Dynamic priority re-allocation scheme for quality of service in IEEE 802.11e WLANs

Ming Li and B. Prabhakaran*
Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083

ABSTRACT

Enhanced DCF (EDCF) is currently under review as the new standard for quality of service in IEEE 802.11 wireless LANs. In EDCF, per-flow differentiation is achieved by maintaining separate queues for different traffic categories (TCs). However, due to its static QoS parameter setting, EDCF does not perform adequately under high traffic load [3]. We present an extended performance model of EDCF and possible conditions for network getting overloaded. With this extended model, we show that the overall throughput of a network can be improved by changing the distribution of the number of active stations† over a set of TCs. Hence, we propose to dynamically re-allocate flow priorities evenly in order to maintain high system performance while providing QoS guarantee for individual real-time flows. Our scheme has several interesting features: (1) Performance of EDCF is improved; (2) Low priority flows are not starved under high traffic load; (3) Misuse of priority‡ can be easily handled. Simulations are conducted for both infrastructure-based and Ad-hoc models. Results show that dynamic priority re-allocation does not decrease throughput of real-time flows under low to medium loads, while considerable improvement over EDCF is obtained even under high loads, making it easy to support multimedia applications.

Keywords: IEEE 802.11e, quality of service, WLAN.

1. INTRODUCTION

EDCF is proposed as the upcoming standard for supporting service differentiation in IEEE 802.11e Wireless LANs. For priority-based scheduling algorithms such as EDCF, priorities are used for backoff interval differentiation such that flows of higher service classes have more chances to access channel than flows of lower service classes. When the backoff interval is smaller, a flow attempts data transmission with higher rate and the packet delay will be smaller. Ideally, different flows should choose different backoff intervals. If all flows are of very high priorities, they may have higher probability of choosing same backoff intervals and thus collide more frequently. This uncontrolled collision leads to significant performance degradation of all flows involved, especially when the network traffic is high. Also, in this case, all flows may experience large delays because of the same chances of channel access. Thus, it is not appropriate to allow flows to choose whatever priority they prefer. Instead, priority re-allocation is needed to minimize the collision rate and guarantee delay constraints for real-time flows.

Several quality of service schemes have been proposed to improve the performance of EDCF [3, 7, 9]. Most of the work focuses on the refinement of the calculation of backoff interval and contention window to minimize the collision rate. However, if all flows select the same priority, their Arbitrary Interframe Space (*AIFS*), Minimum Contention Window (CW_{min}) and Maximum Contention Window (CW_{max}) are essentially the same and good service differentiation is not possible when traffic load is high. Hence, refinement of AIFS and/or the contention window is less effective in EDCF. Furthermore, changing the method of contention window calculation modifies the existing IEEE 802.11e standard. It would be better to provide a scheme that can be incorporated with the existing standard easily.

In this paper, we present an extended model for overall throughput analysis of EDCF (based on an analytical model [9]). This extended model shows that an overall network throughput improvement can be realized by distributing the number of active stations in different sets of traffic categories or classes. Hence, we propose to dynamically re-allocate flow priorities evenly. This priority re-allocation approach is efficient in the sense that it can also maintain good service

* praba@utdallas.edu, phone 1 972 883-4680; fax 1 972 883-2349
† An active station is one that has a pending packet to be sent.
‡ Misuse of priority means that a flow requests much higher priority than necessary.

differentiation for individual flows. We conducted extensive simulations for both infrastructure-based and Ad-hoc models. These simulations show that dynamic priority re-allocation provides considerable improvement even under high traffic load when compared to EDCF. The improvement was not only in terms of satisfying the throughput requirements of real-time or high-priority flows. The mean delays of most real-time flows are also under 100ms. Hence, the dynamic priority re-allocation scheme can help a wireless network to support multimedia applications in a better manner. Also, high traffic load of best-effort or low-priority flows does not seriously affect the performance of real-time flows. Simulation results show that best-effort flows are not starved under high traffic loads either.

The remaining part is organized as follows. Sec. 2. reviews related work. Sec 3. presents an extended model for EDCF performance. Sec. 4. describes the proposed priority re-allocation scheme for IEEE 802.11e WLANs. Sec. 5. discusses simulation results for validation. Sec. 6. discusses the possible applications of the proposed scheme and Sec. 7. concludes this paper with future work.

2. RELATED WORK

Enhanced Distributed Coordinator Function (EDCF) [1] is the newly proposed medium access mechanism by task group E of IEEE 802.11 working group. In EDCF, differentiated DCF access is provided to the wireless medium for prioritized traffic categories (TCs). An EDCF station can implement at most 8 prioritized output queues, one for each traffic category. Values for *AIFS*, CW_{min}, and CW_{max} are set on per-TC basis. Each application sends its preferred traffic category by setting an appropriate priority for priority bit of the data frame. By specifying smaller CW_{min} and *AIFS* values for higher priority queues, delay and throughput of high priority flows can be ensured while minimal service is offered to lowest priority queues. Virtual collisions between competing queues within a station are resolved by granting the Transmission Opportunity (TXOP) to the highest-priority queue involved in the collision. A virtual collision happens when the backoff intervals of two or more queues within one station counts to zero at the same time. EDCF provides good service differentiation for flows with different priorities. However, it uses static QoS parameter setting and it may starve low priority flows in high traffic load [2].

Aad and Ni [6] suggest to slowly decrease *CW*s after each successful packet transmission to reduce collision rate and retransmissions. Then, Romdhani and Ni [3] propose Adaptive EDCF (A-EDCF) to address the issue of static QoS parameter setting of EDCF. In A-EDCF, the contention window of each flow is adapted according to the estimated collision rate in attempt to improve the performance of EDCF under heavy traffic loads. However, A-EDCF is not compatible with EDCF standard. Chen and Lin [4] propose Priority-based Contention Control (PCC), an admission control similar to PCF. In PCC, contention window is dynamically changed in the same way as DFS [13]. A desirable feature of PCC is that it reduces the overhead to minimum by embedding flow information in MAC frames. Benveniste [7] makes further enhancement on EDCF in Tiered Contention Multiple Access (TCMA) and Aad and Castelluccia [5] discuss several per-flow differentiation methods. Kanodia and Li [10] propose distributed priority scheduling for end-to-end quality of service guarantee in multi-hop Ad-hoc networks. In their work, a deadline based priority backoff policy is devised and the priority information is piggybacked with RTS, CTS, and ACK to minimize the extra overhead. Then, to satisfy the end-to-end QoS requirement of flows, downstream stations adjust the priority levels of packets based on their performance upstream by a technique known as multi-hop coordination. However, the MAC access mechanism proposed is different from the IEEE 802.11 standard.

