A Credit-Based On-Demand QoS Routing Protocol over Bluetooth WPANs*

Yuh-Shyan Chen, Keng-Shau Liu, and Yun-Wei Lin
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi, Taiwan, 621, R.O.C.
yschen@cs.ccu.edu.tw

Abstract

The quality-of-service (QoS) communication that supports mobile applications to guarantee bandwidth utilization is an important issue for Bluetooth wireless personal area networks (WPANs). In this paper, we address the problem of on-demand QoS routing with interpiconet scheduling in Bluetooth WPANs. A credit-based QoS (CQ) routing protocol is developed which considers different Bluetooth packet types because of different types of Bluetooth packets have different bandwidth utilization levels. This work improves the bandwidth utilization of Bluetooth scatternets by providing a new interpiconet scheduling scheme. Centralized and distributed algorithms are investigated in this work to improve the bandwidth utilization for the on-demand QoS routing protocol. The performance analysis illustrates that our credit-based QoS routing protocol achieves enhanced performances, compared to existing QoS routing protocols.

1. Introduction

Bluetooth is a low-power short-range wireless network technology. From the IEEE 802.15.1 standard [5], Bluetooth devices establish wireless personal area networks (WPANs) that can be used to provide useful services such as wireless Internet access or mobile multimedia applications in mobile ad hoc network (MANET) systems [2][3]. Many researchers [1][4][6] have recently attempted to develop QoS-extension routing scheduling in Bluetooth scatternets. First, Cordeiro *et al.* [4] proposed dynamic slot assignment (DSA) and enhanced DSA (EDSA) schemes. A direct slave-to-slave communication model is presented in [4] to provide QoS requirements. Unfortunately, no interpiconet scheduling mechanism has been devised when the source and destination nodes are located in distinct piconets. A QoS scheduling mechanism for scatternets was recently investigated by Kim *et al.* [6]. The algorithms proposed by Kim *et al.* are suitable for tree-structure scatternets [12]. The success rate of QoS-aware scheduling algorithms drops off for non-tree-structure scatternets. It also degrades the bandwidth utilization of each Bluetooth device. In this paper, we address these on-demand quality-of-service routing and interpiconet scheduling problems. A credit-based QoS (CQ) routing protocol is developed which considers different Bluetooth packet types which have different bandwidth utilization levels [8]. This work can improve the bandwidth utilization of Bluetooth scatternets. Interpiconet scheduling problems can be resolved by our CQ approach. Both centralized and distributed algorithms are provided to improve bandwidth utilization. The simulation results illustrate that our CQ routing algorithm performs better than Kim *et al.*'s approach [6].

The rest of this paper is organized as follows. Section 2 introduces some preliminary work. In Section 3, we develop the centralized QoS routing protocol and the distributed QoS routing protocol. Section 4 discusses the experimental results. Finally, Section 5 concludes this paper.

Type	Used payload (bytes)	FEC	Utilization (bytes/slot)
DM1	17	YES	17.0
DM3	121	YES	40.3
DM5	224	YES	44.8
DH1	27	NO	27.0
DH3	183	NO	61.0
DH5	339	NO	67.8

Table 1. Bluetooth ACL data packets

2. Preliminary

2.1. ACL/SCO Link Properties of Bluetooth

According to Bluetooth specifications [5], two types of the traffic links are used: one is the asynchronous connectionless (ACL) link and the other one is the synchronous connection-oriented (SCO) link. This paper only discusses QoS problems for data transmission in ACL links. In the following, we only investigate QoS issues in ACL links. The packet of an ACL link may have one, three, or five time slots

* This work was supported by the National Science Council of the Republic of China under grant nos. NSC-92-2213-E-194-022 and NSC-93-2213-E-194-028.

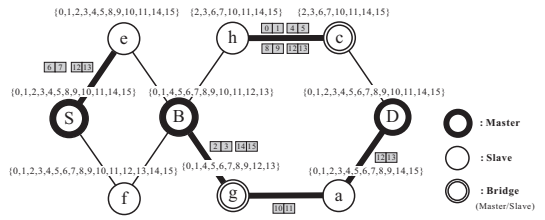


Figure 1. Bandwidth usage of a Bluetooth scatternet.

as shown in Table 1 [5][8]. This fact motivated us to develop an efficient QoS routing protocol in this work by taking different packet types with various bandwidth utilization levels into account.

