Adaptive Frame Concatenation Mechanisms for QoS in Multi-rate Wireless Ad Hoc Networks

Ming Li[1], Hua Zhu[2], Yang Xiao[3], Imrich Chlamtac[4], B. Prabhakaran[5]
[1] Department of Computer Science, California State University, Fresno, CA 93740, USA (mingli@csufresno.edu)
[2] San Diego Research Center, San Diego, CA 92121 USA (hua.zhu@sdrcinc.net)
[3] Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, USA (yangxiao@ieee.org)
[4] Create-Net, Via Solteri 38, Trento 38100, Italy (chlamtac@create-net.org)
[5] Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA (praba@utdallas.edu)

Abstract— **Providing Quality of Service (QoS) to users in a wireless ad-hoc network is a key concern for service providers. With the availability of multiple rates in IEEE 802.11a/b/g wireless LANs, it is desirable to improve the network capacity and temporal fairness by sending multiple consecutive frames (also referred as frame concatenation mechanism in [2]) over high rate links, as proposed in opportunistic auto rate (OAR [1]). However, this scheme does not consider the effect of frame sizes and may yield unsatisfactory performance for high priority multimedia flows transmitting over low rate links. Therefore, a more appropriate frame concatenation strategy and a corresponding service differentiation scheme should be devised to provide better performance for high priority voice/video flows than low priority data flows, under various channel rate scenarios. In this paper, we first analyze the effect of frame size on the performance of OAR. Then, we propose a general concatenation mechanism (GCM), a more accurate frame concatenation mechanism for multi-rate MAC with better fairness. Finally, we propose two mechanisms: adaptive weighted fair frame concatenation mechanism (AWFCM) and adaptive QoS aware frame concatenation mechanism (AQCM), for supporting service differentiation and QoS in multi-rate wireless ad hoc networks. The primary idea is to adjust the number of concatenated frames based on flow weight/priority, frame sizes, link rates, and network traffic. Simulation results show that the proposed mechanisms achieve desirable performance on supporting multimedia applications in multi-rate wireless ad-hoc networks.**

Keywords-multi-rate, wireless LAN, multimedia, QoS

I. INTRODUCTION

Ad-hoc networking has some inherent benefits like ease of mobility, de-centralization and it needs no fixed infrastructure thus it is growing quickly both in terms of importance and applications. One of the major applications is multimedia which includes services such as video-conferencing, on-demand video and audio services, VOIP etc. The basic problem with multimedia services is that they are delay intolerant and require high bandwidth. To improve the network capacity, multi-rate capability is enabled in most wireless cards. For example, IEEE 802.11b standard allows transmission at rates of 2, 5.5, and 11 Mbps, for various channel conditions and transmission range requirements. For normal data transmissions, a higher data rate is preferred if the channel is better, whereas a lower data rate is preferred if the channel is worse. However, transmission range also plays a role in determining data rates since a higher data rate usually indicates a shorter transmission range. A lower data rate has a longer transmission range, therefore increasing coverage and reducing the chances of hidden nodes. For example, control frames (ACK/RTS/CTS) in IEEE 802.11b are all transmitted with the basic rate of 2Mbps to reach large ranges.

However, in standard CSMA/CA MAC protocol, each contending station can only transmit one single frame per each successful channel access, regardless of link rate and frame size. This basic mechanism incurs high overhead, especially when link rate is high and frame size is small. As pointed out in [1] and [2], the capacity of the basic Distributed Coordination Function (DCF) can be significantly improved by the frame concatenation mechanism for the scenarios of high link rate and small frame sizes, respectively. With frame concatenation, more than one frame can be transmitted per each successful channel access. Actually, frame concatenation is also considered as an important technique for improving the network capacity in IEEE 802.11n standard, which is designed for very high throughput.

Therefore, it is a natural idea to combine the link rate based and frame size based frame concatenation mechanisms together to maximize temporal fairness, i.e., flows receiving their *fair* shares in total channel access time regardless of variations of link rates and frame sizes. Note that *fair* is not necessarily *equal*. To support multimedia applications such as voice/video streaming, it is imperative that flows are differentiated in the number of concatenated frames based on their priority or flow weight. Finally, it is desirable that the number of concatenated frames should be changed adaptively according to traffic characteristics and traffic load such that the service differentiation takes effect with the presence of high traffic load, whereas low priority flows can still maintain the temporal fairness to maximize their performance under low traffic load of high priority flows.

In this paper, we first analyze the temporal fairness of OAR protocol and propose a general frame concatenation mechanism (GCM) by considering the frame sizes. This new mechanism is especially efficient for small frame voice data.

Then, we introduce flow weights to distinguish the number of concatenated frames for different traffic such as voice, video, and best effort traffic in conjunction with different channel link rates and frame sizes. This adaptive weighted fair frame concatenation mechanism (AWFCM) provides a baseline for supporting multimedia applications under high traffic load. Based on AWFCM, we devised an adaptive QoS frame concatenation scheme (AQCM) to change the number of concatenated frames according to the traffic load at each priority level. To provide guaranteed quality of service, admission control is enforced for high priority voice and video flows to prevent or at least mitigate performance degradation due to network saturation. Extensive simulation results have shown that the proposed schemes significantly improve the QoS in multi-rate ad hoc networks.

This paper is organized as follows. Section II discusses related works in multi-rate protocols and QoS. Section III proposes a more efficient frame concatenation mechanism. Section IV presents AWFCM and AQCM for service differentiation and QoS in multi-rate ad hoc networks. Section V evaluates the proposed schemes and discusses the simulations results and Section VI concludes this paper.

II. RELATED WORK

Multi-rate MAC: Despite of the multi-rate capability specified by IEEE 802.11a and 802.11b standard and enabled by most wireless adapters at physical layer, the original MAC protocol such as distributed coordination function (DCF) [3] envisages fixed single-rate of transmission between participating nodes in the network. The idea is that for every successful channel access, each contending station can only transmit one frame so that throughput fairness can be well maintained. As an enhancement, Auto Rate Fallback (ARF) protocol [4] was the first commercial MAC implementation with multi-rate capability enabled. With ARF, stations attempt to use higher transmission rates after consecutive successful transmissions, which usually indicate high channel quality, and revert to lower rates after failures. Under most channel conditions, ARF provides a performance gain over fixed single-rate IEEE 802.11. Then, Holland and Vaidya proposed Receiver Based Auto Rate (RBAR) [5] where receivers measure the channel quality using the signal strength of the request-to-send (RTS) message. Based on this signal strength information, receivers choose the most appropriate transmission rate and notify the corresponding sender through the clear-to-send (CTS) message. Since the RTS message is sent shortly before data transmission, the estimation of the channel condition is quite accurate. Thus, RBAR yields significant throughput gains than ARF.

