A Fast Bluetooth Symmetric Link Formation Protocol – SF/CAI (Scan First/Collision Avoidance Inquire)

Han-Chiuan Luo[1], Eric Hsiao-Kuang Wu[2], Gen-Huey Chen[3] and Jui-Hao Chiang[4]

[1,3,4]Computer Science and Information Engineering Department, National Taiwan University, Taipei, Taiwan
[2]Computer Science and Information Engineering Department, National Central University, Chung-Li, Taiwan

E-mail: p88017@csie.ntu.edu.tw,hsiao@csie.ncu.edu.tw, ghchen@csie.ntu.edu.tw, littlejohn@inrg.csie.ntu.edu.tw

ABSTRACT

In the asymmetric point-to-point Bluetooth device connection establishment, the link formation time is dominated by the Random Backoff delay [2], which will take at most 640ms. The state of a node requires to be pre-assigned in the asymmetric scheme. This implies one node starts in INQUIRY (sender) and the other starts in INQUIRY SCAN (receiver). Unfortunately, the asymmetric feature is a pity for an ad hoc fashion connection setup without the need of any role pre-assigned explicitly. This paper proposes an ad hoc symmetric connection establishment protocol- SF/CAI. SF/CAI can achieve expected link formation delay less than 340msec, which approaches the expected time (330msec) of the asymmetric protocol defined in the Bluetooth specification. With the favor of SF/CAI, the speed of BTCP [3] would have a great improvement and allow a new comer to join the network. SF/CAI is much faster than the expected time cost (1sec) of the symmetric protocol proposed in [2]. Free from annoying human effort, SF/CAI provides an improved formation protocol for the invisibility thrust of pervasive computing.

1. INTRODUCTION

It has been a trend to integrate short-range Bluetooth communication into the existing wireless systems (such as 2G, 3G [14] and WLAN [12]) to provide ubiquitous computing service [7][9]. Ubiquitous computing for providing service any place, any time is now evolving as pervasive computing with the new demands. Pervasive computing agenda goes much further than mobile computing [6]. It incorporates four additional research thrusts into its agenda: effective use of smart spaces, invisibility, localized scalability, and masking uneven conditioning [6]. The invisibility ideal expressed by Weiser is complete disappearance of pervasive computing technology from a user's consciousness [6].

The initial application of Bluetooth technology is focused mainly on replacing cables between hand-held devices. However, a general wireless telecommunication such as public Internet access via a Bluetooth-equipped device is expected to be one of the most popular applications in the near future [13].

If someone brings a handheld or wearable Bluetooth device, could this device automatically discover other near-by bluetooth devices and establish connection efficiently without the extra effort or attention? Thus a fast and convenient way to establish a connection is strongly required and brings the several challenges: prompt connection, intelligent assisting point, multimedia traffic transmissions, and ad hoc communication connection set up.

The existing proposed Bluetooth asymmetric link formation protocols are only suitable for the case that the sender/receiver roles are pre-assigned. The required "asymmetric" formation is not a convenient way. Unfortunately, once subscribers are trying to establish links via their Bluetooth devices, they would object to doing explicitly sender/receiver role assignments frequently. The asymmetric way will be annoying enough by drawing subscribers' consciousness to find out the existence of neighboring Bluetooth devices even before role assignment effort. This is a pity for the purpose of prompt connection. Especially, novel applications would like to set up communications between their wearable/hand-held Bluetooth-equipped devices in an ad hoc fashion. Here, ad hoc network is described as the self-organizing and distributed controlled wireless network where every portable terminal has the same function as every other terminal [11][12].

For instance, in a typical conference scenario where individual members bring individual personal digital devices, they could form a wireless network by simply pushing a button. Thus a symmetric mechanism is required to construct connections in an ad hoc fashion without the need of any explicit role pre-assignment. This mechanism should be efficient enough to set up the conference scenario at the start up time, and meanwhile allow a new comer to join the network without bandwidth-waste and conference-disturbance to the formed network.

1.1 Bluetooth

Bluetooth technology has been drawing more and more attentions recently [7][8][9]. The expectations for Bluetooth are to top a billion units by 2005 [5]. The IEEE 802.15 Personal Area Network (PAN) Working Group has made Bluetooth the foundation for a range of consumer network products [5][10]. Bluetooth was developed initially as a short-range (10 meters) cable replacement for linking portable consumer electronic devices [5]. It provides a mechanism for forming small wireless networks of Bluetooth-equipped products on an ad hoc fashion [5].