Zhu and Chlamtac [15] extend a Markov model to EDCF and analyze the performance. Zhao and Zhang [9] present a class-based analytical model for multiple QoS supports in IEEE 802.11. In their approach, an expression of total throughput *G* is formulated as a constrained optimization problem with constraints on bandwidth requirements of each class *l*. This approach can obtain the optimal contention windows, but the constrained optimization problem has to be solved every time the number of active stations in the system changes. Thus, it is not computationally efficient, especially when the number of active stations is large.

3. ANALYSIS OF EDCF PERFORMANCE

The notations involved in our analysis are listed as follows:

G	overall throughput;
$N^{(l)}$	active stations of service class *l*;
$CW^{(l)}$	initial contention window of service class *l*;
$CW_{exp}{}^{(l)}$	expectation of contention window of service class *l*;

$C^{(l)}$	coefficient of average packet size of service class l;
$\tau^{(l)}$	probability of a class l station transmits at a randomly chosen time slot;
$P_s^{(l)}$	probability that a class l station succeeds;
P_c	probability of collision;
P_{tr}	probability that at least one station transmits;
$P_{coll}^{(l)}$	probability that at least one of other stations transmit.
$T_s^{(l)}$	length of a successful transmission of class l;
T_c	length of collision event;
$msg^{(l)}$	average packet size of class l;
L	average packet size of all flows;
σ	duration of slot time;
S	total probability of successful transmission for stations of all service classes;
X	effective traffic load per time slot, defined as $\sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}}$

3.1 Preliminary model for EDCF

Our analysis is based on an analytic model [9] for multiple QoS support in IEEE 802.11. Basic formulae in this model are introduced briefly as follows. Let there be totally M service classes. The service class of a flow is represented by priority of the flow, l, from 0 to $M-1$, and a class l has $N^{(l)}$ active stations. Overall throughput of the system for EDCF with CW differentiation is given as

$$G = \frac{\sum_{l=0}^{M-1}(P_s^{(l)} msg^{(l)})}{P_c T_c + \sum_{l=0}^{M-1}(P_s^{(l)} T_s^{(l)}) + \frac{1-P_{tr}}{P_{tr}}\sigma} \tag{1}$$

Assume that for any two service classes l_1 and l_2, l_1 is of higher service class than l_2 if $l_1 > l_2$. For a transmitting station of service class l, the probability that at least one of other stations transmits is

$$P_{coll}^{(l)} = 1-(1-\tau^{(l)})^{N^{(l)}-1} \prod_{i \neq l}(1-\tau^{(i)})^{N^{(i)}} \tag{2}$$

Let m be the retransmission limit after packet collision occurs, $\tau^{(l)}$ is calculated by the following equation.

$$\tau^{(l)} = \frac{2}{1+CW^{(l)} + P_{busy}^{(l)} CW^{(l)} \sum_{i=0}^{m-1}(2P_{busy}^{(l)})^i}, \quad \text{where} \quad P_{busy}^{(l)} = 1-(1-\tau^{(l)})^{N^{(l)}-1} \prod_{i \neq l}(1-\tau^{(i)})^{N^{(i)}}$$

3.2 Extended model for overall throughput of EDCF

From (1), it is not clear how overall throughput G changes with the number of active stations and contention windows. In this section, we will extend the model of overall throughput and then show that by changing the number of active stations of each distinct service class, it is possible to maintain high overall throughput while providing quality of service guarantee for individual flows.

***Lemma 1*:** If average flow packet sizes of all service classes are the same (we will relax this assumption later), then G only depends on S, the total probability of successful transmission of stations across all service classes.

***Proof*:** Let $msg^{(l)}$ is a constant for any service class l and denote L as the average packet size of all flows, then $msg^{(l)}=L$. Consider that the overhead for sending packets are relatively small, $T_s^{(l)}$ is approximately proportional to $msg^{(l)}$. Thus, $T_s^{(l)}$ is also a constant, denoted as T_s.

When $N^{(l)}$ (the number of active stations) is large, at any time there are always at least one station attempting transmission. In this case, since $\tau^{(l)} = \frac{2}{1+CW_{\exp}^{(l)}} < 1$, $P_{tr} = 1-\prod_{l=0}^{M-1}(1-\tau^{(l)})^{N^{(l)}} \approx 1$.

Let us define $S = \sum_{l=0}^{M-1} P_s^{(l)}$, denoting the total probability of successful transmission for stations of all service classes. Then $P_c = P_{tr} - \sum_{l=0}^{M-1} P_s^{(l)} \approx 1 - \sum_{l=0}^{M-1} P_s^{(l)} = 1 - S$, and (1) can be simplified as

$$G = \frac{L \cdot \sum_{l=0}^{M-1} P_s^{(l)}}{P_c T_c + T_s \cdot \sum_{l=0}^{M-1} P_s^{(l)}} = \frac{LS}{(1-S)T_c + T_s S} = \frac{L}{T_s + T_c \cdot \left(\frac{1}{S} - 1\right)} \tag{3}$$

Since L, T_s, and T_c are constant, total throughput G is solely decided by S.□

Lemma 1 shows that if all service classes have the same average packet size, then the overall throughput only depends on the total probability of successful transmission of all flows. If S is larger, G is also larger, which gives us the idea of focusing on the change of S for the analysis of overall throughput.

***Lemma 2*:** The total probability of successful transmissions (S) increases iff the effective load per time slot is less than or equal to 1, i.e., $X = \sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \le 1$.