Multi-rate Protocols: Sadeghi and Kanodia further extended the idea of RBAR and proposed Opportunistic Auto Rate (OAR) [1]. In OAR, each contending station potentially can transmit multiple frames per successful channel access according to the dynamically-determined channel rate. With this improvement, OAR can achieve temporal fairness and explore the potential of high data rates, thus achieving significant throughput improvements. Wang and Zhai [6] investigated the problem of multi-user diversity in which a node concurrently communicates with several neighbors and proposed a similar frame concatenation scheme as OAR to further improve network performance. However, neither OAR nor OSMA provides service differentiation or QoS guarantee for multimedia flows. To address the issue of unfairness of OAR with IEEE 802.11e EDCF when prioritized flows are present, Kim and Suh [9] proposed an adaptive protocol to ensure fair bandwidth shares for different traffic types even though the corresponding link rates are different.

Frame Concatenation: The IEEE 802.11 standard includes a "frame train" model to handle frame segmentation for large data frames. Xiao [2] proposed a frame concatenation mechanism (CM) of non-segmentation frames for throughput improvement in IEEE 802.11 wireless LANs. In this scheme, each station can send a train of "back-to-back" small frames to significantly reduce the overhead of the legacy DCF protocol and thus achieve much higher network capacity. Zhai and Fang [7] proposed a distributed adaptive packet concatenation (APC) scheme to combine several short frames into a super frame based on a coherent time defined in OAR to eliminate unnecessary protocol overheads. Both schemes and OAR show that CM is a very effective technique for improving network capacity of IEEE 802.11 networks with varying packet sizes and link rates.

QoS in Multi-rate Networks: Li and Prabhakaran [12] devised a reliable QoS strategy for multi-rate multi-hop ad hoc networks. In this scheme, link rate and interference are integrated to compute the route available bandwidth (RAB) of specific path for each flow. Admission control decision is based on the route reliability and RAB. However, no frame concatenation is used. Tan and McLaughlin [10] proposed a framework for multi-rate ad hoc networks. In this framework, OAR-based frame concatenation and differentiated service (DiffServ) [11] model are integrated to support multimedia application. However, the frame concatenation is static only without adaptation according to network conditions.

III. GENEARL CONCATENATION SCHEME

It has been shown that frame concatenation significantly improves network capacity in various scenarios [1][2][7]. Intuitively, since multiple frames are sent back-to-back without channel contention, the overhead of each data transmission is largely reduced and network capacity is increased significantly. On the other hand, the fairness and QoS may be affected when stations use different concatenation numbers. In this section, we focus on the fairness issue and will discuss QoS issue in the next section.

A. *Temporal fairness in multi-rate networks*

In CM [2], it was proposed that every station keeps sending frames until the total data size per channel access reaches a certain threshold. In OAR [1], a more general concept of *temporal fairness*, i.e., stations should be granted approximately the same channel time for data transmissions per each channel access. Based on the principle of maintaining temporal fairness, OAR sends 1, 3, and 5 concatenated frames when the best channel rate is 2, 5.5, and 11 Mbps in IEEE 802.11b, respectively. Obviously, for cases with single rate channel, CM does maintain temporal fairness by sending the same amount of data whenever a station accesses the channel. On the other hand, for cases with uniform frame size, OAR maintains good temporal fairness in that the number of concatenated frames is proportional to the channel data rate. Thus, both CM and OAR maintains temporal fairness under certain conditions.

In this paper, we consider a more general frame concatenation scheme where both frame sizes and link rates are considered. We define our problem statement as follows: *Given the average frame size of a flow and the corresponding link rate, what is the number of concatenated frames so that temporal fairness is maintained?*

B. GCM: General Concatenation Mechanism

To address the above issue, we begin with the basic operation of OAR where a RTS/CTS handshaking is employed, and then for each transmitted frame, an ACK is also used. In a general case, if N frames are concatenated and the determined rate is r Mbps for a flow with average frame size of L, then the total channel time consumption for this particular data transmission can be computed as $T_{L,r}=RTS/2+SIFS+CTS/2+N\cdot(SIFS+L/r+SIFS+ACK/r)$. On the other hand, if a station is transmitting a flow of large frame size $L_{\max}$ with the basic channel rate of 2 Mbps, then only one frame should be transmitted. Thus, the total channel time consumed for this particular data transmission is $T_{L\max,2}=RTS/2+SIFS+CTS/2+SIFS+L_{\max}/2+SIFS+ACK/2$. For temporal fairness, we have that $T_{L\max,2}=T_{L,r}$. Thus $(RTS+CTS)/2+3\cdot SIFS+(L_{\max}+ACK)/2=(RTS+CTS)/2+(2N+1)\cdot SIFS+N\cdot(L+ACK)/r$. Therefore, we can compute N, the number of concatenated frames corresponding to a specific flow average frame size L and channel rate r as

$$N(L,r)=\frac{2\cdot SIFS+(L_{\max}+ACK)/2}{2\cdot SIFS+(L+ACK)/r} \qquad (1)$$

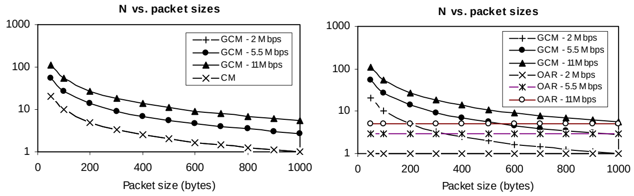
The number of concatenated frames for temporal fairness depends on the actual channel rate and the flow frame size. The higher the channel rate, the larger N is, and vice versa. The smaller the frame size, the larger N is, and vice versa. In an extreme case, N can be quite large under the highest link rate (11 Mbps for IEEE 802.11b) and a very small frame size (e.g., 40 bytes). Nevertheless, from the perspective of temporal fairness, such large N is still reasonable.

