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

Wei Liu and Yuguang Fang
Wireless Networks Laboratory (WINET)
Department of Electrical and Computer Engineering
University of Florida
Gainesville, Florida 32611
{liuw@, fang@ece.}ufl.edu

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 mulithop 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 (CP)* to address this problem. With the recognition of inter-layer 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.**

Keywords—MANET, MAC, System design; Differentiated services; Multihop ad hoc netowrks

I. INTRODUCTION

A multihop mobile ad hoc network (MANET) is a self-configurable, self-organizing multi-hop 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. The salient features such as rapid deployment and self-organization make ad hoc networks very attractive in military and civil applications where fixed infrastructures are unavailable or unreliable, yet fast network establishments and constant reconfiguration are required. Such applications include the disaster rescue after an earthquake and collaborative computing with laptops in a classroom. Though the driven forces of developing the ad hoc networks are so strong and the revenue from such deployment may be promising, the market of such networks have not picked up yet, because many open problems need to be resolved before the expected services with desired quality can be provided. The features termed *system dynamics* [1] of mobile multihop ad hoc networks, such as time-varying and error-prone wireless link, 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 good schemes and solutions for wired networks may not be feasible in the wireless counterpart if we do not make any modification or tailor them to the wireless environments. To conquer these challenges, in recent years, many researchers advocate the so-called cross-layer design philosophy to develop the protocols and applications for MANETs, a departure from the traditional layered design for the Internet. Many researchers believe that *scheduling*, *adaptivity*, and *diversity* are the most important design issues in the context of the cross-layer design [1]. The scheduling can help shape the system dynamics [2][3], for example, the scheduling for data prioritization to support differentiated service. The adaptivity can compensate for or exploit these dynamics using adaptive modulation techniques [4] and adaptive error correction coding [6][7] to improve the throughput. The diversity provides the 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 inter-layer coupling to develop more efficient schemes to handle heterogeneous traffic over wireless links.

To realize the objective to efficiently handle heterogeneous traffic over wireless links, we need to address two problems. The first one is to handle the reliable mobile communications in MANETs. This problem has been extensively studied in recent years, and many proposed routing protocols such as DSR [8], AODV [9], and medium access control mechanisms such as MACAW [10], FAMA [11], and IEEE 802.11 [12], 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, one strategy to support QoS is to set up some kind

This work was supported in part by the U.S. Office of Naval Research under grant N000140210464 (Young Investigator Award) and under grant N000140210554, and U.S. National Science Foundation under grant ANI-0093241 (CAREER Award) and under grant ANI-0220287.

0-7803-8355-9/04/$20.00 ©2004 IEEE.

of priority scheme or service differentiation mechanism [13][14], under which the delay sensitive traffic has higher priority to access the channel over the less time-critical traffic. In the current literature, many scheduling mechanisms for wireless networks, though most of them are not directly designed for MANETs, are proposed for this purpose. 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 [15]. 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 or so-called inter-class effects [16] among traffic with different priorities. Without any precautionary measures, the conflict may lead to the bandwidth starvation for the low priority traffic when the high priority traffic load is high. Meanwhile, most of these scheduling 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 [12], 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* to alleviate the conflict between the throughput and fairness. 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 condition and the resulting dynamic bandwidth, and the dynamic characteristics of the heterogeneous traffic. One noteworthy feature of our scheme is its simplicity: the scheme is suitable for the multihop mobile ad hoc networks with underlying contention-based MAC protocols, though our scheme is applicable in the reservation-based multihop mobile ad hoc networks as well.

The rest of the paper is organized as follows. In Section II, we show the motivation of our proposed scheme. In section III, 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 IV, and performance evaluation is given in Section V. We discuss some related work in Section VI. Finally, we conclude the paper in Section VII.

II. 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 Chugach Mountain Range (i.e., the longest tunnel in North America—the Whittier tunnel in Alaska). The same is from the **S**eward to Whittier. 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 the other traffic would stop and 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, i.e., during the rush hour in the afternoon. In order to quickly pass the tunnel, a better way for other transportation is to check if there is any free space left in the train. If there is, ask for the permission to have a ride at certain cost and according to some rules, for example, how many free space left (counting in some basic unit) 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 with their own ways. Of course, in the real situations, when passengers by car, bicycle or on foot pass through the narrow and dark tunnel in a sequential manner, the traffic usually move very slowly for the sake of safety, thus it is advisable for the car that has free space to piggyback those passengers by bicycle or on foot according to some rules to benefit all the traffic. We can think of these rules as being concerned with HOW MANY-WHO problem. 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 2 units, and a car 6 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: the car, the bicycle, and person on foot.