A high-performance QoS routing protocol is designed if the protocol has better time slot utilization. To simplify our presentation, we only discuss the time-slot reservation scheme for even-numbered time slots to indicate the fact that each traffic pair adds the POLL or ACK. packets.

2.2. Basic Idea of the CQ Protocol

Lower bandwidth utilization of each Bluetooth device is the drawback of Kim *et al.*'s approach [6]. Only using DM1 packets leads to the problem of lower bandwidth utilization. Furthermore, Kim *et al.*'s approach [6] is only suitable for tree-structure scatternets [12]. The lower success rate of searching for a QoS route is obtained for non-tree-structure scatternets. To improve the bandwidth utilization and success rate, a new QoS scheduling scheme was investigated which not only adopts DM1 packets but also uses DM3 and DM5 packets for the QoS scheduling.

In the following, we use an example to point out the drawback of Kim *et al.*'s scheme. This example attempts to find a QoS route (*S*, *e*, *B*, *g*, *a*, *D*) with the QoS requirement of transmitting 51 bytes per polling interval. Fig. 1 shows an example of bandwidth usage of a Bluetooth scatternet before finding a QoS route. The divide-and-conquer approach used in Kim *et al.*'s scheme [6] repeatedly splits the *traffic-load* matrix defined in [6] into *traffic-load* sub-matrices until the *traffic-load* sub-matrix contains only DM1 packets to prevent contention for the time slot reservation. That is to say, only DM1 packets are used to construct a QoS route in Kim *et al.*'s scheme. Let $L(X,Y)$ or $\overleftrightarrow{XY}$ denote the traffic link between Bluetooth devices X and Y. But it failed to find three DM1 packets in link $\overleftrightarrow{ga}$, and therefore it failed to search for a QoS route.

To transmit 51 bytes per polling interval, we can use three DM1 packets or one DM3 packet from Table 1. Our main idea is to possibly use DM5, DM3, and DM1 packets to achieve the goal of using fewer free time slots. This work mainly attempts to improve the bandwidth utilization and promote the success rate of searching for a QoS route. Given the same scenario as in Fig.1, the QoS scheduling result using Kim *et al.*'s scheme fails. Fig. 2 gives a successful QoS scheduling result using our new scheduling scheme. Therefore, a QoS route (*S*, *e*, *B*, *g*, *a*, *D*) with the QoS requirement of transmitting 51 bytes is successfully constructed. Packets with different types of support for

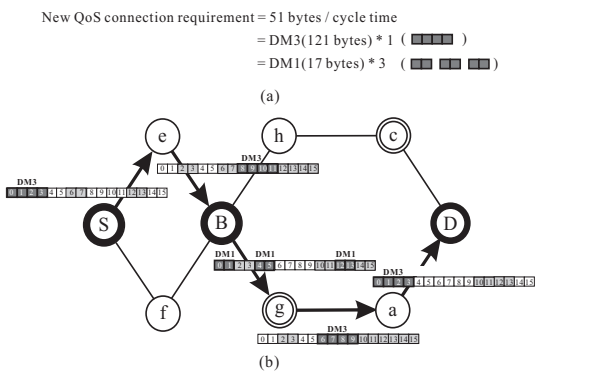


Figure 2. QoS scheduling result using our scheme.

QoS requests produce different bandwidth utilization levels. This obviously causes the problem of low bandwidth utilization and degradation of the success rate. Developing a new scheduling scheme by adopting DM1, DM3, and DM5 packets is the key idea of our work.

3. A Centralized On-Demand QoS Routing Protocol

3.1. Phase I: Free Time-Slot Information Collection

A Bluetooth scatternet is assumed to initially be formed by existing formation protocols [7][9] [12]. The detailed collection of free time-slot information from source to destination nodes is performed. The source node initiates the QoS_REQuest, or BQ_REQ, packet and floods into Bluetooth scatternets until the BQ_REQ packets arrive at the destination node. Each BQ_REQ packet records all free time-slot information of links along a path from the source node to the destination node.