It should be noted that for $r = 2$ Mbps, since $SIFS$ and ACK are relatively small, therefore $N(L,2)\approx L_{\max}/L$, which is the same as the concatenation scheme (CM) proposed by Xiao [2]. On the other hand, if the frame size is L_{max}, since $SIFS$ and ACK are both in very small size, $N(L_{\max},r)\approx r/2$, which is the same as the OAR proposed by Sadeghi and Kanodia [1]. Therefore, equation (1) presents a unified solution to calculate the number of concatenated frames for better temporal fairness under various traffic characteristics and channel conditions. Under a constant link rate or frame size, equation (1) works in the same way as CM or OAR. In this paper, for simplicity, we name our proposed approach for concatenation number computation as *General Concatenation Mechanism* (*GCM*).

However, in OAR, it is assumed that frame size is quite large such that ACK overhead is negligible. However, as shown in equation (1), if L is small, e.g., 100 bytes, ACK imposes a considerable overhead to the performance of GCM. To eliminate this overhead further, we can first concatenate small frames to a big frame before it reaches $L_{\max}$ without an ACK transmitted for each small frame. Then, equation (1) is simplified to

$$N(L,r)=r\cdot L_{\max}/(2L) \qquad (2)$$

For comparison, we calculate N, the concatenation number in GCM, with $L_{\max}$ set to 1000 bytes. As shown in Figure 1, GCM combines the advantages of both CM and OAR. With frame size of L_{max}, GCM works in the same as OAR. However, with small frame sizes, the N values specified in GCM and CM are exactly the same for channel rate of 2Mbps and medium frame sizes (200 bytes and larger). However, under higher rates, GCM sends more frames for temporal fairness.



(a) GCM vs. CM (b) GCM vs. OAR

Figure 1. Comparison of GCM, CM, and OAR

C. Integration with link loss ratio

Basically, given a certain bit error rate (BER) and the maximum channel data rate, the higher the data rate is used, the higher the loss ratio will be. A higher loss ratio may yield lower throughput due to increased data retransmission and collision probability. Thus, the frame concatenation scheme should take into account the effect of loss ratio to improve the overall *successful* transmission per each channel access.

De Couto and Aguayo [15] proposed to measure ETX, the expected number of transmissions per frame as $1/(d_f\times d_r)$, where d_f and d_r are defined as the forward and reverse frame delivery ratio, respectively. ETX is obtained by source nodes sending probe broadcast frames periodically and calculating the probability of how many frames are correctly received. Therefore, given a specific link e with ETX_e, the expected total channel time consumed by a concatenation sequence is $RTS/2+SIFS+CTS/2+ETX_e\cdot N\cdot(SIFS+L/r)$. Then, following the same derivation and simplification in equation (2), we have

$$N(L,r)=\frac{1}{ETX_e}\cdot\frac{r\cdot L_{\max}}{2L} \qquad (3)$$

According to equation (3), a higher loss ratio leads to a bigger ETX_e and thus a smaller N. In other words, to make the frame concatenation more efficient and avoid excessive frame retransmissions, more frames should be concatenated over links that are more reliable.

D. Implementation

In CM, an efficient data format was designed to implement frame concatenation under the cases with and without RTS/CTS handshaking. On the other hand, in RBAR and OAR, RTS/CTS handshaking is necessary since RTS/CTS handshaking is used to decide the appropriate channel rate for data transmissions. In this paper, we adopt the same channel rate measurement scheme proposed in RBAR. Thus, RTS/CTS handshaking is always turned on in GCM. In RBAR, let M_1, M_2, …, M_N be the set of modulation schemes in increasing order of their rate, and θ_i be the *SNR* thresholds at which $BER(M_i)$=1E-5, the modulation scheme is chosen according to the following criteria:

M_1 if $SNR < \theta_1$
M_i if $\theta_i \leq \mathrm{SNR} < \theta_{i+1}$, $i = 1, \ldots, N-1$
M_N otherwise

However, GCM can also works with other rate adaptation schemes such as ARF [4], where the channel rate is determined based on the experience at the sender. In this case, the RTS/CTS handshaking become optional and the proposed schemes can be modified accordingly. Similar to OAR, the NAV value in the CTS is modified to reserve the channel time for the data transmission of all concatenated frames. Furthermore, with RTS/CTS enabled, the performance degradation due to loss of large frame transmission under high BER can be significantly mitigated.

The channel access sequence of GCM is illustrated in Figure 2. With the simple calculation in aforementioned equations, the dynamic frame concatenation can operate on-the-fly in real time. When the sender obtains the best channel rate from the receiver through CTS, it calculates an appropriate concatenation number according to the rate and frame size. Then, the sender keeps inserting the small frames to a big frame until its size reaches $L_{\max}$. The number of such frames is determined by equation (2) using the average size of the frames. In the case that there are no more frame available in the queue, default RBAR procedure is applied. Before the insertion of each small frame, a corresponding header (2 bytes) is included to indicate the size of each frame. Thus, upon the receipt of each frame, the receiver can easily identify each original frame by checking the header information. In summary, GCM operation is different from existing schemes such as CM, OAR, and APC and has the following advantages:

- Compared to OAR, it eliminates the overhead of ACK for small frames and thus can significantly improve the network throughput.
- Compared to CM and APC, it is more compatible with DCF standard and does no impose additional processing for frame loss at the receiver side.
- Forming a regular size frame is more loss resilient than forming super frame in APC. Super frames larger than the segmentation threshold will still be segmented by DCF operation with an individual ACK for each fragment, which is not efficient.