Bluetooth applies pseudo random frequencies hopping sequence to combat interference [11]. The symbol rate is 1 Ms/s. A slotted channel is applied with a nominal slot length of 625 μ s. For full duplex transmission, a Time-Division Duplex (TDD) scheme is used. On the channel, information is exchanged through packets. Each packet is transmitted on a different hop frequency. A packet nominally covers a single slot, but can be extended to cover up to five slots.

The Bluetooth system provides a point-to-point connection (only two Bluetooth units involved), or a point-to-multipoint connection. In the point-to-multipoint connection, the channel is shared among several Bluetooth units. Two or more units sharing the same channel form a **piconet**. One Bluetooth unit acts as the master of the piconet, whereas the other unit(s) acts as slave(s). Up to seven slaves can be active in the piconet. Multiple piconets with overlapping coverage areas form a **scatternet**. Each piconet can only have a single master. However, slaves can participate in different piconets on a time-division multiplex basis. In addition, a master in one piconet can be a slave in another piconet. The piconets shall not be required to time or frequency-synchronized one another. Each piconet has its own hopping channel [1].

1.2 The Primitive Asymmetric Link Formation Protocol Defined in the Bluetooth Baseband Specification

According to the Bluetooth Baseband specification, the Bluetooth asymmetric protocol for link formation starts by the sender starting in the INQUIRY state and the receiver in the INQUIRY SCAN state. Given two units, one operating as a sender and the other as a receiver, the term **Frequency Synchronization delay** [2] (or **FS delay**) refers to the time until the sender transmits Inquiry Access Code (IAC) packets [1] at the frequency the receiver is currently listening on. When the receiver detects an IAC packet from the sender, we say a ***HIT*** occurring (The sender *HIT* the receiver and the receiver is *HIT* by the sender.) There is a FS delay until a hit occurs. After the first hit, the unit in Inquiry Scan (receiver) backs off for an amount of time that is uniformly distributed between 0ms (0 slot) and 639.375ms (1023 slots). The time when the receiver backs off is called the **Random Backoff delay** (or **RB delay**) [2]. When the receiver unit wakes up, it starts listening again at the hop it used to listen to before backing off. After a second FS delay, a second *HIT* occur. Then the receiver sends back to the sender its Frequency Hopping Sequence (FHS) packet [1]. Upon the sender receives the FHS, the paging procedure happens instantaneously and the connection is established. The paging procedure delay is negligible since it immediately follows the inquiry procedure with the time cost of 4 slots (1 slot = 625 μs). Thus the [2] approximates the link formation delay R by the following equation:

$$R = 2FS+RB \quad [2] \qquad (1)$$

Where FS and RB are uniform random variables in $[0,T_{coverage}]$ and $[0,r_{max}]$ respectively [2]. From (1), the link formation delay is dominated by $2T_{coverage}+r_{max}$=2*16+1023=1055slots (659.375ms) for the 32-hop system and 1039slots (649.375ms) for the 16-hop system [2]. And the expected link formation delay is approximately 528slots (330ms). This is the ideal case.

However, the extra human efforts of operations to negotiate sender/receiver role-playing between two Bluetooth devices should be considered. Thus, the actual time cost to form a link will be more than 330ms.

2. RELATED WORK

2.1 A Symmetric Protocol for Link Formation, *ALT600ms*

The protocol proposed in [2] is to force the two nodes to alternate independently between the INQUIRY and INQUIRY SCAN states and try to connect according to the asymmetric protocol during an "on" interval where they meet in opposite states.

Each unit will alternate between INQUIRY and INQUIRY SCAN with the mean state residence time of 600ms at each state following a uniform/exponential distribution. Here we name this protocol as *ALT600ms*.

The expected link formation delay is about 1 sec for the two nodes to connect. It is still much slower than the ideal case, 330ms.

2.2 BTCP Protocol for Constructing Network

The BTCP [5] is an ad hoc asynchronous distributed approach to establish Bluetooth scatternet connection. It collects information of neighbors and assigns roles of scatternet. This protocol uses a coordinator election process to collect all FHS packets of the neighbors in vicinity. Each node has a timeout (ALT_TIMEOUT = 2527.223ms) for the decision of whether the coordinator election is terminated or not. After the coordinator election, the coordinator will get all FHS information, and start to assign each node a role to construct Bluetooth scatternet.

The BTCP makes use of the symmetric link formation protocol *ALT600ms* proposed in [2] (2.1) to find the neighboring nodes. We don't think this is a good choice for the setup speed. The BTCP also has a drawback that no mechanism for a new comer to join after the establishment of the scatternet.