In the following, we describe how to calculate free time slots between two adjacent nodes in Bluetooth scatternets. Let $\{\alpha_1, \alpha_2, ..., \alpha_k\}$ denote a free time-slot set for a Bluetooth device in a Bluetooth scatternet. For instance as shown in Fig. 1, $\{0, 1, 2, 3, 4, 5, 8, 9, 10, 11, 14, 15\}$ is the free time-slot set of node S. Given a pair of adjacent nodes, A and B, free time-slot sets of A and B are $\{\alpha_1,\alpha_2,...,\alpha_{k_1}\}$ and $\{\beta_1,\beta_2,...,\beta_{k_2}\}$, where $k_1 \neq k_2$. As mentioned in Section II, $\overleftrightarrow{AB}$ denotes the link between adjacent nodes A and B. An intersection function, $\cap(\{\alpha_1,\alpha_2,...,\alpha_{k_1}\}, \{\beta_1,\beta_2,...,\beta_{k_2}\}) = \{\gamma_1, \gamma_2, ..., \gamma_{k_3}\}$, is executed for link $\overleftrightarrow{AB}$ to calculate the shared free time slots of nodes A and B, where $\{\gamma_1, \gamma_2, ..., \gamma_{k_3}\} \in \{\alpha_1, \alpha_2, ..., \alpha_{k_1}\}$, $\{\beta_1, \beta_2, ..., \beta_{k_2}\}$, and $k_3 \leq min(k_1, k_2)$. For example, the free time-slot list of link $\overleftrightarrow{Bg}$ is $\{0, 1, 4, 5, 6, 7, 8, 9, 12, 13\} = \cap(\{0, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}, \{0, 1, 4, 5, 6, 7, 8, 9, 12, 13\})$. The BQ_REQ packet is defined as BQ_REQ(S_ADDR, D_ADDR, FT, PL, FTSL, BR,

Packet field	Field description
S_ADDR	Source node address
D_ADDR	Destination node address
FT	Free time slots of the current node
PL	List of node information that records the path from the source tothe current traversed node
FTSL	Free time-slot list of links, each of which records the shared free time slots among the current traversed node and the last node recorded in the PL
BR	QoS requirement of the source host
TTL	Time To Live: limitation of the hop-length in the search path

Table 2. Detailed definition of a BQ_REQ packet

TTL), where the detailed definition is given in Table 2. The algorithm of free time-slot information collection is given below.

(A1) Source node S initiates and floods a BQ_REQ(S_ADDR = S, D_ADDR = D, FT =$\{\alpha_1,\alpha_2,...,\alpha_{k_1}\}$, PL=$\{\}$, FTSL=$\{\}$, BR, TTL) packet into a Bluetooth scatternet, where D is a destination node, BR is the QoS requirement, and TTL is the time-to-live value.

(A2) If node e receives a BQ_REQ(S_ADDR = S, D_ADDR = D, current_FT, current_PL, current_FTSL, BR, current_TTL) packet from node e' in the Bluetooth scatternet, the current_TTL and D_ADDR fields are checked, and four cases are considered.

(B1) If current_TTL is equal to zero and node e is not equal to D_ADDR of the BQ_REQ packet, then the current BQ_REQ packet is dropped.

(B2) If the shared free time slots $\{\gamma_1, \gamma_2, ..., \gamma_{k_3}\}$ of link $\overleftrightarrow{e'e}$ cannot satisfy the QoS requirement, BR, then the current BQ_REQ packet is dropped.

(B3) If node e is equal to D_ADDR, then go to step A3.

(B4) Node e appends e into the current_PL field and adds $\{\gamma_1, \gamma_2, ..., \gamma_{k3}\}$ into the current_FTSL field, and decreases the value of the current_TTL. Node e floods BQ_REQ(S, D, itself_FT, {current_PL, e}, current_FTSL $\cup$ $\{\gamma_1, \gamma_2, ..., \gamma_{k3}\}$, BR, current_TTL - 1) into the Bluetooth scatternet, where itself_FT is the free time-slot list of node e.

(A3) Destination node D waits for a period of time to receive many different BQ_REQ packets from the source node.

3.2. Phase II: Time-Slot Reservation

3.2.1. Credit-based Algorithm To assign time slots to each link while considering different packet types, a better QoS route with high bandwidth utilization is presented. First, each link is assigned a different priority value. This value indicates the degree of influence that link has with its neighboring links. A credit-based algorithm is developed based on the priority value of finding a QoS route with lower influence by neighboring links. The detailed description follows.