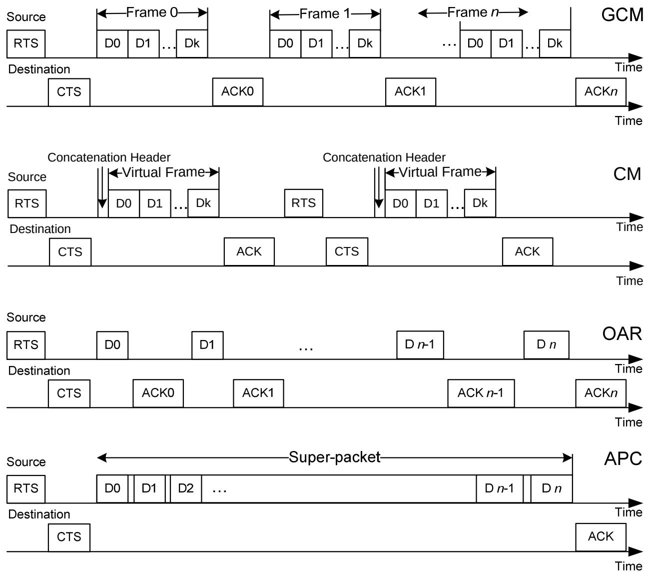


Figure 2. Illustration of various concatenation schemes

When frame size is not constant and the average frame size L is not available, we can calculate N according to $T_{L_{\max},2} = \sum_{i=1}^{N} L_i / r$, where $T_{Lmax,2}$ is calculated as $RTS/2+SIFS+CTS/2+SIFS+L_{\max}/2+SIFS+ACK/2$ and L_i is the actual length of the ith frame in the concatenation sequence. Similarly, equation (3) can also be modified under varying frame sizes according to $T_{L_{\max},2} = \sum_{i=1}^{N} t_i$, where t_i is the actual channel time being consumed for the transmission of ith frame in the concatenation sequence and can be measured by recording the time from the instance when the frame is removed from the MAC interface queue until the instance when either the frame is successfully transmitted or multiple retransmissions have failed. In this way, the temporal fairness is still maintained.

IV. ADAPTIVE QOS SCHEMES IN MULTI-RATE NETWORKS

Despite of the advantage of GCM on ensuring temporal fairness under various link rates and frame sizes, it is not sufficient to support high priority multimedia applications such as voice or video streaming. From the QoS aspect, even under relatively high channel rate and small frame sizes, it is undesirable for low priority data traffic to obtain equal or larger share of channel access time with respect to high priority voice and video traffic. To facilitate service differentiation, the number of concatenated frames should be differentiated such that high priority traffics receive more bandwidth share.

The problem statement of QoS in multi-rate ad hoc networks can be described as: *Given the network condition such as node distribution, channel rates, and prioritized network traffic such as voice, video, and data, how to provide a scheme such that (i) higher priority flows receive more* *channel time share (and preferably more bandwidth share) than lower priority flows; (ii) high priority flows such as voice and video get sufficient throughput to satisfy user requirements; (iii) temporal fairness is well maintained, i.e., under same priority level, different flows consume approximately same channel time share for various channel rates and frame sizes.*

To address this issue, we propose two schemes:

- Adaptive weighted fair frame concatenation mechanism (AWFCM) that adaptively set the concatenation number based on channel rate, frame size, flow weights, and network condition such that each flow gets a fair share of the channel bandwidth.
- Adaptive QoS frame concatenation mechanism (AQCM) that further provides soft QoS based on AWFCM. The core idea of AQCM is to adaptively tune the concatenation number of lower priority traffic and enforce admission control to protect existing high priority voice/video traffic.

A. Adaptive weighted fair frame concatenation (AWFCM)

Table I. Value of *N'* with three priorities

Voice	***φ* =1**			
Frame size (bytes)	**60**	**214**	**512**	**1024**
2 Mbps	10	4	2	1
5.5 Mbps	23	10	5	3
11 Mbps	35	18	9	5
Video	***φ* =0.5**			
Frame size (bytes)	**60**	**214**	**512**	**1024**
2 Mbps	5	2	1	1
5.5 Mbps	11	5	3	2
11 Mbps	17	9	5	3
Data	***φ* =0.25**			
Frame size (bytes)	**60**	**214**	**512**	**1024**
2 Mbps	3	1	1	1
5.5 Mbps	6	3	1	1
11 Mbps	9	5	2	1

To differentiate flows, we introduce the concept of flow weight. Let φ be the *flow weight* for a flow of priority ρ and L be the average frame size, concatenation number N' becomes

$$N'(\rho,L,r)=\lceil N(L,r)\cdot\varphi(\rho)/\varphi(\rho_{\max})\rceil \qquad (4)$$

Where $N(L,r)$ is calculated based on equation (1) or (3) for consideration of channel loss. Given the same frequency of channel access, N' corresponds to bandwidth share. Given that N reflects a fair share of bandwidth for the same traffic priority, N' reflects a weighted fair bandwidth provisioning for prioritized traffic. It is worth noting that φ can be configured according to various policies as long as it's reasonably set for differentiation. For instance, we can set the corresponding *flow weight* φ for voice, video, and data to 1, 0.5, and 0.25, respectively. With such configuration of φ, the values of N' for voice, video, and data under different channel rates and frame sizes are illustrated in Table I.

Setting fixed flow weights may have significant impact on lower priority flows under various network conditions. When there is no voice/video traffic or when the network load is light, it would be a waste of channel resource to limit the bandwidth share of low priority data traffic with a smaller concatenation number. Thus, based on the average frame size and link rate, and the *overheard weights of other flows*, the sender decides (on a per channel access basis) an appropriate frame concatenation number as

$$N'(\rho,S,r)=\lceil x\cdot N(L,r)\rceil \qquad (5)$$

where x is defined as

$$x=\begin{cases}1 & \text{if}\,\varphi(\rho)=\varphi(\rho_{\max})\\ \max(\varphi(\rho)/\varphi(\rho_{\max}),1-\alpha) & \textit{otherwise}\end{cases} \qquad (6)$$

and $\varphi(\rho_{\max})$ is the maximum flow weight overheard from neighboring nodes. Thus, if a flow has the highest weight, then it will use the values calculated by equation (1) so that temporal fairness is maximized. Otherwise, the flow uses a frame concatenation number appropriate to its flow weight and network condition. If the network load is very light, a large N will still be used.

This algorithm is quite adaptive. Initially, when traffic load is very light, every station can use large concatenation number relatively to the one calculated from equation (1). As the traffic load increases, this number reduces gradually until it equals to the lower bound calculated from equation (4). Most of the time, stations choose some values in between of the two bounds. On the other hand, with the starting and finishing of high priority traffic, the maximum flow weight also changes dynamically and stations can recalculate their numbers based on equation (5). For example, when only video and data coexist, video flows will choose flow weight of 1 and use large concatenation numbers for frame transmissions.

