiated services, quality of service, multihop mobile ad hoc networks.

1 Introduction

A multihop mobile ad hoc network (MANET) is a self-configurable, self-organizing multihop mobile wireless network with no fixed infrastructure. Each node not only sends/receives packets to/from adjacent nodes, but also acts as a router and forwards packets for other nodes. Features such as rapid deployment and self-organization make ad hoc networks very attractive in military and civil applications for which fixed infrastructures are unavailable or unreliable, yet fast network establishment and constant reconfiguration are required. Such applications include disaster rescue after an earthquake and collaborative computing with laptops in a classroom. Though the driving forces of developing ad hoc networks are strong and the revenue from such deployment may be promising, the market for such networks have not been developed yet. This may be attributed to the many open problems that still need to be resolved before the expected services with desired quality can be provided.

The system dynamics [1] of multihop mobile ad hoc networks, such as time-varying and error-prone wireless links, dynamic and limited bandwidth, time-varying traffic pattern and user location, and energy constraints, pose new challenges that do not exist in wired networks. Many solutions for wired networks may not be feasible in the wireless counterpart if we do not modify them for the wireless environments. To conquer these challenges, in recent years, many researchers advocate a cross-layer design philosophy to develop protocols and applications for MANETs. This is a departure from the traditional layered design for the Internet. Though the cross-layer design philosophy might not be an optimal solution, it does provide us new network implementations that may better support the amalgamation of user services and QoS requirements [37].

Many researchers believe that scheduling, adaptivity, and diversity are the most important design issues in the context of the cross-layer design [1]. Scheduling can help shape the system dynamics [2], [3]; for example, scheduling for data prioritization can be used to support differentiated services. Adaptivity can compensate for or exploit these dynamics using adaptive modulation techniques [4] and adaptive error correction coding [6], [7] to improve the throughput. Diversity techniques can provide robustness to the unknown dynamics. For example, some rerouting mechanisms or alternative routing mechanisms can be designed to combat the link breakage. In short, the cross-layer design principle attempts to make use of the interlayer coupling to develop more efficient schemes to handle heterogeneous traffic over wireless links.

To efficiently handle heterogeneous traffic over wireless links, we need to address two problems. The first is to handle reliable mobile communications in MANETs. This problem has been extensively studied in recent years, and many proposed routing protocols such as DSDV [8], DSR [9], and AODV [10], and medium access control mechanisms such as MACAW [11], FAMA [12], and IEEE 802.11 [13], aim to achieve efficient reliable communications. The other problem is to provide QoS provisioning for heterogeneous traffic with different quality-of-service (QoS) requirements in terms of BER, throughput, and delay. Since the channel bandwidth in wireless environments is limited,

• *The authors are with the Wireless Information Networking Group (WING), the Department of Electrical and Computer Engineering, University of Florida, PO Box 116130, Gainesville, Florida 32611-6130. E-mail: liuw@ufl.edu, xchen@ecel.ufl.edu, {fang, jshea}@ece.ufl.edu.*

Manuscript received 30 Apr. 2004; revised 8 July 2004; accepted 9 July 2004. For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMCSI-0151-0404.

one strategy to support QoS is to set up some kind of priority scheme or service differentiation mechanism [14], [15], under which delay-sensitive traffic has higher priority to access the channel over less time-critical traffic.

In the current literature, many scheduling mechanisms for wireless networks are proposed for this purpose, though most of them are not directly designed for MANETs. In general, these scheduling mechanisms all attempt to combat the channel impairments and to support heterogeneous traffic with the following goals: providing high wireless channel utilization, long-term fairness, bandwidth guarantees and delay bounds for flows with error-free links, or links with sporadic errors [16]. However, these algorithms may not be practical to be implemented in MANETs. Actually, it is hard, if not impossible, to achieve those goals simultaneously because of their conflicting nature. For example, there is a tradeoff between the throughput and fairness[1] or so-called interclass effects [17] among traffic with different priorities. Without any precautionary measures, this conflict may lead to bandwidth starvation for low-priority traffic when the high-priority traffic load is high. Meanwhile, most of these scheduling mechanisms are suitable for the reservation-based MAC protocols, especially for those designed for cell-structured wireless networks. In networks with contention-based MAC protocols such as IEEE 802.11 [13], the reservation-based scheduling mechanisms may not be applicable because it is not easy for a node to reserve resource in a contention manner.

In this paper, we attempt to avoid the conventional scheduling approach, and propose a novel scheme called Courtesy Piggybacking (CP) to alleviate the conflict between throughput and fairness. The basic idea of CP is to let the high-priority traffic help the low-priority traffic by sharing unused residual bandwidth with courtesy. Our scheme closely follows the cross-layer design principle and exploits the system dynamics as much as possible, i.e., we effectively employ the dynamic channel conditions and the resulting dynamic bandwidth, and the dynamic characteristics of the heterogeneous traffic. Note that not only is our scheme suitable for multihop mobile ad hoc networks with underlying contention-based MAC protocols, but also it is applicable to those with reservation-based or hybrid MAC protocols. Meanwhile, our scheme is shown to be easily implemented.

The rest of the paper is organized as follows: In Section 2, we show the motivation of our proposed scheme. In Section 3, we discuss the relationship between the SNR and the optimal packet length, and come up with a Finite State Markov Chain channel model based on the packet length. Our Courtesy Piggybacking scheme is described in Section 4. We present some preliminary analytical result in Section 5 and evaluate our scheme with extensive simulation in Section 6. We discuss some related work in Section 7. Finally, we conclude the paper in Section 8.

2 Motivation

Consider the scenario depicted in Fig. 1. In a mountain area, the only way from **A**nchorage to **W**hittier (the access to see the spectacular glacier) is to pass a tunnel near **P**ortage running through the **C**hugach Mountain Range (i.e., the longest tunnel in North America—the *Whittier Tunnel* in Alaska). The situation is the same from **S**eward to **W**hittier. People have several choices to pass the tunnel: by train (high priority), by car, by bicycle, or on foot (low priority). Only one direction traffic is allowed during one period of time. To pass the tunnel, when the train approaches the tunnel, all other traffic stops and waits until the train passes the tunnel. Often, there is a long traffic line waiting to pass the tunnel, especially for the direction from **W** to **P** when traffic load is high, e.g., during rush hour in the afternoon. In order to quickly pass the tunnel, a better approach for other transportation users would be to check if there is any free space left in the train. If there is, these users could ask for permission to ride at a certain cost and according to some rules, for example, how much free space in terms of basic units is left and what kind of traffic (priority) the train can accommodate. After passing through the tunnel, the piggybacked traffic can get off the train at **P** and continue on its own way. Of course, in the real situations, when passengers by car, by bicycle, or on foot pass through the narrow and dark tunnel in a sequential manner, the traffic usually moves very slowly for the sake of safety. Thus, it is advisable for cars that have free space to piggyback those passengers by bicycle or on foot according to some rules to benefit all the traffic.

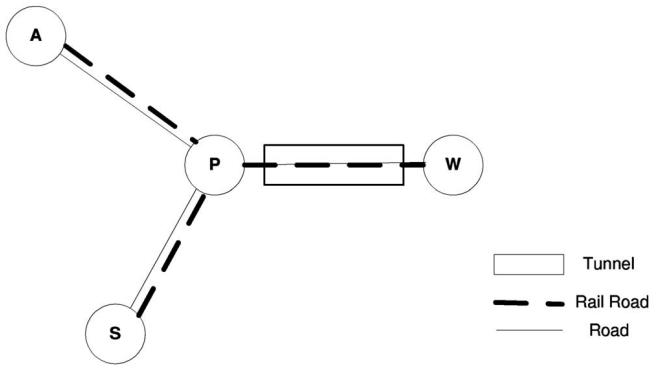


Fig. 1. The Whittier Tunnel scenario.

We can think of these rules as being concerned with the HOW MANY-WHO problem, i.e., how much free space is available and who can enjoy such free space? If we only consider the free space FS in the train as a function of time, then we could consider the following scenario as an example: one person would occupy 1 basic space unit, a bike two units, and a car six units. If we have some predefined objective to meet, then we can design different piggybacking rules to solve the HOW MANY-WHO problem. For example, suppose our objective is to maximize the revenue of the train. With different piggybacking costs, for a given FS, we can achieve the optimal allocation scheme for the free space among different traffic: cars, bicycles, and pedestrians.

The above scenario is very similar to multihop mobile ad hoc networks supporting differentiated services. The piggybacking strategy described above motivates us to develop a more efficient way to alleviate the conflict between throughput and fairness for different prioritized services. First of all, we need to identify the "free space" in a MANET. Fortunately, we do have two sources that can provide us with such free space. The first one comes from the time-varying channel conditions. In recent studies such as [4], [5], the MAC and PHY layers adapt to the channel state by using adaptive transmission schemes to provide

1. We only consider the fairness problem between different classes of traffic, e.g., each class of service should be allocated some bandwidth rather than being completely starved, while the fairness problem between different nodes, e.g., each node should have fair opportunity to access the channel in the short or long term, is out of the scope of this paper.

higher data rates when the channel is good. With a higher data rate, the transmission time for MAC protocol data unit (MPDU) can be shortened, leading to some potential idle time if the transmitting node does not have further data to transmit. If the IEEE 802.11 MAC is used, the NAV (Network Allocation Vector) setting may prevent other nodes from using the medium, even though it is idle (the rule of virtual collision avoidance). This idle period will be the "free space" and should be more effectively used. The second source comes from the traffic characteristics. When we look into the traffic patterns and the stochastic traffic behavior, sometimes the high priority traffic may not have enough data during the reserved slots in a reservation-based system or their transmission period in a contention-based system (e.g., a network with IEEE 802.11) to fully utilize the channel capacity. For example, consider a network with reservation-based MAC protocols. In addition to the "free space" provided by the channel dynamics, when the packets from one high-priority flow are not enough to fill the reserved slots, e.g., during silent periods for voice connections, some "free space" can be harvested to piggyback some bits from the queue(s) with low priorities.