***Proof*:** Probability that a class l station succeeds is calculated as

$$P_s^{(l)} = N^{(l)} \tau^{(l)} (1-\tau^{(l)})^{N^{(l)}-1} \prod_{i \ne l} (1-\tau^{(i)})^{N^{(i)}} = N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \prod_{i=0}^{M-1} (1-\tau^{(i)})^{N^{(i)}}$$

Then, the total probability of successful transmissions

$$S = \sum_{l=0}^{M-1} P_s^{(l)} = \sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \prod_{i=0}^{M-1} (1-\tau^{(i)})^{N^{(i)}} = \prod_{l=0}^{M-1} (1-\tau^{(l)})^{N^{(l)}} \cdot \sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \tag{4}$$

Now, let us consider how S changes dynamically with the arrival of new flows. Suppose we have $N = \sum_{l=0}^{M-1} N^{(l)}$ active flows and a new flow attempts transmission with priority i, $0 \le i \le M-1$. Assume that adding a new flow does not significantly change the contention windows of existing flows because of the increasing collision rate. Then it can be easily shown that the change of S can be expressed as

$$\begin{aligned}
\Delta S &= S_{new} - S \\
&= (1-\tau^{(i)}) \prod_{l=0}^{M-1} (1-\tau^{(l)})^{N^{(l)}} \cdot \left(\sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} + \frac{\tau^{(i)}}{1-\tau^{(i)}} \right) - \prod_{l=0}^{M-1} (1-\tau^{(l)})^{N^{(l)}} \cdot \sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \\
&= \tau^{(i)} \prod_{l=0}^{M-1} (1-\tau^{(l)})^{N^{(l)}} \cdot \left(1 - \sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \right)
\end{aligned} \tag{5}$$

Let us define $X = \sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}}$, as the effective traffic load per time slot, then $\triangle S \ge 0$ iff $X \ge 1$.□

The result of Lemma 2 is interesting in the sense that it provides the possible condition under which a network gets overloaded. X is actually the effective traffic load in any randomly selected time slot in term of the number of transmitting stations. Therefore, if $X \le 1$ in each time slot (on an average), there is at most one station attempting transmission. Otherwise, collision will occur more frequently and the overall throughput will decrease. To prevent X from increasing too fast, for larger $\tau^{(l)}$ (corresponding to smaller contention window), $N^{(l)}$ should be smaller. Therefore, it is not appropriate to have all flows use very high priority since which may lead to early performance degradation.

In the above analysis, we assumed that $msg^{(l)}$ is a constant L for any class l. If that is not the case, we can define $C^{(l)}$, such that $msg^{(l)} = L * C^{(l)}$. Notice that $T_s^{(l)}$ is approximately proportional to $msg^{(l)}$, accordingly we have $T_s^{(l)} = T_s * C^{(l)}$. Therefore, (1) becomes

$$G = \frac{L \cdot \sum_{l=0}^{M-1} C^{(l)} P_s^{(l)}}{P_c T_c + T_s \cdot \sum_{l=0}^{M-1} C^{(l)} P_s^{(l)}} = \frac{L}{T_s + T_c \cdot \left(\frac{1 - \sum_{l=0}^{M-1} P_s^{(l)}}{\sum_{l=0}^{M-1} C^{(l)} P_s^{(l)}} \right)} \quad (6)$$

Let us define $S' = \sum_{l=0}^{M-1} C^{(l)} P_s^{(l)}$ as the normalized total probability of successful transmission of all flows. If S' is larger, *G* is correspondingly larger.

Notice that

$$S' = \sum_{l=0}^{M-1} C^{(l)} P_s^{(l)} = \prod_{l=0}^{M-1} (1-\tau^{(l)})^{N^{(l)}} \cdot \sum_{l=0}^{M-1} C^{(l)} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \quad (7)$$

And similar to (5), the change of S' can be derived as

$$\Delta S' = S'_{new} - S' = \tau^{(i)} \prod_{l=0}^{M-1} (1-\tau^{(l)})^{N^{(l)}} \cdot \left(C^{(i)} - \sum_{l=0}^{M-1} C^{(l)} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \right) \quad (8)$$

S' increases only iff $X' = \sum_{l=0}^{M-1} C^{(l)} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}} \le C^{(i)}$. To prevent *X'* from growing too fast, for large $C^{(l)}$, $N^{(l)}$ should be smaller.

From the above analysis, it can be seen that the overall throughput depends on $\tau^{(l)}$ (probability that a station attempts transmission), $N^{(l)}$ (number of active stations), and average packet size of each service class. Since $\tau^{(l)}$ is decided by CW_{min}, it is necessary to obtain the optimal CW_{min}'s such that *X'* can be minimized. Currently, there are three approaches proposed:

- Zhao and Zhang [9] suggest to obtain optimal CW_{min}'s by solving a constrained optimization problem with minimum bandwidth requirement for each service class. However, this approach has the disadvantage of expensive implementation. Whenever the number of active stations changes, new CW_{min}'s have to be re-calculated. Obviously, this is quite inefficient for a MAC layer protocol, especially with large number of stations.
- Aad and Ni [6] propose to slowly decrease the CW_{min} after a packet is successfully transmitted to accommodate the fact that the network may get congested somehow and its load may not decrease immediately. This technique can effectively avoid excessive channel contention. However, it has the drawback of resulting in higher CW_{min}'s and thus reduces transmission probability $\tau^{(l)}$, which is not good for real-time video/audio applications.
- Romdhani and Ni [3] propose to adjust CW_{min} according to traffic load for better performance. However, since all stations increase their contention windows in the same way, each flow (including real-time flows) experiences significant performance loss under high traffic loads, making it insufficient to support multimedia applications.

We realize that in order to achieve our objective of supporting multimedia applications over wireless LANs, it is not a good idea to compromise the QoS requirements of *all* flows, i.e., to increase CW_{min}'s of all flows. Instead, we feel that a more feasible approach is to modify $N^{(l)}$ so that the number of stations with similar backoff interval can be reduced across different service classes. This modification can result in fewer collisions thereby improving the throughput and delay characteristics. In addition, this approach has the following interesting features.

- This approach does not require the modification of underlying scheduling algorithm and thus is compatible with the existing EDCF standard.
- Changing $N^{(l)}$ helps keeping good flow differentiation, especially in favor of existing real-time high priority flows.
- Changing $N^{(l)}$ also changes CW_{min}, CW_{max}, *AIFS*, and thus enhances service differentiation.

