ADAPTIVE ROLE SWITCHING PROTOCOLS FOR IMPROVING SCATTERNET PERFORMANCE IN BLUETOOTH RADIO NETWORKS

Chih-Yung Chang, Kuei-Ping Shih, Shih-Chieh Lee, Chih-Hsiung Tseng

Dept. Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan, cychang@cs.tku.edu.tw

Abstract - Bluetooth is a low-power, low-cost, and short-range wireless technology. A well structured scatternet with the appropriate number of piconets and bridges for a specific traffic pattern will increase the performance of a Bluetooth network. However, the structure of a scatternet is difficult to control or predefine because that the scatternet is formed using a distributed procedure, with the master and slave connected at random. The participation of mobile Bluetooth devices in a scatternet at different times also increases the difficulty of maintaining a good structure. A badly structured scatternet exhibits the following characteristics. Firstly, too many bridges in the scatternet will create a guard slot overhead associated with bridge switching among the participated piconets, increasing the probability that a packet is lost. Secondly, too many piconets in a communicative range will cause packet collision and thus degrade the performance. Unnecessary piconets also lengthen the routing path, delaying the transmission of packets from source to destination. This work proposes a distributed scatternet reconstruction protocol for dynamically reorganizing the scatternet. The unnecessary bridges and the piconet can be dynamically removed by applying the role switching operation, improving the packet error rate, saving guard slots, and reducing the average routing length. Experimental results reveal that the proposed protocol improves the data transmission performance of a Bluetooth scatternet.

Keywords – Role Switch, Scatternet, Bluetooth, Routing

I. INTRODUCTION

Bluetooth [1] is a low-power, low-cost and short-range wireless technology. A *piconet* consists of a master and ups to seven active slaves that share the same channel as the master. Each piconet has its own hopping sequence that is controlled by the master. The master device transmits packets in even slots and the slave devices transmit packets in odd slots. A device that participates multiple picoents is called a *bridge*. Bridge devices can participate different piconets on a time division multiplex basis. Several piconets connected by common bridges form a connected *scatternet*. The bridge device can relay packets from one piconet to another.

Recently, a comprehensive study proposed scatternet formation protocols [2][3] for constructing a connected scatternet. Most protocols seek to reduce the scatternet formation time or increase the probability of constructing a connected scatternet. The number of piconets and bridges and the role of each device are hard to control or predefine, because the scatternet is formed by a distributed procedure, with the master and slaves connected at random, such that the number of devices to be included in the scatternet and the traffic pattern of devices in a communicative range is unknown. Mobile Bluetooth devices that participate in a scatternet at different times also increase the difficulty of maintaining a well structured scatternet.

A well structured scatternet has the appropriate number of piconets in a communicative range and the proper number of bridges, and every device has its proper role. The performance of the scatternet depends strongly on these factors, which are described below. Each piconet has its own hopping sequence. Increasing the number of piconets will enhance the channel utilization. However, two different piconets' transmitting data on the same channel will result in the loss of packets. As the number of piconets is increased, the hopping sequences of the two devices that are in different piconets are likely to collide with each other[4][5], increasing the probability that a packet is lost and reducing the performance of the scatternet by requiring packet retransmission. Besides, increasing the number of piconets also increases the mean routing path, delaying the construction of the rout and the transmission of packets, and increasing the bandwidth and power consumption. When the packet collision rate exceeds a threshold, appropriately reducing the number of piconets will improve the performance by reducing the packet error rate and the average route length.

As well as the number of piconets, the number of bridges is another key determinant of the performance of a scatternet. Two piconets that share a common bridge can transmit data between each other. However numerous devices are assigned to be bridges, in constructing a connected scatternet. The time for bridge switching from one piconet to another is referred to as the *guard time*. An excess of bridges cannot improve piconet connections and they generate guard slots. Bridge switching among different piconets also makes difficult scheduling among the masters of the piconets in which the bridge participates, worsening the packet loss problem, thereby degrading data transmission. Therefore, properly eliminating unnecessary bridges will improve the performance of a scatternet.