When such free space is available, the next problem would be how to make use of it to fulfill certain objectives such as fair allocation of bandwidth. While one would think that it should be used to better support high-priority traffic in the first place, we argue that it may not necessarily be the case. Rather, the piggybacking rules should be properly designed in light of specific requirements of various applications. If some delay-sensitive applications like voice or video-telephony require that their packets get through the network as quickly as possible, then the free space should be used to meet such needs. On the other hand, if high-priority traffic does not need more resource than needed, a piggybacking rule favoring low priority may be more reasonable. For streaming multimedia applications, as an example, when the QoS requirement of one stream with high priority has already been met, there is no need to piggyback packets belonging to this stream ahead of the scheduled time; instead, piggybacking packets from other low-priority streams may be more beneficial.

In the following sections, we will elaborate more on why free space exists and how piggybacking can be used to achieve our goal—alleviating the conflict between throughput and fairness for different prioritized services.

3 Packet-Length-Based Channel Model

In the current literature, the time-varying channel is commonly modeled as the well-known Gilbert-Elliott two-state Markov channel model (Fig. 2). Each state in the two-state Markov chain model represents a binary symmetric channel (BSC). In "Good" state, the BSC has low crossover probability, P_g, and int the "Bad" state, the BSC has high crossover probability, P_b. The transition probability matrix can be given as:

$$\Re = \begin{bmatrix} P_{GG} & P_{GB} \\ P_{BG} & P_{BB} \end{bmatrix}.$$

Given the transition probability, it is easy to determine that the steady state probabilities are

$$\pi = \left[\frac{P_{BG}}{P_{BG}+P_{GB}} \quad \frac{P_{GB}}{P_{BG}+P_{GB}} \right].$$

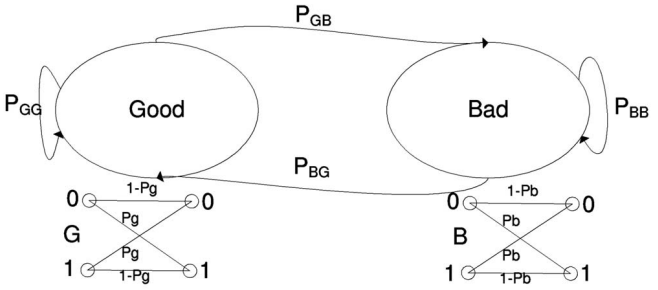


Fig. 2. The Gilbert-Elliott channel model.

We notice that if P_g and P_b are set to 0 and 1, respectively, i.e., a packet succeeds with probability 1 in the "Good" state and is lost with probability 1 in the "Bad" state, the two-state model is reduced to the simplified Gilbert model.

When the channel quality varies dramatically, it is not accurate enough to model the channel as a two-state Gilbert-Elliott model. In this case, a finite-state Markov channel (FSMC) [30] can be used. By using the received signal-to-noise-ratio (SNR) as the only side information, the FSMC provides a mathematically tractable model for time-varying channel. Let γ denote the received SNR that is proportional to the square of the signal envelop. Then, for a Rayleigh fading channel, the probability density function of γ can be written as

$$f_\gamma = \frac{1}{\overline{\gamma}} e^{-\frac{\gamma}{\overline{\gamma}}}, \tag{1}$$

where $\overline{\gamma}$ is the mean of γ (actually, it is an exponential distribution with mean $\overline{\gamma}$). In order to build the finite state Markov chain model, we assume the received SNR remains at a certain level for the duration of a symbol, and we partition the range of the received SNR into a finite number of intervals. Let $0 = \gamma_0 < \gamma_1 < \gamma_2 < \ldots < \gamma_{K-1} < \gamma_K = \infty$ be the thresholds. For each interval, we associate it with a state $S_k, k = 0, 1, 2, 3, \ldots, K-1$. The channel is in the state S_k if γ is in the interval $[\gamma_k, \gamma_{k+1}]$. We know that there is a crossover probability p for a given SNR γ. When BPSK is used, this probability can be written as a function of γ:

$$p(\gamma) = 1 - \Phi(\sqrt{2\gamma}), \Phi(\gamma) = \int_{-\infty}^{\gamma} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt. \tag{2}$$

According to [19], for a given crossover probability p, the optimal packet length, which is a function of p, can be written as

$$PL = \frac{-h\ln(1-p) - \sqrt{-4h\ln(1-p) + h^2\ln(1-p^2)}}{2\ln(1-p)}, \tag{3}$$

where h is the number of overhead bits per packet. Fig. 3 shows the relationship between the received SNR and the optimal packet length. For a given state S_k, the average optimal packet length PL_k for this state can be derived by using (1), (2), and (3) to be

$$\frac{\int_{\gamma_k}^{\gamma_{k+1}} \frac{1}{\overline{\gamma}} e^{\frac{\gamma}{\overline{\gamma}}} \frac{-h\ln(\Phi(\sqrt{2\gamma})) - \sqrt{-4h\ln(\Phi(\sqrt{2\gamma})) + h^2\ln(1-(1-\Phi(\sqrt{2\gamma}))^2)}}{2\ln(\Phi(\sqrt{2\gamma}))} d\gamma}{\int_{\gamma_k}^{\gamma_{k+1}} \frac{1}{\overline{\gamma}} e^{\frac{-\gamma}{\overline{\gamma}}} d\gamma}. \tag{4}$$

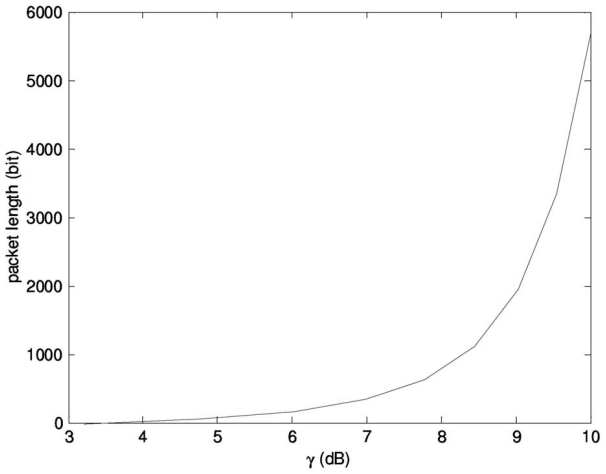


Fig. 3. The optimal packet length (PL) versus SNR (γ), h=128.

Based on the above analysis, we present our packet-length-based FSMC model in Fig. 4. We represent each state as the average packet length PL_k, which is the packet size for a transmission in state k. The transition probabilities between different states are denoted as t_{ij}. Further, we can derive the state steady probability for each state as

$$\pi_k = \int_{\gamma_k}^{\gamma_{k+1}} \frac{1}{\overline{\gamma}} e^{\frac{-\gamma}{\overline{\gamma}}} d\gamma = e^{-\frac{\gamma_k}{\overline{\gamma}}} - e^{-\frac{\gamma_{k+1}}{\overline{\gamma}}}, k = 0, 2, \ldots, K-1. \quad (5)$$

In practice, we may use different modulation schemes (not necessarily BPSK) in different channel states. Moreover, by properly partitioning the range of the received SNR, we may obtain the multiplicative relationship between the average optimal packet lengths.

4 Courtesy Piggybacking

In this section, we present our Courtesy Piggybacking scheme to alleviate the conflict between throughput and fairness and to combat the starvation problem for differentiated services.

4.1 System Assumptions

We consider an ad hoc network consisting of n mobile nodes uniformly distributed in some area. Nodes can communicate with each other directly if they can hear each other or through other relay nodes in a single broadcast channel. They employ some contention-based MAC protocols, such as IEEE 802.11, to support their communications. Each node can generate services with N different priorities destined to other mobile node(s). A node's mobility follows the random waypoint model [36], [38]. At first, a node stays at a position for duration of *pause_time*. After that period, the node chooses a new random position and moves toward that position at a random speed uniformly distributed in the range from *min_speed* to *max_speed*. After reaching the new position, the node will stay there for another *pause_time*. This process will continue for each node until the end of the simulation.

We assume some service differentiation mechanism is employed at the network layer. All the heterogeneous traffic is prioritized at its originating source node. When a packet is handed down from the network layer, it will be kept in the Tx queue corresponding to its priority and wait for its turn to be transmitted at the MAC layer.

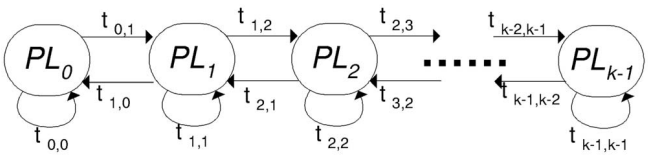


Fig. 4. Packet-length-based Finite-State Markov Channel Model.

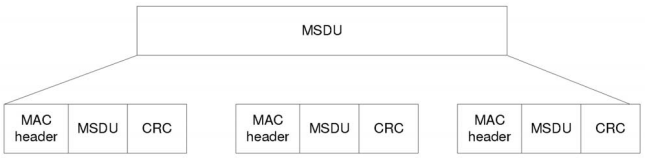


Fig. 5. A fragmentation example.

From the previous section, we know that the packet length is related to the received SNR. The greater the SNR is, the greater the packet length is. In the IEEE 802.11 MAC protocol [13], this packet length may be called as frame length, which equals the fragmentation threshold plus the length of the MAC header and the length of CRC. In the IEEE 802.11 standard, the MAC layer takes a MSDU from the Tx queue and adds MAC header and a CRC to each MSDU to generate a MPDU. In order to reduce the probability of transmission errors, the IEEE 802.11 limits the size of the body of a MPDU to be less than a fixed fragmentation threshold (FT), or it will break the long MSDU into multiple fragments, each of which will be no longer than the FT. In Fig. 5, we show a case where a long MSDU is partitioned into three small MSDUs in the IEEE 802.11. Since the length of the MAC overhead may be kept unchanged, according to the analysis in the previous section, different channel states have different frame lengths, we can say that different channel states have different fragment thresholds (FTs). The greater the received SNR is, the greater the fragment threshold (FT) is. Hereafter, we associate FT_k with each state S_k of the FSMC model as depicted in Fig. 4. In order to improve the channel utilization, we assume that the MAC protocol can adaptively adjust the fragmentation threshold and the transmission rate according to the channel state. To accurately determine the channel state when some packets need to be transmitted, we further assume that we have some channel estimators or predictors, which can provide the accurate channel information for the proper MAC layer fragmentation.