It should be noted that the proposed adaptation does not cause oscillation and is fully distributed. The corresponding concatenation number from equation (4) and (5) is for *each channel access* only. In the next channel access, the corresponding channel condition, maximum flow weight, and frame size may change and a different concatenation number may be obtained accordingly. Furthermore, the channel busy ratio is estimated every second and the flow weight information is piggybacked in RTS and CTS frames. Whenever a RTS/CTS handshaking is overheard, the flow weight information can be retrieved and $\varphi(\rho_{\max})$ is updated if necessary. Thus, the accuracy of the channel condition information can be ensured. For flow weight updates, a station will maintain a variable indicating the maximum value of its own flow weights and all weights overheard from neighbors. This maximum flow weight is reset to the local flow weights periodically to reflect the fact that the highest priority flow may have already finished.

B. Adaptive QoS frame concatenation (AQCM)

The primary limitation of AWFCM is that it does not provide sufficient QoS support. Under heavy traffic load, it is highly possible that there is no bandwidth available for new voice/video flows. In this case, admission control is mandatory to avoid performance degradation of existing voice/video traffic. On the other hand, to further protect voice/video traffic, it is desirable that data traffic reduces their bandwidth share by reducing the number of concatenated frames. Therefore, based on AWFCM, we propose an adaptive QoS frame concatenation mechanism to ensure QoS support for high priority flows.

The idea of AQCM is described as follows. Initially, every station continuously listens to the channel and thus can overhear RTS/CTS/DATA/ACK of their neighbors and thus obtain information in the frame headers. In this algorithm, two fields, *flow weight* and *is_satisfied*, are included in RTS and CTS frame headers. The flag *is_satisfied* indicates whether or not the flow rate requirement is met. Based on this information, a station makes decision on whether or not it should start a new traffic as follows:

- If any of the voice flows is not satisfied, reject any new voice/video flows at this station;
- Otherwise, if any of the video flows is not satisfied, reject any new video flows at this station. However, we can still admit new voice flows since voice flows usually require much less bandwidth than video flows.

Of course, it is difficult to know in advance if the newly accepted voice/video flow can be supported or not. We can adopt a simple method to resolve it. After the admission, the source station of the flow monitors RTS/CTS of other stations for a short period. If at least one of the flows in the same category is not satisfied, the station drops the newly admitted flow to avoid performance effect to existing flows.

To set the *is_satisfied* flag, each station currently sending voice/video flows measures its perceived throughput by periodically counting the number of bytes successfully transmitted. With the RTS/CTS/DATA/ACK handshaking, the source can easily know if a frame is delivered successfully without sending probing frames. Then, before sending RTS/CTS frames, it sets *is_satisfied* flag to 1 if the throughput is at least 95% of the flow rate requirement and 0 otherwise. Note that using 95% can avoid a new flow is incorrectly rejected due to slight throughput fluctuation.

Another enhancement in AQCM is the control of data traffic through dynamic flow weight adjustment. Each source station of data traffic continuously monitors RTS/CTS of other stations. If at least one of the voice/video flows is not satisfied, the station reduces the corresponding flow weight by half to yield bandwidth share to existing high priority flows. In the worst case, only one frame is transmitted per each channel access. If all voice/video flows experience satisfactory performance (mostly likely due to the completion of other high priority flows), the flow weight is still calculated by equation (5) so that more throughput can be received.

C. *Determination of flow weights*

Flow weight has been used for service differentiation in many scheduling protocols such as Self Clocked Fair Queuing (SCFQ [13]) and Distributed Fair Scheduling (DFS [14]). Using flow weight helps provides a fair bandwidth allocation for different flows. In AWFCM and AQCM, we adopt this concept for different traffic types and also provide suggested values for flow weights. Basically, choosing an appropriate flow weight depends on the network configuration. However, it is still helpful to suggest some guidance for flow weights.

Let us assume that there are two flows with flow weight φ_1 and φ_2, respectively and $\varphi_1 > \varphi_2$. Let us define β as the relative weight between these two flows, $\beta = \varphi_2/\varphi_1$. Given the same network configuration such as minimum contention windows ($CW_{\min}$) and interframe spaces (*IFS*), under the assumption of the same channel access probability, it is obvious that the data being transmitted by stations for these two flows are approximately $N(L,r)$ and $\beta{\cdot}N(L,r)$, respectively. Let us assume that each station accesses channel M time within the same sufficiently long period, in the worst case, flow 1 transmits $M{\cdot}L_{\max}$ bytes under 2 Mbps channel rate and largest data frames. In comparison, flow 2 transmits $M{\cdot}\beta{\cdot}N(L,r){\cdot}L$ bytes during the same period. Then, from bandwidth provisioning aspect, three criteria follow:

1) Prioritized bandwidth sharing (PBS): With this criterion, under any situation, flow 1 (with higher weight) should be allocated at least the same bandwidth as flow 2. Thus, $\mathrm{M}{\cdot}\mathrm{L}_{\max} \geq \mathrm{M}{\cdot}\beta{\cdot}\mathrm{N(L,r)}{\cdot}\mathrm{L}$. , which is equivalent to

$$\beta \leq L_{\max} / (L \cdot N(L,r)) \tag{7}$$

If L is not very small, then *SIFS* and *ACK* overheads can be neglected and thus from equation (1) we have that $N(L,r) = r{\cdot}\ L_{max}/(2L)$ and equation (7) can be approximated as $\beta \leq 2/r$. In the best case for flow 2, r=11 Mbps. So, β should be no more than 0.18 for prioritized bandwidth sharing. In this case, we can set flow weights of voice, video, data flows to be 1, 0.18, and 0.03, respectively.