As well as the number of piconets and bridges, role assignment is another key parameter in determining the performance of a scatternet. Devices with inappropriate roles create new piconets and bridges. *Role switching* is an operation that exchanges the roles of master and slave devices very rapidly. A master can also send a role switch request to one of its slaves to require that slave to take over all of its master's resources. In this paper, each device monitors the bandwidth requirement, the guard slot overhead and the packet loss parameters. While these parameters are considered to be the main reasons that drop the performance, a device will initiates the role switch request to reduce the number of piconets and bridges.

This work analyzes the effects of role switching on scatternet performance. A dynamic role switch protocol is developed to improve data transmission performance by restructuring the scatternet in a distributive and dynamic fashion. The remainder of this article is organized as follows. Section 2 introduces the basic operations associated with role switching. Section 3 analyzes the usages of role switching operation. Section 4 proposes a distributed role switching protocol useable by devices to determine when and how to gain advantage from executing role switching. Section 5 experimentally examines the proposed protocol. Session 6 draws conclusions.

II. BASIC OPERATIONS OF ROLE SWITCHING

In a Bluetooth radio network, two devices establish a connection by executing Inquiry/Inquiry Scan and Page/Page scan operations.

0-7803-8523-3/04/$20.00 ©2004 IEEE.

In a piconet, a master is connected no more than seven slaves and controls the scheduling of the transmission of data of its slaves. A device that simultaneously participates in multiple piconets is called a *bridge*. Connected by bridges, several piconets can form a connected scatternet.

Sometimes, a master or slave may have to switch roles[6]. *Role switching* enables two devices to exchange roles very quickly, rather then reconnecting by executing the time-consuming inquiry and inquiry scanning processes. When a slave initiates a role switch request, it firstly sends an LMP_slot_offset command to compensate for the discrepancy between the clocks of the two devices. The slave then sends an LMP_switch_req command to ask to switch roles with the master. A master will send an LMP_not_accepted command back to the slave to indicate declination of the request or will approve the slave's request, and sends an LMP_accepted command, before exchanging BD_ADDR information with the slave. The role switching operation involves fewer slots than the Inquiry/Inquiry Scan and Page/Page Scan operations in switching the devices' roles.

Three major types of role switching operation are considered below.

Type 1: *Combining Operations*

As presented in Fig. 1(a), device *b* may establish a connection in the inquiry state when it is a master. As presented in Fig. 1(a), after the piconet P_2 has been established, devices *b* and *a* act as master and slave, respectively. Hereafter, device *b* may send a role switch request to device *a* so that it can play a slave role in piconet P_1. The role switching operation will combine two piconets P_1 and P_2 into a single piconet P_1, as shown in Fig. 1(b). The role switching operation also eliminates the bridge role of device *a*. This type of role switching can reduce the number of bridges and the number of piconets.

Type 2: *Splitting Operation*

A role switching operation also can split a piconet into two piconets. As depicted in Fig. 1(b), in piconet P_1, slave device *b* intends to create a new piconet P_2, where device *b* is the master. Device *b* accordingly initiates a role switch request to device *a*. As depicted in Fig. 1(a), device *b* creates a new piconet P_2 and plays a master role in P_2. Device *a* then alternatively participates in two piconets P_1 and P_2 and plays master and slave roles, respectively. This kind of role switching increases the number of picoents and bridge devices and is referred to as the *splitting operation*.

Type 3: *Take-Over Operation*

Role switching operation can also be used to enable a slave to take over the resources of its master. For example, as indicated in Fig. 2(a), device *b* initiates a role switching request. On receiving the request, device *a* asks the other slaves *c* and *d* to enter the page scan state and transfer their Bluetooth Address and clock information to device *b*. Device *b* thus enters the page state and tries to act as a master, connecting to the slaves in the page scan state. Thereafter, as shown in Fig. 2(b), the master of the constructed picoent will have been changed to device *b*. This type of role switching operation enables a slave to take over all slaves that belong to its master, which operation is called the *take-over operation*.

Since role switching can be applied to Bluetooth devices without entering the inquiry and inquiry scan states, the role switching can be completed within a very few slots, implying that the role switching operation can be cost-effectively applied to reorganize dynamically the scatternet.

The following definitions are used in detailing the role switching operation and the proposed protocol.