2.3 A Prompt Connecting Mechanism for Constructing Network

This mechanism [4] is asymmetric in link formation. There is a device dedicated in one particular area called the coordinator. The coordinator is pre-assigned in the INQUIRY SCAN state. All of the other devices start at the INAQUIRY state in order to discover the coordinator, while entering into this area. The coordinator will collect FHS information of all devices. If any device desires to connect to each other, it will ask the coordinator to get information to make a successful connection.

This approach allows the new comer to join the established scatternet and prevent the Inquiry bandwidth hog from the established scatternet by the new comer's inquire.

The drawback of this approach is that it must have a dedicated pre-assigned coordinator to discover information of all devices.

There exist other two disadvantages: the one is the co-channel frequency collision while multi devices enter the scatternet at the same time during a short interval doing inquiry; the other is there is one Random Backoff delay cost for one connection establishment. And the Random Backoff delays can't happen concurrently for the sake of the only one INQUIRY SCAN node (coordinator). This is the reason why the setup time is almost linearly to the number of Bluetooth devices. This leads much slower setup time at the beginning of the conference scenario than the BTCP, if too many joiners.

2.4 Our Motivation

The asymmetric method defined in specification [1] is the primitive function of a symmetric method. So the time efficiency of the asymmetric method is the lower bound for a symmetric method. Our motivation is to achieve a symmetric method close to the asymmetric method in efficiency. The research effort is focused on proposing a symmetric method preserving the benefit of Prompt Connecting Mechanism: let the new comer join and no bandwidth wasted in the established scatternet. And finally, the symmetric method will satisfy the invisibility requirement of pervasive computing [6].

3. SF/CAI SYMMETRIC LINK FORMATION PROTOCOL

In this section, the proposed symmetric link formation protocol, SF/CAI will be described. With the insight of Bluetooth system, the proposed protocol strategy is carefully determined. By the probability analysis, we select a proper parameter value for the proposed algorithm. Furthermore, we choose another parameter value by observation of simulation results to make a trade off between the expected average connection establishment time and the worst case average connection establishment time. With the proper choice of the related parameter values, SF/CAI exactly approaches the ideal case, the asymmetric link formation protocol defined in the Bluetooth baseband specification [1].

3.1 Ad Hoc Symmetric Link Formation Strategy Insight

Since the Bluetooth asymmetric link formation protocol yields a short connection establishment delay, the proposed new strategy to design symmetric protocol is focused on the goal of approximating the asymmetric protocol.

We let the first starting Bluetooth device enter the INQUIRY state, and let the latter comers enter the INQUIRY SCAN state. In the ad hoc environment, a Bluetooth device does not know whether it is the first starter. Thus a device assumes itself is the first starter when the *HIT* not occurs after a period of INQUIRY SCAN. But there is a chance that several devices may start at the same time, thus the INQUIRY SCAN period should be random. Our strategy is to let just one device enter INQUIRY mode as soon as possible and the others enter INQUIRY SCAN mode. Although the more devices enter the INQUIRY state, the higher probability the *HIT* happens for the devices at the INQUIRY SCAN state. However, the probability of co-channel frequency collision will increase at the same time Because of multiple IAC packets being sent. If a device is inquiring, the other device will be hit after 16-slot (slot time aligned) or 18-slot (slot time not aligned) continuing scan. And the dominant factor of link formation delay is the RB delay, not the FS delay. So it is not necessary to have a second device sending IAC at the INQUIRY state.

3.2 SF/CAI Algorithm

From the above reasons, the following strategies are considered to design a symmetric link formation protocol:

Scan first for at least 18 slots and a random time more in 18-slot unit, if be *HIT*, stay at the INQUIRY SCAN state; otherwise, enter the INQUIRY state. It alternates between two phases (SF, I/S) shown below:

SF:w*s	Inquiry (not *HIT*)/ Scan (*HIT*): I /I+w*S

Figure 1: the alternating timing

SF: scan first phase.
I/S: Inquiry/Scan phase.
w: Scan Backoff Window = 18 slots.
S: the maximum Scan Backoff Window number.
s : uniformly random from 1 to S.
I: the inquiry slot number for the I/S phase, if not *HIT*.
I+w*S: the scan slot number for the I/S phase, if *HIT*.

