A CENTRALIZED TIME-SLOT LEASING-BASED QOS ROUTING PROTOCOL OVER BLUETOOTH WPANS

Yuh-Shyan Chen and Tsung-Hung Lin
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan
email: {yschen;dust}@cs.ccu.edu.tw

ABSTRACT
In this paper, we investigate a new efficient quality of service (QoS) routing protocol based on the time-slot leasing mechanism over Bluetooth wireless personal area networks (WPANs). In a Bluetooth scatternet, a QoS route path is constructed through a series of QoS slave-master-slave communications. The "transmission holding" problem is incurred because the master node is the communication bottleneck for each slave-master-slave communication. To alleviate this problem, a time-slot leasing scheme is adaptively incorporated into our scheme to provide a completely new QoS routing protocol. This QoS routing protocol can additionally offer extra slave-to-slave QoS communication capability to effectively reduce the workload of the master node and significantly promote the success rate of finding a QoS route. In our proposed QoS routing protocol, QoS slave-master-slave and slave-to-slave communication mechanisms are simultaneously considered in order to achieve a high success rate of QoS routing from the source to destination nodes. Finally, simulation results demonstrate that this time-slot leasing-based QoS routing protocol can significantly improve the success ratio, delay time, throughput, and bandwidth utilization when compared to other existing QoS routing protocols.

KEY WORDS
Bluetooth, time-slot leasing (TSL), quality of service (QoS), wireless personal area network (WPAN), wireless communication.

1 Introduction

Several untethered small hand-held electronic *personal area network* (PAN) devices including mobile phones, cell phones, *personal digital assistants* (PDAs), mp3 players, headphones, digital cameras, sensors, and their peripherals will become part of our daily lives [1]. The tremendous growth of these popular heterogenous PAN devices is increasingly deriving requirements for efficient communications. Bluetooth *wireless personal area network* (WPAN) technologies are continual increasing in interest for their ubiquitous mobile connections and their ability to provide new personal communication opportunities and services [3]. In this paper, we propose a novel integrated technology to improve the performance of Bluetooth scatternets. In Bluetooth scatternets, if the source and destination nodes are located in distinct piconets, the weakness of the interpiconet scheduling mechanism is still a problem. Simplicity and low power consumption are the advantages under the slave-master-slave QoS requirement model of Bluetooth scatternets. Recently, Zhang *et al.* [7][6] proposed a new scatternet QoS approach, called *time-slot leasing* (TSL), and an enhanced version, called *enhanced TSL* (ETSL), to address the problems associated with the slave-master-slave model. The TSL approach establishes temporarily leased slots from the constructed piconet to support slave-to-slave communication. We investigate a new, efficient QoS routing protocol based on the time-slot leasing mechanism over Bluetooth wireless personal area networks (WPANs) in this paper. Finally, simulation results demonstrate that the time-slot leasing-based QoS routing protocol can significantly improve the success ratio, delay time, throughput, and bandwidth utilization when compared to other existing QoS routing protocols. The rest of this paper is organized as follows. Section 2 introduces some preliminaries and basic ideas. Section 3 develops the *centralized time-slot leasing-based QoS* (CTQ) routing protocol. In Section 4, we discuss the experimental results and performances of our *time-slot leasing-based QoS* (TQ) routing protocol. Finally, Section 5 concludes this paper.

2 Preliminaries and Basic Ideas

We discuss the basic ideas concerning our *time-slot leasing-based QoS* (TQ) routing protocol and some Bluetooth background in this section. A recent result presented by Chen *et al.* [2] is described which motivated the investigation of our TQ routing protocol. The devices that use Bluetooth technologies are normally formed piconets which communicate with each other in a slave-master-slave configuration. The bandwidth utilization is calculated by the amount of the used data payload per number of time slots; the bandwidth utilizations of *DM*1, *DM*3, and *DM*5 are 17.0 (17/1), 40.3 (121/3), and 44.8 (224/5) bytes/slot, respectively. In the TQ routing protocol, only an error-prone environment is adopted, and three packets, *DM*1, *DM*3, and *DM*5 packets, are considered. Any slave-to-slave communications must go through the master node in the same piconet, and the master node has to use extra bandwidth to exchange packets between two slave nodes. For example, in Fig. 1(a), if slave node, S_2, wants to transmit packets to another slave node, S_1, then two transmissions are needed. One is from S_2 to the master, denoted S_2*-to-master*, and the other is from the master node to slave node, S_1, denoted *master-to-*S_1. Use of S_2-to-master and master-to-S_1 doubles the bandwidth consumption and communication delay time. In Fig. 1(a), slave node, S_2, and slave node, S_1, switch their roles and become a temporary master node, denoted *Temp-master*, and a temporary slave node, denoted *Temp-slave*, respectively, and then a new temporary piconet with lower master throughput and a shorter communication delay is constructed. The TSL approach need not permanently change the basic pi-