2) Average prioritized bandwidth sharing (APBS): With this criterion, the worst case bandwidth allocation for flow 1 is no less than the average bandwidth share of flow 2. Thus, $M \cdot L_{\max} \geq \sum_{i=1}^{3} M \cdot \beta \cdot N(L, r_i) \cdot L / 3$, which is equivalent to

$$\beta \leq 3L_{\max} / (L \cdot \textstyle\sum_{i=1}^{3} N(L, r_i)) \tag{8}$$

Similarly to equation (7), equation (9) can be approximated as $\beta \leq 6 / \sum_{i=1}^{3} r_i$. For channel rates of 2, 5.5, and 11 Mbps, β should be no more than 0.33 for average prioritized bandwidth sharing. In this case, we can set flow weights of voice, video, data flows to be 1, 0.33, and 0.11, respectively.

3) Average weighted bandwidth sharing (AWBS): With this criterion, the average weighted bandwidth shares of flow 1 and flow 2 are predefined. Thus, we have that

$$\frac{1}{3}\sum_{i=1}^{3} M \cdot N(L_{\max}, r_i) \cdot L_{\max} \geq \gamma \cdot \frac{1}{3}\sum_{i=1}^{3} M \cdot \beta \cdot N(L, r_i) \cdot L \tag{9}$$

Where γ is the ratio of the predefined weighted bandwidth share between flow 1 and flow 2. Thus, we have

$$\beta \leq \frac{1}{\gamma} \cdot \frac{L_{\max}}{L} \cdot \frac{\sum_{i=1}^{3} N(L_{\max}, r_i)}{\sum_{i=1}^{3} N(L, r_i)} \tag{10}$$

Similarly to equation (7) and (9), equation (10) can be approximated as $\beta \leq 1/\gamma$. If we assume a higher priority flow has twice overall bandwidth share of the next level priority flow, we can set flow weights of voice, video, data flows to be 1, 0.5, and 0.25, respectively with γ being 2.

For comparison, Table II lists the suggested flow weights for voice, video, and data, for different bandwidth provisioning criteria, respectively. From this setting, we feel that PBS sacrifices video and data flows heavily to ensure the priority of voice data. On the other hand, APBS and AWBS make a better bandwidth provisioning for all three traffic types. Specifically, AWBS seems to be more reasonable due to the relatively larger weights for video and voice. However, as discussed earlier in equation (6), under non-saturation status, given a relatively small value of β, it is the channel condition, not the relative flow weight that dominates the concatenation number.

Table II. Suggested maximum flow weights

Traffic	Voice	Video	Data
PBS	1	0.18	0.03
APBS	1	0.33	0.11
***AWBS* (γ=2)**	1	0.5	0.25

D. Discussion

An alternative approach of enforcing admission control is to calculate available bandwidth based on the effective link capacity and channel busy ratio [12]. However, with differentiated frame concatenation, it is quite difficult to know exactly how much bandwidth is achievable for flows of different categories. Thus, we believe that measuring the throughput is reasonable and effective for supporting soft QoS. In addition, to further improve per flow performance, we can also measure the frame delays as additional QoS criteria. More enhancements are omitted due lack of space.

For senders transmitting flows of mixed priorities, AWFCM and AQCM can be easily modified as follows. When a sender gets access to the channel, it allocates a channel time share proportional to the flow weight for each traffic type. Then, a corresponding concatenation number for each traffic type is determined and the sender simply concatenates the same number of frames until the length threshold is reached. Furthermore, we believe that the proposed flow weight based concatenation scheme can be integrated with the IEEE 802.11e EDCF for sufficient service differentiation in multi-rate wireless ad hoc networks. Although EDCF can effectively support high priority traffic by varying parameters such as IFS, minimum contention window ($CW_{\min}$), and TXOP, it does not consider the multi-rate capability. In this case, if GCM is used (for desirable capacity improvement), it is still possible that a low priority flow may receive more throughput than a high priority flow if it is transmitted over higher rate links. By differentiating frame concatenation number, EDCF can achieve much higher network capacity while maintaining sufficient service differentiation. These issues will be our future work.

V. PERFORMANCE EVALUATION

GCM, AWFCM, and AQCM are evaluated by extensive simulations with *ns*-2 network simulator. The proposed protocols are implemented by modifying the OAR software [16] provided by Rice Network Group. In all simulations, transmission range is set to 250 meters and the link rates are 11, 5.5, and 2 Mbps for nodes within transmission range of 80, 120, and 250 meters, respectively. 50 nodes are randomly distributed in the area and the size for single and multiple hop networks are 120×120 m^2 and 500×500 m^2, respectively.

A. Performance of GCM

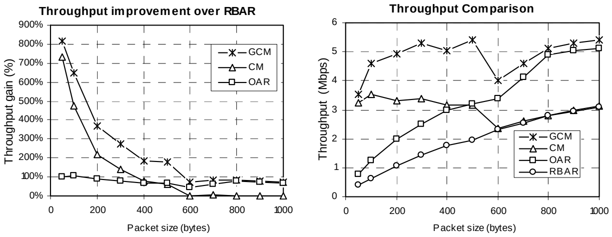


Figure 3. Comparison of total throughput by various schemes without channel loss. Left: Throughput improvement over RBAR; Right: Comparison of throughput achieved by GCM, CM, OAR, and RBAR.

We evaluate the performance of GCM, CM, and OAR under saturation status where 25 source stations send CBR traffic to randomly chosen destinations at a constant time interval of 2 ms. Packet sizes vary from 50 bytes to 1000 bytes but remain constant for each simulation. Figure 3 compares the throughput improvement over RBAR, and total throughput by GCM, CM, OAR, and RBAR. It can be seen that GCM outperforms CM and OAR in all frame sizes. Particularly, with small frame sizes, GCM achieves around two to four times throughput OAR. With large frame sizes, GCM achieves around 80% more throughput than CM. This improvement is due to the fact that GCM combines the advantages of both CM and GCM. At very small frame sizes such as 50 and 100 bytes, CM performs close to GCM. This is due to the fact GCM may have to concatenate a huge number of frames under higher channel rates but there are not always that many frames queued at the MAC interface. Furthermore, CM performs much better than OAR under small frame sizes since it allows more frames to be concatenated. However, when frame sizes are greater than 500 bytes, CM cannot concatenate more frames and performs in the same way as RBAR does.

Figure 4 compares the total throughput achieved by GCM, CM, OAR, and RBAR with varying frame sizes randomly distributed within certain ranges. It is obvious that GCM outperforms CM and OAR significantly even though the average frame sizes are 550 and 600 bytes, respectively. Note that CM performs much better than RBAR under both scenarios since for CM it is highly possible to concatenate more small frames with random packet sizes.

