A QoS Tracking Algorithm for Multimedia Requirements over IEEE 802.11e Multihop Networks

Ju Yong Lee*, Seung Jun Lee†, Jang Keun Jeong†, Jitae Shin†, and Shahrokh Valaee‡

* KAIST Institute for Information Technology Convergence, KAIST, Daejeon, Korea
† School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea
‡ Dept. of ECE, University of Toronto, Toronto, Canada

***Abstract*—In this paper, we propose a QoS tracking algorithm to support different QoS requirements to accommodate delay-sensitive and throughput-sensitive traffics in wireless mesh networks. The proposed scheme is carried out in a totally distributed manner by adjusting the minimum contention window size. Moreover, we investigate whether we can satisfy all QoS requirements in a stable manner through stability analysis. The simulation results show that the proposed algorithm can satisfy both average delay bound and minimum throughput requirements without sacrifice of total system efficiency.**

***Index Terms*—QoS, Multi-hop, Wireless LAN, Wireless mesh network**

I. Introduction

Multimedia streaming applications over wireless networks need a certain level of quality-of-service (QoS) requirement like delay and throughput. QoS in wireless local area networks (WLANs) is related with the control parameters of the medium access control (MAC) mechanism such as the distributed coordination function (DCF) of IEEE 802.11 [1] and Enhanced Distributed Channel Access (EDCA) of IEEE 802.11e [2], which is adopted as the mandatory MAC functions in IEEE 802.11s known as a standard for wireless mesh networks (WMNs). The main MAC parameters are minimum/maximum Contention Window (CW) (shown as CW_{min} and CW_{max}), Arbitrary Inter-frame Space (AIFS), and Transmission Opportunity (TXOP) that apply different value settings to different priority queues, i.e. access category (AC) queues in IEEE 802.11e, within each node in order to provide QoS differentiation gaps among AC queues.

It has been shown in the literature that an appropriate tuning of the IEEE 802.11 MAC parameters can significantly increase the protocol efficiency; refer to [3, 4] for general overview of QoS provisioning in IEEE 802.11. Among directly adjustable parameters per each AC queue, (i.e., CW_{min}, CW_{max}, and $AIFS$) [5, 6] show that CW_{min} is more appropriate for service differentiation for high priority traffic flows with hard QoS requirement such as guaranteed throughput, while CW_{max} and $AIFS$ are more appropriate for moderate service differentiation. In this paper, we focus on CW_{min} adaptation to track different QoS requirements in different AC queues – given end-to-end delay and minimum throughput – in simple but robust ways.

When nodes using IEEE 802.11 DCF are concatenated and a packet traverses multiple nodes, serious unfairness can occur [7]. This problem may decrease throughput and increase end-to-end delay due to the unbalanced channel usage. Recently, there have been some work for resource management in multihop WLAN. Lim *et al.* [8] showed that enhancement of the end-to-end throughput on multihop wireless path can be obtained by adjusting CW_{min}. They changed CW_{min} proportional to the difference between the outgoing rate and the incoming rate. Although throughput unfairness can be overcome in this way, it cannot support explicit QoS requirements such as the end-to-end delay. The APHD scheme [9] is a proposed method that chooses adaptively one of the AC queues with different MAC parameters in all the forwarding nodes in order to satisfy the end-to-end delay. Unfortunately this technique needs complex network-wide monitoring and computations. In effect, most existing schemes need complex algorithms that frequently require information and coordinations from neighboring nodes. Briefly, for QoS differentiation, many schemes such as [5, 10, 11] were proposed to use differently predetermined or dynamic settings of the MAC parameters per AC queue. Despite the complexity of these algorithms, they can only provide differentiated QoS, which may not be suitable for delay-/throughput-sensitive traffics such voice-over-IP (VoIP) and video-on-demand (VoD). But for QoS guarantee, there are usually needed heavy signal exchanges and strong coordination like admission control among wireless mesh nodes [9, 12, 13].

To our best knowledge, there is only few research efforts to satisfy QoS requirements by adaptively adjusting the MAC parameters. Yang *et al.* [14, 15] proposed a distributed optimal contention window control technique that does not impose control message overhead and does not depend on explicit knowledge of channel capacity.