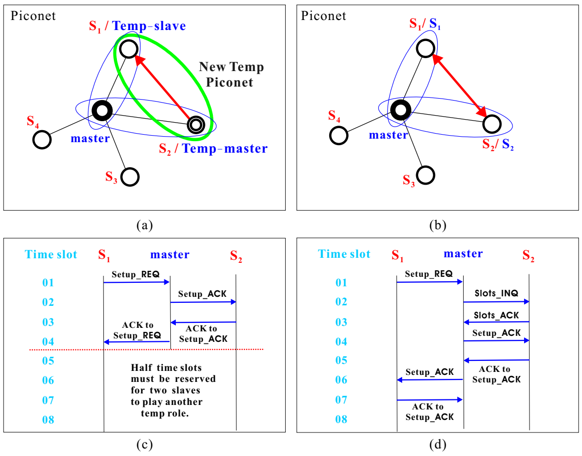

Figure 1. (a) RS architecture (with a new piconet), (b) TSL architecture (without a new piconet), (c) setup procedure of RS, and (d) setup procedure of TSL.

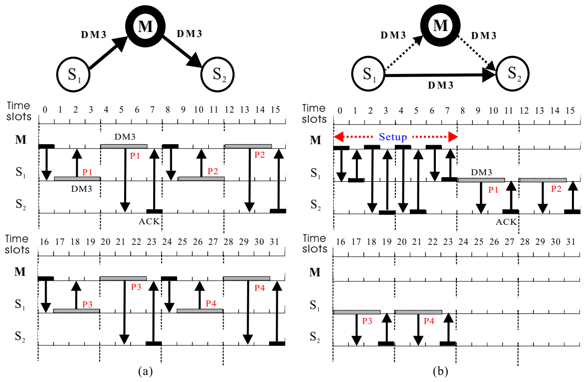

Figure 2. Time slot usages of (a) a normal piconet transmission and (b) a TSL-complemented piconet transmission.

conet structure and has no negative effects on interpiconet communications [7]. For example as shown in Fig. 1(b), if slave node, S_1, needs to send a large file to another slave node, S_2, then S_1 requests the master to lease some time slots for direct transmission between S_1 and S_2. Direct packet transmissions between S_2 and S_1 are accomplished without changing the roles of S_1 and S_2. Figure 1(c) shows the RS setup procedure, and Fig. 1(d) shows the TSL setup procedure. Figure 2 shows a comparison of time slot usages between normal piconet transmissions and TSL-complemented piconet transmissions. The new QoS connection requirement is 51 bytes/cycle time, and the *DM*3 packet type is adopted to send the data packets. The total time slot usages of a new QoS connection requirement in normal piconet communication and TSL-complemented piconet communication are 32 and 24 time slots, respectively, as shown in Fig. 2. The performance of time slot usages in the TSL-complemented scheme is better than that of the non-TSL-complemented scheme. Figure 3 shows the basic idea of time-slot leasing-based QoS route-discovery operations in our TQ routing protocol. If a new QoS connection request arrives with source node, *S*, and destination node, *D*, then the QoS requirement will be *n* bytes/cycle time. Figure 3(a) shows the case when master node, *M*, has insufficient free ($< n$) time slots in the $\overrightarrow{aM}$ and $\overrightarrow{cM}$ input links, and $\overrightarrow{Mb}$ and $\overrightarrow{Md}$ output links, to service the new QoS requirement. In this period, the time-slot leasing-based routing scheme is started up in order to find sufficient ($\geq n$) time slots for the slave-to-slave link. Figure 3(b) shows that the usable slave-to-slave link, $\overrightarrow{ab}$, can replace links $\overrightarrow{aM}$ and $\overrightarrow{Mb}$ using the time-slot leasing-based routing scheme to form the final routing path, $S \rightarrow a \rightarrow b \rightarrow D$.

