Rapid Heterogeneous Connection Establishment: Accelerating Bluetooth Inquiry Using IrDA

Ryan Woodings, Derek Joos, Trevor Clifton, Charles D. Knutson
Department of Computer Science
Brigham Young University
Provo, Utah 84602
{woodings, joos, cliftont, knutson}@cs.byu.edu

Abstract

Bluetooth device discovery is a time-intensive phase of the Bluetooth connection-establishment procedure. In this paper we propose a technique that integrates existing IrDA technology with Bluetooth technology to improve the connection-establishment time of Bluetooth devices. We accomplish this improvement by first establishing an IrDA connection between two devices equipped with both Bluetooth and IrDA capabilities and then exchanging Bluetooth device discovery information via the established IrDA connection. As a result of this cooperative exchange, the devices are able to bypass the time-intensive Bluetooth device discovery procedure. Our research shows that IrDA-assisted Bluetooth connection establishment is up to four times faster than the normal Bluetooth connection establishment procedure. In addition, it provides other time-savings in subsequent device selection procedures.

Keywords
Wireless integration, IrDA, Bluetooth, device discovery, mobile devices

1. Introduction

Short-range wireless connectivity technology from the Infrared Data Association (IrDA) has been integrated into many mobile devices. "IrDA technology is already in over 100 million electronic devices including desktop, notebook, palm PCs, printers, digital cameras, public phones/kiosks, cellular phones, pagers, PDA's, electronic books, electronic wallets, and other mobile devices" [1]. IrDA offers great strengths for dynamic, ad hoc connectivity, but is limited in certain usage models by its requirements of short distance and line-of-sight between devices.

As a recent newcomer to the marketplace, Bluetooth is showing strong signs of initial acceptance. The Bluetooth specification has garnered widespread corporate support with more than 2,000 companies registered in the Bluetooth Special Interest Group. Bluetooth wireless devices are projected to exceed one billion units by the year 2005 [2]. Bluetooth offers flexibility in distance and direction, but is more cumbersome than IrDA in the establishment of dynamic, ad hoc connections between mobile devices.

When Bluetooth was announced, many saw it as a direct competitor to IrDA since both provide short-range point-to-point connectivity between devices. In truth, Bluetooth surpasses IrDA in some regards, while IrDA is stronger than Bluetooth in others [3]. Some suggest that these two technologies will, in fact, co-exist in many mobile devices. Rather than pursuing an either/or scenario, we propose a combination of these two short-range wireless technologies to capture their respective strengths while minimizing their weaknesses. The full potential of Bluetooth and IrDA will not be realized simply by coexistence, but by integration and cooperation of both technologies.

This paper details research into the ways in which IrDA and Bluetooth can be combined to improve Bluetooth's connection establishment time. We first describe how Bluetooth performs device discovery and connection establishment. Next, we outline how IrDA performs these same procedures. Finally, having shown the relative performance of Bluetooth and IrDA device discovery and connection procedures, we show how to use IrDA to retrieve the information required for Bluetooth connection establishment. This enhancement eliminates the need for Bluetooth device discovery and results in a four-fold improvement in connection establishment time. In addition, it dramatically simplifies the Bluetooth device selection procedure and reduces the amount of user intervention required for a successful transaction.

2. Bluetooth Overview

Bluetooth is a short-range Radio Frequency (RF) technology in the 2.4 GHz range, capable of point-to-multipoint connections at speeds up to 1 Mbps. It uses frequency hopping to minimize the effects of signal interference caused by IEEE 802.11, HomeRF, microwave ovens, other Bluetooth devices, and miscellaneous other devices operating in the 2.4 GHz ISM band. Bluetooth signals do not require line-of-sight, can travel through most physical barriers, and have a range of 10 meters.

The following section describes relevant details of Bluetooth device discovery and connection establishment. This discussion helps explain why this procedure is so time-intensive. Finally, the time-consuming aspects of Bluetooth connection establishment are characterized and quantified for later reference.

