JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 24, 1065-1079 (2008)

A Modified EDCA with Dynamic Contention Control for Real-Time Traffic in Multi-hop Ad Hoc Networks

YANG-JING WU, JEN-HUNG CHIU AND TSANG-LING SHEU
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, 804 Taiwan
E-mail: sheu@ee.nsysu.edu.tw

EDCA (Enhanced Distributed Channel Access), developed for providing service differentiation in wireless LANs, has become a MAC standard in IEEE 80.2.11e. However, unexpected packet delay due to collisions and retransmissions may occur when the wireless LAN becomes congested. To support the increasing demand of delay-sensitive multimedia applications, this paper presents a modified EDCA with dynamic contention control (DCC) for real-time traffic in multi-hop ad hoc networks. The proposed scheme not only reduces the average packet delay but also increases the percentage of packets meeting real-time constraints by two approaches: (1) dynamically adjusting the priority level of a traffic flow based on the estimated per-hop delay, and (2) generating a non-uniformly distributed backoff timer for retransmitted frames according to their individual end-to-end delay requirements. For the purpose of evaluation, we perform simulations on ns-2. We compare the performance of the proposed DCC scheme to EDCA and a previous work, AEDCF (adaptive EDCF). The simulation results demonstrate the effectiveness and superiority of our proposed DCC scheme.

Keywords: EDCA, multi-hop ad hoc networks, real-time traffic, per-hop delay, end-to-end delay

1. INTRODUCTION

In the last decade, the emergence of mobile applications has led to the rapid deployment of wireless networks. Instead of infrastructure mode, more and more wireless networks have been deployed in ad hoc mode because of the expensive investment in base stations [1, 2]. One of the important issues in wireless ad hoc networks is to design a medium access control (MAC) with which mobile nodes can share the medium effectively in a distributed manner. There are two configurations for a wireless ad hoc network: fully-connected and multi-hop. A wireless ad hoc network is said to be fully-connected if any pair of nodes can directly communicate with each other and multi-hop otherwise. In this paper, we mainly focus on multi-hop ad hoc networks, since fully-connected configuration is a special case of multi-hop one.

For a wireless ad hoc network, CSMA is the most pervasive MAC scheme that permits a mobile node to start packet transmission based only on the knowledge of whether the channel is idle. Yet, packet collisions occur more and more frequently when the network is getting congested. In IEEE 802.11 [3], a well-known CSMA/CA (collision avoidance) protocol with a uniformly-distributed random backoff timer was designed to alleviate the problem of collisions, but it lacks the QoS provision for multimedia applica-

Received July 27, 2006; revised February 13, 2007; accepted April 3, 2007.
Communicated by Ten-Hwang Lai.

tions. To provide service differentiation at the medium access control layer, IEEE 802.11e [4] has standardized another MAC scheme named EDCA[1] (Enhanced Distributed Channel Access). EDCA supports up to eight user priorities mapped into four access categories (ACs). AC values of 0, 1, 2, and 3 represent best effort, background, video, and voice traffic, respectively. The four traffic categories are differentiated by different contention parameters, including arbitration interframe space (AIFS), minimum/maximum contention window size (CW_{min}/CW_{max}), and persistence factor (PF) for expanding the contention window. The smaller AIFS/CW_{min}/CW_{max}/PF brings the larger probability of winning the channel access. Right after a short interframe space (SIFS), a node will send an acknowledgement (ACK) frame for confirmation on successfully receiving a frame. If a node transmitting a frame does not receive the corresponding ACK within a specified ACK timeout, it will reschedule the frame transmission by invoking a retransmission procedure. Note that the feature of PF was discussed in the developing phase of 802.11e but completely removed in the standard release.

Per-class differentiation used by EDCA provides better service to high-priority traffic while offering minimum service to low-priority traffic. Although EDCA has considered the delay sensitivity of real-time traffic, the QoS improvement is limited because the contention parameters of EDCA can not adapt to the network conditions. Recently, many mechanisms were proposed to improve EDCA by modifying its contention parameters [5-10]. However, the modified schemes mainly focus on single-hop ad hoc networks. Aiming to improve the EDCA performance for real-time traffic in multi-hop ad hoc networks, we propose a modified EDCA with dynamic contention control (DCC) to reduce the packet delay by taking into account end-to-end delay requirements. With the ability to estimate the time consumed at each hop and alleviate the delay incurred by retransmissions, the proposed DCC scheme is expected to be very feasible at the MAC layer, attaining end-to-end delay guarantees for real-time traffic in multi-hop ad hoc network.

The remainder of this paper is organized as follows. Section 2 discusses the related works of our scheme. The proposed DCC scheme is described in section 3. The simulation results and discussions are presented in section 4. Finally, concluding remarks are given in section 5.

2. RELATED WORKS

The schemes previously proposed to improve the EDCA performance can either dynamically adjust the contention window or modify the exponential backoff algorithm with a varying PF. For example, Wong and Donaldson [11] presented an age-dependent backoff (ADB) algorithm, in which the PF associated with high-priority traffic can be dynamically tuned based on the waiting time (*i.e.*, age) for the packet in the transmission queue and the lifetime of the packet. Moreover, real-time packets that are not received by destination nodes within their lifetime are considered to be obsolete. However, the adjustment of PF has less QoS improvement than that of contention window when the network is getting congested. Therefore, most of the modified EDCA schemes are devoted to adjusting contention windows according to the network conditions.