3 CTQ (Centralized Time-Slot Leasing-Based QoS) Routing Protocol

In section 3, the CTQ (centralized time-slot leasing-based QoS) routing protocol is proposed. Over the preformed Bluetooth scatternet, the centralized QoS routing algorithm constructs a routing path from the source node to the destination node to satisfy the QoS requirement. Our TQ routing protocol is an efficient synthesized QoS routing protocol combining a priority-based scheme, i.e., a time-slot leasing-based scheme, and taking the factor of different packet types with different levels of bandwidth utilization into account.

In a comparison with the algorithms of Chen *et al.* [2], our CTQ routing protocol was achieved by developing an *available QoS routing-path search phase*, a *time-slot leasing-based path-recovery phase*, and a *time-slot reservation phase*. First, in the available QoS routing-path search phase, information on free time slots is collected among all of the routing paths constructed from the designated source node to the designated destination node. In the time-slot leasing-based path-recovery phase, backup paths meeting the QoS requirement are found when the request for a normal QoS requirement connection fails. Finally, in the time-slot reservation phase, priority-based, time-slot leasing-based strategies and the optimal solution of QoS time-slot reservation are given.

3.1 Available QoS Routing-Path Search Phase

For a start, collection of the required QoS free time-slot information is performed between all routing paths from the designated source node to the designated destination node. First, the designated source node broadcasts a QoS request packet, named QoS_REQ, into the Bluetooth scatternet and waits for the *route reply* (RREP) to return from the designated destination node. The QoS_REQ packet is defined as QoS_REQ (DS_ADR, DD_ADR, QoS_RQ, Cur_Free_TS, Path_List, Free_TS_link, Time_to_Live, Min_FTP). Next, let $\{\alpha_1, \alpha_2, ..., \alpha_n\}$ denote a set of free time slots for our Bluetooth scatternet environment. For example as shown in Figs. 4(a) and 5(a), {2, 3, 4, 5, 6, 7, 8, 9, 12, 13} is the set of free time slots of node *a* and {10, 11, 12, 13} is the set of free time slots of node *M*. Then, $\overline{aM}$ denotes the link between adjacent nodes *a* and *M*. An intersection function is executed for link $\overline{aM}$ to calculate the shared free time slots between nodes *a* and *M*. The intersection function is shown as $\{\alpha_1, \alpha_2, ..., \alpha_{n_1}\} \cap \{\beta_1, \beta_2, ..., \beta_{n_2}\} = \{\gamma_1, \gamma_2, ..., \gamma_{n_3}\}$, where $\{\gamma_1, \gamma_2, ..., \gamma_{n_3}\} \in \{\alpha_1, \alpha_2, ..., \alpha_{n_1}\}$, $\{\beta_1, \beta_2, ..., \beta_{n_2}\}$, and $n_3 \leq \min(n_1, n_2)$.

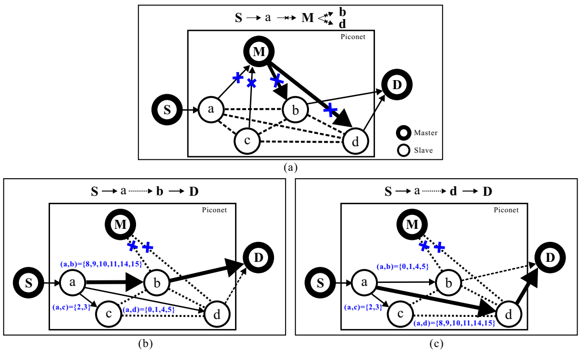

Figure 3. Basic idea of time-slot leasing-based QoS route-discovery operations.

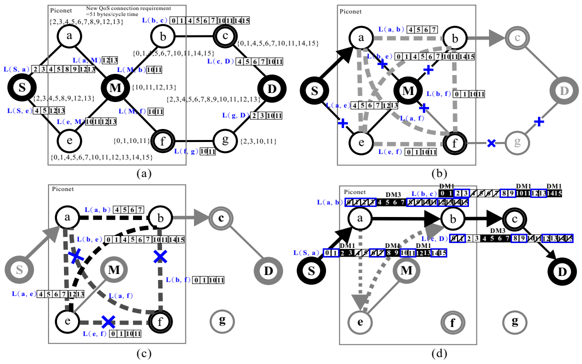

Figure 4. One-hop CTQ approach.