The following algorithm is the proposed ad hoc symmetric link formation protocol:

```
StartUp()
{
SFCAI();
TopologyConstruction:
    Form scatternet and communicate begin;
    Do periodically inquiry scan at idle slots after the
scatternet formed (to enable a new comer join);
}
```

```
SFCAI()
{
    Winner = true; //the winner of CoordinatorElection()
    Connect = false; //link formation succeed
    SFCAI_TimeOut_Enable = false;
        //the trigger of SFCAI_TimeOut
    SFCAI_TimeOut = 2527.223 ms;
    SF_Phase:
    HIT = false;
    TimeOut = w*SlotTime*uniform(1,S);
    //TimeOut is a decreasing timer;
    While (Not HIT) AND
        (SFCAI_TimeOut>0 AND TimeOut>0)
                InquiryScan();  //set HIT or not
    I/S_Phase:
    If (HIT) {  //be HIT during SF phase
        Random Backoff (0,1023) SlotTime;
        TimeOut = (w*S+I) * SlotTime;
        While(SFCAI_TimeOut>0 AND TimeOut>0)
        {           InquiryScan();
```

```
            If (HIT) {
                  Transmit it's own FHS;
                  PageScan();
                  //set Connect if succeed
                  If (Connect) {
                        TimeOut=0;
                        AfterConnect();}
            }
      }
  } Else {  //not be HIT after SF phase
      TimeOut = I*SlotTime;
      While (SFCAI_TimeOut>0 AND TimeOut>0)
      {     Inquire();  //to receive FHS
            If (a FHS received) Page(FHS);
            If (Connect) //Page(FHS) succeed
                  AfterConnect();
      }
  }
  If (SFCAI_TimeOut>0) Goto SF_Phase;
}
```

```
AfterConnect()
{
   CoordinatorElection(); //determine who wins
   If (Winner)
       SFCAI_TimeOut = 2527.223 ms;
   Else
   {
       Pass its collected FHS packets to the winner;
       SFCAI_TimeOut = 0;
       Enter Page Scan State;
   }
   SFCAI_TimeOut_Enable = true;
       //trigger SFCAI_TimeOut
}
```

Based on the above algorithm, a Bluetooth device will start to alternate between SF and I/S phases until it is connected. After a connection happens, by a coordinator election mechanism, it will enter the PAGE SCAN state if it loses the election, or continue to alternate between SF and I/S phases until no more connection succeeds after the SFCAI_TimeOut expires. When the SFCAI_TimeOut expires, the lived winner will coordinate the topology construction to form a scatternet. After the scatternet formed, all the joiners of the scatternet will make use of their idle slots to do INQUIRY SCAN, this won't waste the scatternet's bandwidth to do inquiry. Any latter comer who wants to join the scatternet will take the responsible of inquiry by alternating between SF and I/S phases until it engages into the scatternet.

3.3 Probability Analysis to Decide I Parameter

The alternating timing is depicted at Figure 1. When there are two Bluetooth nodes, we assume the s of the node N_0 is given, and the Inquiry/Scan phase start time I_1 of the node N_1 uniformly distributes within (w*s+I) slots. We also assume the two nodes are slot time aligned. If I_1 locates at the $[w*s+17, w*s+I]^{th}$ slot, the node N_0 will hit the node N_1, because the node N_1 will scan first at least 18 slots before I_1. Similarly, the node N_1 will hit the node N_0, if the I_1 locates at the $[1, w*s-15]^{th}$ slot. So, only when I_1 locates at the critical section between $[w*s-14, w*s+16]^{th}$ slot, the *HIT* may not occurs and the *HIT* probability was (1/16) for any slot of the critical section. Outside of the critical section, the *HIT* probability is 1.

From the above analysis, the not Hit probability is:

$$u(s) = (w-2)/(w*s+I) \quad (2)$$

The Hit probability is:

$$H(s) = 1-u(s) \quad (3)$$

The first *HIT* probability is:

$$H = (1/S)\sum H(s),\ s=1,2,3,\ldots S \quad (4)$$

Once a node was hit, it random backs off (0,1023) slots, and wakes up to stay at the INQUIRY SCAN state for (w*S+I) slots. The hitting node may be at the INQUIRY state at the probability of I/(w*S/2+I). So the second *HIT* probability is:

$$H_2=I/(w*S/2+I) \quad (5)$$

The expected connection-establish time is:

$$T=(528*SlotTime)/H/H_2 \quad (6)$$

From the equation (6), we draw Figure 2, 3.