Along with Yang's distributed concept, we will propose an algorithm to satisfy QoS requirements in multi-hop wireless LAN environment by adapting one of the MAC parameters, CW_{min}. Our objective is to satisfy end-to-end delay for delay-sensitive traffic like VoIP and to guarantee minimum bandwidth for throughput-sensitive traffic like VoD at same time. We identify two types of users: *satisfied* and *unsatisfied*. Satisfied users are the ones that meet their QoS requirements, and unsatisfied users are the ones for which QoS requirements are not met. In the proposed algorithm the minimum congestion window, CW_{min}, of satisfied users is enlarged and the minimum congestion window of unsatisfied users is shortened. Our mechanism is operated in a totally distributed manner, with each queue making an independent decision whether to increase or decrease the minimum window size. Throughput-sensitive traffic users may require minimum bandwidth. However, if there is available bandwidth, they may want to receive more bandwidth. Therefore, delay-sensitive traffic may be affected by throughput-sensitive traffic. In numerical results, we show that our algorithm can meet both delay requirements and throughput requirements more effectively than the previous algorithm [9].

The rest of the paper is organized as follows. In Section II, we propose our tracking algorithm to satisfy the QoS requirements by adjusting CW_{min}. Section III investigates if we can reach the feasible state in a stable manner. We compare our algorithm with the previous algorithm [9] in Section IV. Finally, Section V concludes this paper.

II. MAC Control Parameter Adaptation

We consider wireless mesh networks consisting of multiple WLAN nodes [17]. If the signal can be decoded at a receiver node, the node is in the transmission range of the sender node. The receiver is in the interference range if the signal can affect the receiver by reducing the capacity due to interference.

We assume that each node has multiple priority queues. Different priority queues may have different MAC parameter, CW_{min}. Let $\mathbf{W}$ be a vector of $(W_1, W_2, \cdots, W_n)$, where n is the number of queues in the interference range and W_i is the CW_{min} of the i-th queue. Let $\mathcal{D}$ and $\mathcal{T}$ be the set of queues for delay-sensitive traffic and throughput-sensitive traffic, respectively. Our objective is to allocate adequate resources so that all QoS requirements are satisfied for each priority queue at each node. For delay-sensitive traffic, the QoS measured in terms of the average delay is given as:

$$d_i \leq \hat{d}_i, \quad \forall i \in \mathcal{D}, \tag{1}$$

where d_i and $\hat{d}_i$ are the one-hop average delay and its target value in the i-th queue, respectively. For throughput-sensitive traffic, the QoS requirement is given as:

$$r_i \geq \hat{r}_i, \quad \forall i \in \mathcal{T}, \tag{2}$$

where r_i and $\hat{r}_i$ are the throughput and required minimum throughput of the i-th queue, respectively.

Using the above definitions, we can identify two types of queues. For a queue $i \in \mathcal{D}$, if (1) holds, we call it a "satisfied" queue. Otherwise, if (1) does not hold, we call it an "unsatisfied" queue. Similar definitions hold for queues in $\mathcal{T}$. Let us represent the set of all satisfied queues by $\mathcal{S}$ and the set of all unsatisfied queues by $\mathcal{U}$. Note that $\mathcal{S}$ and $\mathcal{U}$ may contain queues belonging to $\mathcal{D}$ or $\mathcal{T}$.

Suppose that the system starts at time $t = 0$. Let $d_i(k)$ and $r_i(k)$ be the measured average one-hop delay and the measured throughput of the i-th queue during $[(k-1)T, kT)$, $k = 1, 2, \ldots$, respectively. For delay-sensitive traffic, if $d_i(k)$ is less than the delay requirement, $\hat{d}_i$, it is included in $\mathcal{S}(k)$. Otherwise, it is in $\mathcal{U}(k)$. Also, for throughput-sensitive traffic, if $r_i(k)$ is greater than the minimum throughput requirement, $\hat{r}_i$, it is included in $\mathcal{S}(k)$. Otherwise, it is in $\mathcal{U}(k)$.