To reduce delay/jitter for real-time packets and increase the channel utilization, Gannoune and Robert [12] extended EDCA with a dynamic adaptation algorithm that

[1] EDCA used to be called EDCF.

enables each node to periodically generate a new CW_{min} according to traffic load and channel conditions. Actually, their improvement in packet delay is not so significant since only the contention parameter CW_{min} can be dynamically adjustable. An adaptive continuous transmit EDCF (ACT-EDCF) proposed by Wang *et al*. [13] allowed mobile nodes to tune the size of contention window and continuously transmit a certain amount of high-priority packets by monitoring the channel status in terms of transmission attempt failure probability and collision probability. The ACT-EDCF may starve low-priority traffic for the sake of using burst ACK mode in EDCF to support high-priority traffic. Aiming to share the channel capacity efficiently among multi-priority traffic and achieve QoS enhancement for real-time traffic, Romdhani *et al*. [14] presented an adaptive EDCF (AEDCF), which not only can update the contention window for each AC based on the ratio of the number of collisions to the total number of packets sent during a constant period, but also can assign a smaller value of PF to higher-priority traffic. Similar to the idea behind the AEDCF, two schemes [15, 16] were proposed.

The ad hoc network considered in the above-mentioned schemes is single-hop. Ahn *et al*. [1] proposed SWAN to support service differentiation without the need of introduction and management of per-flow state in multi-hop ad hoc networks. Assuming best effort MAC technology, SWAN designs an AIMD rate control algorithm for each hop to adjust the transmission rate of best effort traffic and uses source-based admission control for real-time sessions according to the bottleneck bandwidth detected by a probing packet toward the destination node. Unlike SWAN, QoS provisioning in our work is mainly managed at the MAC layer not at the transport layer because QoS support in wireless networks is highly dependent on the status of a wireless link. Besides, when the network is getting congested, it suffers from longer and longer delay for a real-time admission controller to receive the probe response message. One of the main contributions of our work is that the percentage of packets arriving at receivers in time is increased, since the packets arriving late are obsolete for real-time traffic. With the proposed dynamic contention control (DCC) scheme, mobile nodes can estimate per-hop delay from either the received MAC-layer ACKs or the control packets of a reactive[2] routing protocol. The estimated per-hop delay and the accumulation of per-hop delays are used to dynamically adjust the size of contention window for each user priority at any nodes. On the other hand, the DCC scheme can alleviate packet delay/jitter by generating a non-uniformly distributed backoff timer for retransmitted frames based on their individual end-to-end delay requirements.

3. THE DYNAMIC CONTENTION CONTROL SCHEME

3.1 Model Descriptions

Consider eight user priorities for multimedia traffic in a multi-hop ad hoc network and assume a reactive routing protocol (*e.g*., AODV [18]) is installed at each mobile node to construct routes to any destination nodes. We say the priority i_1 is larger than the priority i_2 if $i_1 > i_2$. The dynamic contention control (DCC) scheme includes procedures to estimate per-hop delay for each priority at the MAC layer and to generate a non-uni-

[2] Routing protocols for multi-hop ad hoc networks can be classified into two broad categories: reactive and proactive [17]. A reactive protocol initiates route computation only on demand.

formly distributed backoff timer for retransmitted frames. A frame can be raised to the higher priority or be dropped at an estimated rate by any hop j depending on the difference between its remaining time constraint and the accumulation of per-hop delays from destination up to hop j. In addition, when a frame is to be retransmitted, the associated random backoff timer is drawn according to a non-uniform distribution with three parameters, the remaining time constraint, the residual hop count, and the number of transmission retries. Following are the denotations for the parameters used in the proposed DCC scheme.

- *Mrtt*[*i*]: the per-hop round trip time measured at the MAC layer for priority-*i* traffic.
- *Srtt*[*i*]: the sum of *Mrtt*[*i*]'s from destination up to source or an intermediate node.
- *Max_d*[*i*]: the maximal endurable end-to-end delay for priority-*i* traffic.
- *Tr*: the remaining time constraint with an initial value being *Max_d*[*i*] for priority-*i* traffic.
- *Tc*: the minimum time period to initiate dynamic priority adjustment.
- *Rd*[*i*]: the percentage of priority-*i* packets with $Tr \leq Srtt[i]$, calculated once per *Tc*.
- *Max_hc*: the maximum hop count along a routing path.
- *R_hc*: the residual hop count along a routing path.
- *Rlimit*: the upper limit of transmission retries for a frame before being dropped.