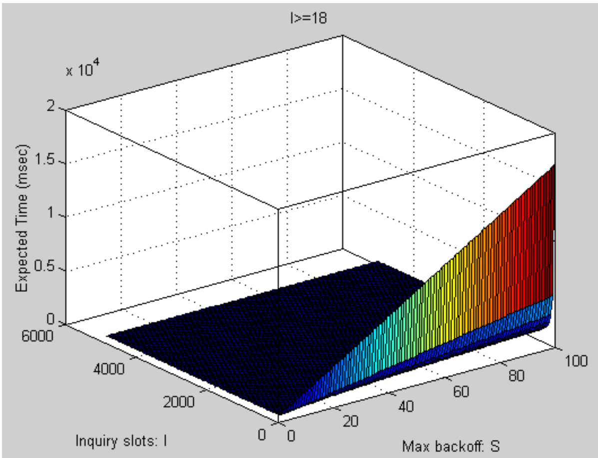

Figure 2: the expected connection-establish time for I>=18

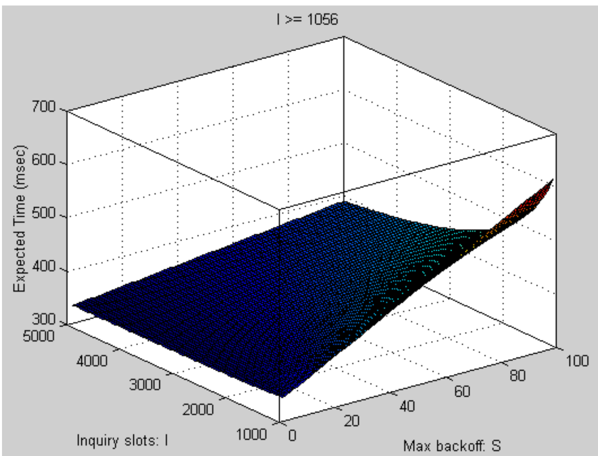

Figure 3: the expected connection-establish time for I>=1056

From Figure 2 and Figure 3, we choose I=16+1024+16 = 1056 slots. The S will be decided later by simulation.

3.4 Simulation to Decide S Parameter

Someone may think S=2 is a good choice from the view of Figure 3. But Figure 3 shows only ideal expected case for two Bluetooth devices to form a connection. The worst case may happen for 2 Bluetooth nodes start at the same time. If there are more than 2 nodes in the vicinity, the increase of node number will increase the sending of IAC packet, this is a profit to *HIT*. But with the increase of IAC packets sent at the same time, the IAC packets might be destroyed by each other cause of co-channel frequency collision. Thus there exists a tradeoff to select S value. We will do this decision by simulation results.

In our simulation, the interference not came from the Bluetooth system isn't taken into account. The only considered interference came from Bluetooth device is co-channel collision while radios work at the same frequency hop. We use 3 cases to simulate. Case 1 is **No offset**: The other nodes start at the same time as the first start node. Case 2 is **iid1000**: The other nodes will start within iid distributed time in 1000ms after the first starter. Case 3 is **iid2000**: The other nodes will start within iid distributed time in 2000ms after the first starter.

The following Figures are our simulation results. The X-axis is the Backoff S. In Figure 4/5/6, the Y-axis is the later starter's expected connection time after it starts. In Figure 7/8/9, there are 30 nodes within each other's radio range, and the Y-axis is the expected time to make the 30 nodes forming a fully connected topology after the first node starts.

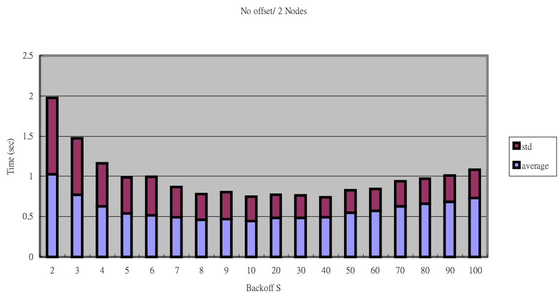

Figure 4: 2 Nodes/Case 1.

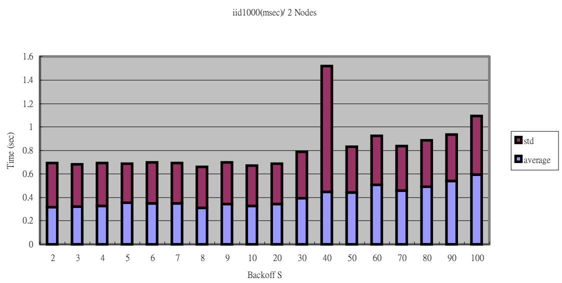

Figure 5: 2 Nodes/Case 2

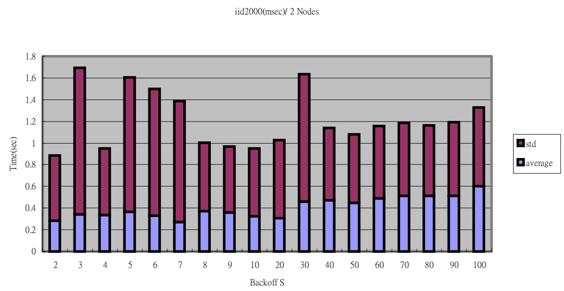

Figure 6: 2 Nodes/Case3.