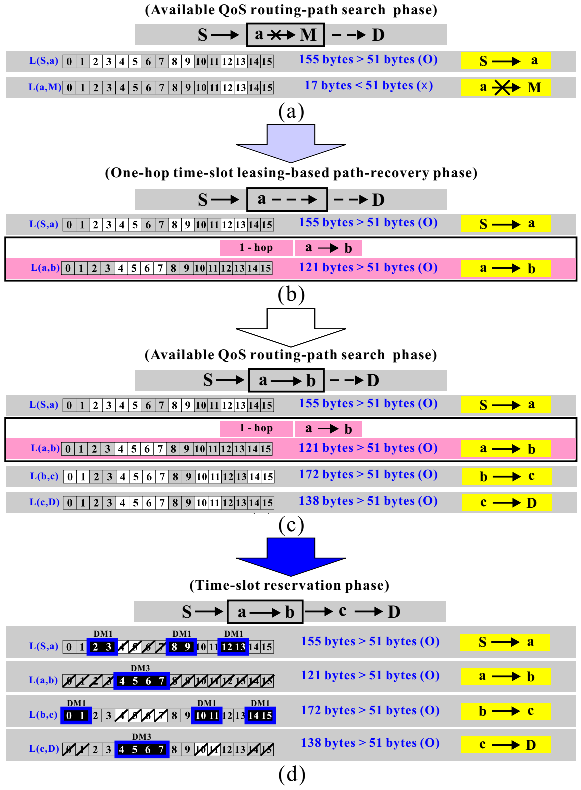

Figure 5. Time slot reservation steps using the one-hop CTQ approach.

(A1) The designated source node initially broadcasts the QoS_REQ (DS_ADR=*S*, DD_ADR=*D*, DQ_REQ=*QR*, Cur_Free_TS=$\{\alpha_1, \alpha_2, ..., \alpha_{n_1}\}$, Path_List={}, Free_TS_Link={}, Time_to_Live=*HN*, Min_FTP=0) into the Bluetooth scatternet, where *S* is the designated source node, *D* is the designated destination node, *QR* is the QoS requirement, and *HN* is the hop numbers of the time-to-live.

(A2) If an intermediate node between nodes *S* and *D* receives the QoS_REQ (*S*, *D*, *QR*, *Cur_FT*, *Cur_PL*, *Cur_FTL*, *Cur_HN*, *MinPK*) from the other nodes in the scatternet, where *Cur_FT* is the currently free time slots, *Cur_PL* is the current path list, *Cur_FTL* is the currently free time slots of link, *Cur_HN* is the remaining useable hop numbers, and *MinPK* is the packet numbers of the current minimum available free time slots. Then, the current node checks the *Cur_HN* and *D*, and follows the suitable cases below.

  (B1) If the current node is destination node, *D*, then jump to A3.

  (B2) If the current node is not destination node, *D*, and the remaining hop number *Cur_HN* is equal to zero, then drop the current QoS_REQ.

  (B3) If the current node, assumed to be N_c, is not the destination node, and the shared free time slots between N_c and the corresponding node, assumed to be $N_{c'}$, in the *Cur_FTL* cannot satisfy the QoS value *QR*, then keep the failed link, $\overline{N_c N_{c'}}$, and jump to the *time-slot leasing-based recovery phase* (C1).

  (B4) If the current node is not destination node, *D*, then add the information of the current node to the QoS_REQ and flood the new modified QoS_REQ into the Bluetooth scatternet, where the new QoS_REQ is (*S*, *D*, *QR*, {*free time slots of current node*}, {*Cur_PL*} ∪ {*current node*}, [{*Cur_FTL*} ∪ {*shared free time slots of current node*}], *Cur_HN* - 1, *new updated MinPK*).

(A3) The designated destination node waits for a reasonable time period to receive the different QoS_REQ packets from the designated source node. If the reasonable time period passes, then the list of QoS_REQ packets passes into the *time-slot reservation phase* (E1), and this phase is finished.

3.2 Time-Slot Leasing-Based Path-Recovery Phase

To increase the system performance, the time-slot leasing-based scheme is started up when the routing path of the traditional free time-slot information-collection phase fails. The time-slot leasing-based slave-to-slave architecture replaces the traditional slave-master-slave architecture to increase system throughput and decrease system delay times. The algorithm of the time-slot leasing-based path-recovery phase is described as follows.

