An Integrated Approach to Scatternet Traffic Management in Bluetooth Ad Hoc Networks

Liza Lai-Yee Shek and Yu-Kwong Kwok
Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong

Abstract— **Scatternet management remains to be one of the most crucial research issues for Bluetooth networks, despite that Bluetooth devices have proliferated in the commercial market. In this paper, we describe our proposed integrated scheme for effective scatternet management. Our proposed scheme contains four main mechanisms to address the different facets of the problem, namely Compensation-Based Time-Slot Assignment (CTSA), Traffic Differentiation Queueing (TDQ), Adaptive Master-Slave Switching (AMSS), and an Enhanced AODV algorithm for ad hoc routing. We have built a comprehensive Bluetooth simulator and performed extensive simulations to evaluate the proposed IARTSS. We find that our proposed scheme can perform well under a wide variety of practical circumstances, and provides efficient and high performance intra-piconet and inter-piconet communications.**

Keywords: Bluetooth, scatternet, ad hoc networks, routing, time-slot scheduling, pervasive.

I. Introduction

Wireless networks have proliferated during the last decade and have become an essential part of our daily lives. Wireless networks can be classified into infrastructured and infrastructureless networks. Infrastructured networks refer to those networks with fixed and wired gateways or access points. The most common form of infrastructured networks is manifested by a cellular network that is used in the mobile phone system. In such a system, base-stations act as bridges to connect the mobile terminals together. Infrastructureless mobile networks, which are also known as *ad hoc networks*, refer to those networks that have no *fixed* routers or access points. Their main characteristic is that all of the nodes are peers, and are capable to move around in the network and connected among each other dynamically in an arbitrary manner. Nodes in this network function as routers as well as to discover and maintain routes to other nodes in the network. Typical examples of usage are in meetings and conferences where people wish to share information quickly without the need to set up an infrastructure beforehand. Since early 1970's, much research has been done on developing efficient protocols for ad hoc mobile networks [6]. Our study is focused on traffic management in ad hoc networks[1] based on a popular short-range wireless technology—Bluetooth [3].

A single Bluetooth network is called a *piconet*, involving one master and several slaves. With the formation of a network of Bluetooth piconets, called a *scatternet* in which there are some *bridge nodes*, there is a need to develop an ad hoc routing protocol to allow efficient route setup so that nodes in different piconets can communicate. For example, imagine a home user walks into the kitchen, which is 15 m away from his/her bedroom, but he/she wants to switch off the TV in the bedroom. It would then be desirable to have a route swiftly set up between the user's Bluetooth device (e.g., a PDA with remote control capability) and the TV so that the command can be sent to the TV from the PDA.

The specification of scatternet formation/management has not yet been defined and thus, there is a pressing need to address the details of developing an ad hoc routing protocol in Bluetooth. Many ad hoc routing protocols have been proposed, but they all assume a wideband network and large packet size environment such as in IEEE 802.11. They do not pinpoint the characteristics of Bluetooth. In [2], a routing method is proposed for Bluetooth network; however, it does not address the details of developing an ad hoc routing protocol in Bluetooth. Also, since it is developed based on the routing method of DSR, there is a drawback that it requires an unbounded packet size for creating and storing the route information. Furthermore, its vector notation for Bluetooth nodes cannot address the problem of link failures, and this may result in false route formation. Bluetooth network has much more constraints and resources limitations than many other wireless network. Hence, an efficient and practical ad hoc routing protocol should be developed. There are some other recent work on issues related to scatternet traffic management [1], [2], [4],

This research was supported by a grant from the Hong Kong Research Grants Council under project number HKU 7162/03E.

[1] A Bluetooth network is still *ad hoc* in that the master-slave relationship is not *fixed* because a slave can become a master by performing a master-slave switching [3].

[5] but these efforts are largely focused on rather restricted facets of the whole problem.