2.1. Bluetooth Device Discovery and Connection Procedure

The device discovery and connection establishment procedure begins when a Bluetooth device enters the inquiry substate to discover other Bluetooth devices. The Bluetooth specification defines inquiry access codes that allow a device to specify the type of device it is seeking, such as PDAs, printers, or LAN access points. During inquiry, devices generate an inquiry hopping (channel changing) sequence. This inquiry hopping sequence is derived from the local device's clock and the chosen inquiry access code. This hopping sequence covers a 32-channel subset of the available 79 Bluetooth channels. Once a device generates an inquiry hopping sequence, it broadcasts inquiry messages as it sequentially switches to each channel defined in the hopping sequence.

Discoverable devices will periodically enter the inquiry scan substate. In this substate, devices hop according to the inquiry scan hopping sequence, which is also based on the inquiry access code and the local clock. If the device performing the inquiry scan receives an inquiry message, it enters the inquiry response substate and replies with an inquiry response message. The inquiry response includes the remote device's address and clock, both of which are needed to establish a Bluetooth connection [4].

All discoverable devices within the 10-meter broadcast range will respond to the device inquiry. This typically requires the user to manually select the desired Bluetooth device from a list of discovered devices. After spending time inquiring all devices in range, an indeterminate amount of time must now be spent by the user in order to select the desired device.

After obtaining and selecting a remote device's Bluetooth address, the local device enters the paging substate to establish a connection with the remote device. In the paging substate, the local device generates a hopping sequence based on the remote device's address and estimated current clock. The paging device then repeatedly sends page messages as it hops through the generated sequence of channels. If a device allows other devices to connect to it, it will periodically enter the page scan substate. In the page scan substate, a hopping sequence is generated based on the local address and clock.

When the remote (slave) device receives a page packet, it responds to the local (master) device with a page response packet. Upon receiving the response, the master sends a Frequency Hopping Synchronization (FHS) packet to the slave. The FHS packet includes the master's Bluetooth address and clock. Once the slave receives the FHS packet, it sends an acknowledgement to the master. When the master device receives the acknowledgement, it generates a new hopping sequence from its own address and its own clock. The slave then uses the master's address and the master's clock to generate a hopping sequence identical to the master's hopping sequence. The identical hopping sequences allow the devices to hop on common channels while remaining connected.

Once the paging process is complete, the devices move to the connection state. The master sends a poll packet to the slave verifying that the transition from the page hopping sequence to the new hopping sequence is successful. If successful, the two devices continue frequency hopping in a pseudo-random pattern based on the master device's address and clock for the duration of the connection [4].

2.2. Reasons for slow Bluetooth Device Discovery and Connection

According to the Bluetooth specification, "the inquiry substate may have to last for 10.24 seconds unless the inquirer collects enough responses and determines to abort the inquiry substate earlier" [4]. In an error-prone environment, it is difficult to determine the maximum time required for device discovery. Spending 10.24 seconds (or longer) just to discover devices that are in range is unacceptable in many situations. As an example, if a consumer spends 10 or more seconds waiting for a mobile device as it discovers all the cash registers at the front of a store, and then is required to manually select the correct cash register in order to pay electronically, it may seem like an eternity to the consumer, the clerk, and the people waiting behind him in line. This lengthy discovery time also becomes critical when devices are actively moving during discovery. For example, if one device is actively moving past a second device, the time required to perform discovery may in fact exceed the time during which the two devices are in range of one another, effectively rendering the devices unable to communicate in a meaningful fashion. This makes Bluetooth an unsatisfactory solution in certain situations [5].