<u>*Definition*</u>: Number of Slaves *NOS*(*d*)

The number of active slave managed by master *d* is denoted as *NOS*(*d*). ■

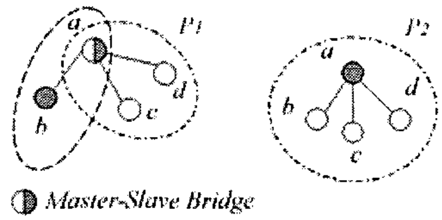
<u>*Definition*</u>: Number of Participating Piconets *NOP*(*d*)

For any device *d*, *NOP*(*d*) is the number of piconets in which device *d* participates. In the case that *Role*(*d*)=*M* or *S*, value *NOP*(*d*) is one. If *d* is a bridge and *Role*(*d*)=*S*/*S* or *M*/*S*, then *NOP*(*d*) >= 2. ■

Three-type role switching is defined below.

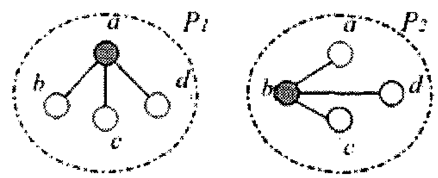
<u>*Definition*</u>: *Combine*(*b*, *a*), *Split*(*b*, *a*), *TakeOver*(*b*, *a*)

Operation *Combine*(*b*, *a*), *Split*(*b*, *a*), *TakeOver*(*b*, *a*) respectively represents the Combination, Splitting, and TakeOver operations, where *b* represents the device that initiates the request and *a* represents the device that responds to the request. ■



(a) Before executing the combination operation
(b) After executing the combining operation

Figure 1: The combining operation of role switching.



(a) Original piconet
(b) Piconet after executing the take-over operation

Figure 2: The execution of take-over operation.

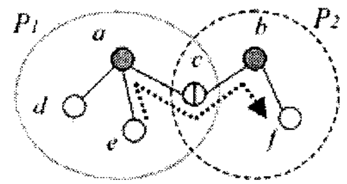
III. ANALYSIS OF ROLE SWITCHING OPERATION

This paper presents an adaptive scatternet restructuring protocol, *ARSP*, for removing unnecessary piconets and bridges and reducing the average length of routes. This section analyzes the effect of role switching. Examples are also presented to show that applying the role switching operaton reduces the packet collision rate, the number of guard slots, and the average length of the routing paths. The *ARSP* is a cross-layer protocol of L2CAP and LMP. *ARSP* monitors factors that include bandwidth requirement, the guard slot overhead, and the packet loss rate of L2CAP. If the performance drops because of the monitored factors, *ARSP* will initiate the role switching request to the LMP layer to reduce the number of piconets and bridges.

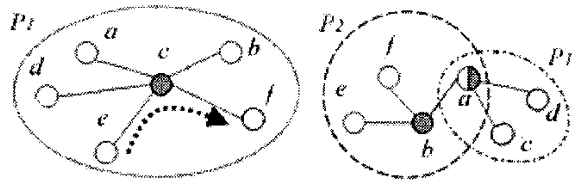
Figure 3 shows the problem that arises from the poor structure of a scatternet. In Fig. 3(a), bridge device *c* simultaneously participates in piconets P_1 and P_2, which are managed by devices *a* and *b*, respectively. When bridge *c* initiates a role switching request to devices *a* and *b*, two piconets P_1 and P_2 will be combined into one piconet P_3 , as shown in Fig. 3(b). The new piconet P_3 includes a new master device *c* and slaves *a*, *b*, *d*, *e*, and *f*. In a scatternet, if several bridges asynchronously apply the *TakeOver* operation to the piconets in which they participate, then the excess piconets and bridges can be effectively eliminated, improving the transmission of packets.

Another purpose of role switching is to reduce the route length. In a scatternet, if the source and destination devices belong to different piconets, the packet should pass through a routing path that connects the source and destination piconets. The length of the routing path governs the performance of data transmission along this route. A long path is associated with a long delay and greater power consumption, increasing the probability of route breakage and packet loss. Yet another function of role switching operation is to reduce the length of a routing path in a scatternet. As shown in Fig. 3(a), a routing path $e \rightarrow a \rightarrow c \rightarrow b \rightarrow f$ exists between the source device e and the destination device f. When bridge device c initiates *TakeOver* operations *TakeOver*(c, a) and *TakeOver*(c, b), the resulting scatternet has the structure depicted in Fig. 3(b), wherein the original routing path has been changed to $e \rightarrow c \rightarrow f$. The number of hops along the route from e to f has been reduced from four to two. Factors such as power consumption, delay time, packet loss, and routes broken, which impact performance, can be improved.