We adjust the minimum contention window dynamically to satisfy the QoS requirements. Let $\mathbf{W}(k)$ be $(W_1(k), W_2(k), \ldots, W_n(k))$, where $W_i(k)$ is the minimum contention window of the i-th queue at updated time interval k. Based on the estimated performance $d_i(k)$ and $r_i(k)$, we update $W_i(k)$ as follows:

$$W_i(k+1) = \begin{cases} W_i(k)(1+\epsilon) & \text{if } i \in \mathcal{S}(k), \\ W_i(k)(1-\epsilon) & \text{if } i \in \mathcal{U}(k), \end{cases} \tag{3}$$

where ϵ is a small positive step size. If the i-th queue is included in $\mathcal{S}(k)$ in the k-th interval, $W_i(k+1)$ is greater than $W_i(k)$ to reduce the allocated channel resource. On the other hand, if the QoS requirement of the i-th queue is not satisfied for the k-th interval, $W_i(k+1)$ is less than $W_i(k)$. Each queue controls the channel resources to satisfy the QoS requirements by updating $W_i(k)$ at every interval of length T. Note that our algorithm manages the channel resources in a totally distributed manner.

III. Stability Analysis

In this section, we investigate whether we can satisfy all QoS requirements using the proposed algorithm in a stable manner. We define the feasible region, $\mathcal{M}$ as follows:

$$\mathcal{M} = \{\mathbf{W} | d_i(\mathbf{W}) \leq \hat{d}_i, i \in \mathcal{D} \;\text{ and }\; r_i(\mathbf{W}) \geq \hat{r}_i, i \in \mathcal{T}\}. \tag{4}$$

Note that if $\mathbf{W}$ is in the set $\mathcal{M}$, all QoS requirements for delay-sensitive traffic and throughput-sensitive traffic are satisfied. Our goal is to find a vector $\mathbf{W}$ in $\mathcal{M}$.

Let B be the set of backlogged queues. In [15], the actual throughput of the i-th queue at time t, c_i is approximated as:

$$c_i \approx C_i \frac{L_i / W_i}{\sum_{j \in B} L_j / W_j}, \tag{5}$$

where C_i is an effective capacity, which is adopted from [15] and is given statically as successful channel transmission, of the i-th queue and L_i is the transmission rate at the physical layer multiplied by the duration of a successful transmission (including the DIFS and RTS/CTS/DATA/ACK handshake) at node.

Now, we examine the dynamics of (3). If ϵ is sufficiently small, we can obtain the following:

$$\frac{1}{W_i(k+1)} = \begin{cases} \frac{1}{W_i(k)}(1-\epsilon) & \text{if } i \in \mathcal{S}(k), \\ \frac{1}{W_i(k)}(1+\epsilon) & \text{if } i \in \mathcal{U}(k). \end{cases} \tag{6}$$

Suppose that $i \in \mathcal{U}(k)$. Then, it follows from (5) and (6) that

$$
\begin{aligned}
& c_i(k+1) && (7)\\
&= C_i \frac{L_i/W_i(k+1)}{\sum_{j\in B} L_j/W_j(k+1)} \\
&= C_i \frac{\frac{L_i}{W_i(k)}(1+\epsilon)}{\sum_{j\in B, j\in \mathcal{S}} \frac{L_j}{W_j(k)}(1-\epsilon) + \sum_{j\in B, j\in \mathcal{U}} \frac{L_j}{W_j(k)}(1+\epsilon)} \\
&> C_i \frac{L_i/W_i(k)}{\sum_{j\in B} L_j/W_j(k)} = c_i(k). && (8)
\end{aligned}
$$

It means that allocated bandwidth increases if the QoS requirements are not satisfied. Therefore, we can infer that if $i \in \mathcal{U}(k)$, then

$$d_i(k) > d_i(k+1) \quad \text{for } i \in \mathcal{D}, \qquad (9)$$

$$r_i(k) < r_i(k+1) \quad \text{for } i \in \mathcal{T}. \qquad (10)$$

We define a discrete-time Lyapunov function, $V(k)$ as follows:

$$V(k) = \sum_{i\in\mathcal{D}, i\in\mathcal{U}(k)} (d_i(k) - \hat{d}_i)^2 + \sum_{i\in\mathcal{T}, i\in\mathcal{U}(k)} (\hat{r}_i - r_i(k))^2. \qquad (11)$$

Then, $V(k) \geq 0$. Suppose that $\mathbf{W}(k) \in \mathcal{M}$. Then, all i is in $\mathcal{S}(k)$. Thus, $V(k) = 0$. If $\mathbf{W}(k) \notin \mathcal{M}$, there exists i such that $i \in \mathcal{U}(k)$. Thus, $V(k) > 0$, and it follows from (9), (10) and (11) that $V(k) > V(k+1)$. Therefore, the feasible region, $\mathcal{M}$ is a globally asymptotically stable region [20].