The inquiry substate contains two 16-channel subsets known as trains. Each train takes 10 ms to complete. By specification, each train must be repeated 256 times to allow sufficient time to collect all inquiry responses. The specification also dictates that at least three train switches must occur, meaning that there must be two iterations of each train. Running both trains twice, at 256 times per iteration, allows the inquiry device to ensure that all listening devices in range will be on a common frequency and be in the inquiry scan substate during at least one inquiry time slot. The resulting total is 10.24 seconds, as shown in Equation 1. It should be noted, however, that in a noisy or error-prone environment, there is no guarantee of successful inquiry even if both devices are on the same frequency at the same time, since packets transmitted at that time may be corrupted. In such situations, the inquiry time may far exceed the default time of 10.24 seconds.

$$\boxed{2\ trains \times 2\ iterations \times 256\ times \times 0.01\ s = 10.24 s}$$

Equation 1. Minimum Time Required for Bluetooth Device Discovery.

3. IrDA Overview

IrDA is a short-range infrared wireless technology, capable of point-to-point connections at speeds up to 16 Mbps. IrDA signals require line-of-sight, with a range of up to one meter. IrDA provides "low-cost, short-range, cross-platform, point-to-point communications at a wide range of speeds" [6].

The following section describes IrDA device discovery and connection procedures. We then explain why the simplicity of IrDA device discovery and connection establishment makes it quicker than Bluetooth device discovery and connection establishment.

3.1. IrDA Device Discovery and Connection Procedures

IrDA device discovery uses a polling scheme to collect responses from all devices in line-of-sight within a one meter range. The device performing discovery is called the primary device and the devices that respond are called secondary devices. The primary device broadcasts a message to initiate device discovery. This message identifies the number of discovery time slots in which the secondary device may respond. Device discoveries may contain 1, 6, 8, or 16 time slots. Each secondary device generates a random number specifying the slot in which it will respond. The primary device sends out a device discovery packet at the beginning of each time slot. If the time slot number matches the random number chosen by the secondary device, it will send a discovery response packet to the primary device. Each time slot must last at least 25 milliseconds, with each response beginning within 10 milliseconds and completing within 70 milliseconds of the end of the primary's device discovery packet. If a device discovery is performed using the maximum number of slots (16 slots), the device discovery time will be 1.12 seconds as shown in Equation 2.

$$16 \ timeslots \times 0.070 \ s = 1.12 \ s$$

Equation 2. Time Required for IrDA Device Discovery.

Since IrDA devices communicate via a relatively short (one meter) and narrow (15 degree half-angle) infrared cone, the number of potential secondary devices visible to a given primary during discovery is quite small (typically from one to three). As a result, most devices do not use the maximum 16 discovery slots. Rather, our review shows that IrDA-enabled devices almost exclusively use either 6 or 8 slots for discovery. As result, a typical IrDA device discovery procedure consumes approximately half the time indicated in Equation 2, or from 0.42 to 0.56 seconds.

After completing the IrDA device discovery, the primary device can establish a connection by sending a connection request packet to the desired secondary device. The connection request packet contains the supported values of the connection parameters for the primary device (baud rate, window size, etc). If the secondary device accepts the connection request it will respond with a connection request response packet, which contains its supported values for the connection parameters. After switching to the most suitable common parameter values, the devices are connected [7] [8].

The line-of-sight requirement and limited signal range of IrDA substantially reduce the possible number of devices in range, typically reducing (or entirely eliminating) the need to perform a complex device selection process after the device discovery is completed. Also, IrDA device discovery does not suffer from the complications caused by frequency hopping in Bluetooth, allowing IrDA to perform a complete device discovery ten to twenty times faster than a Bluetooth device discovery.

4. IrDA-Assisted Bluetooth Device Discovery

The preceding descriptions of IrDA and Bluetooth reveal their weaknesses and underscore the need for improvement. Our study showed that in certain scenarios, a combination of Bluetooth and IrDA can achieve connection time improvements of up to 400%. These improvements can be realized when mobile devices are equipped with both IrDA and Bluetooth capabilities.

The current Bluetooth connection establishment procedure consists of the two steps, or substates, mentioned in Section 2.1. These are the inquiry substate and the page substate. As shown in Figure 1, the inquiry substate of the Bluetooth connection takes 10.24 seconds to complete before entering the paging substate. In our experiments the paging substate took an average of 1.804 seconds, after which two Bluetooth devices are fully connected. Our hybrid approach to device discovery aims to improve the time of the inquiry substate in the Bluetooth connection process.