After collecting many BQ_REQ packets from a source node, all free time-slot information is obtained at the destination node. The destination node chooses one of them and performs the following operation. Without loss of generality, a path $(s_0, M_1, s_1, M_2, \cdots, M_i, s_i)$ is chosen where the source and destination nodes are s_0 and s_i. A *shared free-time slot* matrix, M_f, is used to indicate information of the shared free time slots of links $\overleftrightarrow{s_0M_1}$, $\overleftrightarrow{M_1s_1}$, $\overleftrightarrow{s_1M_2}, \cdots$, and $\overleftrightarrow{M_is_i}$ of route $(s_0, M_1, s_1, M_2, \cdots, M_i, s_i)$. Each row of matrix M_f denotes the shared free time slots of links $\overleftrightarrow{s_0M_1}$, $\overleftrightarrow{M_1s_1}$, $\overleftrightarrow{s_1M_2}, \cdots$, and $\overleftrightarrow{M_is_i}$. For instance, consider route (S, e, B, g, a, D), matrix M_f is constructed for links $L(S, e)$, $L(e, B)$, $L(B, g)$, $L(g, a)$, and $L(a, D)$ as illustrated in Fig. 3(a). The first row of matrix M_f is $\{0, 1, 2, 3, 4, 5, 8, 9, 10, 11, 14, 15\}$ for $L(S, e)$. Let $F(X, Y)$ and $B(X, Y)$ denote the free time slots and busy time slots of link $L(X, Y)$ or $\overleftrightarrow{XY}$. Given route $(s_0, M_1, s_1, M_2, \cdots, W, X, Y, Z \cdots, M_i, s_i)$, we consider three adjacent links, $\overleftrightarrow{WX}$, $\overleftrightarrow{XY}$, and $\overleftrightarrow{YZ}$, along the route. Number list $P(\delta)_{L(X,Y),L(Y,Z)}$ is constructed by $P(\delta_i)_{L(X,Y),L(Y,Z)}$, where $0 \le i \le$ polling_interval. Each $P(\delta_i)_{L(X,Y),L(Y,Z)}$ denotes a credit value of i-th time slots for links $L(X, Y)$ and $L(Y, Z)$ as follows.

$$P(\delta_i)_{L(X,Y),L(Y,Z)} = \begin{cases} 0, \text{ if } \delta_i \in B(X,Y), \\ 1, \text{ if } \delta_i \in F(X,Y) \cap F(Y,Z), \\ 2, \text{ otherwise.} \end{cases}$$
$$\text{where } 0 \le i < \text{polling\_interval.}$$

Similarly, number list $P(\delta)_{L(X,Y),L(W,X)}$ is used to denote the credit values of links $L(X, Y)$ and $L(W, X)$, where every $P(\delta_i)_{L(X,Y),L(W,X)}$ is defined below.

$$P(\delta_i)_{L(X,Y),L(W,X)} = \begin{cases} 0, \text{ if } \delta_i \in B(X,Y), \\ 1, \text{ if } \delta_i \in F(X,Y) \cap F(W,X), \\ 2, \text{ otherwise.} \end{cases}$$
$$\text{where } 0 \le i < \text{polling\_interval.}$$

Let p denote the permutation of any packet types to satisfy the QoS requirement. For example, the QoS requirement is 224 bytes per cycle time, and four different packet types are produced, i.e., $p = 4$; (1) a DM5 packet, (2) two DM3 packets, (3) one DM3 packet and eight DM1 packets, and (4) 16 DM1 packets. For the other example shown in Fig. 3(a), $p = 2$ for the QoS requirement of 51 bytes for $L(e, B)$ and one DM3 packet and three DM1 packets are produced.

With $P(\delta)_{L(e,B),L(S,e)} + P(\delta)_{L(e,B),L(B,g)}$, there are m conditions of time reservation if we only consider one kind of packet type to satisfy the QoS requirement. For the same example of $L(e,B)$ shown in Fig. 3(a), if we consider one DM3 packet, it can be reserved for time slot (8, 9, 10, 11). Therefore, $m = 1$. If we consider three DM1 packets, they can be reserved on (0, 1), (4, 5), and (8, 9) or (0, 1), (4, 5) and (10, 11). Therefore, $m = 2$. Therefore, a computation time of $O(m \cdot p)$ is needed for the one-hop time reservation.