We can note that the Figure 5/6 agree with our probability analysis shown in Figure 3. From the observation of Figure 4/5/6/7/8/9, we determine S=4 for our proposed protocol.

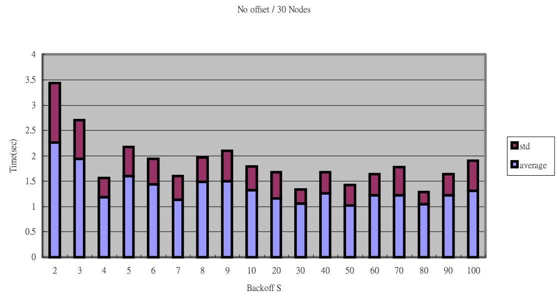

Figure 7: 30 Nodes/Case 1.

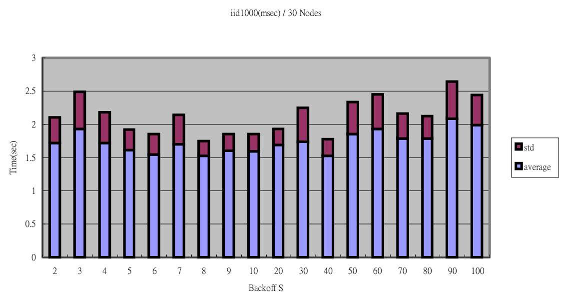

Figure 8: 30 Nodes/Case 2.

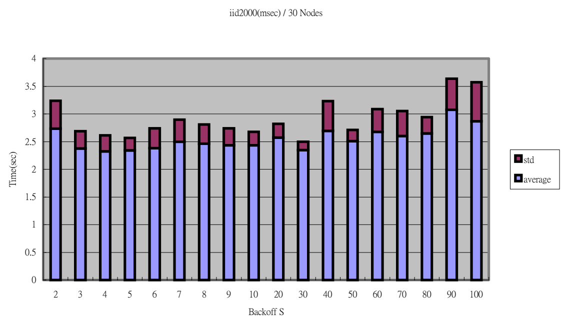

Figure 9: 30 Nodes/Case 3.

3.5 SF/CAI Approaches the Ideal Lower Bound

From Figure 5/6, when S=4 and I=1056, the expected link formation time is less than 0.34 sec, which approaches the ideal expected link formation delay 0.33 sec of the asymmetric link formation protocol defined by the Bluetooth specification.

In spite of the worst case happens (Figure 4), the expected average link formation delay is about 0.63 sec.

Our SF/CAI symmetric protocol is much better than the symmetric protocol *ALT600ms* proposed in [2], which needs 1 sec on average to form a link between 2 Bluetooth devices.

4. PROTOCOL PERFORMANCE

Now we introduce our SF/CAI (S=4, I=1056) protocol into the BTCP [3], using the same simulation conditions, we get the Figure 10. In the SFCAI() procedure, we use SFCAI_TimeOut = 2527.223 ms. This is borrowed from ALT_TIMEOUT = 2527.223ms of BTCP [3].

T_{ideal} is the time where the coordinator is actually elected. The actual network connection time T_{actual} will be:

$$T_{actual} = T_{ideal} + ALT\_TIMEOUT \;\; [3] \qquad (7)$$

The graphs "exp1000" and "exp2000" show the ideal network formation delay when each user is expected to "arrive" after the first user within 1s and 2s in the average according to the truncated exponential distribution.

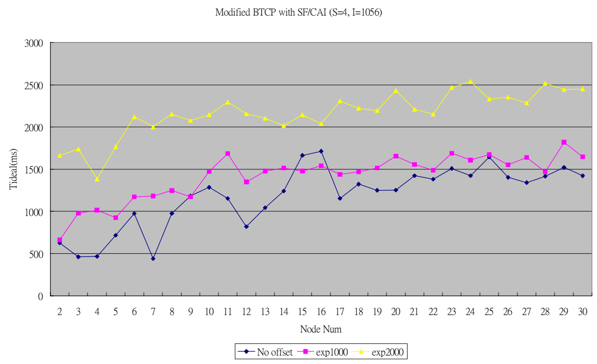

Figure 10: Average ideal connection establishment time for various application scenarios. Modified BTCP with SF/CAI (S=4, I=1056).