Modification of $N^{(l)}$ can be realized by dynamically re-allocating MAC priorities of flows and the proposed priority re-allocation scheme is described in the following section.

4. PRIORITY RE-ALLOCATION

4.1 Optimal Priority Distribution

To intuitively show how priority re-allocation is performed, let us consider the following situation. Suppose we have a large number of flows of different priorities. If the first a few flows are of high priorities, X will reach 1 even when N is still small. That means, if priorities are not re-allocated from the arrival of the first flow, the overall throughput is going to decrease very quickly. One possible solution is that new flows after $X \geq 1$ should use large $CW_{min}^{(l)}$ such that the overall throughput decreases more slowly. However, this is not desirable for the following reasons:

- Assigning large $CW_{min}^{(l)}$ for new flows will prevent from supporting new arriving real-time flows.
- Misuse of priority by one single flow may significantly affect the performance of later arriving flows.
- The optimal priority distribution depends on the predefined $CW_{min}^{(l)}$, i.e., for different configurations of $CW_{min}^{(l)}$, the optimal distribution can be different. Thus, it is quite difficult to decide the optimal $CW_{min}^{(l)}$ for a new flow.

Hence, instead of re-allocating priorities after $X \geqslant 1$, we propose to reassign flow priorities from the beginning. Furthermore, for the sake of efficiency, we suggested to make the number of active flows of each priority to be evenly distributed [14]. Under low to medium traffic load, assigning a lower priority to a flow does not decrease its received throughput because of the low collision rate. Under high traffic load, re-allocating priorities can effectively maintain low collision rate and also keep good service differentiation such that the delay constraints of most real-time flows can be satisfied. Another desirable feature of priority re-allocation is that it can easily handle misuse of priority by assigning lower priority to misbehaving flows and protect other flows. The effectiveness of even priority distribution is illustrated in the following simulations.

In this simulation, we measured the overall throughput and collision rate of the network with respect to the number of priorities. Here, "number of priorities" means that how many priorities are evenly assigned to all flows. To be more specific, if the number of priorities is i, $1 \leq i \leq 8$, then flow priorities are evenly distributed in the range of [7-i+1, 7]. For EDCF, the number of priorities also means the number of queues to be maintained. In one extreme scenario, we have all flows in the queue of highest priority, 7. In another scenario, we evenly assign flow priorities such that the difference of queue length between any two priority queues is no more than 1. We simulate 24 flows, each with rate 80KBits/sec. In the simulation, we evenly assign the priority for all flows, but vary the range of the number of priorities from 1 to 8, respectively. Fig. 1 and Fig. 2 depict the result of this simulation. It can be seen that when flow priorities are evenly distributed among more queues, better performance can be achieved.

Then, simulations are conducted to show the performance of real-time flows with different priority distribution under different traffic loads. All flows are real-time flows. For first four simulations, priorities are set to 7 (the highest priority), 6, 5, 4, respectively, for all flows. Then flow priorities are re-allocated according to even distribution to show the improvement of the proposed scheme. Fig. 3 and Fig. 4 depict the overall received throughput and average normalized throughput, respectively. It can be seen that when all flows are of the same priority, the overall throughput is decreasing after the number of flows exceeds 10. The higher the priorities, the earlier the decrease occurs. This result is consistent with our previous analysis that under low and medium traffic load, re-allocating priorities does not decrease the overall throughput because of the low collision rate, and when traffic load becomes high, low priority is preferred for smaller collision rate. When the number of flows is 20, the collision rate is large in all cases. However, high overall throughput is always obtained with even priority distribution and the improvement is up to 44%. This result also shows that using *slow decrease* approach (if all flows have priority of 7, re-allocating their priorities to 6, then to 5, and so on, with the increase of traffic load) for priority re-allocation does not provide good performance gain. Thus, our approach is better than slow decrease scheme for priority re-allocation.

4.2 Algorithm for Priority Re-allocation

From the previous section, it can be seen that by evenly re-allocating the priorities among all flows, system performance can be significantly improved. However, in the simulation, we assume that we are only concerned about the overall performance. In designing a general approach for priority-reassignment, we need to consider the performance of individual flows, i.e, we must make sure that priority re-allocation does not make the performance of an individual real-time flow worse than EDCF. For example, assigning priority of 2 to a flow which requests priority of 7 will lead to serious performance loss for that flow. On the other hand, it is not necessary to re-allocate high priority for best-effort flows. Therefore, it is necessary to set lower bound for high-priority flows (less delay-tolerant flows) and upper bound

for low-priority flows (delay tolerant flows). Generally, the priorities of flows are set by applications according to Table 1. In EDCF, at most eight TC queues can be maintained. So, this table can be directly mapped to EDCF.

It is noted that traffic types within the priority range from 4 to 7 are actually real-time flows, while those whose priorities are from 0 to 3 are best-effort flows. Thus, we classify flows into two classes, *high priority*[§] *class*, and *low priority*[**] *class*, which have priority from 4 to 7, and 0 to 3, respectively. The flows in these classes are called high priority flows and low priority flows, correspondingly. Two flows are said to be *of the same class* (high or low) if they are in the same range specified. Also, *Flow_length* of a priority *p*, *Flow_length*(*p*), is defined as the total bandwidth demands of all flows of priority *p* in the network. Since flow rate is related to packet size, we actually include the effect of packet size in our priority re-allocation algorithm by considering the flow rate. The pseudo-code for dynamic priority re-allocation is described in Fig. 5.

If all flows have the same bandwidth demand, then *Flow_length* is equivalent to the total number of flows with the same priority. In order to make sure that the performance of an individual flow does not decrease significantly after priority re-allocation, the new priority should also be of the same class as the original one. For example, if the original priority of a new flow is 7 and *Flow_length*(1) = *min* {*Flow_length (p)*, $0 \le p \le 7$}, and *Flow_length*(5) = *min*{*Flow_length (p)*,$4 \le p \le 7$}, we should assign the flow new priority 5, instead of 1, since 1 and 7 are not of the same class. By setting the boundaries for priority re-allocation, we can ensure the aggregate performance of real-time flows.