Apart from the routing protocol, another concern for Bluetooth is its scheduling algorithm. Bluetooth uses a centralized polling scheme to distribute services among the nodes in the network by allocating time-slots through masters. This centralized polling scheme controls the traffic and throughput of the network as it controls the amount of uplink traffic. With the formation of scatternet and availability of ad hoc routing, communications between distant devices become possible. As a result, a more carefully designed scheduling algorithm is needed.

In this paper, we propose an integrated ad hoc routing and time-slot scheduling scheme (IARTSS), which is composed of four novel resource management mechanisms:

- Compensation-Based Time-Slot Assignment (CTSA)
- Traffic Differentiation Queueing (TDQ)
- Adaptive Master-Slave Switching (AMSS)
- Enhanced AODV (Ad Hoc On-Demand Distance Vector) [7] for Ad Hoc Routing

Due to space limitations, in this paper we omit the detailed and formal descriptions of the above four mechanisms, which can be found in the thesis [8]. Instead, we provide a summary of their characteristics below.

Compensation-Based Time-Slot Assignment (CTSA) is a scheduling algorithm tailor-made for Bluetooth. With the formation of scatternet, some nodes have to take up the role of routers or bridges to interconnect different piconets together. As Bluetooth uses a frequency hopping scheme, a node cannot listen to two piconets simultaneously. As a result, it can only synchronize and participate in one piconet at a time. This implies that these bridge nodes have to multiplex their time among different piconets. However, as Bluetooth adopts a centralized polling scheme for time-slots allocations, these nodes have to compete for time-slots with all the other nodes when they switch back to another piconet. Consequently, the resources acquired by these nodes from the piconet are fewer. In a scatternet environment, traffic routing across the network can be abundant. The bridge nodes play an important role in routing packets across different piconets and they become the bottlenecks of the system. Most of the existing scheduling algorithms allocate time-slots according to the loads of the queues, and if we apply these algorithms to Bluetooth directly, we may have ignored the bridging and uplink traffic problems. CTSA is a carefully designed scheduling scheme for Bluetooth. It exploits the properties of the absence of bridge nodes to calculate the time-slots allocation for each nodes more effectively.

Traffic Differentiation Queueing (TDQ) is a technique to separate different traffic to different queues in Bluetooth's link layer. Traffic is differentiated as "self traffic" and "forward traffic" in this scheme and they are put into two different queues. Those being generated by the Bluetooth host itself are classified as "self traffic" while those being forwarded to it are classified as "forward traffic". The Bluetooth link layer serves these queues based on an adaptive service ratio. This scheme helps routing packets across scatternet more efficiently and can significantly improve the throughput and delay.

Adaptive Master-Slave Switching (AMSS) is a hold time calculation scheme between bridge nodes, which decides the duration for a node to stay in a piconet. As bridge nodes in Bluetooth network need to multiplex their time to different piconets in a scatternet, the time spent on a piconet greatly affects the overall system performances. Hence, an adaptive calculation of the hold time is required. AMSS addresses this problem and uses an adaptive method based on queue lengths and utilization ratio to determine the hold time for a piconet. This scheme allocates time between piconets in a more effective way.

Enhanced AODV for Ad Hoc Routing is a routing protocol developed to facilitate ad hoc routing in Bluetooth network. AODV is used because it is one of the most representative ad hoc routing protocols and it performs well in most of the wireless network environment. In our enhanced AODV routing protocol, we have modified AODV for Bluetooth so that it becomes suitable for Bluetooth's narrow bandwidth and small packet size features.

We have investigated the possible problems of Bluetooth scatternet, and evaluated the performances of some existing scheduling algorithms in Bluetooth network in this research. As indicated by the simulation results shown in Section II, we find that our proposed IARTSS outperforms the evaluated scheduling algorithms and is more suitable in the Bluetooth scatternet environment.

II. Performance Results

A. Overview

We have implemented three existing scheduling algorithms and the IARTSS scheme in our Bluetooth simulator. We have also conducted extensive simulations under various traffic loads and measured the throughput, the end-to-end packet latency, and the delivery rate for all the scheduling algorithms. The results show that IARTSS works well in Bluetooth networks. In this section, we first discuss the simulation environment in detail, followed by a presentation of the results and our interpretations.