4.2 The Courtesy Piggybacking Scheme

In practice, the size of a packet generated by an application may be fixed or may vary from a minimum allowed size to a maximum value PK_{max}. We argue that the PK_{max} should be properly chosen to reduce the overall overhead. Suppose we want to transmit c Mbits traffic. Packets are generated according to the PK_{max}. We assume that each packet can be correctly received without any retransmission.[2] Then, the total overhead should be the sum of the overheads O_{ip} at the IP layer (e.g., 20 bytes for IPv4), O_{mac} at the MAC layer (e.g., 34 bytes for the IEEE 802.11) and O_{phy} at the PHY layer

2. For simplicity, we only consider the case without any retransmission and view the resulting overhead as a lower bound. Apparently, the overhead with retransmissions is larger than this lower bound.

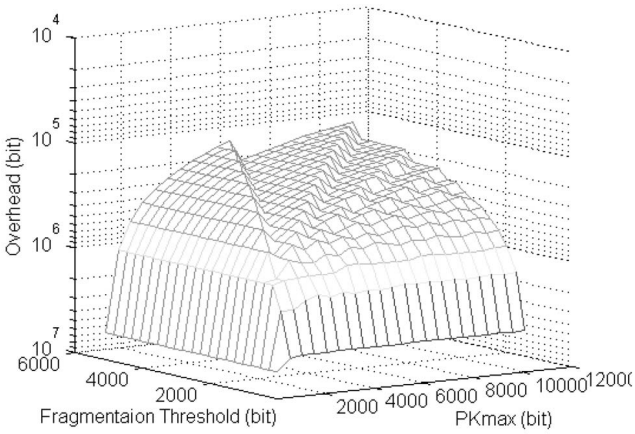


Fig. 6. The total overhead with PK_{max} and FT.

(e.g., 16 bytes). Thus, the total overhead to transmit the c Mbits traffic can be written as

$$\left\lceil \frac{c}{PK_{\max}} \right\rceil \times \left(O_{ip} + \left\lceil \frac{PK_{\max}}{FT} \right\rceil \times \left(O_{mac} + O_{phy} \right) \right),$$

where $\lceil \bullet \rceil$ is the function to round the element to the nearest integer greater than the element. We show the relationship of overhead versus PK_{max} and FT when $c = 1$ in Fig. 6.

From Fig. 6, we observe that PK_{max} should be reasonably chosen when multiple fragmentation thresholds are used. It cannot be too small, as it may cause too much overhead; neither can it be too large, as it may generate too many fragments when the FT is small, which may further degrade the overall throughput. For example, a large packet may be partitioned into many small fragments. Each fragment is augmented with an individual header, sent as an independent transmission, and acknowledged individually. Hence, a large packet will cause lots of DATA/ACK exchanges and result in suboptimal performance. Thus, there must exist an optimal value of $PK^*_{\max}$ such that the overall overhead associated with the successful transmission of a message is minimized. Assume we obtain $PK^*_{\max}$, which may not be equal to any of $FT_k (k = 0, 1, 2, \ldots, K-1)$; it is thus advisable to approximate $PK^*_{\max}$ with the closest fragmentation threshold corresponding to a certain channel state, say S_m. Therefore, we set PK_{max} to FT_m.

Next, we want to show where the "free space" comes from. When a packet with length strictly less than the PK_{max} is transmitted in the channel state with FT less than FT_m, the packet may be fragmented, and there is no "free space" available for all fragments possibly except the last one. However, due to the time-varying nature of the channel, when the packet is transmitted in the channel state with FT greater than FT_m, one packet does not have enough bits to utilize the full capacity the channel provides in one transmission. We argue that we could take advantage of the "free space" to pack more bits as the channel allows. As a matter of fact, we have shown, in our recent studies, that the fragmentation threshold can be up to 10K bits when the SNR is close to 20 dB and 64 QAM modulation scheme is used with targeted FER 8% (Frame Error Rate) [5]. On the other hand, we observe that in contention-based MAC protocols, it may take a long time for a node to seize the channel, and the node that has seized the channel should treasure every transmission opportunity to transmit as many bits as possible, especially when the channel condition is good. From now on, we call state S_k the free-space-effective state when k is greater than m, otherwise the non-free-space-effective state, even though such a state may still have the possibility to pack more bits when the traffic dynamic is taken into account.

Now, we describe how the Courtesy Piggybacking scheme makes use of the free space. When a mobile node seizes the channel, it will first check the channel state and determine if it is in a free-space-effective state and if it is capable of piggybacking more packets in one transmission. If it is not in a free-space-effective state, only one packet (MSDU) with highest priority from the queues will be served, as the current MAC protocol does. If the channel is in a free-space-effective state, the node can transmit in one transmission as many bits as channel allows and, thus, can piggyback more packets (MSDUs) from the queue(s), which may have different priorities but the same next hop in the routing table. Since the Courtesy Piggybacking scheme follows the cross-layer design principle so that the MAC layer has the access to the routing information, it is possible for the MAC layer to obtain such packets from the Tx queues.

After identifying the existence of the free space, we now discuss the piggybacking rules that guide the MAC layer to assemble enough and proper bits from the Tx queues (the HOW MANY-WHO problem) and piggyback them to the next hop to alleviate the conflict we intend to address. Since the channel state determines "HOW MANY" MSDUs the node can pack and transmit in one transmission, the fundamental issue of the rules should specify "who" plays the role of "train" that offers the piggybacking service to others, and "who" can enjoy such piggybacking service. Without any scheduling mechanism, the role of "train" is always taken by the MSDU located at the head of a nonempty queue with the highest priority currently. Thus, the piggybacking rules should primarily address "who" has the privilege to enjoy such "free" piggybacking service. As a guideline, the basic idea for such piggybacking rules is that, under different channel states, the node assembles multiple MSDUs that may have different priorities but share the same next hop in the routing table, to form an MPDU whose length is channel dependent. In this way, we can achieve some extent of fairness between different prioritized services. When the channel is not in a free-space-effective state, only the highest priority service in the Tx queues is supported, and the packets are fragmented if needed and are treated as usual. When the channel changes to a free-space-effective state, according to the rules we define, our Courtesy Piggybacking scheme can pack other services, possible with lower priorities, to share the residual bandwidth with the high-priority traffic. One such rule is to give preference to high-priority services. It always, if possible, packs the high-priority services destined to the same next hop in queue(s). Only when there are no more bits from the high-priority traffic fitting into the free-space[3] will the bits from the lower-priority queue(s) be considered for piggybacking. Other rules may not prefer the high-priority service; for example, a high-priority service may trade-off its own performance for fairer channel utilization

3. When we say "no more bits in one queue fitting into the free-space," it means either no packet is left in the queue or packets in the queue do not share the same next hop with the MSDU who offers the piggybacking service.

by its **courtesy**—piggybacking the low-priority service. One such rule is to always piggyback the MSDUs from the longest Tx queue. With such piggybacking rules, the traffic dynamics (different packet arrival time and destinations) and channel dynamics are jointly utilized to strike a good balance between throughput and fairness. Note that the piggybacking rules are not necessarily defined a priori; they could be designed to adapt to both channel and traffic uncertainty in the runtime.

Intuitively, the Courtesy Piggybacking scheme can improve the performance of the low-priority traffic since some low-priority packets may be packed with high-priority packets and be delivered to the next hop for free; thus, it can statistically reduce the time taken to contend for accessing the channel for the low-priority services. This benefit will be more pronounced in mobile ad hoc networks using service differentiation based MAC protocols [14], [18] where the MAC protocols sacrifice quality of the low-priority service to support high-priority service through either time spacing (differentiation of Interframe Space (IFS)) or backoff parameters [13]. On the other hand, the reduction of contention from low-priority services can, in turn, benefit the high-priority services: One node's courtesy piggybacking of low-priority services may help its neighbors to transmit high-priority traffic because less low-priority traffic will reduce the contention the high-priority traffic may encounter. One may wonder why we do not simply release the channel so that other low-priority traffic can use the channel, i.e., the so-called complete sharing scheme. The problem is that the time, for which the residual resource is available, is too short to be given to other services due to the overhead associated with successfully seizing the channel. Besides, some MAC protocols such as the IEEE 802.11 family forbid others to use the channel during the time period specified by the Network Allocation Vector (NAV). Even if the NAVs are reset, the contention process may take too long to render the harvested resource from the rate adaptation useless. Thus, the courtesy piggybacking by high-priority traffic flows makes more sense. In short, our Courtesy Piggybacking is able to achieve better channel utilization and further improves the fairness between different prioritized traffic by the following two means: lowering the contention of the network and decreasing the overhead required for a transmission. And, as we show in simulation later, our scheme significantly improves the performance of low-priority traffic, while improves or at least keeps unchanged the performance of high-priority traffic with appropriate piggybacking rules.

To illustrate the Courtesy Piggybacking scheme, we demonstrate the operation of the scheme in Fig. 7. First, prioritized packets, called MSDUs, arrive from the network layer as b-MSDUs (basic MSDUs, the basic unit) whose lengths agree with the FT_m. We assume the maximum packet length PK_{max} is strictly enforced at the upper layer; if not, an oversized MSDU will be further broken down into several b-MSDUs and the resulting b-MSDUs will inherit the IP header of the original MSDU. The b-MSDUs are kept in the queues corresponding to their priorities. The dequeue controller operates according to the predefined piggybacking rule, dequeues one or more b-MSDUs with the same next hop, and forms a MPDU satisfying the FT corresponding to the channel state. In order to reduce the overhead and the work to break a MSDU at the transmitter and to assembly the MSDU at the receiver, it is advisable to limit the packet length at the network layer to be no longer than

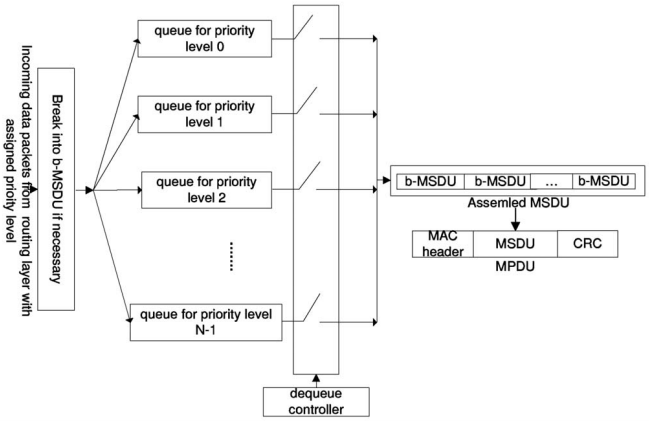


Fig. 7. Illustration of piggybacking scheme.