The role switching operation can also change the relationships among roles in a piconet. For example, in Fig. 3(c), device a participates in piconets P_1 and P_2 and as master and slave, respectively. When device a switches to piconet P_2, slaves c and d in piconet P_1 will be idle since their master device a is inactive in piconet P_1. The bandwidth utilization falls accordingly. However, role switching between devices a and c will solve this problem. Inasmuch as the scatternet includes several bridges with poor roles, the role switching operation will save the guard slots and improve the roles.



(a) Bridge device c participates two Piconets.



(b) The piconet after bridge c executing *TakeOver* Operation.

(c) An example that a M/S bridge a participates two piconets

Figure 3: The problems of poor structure of scatternet.

In the following, an estimation of the benefit of role switching is discussed.

<u>*Definition*</u>: *Benefit*(d, a)

Let *traffic* T_i of a routing path R_i represent the number of packets that pass through a route in unit time. Assume that k routing paths $R_1, R_2, \ldots, R_k$ pass through device d. The benefit to device a of the role switching operation initiated by device d is defined as,

$$\sum_{i=1}^{k} T_i \Delta_i$$

, where Δ_i denotes the reduction in the length of route R_i obtained by executing the role switching operation. ■

Role switching exchanges not only the roles of master and slave, but also the topology of the piconet, the packet transmission slot and the controller of the piconet. To ensure that role switching improves the performance of the scatternet, each device should analyze its effect of the scatternet's structure before it initiates the role switching request. The following observations determine when to gain advantage from executing role switching operation.

The next section will present policies for executing role switching, based on the foregoing observations.

IV. ADAPTIVE ROLE SWITCHING PROTOCOL

The preceding section, introduced some good opportunities for executing role switching. Whether or not the execution of the role switching operation is appropriate depends on the role of the initiated device and the context.

<u>*Definition*</u>: $g_t(\gamma, o)$

Consider a Bluetooth bridge $d \in D$, *Role(d)*= *M/S* or *S/S*. Let γ and o represent the packet error rate and the number of packets transmitted from device d in unit time. A role switching *decision function* $g_t(\gamma, o)$ is defined by the value $g(\gamma, o)$ at time t. Each bridge device d will maintain $g_t(\gamma, o)$ to determine whether it should initiate a role switching request at time t. ■

As the packet error rate γ is increased, the role switching operation can effectively reduce the number of piconets. The passing of many packets through a bridge device will cause a bottleneck of inter-piconet packet transmission, so another parameter o is included in function g. Packet error associated with collisions between hopping sequences can therefore be mitigated. The protocol uses a threshold value δ_1 set to a default value. When $g_t(\gamma, o) > \delta_1$, the bridge will execute a role switching operation and determine whether the packet error rate drops. However, the packet error may have another cause, such as interference between other signals, rather than Bluetooth. Consequently, reducing the number of piconets cannot reduce the packet error rate but can reduce parallel transmission among these piconets. The performance improvement must be monitored to guarantee that the role switching operation effectively promotes the suppression of packet collisions and does not eliminate opportunities for parallel transmissions of packets. A threshold value δ_2 is also set in the developed protocol for evaluating the performance improvement associated with the role switching operation. Let $\nabla g_t(\gamma, o) = g_{t+1}(\gamma, o) - g_t(\gamma, o)$, $\nabla g(\gamma, o) < \delta_2$ implies that the packet error may be caused by interference. The role switching operation cannot reduce the packet error rate, so the original bridge will again execute reverse role switching, the *split operation*, to retain the benefits of parallel transmission of packet.

At the moment when a bridge intends to execute a role switching operation, it must collect information from all connected masters. The obtained information is stored in a decision table to enable the bridge to evaluate which master is the most appropriate for initiating role switching.

All devices that intend to execute the role switching operation must follow the following rules to guarantee that the adaptive role switching operation can effectively reduce the number of piconets, guard slots and the length of the routing path.