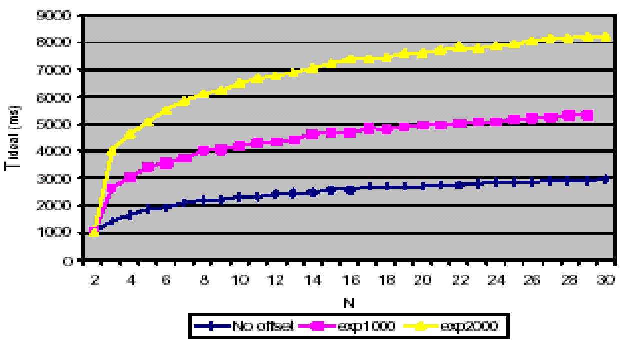

Figure 11: Average ideal connection establishment time for various application scenarios. Original BTCP with *ALT600ms* [3]

4.1 Compare to the Original BTCP

Compare to Figure 11 (the performance of the original BTCP protocol [3] with *ALT600ms*), our SF/CAI protocol makes a great performance improvement.

A Bluetooth device following the BTCP will stop its neighbor finding if no more one for lasting ALT_TIMEOUT. This is why the BTCP can't welcome the new comer's join.

But following SF/CAI protocol, a device will be continuing to try to find a neighbor till it successes. Then the SFCAI_TimeOut will be triggered, i.e., if no more neighbors found in 2527.223 ms, the device will enter the CONNECTION state to communicate and do periodically INQUIRY SCAN at the idle slots when no communication. Thus our SF/CAI protocol can let a new comer to join the network.

4.2 Compare to the Prompt Connection

The inquiry role is also played at the side of a new comer, thus there is no bandwidth hog of the preformed network. The SF/CAI does not need a pre-assigned dedicated coordinator like the Prompt Connection mechanism [4]. Any node already in the network will do INQUIRY SCAN periodically. This makes the connections happen concurrently, if there are many new comers entering at the same time. However, the Prompt Connection has the only coordinator to do periodically INQUIRY SCAN after the network setup. Therefore, the Prompt Connection establishes one connection after one connection. For each link establishment, the Prompt Connection needs another 330ms about. Thus when the new arrivals during 330ms are no more than one, the Prompt Connection is fine. But, if there are more than one arrival during a 330ms interval, it is not time efficient.

The Prompt Connection is asymmetric that means human effort to select a proper role for each device. This is another time cost should be taken into account.

The SF/CAI also prevents many nodes from entering INQUIRY states to get rid of co-channel frequency collision. This point doesn't be thought of in the Prompt Connection [4] and the BTCP [3]. Too many IAC radios of new comers will jam the communication of the functioning network.

The Prompt Connection needs a dedicated device to play the coordinator role. This is an additional money cost.

The SF/CAI is superior much to the Prompt Connection mechanism in setup time efficiency, bandwidth usage, and money saving.

4.3 Compare to the Asymmetric Standard Protocol

Figure 12 shows the ideal performance of the primitive asymmetric link formation protocol defined in the Bluetooth baseband specification. The first starter operates as a sender and the others as receivers.

Compare Figure 10 and Figure 12, the performance of our SF/CAI is near to the asymmetric standard protocol. But, never forget the human effort to negotiate sender/receiver role is a great time cost. Thus the actual network connection time T_{actual} of the asymmetric standard protocol will be:

$$T_{actual} = T_{ideal} + ALT\_TIMEOUT + \text{Human\_Effort} \quad (8)$$

Therefore, we can claim our SF/CAI is faster than the standard protocol.

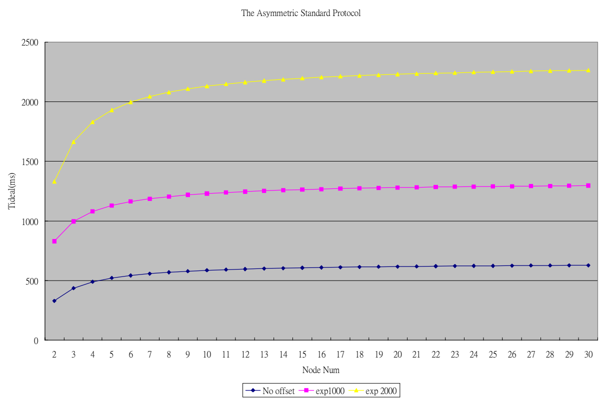

Figure 12: Average ideal connection establishment time for various application scenarios. The asymmetric standard protocol.

Besides, our SF/CAI will work smoothly without drawing any user's consciousness except to push a start button. For an example, a person wearing a Bluetooth watch walks around an exhibition hall in order to exchange digital name cards with someone passing across. They don't need to stop on going walking to figure out each other's device and negotiate each other's sender/receiver role for link establishment. For another example, when a latter comer wants to join the Bluetooth conference network that have formed, with our SF/CAI protocol, he makes no disturbance; with the asymmetric standard protocol, he should acquire a pause of the conference and negotiate sender/receiver role playing to reconstruct the network.