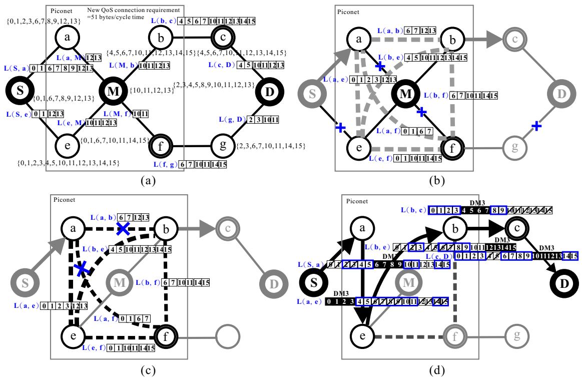

Figure 6. Two-hop CTQ approach.

(C1) If the failed link is $\overline{N_c N_{c'}}$, then all of the slave-to-slave information is collected in the following sub-steps.

(D1) If N_c is the master node, then link information is collected of all links from all other slave nodes to slave node, $N_{c'}$, in the same piconet with master node, N_c.

(D2) If $N_{c'}$ is the master node, then link information is collected of all links from slave node, N_c, to other slave nodes in the same piconet with master node, $N_{c'}$.

(C2) If all of the links have failed, then the QoS_REQ is dropped, and the algorithm is broken by returning the message of the failed QoS connection requirement.

(C3) The fewer-hops-first strategy will preferentially adopt a choice of time-slot leasing-based replacement paths. If more than one available link exists, then the group possessing the fewest number of hops is chosen.

(C4) When the group with fewer hop numbers is chosen, the maximum-free-packet-first strategy continuously chooses the time-slot leasing-based replacement path. If more than one available link exists, in the chosen group, then the link with the maximum free packets is chosen. A complete two-hop CTQ approach is shown in Fig. 6.

(C5) When the replacement link is chosen, then return to the previous *available QoS routing-path search phase*.

3.3 Time-Slot Reservation Phase

Finally, the time-slot reservation phase is proposed to reserve the time slots to support the QoS requirement. In this phase, a priority-based QoS routing scheme is also proposed to increase the high bandwidth utilization by assigning time slots with different packet types into each link. The main purpose of the priority-based algorithm is developed based on the priority value to find a lower-influence QoS route. In comparison with the definition of Chen *et al.* [2], we describe our priority-based definition clearly as follows.

Let BTL(X, Y) and FTL(X, Y) denote the time slots with a busy status of link $\overline{XY}$ and time slots with a free status of link $\overline{XY}$, respectively. The status of time slot i of L(A, B) is denoted $TS(i)_{L(A,B)}$. A priority credit value list $CN(\delta)_{L(X,Y,Z)}$ is constructed by priority credit values $CN(\delta_i)_{L(X,Y,Z)}$, where $0 \leq i \leq$ the polling interval. We show the rules of the time-slot reservation phase as follows.

(E1) If $TS(i)_{L(Y,Z)}$ is in BTL(Y, Z), then set $CN(i)_{L(X,Y,Z)}$ to 0. This means that the current link, L(Y, Z), is busy and unusable, even though the prefix link, L(X, Y), is free. We give the priority credit value 0 to the connection reference $CN(i)_{L(X,Y,Z)}$.

(E2) If $TS(i)_{L(Y,Z)}$ and $TS(i)_{L(X,Y)}$ are in the FTL(Y, Z) and FTL(X, Y), respectively, then set $CN(i)_{L(X,Y,Z)}$ to 1. This means that the current link, L(X, Y, Z), is free, and we can use this time slot for a QoS connection. We give the priority credit value 1 to the connection reference $CN(i)_{L(X,Y,Z)}$.

(E3) If $TS(i)_{L(Y,Z)}$ is in the FTL(Y, Z) and $TS(i)_{L(X,Y)}$ is in the BTL(X, Y), then set $CN(i)_{L(X,Y,Z)}$ to 2. This means that the prefix link, L(X, Y), is unusable and the current link, L(Y, Z), is still unusable even though the current link, L(Y, Z), is free. We give the priority credit value 2 to the connection reference, $CN(i)_{L(X,Y,Z)}$.

(E4) If $PL(i)_{L(W,X,Y,Z)}$ is denoted the priority list of the time-slot-chosen strategy, then, $PL(i)_{L(W,X,Y,Z)}$ = $CN(i)_{L(W,X,Y)}$ + $CN(i)_{L(X,Y,Z)}$. The higher value in the sum number list, $PL(i)_{L(W,X,Y,Z)}$, will be chosen first by the property of lower influence capability.

