

# A Scheme to Provide Proportionally Differentiated End-to-end Packet Delay in Wireless Multi-Hop Ad Hoc Networks

Dan Li<sup>1,2</sup> and Peng-Yong Kong<sup>1</sup>

<sup>1</sup> 21 Heng Mui Keng Terrace, Institute for Infocomm Research, 119613 Singapore.

<sup>2</sup> Electrical & Computer Engineering Department, National University of Singapore.

**Abstract.** This paper proposes a scheme to provide in a CSMA/CA based multi-hop wireless ad hoc network, a consistent and accurate proportional differentiation in average end-to-end packet delay. The proposed scheme, called PDMED uses a cross-layer approach that requires a distributed scheduler to adapt to the information from a QoS monitor, a route monitor and a channel monitor. Conceptually, the distributed scheduler dynamically adjusts the backoff duration of a flow based on its instantaneous deviation from the maximum average end-to-end packet delay. This is done such that a flow with a larger deviation from the maximum is given a longer backoff duration to give way to transmissions from other flows with smaller deviations. PDMED has been extensively evaluated through random event simulations using OPNET. The results confirm that it is capable of providing a consistent and accurate proportional differentiation, which is otherwise not achievable under various traffic conditions.

**Key words:** Proportional Differentiation, Multi-hop, Ad Hoc, End-to-end QoS.

## 1 Introduction

Wireless multi-hop ad hoc networks can be used to inter-connect various types of sensors without any pre-existing infrastructure. As a result of not relying on any existing infrastructure, multi-hop ad hoc networks have several salient and unique features. First, the network topologies are dynamic and changed often rapidly because of unpredictable and arbitrary movement of nodes. Also, the shared medium nature makes the availability of resource at one node being affected by its contending neighbors. Thus, node interconnectivity and link properties such as capacity and bit error rate cannot be pre-determined. Second, distance between the two ends of a link, obstacles in the environment, externally generated noise and interference caused by other transmissions will make the capacity of a wireless link reduced and apt to be highly variable. Therefore, the wireless link has a bandwidth-constrained and variable capacity. Third, multi-hop ad hoc networks are power-constrained because of lightweight batteries. The limited power supply limits the transmission range, data rate, communication activity and processing speed of the devices. Forth, the multi-hop networks need not have a centralized administration and thus, only local but not global

information is available to any node in the network. This implies distributed operations on every node are required.

Given the features presented above, multi-hop ad hoc networks suffer from resource constraints and operation vulnerability and therefore, quality of service (QoS) support in the network becomes a very demanding task [1]. Despite difficult, QoS provisioning in a multi-hop ad hoc network is unavoidable because sensor data do need timely delivery. For example, packets from an image sensor must be delivered real-time so that any illegal intruder can be detected immediately. Also, different types of sensor will require different QoS levels. For instance, packets from a temperature sensor that captures data once every 5 minutes should not be dropped in the presence of an instantaneous resource constrain compared to packets from an image sensor that generates a continuous stream of data.

IEEE 802.11 working group has taken the effort to define a standard mechanism to collectively adjust backoff duration and distributed inter-frame spacing (DIFS) to achieve efficient QoS differentiations [2]. The effort yields CSMA/CA based 802.11e protocol which has been extensively studied in the literatures [3], [4]. From the studies, controlling backoff duration is effective in introducing throughput differentiation while adjusting DIFS duration amplifies the differentiation. The studies also show that 802.11e can provide differentiation when there is a fixed number of active nodes within a radio range in an idealistic channel even though the traffic load is at a saturated level. However, the differentiation is vulnerable to changes in the number of nodes and traffic load. This vulnerability is partly due to the definition of its differentiation where a flow can choose one amongst a small number of service classes (or priorities) that best meet its QoS requirement, based on the assurance that the perceived QoS of higher classes will be better, or at least no worse than that of lower classes. This type of differentiation is called relative differentiation compared to proportional differentiation which offers predictable and controllable differentiations between different service classes [5].

For accurate proportional differentiation in terms of throughput, there exist various methods to map the virtual clock of a fair queuing model into the backoff duration of a CSMA/CA MAC protocol [6], [7]. Unfortunately, all these works can only achieve proportional differentiation locally or globally between two nodes over one hop. With multiple hops, the proportional differentiation should be achieved in an end-to-end manner across all hops but not limited to a concatenation of local proportional differentiations at each hop.