As mentioned earlier, most of the existing scheduling algorithms are designed for point-to-point or point-to-multipoint uni-directional traffic and are unsuitable for

Bluetooth's centralized polling networking environment. Thus, in order to evaluate the existing scheduling algorithms across the Bluetooth network, we have modified well-known Deficit Round Robin (DRR) [9] in two different enhanced versions so that the DRR approach can be used in a Bluetooth network. Apart from the two modified deficit round robin scheduling algorithms, the straightforward round robin (RR) scheduling algorithm is also used for baseline comparison. These algorithms are the only approaches currently being used in practical Bluetooth systems. The two versions of deficit round robin algorithms are described in detailed in [8].

B. Simulation Environment and Parameters

In our performance study, Bluetooth networks with a variable number of nodes are examined. The Bluetooth topology created is a Bluetooth network with a maximum of three masters and two relays, and thus, restricting the number of piconets to be a maximum of three in our simulation environment. Unlike the scatternet topology assumed in [10], where most of the links are parallel and connected like a linked list. Our simulation environment simulates the worse case situation where up to seven links exist in a piconet and a master has to allocate its time-slots among the maximum number of slaves. In addition to using different topologies, we also varied different load and scatternet traffic across the network and study their effects in the implemented algorithms.

We investigated the performance of scatternet traffic in two major dimensions, the number of hops and the scatternet size. The number of hops varies from single hop within a piconet to a maximum of 6 hops across the whole scatternet. The size of the scatternet varies from 5 nodes to 20 nodes to simulate the effects of a single piconet to multiple piconets. Their effects on the throughput, the delay and the delivery rate on piconet and scatternet sessions are studied thoroughly.

C. Performance Metrics

We evaluated our proposed IARTSS with three main metrics: (1) the average end-to-end latency, (2) the average throughput, and (3) the delivery rate. The end-to-end latency is measured by the time required for a packet to be created at a source node until it is received and processed by its destination. Its average value is calculated by the total end-to-end latency divided by the total number of packets received. The average throughput is calculated by the received number of bits divided by the packet's end-to-end latency. The delivery rate is defined as the total number received in the network divided by the total number of packets created in the network.

Bluetooth supports various data packet types, with different payload sizes and transmission time-slot requirements. Thus, in our performance study, we have used two major types of data packet for testing and they are the DM3 and the DM5, respectively.

As Bluetooth networks have not been widely used yet, the traffic patterns are still unknown. Therefore, in this performance study, we use two major types of hypothetical traffic sources to represent some common traffic patterns in wireless network and they are the Constant Bit Rate (CBR) traffic source and the Exponential (EXP) traffic source, respectively. We used 8 kbps CBR traffic source that transmits 160 bits in 20 ms time interval to represent applications that send small sized packets regularly. These applications vary from simple Bluetooth enabled mice control to voice transmissions. The EXP traffic represents applications with sudden burst of heavy packets. These applications include various file transfer applications such as E-mail applications and FTP. In this performance study, we use EXP source of 56 kbps to represent the rate of a modem. It should be noted that higher bit rates are not suitable for a Bluetooth network because the theoretical peak rate (rarely achievable in practice) for a unidirectional link in the simplest piconet environment (i.e., having only one master and one slave) is only 732 kbps.

To evaluate the system performance of Bluetooth scatternet, we studied the three mentioned metrics to piconet traffic sessions and scatternet traffic sessions in each simulation. We varied different traffic load in the studies of effects of number of hops, and they are the heavy traffic, the medium traffic, and the light traffic. The heavy traffic consists of a total of 8 CBR sources and 8 EXP sources. The medium traffic consists of a total of 4 CBR sources and 4 EXP sources while the light traffic consists of a total of 2 CBR sources and 2 EXP sources. In each cases, there is 1 CBR and 1 EXP sources that transmit across the scatternet. The rest of the traffic are confined in piconet sessions to simulate the influences of various background traffic on scatternet traffic. The simulation environment is justified because Bluetooth network is originally developed for short-ranged intra-piconet communications and thus, most traffic load is contributed by intra-piconet transmissions.