In practice, we measure $d_i(k)$ and $r_i(k)$. Therefore, there exists a measurement error, which results in performance fluctuation. The step size ϵ also affects performance fluctuation and convergence speed. As ϵ increases, performance fluctuation is high and the system converges fast.

IV. Simulation Results

In this section, we present simulation results to validate our algorithm. We consider the wireless mesh network in University of Toronto as shown in Fig. 1, which is a heterogeneous network accommodating delay-/throughput-sensitive and background traffics. There are 50 wireless mesh point (MP) nodes.

We use the NS-2.31 [19]. The UDP protocol is used for transport layer and the 802.11e module in the NS simulator has been modified for MAC layer. We use IEEE 802.11a for the PHY model. Thus, transmission rate in the physical layer is $54Mbps$. We fix the packet size as $1000bytes$. Accordingly, we can obtain the maximum single hop transmission rate, $33Mbps$ in the application layer. All mesh nodes are assumed to be in the same interference range. Transmission occurs only among neighboring MP nodes and transmission paths are fixed.

There are two types of QoS flows. Arrow or dotted arrows denote the path of delay-sensitive traffic flows or throughput-sensitive traffic flows, respectively. Delay-sensitive traffic has an end-to-end average delay bound as the QoS requirement. All queues in the nodes that delay-sensitive traffic flows transverse have $20msec$ hop-by-hop delay requirements. We assume that the throughput-sensitive traffic has a minimum

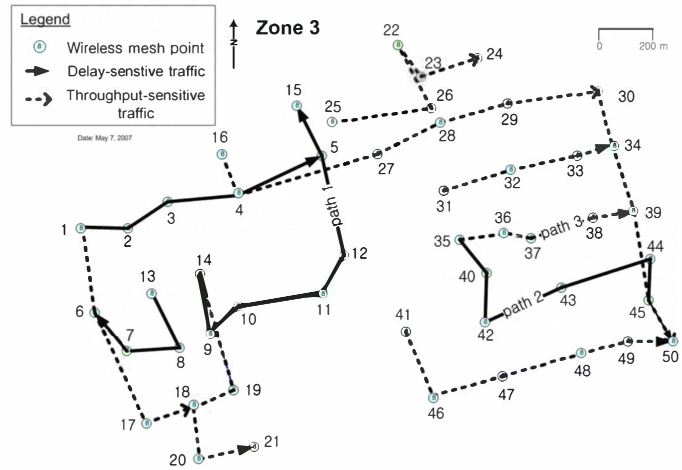


Fig. 1. Wireless mesh network in University of Toronto

TABLE I
QoS traffic flows

path	route
1 (delay-sensitive)	14 → 9 → 10 → 11 → 12 → 5 → 15
2 (delay-sensitive)	35 → 40 → 42 → 43 → 44 → 45 → 50
3 (throughput-sensitive)	35 → 36 → 37 → 38 → 39

bandwidth guarantee as the QoS requirement. The queues for throughput-sensitive traffic have $1Mbps$ bandwidth requirements. The throughput-sensitive traffic is assumed to have a high priority. Delay-sensitive traffic flows are generated with 200 superposed ON/OFF sources, where ON and OFF intervals have exponential distributions. The expected values of the ON and OFF intervals are $0.4sec$ and $0.6sec$, respectively. The transmission rate during the ON intervals is $64kbps$. Generally, the throughput-sensitive traffic may generate more than its requirements, and users expect to receive more bandwidth if there is available capacity. Therefore, throughput-sensitive traffic flows are generated with constant bit rate (CBR) with $2Mbps$, while their requirements are $0.5Mbps$. In all MP's, there exists a background traffic, which is two superimposed ON/OFF traffics. The ON and OFF intervals of the background traffic have Pareto distribution, where the mean and the shape parameter of the ON intervals are $0.1sec$ and $1.5sec$, and those for the OFF intervals are $0.9sec$ and $1.5sec$. The transmission rate during the ON intervals is $1Mbps$. We assume that the queue size is infinite. Paths 1 and 2 are for delay sensitive flows, and Path 3 is for throughput-sensitive flows. Table I describes routes of three paths.