Rule-1: *Only a bridge device will initiate role switching.* ■

As stated above, a role switching operation initiated by a master or slave cannot reduce the number of piconets or bridges. Rule 1 allows only the bridge device to execute role switching to reduce the packet error rate and guard time overhead. This rule also prevents too many devices from initiating the role switching operation, causing the lock operation executed by the neighbor to increase the complexity of role switching.

Rule-2: *When* $g_t(\gamma, o) > \delta_1$, *the role switching operation will be triggered. If the condition* $g_{t+1}(\gamma, o) - g_t(\gamma, o) < \delta_2$ *is met, then a device that has already executed the role switching operation will execute a split operation to recover the structure of the scatternet at time* $t+1$. ■

One main reason for the high packet error rate is an excessive number of piconets in the scatternet, causing hopping sequences to collide. Another major reason is that the master polls a bridge that is active in another piconet, causing packets to be lost. The role switching operation can effectively reduce the number of piconets and bridges, helping to suppress the packet error or packet lost. Hence, when the decision function $g_t(\gamma, o)$ maintained by bridge d exceeds δ_1, the role switching operation will automatically be triggered by device d. Device d will initiate a role switching request to proper masters. Details will be presented below.

If the packet error arises from interference with a device that is not a Bluetooth device, then reducing the number of piconets is ineffective in alleviating this packet error. Hence, after device *d* initiates a role switching operation, it continues to monitor the performance for one time unit. If the performance improvement is not sufficiently great, say, below a predefined value δ_2, then, the cause of the packet error is considered to be interference from nonBluetooth devices. The developed protocol will execute the reverse role switching operation, the *split* operation, to recover the original structure of the scatternet, and then send a message to upper layer that contains the suggestion to reduce the packet size, to enable the packet retransmission overhead to be reduced. For example, the DH5 packet is replaced by many DH3 or DH1 packets.

By applying rules described above, the goal for bridge device to execute the role switching is described below.

Goal: Given a bridge device d, $d \in D$, $Role(d)=S/S$ or M/S, a proper role switching operation will satisfy the following requirement. Min($NOP(d)$) and Max($NOS(d)$) ■

When a bridge device intends to initiate a role switching request to the masters to which it is connected, it will establish a *Decision Table* to help bridge devices determine the order of the masters to which to apply role switching operation. The Decision Table is defined as follows.

Definition: *Decision Table for Role Switching DT(d)*

At the moment when a bridge tries to execute a role switching operation, a decision table $DT(d)$ is established by a bridge device d to collect information from all the connected masters about the number of slaves (noted by *NOS*) and the traffic. In *DT*, each row records information about one master to which the bridge is connected. *DT* includes two fields - the *NOIS* and the *Benefit* fields. The *NOIS* records the *Number Of Increased Slaves* if the role switching operation is applied to the master, whereas the *Benefit* field evaluates the benefit if the role switching operation is applied to the master. ■

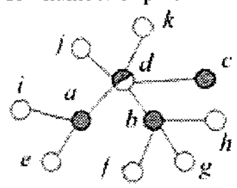
Assume that k routing paths $R_1, R_2, \ldots, R_k$ pass through a bridge device h. The *Traffic* T_i of a routing path R_i represents the number of packets that pass through a route within unit time. The benefit to device a of role switching initiated by device d is defined by $benefit(d, a)$ and equals, $\sum_{i=1}^{k} T_i \Delta_i$

, where Δ_i denotes the length of route R_i saved by executing the role switching operation. The bridge device will firstly check whether the benefit of role switching is positive. If the benefit to the master of initiating a role switching request is positive, then the bridge device will consider initiating role switching with the master.