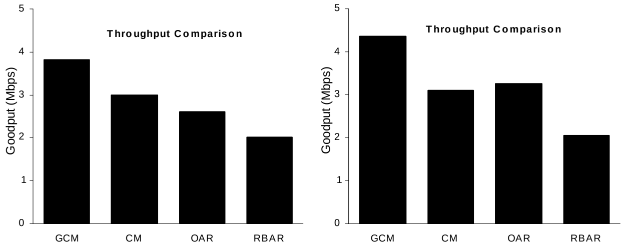


Figure 4. Total throughput with varying frame sizes. Left: frame sizes from 50 to 1000 bytes; Right: frame sizes from 200 to 1000 bytes.

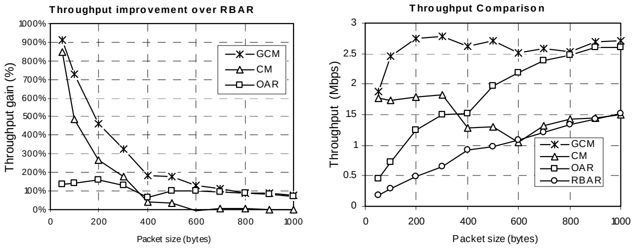


Figure 5. Comparison of total throughput by various schemes with channel loss. Left: Throughput improvement over RBAR; Right: Comparison of throughput achieved by GCM, CM, OAR, and RBAR.

We also compare the performance of GCM when channel loss is considered. In this simulation, the loss ratio of each wireless link between a source/destination pair is randomly selected uniformly within the range from 0 to 0.3. Thus, the average loss ratio is 0.15 among all links. Then, the loss ratio of forward and reverse links are piggybacked in the RTS/CTS frames such that ETX and corresponding concatenation number of a link can be calculated through equation (3). Figure 5 compares the throughput improvement over RBAR, and total throughput by GCM, CM, OAR, and RBAR. It can be seen that GCM outperforms CM and OAR in all frame sizes. Particularly, with small frame sizes (smaller than 400 bytes), GCM achieves around 2-4 times throughput of OAR.

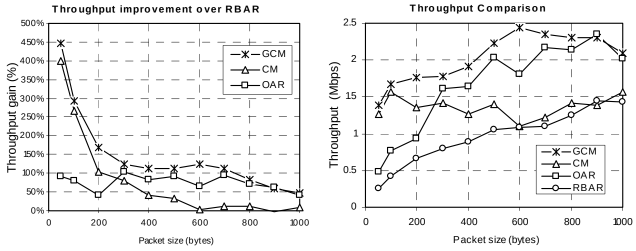


Figure 6. Comparison of total throughput by various schemes in multi-hop ad hoc networks. Left: Throughput improvement over RBAR; Right: Comparison of throughput achieved by GCM, CM, OAR, and RBAR.

Finally, GCM is compared with other schemes in a multi-hop ad hoc network. Ten stations send CBR traffic to randomly chosen destinations at a constant time interval of 2 ms. Figure 6 shows the corresponding simulation results. It can be seen that GCM outperforms OAR and CM in multi-hop ad hoc networks under various packet sizes. Of course, due to the multi-hop interference among links on the same flow, the throughput gain in multi-hop ad hoc networks is smaller than the gain in single-hop ad hoc networks (shown in Figure 3 and Figure 5).

B. *Performance of AWFCM*

Three different traffic types are involved in all simulations. Each voice stream is a CBR flow with rate of 64Kbps, generated by a constant inter-arrival time of 25ms and a fixed payload size of 200 bytes. Each video stream is traffic-shaped CBR flow with rate of 320Kbps, generated by a constant inter-arrival time of 20ms and a fixed payload size of 800 bytes. We also introduce background data traffic, which is a UDP flow with payload size of 1000 bytes and frame arrival interval of 5 ms. Flows are initiated every two seconds to gradually increase the traffic load in the network. The ratio of number of voice, video, and data flows is 2:2:1. All flows last for 50 seconds.

Table III. THROUGHPUT COMPARISON OF AWFCM

Throughput	**Voice (Kbps)**	**Video (Kbps)**	**Data (Kbps)**	**Overall (Mbps)**
GCM	55.29	263.47	608.96	3.34
AWFCM (PBS)	59.36	253.00	317.40	2.54
AWFCM (APBS)	58.51	271.31	320.15	2.63
AWFCM (AWBS)	57.06	275.36	428.59	2.95

Table III shows the performance of GCM and WFCM with different weight selection schemes. We can see that since GCM uses flow weight of 1 for all traffic types (thus the biggest concatenation numbers), it achieves highest overall throughput. However, it sacrifices the performance of highest priority flows, especially the voice traffic. Among the three flow weight selection schemes, PBS and APBS enforce very strict differentiation and thus provides highest throughput for voice traffic. However, it suppresses video and data traffic significantly and thus the overall throughput is quite low. In contrast, AWBS seems to have a better compromise between fairness and network capacity with higher flow weight assigned to video and data traffic. So, we decide to use AWBS in the rest of the section. This simulation results also suggest that optimal flow weights should be investigated.

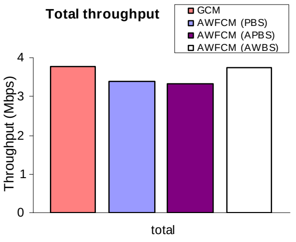


Figure 7. AWBS performance: comparison of total throughputs.

To show the effect of flow weight adaptation in AWFCM under dynamic traffic characteristics, we use TCP stream as the data traffic in the previous simulation. The TCP flow has the payload size of 1000 bytes and continues until the end of the simulation. Figure 7 compares the total throughputs with various schemes. We can see that in this scenario, AWBS receives very similar overall throughputs with different flow weight determination methods. The reason for this is that TCP streams all use high flow weight after voice/video flows finish at around 96 second.

Figure 8 depicts the received throughput by the first TCP flow. It can be seen that initially, traffic load is low and even data traffic uses quite large concatenation number and thus receives good throughput. Then, due to severe channel contention from 20 to 80 seconds, TCP traffic is significantly suppressed to accommodate voice/video flows. After 96 seconds, voice and video flows finish and TCP flows use flow weight of 1 and thus receive much better performance.