The above scenario is very similar to the multihop mobile ad hoc networks with differentiated services. The piggyback strategy described above motivates us to develop a more efficient way to alleviate the conflict between throughput and fairness for different prioritized service. First of all, we need to identify the "free space" in a MANET. Fortunately, we do have two sources that can provide us with some free space. The first one comes from the channel condition. In recent studies such as [4][5], by taking the channel state into consideration at the MAC and PHY layer, adaptive transmission schemes can be designed to provide higher data rate. With higher data rate, the transmission time for MAC protocol data unit (MPDU) can be shortened, leading to some

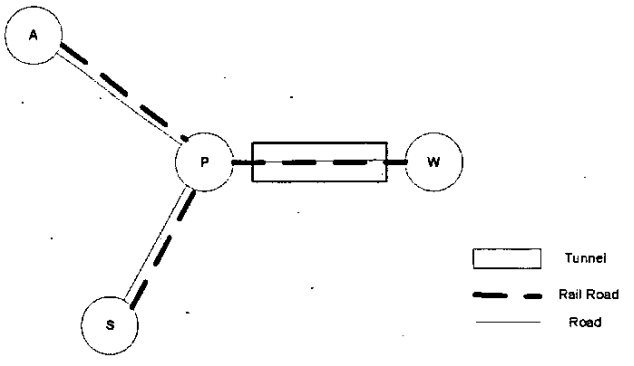

Figure 1. A tunnel scenario

potential idle time if the transmitting node does not have further data to transmit. If the IEEE 802.11 MAC is used, due to the setting of NAV (Network Allocation Vector), other nodes cannot use the medium although 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 the contention-based system (e.g., in IEEE 802.11) to fully utilize the channel capacity. To harvest such "free space", we need to design some piggybacking rules with certain objectives. In the following sections, we will elaborate more on why there exist free spaces and how the piggybacking can be used to achieve our goal —alleviating the conflict between throughput and fairness for different prioritized services.

III. PAKCET-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). The "Good" state in the BSC has low crossover probability, P_g, and the "Bad" state 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

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

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) [29] can be used. By using the received signal-to-noise-ratio (SNR) as the only side information, the FSMC

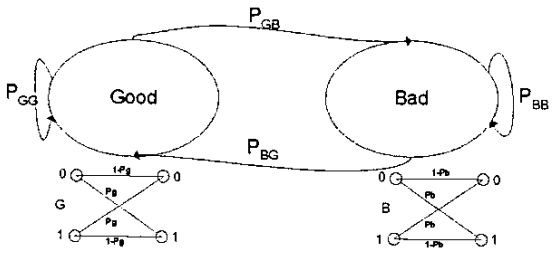

Figure 2. Gilbert-Elliott channel model

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

$$f_\gamma = \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}, \quad \gamma \geq 0. \tag{1}$$

where $\bar{\gamma}$ is the mean of γ (actually it is an exponential distribution with mean of $\bar{\gamma}$). In order to build the finite state Markov chain, 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 $0=\gamma_0<\gamma_1<\gamma_2<\cdots\cdots<\gamma_{k-1}<\gamma_k=\infty$ be the thresholds. For each interval, we associate it with a state S_k, k=0,1,2,3...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 value γ. When BPSK is used, this probability can be written as a function of γ:

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

According to [18], 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 for this state can be derived using (1)(2)(3) as equation (4).

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

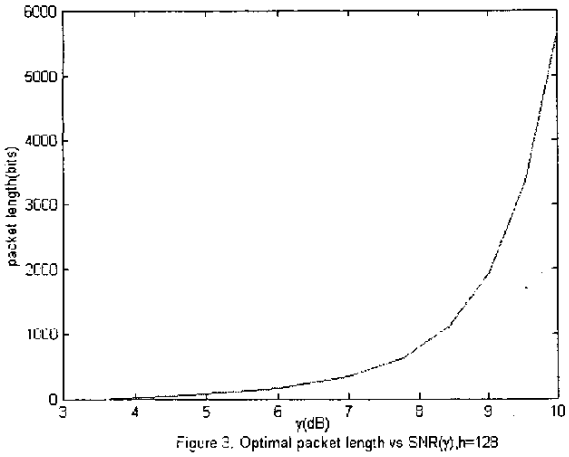

Figure 3. Optimal packet length vs SNR(γ),h=128

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

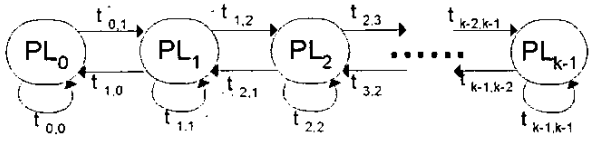

Figure 4 . Packet-Length-Based Finite-State Markov Channel

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

IV. Courtesy Piggybacking

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

A. 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.

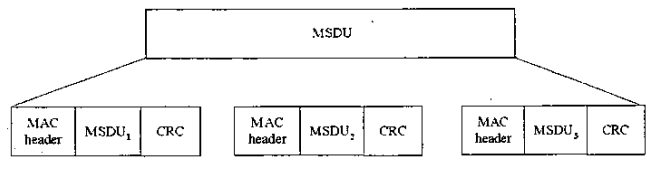

Figure 5. Fragmentation in IEEE 802.11

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. 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 towards that position at a random speed uniformly distributed in the range from 0 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 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.