3.2 Computations of *Mrtt*[*i*] and *Srtt*[*i*]

As shown in Fig. 1, the routing table cached at a mobile node is modified by the DCC scheme so that *Mrtt*[*i*] and *Srtt*[*i*] for $0 \leq i \leq 7$ can be recorded with each route entry destined to a certain destination IP address. Initially, *Mrtt*[*i*] and *Srtt*[*i*] are set to *Max_d*[*i*]/*Max_hc* and zero, respectively to ensure that no packets are regarded as obsolete ones in the beginning. Then, *Mrtt*[*i*] and *Srtt*[*i*] with respect to any destinations are measured from received MAC-layer ACKs when the routes function normally. Fig. 2 illustrates how *Mrtt*[*i*] is measured and updated when no link failure occurs. Along a routing path of priority *i*, each sending node measures *Mrtt*[*i*] by starting a timer at the instant when an MPDU (MAC protocol data unit) enters the transmission queue and terminating the timer at the instant when the corresponding ACK frame is received. The difference between the two above-mentioned time instants is used to update *Mrtt*[*i*], and the latest value of *Srtt*[*i*] is obtained by summing up the new *Mrtt*[*i*] measurement and the value recorded in a new-added field, *Srtt*, of the received ACK frame. The format of an MAC-layer ACK frame is modified with the DCC scheme as shown in Fig. 3. The fourth, fifth, and sixth fields in the modified format are new-added. The 4-byte *Srtt* field is used to deliver the value of *Srtt*[*i*] (in unit of microseconds) recorded in the routing table of a receiving node when an ACK is replied. Note that *Srtt*[*i*] at any destination nodes is always zero. Since *Srtt*[*i*] is concerned with end-to-end time lag instead of hop-by-hop behavior, the *DST IP* field is added to contain the destination IP address encapsulated by the successfully

DST IP	Next Hop	Hop Count	*Mrtt*[0] ~ *Mrtt*[7]	*Srtt*[0] ~ *Srtt*[7]
⋮	⋮	⋮	⋮	⋮

Fig. 1. The modified routing table cached at a mobile node.

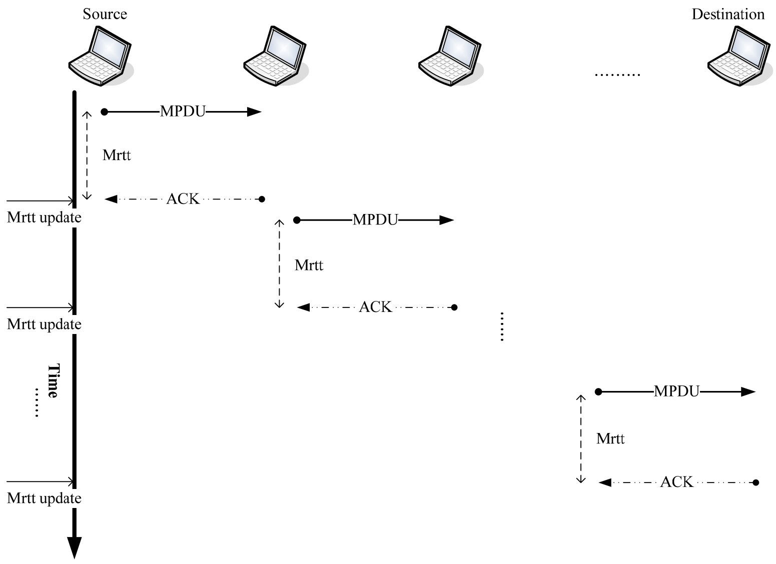


Fig. 2. Timing diagram of a flow traversing on a route without link failure.

Frame Control (2 bytes)	Duration (2 bytes)	RA (6 bytes)	DST IP (4 bytes)	QoS Control (2 bytes)	*Srtt* (4 bytes)	CRC (4 bytes)

Fig. 3. The format of a modified MAC-layer ACK frame.

transmitted frame. The *QoS Control* field indicates the priority level of the successfully transmitted frame as in EDCA. To minimize the impact of transient network conditions, the estimated *Mrtt*[*i*] is smoothed by using the low-pass filter

$$Mrtt[i] = (1 - \alpha)Mrtt[i]_{new} + \alpha Mrtt[i]_{old},\ (0 < \alpha < 1). \quad (1)$$

If link failure occurs along a routing path, some intermediate nodes will be changed after route repair. Since old *Mrtt*[*i*] becomes invalid for the new route, the DCC scheme sets the initial value of *Mrtt*[*i*] at each of the new hops by monitoring RREQ (Route Request) and RREP (Route Reply) messages issued by a reactive routing protocol to repair the failed route. As illustrated in Fig. 4, when detecting that node C fails in the original route, node B starts to broadcast an RREQ message to the neighboring nodes. Then, nodes E and F will forward the received RREQ message or reply an RREP message if they have the route entry destined to the IP address of node G. Finally, node D issues an RREP message on receiving the RREQ message forwarded by node F and then, node F forwards the RREP message to node B. *Mrtt*[*i*] at each new hop is estimated as one-second[3] of the interval between the time when an RREQ message is issued and the time when either an RREP message corresponding to the issued RREQ or another RREQ message is received. Similarly, the DCC scheme modifies the format of an RREP packet of AODV as shown in Fig. 5 to obtain new *Srtt*[*i*]. The 4-byte *Srtt* field delivers the updated *Srtt*[*i*] value (in unit of microseconds) to the previous node on the repaired route requested by the node with IP address indicated by the *RREQ_Bcast_ID* field.

[3] Two times of access contention are required for transmitting an RREQ and receiving an RREP/RREQ.