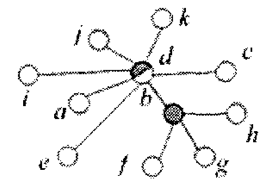
Table 1: Decision Table of device *d*

	NOIS	*Benefit*
c	1	*Be (d, c)*
a	3	*Be (d, a)*
b	5	*Be (d, b)*

Consider Fig. 4(a). When bridge device d detects that the condition $g_i(\gamma, o) > \delta_1$ satisfied and it evaluates the benefits of executing switching roles with masters a, b, and c are positive, device d will send a role switching request to masters a, b and c. On receiving a request, masters a, b and c reply with information on their *NOS* numbers and traffic loads to device d. Device d will store this information in its *decision table*, as shown in Table 1, in which $NOIS(c)=1$ represents that the number of slaves is increased by one if device d initiates a role switching operation $TakeOver(d, a)$ with device c. In such a case, if device d applies the role switching operations $TakeOver(d, a)$, $TakeOver(d, b)$ and $TakeOver(d, c)$ to masters a, b, and c, respectively, then the total number of slaves in the newly combined piconet will violate the condition $NOS(d) <= 7$. Then, device d will apply the role switching operation to these masters in ascending order of *NOIS* value recorded in the *DT*. Device d will initiate role switching operations with those masters until the total number of slaves reaches the upper limit to reduce the number of piconets as much as possible. Hereafter, device d will initiate role switching operations $TakeOver(d, a)$ and $TakeOver(d, b)$ with masters a and b. Figure 4 (b) shows the resulting structure of the scatternet. The number of piconets has been reduced from four to two. After device a initiates $TakeOver(a, b)$, the structure is shown in Fig. 4 (b). The number of piconet has been reduced to one.



(a) A scatternet structure for illustrating the TakeOver operation.



(b) Scatternet structure after executing TakeOver operation.

Figure 4: An example of executing *TakeOver* operation.

V. PERFORMANCE STUDY

This section examines the performance of the *Adaptive Role Switching Protocol*, in terms of the rate of successful packet transmissions, number of bridges, number of piconets, degree of bridges, and the average route length. The simulation environment is as follows. The size of the scatternet region is 7×7 units, and the radio transmission range of a Bluetooth device is set to a constant ten units, such that all devices are in the communication range. Initially, a connected scatternet, namely '*origin*', is randomly constructed. That is, each device alternatively executes an inquiry/page or an inquiry scan/page scan operation to connect with other devices at random. The proposed protocol applies role switching operations to obtain the structure of the original scatternet, generating another scatternet, namely '*ASRP*'. The number of devices ranges from ten to 100. The number of piconets is maintained between zero and 20. Parameters such as the number of devices, the number of piconets, the number of routing paths, and the average degree of the bridges are varied to elucidate the performance of *ARSP*.

Each Bluetooth device hops to another channel, according to master's hopping sequence. Increasing the number of piconets in a specific region will increase the likelihood that more than one device hops to the same channel simultaneously, causing collision and packet retransmission, which considerably reduce performance.

Figure 5 shows that the number of piconets increases with the number of devices. As the number of piconets increases, a device will link to more piconets, causing the number of bridges also to increase. Applying the proposed ARSP can reduce the number of piconets in a distributed manner, reducing the likelihood that a packet collision will be caused by the sharing of a channel to transmit packets. When piconet combining operation is applied, the ARSP merges two different piconets into one. Consequently, the master degree increases on average, as shown by Fig.6.

Packet transmission from one piconet to another relies on a bridge device. The use of the appropriate number of bridges can maintain the connections of the scatternet and guarantee that the packets from any Bluetooth device can be bridged to another device that belongs to another piconet. However, a bridge device that switches among piconets will waste of guard time and increase the problem of packet loss, because of the difficulty of scheduling among numerous piconets. Hence, reducing the number of bridges while maintaining a connected Scatternet is effective in reducing the routing delay and increasing the throughput of the Scatternet, significantly improving the performance. Figure 7 shows the difference between the numbers of bridge devices in a system after the execution of ARSP and the same system before the execution of ARSP. ARSP efficiently reduces the number of bridge devices, potentially improving the route delay and the rate of successful packet transmission from one piconet to another.

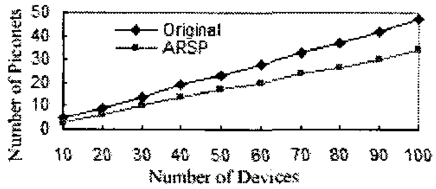


Figure 5: The number of piconets in varying number of devices.

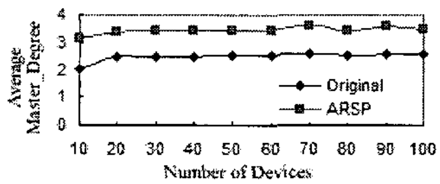


Figure 6: ARSP reduces the number of picoents thus increases the average master degree.