FT_m. In order to avoid further fragmentation of a b-MSDU to fit the free space and assembly the b-MSDU, it is advisable to maintain the multiplicative relationship between the fragmentation threshold (FT) of the free-space-effective state and FT_m, i.e., the frame length for state k satisfies $FT_k = g_k \times FT_m$, where g_k is a positive integer. This can be achieved by properly partitioning the range of received SNR and adopting channel-dependent modulation schemes. To reduce the transmission time for a long frame, rate adaptive transmission schemes may be used, so that the time for transmitting a frame does not vary too much. To avoid making too many modifications to the MAC layer, we prefer packing the b-MSDUs with the same next hop in the routing table. To facilitate a receiver in unpacking the bound packets, an unused bit [33] in the IP header of each b-MSDU is set to 1 at the transmitter to indicate that one bound b-MSDU is followed this b-MSDU, and the corresponding bit in the last b-MSDU is set to 0. At the receiver, the only thing it needs to do is to acknowledge the received long frame and unpack the packed packets one by one according to the value of the unused bit.

4.3 Discussion

4.3.1 Some Properties of the CP

If we examine the destination of the bits in a single piggybacked transmission, though these bits share the same next hop, we find out these bits may be destined to the next hop or other nodes. Fig. 8 shows three scenarios for the Courtesy Piggybacking. Consider that a mobile node A sends some bits (consisting of two b-MSDUs) to the next hop B which have three neighbors, including this mobile node A. Suppose that packet 2 is piggybacked by packet 1, both packets should have the same next hop B. After the packets 1 and 2 arrive at B, there are three cases at node B to process these two packets if we do not distinguish the difference between packet 1 and packet 2. Case 1 shows that packet 1 and packet 2 may be destined to different nodes and have different next hops at node B. Case 2 shows the case when both packets have the same destination B. Case 3 shows that one packet is destined to B while another one is destined to a node other than B.

Since the probability that a packet is piggybacked largely depends on channel conditions and traffic pattern, the piggybacking may induce some delay jitter. For instance, due to our restriction that only packets sharing the same

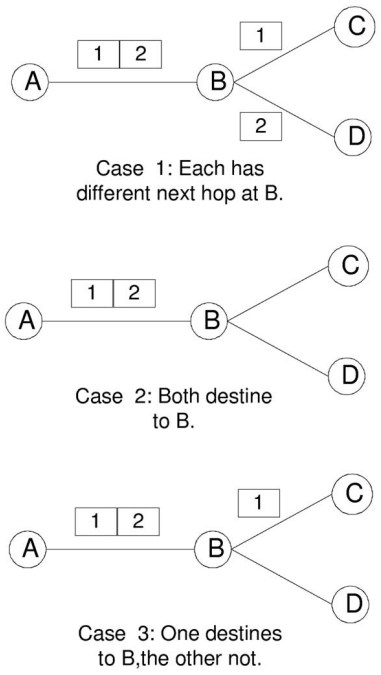


Fig. 8. Three piggybacking cases.

next hop can be piggybacked together, one packet that arrives at one node later may leave the node earlier than some earlier arriving packets.

4.3.2 The Significance of Piggybacking Rules

The piggybacking rules may play an important role in allocating the bandwidth among the different prioritized traffic. Here, we want to discuss the design of a piggybacking rule based on a special case. When we have plenty of different priority packets in the Tx queues waiting to be served, the design of the piggybacking rule can be viewed as an allocation problem. As discussed above, by properly partitioning the range of received SNR, the fragmentation threshold of the free-space-effective state k satisfies $FT_k = g_k \times FT_m$, where g_k is a positive integer. For other non-free-space-effective states, let $g_k = 1$. Suppose we have a total of K different channel states, and N different priority levels. Let α_{kj} denote the number of b-MSDUs of j priority level to be packed when the channel is in state k. We should point out that in the non-free-space-effective state, only the highest priority packet is served when there is plenty of traffic in the waiting queue, thus $\alpha_{k0} = 1$ and $\alpha_{kj} = 0$ for $0 \le k \le m$ and $1 \le j \le N-1$. If we neglect the MAC layer overhead, then the design problem can be reduced to choose a_{kj} such that

$$\begin{cases} 0 \le \alpha_{kj} \\ \sum_j \alpha_{kj} \le g_k \quad 0 \le k \le K-1, 0 \le j \le N-1. \end{cases} \tag{6}$$

Thus, an upper bound for the expected value of throughput of priority level j at one node is given by $\sum_k R_k p_k \alpha_{kj}, 0 \le j \le N-1$, where p_k is the probability that the channel is in state k, and R_k is the transmission rate in state k.

4.3.3 Some Measures May Improve the CP

To avoid fragmentation of the b-MSDUs in a free-space-effective state, in our piggyback scheme, we should maintain the multiplicative relationship between the fragmentation threshold (FT) in the free-space-effective state and the FT_m. Actually, we can relax this requirement in the high traffic load case by allowing fragmentation of the low-priority services at will to fit into the free space the channel provides because, at very heavy traffic load, the piggybacking rule favoring high-priority services may still lead to bandwidth starvation for low-priority services. By allowing the fragmentation of the b-MSDUs from low-priority traffic, at least some low-priority traffic can be served by piggybacking.

Our Courtesy Piggybacking scheme does not preclude scheduling mechanisms; in fact, scheduling can still be employed at higher layers to enhance the management of the heterogeneous traffic. For example, we can use EDF (Earliest Deadline First) policy to manage each priority queue, such that in a single queue, the packet with early deadline may be put at the head of the queue and get transmitted earlier than those with a later deadline. At the same time, the piggyback scheme can still take effect once the channel is in a free-space-effective state.

In our preliminary implementation of the piggybacking, the b-MSDUs are organized in the queues according to their priorities. When a transmitter wants to pack more bits to the same receiver, an exhaustive search is carried out to find the proper bits in candidate queues according to the piggybacking rules. In addition, some processing time is also needed at the receiver to unpack the packets. Thus, in the first place, we may expect some additional delay caused by courtesy piggybacking; however, as can be seen from our performance evaluation, the incurred delay is negligible and, hence, acceptable compared with the benefit gained from the Courtesy Piggybacking scheme. Furthermore, we argue that with more efficient ways to organize the b-MSDUs and to quickly acquire the proper b-MSDUs for packing, the benefit from the piggybacking scheme should be more visible.

In our proposed Courtesy Piggybacking scheme, only the traffic sharing the same next hop can be packed. One may wonder if it can be extended to the different next hop scenarios. We are cautions to make this move. The main concern comes from the MAC layer. The prevalent MAC protocols including the IEEE 802.11 do not support multiple receivers at the same time. Undoubtedly, it is a arduous task to coordinate the transmission and reception activities between multiple receivers. Especially when multiple receivers are involved in one transmission, the interference area should be much larger than that of one receiver case and the coordination issue gets more complicated. Though a data flushing mechanism for multiple receiver was presented in [32], while without consideration of the enlarged interference area, the benefit may not outweigh the negative impact of their approach.

In our exemplary piggybacking illustration (Fig. 7), all the dequeued b-MSDUs are first assembled as one MSDU and further encapsulated with only one MAC header and CRC. This exemplary piggybacking method imposes no modification on current MAC protocol, e.g, IEEE 802.11, but it lacks flexibility. For example, when the receiver detects something wrong with the received frame, it is not able to inform the transmitter which b-MSDU is damaged, and as a

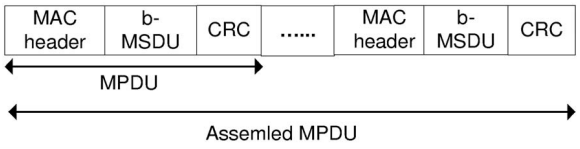


Fig. 9. An alternative piggybacking method.

result, the transmitter has to retransmit the whole frame again. In fact, we can adopt an alternative method as the following (Fig. 9). Different from the above method, each b-MSDU will generate a separate MPDU with its own CRC and MAC header, and the resulting MPDUs will finally be combined as an assembled MPDU and be sent out as one unit in one transmission. At the receiver side, the receiver can check the CRC for each b-MSDU and positively or negatively acknowledge each b-MSDU in a single short message, e.g., ACK. With this alternative method, the transmitter can retransmit only those b-MSDUs failed CRC checking. In fact, these retransmitted b-MSDUs can be further bound with other fresh b-MSDUs from the queues to form another assembled MPDU. Comparing the alterative one with our exemplary one, the alterative method provides some degree of flexibility, but requires some modification on MAC protocol to facilitate the above communications. This modification can be possibly made similar to that in [32]. On the other hand, our exemplary piggybacking method may incur less overhead of MAC header and CRC than the alternative one. In practice, some adaptive piggyacking methods can be designed to accommodate the time-varying channel condition, meanwhile, keep the overhead reasonable.

5 Performance Analysis

In this section, we present the performance analysis in order to theoretically show the effectiveness of our proposed piggybacking scheme in alleviating the conflict between throughput and fairness for different prioritized services.

To simplify the analysis, we consider the piggybacking at one node, and assume all the packets are destined for the same next hop. Again, we assume that there are a total of N different priority levels, $P_0, P_1, \ldots, P_{N-1}$, whereby P_i has higher priority than P_j if $i < j$. The arrivals of each priority P_i service are *Poisson* processes with arrival rate λ_i. The channel is modeled as FSMC as discussed in Section 3: $FT_k = g_k \times FT_m$, where $g_k = 1$ if $k \leq m$;[4] otherwise, g_k is an integer greater than 1 and $g_i > g_j$ if $i > j \geq m$. When the channel is in state $j > m$, the channel adaptation scheme is used such that the time for transmitting FT_j is almost the same as that for FT_m. Moreover, we assume that the packet length is $PK_{max} = FT_m$. Under such assumptions, our Courtesy Piggybacking scheme can be modeled as a multiple-server queue system with nonpreemptive priority, where the service rate is dependent on the channel state. More specifically, the service discipline is the following if we limit the number of servers to two. Server SR_0 operates

4. Given fixed transmitting power and modulation schemes, when the channel is in any state $j < m$, a fragmentation threshold that is smaller than FT_m is required to maintain the same frame error rate. For simplicity, we assume FT_m is also used in those states.