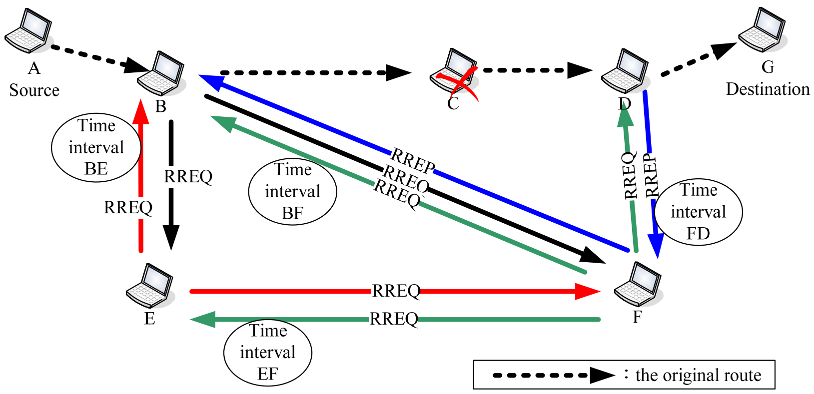


Fig. 4. An illustration of route repair by a reactive routing protocol.

Original RREP Packet
Srtt (4 bytes)
RREQ_Bcast_ID (4 bytes)

Fig. 5. The format of a modified RREP packet.

3.3 Non-uniform Random Backoff Timer

To further reduce the possibility of continuous collisions, we design a non-uniform random backoff timer for retransmitted frames. When frames are to be retransmitted, the associated backoff timers are drawn from the contention window with a non-uniform distribution to let the retransmitted frames have unequal probabilities to get different backoff timers so that continuous collisions among them can be avoided. With the DCC scheme, the retransmitted frame having the smallest remaining time constraint and the largest residual hop count is intended to draw the shortest backoff timer among the frames undergoing the same retransmission retries.

Similar to EDCA, the contention window size is doubled after each unsuccessful transmission until a maximum window size is reached. Assuming the contention window size for a priority-i frame at the nth retransmission is $CW[i]$, the probability that any backoff timer drawn from a uniform contribution is just $1/CW[i]$. Additionally, the DCC scheme divides both of the maximal endurable end-to-end delay $Max\_d[i]$ and the contention window size $CW[i]$ into $(n + 1)$ intervals and maps the remaining time constraint (Tr) of the retransmitted frame to the interval number m ($0 \le m \le n$)

$$m = \lfloor Tr/(Max\_d[i]/(n+1)) \rfloor. \tag{2}$$

Then, the non-uniform random variable X distributed over $[0, CW[i]]$ has the maximum probability located in the interval m

$$P(x_m) = \frac{1}{CW[i]} + \frac{1}{CW[i]}\left(\frac{R\_hc}{Max\_hc}\right), \tag{3}$$

where *Max_hc* (*R_hc*) denotes the maximum (residual) hop count along a routing path as defined in section 3.1. Note that $P(x_m)$ is always smaller than $2/CW[i]$ since *R_hc* is never larger than *Max_hc*. That's, a frame with *n* transmission retries has less probability than one with smaller transmission retries to get a backoff timer smaller than a value in the interval *m*. For the probabilities located in the other *n* intervals of $CW[i]$,

$$P(x_k) = \begin{cases} P(x_m) - (m-k)D, & \text{for } 0 \le k < m \\ P(x_0) - (k-m)D, & \text{for } m < k \le n \end{cases}, \tag{4}$$

where

$$D = \frac{1}{CW[i]}\left(\frac{R\_hc}{Max\_hc}\right) \Big/ \left(\frac{n(n+1)}{2}\right). \tag{5}$$

It is obvious that $P(m-1), P(m-2), \ldots, P(0), P(m+1), P(m+2), \ldots, P(n)$ forms an arithmetic sequence with a common difference $-D$ and

$$\begin{aligned} \int P(x)dx &= \sum_{k=0}^{n} P(x_k)\Delta x_k = \Delta x\left(\sum_{k\neq m} P(x_k) + P(x_m)\right) = \frac{CW[i]}{n+1}\left(\sum_{k\neq m} P(x_k) + P(x_m)\right) \\ &= \frac{CW[i]}{n+1}\left(\frac{n}{CW[i]} - \frac{n(n+1)D}{2} + \frac{1}{CW[i]} + \frac{1}{CW[i]}\left(\frac{R\_hc}{Max\_hc}\right)\right) \\ &= 1. \end{aligned} \tag{6}$$

Thus, $P(x)$ is a probability density function.

To record the remaining time constraint of a packet along a routing path, the header of an IP datagram is extended as shown in Fig. 6. The *Code* field is set to 00100001 for the extended format, the *Length* field is set to 8, the *Pointer* field is set to 5, the *Delta_Priority* field contains how many priority levels of a packet are raised for priority recovery at the next hop, and the *Remaining_Time_Constraint* field records the residually endurable end-to-end delay (in unit of microseconds) of a packet after traversing from source to the current hop.