If the QoS connection requirement is 200 bytes/cycle time and the three inequalities (1*224=224 $>$ 200 $>$ 0*224=0), (2*121=242 $>$ 200 $>$ 1*121=121), and (12*17=204 $>$ 200 $>$ 11*17=187) are established, then we can use either one packet of the *DM*5 packet type, two packets of the *DM*3 packet type, or twelves packets of the *DM*1 packet type to satisfy the QoS requirement under the three inequalities. In particular, the *DM*1 and *DM*3 packet types of all the examples in this paper are the only chosen types that are adopted to support the QoS connection requirement. Now, a path, $(M_0, s_0, M_1, s_1, ..., W, X, Y, Z, ..., M_i)$, named *PA*, and a priority list, $PL(i)_{L(W,X,Y,Z)}$, are given. *PA* is received by the designated destination node. Then, the time slot reservation algorithm is described as follows.

(F1) Randomly select a link, L(X, Y), with minimum shared free time slots from *PA* and then divide the *PA* into two sub-paths, PA_1=(M_0,..., X) and PA_2=(Y,..., M_i), where *PA*=PA_1+PA_2.

(F2) For the QoS connection requirement, the priority list, $PL(i)_{L(W,X,Y,Z)}$, is used, and the *DM*5 packet type is first adopted to try the QoS connection requirement if the number of free time slots is sufficient. Continuing, the *DM*3 packet type is tried to see if it can satisfy the QoS requirement if the *DM*5 packet type failed. If it still does not satisfy the QoS requirement, the *DM*1 packet type is tried to see if it can satisfy the QoS requirement.

(F3) Recursively perform the time slot reservation operations of steps F1 and F2 until all of the sub-paths are processed and all of the paths that are received by the designated destination node are processed.

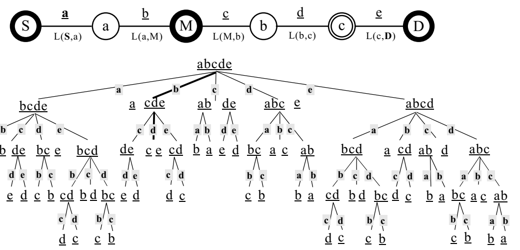

Figure 7. Conditions of time-slot reservation with a normal optimal algorithm.

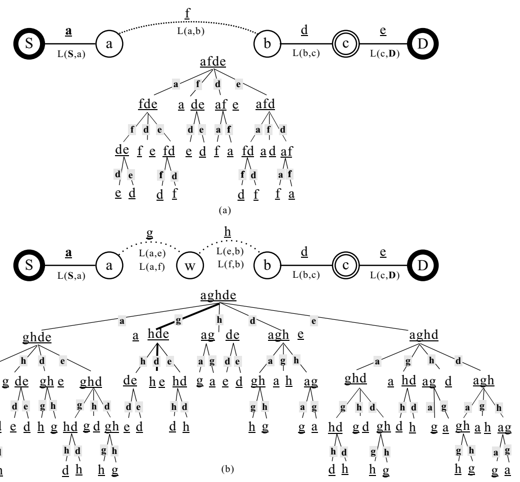

Figure 8. Conditions of time-slot reservation with the (a) one-hop and (b) two-hop time-slot leasing-based optimal algorithm.

When the time-slot reservation phase is recursively performed and the time slots with the QoS connection requirement are reserved, the designated destination node sends the acknowledgement packet (RREP) back to the designated source node to reserve the time slots along the chosen path. All unselected time slots of all existing routing paths that were received by the designated destination node are released. The detailed optimal priority-based algorithm of the time-slot reservation phase follow the rules of Chen *et al.* [2]. Without loss of generality, a path $(M_0, s_0, M_1, s_1, ..., W, X, Y, Z, ..., M_i)$, named *PA*, and a priority list, $PL(i)_{L(W,X,Y,Z)}$, are also given, and the modified rules are presented as follows.

(G1) Randomly select link L(*X*, *Y*) with minimum shared free time slots from *PA*, and then divide *PA* into two sub-paths, PA_1=(M_0,..., *X*) and PA_2=(*Y*,..., M_i), where *PA*=PA_1+PA_2.

(G2) Assign a different priority value that indicates the affected degree with neighboring links to each link collected in the preceding two phases, the *available QoS routing-path search phase* and *time-slot leasing-based path-recovery phase*.

(G3) Apply the priority-based scheme to reserve time slots on link L(*X*, *Y*).

(G4) Recursively execute the time-slot reservation operations of steps G1 and G2 on all paths received by the destination node until all of the links have been selected for time-slot reservation.