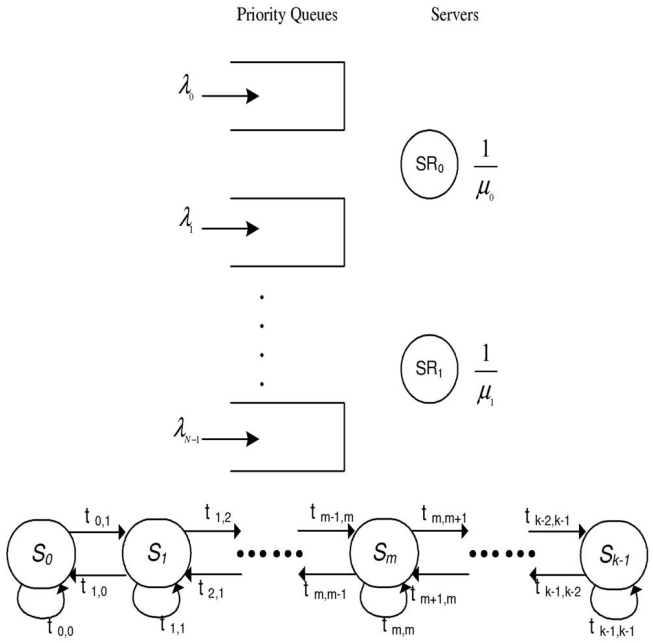


Fig. 10. The queue model for piggybacking.

in all channel states with average service time X. For server SR_1, it operates when the channel is in the state $S_k, k > m$, and does not work in any state $S_k, k \leq m$; the service time for server SR_1 is $\frac{X}{g_k - 1}$. This means that, when server SR_0 serves one nonpiggyback packet, server SR_1 can serve $g_k - 1$ piggybacked packets. The nonworking probability of server SR_1 is $\sum_{k=0}^{m} \pi_k$, where π_k is the steady-state probability of state S_k. Server SR_0 operates with nonpreemptive head-of-line priority service discipline, while server SR_1 operates according to the discipline defined in the piggybacking rules. For example, it may first serve the traffic with the longest waiting queue. Within a priority traffic, service is provided on a first-come-first-served basis. Fig. 10 shows the analytical model.

For the purpose of obtaining tractable analysis, we study the scheme in a simple scenario in which the traffic is of two priorities, and the channel has two states, with $FT_1 = 2 \times FT_0$. Accordingly, the service time for each server is X. As discussed above, server SR_0 works all the time with nonpreemptive service discipline, whereas SR_1 works only when the channel state is in S_1 and does not work otherwise. Then, our piggybacking queue model can be reduced to an M/D/2 nonpreemptive priority queuing system, in which one server works all the time, while the other works with some probability. This system seems to be simple, however, it is hard to quantitatively analyze it. To our knowledge, there is no ready solution to this interesting queue system. Hence, we seek to get some bounds on the performance metrics such as average waiting time (i.e., the average queuing delay) and queue size. We only focus on the average waiting time hereafter as the average waiting time and the queue size lead to each other according to Little's Law.

We first consider the upper bound of the average waiting time for the system. Obviously, if a transmitter does not have any knowledge of the channel status and, hence, does not adopt any rate adaptation based on the channel state, both traffic types, i.e., the high priority traffic and the low

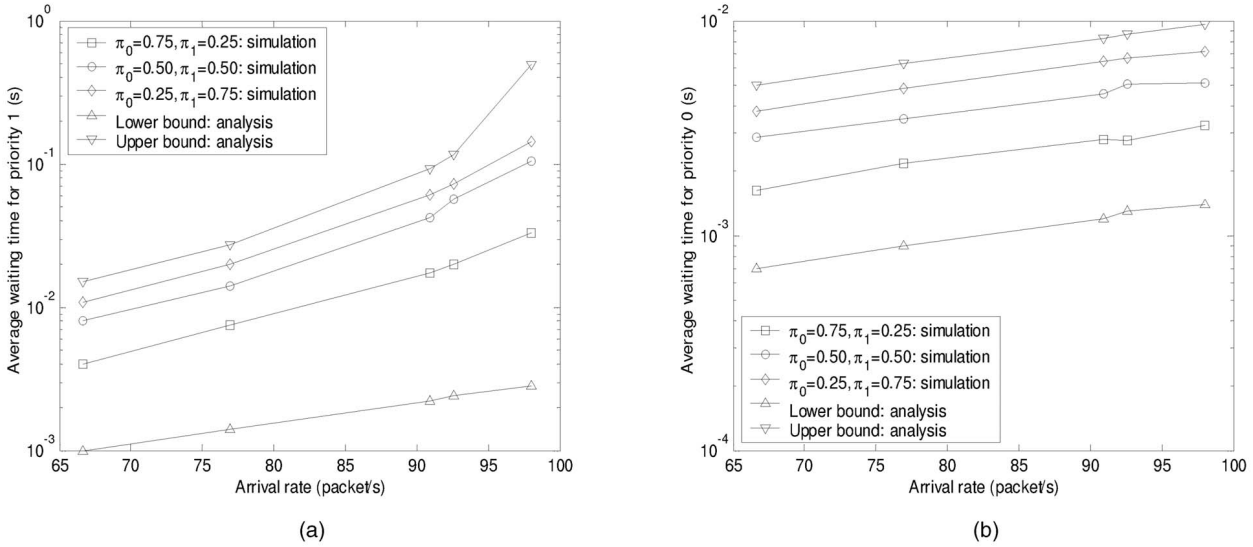


Fig. 11. Average waiting time. (a) Average waiting time of priority 1 service. (b) Average waiting time of priority 0 service.

priority traffic, will suffer from longer delay than they will when piggybacking is used. In this case, only one server, i.e., server SR_0 works. The system thus is an M/D/1 priority queue system [34], and it is easy to obtain the average waiting time for each type of traffic:

$$\begin{cases} W_0 = \frac{(\lambda_0+\lambda_1)X^2}{2(1-\lambda_0 X)} \\ W_1 = \frac{(\lambda_0+\lambda_1)X^2}{2(1-\lambda_0 X)(1-\lambda_0 X-\lambda_1 X)}. \end{cases} \quad (7)$$

Recall that server SR_0 works all the time and server SR_1 works only when channel state is in S_1. If the piggybacking rule is set such that, when SR_1 operates, it always serves the high-priority traffic first, the following case will provide the lower bound for the average waiting time. In this case, the transmitter is aware of the channel state and adopts the channel dependent transmission rate. Moreover, the channel is always in a good state. Therefore, the system becomes a two-server queue system with nonpreemptive priority, where the two servers, server SR_0 and SR_1, work all the time. According to the results in [35], we can obtain the average waiting time for each type of traffic:

$$\begin{cases} W_0 = \frac{4(\lambda_0+\lambda_1)^2X^3}{3(2+(\lambda_0+\lambda_1)X)(2-\lambda_0 X)} \\ W_1 = \frac{4(\lambda_0+\lambda_1)^2X^3}{3(4-(\lambda_0+\lambda_1)^2X^2)(2-\lambda_0 X)}. \end{cases} \quad (8)$$

To validate the upper and lower bounds, we implement the above queue model in OPNET [31] and present the analytical results and simulation results in Fig. 11, where the service time $X = 0.01$s. It can be observed that the average waiting time for the piggybacking scheme in three different channel models as specified in Table 1 is completely bounded by the obtained bounds. Therefore, we know the gain that can be accrued by adopting the piggybacking scheme as opposed to the case without piggybacking.

6 Performance Evaluation

In this section, we implement our proposed piggybacking scheme in OPNET with DSDV and the IEEE 802.11 as underlying routing and MAC protocols, and conduct extensive simulations to evaluate the performance of the piggybacking scheme. The simulation results validate the effectiveness and efficiency in improving the channel utilization and fairness, thus alleviating the conflict between throughput and fairness.

6.1 Simulation Setup

In our simulation study, we assume the physical channel is a slow fading channel with only two states satisfying $FT_1 = 2 \times FT_0$. Three different channel settings are adopted in our studies. The channel statistics are listed in Table 1. While setting 2 represents a "neutral" channel in the sense that the steady state probabilities of both states are identical, channel setting 1 and 3, respectively, represent a relatively "bad" channel and a relatively "good" channel. Without loss of generality, we limit the number of priorities to 2. We simulate an ad hoc network consisting of 50 mobile nodes,[5] whose mobility follows the random waypoint mobility model in a $1500 \times 300m^2$ area. The transmission range of each node is $250m$. Each node generates traffic according to a Poisson process with parameter λ, and the destination for each generated packet is randomly chosen among all other nodes. We assume that the packet length is 1,024 bits that agrees with FT_0 and each packet is a b-MSDU. The generated traffic is further assigned with 1 (low priority) or 0 (high priority) with probability 0.5. All packets are buffered in the queues according to their priorities. In our simulations, when the channel is in the state corresponding to FT_0, we use a basic transmission rate of 1Mbps, while for state corresponding to FT_1 we use 2Mbps, so that the transmission time for one fragment with channel-dependent length in two states does not change too much. Each simulation runs for 300 seconds.

5. To minimize the possible unfairness between different nodes due to unequal channel access, we use such a configuration and treat all the nodes equally, e.g., using the same traffic and mobility pattern.

TABLE 1
Channel Model Statistics

Setting	State i	$Pr(i)$	$t_{1,i}$	$t_{0,i}$
1	0	0.75	0.0075	0.9975
	1	0.25	0.9925	0.0025
2	0	0.5	0.002	0.998
	1	0.5	0.998	0.002
3	0	0.25	0.0025	0.9925
	1	0.75	0.9975	0.0075

We define two piggybacking rules for comparison. Rule 1 favors the high priority traffic. When the channel state is in FT_0, a priority-1 packet can be served only when no priority-0 packet exists in the queue. When the state is in FT_1, a priority-1 packet is piggybacked by priority-0 packets only when no priority-0 packet is left in the queue (of course, the packets should share the same next hop in the routing table). In contrast, rule 2 favors the low priority traffic. In rule 2, when the channel is in FT_0, the stations act as in rule 1. When the state is in FT_1, a priority-1 packet is piggybacked no matter how many priority-0 packets are left in the queue. We study the performance of the network in four scenarios:

- Case 1. The network is unaware of channel states.
- Case 2. The network is aware of channel states and, thus, adopts dynamic transmission rate.
- Case 3. The network employs the Courtesy Piggybacking with rule 1.
- Case 4. The network employs the Courtesy Piggybacking with rule 2.