In order to provide QoS across multiple hops, [8] has proposed a distributed packet scheduling algorithm for CSMA/CA based MAC protocols to achieve an accurate transmission order as if in a centralized scheduler that provides QoS differentiation. Based on the desired transmission order, the scheduling algorithm assigns to every packet an appropriate priority. With the priority of a head packet, each node can rank itself against all its neighboring nodes after overhearing their head packets' priorities which are piggybacked on other transmissions. According to the rank, a node will determine its backoff duration to achieve the desired transmission order. Although the algorithm is capable of ensuring an accurate transmission order in a multi-hop setting, it is for packet and not flow. Further, there is no end-to-end performance objective.

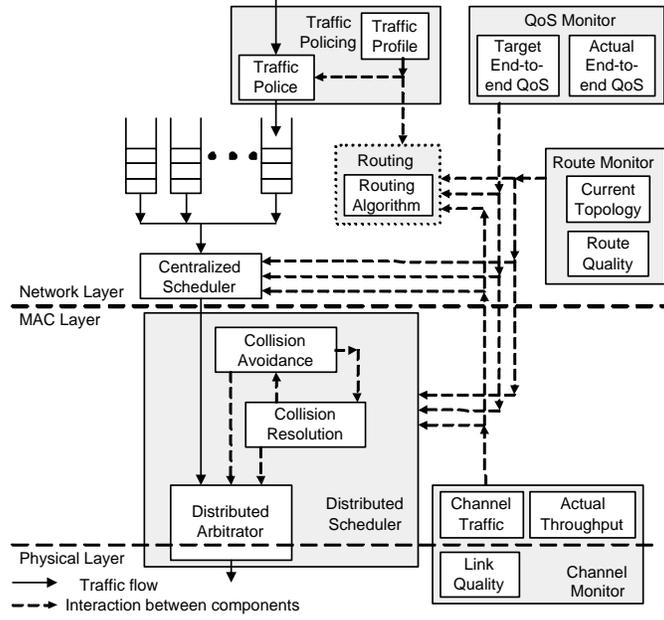
For different QoS to different flows across multiple hops, [9] proposes a coordinated multi-hop packet scheduling algorithm that requires some modifications to and co-operations from the CSMA/CA MAC protocol. In [9], the end-to-end QoS requirement of a flow is transformed into an instantaneous priority by the packet scheduling algorithm. Here, a packet that has not been offered sufficient service in the previous hop will be given a higher priority in the future hops and vice versa. The priority of the current and the next packets will be piggybacked onto RTS/CTS and DATA/ACK packets, respectively. Hence, all nodes within a hop know each other's instantaneous priorities and only the node with the highest relative priority will contend for the channel while the other nodes defer their own transmissions. It is the mechanism of adjusting a packet's priority at a hop based on its experience in previous hops that enables end-to-end QoS across multiple hops. The similar service compensation mechanism has been adopted by [10] for the same goal. More aggressively, [10] intends to provide a guarantee in end-to-end packet delay through admission control. Since there is no intuitive way to compute the capacity of a multi-hop ad hoc network, the admission control is done using an admit-then-test method. Specifically, a flow with end-to-end delay requirement is first admitted and then, its impact on the channel idle time is monitored. If the idle time becomes too short as a result of the new flow, another flow that has no end-to-end delay requirement is selected for rejection. Thus, an admitted flow may be dropped. Also, none of these schemes is capable of supporting the end-to-end proportional differentiations which are more controllable and predictable compared to other QoS offerings in a multi-hop ad hoc network.

We have learnt that there are numerous mechanisms across the protocol layers and time scales for QoS delivery in multi-hop ad hoc networks. Among these mechanisms are QoS routing protocols, admission control policies, resource reservation schemes, packet scheduling algorithms, QoS capable MAC protocols, etc. Unfortunately, none of these existing mechanisms is alone capable of providing satisfactory end-to-end proportional differentiations. Logically, a combination of these mechanisms have to work collaboratively to achieve the goal. For example, we may need a packet scheduling algorithm that transforms the QoS requirements into medium access priorities and works with a MAC protocol that provides the multiple priorities. Therefore, this paper contributes in developing a cross-layer scheme to provide proportional differentiation in end-to-end packet delay in wireless multi-hop ad hoc networks.

The remainder of this paper is organized as follows. Section 2 presents in details the proposed cross-layer scheme called Proportionally Differentiated Multi-hop End-to-end Delay (PDMED). The PDMED scheme has been evaluated through random event simulation using OPNET and simulation results are discussed in Section 3. The paper ends with concluding remarks in Section 4.