In studying the effects of scatternet size, the total number of traffic sources equal to the total number of nodes, where each node has either one of the traffic sources. Two random nodes are chosen to perform the scatternet traffic, while all the others have intra-piconet traffic only. In the simulation evaluations, we also separate the CBR and EXP traffic for comparisons in each case. We now move on to introduce the detail parameters used in each simulations.

As mentioned earlier, existing Bluetooth networks do not support any routing protocol, we use the Enhanced AODV for Ad Hoc Routing implemented as a routing protocol to test different existing scheduling algorithms in our performance study.

In the RR scheduling algorithm, we tested the effects of number of hops and scatternet size on the average delay, throughput and delivery rate. This is a simple scheduling algorithm which requires no information from the masters and the slaves. Therefore, all the parameters used in the algorithm are retained as such in the original Bluetooth standard.

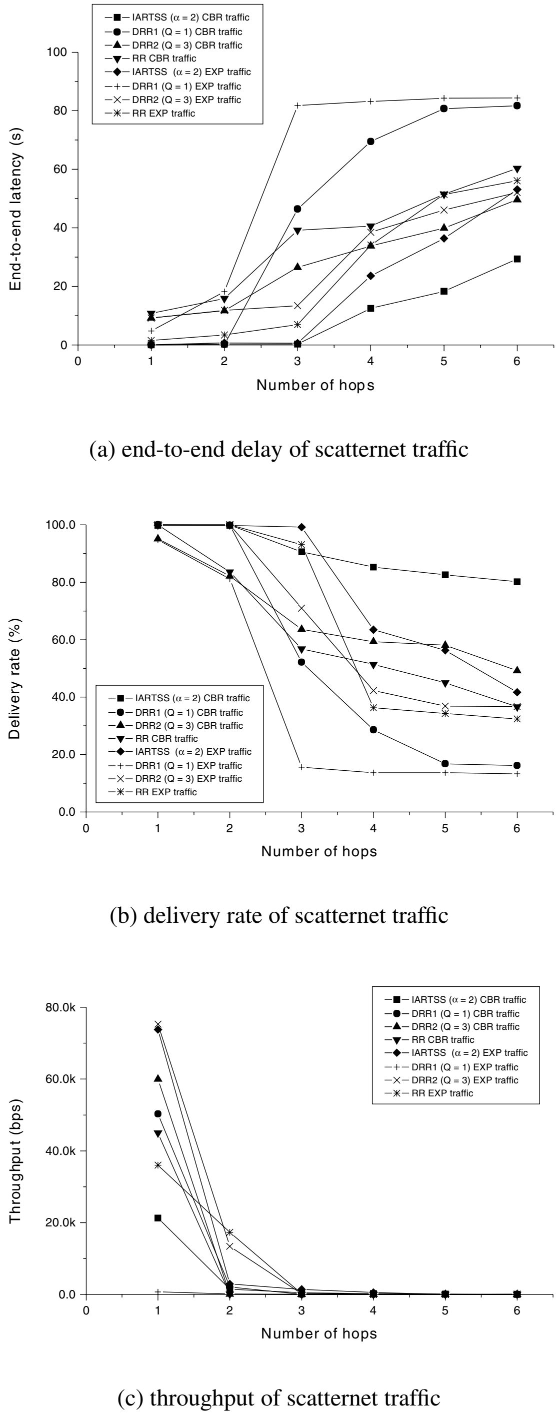
For DRR1 and DRR2, we varied different quantum (Q) sizes for testing. This value is added to the deficit counters of all active queues when the master has served one complete round of the queues. It is set to the size of 1 payload, 2 payload and 3 payload respectively for the cases Q = 1, 2, and 3. The size of the payload varies according to the type of data packets used. For example, if data packet type is DM3, with a payload of 1000 bits. In case Q = 1, it represents a quantum size of 1000. In case = 2, it represents a quantum size of 2000 while in case Q = 3, it represents a quantum size of 3000. In these two algorithms, the value of Maximum Quanta is set to be 5 times the payload. In the proposed IARTSS, the constant which governs the weighting between the elapsed time-slot factors and queuing factors are set to be 1 and 2 for our experiments.