In the APHD scheme in [9], each MP supports multiple queues as in the IEEE 802.11e. Delay-sensitive traffic, throughput-sensitive traffic and background traffic are severed in different queues. We assume that throughput-sensitive traffic is served with high priority. Accordingly, $CW_{min} = 8$ and $CW_{max} = 16$. On the other hand, in our algorithm, each queue competes to access wireless medium with different MAC parameter values. We update CW_{min} at each beacon interval, $100msec$ according to (3). We set the initial value of CW_{min} at 32 and the range of CW_{min} is from 8 to 128. If the range of CW_{min} variations is sufficiently large, from (5), we can support a differentiated bandwidth with high granularity. However, if CW_{min} is too large, bandwidth can be wasted. We set the maximum contention window, CW_{max} as $2^5 \times CW_{min}$, where the maximum backoff stage is 5. The step size, ϵ is fixed at 0.1. If ϵ is too small, adaptation may be slow, and if ϵ is too high, high fluctuation can occur.

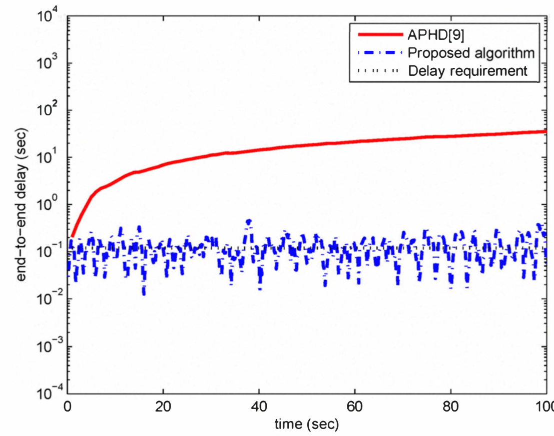
Fig. 2 compares the performance APHD and the proposed algorithm. The average throughput is depicted using:

$$\bar{r}_i(k) = 0.1 * r_i(k) + 0.9 * \bar{r}_i(k-1) \quad (12)$$

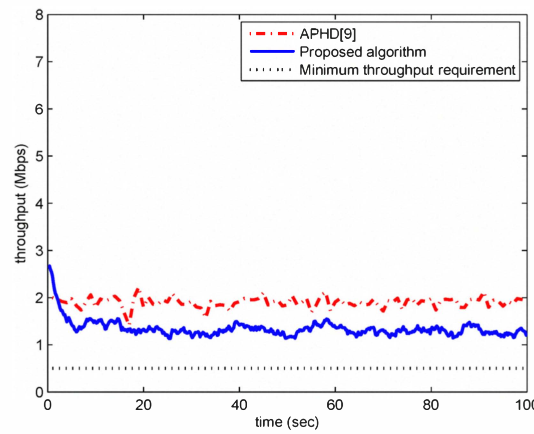
where $r_i(k)$ is the average throughput in the k-th interval. In APHD, if the delay budget in the current node is not satisfied, the packet will be served with the high priority queue in the next node. Even if the algorithm uses multiple queue schemes for supporting QoS, all flows contend with each other since delay-sensitive flows and throughput-sensitive flows are served with high priority. Therefore, the resources for the delay-sensitive traffic may be reduced by heavy throughput-sensitive traffic flows. Thus, the hop-by-hop delay for the delay-sensitive traffic flow increases severely as shown in Fig. 2(a). We need to regulate the high priority traffic to protect the delay-sensitive traffic. In our algorithm, the minimum bandwidth of throughput-sensitive traffic, 0.5*Mbps*, can be guaranteed. The channel resource that the throughput-sensitive traffic occupies is reallocated to the delay-sensitive traffic. Thus, the end-to-end average delay requirement of the delay-sensitive traffic, 120*msec*, can be met in average sense (Fig. 2(a)).