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 IEEE 802.11 MAC protocol [12], this packet length may be called as frame length, which equals to the fragmentation threshold plus the length of the MAC header, the length of CRC and the overhead of PHY. In 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 IEEE 802.11. Since the length of the MAC overhead and PHY overhead may be kept unchanged or very little changed, according to the analysis in the previous section, different channel states have different frame length, we can say that different channel state will have different fragment thresholds (FTs). The greater the received SNR is, the greater the fragment threshold (FT) will be. In order to improve the channel utilization, we assume the MAC protocol can adaptively adjust the fragmentation threshold and transmission rate according to the channel state. To accurately figure out 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.

B. 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. Then, the overall overhead should be the total of the overheads O_{ip} at the IP layer (e.g., 20 bytes for IPv4), O_{mac} at the MAC layer (e.g., 34bytes for IEEE 802.11) and O_{phy} at the PHY layer (e.g., 16 bites). 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 vs. 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 will cause lots of DATA/ACK exchanges before the successful transmission of the packet, even when the channel is not bad. 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 equal to any of FT_k (k = 0, 1, 2, … , 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. 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 when the SNR is close to 20 dB and 64 QAM modulation scheme is used [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 which has seized the channel should treasure every transmission opportunity to transmit as many bits as possible, especially when the channel is in good conditions. From now on, we call the state S_i the free-space-effective state when i is greater than m, otherwise the non-free-space-effective state, although such a state may still provide the possibility to pack more data 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 and transmits a packet, it will first check the channel state and determine if it is in a free-space-effective state and is capable of packing more bits to piggyback more packets in one transmission. If the channel is in a free-space-effective state, the node can transmit more bits, and can piggyback some more bits from the queue(s), which may have different priorities but with the same next hop in 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 bits from the Tx queues.

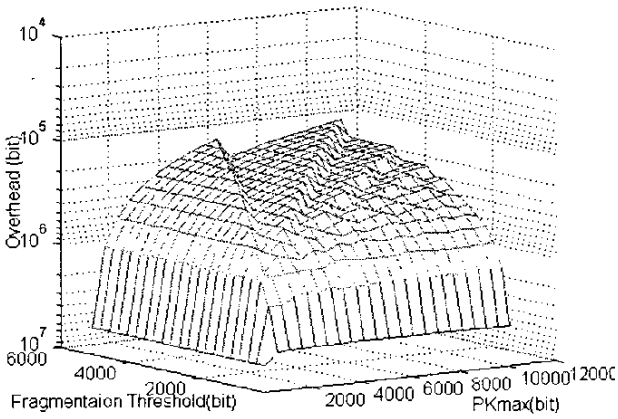

Figure 6. The total overhead with PKmax & FT

After identifying the existence of the free space, the most fundamental part in the Courtesy Piggybacking scheme is to design rules that guide the MAC layer to assemble enough and proper bits from the Tx queues (HOW MANY- WHO problem) and piggyback them to the next hop to alleviate the conflict we intend to address. The basic idea for such piggybacking rules is that under different channel states, we assembly 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, we can pack other services possible with lower priority to share the residual bandwidth with the high priority traffic. One of such rules is the one that prefers the 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, 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 scarify some of its own performance for more efficient channel utilization by its **courtesy**--piggybacking the low priority service. One of such rules is to always piggyback the MSDUs from the longest Tx queue.