Figure 7 shows a traversal tree, also called a *time-slot reservation tree* (TSR-tree), to present all of the recursive operations without using the time-slot leasing-based approach. Figure 8(a) shows the one-hop time-slot leasing-based TSR-tree and routing graph to present all of the recursive operations when the one-hop time-slot leasing-based approach is adopted. Figure 8(b) shows the two-hop time-slot leasing-based TSR-tree and its routing graph.

A distributed time-slot leasing-based QoS routing (DTQ) protocol can be easily applied if every three hop neighboring nodes are considered to simply perform the CTQ protocol. We omit the details herein.

4 Performance Analysis and Comparison Results

In this section, we implement five algorithms to verify our TQ routing protocol's analytic observations. In the following simulator, we use "CTQ", "DTQ", "CCQ", "DCQ", and "KIM" to denote our CTQ algorithm, our DTQ algorithm, Chen *et al.* [2]'s CCQ algorithm, Chen *et al.* [2]'s DCQ algorithm, and Kim [4]'s algorithm, respectively. C++ simulation programs were developed to achieve the requirements of five algorithms. The simulation programs use the same environments of Network Simulator (ns-2) and BlueHoc (IBM Bluetooth simulator) [5]. The performance metrics of the simulations are given as follows.

- *Success rate*: the value of the sum of successful QoS route requests divided by the sum of total QoS route requests.
- *Throughput*: the value of all data bytes received by all devices per unit time.

A high success rate, high throughput, high bandwidth utilization, and low slot occupation are the main issues of an efficient QoS routing protocol in Bluetooth WPANs. In the following, we demonstrate that our time-slot leasing-based QoS routing protocol can significantly improve the success rate, throughput, slot occupation, and bandwidth efficiency from several prospects.

4.1 Performance of Success Rate

We investigate the effect of various numbers of QoS requests on the success rate. Figure 9 gives the simulation results between success rates vs. the number of QoS requests for four QoS requirement scenarios. The four QoS requirement scenarios are *DM*1: *DM*3: *DM*5 = 1: 1: 1, 3: 1: 1, 1: 3: 1, and 1: 1: 3 as illustrated in Fig. 9(a)-(d), respectively. In general, the success rate of our CTQ was greater than that of Chen *et al.* [2]'s CCQ. Similarly, the success rate of our DTQ was greater than that of Chen *et al.* [2]'s DCQ. This is because that our CTQ routing protocol additionally applied the slave-master-slave network to construct the QoS routing path. In case the connection request fails, the routing is immediately switched to the backup slave-to-slave time-slot leasing-based routing paths to support the

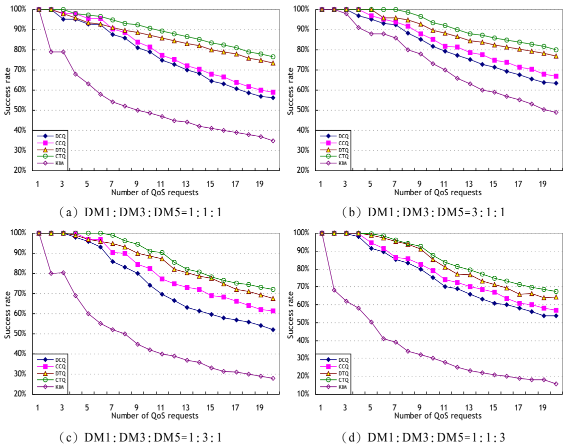

Figure 9. Success rate vs. the number of QoS requests.

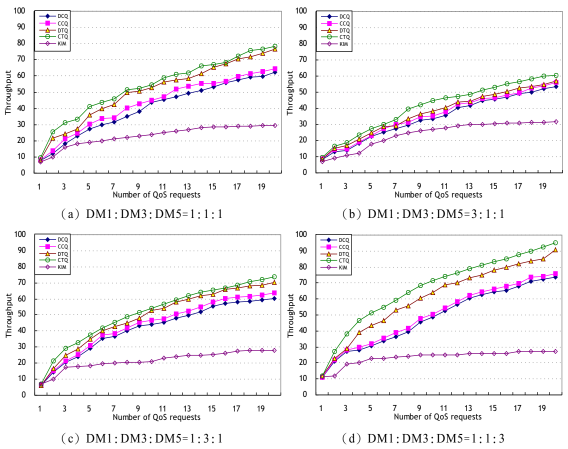

Figure 10. Throughput vs. the number of QoS requests.

transmission QoS. This allowed CTQ and DTQ to have better results for the success rate than CCQ, DCQ, and KIM. General speaking, the success rate of CTQ $>$ that of DTQ $>$ that of CCQ $>$ that of DCQ $>$ that of KIM as illustrated in Fig. 9. On the other hand, most time slots were free for request, and the success rate was almost 100% in the initial stage of simulation. Higher QoS requests produced lower success rates. Figure 9(b)-(d) show that the values of success rates in average case are 3: 1: 1 $>$ 1: 3: 1 $>$ 1: 1: 3. This is because if the frequency of a high traffic pattern is higher than that of a low traffic pattern, then the system throughput increases, the time slot occupation increases, and the success rate decreases.