## 2 The PDMED Scheme

As illustrated in Fig. 1, PDMED consists of a traffic police, a routing algorithm, a centralized scheduler and a distributed scheduler. These are in turn assisted by a QoS monitor, a route monitor and a channel monitor. In this paper, we assume that all the traffic flows are self-disciplined such that no traffic policing is required. We further



**Fig. 1.** The cross-layer PDMED scheme

assume that all the nodes are not mobile and have a deterministic route quality so that the static shortest path routing protocol can be adopted. We also assume the use of CSMA/CA MAC protocol. This implies the collision avoidance function consists of RTS/CTS exchange and carrier sensing. Also, the collision resolution function is based on the paradigm that each flow has its own contention window size. Thus, collisions can be resolved by dynamically adjusting the contention window size based on which the backoff duration of a flow is determined. Let  $W_i$  be the contention window size of a flow  $i$ . Then, the backoff duration of a flow  $i$ ,  $\Delta_i$  in terms of number of discrete intervals is decided as follows:

$$\Delta_i = U[0, W_i - 1], \quad (1)$$

where  $U[x, y]$  is a function that generates random integer numbers within the range  $[x, y]$ . In (1),  $W_i$  is adjusted depending on the number of retransmissions,  $m$  the current flow  $i$ 's packet has experienced such that  $W_i = 2^m \times W_{min}$ , where  $W_{min}$  is the minimum contention window size of all flows. While  $W_i$  increases with the number of retransmissions, it is upper bounded by  $W_{max}$ . The adoption of CSMA/CA also means that the centralized scheduler is implicit. Specifically, with CSMA/CA, only the local flow that has finished first counting down its backoff duration can contend for medium access with the other flows from neighboring nodes.

With the assumptions given above, the task of providing an accurate end-to-end proportional differentiation falls mainly on a distributed scheduler which is presented next. We let the QoS be defined in terms of average end-to-end packet delay. Thus, the

target end-to-end QoS of the QoS monitor in Fig. 1 can be written as follows:

$$\frac{d_i(t)}{\phi_i} - \frac{d_j(t)}{\phi_j} = 0; \forall i, j, t, \quad (2)$$

where  $\phi_i$  is the proportional differentiation parameter and  $d_i(t)$  is the actual average end-to-end packet delay for flow  $i$  at time  $t$ . In practice,  $d_i(t)$  must be measured at the destination node of flow  $i$ . From the expression above, the target QoS can be interpreted as achieving among all flows an equality in their normalized end-to-end packet delays and the deviation of a flow  $i$  from the target QoS at time  $t$  can be quantified by  $\beta_i(t)$  as follows:

$$\beta_i(t) = \max_{\forall j/i} \left\{ \frac{d_j(t)}{\phi_j} \right\} - \frac{d_i(t)}{\phi_i}. \quad (3)$$

From the equation,  $\beta_i(t)$  is a positive real number where the smaller its value means closer it is to the QoS target, i.e.,  $\beta_i(t) = 0$ . Thus,  $\beta_i(t)$  is also used as the measurement for the actual QoS of flow  $i$  at time  $t$ .

In order to make  $\beta_i(t)$  as close as possible to its target value 0, we propose to dynamically adjust the backoff duration of a flow based on its instantaneous deviation from the equality such that a flow with a relatively smaller  $\beta_i(t)$  is given a shorter backoff duration to reduce its end-to-end packet delay. On the other hand, a flow with a relatively larger  $\beta_i(t)$  is given a longer backoff duration to give way to transmissions from other flows with a smaller  $\beta_i(t)$ . However, there is no intuitive best known method to perform the adjustment because of the following two problems: (a) The average end-to-end packet delay,  $d_i(t)$  that is measured at the destination node is not readily available to the intermediate nodes and source node of the flow, and (b) The normalized end-to-end packet delay of a flow is only known to the flow itself but the computation of  $\beta_i(t)$  requires the normalized delays of other contending flows.