To illustrate the Courtesy Piggybacking scheme, we demonstrate the operation of the scheme in Fig. 7. First, different priority packets called MSDUs arrive from the network layer as b-MSDUs (basic MSDUs, the *basic unit*) whose length agrees with the FT_m. We assume the packet maximum value PK_{max} is strictly enforced at the upper layer; if not, the oversized MSDUs 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 queue corresponding to their priorities. The dequeue controller operates according to the piggybacking rule, dequeues one or more b-MSDUs with the same next hop for the routing purpose and form a MPDU satisfying the FT

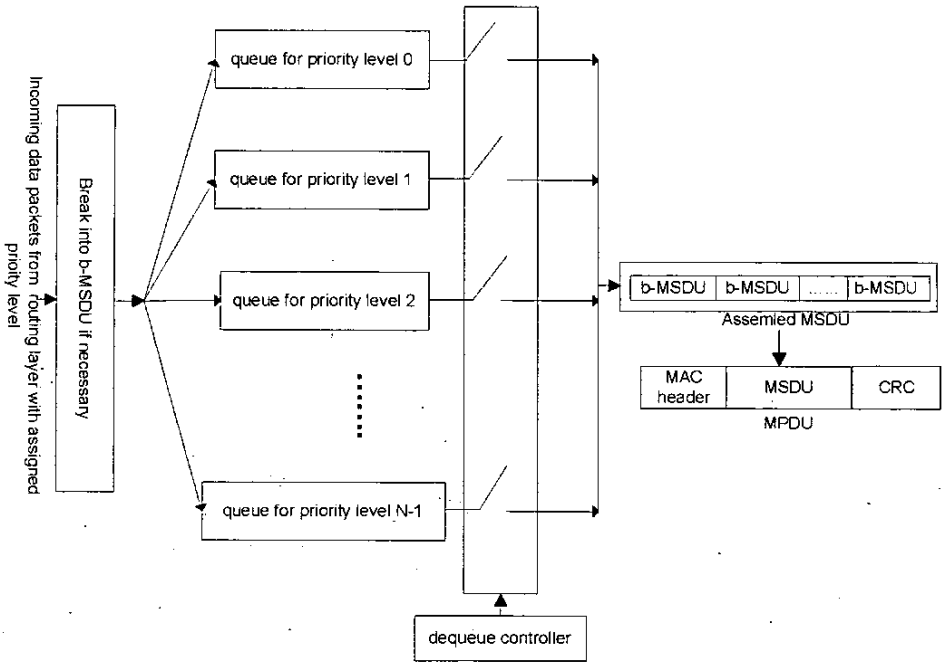

Figure 7 . Illustration of Courtesy Piggybacking Scheme

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 no longer than FT_m. In order to avoid further fragmentation of b-MSDUs 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 the FT_m, i.e., the frame length for state i satisfies $FT_i=g_i \times FT_m$, where g_i is a positive integer. This can be achieved by properly partition the range of received SNR and channel-dependent modulation schemes. To reduce the transmission time for a long frame, rate adaptive transmission scheme 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 the receiver in unpacking the bound packets, an unused bit 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, unpack the bound packets one by one according to the unused bit value.

C. *Some Further Discussions*

If we examine the destination of the bits in a single piggybacked transmission, 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 2 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 node 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.

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 packet transmissions and delivered to the next hop for free, thus it can statistically reduce the time taken to contend to access the channel for the low priority services. This benefit will be more pronounced in mobile ad hoc networks using service differentiation based MAC protocols [13][17] where the MAC protocols scarify the low priority service quality to support high priority service through either time spacing (differentiation of Interframe Space (IFS)) or backoff parameters [12]. On the other hand, the reduction of contention from low priority services can in turn bring some benefit to the high priority services. One node's courtesy piggybacking of low priority services may help its neighbors' transmissions of high priority traffic, because less low priority traffic will reduce the contention for the high priority traffic. We also want to point out that the piggybacking may increase some delay jitter. And one may expect that even with the same

priority level, one packet arrives at one node later may leave the node earlier than some earlier arrival packets.

The piggybacking rule 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 for being served, the design of piggybacking rule can be viewed as an allocation problem. As discussed above, by proper partitioning the range of received SNR, the fragmentation threshold of the free-space-effective state i satisfies $FT_i=g_i \times FT_m$, where g_i is a positive integer. For other non-free-space-effective states, let $g_i=1$. Suppose we have totally K different channel states, and N different priority levels. Let α_{ij} denote the number of b-MSDUs of j priority level to be piggybacked when the channel is in state i. We 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 $\alpha_{iN-1}=1$, and $\alpha_{ij}=0$ for $0 \le i \le m$ and $0 \le j \le N-2$. If we neglect the MAC layer overhead, then the design problem can be reduced to choose α_{ij} such that

$$\begin{cases} 0 \le \alpha_{ij} \\ \sum_j \alpha_{ij} \le g_i \end{cases} \qquad 0 \le i \le K-1, 0 \le j \le N-1$$

Thus, the expected value of throughput of any priority level at one node should be $R_i \sum p_i \alpha_{ij}$, where p_i is the probability that the channel is in state i, and R_i is the transmission rate in state i.