We choose two metrics to analyze and compare the performance of piggybacking: **End-to-end delay** and **Packet delivery ratio**. End-to-end delay measures the average one-way latency observed between the time instant that the packet is generated at the source and the time instant the packet is received at the destination. This metric should count in all the delays, such as propagation delay, queuing delay, and transmission delay, which the packet has experienced during the whole process. Packet delivery ratio measures the ratio of the total number of packet successfully received by the destinations to the total number of packets generated at the sources. This metric reflects the overall throughput and fairness of each prioritized traffic.

To observe the effect of the channel characteristics, we first disable node mobility and compare the performance of piggybacking under different channel settings and under different traffic loads. Then, we fix the channel setting as setting 2, and enable node mobility according to the random waypoint mobility model described in the Section 4 with $min\_speed = 1m/s$ and $max\_speed = 19m/s$, in order to study the effect of traffic load and mobility on the performance of piggybacking.

6.2 Impact of Channel Characteristics

Fig. 12 illustrates the performance of the network in the four different cases, when three channel settings are specified as in Table 1. Since in Case 1, the network has no knowledge of the channel, the performance is the same no matter what channel settings are used. In all the other three cases, the

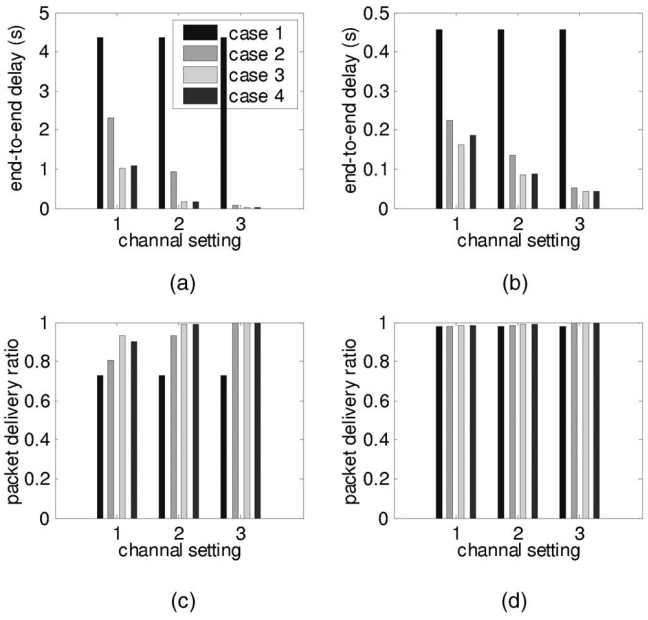


Fig. 12. Simulation results with different channel settings.

performance changes with the channel setting. More specifically, as the channel is getting better, i.e., the channel setting changes from 1 to 2, and to 3, for each traffic priority, the end-to-end delay decreases and the packet delivery ratio increases. This shows that the dynamic channel conditions are valuable resources that should be explored. More improvements in terms of both metrics can be observed when piggybacking is adopted, e.g., Case 3 and Case 4, compared to Case 2. Further, our piggybacking scheme can achieve a better fairness between these two prioritized traffic while still maintaining a higher packet delivery ratio, thus higher throughput. This is because, in addition to taking advantage of channel states, the piggybacking scheme reduces the time to contend for the channel, thereby further improving channel utilization.

6.3 Impact of Traffic Load

The performance of the network is studied under different traffic loads as well, as shown in Fig. 13, in which the average packet interarrival time of 0.3s represents the relatively light traffic load and the average packet interarrival time of 0.25s represents relatively heavy traffic load. From Fig. 13, we can see the interclass effects in the differentiated service system, which becomes more pronounced in the case where the high-priority traffic load is high. The delay for priority 1 (7.89 seconds) is far greater than that for priority 0 (0.30 seconds), and the packet delivery ratio for priority 1 is much smaller than that for priority 0. It can also be observed that Cases 3 and 4, where our piggybacking scheme is employed, have better performance than Cases 1 and 2. Under both heavy traffic load and light traffic load, the piggybacking scheme can greatly reduce the end-to-end delay and improve the packet delivery ratio for both priorities. It is noteworthy that, although channel aware mechanisms (Case 2) can improve both metrics compared to Case 1, our piggybacking scheme provides more benefits than the channel aware mechanism alone cannot achieve for the reasons described above.

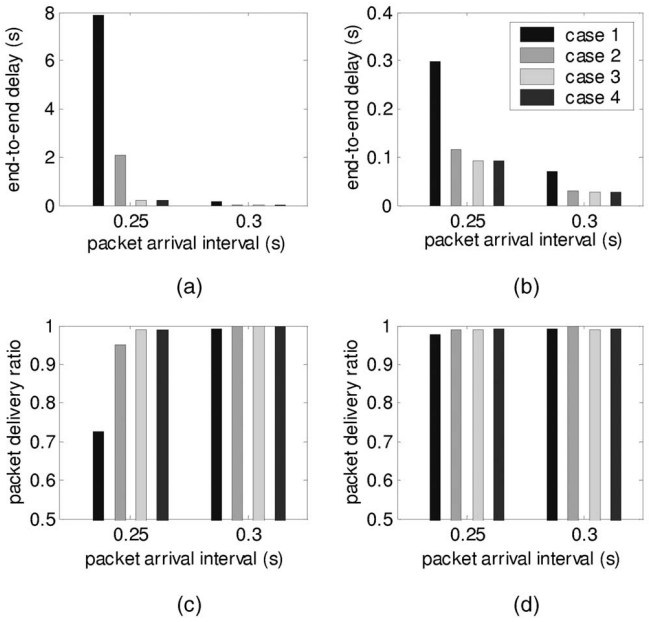


Fig. 13. Simulation results with different packet arrival rates.

6.4 Impact of Node Mobility

Next, we study the impact of mobility on the performance of the proposed Courtesy Piggybacking scheme when the average packet interarrival time is 0.25s. In Fig. 14, the larger the *pause_time*, the lower the mobility. We observe that in all the cases, the performance is sensitive to mobility and degrades as mobility increases. The reason for this is that high mobility may frequently cause route breakages and packet losses, and hence increase delay and decrease packet delivery ratio. In addition, we can clearly observe that Case 2, Case 3, and Case 4 have better performance than Case 1, which is consistent with our observations in previous simulations. In general, these three cases have shorter end-to-end delay and higher packet delivery ratio than Case 1 for both priorities, especially for the low priority traffic, the disadvantageous traffic in the differentiated service system. Since all the cases except Case 1 make use of the channel conditions and rate adaptation, again, we validate that the dynamic channel conditions can be used to improve the channel utilization.

We further compare Cases 3 and 4 as a group with Case 2 to show the effectiveness of our Courtesy Piggybacking scheme. From Fig. 14, we can clearly see that our scheme can further shorten the end-to-end delay and improve the packet delivery ratio for both types of traffic. Moreover, the Courtesy Piggybacking scheme not only improves the performance of the priority-0 traffic, but also significantly improves the performance of the priority-1 traffic. This verifies that our Courtesy Piggybacking scheme is capable of alleviating the conflict between the different prioritized traffic. As discussed in Section 4, all these gains come from the Courtesy Piggybacking scheme. In Case 2, the channel state information is exploited only to some extent, but not fully harvested in the sense that the "free space" cannot completely be utilized. However, our piggybacking scheme can make use of these system dynamics, not only the channel dynamics but also the traffic dynamics, so that the "free space" can be best exploited without any waste.

6.5 Impact of Piggybacking Rules

Finally, we focus on Cases 3 and 4 and study the impact of the piggybacking rules. The piggybacking rule in Case 3 favors the high-priority traffic in the system, priority 0, while the rule in Case 4 favors the low-priority traffic, priority 1. Thus, there is no surprise that in Figs. 14a and 14c, the end-to-end delay for the priority 1 in Case 4 is generally shorter than that in Case 3, and the packet delivery ratio is generally greater than that in Case 3. For the priority-0 traffic, all the measured metrics generally have better performance in Case 3 than those in Case 4. Compared with the performance in Case 3, the Courtesy Piggybacking in Case 4 sacrifices the priority-0 traffic a little bit to piggyback the priority-1 traffic. In Fig. 14d, we also observe some oscillations in the packet delivery ratio when the mobility is high, e.g., when the *pause_time* is less than 60. The packet delivery ratio of priority 0 in Case 3 seems very sensitive to the high mobility, and has worse performance than that in the Case 4, the one with piggybacking rule preferring the low priority. This can be explained as follows: When the mobility is high, the packet loss may primarily result from the mobility of nodes involved in the communications, not necessarily from the channel impairments due to other factors. On the other hand, the high mobility prolongs packet delivery and brings down the packet delivery ratio, which further results in many waiting packets of both types in the queues. In Case 3, since the piggybacking rule prefers the traffic of priority 0, quite often we may have two priority-0 packets packed together for transmission to the next hop when the channel is in state 1. If the receiver does not receive them successfully due to high mobility in this case, then more packets of priority 0 will be dropped, leading to lower packet delivery ratio, thus the packet loss due to high mobility under piggybacking rule in Case 3 may be amplified and accordingly degrades the performance further than in Case 4 for high-priority traffic. On the contrary, in Case 4, instead of packing two priority-0 packets when possible, a sender packs one packet of priority 1 with one packet of priority 0. When the packed packets cannot be successfully received due to high mobility, only one packet of each priority is involved, hence the impact on the high-priority traffic is less severe. Thus, the Courtesy Piggybacking with properly designed piggybacking rules may compensate for the negative effect of high mobility.