Solving the two problems are the functions of the QoS monitor and channel monitor (refer to Fig. 1), respectively. In the QoS monitor, a backward propagation scheme is proposed so that  $d_i(t)/\phi_i$  computed at the destination node will be known by the flow's intermediate and source nodes. According to the backward propagation scheme, when a packet arrives at a flow  $i$ 's destination node at time  $t$ , its average end-to-end delay is updated as follows:

$$d_i(t) = \frac{\tau_i(t) + (n(t) - 1) \times d_i(t')}{n(t)}, \quad (4)$$

where  $\tau_i(t)$  is the end-to-end delay of the packet that arrives at time  $t$ ,  $n(t)$  is the total number of packets including the newly arrived one up to time  $t$ , and  $d_i(t')$  is the previous average packet delay. Through the updating process, the destination node always has the latest value of normalized average end-to-end packet delay, i.e.,  $d_i(t)/\phi_i$ . The latest value together with its respective flow identity will be piggybacked onto the MAC ACK frames that are transmitted in response to each successfully received MAC DATA frame of the flow. At the intermediate nodes, the piggybacked information will be extracted from the received MAC ACK frames and stored locally before being

similarly piggybacked onto the upcoming MAC ACK frames of the flow. As such, the actual normalized end-to-end packet delay of each flow can be propagated from the destination node to the source node. We notice that there will be a time lag between the computation of an instantaneous normalized average end-to-end delay and its arrival at the intermediate and source nodes. In practice, the extension of the time lag depends on the number of hops and its impact on the QoS target will be extensively studied through simulation in the next section.

In the channel monitor, a sniffer is proposed to read all the transmitted MAC ACK frames within a broadcast region. With the sniffer, each node can maintain a table containing the identities of all neighboring flows and their respective latest normalized average end-to-end delays. The table is updated each time a MAC ACK frame is received. With the up-to-date table,  $\beta_{i,k}(t)$ , i.e., the value of  $\beta_i(t)$  (refer to (3)) at the  $k$ -th hop of flow  $i$  can be computed as follows:

$$\beta_{i,k}(t) = \max_{\forall j \in \mathcal{I}_{i,k}/i} \left\{ \frac{d_j(t)}{\phi_j} \right\} - \frac{d_i(t)}{\phi_i}, \quad (5)$$

where  $\mathcal{I}_{i,k}$  is the set of flow  $i$ 's neighboring flows at its  $k$ -th hop. Based on the computed  $\beta_{i,k}(t)$ , flow  $i$  can rank itself among all its neighboring flows. Specifically, the flow will be given the rank  $\ell$  if its  $\beta_{i,k}(t)$  is the  $\ell$ -th highest among all the neighboring flows.

Let  $r_{i,k}$  be the rank of flow  $i$  at its  $k$ -th hop when it has a packet to transmit there but sense a busy channel. In case no ranking can be performed, the default value for  $r_{i,k}$  is unity. Also, let  $W_{i,k} = 2^{m_{i,k}} \times W_{min}$  be the flow's contention window size at its  $k$ -th hop when the packet is making the  $m_{i,k}$ -th retransmission attempting and  $m_{i,k} = 0$  for a fresh packet. Then, instead of using the original CSMA/CA method in (1), the distributed scheduler will decide a flow's backoff duration,  $\Delta_{i,k}$  as follows:

$$\Delta_{i,k} = \begin{cases} U[0, W_{min} - 1] + I_{r_{i,k} \geq 2} \times \gamma_{i,k} \times W_{min} & \text{if } m_{i,k} = 0, \\ U[0, \frac{W_{i,k}-1}{h_i}] + W_{i,k} \times \left( \frac{h_i-k}{h_i} + r_{i,k} - 1 \right) & \text{otherwise,} \end{cases} \quad (6)$$

where  $h_i$  is the total number of hops for flow  $i$  and it is provided to the distributed scheduler by the route monitor in Fig. 1. In (6), the term  $I_A$  is an indicator function defined as follows:

$$I_A = \begin{cases} 1 & \text{if } A, \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

and  $\gamma_{i,k}$  is a dynamic control parameter for flow  $i$  at its  $k$ -th hop. The control parameter has an initial value of unity and it is dynamically adjusted only for a fresh packet at time  $t$  based on the actual normalized average end-to-end delay as follows:

$$\gamma_{i,k} = \begin{cases} \gamma'_{i,k} + 1 & \text{if } 0 < \beta_{i,k}(t') < \beta_{i,k}(t) \\ \gamma'_{i,k} - 1 & \text{if } \beta_{i,k}(t) = 0 \text{ and } \gamma_{i,k} > 1 \\ \gamma'_{i,k} & \text{otherwise,} \end{cases} \quad (8)$$

where  $\beta_{i,k}(t')$  and  $\gamma'_{i,k}$  are the previous values of  $\beta_{i,k}(t)$  and  $\gamma_{i,k}$ , respectively.