<table>
<tr><td colspan="4">Original IP Header</td></tr>
<tr><td>Code (1 byte)</td><td>Length (1 byte)</td><td>Pointer (1 byte)</td><td>Delta_Priority (1 byte)</td></tr>
<tr><td colspan="4">Remaining_Time_Constraint (4 bytes)</td></tr>
</table>

Fig. 6. IP header with optional fields.

3.4 The DCC Algorithm

The pseudo-code of the DCC algorithm without route failure is shown in Fig. 7. When a node *k* has a frame of priority *i* to transmit, it will first look up the cached routing table to get *Srtt*[*i*] and *Mrtt*[*i*] associated with the destination IP address. According to the difference between *Tr* and *Srtt*[*i*], either of the following two cases is invoked to deal with the packet before contending for the medium access with EDCA.

```
For node k with a priority-i frame do
{
Start a timer;     // time is t1
Look up the cached routing table;     // get Srtt[i] and Mrtt[i]
If (Tr > Srtt[i])     // the remaining time constraint is larger than Srtt[i]
   IP_Header -> Remaining_Time_Constraint = Tr – Mrtt[i];
   IP_Header -> Delta_Priority = 0;
Else     // the remaining time constraint is smaller than or equal to Srtt[i]
   Look up the cached routing table;     // get Srtt[j] and Mrtt[j]
   If (j is the highest priority for the traffic flows on the route & Tr < Srtt[j])
      Drop the frame at a rate Rd[i];
   Elseif (no priority-j packets are forwarded during Tc)     // guarantee high-priority traffic (*optional)
      IP_Header -> Remaining_Time_Constraint = Tr – Mrtt[j];
      IP_Header -> Delta_Priority = j – i;     // the priority of the frame is raised to j
Contend for the medium access with EDCA;
If (ACK is received)     // within an ACK timeout
   End the timer;     // time is t2
   Mrtt[ACKframe -> QoS_Control] = t2 – t1;     // compute Mrtt
   Srtt[ACKframe -> QoS_Control] = ACKframe -> Srtt + (t2 – t1)     // compute Srtt
   Update the routing table;     // for the values of Mrtt and Srtt
Else     // ACK is not received within an ACK timeout
   Retry transmission;     // with a non-uniform random backoff timer
   If (number of retransmission retries > Rlimit)
      Drop the frame;
}
```

Fig. 7. The DCC algorithm.

Case 1: When $Tr > Srtt[i]$

Update the *Remaining_Time_Constraint* field by the value of ($Tr - Mrtt[i]$) and set the *Delta_Priority* field to zero in the extended IP header of the packet.

Case 2: When $Tr \leq Srtt[i]$

Look up the routing table to get $Srtt[j]$ associated with the destination IP address, where j ($j > i$) is the smallest among the priorities j' with $Srtt[j'] < Tr$. Then, update the *Remaining_Time_Constraint* field by the value of ($Tr - Mrtt[j]$) and set the *Delta_Priority* field to ($j - i$) in the extended IP header of the packet.[4] On the other hand, if j is the highest priority for the flows on the routing path and Tr is still less than $Srtt[j]$, the packet can be dropped at a rate $Rd[i]$. Note that in order to guarantee high-priority traffic, raising the priority to j is allowed only when no priority-j packets are forwarded within the time interval Tc.

After the medium access contention, compute the values of $Mrtt$ and $Srtt$ on receiving a corresponding ACK and record them in the routing table. Otherwise, do retrans-

[4] The MAC-layer priority corresponding to the packet is raised to j and will be set back to i at the next hop.

mission attempts with a non-uniform random backoff timer until the upper limit of transmission retries (*Rlimit*) is reached.

4. SIMULATION RESULTS

We evaluate the performance of the proposed DCC scheme via simulations on NS-2 [19]. As shown in Fig. 8, the multi-hop ad hoc network we consider consists of 14 mobile nodes spread on a 900m × 300m area. Nodes 0 and 1 are sources responsible for generating traffic flows destined to nodes 12 and 13, respectively. In the beginning, any two neighboring nodes are separated by 100 meters on the inner rectangle 600m × 100m. During the simulation time, all of the intermediate nodes move in a random way inside a square area of 50m × 50m and two nodes randomly selected from the intermediate ones move outward at a speed of 50m/sec to cause route failure. All parameters of the modified EDCF with DCC are set as shown in Table 1. Each source generates one AC-3 UDP (User Datagram Protocol) flow as high-priority traffic, one AC-2 UDP flow as low-priority traffic, and at most eight AC-1 UDP flows as background traffic. For comparison, each performance measure is obtained in the proposed DCC scheme along with EDCF [4] and a previous work, AEDCF [14]. Unless explicitly specified, all simulation results are obtained from averages of 10 random samples, each with a running period of 30 seconds.

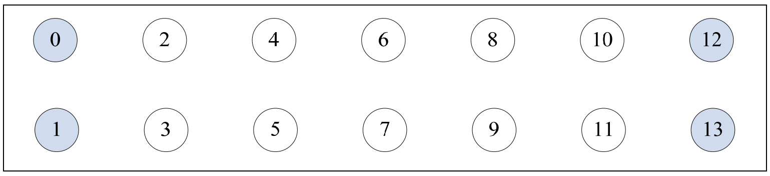


Fig. 8. Network topology used in NS-2 simulation.