7 Related Work

As we mentioned in Section 1, scheduling is one promising way to support heterogonous traffic with different QoS requirements. For scheduling mechanisms, throughput and fairness are two main objectives to be met through bandwidth allocation with admission control and congestion control. Many scheduling algorithms such as fair queuing scheduling [20] and virtual clock [21] are capable of providing certain QoS guarantee for wireline networks, and many scheduling algorithms such as IWFQ [22], CIF-Q [2], CSDPS [3], and CSDPS + CBQ [23] are proposed for the wireless networks, especially for wireless cellular networks. However, little progress has been made along this direction in wireless mobile ad hoc networks with underlying contention-based MAC protocols. CSDPS and its improved version CSDPS + CBQ are two of the scheduling mechanisms that may be applicable to the ad hoc networks with contention-based MAC protocols. In CSDPS, packets to be transmitted to the same receiver are queued in the same

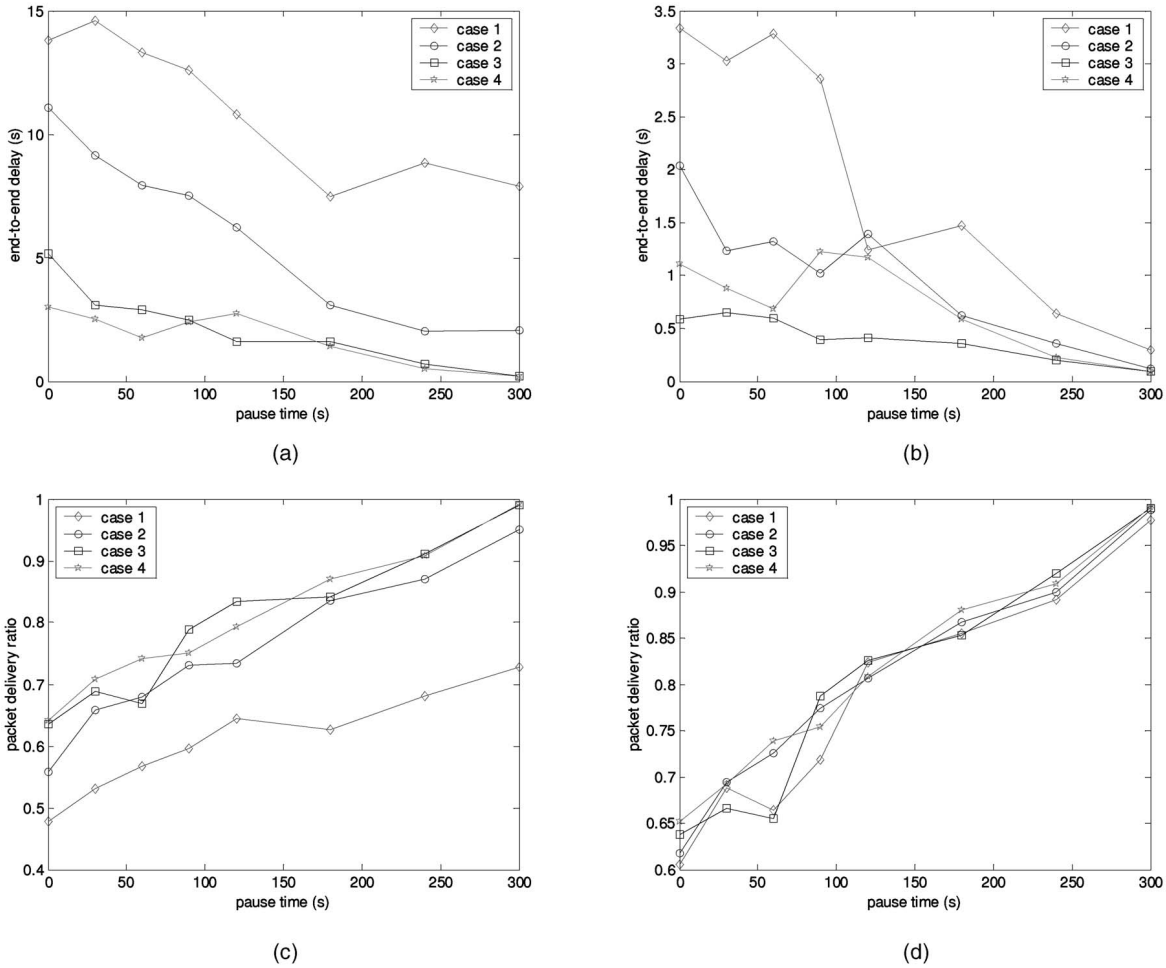


Fig. 14. Simulation results with mobility. (a) Average end-to-end delay of priority-1 service. (b) Average end-to-end delay of priority-0 service. (c) Packet delivery ratio of Priority-1 service. (d) Packet delivery ratio of Priority-0 service.

queue and are served in an FIFO fashion. At a node, the different queues are served according to some policies such as round robin, earliest timestamp first, or longest queue first. The basic idea of CSDPS is as follows: When the link toward a receiver is bad, the node should defer the transmission of packets in the queue corresponding to that receiver. With CSDPS, it is easy to alleviate the head of line (HOL) problem when a single FIFO queue is used. Since CSDPS makes use of the channel state information, it can achieve high data throughput and channel utilization. However, it does not address the fairness issue. To improve the fairness in CSDPS, class-based queuing (CBQ) [24] is used together with the CSDPS. By using CBQ, a hierarchical channel-sharing mechanism, it can achieve certain fairness, and ensure that different traffic classes can share the overall bandwidth, while maintaining the features of CSDPS to deal with the channel variations. Unfortunately, this scheme is also complicated in keeping track of the amount of service each class has been served. Efficient and less expensive mechanisms are very desirable to alleviate the conflict of throughput and fairness in MANETs. More and comprehensive materials on scheduling can be found in [16]. Besides, some QoS adaptive schemes such as SWAN [39] and Havana [40] are also available in the literature. These schemes adaptively perform admission control and rate control according to the user QoS requirements and channel states.

The main reason leading to the conflict between throughput and fairness is the limited bandwidth of the wireless link. If the system can provide plenty of bandwidth, the conflict problem would not be so significant. Recently, many adaptive transmission techniques are proposed to exploit the channel dynamics to provide more bandwidth. These schemes can adaptively adjust the parameters such as modulation level and symbol rate to maintain an acceptable BER without wasting much bandwidth. In [4], the authors integrated adaptive transmission techniques, resource allocation, and power control for a TDMA/TDD system so that higher modulation levels can be assigned to users in good channels to enhance the throughput, while power control can be used to reduce the interference and increase the system capacity. In addition to these schemes proposed for wireless cellular networks, some rate-adaptive schemes are also proposed to improve the system throughput in WLANs. In [25], the authors propose a rate adaptive MAC protocol, called RBAR, which uses the RTS/CTS to exchange the channel state information and the optimal rate on a per-packet basis. Unfortunately, this scheme needs to make some modifications to the IEEE 802.11 MAC protocols. To avoid this modification, in [26], the authors propose a scheme to select the optimal rate only with the local information at the transmitter. This scheme is based on the history of attempted transmissions. It uses one successful transmission count and one failed

transmission count to indicate the channel state and to determine the optimal rate the transmitter can use. For IEEE 802.11 MAC protocols, adaptive fragmentation schemes can also be designed with the rate adaptation to enhance the system throughput [5], [27], [28].

For all the scheduling mechanisms and other channel-dependent schemes, including our Courtesy Piggybacking scheme, designed for wireless networks, they all have to monitor the channel quality based on the symbol error rate, bit error rate, and receiver signal strength. The more accurate the channel information is, the more benefits these schemes can bring to the system design. In general, the channel estimation can be performed by the sender or by the receiver. Since the channel information used in all channel-dependent schemes is the one seen by the receiver, the receiver-based channel estimation is more attractive. However, the channel information needs to be sent back to the sender, which is sometimes costly in terms of the resource used to transmit the channel information, thus certain performance tradeoff has to be made between estimation accuracy and overhead. More details about channel quality estimation can be found in [29].

8 Conclusions

In this paper, we propose a novel Courtesy Piggybacking scheme to alleviate the conflict between throughput and fairness for different prioritized services in mobile ad hoc networks. By making use of system dynamics, such as the variable channel quality and changing traffic conditions, it can harness the available residual bandwidth that would otherwise be wasted. Thereby, it significantly improves the end-to-end delay and packet delivery ratio. More precisely, when the traffic load is light, it can shorten the end-to-end delay; when the traffic load is high, it cannot only shorten the end-to-end delay but also improve the packet delivery ratio for all prioritized services. With properly defined piggybacking rules, the piggybacking scheme can flexibly allocate the bandwidth among different types of traffic, thus achieve good fairness between different priority services without using conventional costly scheduling mechanisms. By deriving the delay bounds, we explicitly specify the performance gain the piggybacking scheme can achieve in some simplified scenarios. Further, extensive simulations verify the performance of our proposed piggybacking scheme. Our scheme is also shown to be easily implemented in a distributed fashion and, thus, could be incorporated into many scheduling schemes to provide better support of the differentiated and heterogeneous services in both mobile ad hoc networks and traditional wireless networks.

Acknowledgments

The authors would like to thank all the anonymous reviewers for their valuable suggestions that helped improve the quality of this paper. They would also like to thank Hongqiang Zhai and Dr. Xuejun Tian for their helpful discussion on the analysis of Courtesy Piggybacking. The work of Wei Liu, Xiang Chen, and Yuguang Fang was supported in part by the US Office of Naval Research under grant N000140210464 (Young Investigator Award) and under grant N000140210554, and the US National Science Foundation under grant ANI-0093241 (CAREER Award) and under grant ANI-0220287. The work of John M. Shea was supported in part by the Department of Defense Multidisciplinary University Research Initiative administered by the US Office of Naval Research under grant N000140010565. A preliminary version of this paper was presented at the 23rd IEEE International Conference on Computer Communications (INFOCOM), March 2004, Hongkong, China.

References

[1] "Defining Cross-Layer Design for Wireless Networking," *Proc. IEEE Int'l Conf. Comm. (ICC '03),* http://www.eas.asu.edu/junshan/ICC03panel.html, 2003.

[2] T.S.E. Ng, I. Stoica, and H. Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors," *Proc. IEEE INFOCOM '98 Conf.,* pp. 1103-1111, 1998.

[3] P. Bhagwat, A. Krishna, and S. Tripathi, "Enhance Throughput over Wireless LANs Using Channel State Dependent Packet Scheduling," *Proc. IEEE INFOCOM '96 Conf.,* pp. 1133-1140, Mar. 1996.