Compare (6) to (1), we notice that PDMED scheme gives priority to a flow that experiences excessive normalized average end-to-end delay by allowing a smaller backoff duration. In order to ensure a high responsiveness,  $\gamma_{i,k}$  provides an additional degree of freedom when ranking and prioritization alone are not sufficient to quickly bring down a high excessive normalized delay. Also, PDMED gives priority to a retransmitted packet compared to a fresh packet. This is to avoid the situation where multiple packets from a same flow are contending with each other arbitrarily. Among all the retransmitted packets, based on the heuristic disclosed in [11], the packet that is closer to the destination node will be given the priority to transmit so that the overall end-to-end delay can be reduced.

### 3 Performance Evaluation

We have evaluated PDMED using OPNET. For the purpose of simulation, the general network topology as illustrated in Fig. 2 is used. In the network, there are only two flows, namely Flow 1 (S1-D1) and Flow 2 (S2-D2). From the figure, Flow 1 and Flow 2 have 3 and 2 hops, respectively. For the flows, their differentiation parameters are denoted by  $\phi_1$  and  $\phi_2$ , respectively.

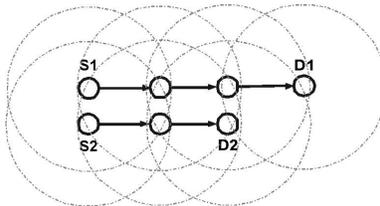
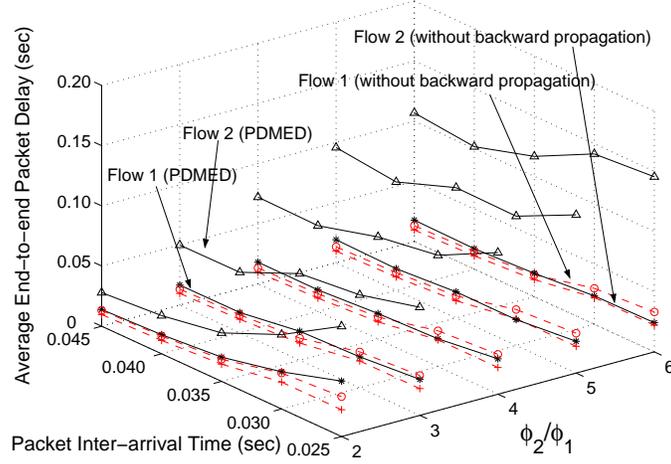


Fig. 2. Network topology

In the simulations, traffic for each flow is generated using a Poisson arrival process with a fixed packet size,  $L_m$  and a packet arrival rate,  $\lambda$ . Hence, the packet inter-arrival time is exponentially distributed with mean  $\lambda^{-1}$ . Hereafter,  $L_m$  is fixed at 500 bytes unless specified otherwise. In the evaluation, the raw bit rate of communication channel is 1 Mbps. Also,  $W_{min}$  and  $W_{max}$  are fixed at 16 and 1024 time slots, respectively. Here, the duration of each time slot,  $T_{slot} = 50\mu$  second.

First of all, we perform simulations to study the usefulness of the backward propagation scheme adopted by the QoS monitor to inform the nodes of a flow's instantaneous normalized end-to-end delay. Recall that the backward propagation is achieved by piggybacking the latest normalized average delay value onto the MAC ACK frames. We disable the piggybacking in some simulations and compare the results with those of PDMED. The comparison is depicted in Fig. 3 which shows the performance in terms of average end-to-end packet delay. The delay of a packet is the time elapsed since the packet's arrival at the MAC layer of its source node until the packet's subsequent arrival at the MAC layer of its destination node. These packets