In our proposed courtesy piggybacking scheme, only the traffic sharing the same next hop can be packed. If we want to extend it to the different next hop scenarios, further modifications to the MAC layer are needed to provide the data link layer acknowledgements. In addition, to avoid fragmentation of the b-MSDUs in the free-space-effective state, in our piggyback scheme we should maintain the multiplicative relationship between the fragmentation threshold (FT) at 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 heavy traffic load, the piggybacking rule favorable of high priority services may 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.

In order to fully make use of the channel dynamics, it is advisable to give the nodes with better channel condition better chances to seize the channel. To achieve this, differentiation of Interframe Space (IFS) and backoff parameters can be appropriately designed. Moreover, our Courtesy Piggybacking scheme does not exclude the scheduling mechanisms; in fact, the scheduling can be still used at higher layers to enhance management of the heterogeneous traffic. In our preliminary implementation of the piggybacking, the b-MSDUs are organized in the queues according to their priorities. When the transmitter wants to pack more bits to the same receiver, exhaustive search is

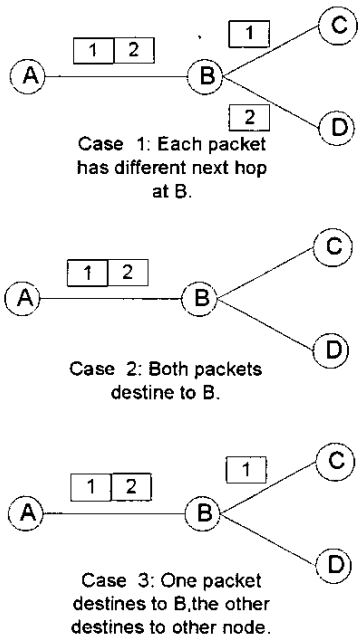

Figure 8. Three piggybacking cases

carried out to find out the proper bits in candidate queues according to the piggybacking rules. This may not be the most efficient way, and further investigations on how to efficiently organize the b-MSDUs and fetch the proper b-MSDUs are ongoing.

In this paper we only consider the ad hoc networks using contention-based MAC protocols. We should point out that our Courtesy Piggybacking scheme also works in the networks with reservation-based MAC or hybrid MAC protocols. In addition to the free space provided by the channel dynamics, in the reservation-based MAC protocols, 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.

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 the residual resource is too short to be given to other services due to the overhead of establishing a new connection. Besides, some MAC protocols such as 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 make the harvested resource from the rate adaptation useless. Thus, the courtesy piggybacking by high priority traffic flows makes more sense.

V. Performance Evaluation

In order to evaluate the performance of the proposed Courtesy Piggybacking scheme, we use the simulator OPNET

[30]. In our simulation study, we assume a slow fading channel with only two states and $FT_1=2\times FT_0$. The duration for which the channel stays in either state is exponentially distributed with mean 0.5 second, and the transition probability is 0.5 for each state. Without loss of generality, we limit the number of priorities to 2. We simulate an ad hoc network consisting of 50 mobile nodes, whose mobility follow the waypoint mobility model in a $1500\times300m^2$ 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 1024 bits/packet that agrees with the FT_0 and each packet is a b-MSDU. The generated traffic is further assigned with 0 (low priority) or 1 (high priority) with probability 0.5. All packets are buffered in the queues according to their priorities. We define two piggybacking rules for comparison. Rule 1 prefers the higher priority. When the channel state is in FT_0, a priority 0 service can be served only when no priority 1 services exist in the queue. When the state is FT_1, the priority 0 service is piggybacked by priority 1 service only when no more priority 1 services left in the queue (of course, the packets should share the same next hop in the routing table). In rule 2, when the channel is FT_0, the stations act as the rule 1. When the state is FT_1, the priority 0 service is piggybacked no matter how many priority 1 services left in the queue. We study the performance of the networks with four scenarios: case 1, the networks unaware of channel states; case 2, the networks aware of channel states with dynamic transmission rate; case 3, the networks employing the Courtesy Piggybacking with rule 1; and case 4, the networks employing the Courtesy Piggybacking with rule 2. We compare their performance in terms of end-to-end delay and packet delivery ratio. In our simulations, when the channel is in state with FT_0, we use basic transmission rate 1Mbps, while for state with 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. We run each experiment for 300 seconds.