D. Results and Interpretations

The performance results comparing different approaches (IARTSS, DDR1, DRR2, RR) using the three performance metrics (delay, delivery rate, throughput) under heavy load condition are shown in Figure 1 for the multi-hop scatternet traffic using DM3 packet type (results for light and medium load conditions, as well as results for DM5 packet type can be found in [8]). Results for varying number of nodes in the network are shown in Figure 2. The performance results for the background intra-piconet traffic are shown in Figure 3.

The average end-to-end packet latency generally increases significantly when the number of hops increases to 3 while the throughput and the delivery rate drops drastically. It is because when the number of hops increases to 3, packets are needed to route through a bridge node where half of time is being shared by another piconet.

In the effects of scatternet size, the performance is generally better than in the effects of the number of hops. It is because the source and destination pairs for scatternet traffic are generated randomly, and thus, varying the average number of hops each time. For most of the algorithms, the effects change more greatly when the number of nodes

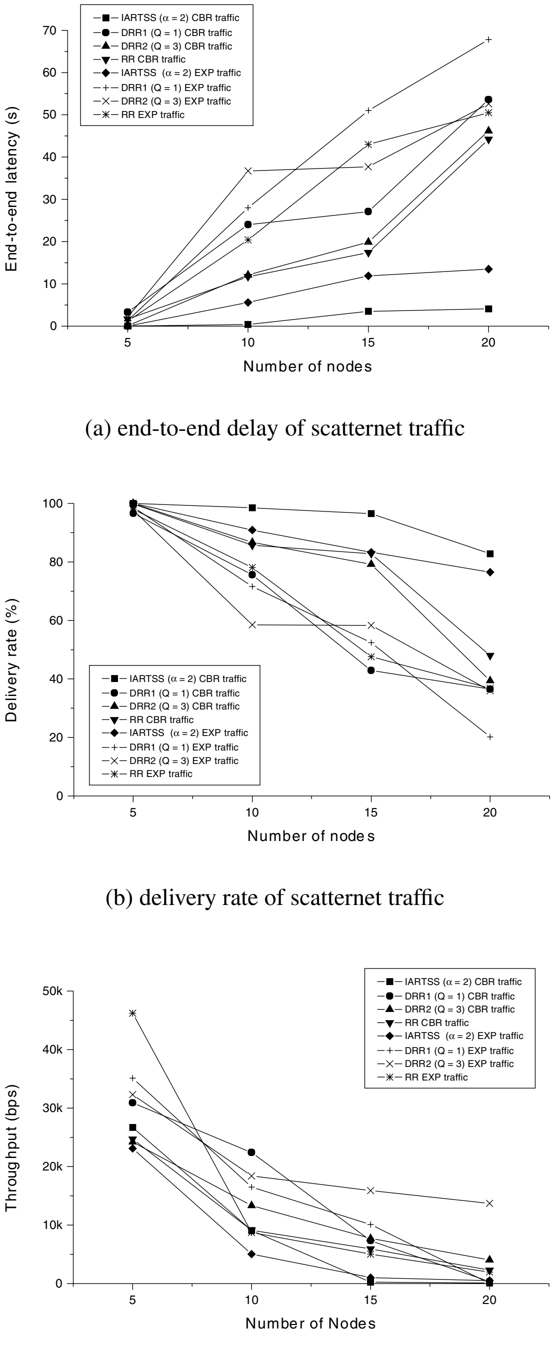


(a) end-to-end delay of scatternet traffic

(b) delivery rate of scatternet traffic

(c) throughput of scatternet traffic

Fig. 1. Comparison of different approaches using the DM3 packet type under the heavy load conditions.