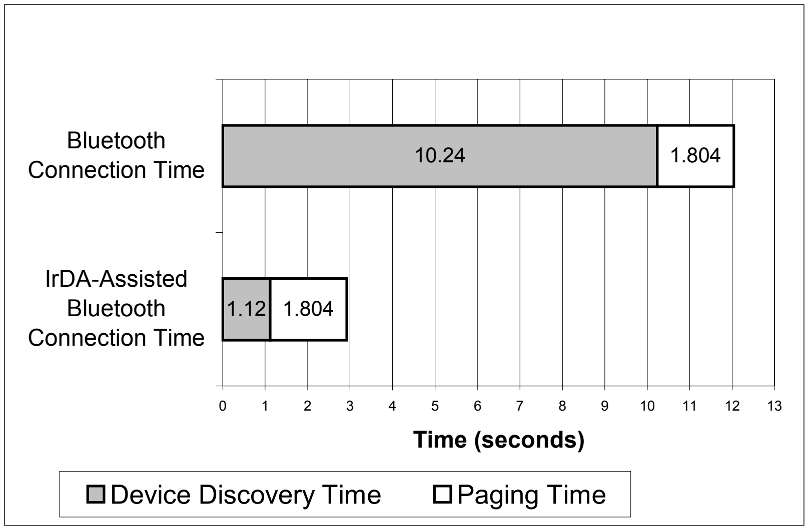

Figure 1. Connection Time Analysis

By utilizing the quick connection-establishment capabilities of IrDA, the device gathers the information necessary to perform a Bluetooth connection in significantly less time than the 10.24 seconds required by the Bluetooth device discovery procedure. Establishing an IrDA connection to the remote device also eliminates the need for the Bluetooth device selection procedure which occurs after discovery. The Bluetooth device selection procedure may require user involvement in selecting from a list of discovered devices. This factor not only increases the complexity of software (particularly the user interface) but also involves an additional (largely indeterminate) amount of time for the user to make a decision and communicate that choice to the device.

We first look at a few scenarios that benefit from IrDA-assisted Bluetooth connection establishment. This is followed by a technical overview of the IrDA-assisted Bluetooth connection establishment procedure. Finally, we explain how we implemented this procedure in a sample application.

4.1. Usage Scenarios

There are a number of scenarios in which accelerating Bluetooth inquiry using IrDA is advantageous. One common scenario for using Bluetooth is transferring a file between two handheld devices. Before the file transfer, a Bluetooth inquiry must first occur, requiring up to 10.24 seconds. After the inquiry is complete, the user must choose the correct device from among those that respond to the inquiry. The user is required to know either the exact Bluetooth address or the user-friendly name of the remote device. After the user chooses the correct device, a Bluetooth connection is established and the file is transferred using the connection enhancement. This scenario can be significantly improved if users can simply point the devices at each other for one or two seconds (long enough to establish the IrDA connection and retrieve the Bluetooth connection information). After the Bluetooth connection information is retrieved, the IrDA link can be disconnected and the Bluetooth connection established. The users are then free to move their devices around, out of line-of-sight, while the file continues to be transferred over Bluetooth.

Another scenario that benefits from IrDA-assisted Bluetooth connection establishment is an Internet connection using a PDA or laptop via a cell phone. The user can momentarily point the device at the cell phone and use IrDA to complete the device discovery, thus bypassing the lengthy device discovery selection procedure of the Bluetooth connection process. The user can then pocket the cell phone and surf the Internet on the PDA or laptop.