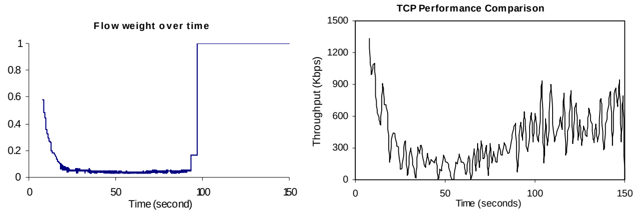


Figure 8. AWBS performance under dynamic traffic characeteristics. Left: TCP flow weight over time; Right: TCP throughput over time.

C. Performance of AQCM

Table IV. THROUGHPUT COMPARISON OF AQCM

Throughput	**Voice (Kbps)**	**Video (Kbps)**	**Data (Kbps)**	**Overall (Mbps)**
GCM	55.29	263.47	608.96	3.34
AQCM (AWBS)	61.93	308.64	784.75	2.95

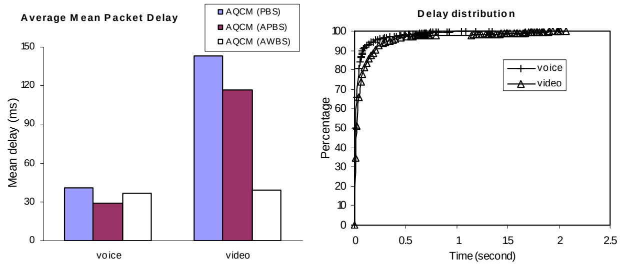


Figure 9. Packet delay with AQCM. Left: averaged mean delays; Right: delay distribution of selected voice and video flows.

The same scenario with UDP as data traffic is used for AQCM performance evaluation. Table IV compares the performance of AQCM (AWBS) with GCM. It can be seen that admitted voice and video flows received 96% and 91% throughput on average, respectively. In contrast, this percentage is only 86% and 81% for voice and video flows with GCM. Also, with the rejection of some high priority flows, the UDP traffics receive much higher performance with AQCM. Figure 9 shows the averaged mean packet delays and delay distribution of voice and video flows, respectively. It can be seen that on average, with various flow weight determination schemes, voice and video flows experiences packet delay of 30-40ms and 40-140ms on average, respectively. It should be noted that AWBS perceives quite low delay for video flows due to the fact that it allocates more bandwidth share to data traffic and thus admits less video flows. From Figure 9, we can also see that for voice and video flows, the percentages of packets experiencing delay lower than 100ms are 92% and 81% respectively. For soft QoS, this is quite desirable.

VI. CONCLUSTION

Capacity and quality of service in multi-rate networks are two critical issues for supporting bandwidth constrained multimedia applications. In this paper, we investigated the temporal fairness from a more comprehensive aspect and proposed GCM, a general frame concatenation mechanism. Compared with existing schemes such as CM and OAR, GCM achieves better temporal fairness and thus higher throughput under various frame sizes and channel rates. Based on GCM, we designed two adaptive algorithms, AWFCM and AQCM, to adjust frame concatenation number according to various factors such as channel rate, frame size, flow weights, and network traffic load. The proposed adaptive algorithms help achieve better temporal fairness and desirable service differentiation. With these schemes, performance of multimedia applications can be further improved in the wireless ad-hoc networks. In the future, we will investigate interesting issues as extensions of GCM such as handling traffic of mixed priorities and integration with EDCF.

REFERENCES

[1] B. Sadeghi, *et. al*, ``Opportunistic Media Access for Multirate Ad Hoc Networks,” in Proc. of ACM MobiCom'02.

[2] Y. Xiao, "IEEE 802.11 Performance Enhancement via Concatenation and Piggyback Mechanisms," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 5, Sep. 2005, pp. 2182- 2192.

[3] IEEE, “IEEE std 802.11 – wireless LAN medium access control (MAC) and physical layer (PHY) specification”, 1997.

[4] A. Kamerman and L. Monteban, “WaveLAN II: A high-performance wireless LAN for the unlicensed band, ” in *Bell Labs Technical Journal*, pages 118-133, Summer 1997.

[5] G. Holland, *et. al*, “A rate-adaptive MAC protocol for multi-hop wireless networks”. In Proceedings of ACM MOBICOM'01.

[6] J. Wang, et.al, "OMAR: Utilizing Multiuser Diversity in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1764-1779, Dec. 2006.

[7] H. Zhai and Y. Fang, "A Distributed Adaptive Packet Concatenation Scheme for Sensor and Ad Hoc Networks," in Proc. of IEEE MilCom'05.

[8] H. Zhai, *et. al*, "A Call Admission and Rate Control Scheme for Multimedia Support over IEEE 802.11 Wireless LANs," *ACM Wireless Networks*, vol. 12, no.4, pp. 451-463, August 2006.

[9] E. Kim and Y.-J. Suh, “ATXOP: An Adaptive TXOP Based on the Data Rate to Guarantee Fairness for IEEE 802.11e Wireless LANs,” in Proc. of IEEE VTC 2004-Fall.

[10] Y.-Y. Edwin Tan, *et. al*, “Quality-of-Service (QoS) Framework for Multi-rate Wireless Ad-hoc Network (MWAN)”, in Proc. of IWMAN 2005.

[11] S. Blake, *et. al*, “An Architecture for Differentiated Service”, RFC 2475, December 1998.

[12] M. Li, B. Prabhakaran, "On Supporting Reliable QoS in Multi-hop Multi-rate Mobile Ad Hoc Networks", in Proc. of IEEE WoNGeN'05.

[13] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications," in Proceedings of IEEE INFOCOM,1994.

[14] N. H. Vaidya, *et. al*, “Distributed Fair Scheduling in Wireless LAN”. In Proceedings of ACM MOBICOM 2000.

[15] D. De Couto, et.al, “High-throughput path metric for multi-hop wireless routing”. In ACM MOBICOM 2003.

[16] OAR implementation, Rice network group. Available at: http://www-ece.rice.edu/networks/software/OAR/OAR.html.