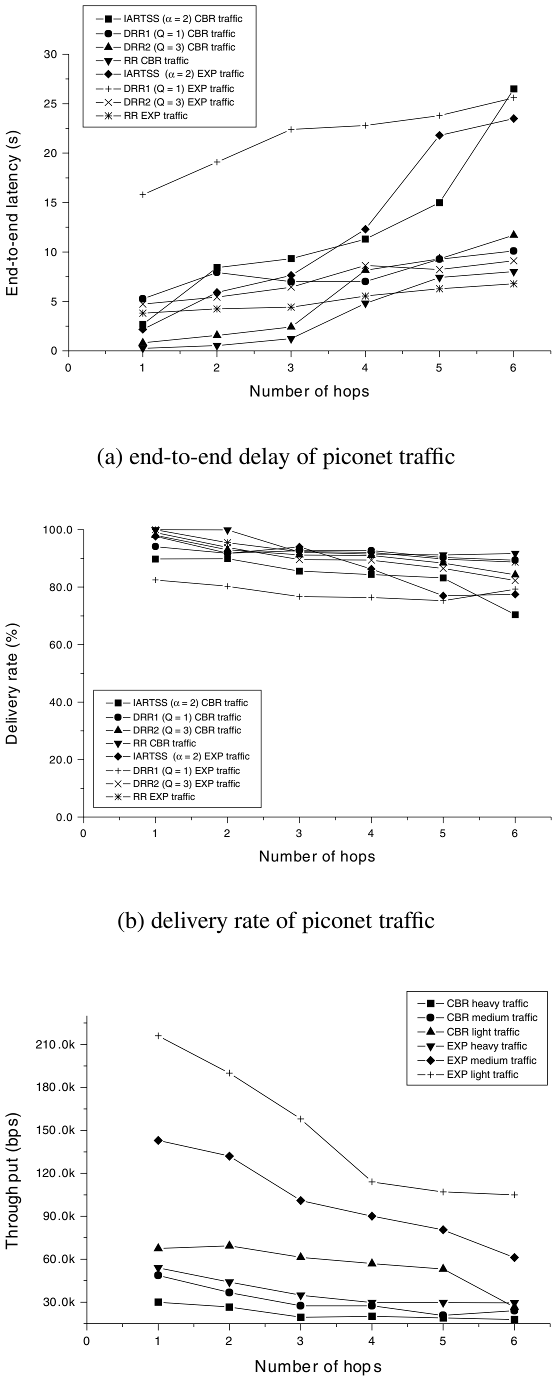


(a) end-to-end delay of scatternet traffic

(b) delivery rate of scatternet traffic

(c) throughput of scatternet traffic

Fig. 2. Comparison of different approaches using the DM3 packet type with various number of nodes in the scatternet.



(a) end-to-end delay of piconet traffic

(b) delivery rate of piconet traffic

(c) throughput of piconet traffic

Fig. 3. Comparison of the performance of the background piconet traffic using the DM3 packet type under the heavy load conditions.

increase from 5 to 10 and from 15 to 20. It is because for these cases, it involves setting up more piconets across the scatternet, and therefore, increasing the number of bridge nodes in the scatternet.

The throughput for EXP traffic are usually much higher than those of CBR traffic. It is because EXP traffic has more bits in a packet and can have more efficient use of the payload of the data packets. Thus, the destination can receive more bits in a single transmission and results in a higher effective throughput.

DRR1 has been investigated in 3 cases and they are the cases of Q1, Q2 and Q3. It is found that Q2 has higher piconet throughput than Q1, but Q1 has higher scatternet throughput than Q2. It is because Q2 favours the downlink traffic more than Q1 and hence makes piconet traffic more favorable. For the case of Q3, it shows similar delay and throughput as Q1 and Q2, but a better performance in delivery rate. The reason is that Q3 further favours the downlink traffic and allows traffic to get through during the turn for the master to forward it.

DRR2 has also been investigated in 3 cases of Q1, Q2 and Q3. In DRR2, we found that Q1 works better than Q3, and Q3 works better than Q2 in terms of all 3 measured aspects. It is because DRR2 tends to discriminate the link of the inactive and low traffic sessions more. This allows higher traffic to send across the network and results in a better overall system performance. However, it is unfair to other links.