We first study the performance of our Courtesy Piggybacking under different traffic loads. In the relative light traffic load, the average packet inter-arrival time is 0.3s, while in the scenario with relative heavy traffic load, the average packet inter-arrival time is 0.25s. In these two scenarios, no mobility is considered. From Fig. 9, we can see the inter-class effects in the differentiated service system, especially when the high priority traffic load is high. From Fig. 9.a and 9.b, we can clearly observe that the delay for priority 0 (7.89 second) is far greater than that for priority 1 (0.299 second). The same phenomenon happens to the packet delivery ratio. We also observe that the cases employing our piggybacking scheme (case 3 and case 4) have better performance than those without piggybacking scheme (case 1 and case 2). When the traffic load is heavy, our piggybacking scheme can improve the packet delivery ratio for both priorities. The courtesy piggybacking scheme can greatly shorten the end-to-end delay for both priorities as well. When the traffic is light, for all cases, the packet delivery ratio is very close to 1. However, our scheme still has the ability to shorten the end-to-end delay for both types of traffic. From Fig. 9, though channel aware mechanisms can improve both measured metrics in case 1, our piggybacking scheme provides more benefits that the channel aware mechanism alone cannot achieve.

Next, we study the impact of mobility on the performance of the proposed Courtesy Piggybacking scheme when $1/\lambda=0.25$. Fig. 10.a to Fig. 10.d show the simulation results. We observe that all the cases are sensitive to mobility and the performance degrades as the mobility increases, which is consistent with our intuition. When the *pause_time* is 300 with minimal mobility consideration, all the measured metrics are optimal. When the mobility reaches the highest point, i.e., *pause_time*=0, the measured metrics reach their worst value. In addition, we can clearly observe that case 2, case 3, and case 4 have better performance than case 1. In general, the three cases have shorter end-to-end-to delay and higher packet delivery ratio than case 1 for both priorities, especially for low priority 0, the most disadvantaged traffic in the differentiated service system. Since all the cases except case 1 make use of the channel states and rate adaptation, we validate that the dynamic channel states can be used to improve the channel utilization.

We can also compare case 3 and case 4 as a group with case 2 and study the effectiveness of our courtesy piggybacking scheme. From Fig. 10.a to 10.d, 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. We observe that the piggybacking scheme not only improves the performance of the priority 1 traffic, the highest priority traffic in the system, but also improves significantly the performance of the priority 0 traffic. This validates that our courtesy piggybacking scheme is capable of alleviating the conflict between the different prioritized traffic. According to our discussion in Section IV, all these gains beyond those in case 1 should come from the courtesy piggybacking scheme. In case 1, 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. While 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.

Finally, we focus on the case 3 and case 4 and study the impact of the piggybacking rules. The piggybacking rule in case 3 is the one that prefers the high priority traffic in the system, the priority 1 while the rule in case 4 is the one that prefers the low priority 0 traffic. Thus, there is no surprise that in Fig. 10.a and Fig. 10.c, the end-to-end delay for the priority 0 traffic 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 1 traffic, all the measured metrics generally have better performance in case 3 than those in case 4. We can see that the courtesy piggybacking in case 4 sacrifices the priority 1 traffic to piggyback the priority 0 traffic. In Fig. 10.d, 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 1 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 high priority 1, quite often we may have two priority 1 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 1 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 contrary, in the case 4, instead of packing two priority 1 packets when possible, a sender packs one packet of priority 0 with that packet of priority 1. 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.

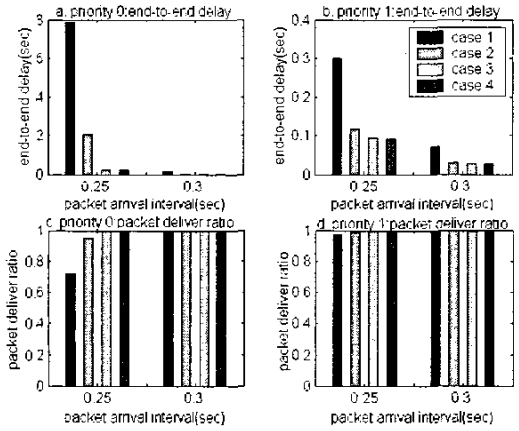

Figure 9. Simulation results with different packet arrival rates.