Of course, some flows may get lower priority than their requested value and thus receive lower quality of service than expected. However, this is necessary for quality of service guarantee of overall performance of real-time flows. It can be seen from simulation results that in high traffic load, priority re-allocation ensures that most flows receive satisfactory performance. Whereas for original EDCF scheme, all flows suffer from serious performance degradation.

In addition, priority re-allocation affects the fairness of flows, i.e., for flows of the same priority, some flows are reassigned lower priority and some may get higher priority, depending on the distribution of flow priorities at the time of a flow's arrival. For better fairness, it is possible to consider the order of flow arrival as a factor for choosing the loser and thus providing more stable QoS guarantee for existing flows.

4.3 Implementation

Since re-allocation of priorities requires the knowledge of $N^{(l)}$ and flow rates, information of flows transmitting through other stations needs to be collected. Similar to the approach followed in PCC [4], we embed flow information in RTS/CTS. In infrastructure model, every RTS message is relayed by access point (AP) and AP collects flow information within its cell and sends back the re-allocated priorities to the senders through CTS. In Ad-hoc model, a station listens to the channel, gets information from RTS messages being transmitted, and then re-allocates its own priority locally. By collecting the flow information, a station can obtain the priority distribution of flows transmitting within its broadcast region. Based on this distribution, a station can dynamically re-allocate its flow priority.

Next, we explain how the MAC frame is modified to accommodate our proposed scheme. We insert a field called *Extensible Field* (EF), which contains a unique *flow id*, *flow priority*, and *demand rate*. The new frame format is shown in Fig.6. In RTS, the flow priority is the original one and in CTS, it may be the re-allocated one. The *Rate* field is 11 bits, corresponding to 2047. So a flow can request up to 2MB bandwidth, which is large enough for almost all cases.

5. PERFORMANCE EVALUATION

The performance metrics we are going to measure are as follows.

- **Normalized throughput** is the percentage of demand throughput received by an individual flow or all flows. This metric is used to show how well qualities of service of flows are guaranteed.
- **Collision rate** is the number of collisions per second. It is used to show how serious the collision is in a network.
- **Medium utilization** is the metric which shows how good the channel is effectively used and is calculated as

$$M_u = \frac{total\_time - collision\_time - idle\_time}{total\_time} * 100\%$$

- **Mean delay** is the average packet delay incurred for an individual flow.

[§] In this paper, we use high priority flows and real-time flows interchangeably.

[**] In this paper, we use low priority flows and best-effort flows interchangeably.

- **Delay distribution** shows the percentage of flows whose mean delay is less than a certain delay constraint. We can use this metric to show how well flows' delay constraints are satisfied.

ns-2 network simulator is used to simulate the proposed scheme on Linux machines. The physical channel bandwidth is set to be 2Mbps, and it is assumed that a station only initiates one flow at a time and all stations are stationary. In this simulation, we have n+1 number of stations and n/2 number of flows. Without loss of generality, a flow i ($1 \le i \le n/2$) is setup from station 2i-1 to station 2i. For simulation of the infrastructure model, station 0 is specified as the AP. The duration of the simulation is set such that measured metrics converge to fixed values. The values of CW_{min} 's for different priorities are listed in Table 2. The packet size is 800bytes and all flows have the same rate of 80Kbits/sec. For convenience, we call our scheme "EDCF+PRIO". It should be observed here that we have not considered packet loss due to environmental conditions as well as the mobility of stations. We plan to consider these effects in our future work.

5.1 Infrastructure Model

First, a simulation is conducted to see how well real-time flows sustain under high traffic load of best-effort flows since EDCF is claimed to provide guaranteed service differentiation for high priorities flows. We fixed the number of high priority flows to be 10 (with priority of 6), and then vary the number of low priority flows (with priority 1) from 0 to 20. The average normalized throughput of high priority flows is depicted in Fig. 7. It is obvious that with the increase of the number of low priority flows, high priority flows receive only 60% of the requested bandwidth, on an average. However, with priority re-allocation, more than 80% of the requested bandwidth is offered, which is quite good for soft-QoS guarantee. Thus, it can be seen that our priority re-allocation scheme can effectively minimize the effect of best effort flow traffics to high priority traffics. From Fig. 8, it can be seen that in this simulation, the gain in average normalized throughput increases with the number of low priority flows. When the number of low priority flows is 20, the gain is 34%.

Next, we looked at the performance of low priority flows since we also re-allocate priorities for low priority flows. It can be seen from Fig. 9 that with our scheme, the average normalized throughput of low priority flows is almost doubled compared to EDCF. When the number of low priority flows is 10, low priority flows receive 10% of the requested rate with our scheme, i.e, low priority flows are not starved. But EDCF only offers 4.7%, which is comparably low. We also monitor the number of dropped packets and the result is shown in Fig. 10. It can be seen that using our proposed algorithm, the number of packets being dropped is much smaller.

Then, we compare the collision rate and channel utilization under our scheme and EDCF in Fig. 11 and Fig. 12, respectively. It can be seen that the collision rate and channel utilization are very similar with these two approaches. Note that the low collision of EDCF is caused by the selection of priority 6 and 1 for high and low priority flows, respectively. With these two priorities, contention windows are not very small and collision rate is not high. This also shows that low collision rate does not necessarily lead to high performance in EDCF, while priority re-allocation can always maintain high performance and low collision rate.

Finally, we measured the mean delays of high priority flows. The distribution of mean delays of all high priority flows are depicted in Fig. 13 and Fig. 14 for the case of when the number of low priorities is 0 and 10, respectively. It can be seen that when traffic load is low, all high priority flows experience very low delay of 20-30ms. However, when the network is heavily loaded, all flows have mean delay of more than 0.5 second under EDCF. But with priority re-allocation, 60% of the high priority flows have mean delays of less than 0.1 second.