Given path $(s_0, M_1, s_1, M_2, \cdots, W, X, Y, Z \cdots, M_i, s_i)$, if we have matrix M_f and information on the summed number lists $P(\delta)_{L(e,B),L(S,e)} + P(\delta)_{L(e,B),L(B,g)}$, then time-slot reservation is given as follows.

(C1) Link $L(X,Y)$ is selected from $(s_0, M_1, s_1, M_2, \cdots, W, X, Y, Z \cdots, M_i, s_i)$ with a lower number of shared free time slots, such that route $(s_0, M_1, s_1, M_2, \cdots, W, X, Y, Z \cdots, M_i, s_i)$ can be divided into two sub-paths, $(s_0, M_1, s_1, M_2, \cdots, W, X)$ and $(Y, Z \cdots, M_i, s_i)$, with two sub-matrices, M'_f and M''_f, where $M_f = M'_f + M''_f$. If there is more than one link with the same fewer number of free time slots, then we randomly select one link from them.

(C2) With the QoS requirement, we first try possible DM5 packets to satisfy the QoS requirement of link $L(X,Y)$. If these do not satisfy the QoS requirement, we continue to try possible DM3 packets to satisfy the QoS requirement. Then, if the QoS requirement is still not satisfied, we continue to try possible DM1 packets to satisfy it. All of the above operations depend on the priority of the summed number lists of $P(\delta)_{L(X,Y),L(Y,Z)} + P(\delta)_{L(X,Y),L(W,X)}$.

  (D1) The time-slot reservation operations of steps C1 and C2 on sub-path $(s_0, M_1, s_1, M_2, \cdots, W, X)$ are recursively performed with sub-matrix M'_f.

  (D2) The time-slot reservation operations of steps C1 and C2 on sub-path $(Y, Z \cdots, M_i, s_i)$ are recursively performed with sub-matrix M''_f.

For instance as shown in Fig. 3(a), a route (S, e, B, g, a, D) with M_f is split into two sub-paths, (S, e) and (B, g, a, D), with M'_f and M''_f, since $\overleftrightarrow{eB}$ has eight free time slots. The QoS requirement is 51 bytes per cycle time. As illustrated in Fig. 3(a), DM3 is reserved to $L(e,B)$, and DM3 is recursively reserved to $L(S,e)$. Sub-matrix M''_f for (B, g, a, D) is split into (B, g) and (a, D) after DM3 is allocated to $L(g,a)$. Finally, 3 DM1 packets are reserved in $L(B,g)$ and one DM3 is allocated to $L(a,D)$.

After performing the time-slot reservation operation, the destination node replies with a REPly (RREP) packet from the destination node to the source node to reserve time slots with the QoS requirement and which releases all unrelated time slots in all other routes. The time complexity of the credit-based algorithm is given.

3.2.2. Optimal Algorithm To provide an optimal solution for constructing a QoS route, an optimal algorithm is presented as follows. Given route $(s_0, M_1, s_1, M_2, \cdots, W, X, Y, Z \cdots, M_i, s_i)$ and matrix M_f, the optimal time slot reservation is given.

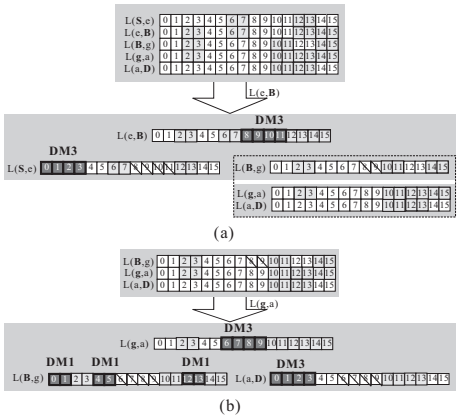


Figure 3. Time slots reserved step-by-step by the CCQ approach.