4.2 Performance of Throughput

The greater the number of data bytes received, the higher the throughput will be. This means that the higher the success rate is, the higher the throughput will be. Figure 10 shows the simulation results of throughput vs. the number of QoS requests under four QoS requirement scenarios. The throughput of our CTQ was greater than that of CCQ, and the throughput of our DTQ was also greater than that of DCQ, as displayed in Fig. 10. This is because the success rate of our CTQ is greater than that of CCQ and the success rate of our DTQ is greater than the success rate of DCQ, as shown in Fig. 9. For instance as illustrated in Fig. 9(a), when the number of QoS requests exceeds than 10, the success rate of CTQ is trivially greater than that of CCQ, then the throughput of CTQ is still greater than that of CCQ as shown in Fig. 10(a). In the four QoS requirement scenarios, the values of throughput in an average case are 3: 1: 1 $<$ 1: 3: 1 $<$ 1: 1: 3, as displayed in Fig. 10(b)-(d). This is because the numbers of transferred packets are 3: 1: 1 $<$ 1: 3: 1 $<$ 1: 1: 3. Figure 10(c) shows that the ratio of traffic pattern is $DM1$: $DM3$: $DM5$ = 1: 3: 1. The throughput of CTQ is nearly 74 kbytes/s and greater than the CTQ value of Fig. 10(b). Generally speaking, the results of throughput are CTQ $>$ DTQ $>$ CCQ $>$ DCQ $>$ KIM and $DM1$: $DM3$: $DM5$ = 3: 1: 1 $<$ $DM1$: $DM3$: $DM5$ = 1: 3: 1 $<$ $DM1$: $DM3$: $DM5$ = 1: 1: 3.

5 Conclusions

To provide the QoS service, our integrated interpiconet scheduling approach, named the TQ routing protocol, which offers QoS guarantees of Bluetooth scatternets in WPANs, is presented in this paper. The slave-master-slave communication is the regular data transmission operation in Bluetooth networks and the master node is the communication bottleneck. To alleviate the ”transmission holding” problem, we have proposed a new time-slot leasing-based scheme to provide extra slave-to-slave QoS communication capability to reduce the workload of master nodes and lower the missing rate of QoS requests. Finally, our simulation results have demonstrated that the time-slot leasing-based QoS routing protocol can significantly improve the success ratio, delay time, throughput, and bandwidth utilization.

References

[1] J. Bray and C. F. Sturman. ”Bluetooth: Connect without Cables”. Prentice Hall, 2001.

[2] Y. S. Chen and K. S. Liu. ”A Credit-Based On-Demand QoS Routing Protocol over Bluetooth WPANs”. *Submitted to Wireless Personal Communications*, Nov. 2004.

[3] 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, Mar. 2004.

[4] Y. M. Kim, T. H. Lai, and A. Arora. ”A QoS-Aware Scheduling Algorithm for Bluetooth Scatternets”. In *Proceedings of IEEE the International Conference on Parallel Processing*, pages 455–462, 6-9 Oct. 2003.

[5] IBM research. ”BlueHoc, IBM Bluetooth Simulator”. Technical report, http://www-124.ibm.com/developerworks/opensource/bluehoc/, Feb. 2001.

[6] W. Zhang, H. Zhu, and G. Cao. ”On Improving the Performance of Bluetooth Networks Through Dynamic Role Management”. Technical report, CSE-01-018 Pennsylvania State University, also available at http://www.cse.psu.edu/ gcao/paper/bluetooth.ps, May 2001.

[7] W. Zhang, H. Zhu, and G. Cao. ”Improving Bluetooth Network Performance Through a Time-Slot Leasing Approach”. In *Proceedings of Wireless Communications and Networking Conference (WCNC)*, volume 2, pages 592–596, Orlando, FL, Mar. 2002.