In Table 3, the mean delays of each high priority flow (the number of low priority flows is 20) are listed. It can be seen that even under heavy traffic load, flows with new re-allocated priority of 6 and 7 only experience 30-90ms delays, which is very desirable for audio and video applications. But EDCF only guarantees the mean delay of about 600ms, which is not tolerable for multimedia applications. If we set the delay target as 100ms, most of the packets would be dropped. Therefore, after priority re-allocation, new priorities can be seen as levels of guaranteed of quality of service that can be offered under the current network condition. Notice that our scheme has similar collision rate as EDCF. However, in EDCF, more collisions happen between high priority flows and thus seriously increase the delay of colliding packets. However, with priority re-allocation, most collisions happen between low priority flows. Therefore, high priority flows still experience low packet delays.

Flows with new priorities of 4 and 5 experience high delays of seconds, which is not acceptable. This is unavoidable since with re-allocating the priorities, some of the high priority flows have to be sacrificed. However, consider that EDCF actually cannot guarantee any QoS when the traffic is high, it is better to provide QoS guarantee to most flows

(in this simulation, it's more than 60%) than to guarantee none of them. Therefore, with priority re-allocation, considerable improvement can be achieved.

5.2 Ad-hoc network

In ad-hoc model, we assume all stations are within the transmission range of each other such that routing is not necessary. (Studying the effect of priority re-allocation with different routing algorithms under different mobility patterns is more involved. We are currently working on this study and will present the results in future). In this simulation, we increase the number of flows gradually from 1 to 20. All flows are high priority flows with initial priorities of 6. Fig. 15 and Fig. 16 show the normalized overall throughput and the mean delay distribution when there are 20 flows in the network. It can be seen that in one-hop ad-hoc network, the overall throughput is higher than for infrastructure model because in infrastructure model, all packets have to be relayed by AP, which decreases the overall throughput. This result shows that under various traffic loads, priority re-allocation can always maintain high overall throughput. Also, considerable improvement is achieved on minimizing packet delay with priority re-allocation. From Fig. 16, we can see that with our scheme, more than 50% flows have mean delay of less than 50ms and 65% flows have mean delay of less than 100ms. However, for EDCF, 95% flows have mean delay of greater than 100ms. That means, under high traffic load, EDCF cannot guarantee the delay constraints of real-time flows, but our scheme can still provide delay guarantee for most of them. Also, we measured collision rate and channel utilization for comparison of our scheme and EDCF and the result is shown in Fig. 17 and Fig. 18. It can be seen that our scheme always maintains low collision rate and high channel utilization with the increase of traffic load.

6. APPLICATIONS

With our proposed priority re-allocation algorithm, the following issues can also be addressed.

6.1 Admission Control

We have shown the effectiveness of our priority re-allocation algorithm. However, as pointed out in Sec. 3, when the number of stations is large, the overall throughput of the network will decrease. In this case, flow admission control becomes necessary. There are several general admission control approaches, which might be considered. Shah and Chen [11] propose a measurement-based approach for admission control, based on the available bandwidth of the channel. Barry and Campbell [8] devise virtual MAC and virtual source scheme for estimating the incurring packet delay for a new flow. However, notice that in *Lemma* 2, the overloading condition can be combined with our priority re-allocation algorithm for admission control, which is described in following procedure.

- Define Saturation Threshold (*ST*) such that *ST* is closed to 1, e.g., 0.8 or 0.9.
- When AP receives RTS from a station, it re-allocates priority for the corresponding flow with the proposed priority re-allocation scheme.
- Then, AP calculates effective traffic load per time slot, i.e., $X = \sum_{l=0}^{M-1} N^{(l)} \frac{\tau^{(l)}}{1-\tau^{(l)}}$.
- If $X \leq ST$, the flow is accepted. Otherwise, the flow is rejected.
- For low priority flows, no admission control is needed.

Notice that since flow information has already been collected for priority re-allocation, enforcing admission control does not involve any extra signaling overhead. Of course, the accuracy of this admission control strategy depends on the accuracy of $N^{(l)}$ (actual number of active stations) and $\tau^{(l)}$ (transmitting probability of a class l station within a randomly chosen slot). Investigation of this new admission control strategy is of our future work.

6.2 Control over misbehaving flows

Since all stations within a WLAN contend for chancel access, the setting of flow priorities has significant impact on the throughput received by individual flows as well as the performance of the system. Therefore, it is very important that all flows do not acquire much higher priorities than necessary and misuse of priority should be effectively handled to ensure the stable performance of existing flows. Note that misbehaving users may or may not be authorized. For unauthorized flows, they can be discarded anyway. Here, we focus on authorized users who misuse priorities by requesting much higher service classes than necessary. With our proposed priority re-allocation algorithm, only minimum amount of work is needed to deal with this situation. If a flow is found for misusing its priority (by checking the QoS requirements), then its priority can be modified appropriately according to the mapping table, or the flow can

be *punished* by assigning it a very low priority. Even if it is not possible to identify misbehaving flows, a new flow's priority may still be re-allocated to guarantee the overall performance such that presence of misbehaving flows does not necessarily affect the overall performance of the network. Therefore, misuse of priority can be easily handled by our priority re-allocation algorithm.

6.3 Adaptive Resolution Control

When the network becomes heavily loaded, it is generally impossible to support the original spatial or temporal resolution of video applications. Therefore, adaptive resolution control technique should be applied so that the video source coding rate can be changed according to the network load. Tasaka and Nakanishi [12] propose a four-level temporal resolution control scheme, where the degree of network congestion is used as the criteria for the adjustment of the video temporal resolution level. With the proposed priority re-allocation scheme, we can do one-to-one mapping between the resolution level (from 0 to 3) and the flow priority (from 7 to 4), respectively, and a video application decides its resolution level by the assigned priority value from the AP. From our simulation, it is shown that with priority re-allocation, high priority flows still receive the desired quality of service under high traffic load while low priority flows do not. Thus, by this simple priority based dynamic resolution control scheme, smooth video transmission over WLANs can be achieved.

7. SUMMARY

In this paper, we presented an extended performance model of EDCF and the possible condition for a network getting overloaded. Based on the analysis of our extended model, we showed that the overall throughput of a network can be improved by evenly distributing the number of active stations over a set of traffic categories. Hence, we proposed a dynamic priority re-allocation algorithm to make the number of active flows of each priority to be evenly distributed.