(E1) Every link $L(X,Y)$ is selected from $(s_0, M_1, s_1, M_2, \cdots, W, X, Y, Z \cdots, M_i, s_i)$ by splitting $(s_0, M_1, s_1, M_2, \cdots, W, X, Y, Z \cdots, M_i, s_i)$ with matrix M_f into the two sub-paths, $(s_0, M_1, s_1, M_2, \cdots, W, X)$ and $(Y, Z \cdots, M_i, s_i)$, with the two sub-matrices, M'_f and M''_f, where $M_f = M'_f + M''_f$.

(E2) A credit-based scheme is applied to reserve time slots on link $L(X,Y)$. This work takes a computation time of $O(m \cdot p)$.

(E3) The time-slot reservation operations of steps E1 and E2 are recursively carried out on sub-path $(s_0, M_1, s_1, M_2, \cdots, W, X)$with sub-matrix M'_f until all links in sub-path $(s_0, M_1, s_1, M_2, \cdots, W, X)$ have been selected for the time-slot reservation.

(E4) The time slot reservation operations of steps E1 and E2 are recursively carried out on sub-path $(Y, Z \cdots, M_i, s_i)$ with sub-matrix M''_f until all links in sub-path $(Y, Z \cdots, M_i, s_i)$ have been selected for the time-slot reservation.

The centralized QoS on-demand routing protocol suffers from a scalability problem. To reduce the scalability problem, a distributed credit-based QoS (DCQ) protocol can be easily applied if every three-hop neighboring nodes simply execute the CCQ protocol. We omit the details herein.

4. Performance Analysis

Our study mainly presents a new credit-based time-slot reservation protocol. To evaluate our credit-based protocol and Kim *et al.*'s protocol [6], we have implemented them using the Network Simulator (ns-2) [10] and BlueHoc [11]. The performance metrics of the simulation are given below.

- *Success rate*: the number of successful QoS route requests divided by the total number of QoS route requests.

Parameters	Value
Bluetooth devices number	18
Network region	17 $\times$17 m^2
Radio propagation range	10 m
Mobility	No
Cycle time	16 time slots
Packet type	DM1 or DM3 or DM5
QoS requirements	$DM1$: $DM3$: $DM5 = 1$: 1: 1 $DM1$: $DM3$: $DM5 = 3$: 1: 1 $DM1$: $DM3$: $DM5 = 1$: 3: 1 $DM1$: $DM3$: $DM5 = 1$: 1: 3

Table 3. Detailed simulation parameters

- *Bandwidth efficiency*: the average number of data bytes which can be transmitted per time slot for a successful QoS route.
- *Slot occupation*: the average number of time slots occupied by a successful QoS route.
- *Throughput*: the number of data bytes received by all Bluetooth devices per unit time.

4.1. Performance of Success Rate

Fig. 4 shows the success rate vs. number of QoS requests for four QoS requirement scenarios: $DM1$: $DM3$: $DM5 =$ 1: 1: 1, 3: 1: 1, 1: 3: 1, and 1: 1: 3 as respectively illustrated in Fig. 4(a), (b), (c), and (d). In general, COQ, CCQ, and DCQ had higher success rates than did KIM. This is because KIM wastes too many POLL time slots using DM1 packets. The low success rate for KIM seriously occurred in the case of $DM1$: $DM3$: $DM5 = 1$: 1: 3 as shown in Fig. 4(d).

4.2. Performance of Throughput

Fig. 6 shows throughput vs. the number of QoS requests under four QoS requirement scenarios. The higher the success rate is, the higher the throughput will be. In addition, Fig. 6(a)(c)(d) show the lower performance of KIM when the number of QoS requests increases. Therefore, the KIM has the lowest throughput. This also explains why $DM1$: $DM3$: $DM5 = 3$: 1: 1 has smooth curves for the success rate and throughput in Fig. 4(b) and Fig. 6(b), respectively. Fig. 6 shows that the average throughput of KIM is about 50% of those values for COQ, CCQ, and DCQ.

4.3. Performance of Bandwidth Efficiency

Fig. 7 shows the bandwidth efficiency vs. number of QoS requests under four QoS requirement scenarios. Basically, Fig. 7 illustrates that the bandwidth efficiency of COQ $>$ that of CCQ $>$ that of DCQ $>$ that of KIM for the four QoS requirement scenarios.