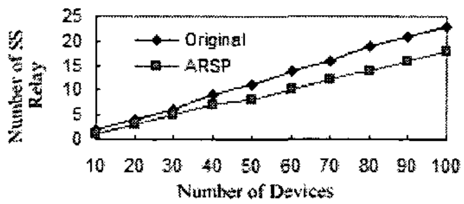


Figure 7: ARSP reduces the number of bridges thus saves the guard time and reduce the probability of packet lost.

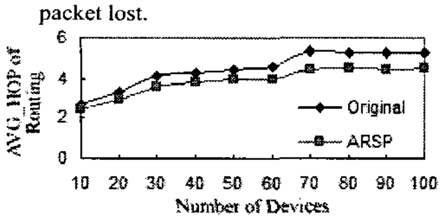


Figure 8: The effect of ARSP on the route length.

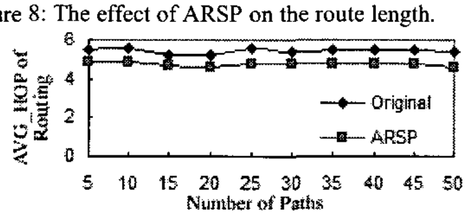


Figure 9: The comparison of ARSP and Original in average route length.

In a piconet, two slaves can exchange their information in a manner that enables the slave sender firstly to transmit information to the master and then enables the master to bridge the message to another slave. Two-hop routing is required. However, the sender and receiver may not belong to the same piconet. A route that connects sender and receiver and passes through many master and bridge devices should be constructed. The transmission delay of a route depends strongly on its length and the guard time associated with bridge switching the participated piconets. Ten routes with randomly generated sources and destinations are considered to investigate the effect of *ARSP* on route length. As shown in Fig. 8, the route length is significantly reduced as the number of piconets is increased. Figure 9 shows that the average route length is reduced constantly as the number of routes increases. Generally, applying ARSP locally changes the topology of the scatternet by reducing the number of piconets, the number of bridges and the route length. Scatternet performance is thus significantly improved.

VI. CONCLUSIONS

The improper structure of a scatternet will not only cause packet error and guard time overheads but also raise the problem of transmission delay. This work presents an adaptive role switching protocol (*ARSP*) for reorganizing the structure of an improper scatternet. Three types of role switching operations and their effects on scatternet performance are analyzed. The adaptive role switching protocol uses role switching operations to remove the unnecessary piconets and bridges and shorten the route in a distributed and dynamic manner. By monitoring the packet error rate, the guard time cost, and the traffic, the bridge device can send a role switching request to masters in the best order to maximize the benefit obtained by changing the structure of the scatternet. By maintaining appropriate numbers of piconets and bridges and the role of each device, the proposed protocol reduces the overheads increased by packet loss and guard time cost, as well as the route length. Experimental results indicate that the protocol reduces the packet loss rate, the guard time overhead and the route length, and so improves the performance of scatternets.

ACKNOWLEDGMENT

The author would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 93-2219-E-032-002.

REFERENCES

[1] Specification of the Bluetooth system-CORE v1.0A, July 26th 1999

[2] Ching Law, Amar K. Mehta, Kai-Yeung Siu, "A New Bluetooth Scatternet Formation Protocol", *ACM Mobile Networks and Applications Journal*, 2002

[3] Godfrey Tan, Allen Miu, John Guttag, Hari Balakrishnan, "Forming Scatternets from Bluetooth Personal Area etworks", *MIT Technical Report*, 2001

[4] Slim Souissi,Eric F,meihofer "Performance Evaluation of a Bluetooth Network in the Adjacent and Co-channel Interference", *in Motorola Personal Communications* , sector 5401 North Beach Street, Fort worth, TX76137USA.

[5] Zilrbes, S.; Stahl, W.; Matheus, K.; Haartsen, J "Radio network performance of Bluetooth", *Communications, 2000. IEEE International Conference*,Vol.: 3, pp. 1563 –1567,2000.

[6] Lakshmi Ramachandran, Manika Kapoor, Abhinanda Sarkar, Alok Aggarwal, "Clustering Algorithms for Wireless Ad Hoc Networks", *Proceedings of the 4th International Workshop on Discrete algorithms and methods for mobile computing and communications 2000*