Simulation shows that this scheme can guarantee high overall throughput under various traffic loads. Also, under high traffic load, EDCF cannot guarantee any QoS for real-time flows. Whereas the proposed scheme can satisfy the required quality of service for most of them. Specifically, using dynamic priority re-allocation, the mean delays of most real-time flows are under 100ms. Hence, the re-allocation scheme can help wireless networks to support multimedia applications better. Furthermore, with priority re-allocation, low priority flows are not starved when traffic load is high. Applications of our proposed scheme include admission control, control over misuse of priority, and adaptive resolution control for video applications. Our future work will be to investigate the performance of our scheme in ad hoc networks where the effect of routing protocols and node mobility patterns needs to be considered. We will also study the impact of propagation errors and other environmental conditions on the proposed priority re-allocation scheme.

REFERENCES

1. M. Benveniste, G. Chesson, M. Hoeben, A. Singla, H. Teunissen, M. Wentink, *EDCF proposed draft text*, IEEE working document 802.11-01/13lrl, March 2001.
2. A. Lindgren, A. Almquist, O. Schelén, "Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs". *In Proceedings of the 26th Annual IEEE Conference on Local Computer Networks*, pp. 348-351, Tampa, 2001.
3. L. Romdhani, Q. Ni, T. Turletti, "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad Hoc Networks". *In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, Vol. 2, pp. 1373 –1378, New Orleans, 2003.
4. W. T. Chen, Y. Lin, S. Lo, "Priority-Based Contention Control in IEEE 802.11 Wireless LANs". *In 16th International Workshop on Communications Quality & Reliability (CQR)*, Okinawa, Japan, 2002.
5. I. Aad, C. Castelluccia, "Remarks on per-flow differentiation in IEEE 802.11". *In EuropeanWireless 2002*, Florance, Italy, 2002.
6. I. Aad, Q. Ni, C. Castelluccia, T. Turletti. *Enhancing IEEE 802.11 Performance with Slow CW Decrease*. IEEE 802.11e working group document 802.11-02/674r0, 2002.
7. M. Benveniste, "Tiered contention multiple access' (TCMA), a QOS-based distributed MAC protocol". *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 2, pp. 598-604, Lisboa, Portugal, 2002.
8. M. Barry, A. T. Campbell, A. Veres, "Distribued Control Algorithms for Service Differentiation in Wireless Packet Networks". *In Proceedings of IEEE INFOCOM*, Vol. 1, pp. 582-590, Anchorage, Alaska, 2001.
9. J. Zhao, Q. Zhang, W. Zhu, Y.-Q. Zhang, "Throughput and QoS Optimization in IEEE 802.11 LAN". *International Conference on Third Generation Wireless and Beyond*, San Francisco, 2002.
10. V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, E. Knightly, "Distributed Priority Scheduling and Medium Access in Ad Hoc Networks". *In ACM Wireless Networks Journal*, **8**, pp. 455-466, 2002.

11. S. H. Shah, K. Chen, K. Nahrstedt, "Dynamic Bandwidth Management for Single-hop Ad Hoc Wireless Networks". *In Proceddings of 1st IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 195-203, Dallas-Fort Worth, 2003.
12. S. Tasaka, H. Nakanishi, Y. Ishibashi, "Dynamic Resolution Control and Media Synchronization of MPEG in Wireless LANs". *In Proceedings of GLOBECOM '97*, vol. 1, pp. 138-144, Phoenix, 1997.
13. N. H. Vaidya, P. Bahl, S. Gupta, "Distributed Fair Scheduling in Wireless LAN". *In Proceedings of ACM MOBICOM*, pp. 167-178, Boston, 2000.
14. M. Li, B. Prabhakaran, S. Sathyamurthy, "On Flow Reservation and Admission Control for Distributed Scheduling Strategies in IEEE802.11 Wireless LAN," *in Proceeding of the Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pp. 108-115, San Diego, 2003.
15. H. Zhu, I. Chlamtac, "An analytical model for IEEE 802.11e EDCF differential services," *In Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN)*, Dallas, Oct. 2003.

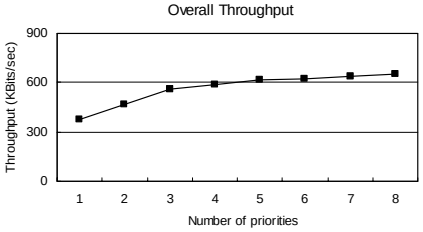

Fig. 1. Overall throughput

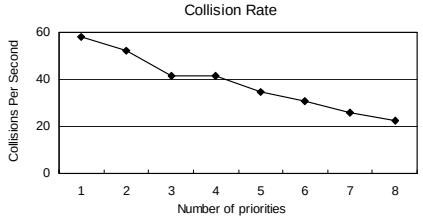

Fig. 2. Collision rate

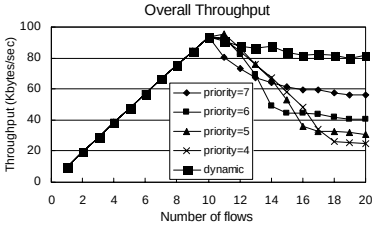

Fig.3. Overall throughput

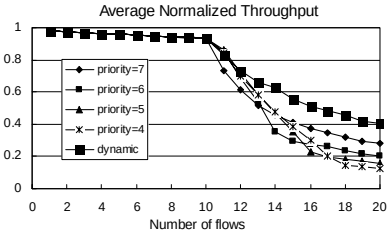

Fig.4. Average normalized throughput

Algorithm: Priority Re-allocation

```
int Reallocate_priority (flowid, priority, rate)
List: flow_list;
if flowid is not found in flow_list
   if  4 ≤ priority ≤ 7
       find   priority p ( 4 ≤ p ≤ 7 )   with the smallest
       Flow_Length;
       new_priority = p;
   if  0 ≤ priority ≤ 3
       find   priority p ( 0 ≤ p ≤ 3 )   with the smallest
       Flow_Length;
       new_priority = p;
   insert flowid, priority, new_priority, rate in flow_list;
   return new_priority;
else
   if the transmitted packet is the last flow   packet
       delete the flow from flow_list;
       assign the priority to a flow whose old priority is closest;
       return priority;
   else
       new_priority is obtained from flow_list;
   return new_priority;
end
```