5. CONTRIBUTION AND CONCLUSIONS

The proposed novel symmetric protocol offers a connection time close to the speed of the ideal case (the asymmetric protocol defined in the Bluetooth specification). When we choose I=1056 and S=4, the SF/CAI protocol achieves expected connection establish time less than 340ms. From Figure 5/6, we can say SF/CAI approximates the speed of the Bluetooth asymmetric link formation protocol, which has expected connection establish time 330ms, the lower bound for a symmetric method making use of the primitive asymmetric function. Thus SF/CAI is a fast symmetric link formation protocol for Bluetooth system.

And SF/CAI is totally distributed getting rid of role pre-assignment drawback. Taking the human efforts into account, SF/CAI is the fastest mechanism to establish connection and convenient enough.

In addition, SF/CAI allows a new comer to join the ad hoc network after the scatternet has formed without the disturbance to the network joiners and the drawback of inquiry bandwidth hog. SF/CAI avoids co-channel collision of many simultaneous IAC sending radios. Therefore, the proposed SF/CAI is convenient to subscribers and bandwidth efficient to the scatternet.

Up to now, SF/CAI is the best choice for ad hoc fashion Bluetooth device connection establishment by making use of the primitive mechanism defined in the Bluetooth Baseband asymmetric link formation protocol. SF/CAI works very well in terms of time efficiency, bandwidth efficiency, and also invisibility to users, that is a very thrust of pervasive computing.

ACKNOWLEDGMENT

This research was supported in part by the Development of Communication Software Core Technology project of Institute for Information Industry and sponsored by MOEA, R.O.C.

REFERENCES

[1] "Specification of the Bluetooth system-core Version 1.1", February 22 2001.

[2] Theodoros Salonidis, Pravin Bhagwat and Leandros Tassiulas, "Proximity Awareness and Fast Connection establishment in Bluetooth", Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on 2000, Page(s): 141 –142.

[3] Theodoros Salonidis, Pravin Bhagwat, Leandros Tassiulas, and Richard LaMaire,"Distributed Topology Construction of Bluetooth Personal Area Networks", INFOCOM 2001. Proceedings. IEEE, Volume: 3, 2001, Page(s): 1577 –1586.

[4] Yao-Hsin Chou, Hsiao-Kuang Wu, "Prompt Connecting Mechanism and Multiple-code Transmission Support for Bluetooth Personal Area Network", Master thesis (2001), Department of Computer Science and Information Engineering, National Central University, Taiwan, ROC.

[5] Ron Schneiderman, "Bluetooth's slow dawn", IEEE Spectrum, Volume: 37 Issue: 11, Nov. 2000, Page(s): 61 –65.

[6] M. Satyanaraynan, "Pervasive Computing: Vision and Challenges", IEEE Personal Communications, Volume: 8 Issue: 4, Aug. 2001, Page(s): 10 –17.

[7] Graham Kirby, Mobile Computing Group, Intel Corporation, "Integrating Bluetooth Techonology into Mobile Products", Intel Technology Journal Q2, 2000, Page(s): 1-8.

[8] Urban Bilstrup and Per-Arne Wiberg, "Bluetooth in Industrial Environment", WFCS-2000, September 6-8, Porto, Portugal, Page(s): 239-246.

[9] Nusser, R.; Pelz, R.M. "Bluetooth-based Wireless Connectivity in an Automotive Environment", Vehicular Technology Conference, 2000. IEEE VTS Fall VTC 2000. 52nd, Volume: 4, 2000 Page(s): 1935 –1942.

[10] Thomas M. Siep, Ian C. Gifford, Richard C. Braley and Robert F. Heile, "Paving the Way for Personal Area Network Standards: An Overview of the IEEE P802.15 Working Group for Wireless Personal Area Networks", IEEE Personal Communications February 2000, Page(s): 37-43.

[11] Jaap C. Haartsen, "The Bluetooth Radio System", IEEE Personal Communications, February 2000, Page(s): 28-36.

[12] D. Cox, "Wireless Personal Communications: What is it? ", IEEE Personal Communication Magazine, April 1995, Page(s): 20-35.

[13] Yujin Lim, Jesung Kim, Sang Lyul Min, and Joong Soo Ma, "Performance evaluation of the bluetooth-based public internet access point", Information Networking, 2001. Proceedings. 15th International Conference on, 2001 Page(s): 643 –648.

[14] Qi Bi, George I.George I. Zysman, and Hank Menkes, "Wireless Mobile Communications at the Start of the 21 Century", IEEE Communication Magazine, January 2001, Page(s): 110-116.