5. Conclusions

In this paper, we address on-demand QoS routing and interpiconet scheduling problems. A centralized credit-based QoS routing protocol was developed which considers different Bluetooth packet types. Both centralized and distributed algorithms were investigated to improve the bandwidth utilization. The simulation result illustrates that our algorithm had better performance compared to Kim *et al.*'s approach. The QoS scheduling is expectably designed in the L2CAP to possibly implement our scheduling scheme in actual Bluetooth devices.

References

[1] S. Baatz, M. Frank, C. Kuhl, P. Martini, and C. Scholz. "Bluetooth Scatternets: An Enhanced Adaptive Scheduling Scheme". In *proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM)*, volume 2, pages 782–790, New York, June 2002.

[2] Y.-S. Chen, Y.-C. Tseng, J.-P. Sheu, and P.-H. Kuo. "An On-Demand, Link-State, Multi-Path QoS Routing in a Wireless Mobile Ad-Hoc Network". *Computer Communications*, 27:27–40, January 2004.

[3] Y.-S. Chen and Y.-T. Yu. "Spiral-Multi-Path QoS Routing Protocol in a Wireless Mobile Ad-Hoc Network". *IEICE Transactions on Communications*, E87-B:104–116, January 2004.

[4] C. Cordeiro, S. Abhyanker, and D. P. Agrawal. "Design and Implementation of QoS-driven Dynamic Slot Assignment and Piconet Partitioning Algorithms over Bluetooth WPANs". In *proceedings of the Twenty-third annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM)*, Hong Kong, March 2004.

[5] Bluetooth Special Interest Group. "The Bluetooth Specification Version 1.2". Technical report, http://www.bluetooth.com, November 2003.

[6] Y. M. Kim, T.-H. Lai, and A. Arora. "A QoS-Aware Scheduling Algorithm for Bluetooth Scatternets". In *proceedings of the International Conference on Parallel Processing*, pages 455–462, Kaohsiung, Taiwan, October 2003.

[7] T.-Y. Lin, Y.-C. Tseng, K.-M. Chang, and C.-L. Tu. "A New BlueRing Scatternet Topology for Bluetooth with Its Formation, Routing, and Maintenance Protocols". *Wireless Communications and Mobile Computing*, 3(4):517–537, 2003.

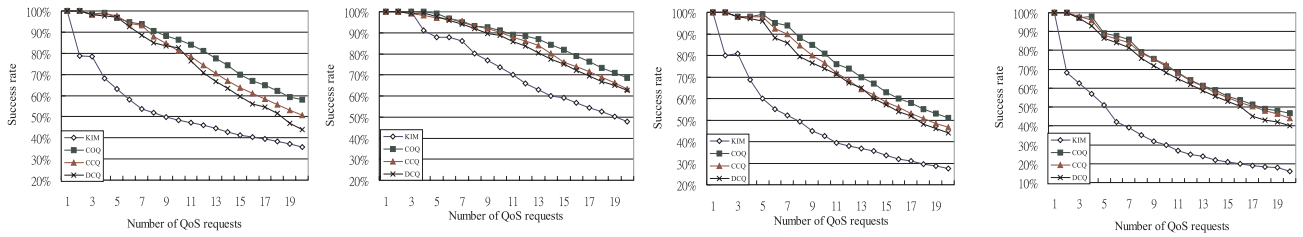
[8] T.-Y. Lin, Y.-C. Tseng, and Y.-T. Lu. "An Efficient Link Polling Policy by Pattern Matching for Bluetooth Piconets". *Computer Journal*, 47(2):169–178, March 2004.

[9] C. Petrioli, S. Basagni, and M. Chlamtac. "Configuring BlueStars: Multihop Scatternet Formation for Bluetooth Networks". *IEEE Transactions on Computers*, 52(6):779–790, June 2003.

[10] VINT Project. "Network Simulator version 2 (ns2)". Technical report, http://www.isi.edu/nsnam/ns, June 2001.