Two cases for IARTSS were studied in this performance study. They are the $\alpha = 1$ and $\alpha = 2$ cases. The other values were not tested because larger values make the elapsed time factor too large and consequently, the effect of queue size would be incomparable to the effect of the elapsed time. It is found that the case of $\alpha = 2$ has higher delay than $\alpha = 1$, but the throughput and the delivery rate are improved. It is because when a higher weighting is given to the elapsed time factor, it allows a master to allocate more time-slots to the bridge nodes that switch back to its piconet. As a result, more packets are allowed to send across the scatternet and it explains the reason why the throughput and the delivery rate are improved. As the masters tend to allocate more time-slots to the bridge nodes when they return to their piconet, the factor of utilization in the calculation of hold time increases. The hold time calculated for a piconet is therefore smaller, and this makes the hold time for another piconet longer. As a result, the delay increases.

It is found that IARTSS generally works better than the other algorithms in scatternet environment. It is because with the three novel mechanisms, we can have better time-slots allocations and control of hold time to minimize the synchronization overhead without affecting the delays greatly. DRR1 and DRR2 generally work badly because they tend to favour downlink traffic and discriminate the low traffic uplinks. Nevertheless, sometimes they perform better because these discrimination effects allow more downlink traffic to be served.

However, in heavy traffic piconet environment, IARTSS may not work as well as the other algorithm. It is because when the network traffic is in saturation, like in the case of heavy traffic, IARTSS tends to favour the scatternet traffic because of the effect of TDQ, where forward traffic is given a higher priority. For example, in delay, we found that when it is in heavy traffic environment, the result for IARTSS in piconet may be doubled when compared with other algorithms. However, it improves the scatternet traffic delay by decreasing the delay to 25% of the other values. There is also a large improvement in delivery rate for heavy traffic scatternet session. There is a 60% improvement in scatternet sessions with only less than 10% sacrifice of delivery rate in piconet sessions. IARTSS allocates time-slots in a more efficient way, and thus can improve the performance under various practical scenarios.

The proposed IARTSS generally works well in heavy traffic environments for both CBR and EXP traffic source than other existing fair queuing and scheduling schemes.

References

[1] S. Baatz, M. Frank, C. Kuhl, P. Martini, and S. Christoph, “Bluetooth Scatternets: An Enhanced Adaptive Scheduling Scheme,” *Proc. IEEE INFOCOM’2002*, pp. 782–790, June 2002.

[2] P. Bhagwat and A. Segall, “A Routing Vector Method (RVM) for Routing in Bluetooth,” *Proc. 1999 IEEE Int’l Workshop on Mobile Multimedia Communications (MoMuC’99)*, pp. 375–379, Nov. 1999.

[3] J. Bray and C. F. Sturman, *Bluetooth: Connect Without Cables*, Prentice-Hall, 2001.

[4] A. Capone, R. Capoor, and M. Gerla, “Efficient Polling Schemes for Bluetooth Picocell,” *Proc. ICC’2001*, vol. 7, pp. 1990–1994, June 2001.

[5] V. B. Misc and J. Misic, “On Bluetooth Piconet Traffic Performance,” *Proc. IEEE PIMRC’2002*, Sept. 2002.

[6] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” *Proc. ACM SIGCOMM’94*, vol. 24, pp. 234–244, Oct. 1994.

[7] C. E. Perkins and E. M. Royer, “Ad-Hoc On-Demand Distance Vector Routing,” *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applicatons (WMCSA’99)*, pp. 90–100, Feb. 1999.

[8] L. L.-Y. Shek, *Ad Hoc Routing and Time-Slot Scheduling in Bluetooth Networks*, M.Phil. Thesis, The University of Hong Kong, Dec. 2002.

[9] M. Shreedhar and G. Varghese, “Efficient Fair Queueing Using Deficit Round Robin,” *IEEE/ACM Trans. Networking*, vol. 4, no. 3, pp. 375–385, June 1996.

[10] G. Tan, *Self-Organizing Bluetooth Scatternets*, S.M. Thesis, MIT, Jan. 2002.