Table 1. Parameters and their values.

Parameter	Value
AIFS[1]	3
AIFS[2]	3
AIFS[3]	2
$CW_{min}[1]$, $CW_{max}[1]$	31, 500
$CW_{min}[2]$, $CW_{max}[2]$	15, 100
$CW_{min}[3]$, $CW_{max}[3]$	5, 64
Max_d[2]	150 ms
Max_d[3]	100 ms
Rlimit	7
PF	2
Tc	0.1 second
Channel Capacity	11 Mbps
Packet Size	160 bytes

4.1 Experiment: Scenario One

In the first experiment scenario, the packets of each flow are generated at a constant bit rate (CBR) with the packet interval being 15ms. The decision made to guarantee high-priority traffic in the DCC algorithm is not enabled. Fig. 9 shows the percentage of packets meeting end-to-end delay requirements for high- and low-priority traffic with respect to the number of background traffic flows. It is observed that the percentage of in-time-packets for high-priority traffic with DCC is very close to but a little smaller than those with EDCF and AEDCF, since the decision made to guarantee high-priority traffic is not enabled. However, the more the number of background traffic is increased, the larger the percentage of in-time-packets for low-priority traffic with DCC becomes in comparison with the other two schemes. This is because our scheme can dynamically raise the priority of a packet that could arrive late at destination by referring to its remaining time constraint (*Tr*). Similar results can be seen in Fig. 10, which alternatively shows the average good throughputs (AGT) only considering the packets meeting end-to-end delay requirements.

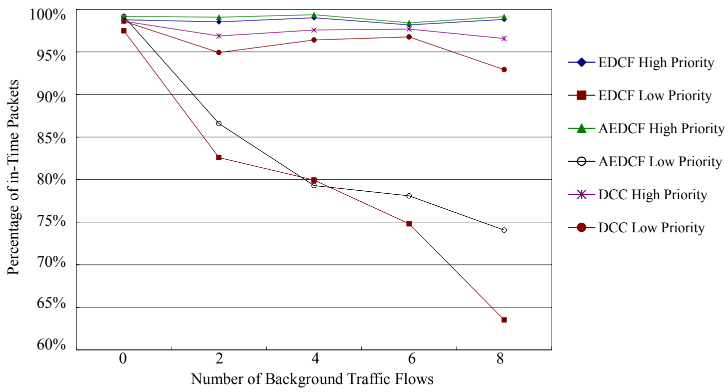


Fig. 9. Percentage of in-time packets without high-priority guarantee.

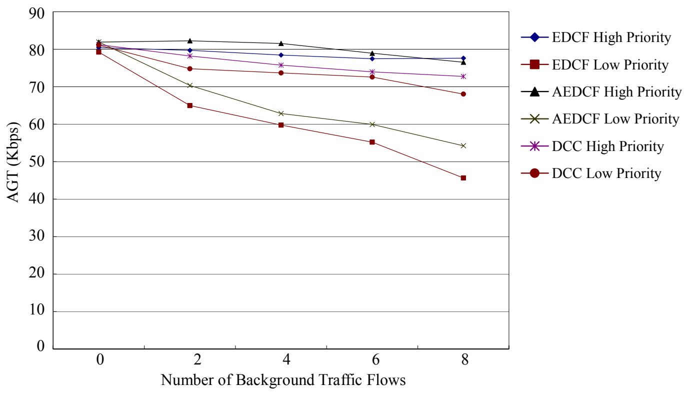


Fig. 10. Average good throughput without high-priority guarantee.

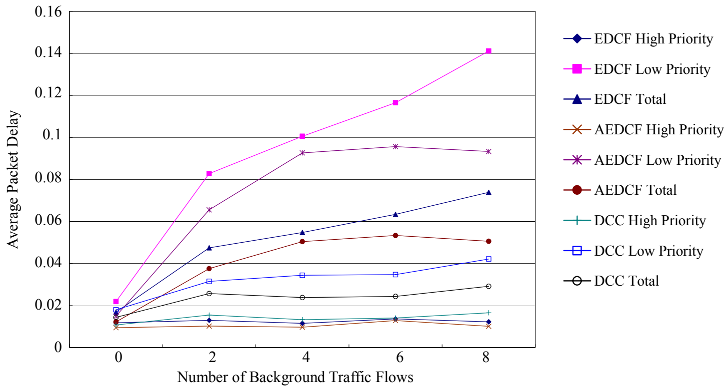


Fig. 11. Average packet delay without high-priority guarantee.

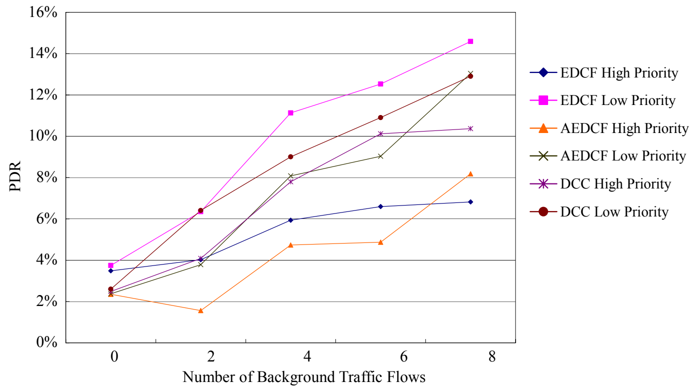


Fig. 12. Packet dropping ratio without high-priority guarantee.