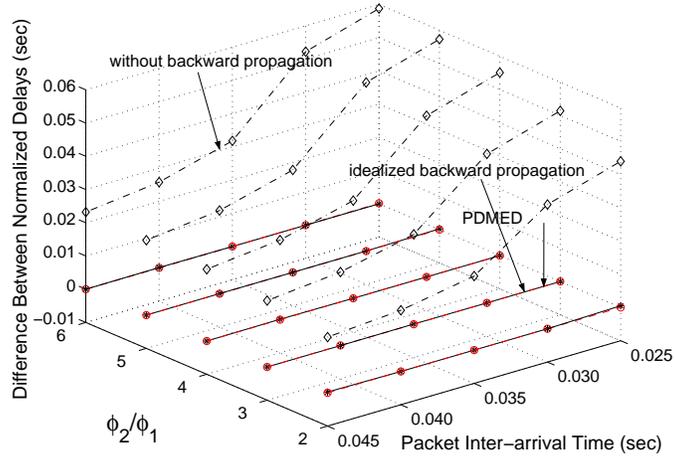


**Fig. 3.** Average end-to-end packet delay with and without the backward propagation scheme

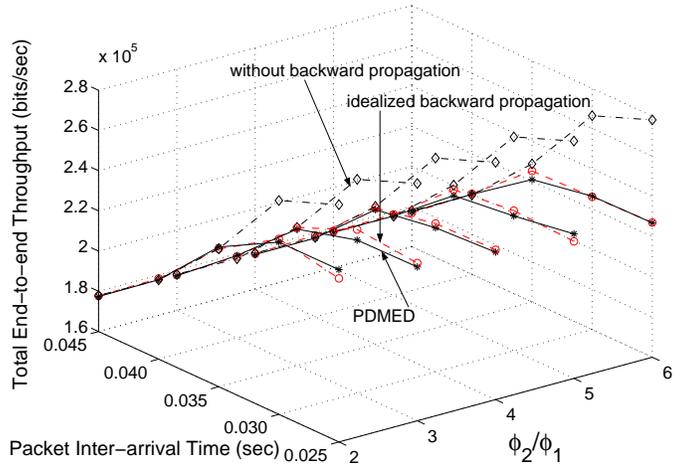
from their respective traffic sources are queued above but not in the MAC layer to avoid distortion in packet delay at high traffic rate,  $\lambda^{-1}$  when the delays of all flows increase exponentially making any difference in their values not noticeable. In Fig. 3, different  $\phi_2/\phi_1$  ratios are achieved by fixing  $\phi_1$  at 1 while varying  $\phi_2$ . The results show that PDMED can indeed provide a proportional differentiation in average end-to-end packet delay despite that the flows are going through different numbers of hops. When there is an increase in  $\phi_2/\phi_1$ , the proportional differentiation is indicated by a rapid increase in Flow 2's end-to-end delay and a slow decrease in Flow 1's end-to-end packet delay although Flow 2 has fewer hops compared to Flow 1. Also, the delays of Flow 1 and Flow 2 increase and keep a fixed differentiation ratio with respect to an decrease in  $\lambda^{-1}$ .

Fig. 3 has confirmed the importance of the backward propagation scheme because, without it, the difference between the two flow's delays is not obvious at various  $\phi_2/\phi_1$  ratios. This is further verified in Fig. 4 where the difference between the two flow's normalized average end-to-end packet delay is plotted. Ideally, the difference should be zero because, as stated in (2), the performance goal is to achieve equality in the normalized delays. From Fig. 4, PDMED can indeed approximate the performance goal regardless of the traffic rate and  $\phi_2/\phi_1$  ratio. On the other hand, the performance goal is not achievable when there is no backward propagation. This happens because, in the absence of the backward propagation, the intermediate nodes do not know the actual end-to-end delay and thus, cannot adjust its backoff duration appropriately to meet the performance goal.

In the evaluation above, the backward propagation scheme is disabled by simply not piggybacking the computed normalized delay on ACK frames. While this leads to a failure in accurate proportional differentiation, there is a noticeable gain in total end-to-end throughput of the two flows as depicted in Fig. 5. This is because, without the instantaneous normalized delay, an intermediate node cannot correctly compute

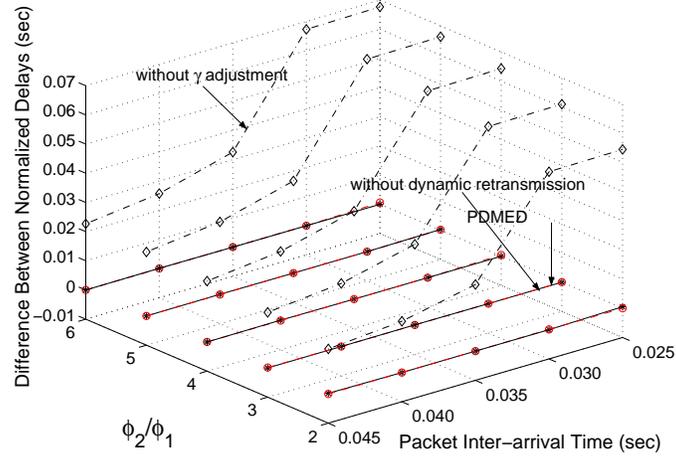


**Fig. 4.** Difference in normalized end-to-end packet delays with and without the backward propagation scheme



**Fig. 5.** Total end-to-end throughput as measured at the respective destination nodes, with and without the backward propagation scheme

$\beta_{i,k}(t)$  according to (5) and consequently, will not perform the ranking mechanism and adjust  $\gamma_{i,k}$  according to (8). Without the ranking and adjustment,  $r_{i,k}$  and  $\gamma_{i,k}$  stay at their default values of unity. Thus, the backoff duration will always be selected from a range upper bounded by  $W_{min} - 1$  compared to a potentially much larger range adjusted by ranking and  $\gamma_{i,k}$  according to (6). The smaller backoff duration is the cause of the better end-to-end throughput when there is no backward propagation. In the presence of backward propagation, we treat the reduction in throughput as the cost to pay for the accurate proportional differentiation.