Fig. 3 represents the variation of CW_{min} of the queues for the delay-/throughput-sensitive traffic in MP 35 when the proposed algorithm is applied. The CW_{min} values of the delay-sensitive traffic flow fluctuate according to whether per-node delay requirement is satisfied or not. On the other hand, the CW_{min} values of the throughput-sensitive traffic flows increase since the throughput satisfies the minimum throughput requirement. By differentiating CW_{min}, more channel resources can be allocated to the delay-sensitive traffic with relatively small CW_{min}. The end-to-end delay of the delay-sensitive traffic satisfies the QoS requirement with an adequate CW_{min}.

Finally, we investigate the number of the delay-sensitive traffic users that each scheme can accommodate. Fig. 4 represents the end-to-end delay as the number of ON/OFF sources in flow 1 and 2 increases. For the APHD scheme, when the number of users approaches 80, the average end-to-end delay increases severely, which means APHD cannot accommodate more than 80 users. By increasing the number of users of flow 1 and 2, the network capacity is insufficient and the average end-to-end delay for the delay-sensitive traffic can increase. We observe that our proposed algorithm can accommodate additional 30 users and 70 users for path 1 and 2, respectively, as compared with the APHD scheme. Path 2



(a) End-to-end delay of the flows in path 2



(b) Throughput of the flows in path 3

Fig. 2. Performance comparison of APHD and the proposed scheme [9]

has less interfering delay-sensitive traffic than path 1. Thus, path 2 can accommodate more delay-sensitive traffic users.

V. Conclusion

We propose a robust and distributed parameter-adaptation algorithm to track different QoS requirements in IEEE 802.11e multihop networks and prove its performance by analysis and simulations. Delay-sensitive traffic has an average delay requirement and throughput-sensitive traffic has a minimum guaranteed bandwidth requirement. According to the estimated delay and throughput, all AC queues in nodes change their minimum contention window dynamically and independently to satisfy different QoS requirements. We have shown that our proposed algorithm finds a feasible solution if such a solution exists. We proved this claim by using the Lyapunov stability technique. The simulation results show that our proposed algorithm can protect delay-sensitive traffic even when throughput-

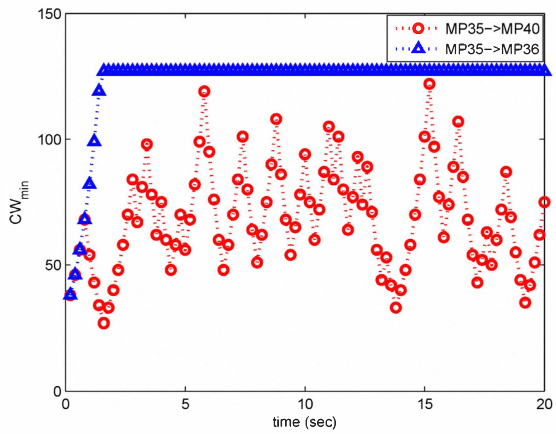


Fig. 3. CW_{min} variation of the proposed algorithm

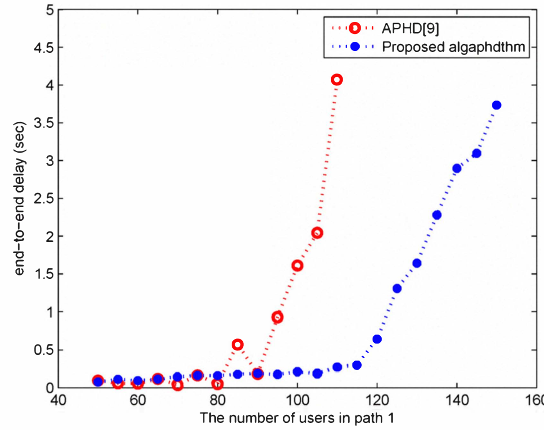
sensitive users send more than their requirements, while delay can increase severely due to high priority traffic in APHD [9].