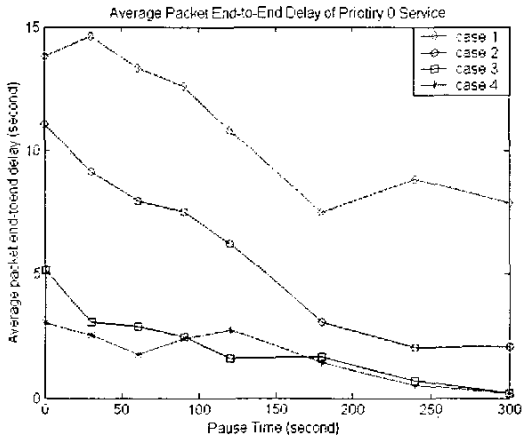

Figure 10.a. Average end-to-end delay of priority 0 service

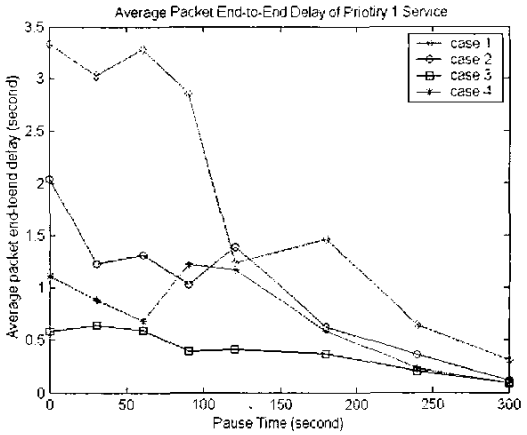

Figure 10.b Average end-to-end delay of priority 1 service

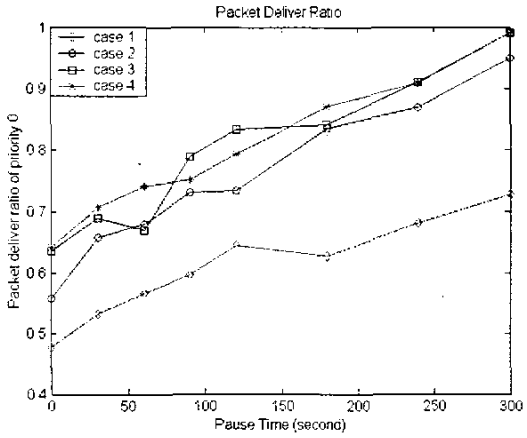

Figure 10.c. Packet delivery ratio of Priority 0 service

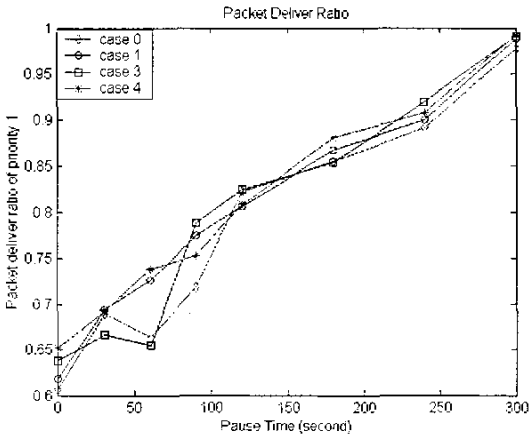

Figure 10.d. Packet delivery ratio of Priority 1 service

VI. Related Work

As we discussed in Section I, the scheduling is one promising way to support heterogonous traffic with different QoS requirements. For the scheduling mechanisms, the

throughput and fairness are two main objectives to meet through bandwidth allocation with admission control and congestion control. Many scheduling algorithms such as fair queuing scheduling [19], and virtual clock [20] are capable of providing certain QoS guarantee for wireline networks, and many scheduling algorithms such as IWFQ [21], CIF-Q [2], CSDPS [3], and CSDPS + CBQ [22] 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 scheduling mechanisms that may be applicable to the ad hoc networks with contention-based MAC protocols. In the CSDPS, the packets to be transmitted to the same receiver are queued in the same 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 towards 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 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) [23] 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 compressive materials can be found in [15].

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 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 [24], 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 [25], 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] [26][27].

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, certain performance tradeoff has to be made. More details about channel quality estimation can be found in [28].

VII. Conclusions

In this paper, we propose a novel scheme, called *Courtesy Piggybacking*, to alleviate the conflict between throughput and fairness for different prioritized traffic in a differentiated service system in mobile ad hoc networks. By making use of the system dynamics such as the channel dynamics and traffic dynamics, our piggybacking scheme can improve the end-to-end delay and packet delivery ratio significantly. When the traffic load is light, our piggybacking scheme can shorten the end-to-end delay. When the traffic load is high, our piggybacking scheme can not only shorten the end-to-end delay, but also improve the packet delivery ratio for all priorities. Our piggybacking scheme with proper piggybacking rule functions well for MANETs with high mobility. We also investigate the impacts of different piggybacking rules and show that a properly designed rule has the ability to "softly" allocate the bandwidth among different types of traffic. Extensive simulation studies show that our piggybacking scheme can harvest the residual bandwidth that might be left unused when the information of channel and traffic is used.

From the simulation studies, we can also observe that our courtesy piggybacking scheme is an efficient way to alleviate the conflict of throughput and fairness among different traffic with different priorities. Moreover, our scheme is easy to implement and can be implemented in a distributed fashion. Finally, it is also possible to incorporate our courtesy piggybacking into many scheduling schemes to provide better support of the differentiated and heterogeneous services in mobile ad hoc networks and traditional wireless networks.