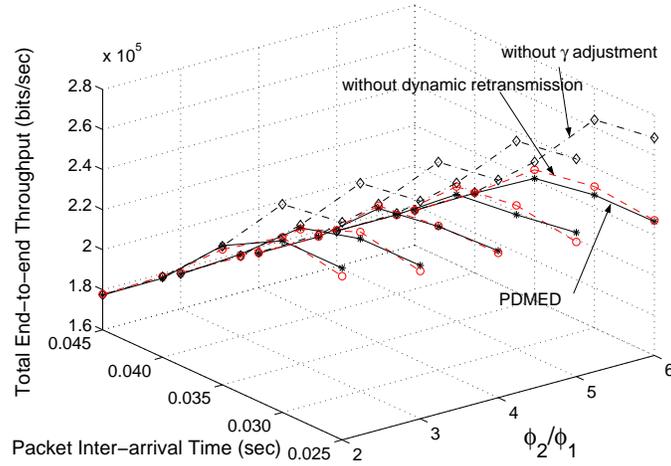


**Fig. 6.** Difference in normalized end-to-end packet delay with and without the  $\gamma_{i,k}$  adjustment and the dynamic retransmission

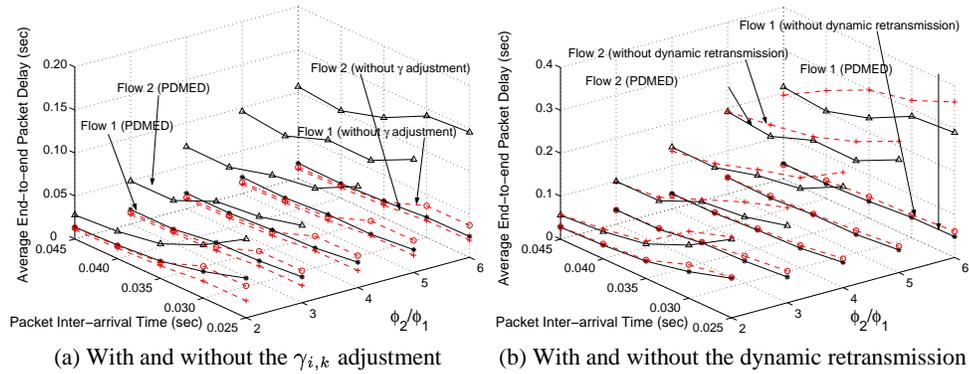
The ranking in PDMED may not always be based on the latest instantaneous normalized delay because the backward propagation scheme takes time to distribute the delay across multiple hops after it is computed at the destination node. Specifically, there is always a time lag before the latest normalized delay is available at an intermediate node. Fortunately, this time lag has no significant impact in achieving an accurate proportional differentiation in average end-to-end delay as illustrated in Fig. 4. In the figure, there is no obvious difference in performance when PDMED is equipped with an idealized backward propagation scheme. Compared to the original scheme, the idealized scheme does not require piggybacking of the latest delay on ACK frames. Instead, the simulation program makes the delay known to all the intermediate nodes as soon as it is computed. Without piggybacking, the idealized propagation scheme consumes less bandwidth. However, as shown in Fig. 5, there is no obvious throughput difference between the original and idealized back propagation schemes. This implies the backoff propagation scheme is efficient as it introduces only very small overhead.

Thus far, we have shown the importance and effectiveness of the backward propagation scheme in PDMED. In short, the backward propagation is needed so that intermediate nodes can obtain the instantaneous normalized delay for ranking and  $\gamma_{i,k}$  adjustment to achieve an accurate proportional differentiation. Next, we want to show that the ranking itself, without  $\gamma_{i,k}$  adjustment is not sufficient. For this purpose, we have repeated the simulations after disabling the adjustment algorithm in (8). Fig. 6 shows the difference between the two flows' normalized average end-to-end packet delay. Compared to PDMED, the larger difference indicates a less accurate proportional differentiation when there is no  $\gamma_{i,k}$  adjustment. This means the ranking mechanism alone is not enough in the channel monitor.