Fig.5. Algorithm of priority re-allocation

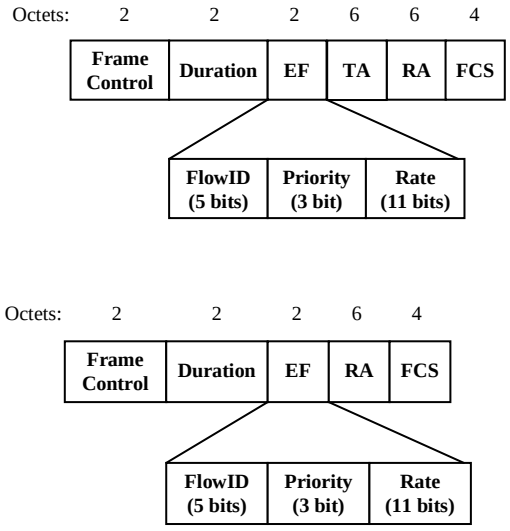

Fig.6. New frame format for RTS(top) and CTS(bottom).

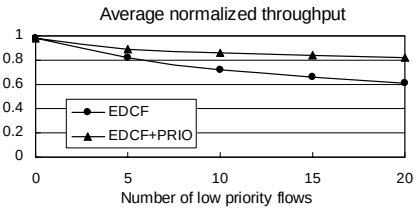

Fig. 7. Average normalized throughput of high priority flows

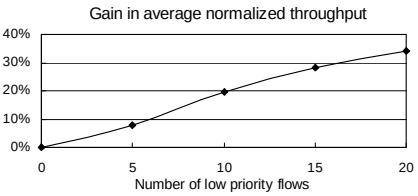

Fig. 8. Gain in average normalized throughput of high priority flows

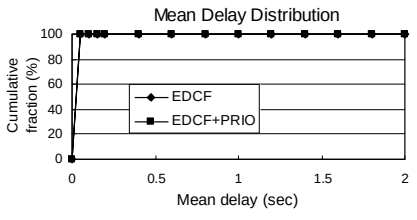

Fig. 13. Mean delay distribution (0 low priority flows)

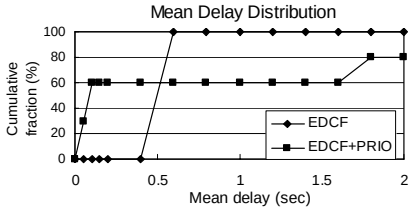

Fig. 14. Mean delay distribution (10 low priority flows)

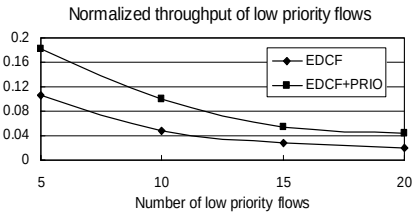

Fig. 9. Average Normalized throughput of low priority flows

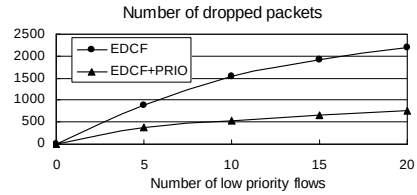

Fig. 10. Number of dropped packets

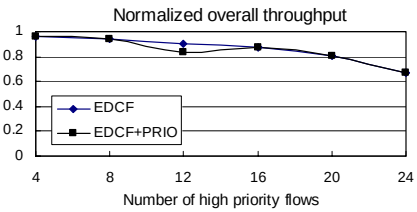

Fig. 15. Normalized overall throughput

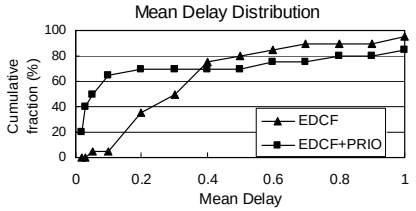

Fig. 16. Delay distribution (20 flows)

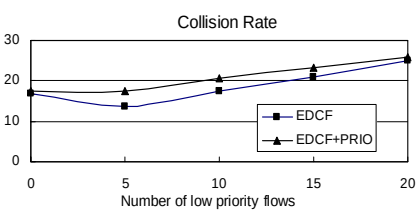

Fig. 11. Collision rate

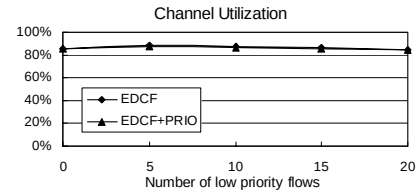

Fig. 12. Channel utilization

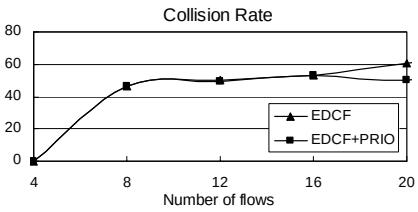

Fig. 17. Collision rate

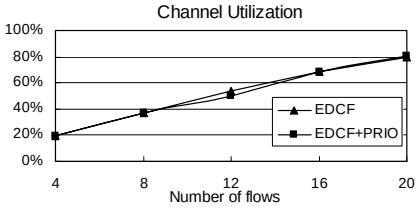

Fig. 18. Channel utilization

User priority	Traffic type
7	Network Control
6	Voice
5	Video
4	Controlled Load
3	Excellent Effort
2	Spare
1	Background
0 (default)	Best Effort

Table 1. Priority assignment to traffic types

Priority	0	1	2	3	4	5	6	7
CW_{min}	512	512	255	127	63	31	15	7

Table 2. $CW_{\min}$ used for simulation of EDCF.

Flowid	Old Priority	New Priority	EDCF	EDCF+PRIO
1	6	6	547.136	89.417
2	6	7	549.872	39.624
3	6	5	603.457	1716.92
4	6	4	581.069	7222.5
5	6	6	593.084	120.652
6	6	7	586.061	33.412
7	6	5	597.242	3904.52
8	6	4	608.267	8587.07
9	6	6	643.989	93.617
10	6	7	613.274	35.361

Table 3. Mean delay of high priority flows (ms)