On the other hand, the average packet delays (in seconds) of high-priority, low-priority, and all of traffic are illustrated in Fig. 11. The difference between DCC and either of the other two schemes is contrary to that observed in Fig. 9. Besides, it is demonstrated that the average delay for all of the packets with DCC (solid line with circles) is the smallest since our scheme can greatly lower the packet delay of low-priority traffic at the cost of slightly increasing the high-priority packet delay and can drop the packets that could be late at intermediate nodes. Fig. 12 shows the variation of packet dropping ratios (PDR) when the number of background traffic is increased. It is interesting to see that the PDR of low-priority traffic with EDCF (solid line with squares) is the largest while the PDR of high-priority traffic with DCC (solid line with asterisks) is apparently larger than those with the other two schemes due to dropping obsolete packets early.

4.2 Experiment: Scenario Two

Different from Scenario one, this scenario enables the decision made to guarantee high-priority traffic in the DCC algorithm. The packets for each AC-3/AC-2 flow are generated at a viable bit rate (VBR) with the average rate, burst time, and idle time being

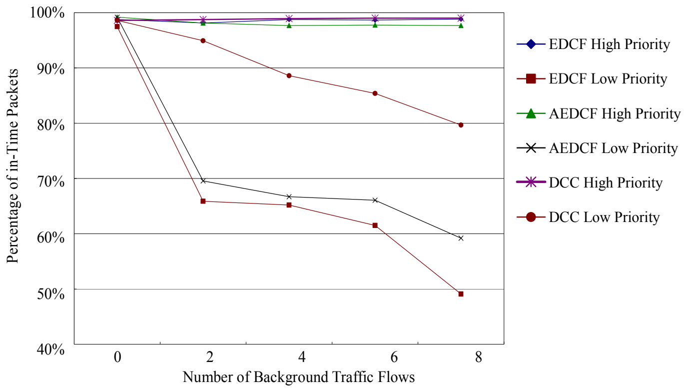


Fig. 13. Percentage of in-time packets with high-priority guarantee.

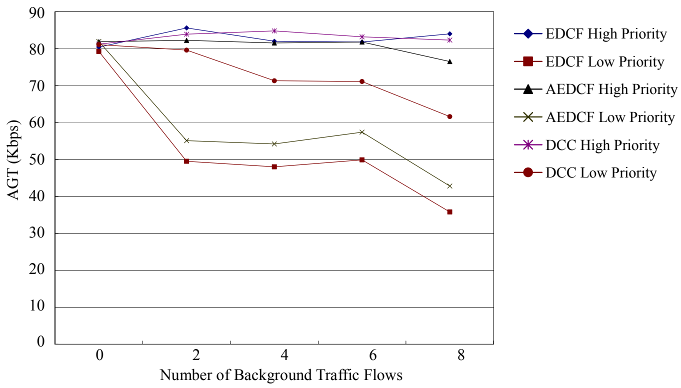


Fig. 14. Average good throughput with high-priority guarantee.

100 Kbps, 500 ms, and 50 ms, respectively, while those for each AC-1 flow are generated at a constant bit rate with the packet interval being 15ms. Fig. 13 shows the percentage of packets meeting end-to-end delay requirements for high- and low-priority traffic with respect to the number of background traffic flows. Contrasting Figs. 9 and 13, one can find from Fig. 13 that the percentage of in-time packets for high-priority traffic with DCC is adversely a little larger than the other two schemes, since in addition to providing QoS guarantees for high-priority traffic, the DCC scheme takes end-to-end delay requirements into account and develops a non-uniform random backoff timer to avoid continuous collisions. In Scenario two, the percentage of in-time packets for low-priority traffic with DCC remains the largest. However, the service differentiation between the high- and low-priority traffic in Scenario two becomes more distinctive than that in Scenario one because the former considers QoS guarantees for high-priority traffic. Alternatively, Fig. 14 shows the average good throughputs (AGT) in the case where only packets meeting end-to-end delay requirements are considered.

5. CONCLUSIONS

We have presented a modified EDCA with dynamical contention control (DCC) for real-time traffic in multi-hop ad hoc networks. With the proposed DCC scheme, a mobile node can estimate per-hop delays from the received ACK frames that are replied for the successfully transmitted frames. When route failure occurs, new per-hop delays can be estimated by monitoring RREQ and RREP messages issued by a reactive routing protocol. Before contending for the medium access to transmit a frame, a mobile node can dynamically raise the priority of the frame based on the estimated per-hop delay. Besides, we have designed a non-uniformly distributed random backoff timer for retransmitted frames according to their individual remaining end-to-end delays and residual hop counts to effectively avoid continuous collisions.

In order to guarantee high-priority traffic, there is an optional procedure made in the DCC algorithm to decide whether the priority of a frame is to be raised. From the simulation results, it is demonstrated that the proposed DCC scheme not only can reduce the average packet delay but also can increase the percentage of packets meeting end-to-end delay requirements. However, the service differentiation between the high- and low-priority traffic becomes more distinctive when DCC considers QoS guarantees for high-priority traffic. Although the minimum time period (*Tc*) to initiate dynamic priority adjustment is given and fixed in this paper, we are working on adaptively adjusting *Tc* according to the network conditions to achieve better QoS guarantees for prioritized traffic.