Although the absence of  $\gamma_{i,k}$  adjustment cannot produce an accurate proportional differentiation, it results in a higher total end-to-end throughput as illustrated in Fig. 7.



**Fig. 7.** Total end-to-end throughput as measured at the respective destination node with and without the  $\gamma_{i,k}$  adjustment and the dynamic retransmission



**Fig. 8.** Average end-to-end packet delay

Refer to (6), this is because the backoff duration tends to be smaller when  $\gamma_{i,k}$  is not dynamically adjusted but fixed at its initial value of unity. The better throughput without  $\gamma_{i,k}$  adjustment also leads to a lower end-to-end packet delay as illustrated in Fig. 8(a). Despite a lower delay, when there is no  $\gamma_{i,k}$  adjustment, the difference in delay does not follow the  $\phi_2/\phi_1$  ratio and thus does not constitute an accurate proportional differentiation. This is not the case in Fig. 8(b) where we show the impact of the dynamic retransmission scheme in PDMED. As given in (6), a retransmission is indicated by  $m_{i,k} > 0$  and the dynamic retransmission scheme gives higher priority to transmissions from a node closer to a flow's destination node. As such, PDMED can deliver a smaller end-to-end delay compared to the case without the dynamic retransmission scheme. The simulations without the retransmission scheme have been

performed by simply selecting the backoff duration, i.e.,  $\Delta_{i,k}$  in (6) from the range  $[0, W_{i,k} - 1]$  when capable of reducing end-to-end delay, it does not compromise the accuracy of proportional differentiation and total throughput as illustrated in Fig. 6 and Fig. 7, respectively.

We have evaluated PDMED under various other conditions and benchmarked against IEEE 802.11e using video traces. However, these results are not presented here due to space limitation.

## 4 Conclusions

Noticing the lack of support in providing proportional differentiation in end-to-end packet delay in a wireless multi-hop ad hoc network, this paper proposes PDMED to do so. PDMED consists of a few mechanisms and monitors which operate across different protocol layers and time scales. PDMED has been extensively evaluated through random event simulation. The results indicate that an accurate and consistent proportional differentiation in end-to-end packet delay which cannot be achieved otherwise, can now be achieved.

## References

1. M. S. Corson, "Issues in supporting quality of service in mobile ad hoc networks", *IFIP 5th Int. Workshop on Quality of Service (IWQOS'97)*, May 1997.
2. M. Benveniste, G. Chesson, M. Hoehen, A. Singla, H. Teunissen and M. Wentink, "EDCF proposed draft text", *IEEE working document 802.11-01/131-rl*, March 2001.
3. J. Kim and C. Kim, "Performance analysis and evaluation of IEEE 802.11e EDCF", *Wireless Communications and Mobile Computing*, Vol. 4, No. 1, pp. 55-64, February 2004.
4. B. Li and R. Battiti, "Performance analysis of an enhanced IEEE 802.11 distributed coordination function supporting service differentiation", *QoS LNCS 2811*, pp.152-161, 2003.
5. C. Dovrolis, D. Stiliadis and P. Ramanathan, "Proportional differentiated services: Delay differentiation and packet scheduling", *IEEE/ACM Trans. Networking*, Vol. 10, No. 1, pp. 12-26, February 2002.
6. H. Luo, S. Lu, V. Bharghavan, J. Cheng and G. Zhong, "A packet scheduling approach to QoS support in multi-hop wireless networks", *Mobile Network and Applications*, Vol. 9, No. 3, pp. 193-206, June 2004.
7. A. K. Somani and J. Zhou, "Achieving fairness in distributed scheduling in wireless ad-hoc networks", *IEEE IPCCC*, pp. 95-102, April 2003.
8. V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi and E. Knightly, "Ordered packet scheduling in wireless ad hoc networks: Mechanisms and performance analysis", *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 58-70, 2002.
9. V. Kanodia, C., Li, A. Sabharwal, B. Sadeghi and E. Knightly, "Distributed priority scheduling and medium access in ad hoc networks", *ACM/Baltzer Wireless Networks*, Vol. 8, No. 5, pp. 455-466, September 2002.
10. Y. Yang and R. Kravets, "Distributed QoS guarantees for realtime traffic in ad hoc networks", *IEEE SECON*, 2004.
11. B. G. Chun and M. Baker, "Evaluation of packet scheduling algorithms in mobile ad hoc networks", *Mobile Computing and Communications Review*, Vol. 1, No. 2, June 2002.