[4] I. Koutsopoulous and L. Tassiulas, "Channel State-Adaptive Techiniques for Throughput Enhancements in Wireless Broadband Networks," *Proc. IEEE INFOCOM '01 Conf.,* pp. 757-766, Apr. 2001.

[5] B. Kim, Y. Fang, T.F. Wong, and Y. Kwon, "Dynamic Fragmentation Scheme for Rate-Adaptive Wireless LAN," *Proc. IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm. (PIMRC '03),* Sept. 2003.

[6] S. Yajnik, J. Sienicki, and P. Agrawal, "Adaptive Coding for Packetized Data in Wireless Networks," *Proc. IEEE Int'l Symp. Personal, Indoor, and Mobile Radio Comm. (PIMRC '95),* pp. 338-342, 1995.

[7] M. Elaud and P. Ramanathan, "Adaptive Use of Error-Correction Codes for Real-Time Communication in Wireless Networks," *Proc. IEEE INFOCOM '98 Conf.,* pp. 548-555, Mar. 1998.

[8] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM '94 Conf.,* pp. 234-244, Sept. 1994.

[9] D.B. Johnson, D.A. Maltz, and Y.-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet Draft, draft-ietf-manet-dsr-09.txt, Apr. 15 2003.

[10] C.E. Perkins, E.M. Belding-Royer, and S. Das, "Ad Hoc on Demand Distance Vector (AODV) Routing," RFC 3561, July 2003.

[11] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *Proc. ACM SIGCOMM '94 Conf.,* 1994.

[12] C. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Acess (FAMA) for Packet Radionetworks," *Proc. ACM SIGCOMM '95 Conf.,* 1995.

[13] IEEE 802.11WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Standard, IEEE, Aug. 1999.

[14] M. Barry, A.T. Campbell, and A. Veres, "Distributed Control Algorithm for Service Differentiation in Wireless Packet Networks," *Proc. IEEE INFOCOM '01 Conf.,* 2001.

[15] X. Pallot and L.E. Miller, "Implementing Message Priority Policies over an 802.11 Based Mobile Ad Hoc Network," *Proc. IEEE MILCOM 2001 Conf.,* Oct. 2001.

[16] Y. Cao and V.O.K. Li, "Scheduling Algorithms in Broad-Band Wireless Networks," *Proc. IEEE,* vol. 89, no. 1, pp. 76-87, 2001.

[17] J. Wang and K. Nahrstedt, "Hop-by-Hop Routing Algorithms for Premium-Class Traffic in DiffServ Networks," *Proc. IEEE INFOCOM '02 Conf.,* 2002.

[18] I. Ada and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," *Proc. IEEE INFOCOM '01 Conf.,* 2001.

[19] E. Modiano, "An Adaptive Algorithm for Optimizing the Packet Size Used in Wireless ARQ Protocols," *Wireless Networks,* vol. 5, no. 5, pp. 279-286, 1999.

[20] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuein Algorithm," *Proc. ACM SIGCOMM '89 Conf.,* pp. 3-12, 1989.

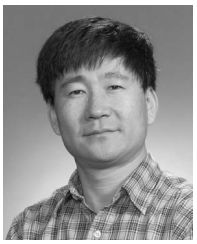
[21] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Netowrks," *Proc. ACM SIGCOMM '90 Conf.,* pp. 19-29, 1990.

[22] S. Lu and V. Bharghavan, "Fair Scheduling in Wireless Packet Networks," *IEEE/ACM Trans. Netowrking,* vol. 7, no. 4, pp. 473-489, 1999.

[23] C. Fragouli, V. Sivaraman, and M. Srivastava, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State Dependent Packet Scheduling," *Proc. IEEE INFOCOM '98 Conf.,* pp. 572-580, Mar. 1998.
[24] S. Floyd and V. Jacobson, "Link-Sharing and Resource Management Models for Packet Netwroks," *IEEE/ACM Trans. Networking,* vol. 3, pp. 365-386, Aug. 1995.
[25] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adpative MAC Protocol for Multihop Wireless Networks," *Proc. ACM MOBICOM '01 Conf.,* 2001.
[26] P. Chevillat, J. Jelitto, A.N. Barreto, and H.L. Truong, "A Dynamic Link Adaptation Algorithm for IEEE 802.11a Wireless LANs," *Proc. IEEE Int'l Conf. Comm. (ICC'03),* May 2003.
[27] J. Tourrilhes, "Dwell Adaprtive Fragmentation: How to Cope with Short Dwells Required by Multimedia Wireless LANs," *Proc. IEEE GLOBECOM Conf.,* 2000.
[28] D. Qiao and S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," *Proc. IEEE Int'l Conf. Comm. (ICC '01),* 2001.
[29] K. Balachanran, S.R. Kadaba, and S. Nanda, "Channel Quality Estimation and Rate Adaption for Cellular Mobile Radio," *IEEE J. Selceted Areas in Comm.,* vol. 17, no. 7, pp. 1244-1256, July 1999.
[30] H. Wang and N. Moayeri, "Finite-State Markov Channel-A Useful Model for Radio Communication Channels," *IEEE Trans. Vehicular Technology,* vol. 44, no. 1, Feb. 1995.
[31] http://www.opnet.com, 2004.
[32] S.-T. Sheu, T. Chen, J. Chen, and F. Ye, "An Improved Data Flushing MAC Protocol for IEEE 802.11 Wireless Ad Hoc Network," *Proc. IEEE Vehicular Technology Conf. (VTC),* 2002.
[33] A.S. Tanenbaum, *Computer Networks,* third ed. Prentice Hall, 1996.
[34] D. Bertsekas and R. Gallager, *Data Networks,* second ed. Prentice Hall, 1992.
[35] K. Altinkemer, I. Bose, and R. Pal, "Average Waiting Time of Customers in an M/D/k Queue with Nonpreemptive Priorities," *Computers & Operations Research,* vol. 25, no. 4, pp. 317-328, 1998.
[36] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing,* T. Imielinski and H.F.Korth, eds., vol. 353, Kluwer Academic Publishers, 1996.
[37] S. Shakkottai, T.S. Rappaport, and P.C. Karlsson, "Cross-Layer Design for Wireless Networks," *IEEE Comm. Magazine,* Oct. 2003.
[38] J. Yoon, M. Liu, and B. Noble, "Sound Mobility Models," *Proc. ACM MOBICOM '03 Conf.,* Sept. 2003.
[39] G.-S. Ahn, A.T. Campbell, A. Veres, and L.-H. Sun, "Supporting Service Differentiation for Real-Time and Best Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN)," *IEEE Trans. Mobile Computing,* vol. 1, no. 3, pp. 192-207, July-Sept. 2002.
[40] J. Gomez and A.T. Campbell, "Havana: Supporting Application and Channel Dependent QOS in Wireless Networks," *ACM J. Wireless Networks (WINET),* vol. 9, no. 1, pp. 21-35, Jan. 2003.

Wei Liu (S'03) received the BE and ME degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively. He is currently pursuing the PhD degree in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, where he is a research assistant in the Wireless Networks Laboratory (WINET). His research interest includes QoS, secure and power efficient routing, and MAC protocols in mobile ad hoc networks and sensor networks. He is a student member of the IEEE.

Xiang Chen (S'03) received the BE and ME degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1997 and 2000, respectively. Afterward, he worked as a MTS (member of technical staff) at Bell Laboratories, Beijing, China. He is currently working toward the PhD degree in the Department of Electrical and Computer Engineering at the University of Florida. His research interest includes resource allocation, call admission control, and QoS in wireless networks, including cellular networks, wireless LAN, and mobile ad hoc networks. He is a member of Tau Beta Pi and a student member of the IEEE.



Yuguang Fang (S'92-M'94-S'96-M'97-SM'99) received the PhD degree in systems and control engineering from Case Western Reserve University, Cleveland, Ohio, in January 1994, and the PhD degree in electrical engineering from Boston University, Massachusetts, in May 1997. From September 1989 to December 1993, he was a teaching/research assistant in the Department of Systems, Control and Industrial Engineering at Case Western Reserve University, where he held a research associate position from January 1994 to May 1994. He held a postdoctoral position in the Department of Electrical and Computer Engineering at Boston University from June 1994 to August 1995. From September 1995 to May 1997, he was a research assistant in the Department of Electrical and Computer Engineering at Boston University. 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 Electrical and Computer Engineering at New Jersey Institute of Technology, Newark, New Jersey. In May 2000, he joined the Department of Electrical and Computer Engineering at University of Florida, Gainesville, Florida, where he got the early promotion with tenure in August 2003 and has been an associate professor since then. His research interests span many areas including wireless networks, mobile computing, mobile communications, automatic control, and neural networks. He has published more than 100 papers in refereed professional journals and conferences. He received the US National Science Foundation Faculty Early Career Award in 2001 and the US Office of Naval Research Young Investigator Award in 2002. Dr. Fang has actively engaged in many professional activities. He is a senior member of the IEEE and a member of the ACM. He is and has been an editor for several journals including the *IEEE Transactions on Communications*, *IEEE Transactions on Mobile Computing*, the *IEEE Journal on Selected Areas in Communications: Wireless Communications Series*, etc. He also has been actively involved with many professional conferences including the 2000 IEEE Wireless Communications and Networking Conference (WCNC'2000) where he served as program vice chair and received the IEEE Appreciation Award for the service to this conference.

John M. Shea (S'92-M'99) received the BS (with highest honors) degree in computer engineering from Clemson University in 1993 and the MS and PhD degrees in electrical engineering from Clemson University in 1995 and 1998, respectively. He is currently an assistant professor of electrical and computer engineering at the University of Florida. Prior to that, he was a postdoctoral research fellow at Clemson University from January 1999 to August 1999. He was a research assistant in the Wireless Communications Program at Clemson University from 1993 to 1998. He is currently engaged in research on wireless communications with emphasis on error-control coding, cross-layer protocol design, cooperative diversity techniques, and hybrid ARQ. Dr. Shea was a US National Science Foundation Fellow from 1994 to 1998. He received the Ellersick Award from the IEEE Communications Society in 1996. He is an associate editor for the *IEEE Transactions on Vehicular Technology*. He is a member of the IEEE.