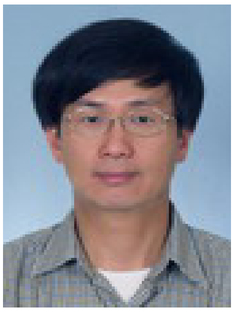
REFERENCES

1. G. S. Ahn, *et al*., "Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN)," *IEEE Transactions on Mobile Computing*, Vol. 1, 2002, pp. 192-207.
2. H. Zhu, *et al*., "A survey of quality of service in IEEE 802.11 networks," *IEEE Wireless Communications*, Vol. 11, 2004, pp. 6-14.
3. IEEE 802.11 – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
4. IEEE 802.11e – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), 2006.
5. D. Gu and J. Zhang, "QoS enhancement in IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, Vol. 41, 2003, pp. 120-124.
6. J. W. Robinson and T. S. Randhawa, "Saturation throughput analysis of IEEE 802.11e enhanced distributed coordination function," *IEEE Journal of Selected Areas in Communications*, Vol. 22, 2004, pp. 917-928.
7. I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proceedings of INFOCOM*, Vol. 1, 2001, pp. 209-218.
8. W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee, "Distributed mechanisms for quality of service in wireless LANs," *IEEE Wireless Communications*, Vol. 10, 2003, pp. 26-34.
9. A. Banchs, *et al*., "Applications and challenges of the 802.11e EDCA mechanism: an

experimental study," *IEEE Network*, Vol. 19, 2005, pp. 52-58.
10. J. Hui and M. Devetsikiotis, "A unified model for the performance analysis of IEEE 802.11e EDCA," *IEEE Transactions on Communications*, Vol. 53, 2005, pp. 1498-1510.
11. G. W. Wong and R. W. Donaldson, "Improving the QoS performance of EDCF in IEEE 802.11e wireless LANs," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Vol. 1, 2003, pp. 392-396.
12. L. Gannoune and S. Robert, "Dynamic tuning of the contention window minimum (CW_{min}) for enhanced service differentiation in IEEE 802.11 wireless ad hoc networks," in *Proceedings of IEEE Vehicular Technology Conference*, Vol. 4, 2004, pp. 2956-2961.
13. X. Wang, Q. Zhang, and X. Li, "Protocol enhancement for IEEE 802.11e EDCF," in *Proceedings of the 12th IEEE International Conference on Networks*, Vol. 1, 2004, pp. 80-84.
14. L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, Vol. 2, 2003, pp. 1373-1378.
15. Y. Wang and B. Huang, "Performance improvement of IEEE 802.11e," in *Proceedings of the IEEE Workshop on Local and Metropolitan Area Networks*, 2004, pp. 51-54.
16. H. Zhu, *et al.*, "EDCF-DM: a novel enhanced distributed coordination function for wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Communications*, Vol. 7, 2004, pp. 3886-3890.
17. C. Perkins, *et al.*, "Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications*, Vol. 8, 2001, pp. 16-28.
18. C. Perkins, *et al.*, "Ad hoc on-demand distance vector (AODV) Routing," IETF, RFC 3561, 2003.
19. The UCB/LBNL/VINT Network Simulator – ns (version 2), http://mash.cs.berkeley.edu/ns.

Yan-Jing Wu (吳妍靚) received the B.S. degree from the Department of Electronic Engineering, National Taiwan University of Science and Technology in 1992 and the M.S. degree from the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, in 1994. From Sep. 1995 to 2001, she taught at the Department of Electronic Engineering, I-Shou University, Kaohsiung County, Taiwan. She received the Ph.D. degree at the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, in Dec. 2006. In Feb. 2007, she became an assistant professor at the Department of Information Technology and Communication, Shih Chien University, Kaohsiung, Taiwan. Her research interests include wireless resource allocation and mobility management for multimedia traffic.

Jen-Hung Chiu (邱仁鴻) received the B.S. degree from the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan in 2002, and the M.S. degree from the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, in 2005. His research interest was in the area of wireless multi-hop ad hoc networks. From Jan. 2006, he has been working with the Electrical and Electronic Engineering Division in China Motor Corporation, Yang Mei, Taiwan. His major work is to develop a vehicle multimedia system and to integrate automotive electronic control units into Control Area Networks.



Tsang-Ling Sheu (許蒼嶺) received the Ph.D. degree in computer engineering from the Department of Electrical and Computer Engineering, Penn State University, University Park, Pennsylvania, USA, in 1989. From Sept. 1989 to July 1995, he worked with IBM Corporation at Research Triangle Park, North Carolina, USA. In Aug. 1995, he became an associate professor, and was promoted to full professor in Jan. 2006 at the Dept. of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan. His research interests include wireless mobile networks and multimedia networking. He was the recipient of the 1990 IBM Outstanding Paper Award. Dr. Sheu is a senior member of the IEEE, and the IEEE Communications Society.