References

[1] "Defining cross-layer design for wirelss networking," *Panel in Proc. of ICC '03*, http://www.eas.asu.edu/~junshan/ICC03panel.html.

[2] T.S. Eugene Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks with locaiton-dependent errors," in *Proc. of INFOCOM'98*, Mar. 1998, pp.1103-1111.

[3] P. Bhagwat, A. Krishna, and S. Tripathi, "Enhance throuthgput over wireless LANs using channel state dependent packet scheduling," in *Proc. of INFOCOM'96*, Mar. 1996, pp. 1133-1140.

[4] I. Koutsopoulous and L. Tassiulas, " Channel state-adptive techiniques for throuthgput enhancements in wireless broadband networks," in *Proc. of INFOCOM'01*, Apr. 2001, pp. 757-766.

[5] B. Kim, Y. Fang, and T. F. Wong, "Dynamic Fragmentation Scheme for Rate-Adaptive Wireless LAN," in *Proc. of PIMRC '03*, Sept. 2003.

[6] S.Yajnik, J. Sienicki, and P. Agrawal, "Adaptive coding for packetized data in wireless netiwrks," in *Proc. of PIMRC'95*, vol.1995, pp.338-342.

[7] M. Elaud and P. Ramanathan, "Adaptive use of error-correction codes for real-time communication in wireless networks," in *Proc. of INFOCOM'98*, Mar. 1998, pp. 548-555.

[8] D. B. Johnson, D. A. Maltz, Y-C. Hu, The dynamic source routing protocol for mobile ad hoc networks. *IETF Internet Draft*, draft-ietf-manet-dsr-09.txt, April 15, 2003.

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

[10] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, "MACAW: A media access protocol for wireless LANs," in *Proc. of ACM SIGCOMM'94*, 1994.

[11] C. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple acess (FAMA) for packet radionetworks," Acm SIGCOMM'95, 1995.

[12] *IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*,1997.

[13] M. Barry, A. T. Campbell and A. Veres, "Distributed control algorithm for service differentiation in wireless packet networks," in *Proc. of INFOCOM'01*.

[14] X. Pallot and L. E. Miller, "Implementing message priority policies over an 802.11 based mobile ad hoc network," in *Proc. of IEEE MILCOM 2001*, Oct. 2001.

[15] Y. Cao and V.O.K. Li, "Scheduling algorithms in broad-band wirelss networks," in *Procedings of the IEEE*, vol.89, no.1, pp. 76-76-87,2001.

[16] J. Wang and K. Nahrstedt, "Hop-by-Hop routing algorithms for premium-class traffic in DiffServ networks." in *Proc. of INFOCOM'02*, 2002.

[17] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11." in *Proc. of IEEE INFOCOM'01*, 2001.

[18] Eytan Modiano, "An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols," *Wireless Networks*,Vol.5, No.5, pp.279-286,1999.

[19] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queuein algorithm," in *Proc. of ACM SIGCOMM'89*, 1989, pp.3-12.

[20] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switching netowrks, " in *Proc. of ACM SIGCOMM'90*, 1990, pp. 19-29.

[21] S. Lu and V. Bharghavan, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. Netowrking*, vol.7, no. 4, pp. 473-489,1999.

[22] C. Fragouli, V. Sivaraman, and M. Srivastava, "Controlled multimedia wirelss link sharing via enhanced class-based queuing with channel-state dependent packet scheduling," in *Proc. of INFOCOM'98*, Mar. 1998, pp. 572-580.

[23] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet netwroks," *IEEE/ACM Trans. Netwroking*, vol.3, pp. 365-386, Aug. 1995.

[24] G. Holland, N. Vaidya and P. Bahl, "A rate-adpative MAC protocol for multi-hop wireless networks," in *Proc. of ACM MOBICOM'01*, 2001.

[25] P.Chevillat, J. Jelitto, A. Noll Barreto, and H. L. Truong, " A dynamic link adaptation algorithm for IEEE 802.11a wireless LANs," in *Proc. of ICC'03*, May 2003.

[26] J. Tourrilhes, " Dwell adaprtive fragmentation: how to cope with short dwells required by multimedia wirelss LANs," in *Proc. of IEEE Globecom 2000*.

[27] D. Qiao and S. Choi, " Goodput Enhancement of IEEE 802.11a Wireless LAN via link adaptation," in *Proc. of ICC'01*,2001.

[28] K. Balachanran, S. R. Kadaba, and S. Nanda, "Channel quality estimation and rate adaption for cellular mobile radio." *IEEE Journal on Selceted Areas in Communications*, vol.17, no.7, pp. 1244-1256, July 1999.

[29] 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.

[30] http://www.opnet.com.