Acknowledgement

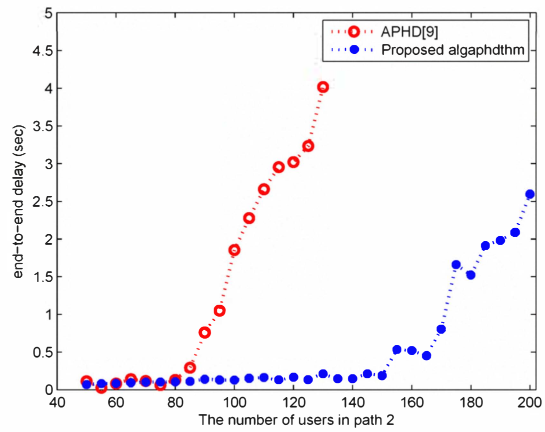
This work was supported by the IT R&D program of MKE/KEIT. [10035567, P2P mass storage multimedia contents distribution/ intermediary system development based on multiple wireless network]

References

[1] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11, 1997.
[2] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11e, 2005.
[3] H. Kim, J. Hou, C. Hu, Y. Ge, "QoS provisioning in IEEE 802.11-compliant networks: Past, present, and future," *Computer Networks*, vol. 51, 2007, pp. 1922–1941.
[4] N. Ramos, D. Panigrahi, and S. Dey, "Quality of Service Provisioning in 802.11e Networks: Challenges, Approaches, and Future Directions," *IEEE Network*, July/August 2005, pp. 14–20.
[5] H. Zhu and I. Chlamtac, "Performance Analysis for IEEE 802.11e EDCF Service Differentiation," *IEEE Trans. Wireless Communications*, vol. 4, no. 4, pp. 1779–88, 2005.
[6] Y. Xiao and H. Li, "Voice and Video Transmissions with Global Data Parameter Control for the IEEE 802.11e Enhance Distributed Channel Access", *IEEE Trans. Parallel Distrib. Sys.*, vol. 15, no. 11, pp. 1041–53, 2004.
[7] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?" *IEEE Communications Magazine*, vol. 39, issue 6, pp. 130–137, 2001.
[8] H. Lim, D. Jung, J. Hwang, K. Park and J. C. Hou, "Contention Control for High End-to-End Throughput Performance of Multihop Wireless Networks," in *Proc. of the 3-rd WICON*, 2007.
[9] J. Li, Z. li and P. Mohapatra, "Adaptive per hop differentiation for end-to-end delay assurance in multihop wireless networks," *Ad Hoc Networks*, vol. 7, 2009, pp. 1169-1182.
[10] Luca Scalia, Ilenia Tinnirello, Juki Wirawan Tantra, Chuan Heng Foh, "Dynamic MAC Parameters Configuration for Performance Optimization in 802.11e Networks," in *Proc. IEEE ICC*, May 2008.
[11] Ye Ge, Jennifer C. Hou, Sunghyun Choi, "An analytic study of tuning systems parameters in IEEE 802.11e enhanced distributed channel access," *Computer Networks*, vol. 51, issue 8, June 2007, pp1955-1980.
[12] Y. Xiao, H. Li, and S. Choi, "Protection and Guarantee for Voice and Video Traffic in IEEE 802.11e Wireless LANs," in *Proc. IEEE INFOCOM '04*, vol. 3, pp. 2152–62, Hong Kong, Mar. 2004.



(a) Path 1



(b) Path 2

Fig. 4. The end-to-end delay as a function of the number of delay-sensitive users

[13] L. Zhang and S. Zeadally, "HARMONICA: Enhanced QoS Support with Admission Control for IEEE 802.11 Contention-based Access," in *Proc. IEEE RTAS '04*, pp. 64–71, Toronto, Canada, May 2004.
[14] Y. Yang and R. Kravets, "Achieving Delay Guarantees in Ad Hoc Networks through Dynamic Contention Window Adaptation," *IEEE INFOCOM*, Apr. 2006.
[15] Y. Yang, J. Wang, R. Kravets, "Distributed optimal contention window control for elastic traffic in Single-Cell Wireless LANs," *IEEE/ACM Transactions on Networking*, pp.1373-1386, vol. 15, Dec. 2007.
[16] D. Gao, J. Cai, K. N. Ngan, "Admission Control in IEEE 802.11e Wireless LANs," *IEEE Network*, vol. 19, pp. 6–13, 2005.
[17] IEEE Draft Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE P802.11s, D1.06, July, 2007.
[18] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.
[19] http://www.isi.edu/nsnam/ns/
[20] S. S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control.* Springer-Verlag, 1999.