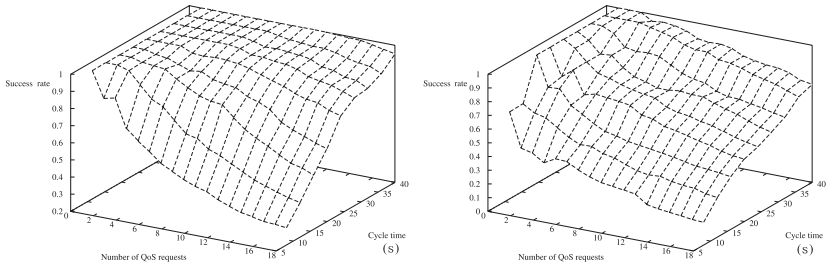
[11] IBM research. "BlueHoc, IBM Bluetooth Simulator". Technical report, http://www-124.ibm.com/developerworks/opensource/bluehoc/, February 2001.

[12] M. T. Sun, C.-K. Chang, and T.-H. Lai. "A Self-Routing Topology for Bluetooth Scatternets". In *proceedings of International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, pages 13–18, Makati City, Metro Manila, Philippines, May 2002.



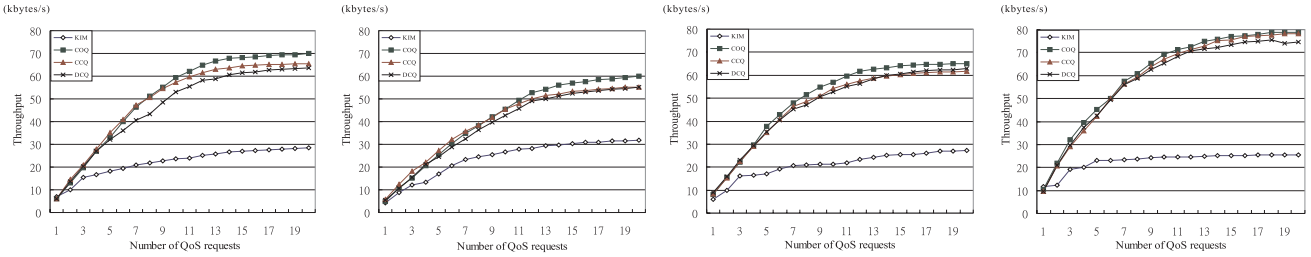
(a) DM1: DM3: DM5 = 1: 1: 1 (b) DM1: DM3: DM5 = 3: 1: 1 (c) DM1: DM3: Dm5 = 1: 3: 1 (d) DM1: Dm3: DM5 = 1: 1: 3

Figure 4. Success rate vs. the number of QoS requests.



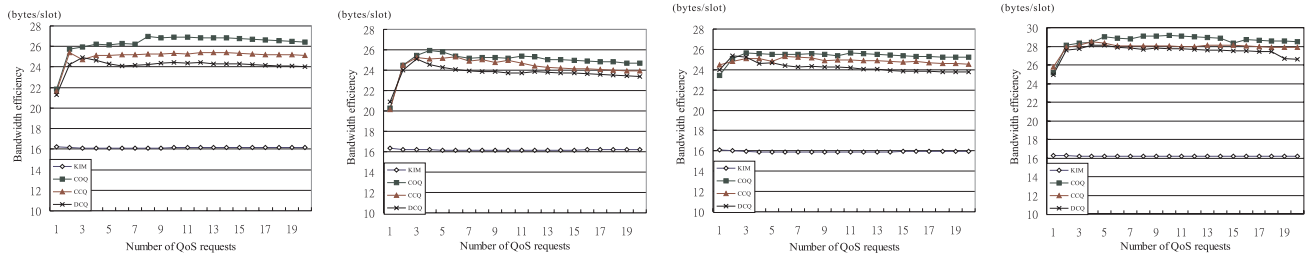
(a) (b)

Figure 5. Success rates of (a) CCQ and (b) the KIM vs. different cycle times.



(a) DM1: DM3: DM5 = 1: 1: 1 (b) DM1: DM3: DM5 = 3: 1: 1 (c) DM1: DM3: DM5 = 1: 3: 1 (d) DM1: DM3: DM5 = 1: 1: 3

Figure 6. The throughput vs. the number of QoS requests



(a) DM1: DM3: DM5 = 1: 1: 1 (b) DM1: DM3: DM5 = 3: 1: 1 (c) DM1: DM3: DM5 = 1: 3: 1 (d) DM1: DM3: DM5 = 1: 1: 3

Figure 7. The bandwidth efficiency vs. the number of QoS requests