As a final example, mCommerce promises to revolutionize the way we pay for groceries. Imagine that Grandma is standing at the checkout counter of a local grocery store that has just added Bluetooth point-of-sale (POS) terminals. Grandma takes out her Bluetooth-equipped Palm device and begins to do a Bluetooth inquiry. The Bluetooth inquiry discovers the POS terminals at all the checkout stands in the store, plus the PDAs and cell phones of the other customers standing in several lines. Grandma has no idea which device she needs to select, so the clerk must help her select the appropriate POS terminal. The other customers standing behind her grow extremely impatient after waiting 20-30 seconds (or longer) for her to complete her transaction.

If Grandma had an IrDA-enabled PDA and the POS terminals were IrDA-equipped, she might possibly complete the transaction using only IrDA, eliminating the need to select the appropriate POS terminal. But since IrDA is line-of-sight, Grandma would need to keep her Palm pointed at the POS terminal throughout the transaction. Unfortunately, Grandma’s hands aren’t as steady as they used to be, and she can’t necessarily hold her palm device steady long enough to complete the IrDA transaction.

Now, if both devices used the IrDA-assisted Bluetooth connection method, Grandma would be able to hold her device in position for the 1 to 1.5 seconds required for her PDA to establish an IrDA connection to the POS terminal and retrieve the Bluetooth connection information. Her PDA could then establish a Bluetooth connection to the POS terminal and complete the transaction without concern for line-of-sight. In this situation, there is no need for Grandma to select the POS terminal from a huge list of Bluetooth devices, nor for her to hold her PDA steady during the entire transaction.

4.2. Technical Overview

In order to take advantage of our proposed algorithm, each device must be equipped with a Bluetooth stack that includes L2CAP, HCI and hardware layers. Each device must also have an IrDA stack consisting minimally of IAS, IrLMP, IrLAP, and hardware layers. The Bluetooth and IrDA stack architectures necessary for IrDA-assisted Bluetooth Device Discovery are shown in Figure 2 below.

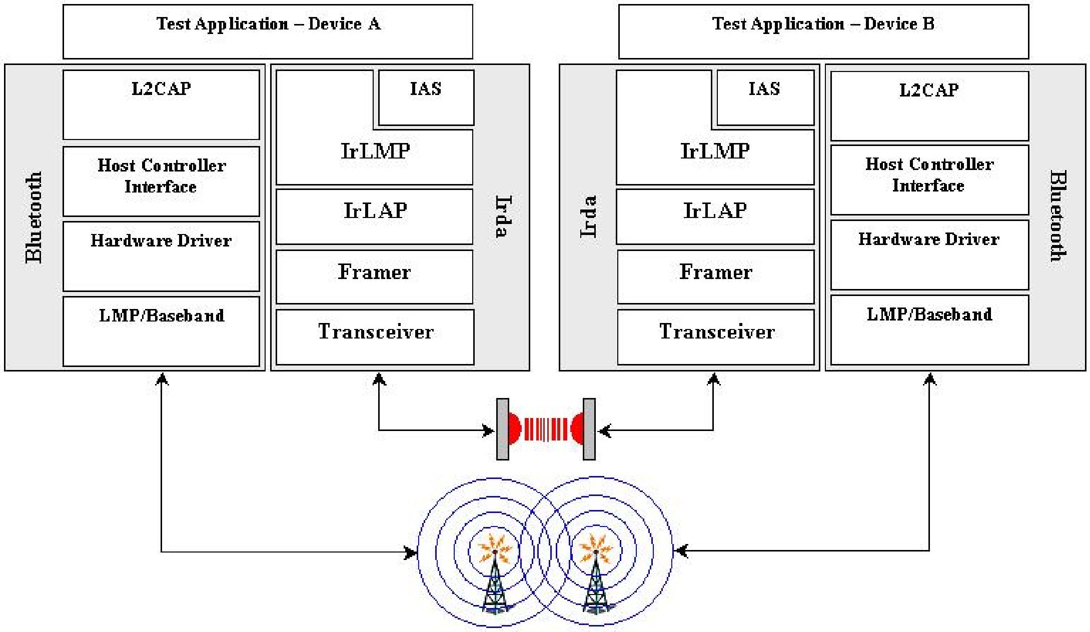

Figure 2. Bluetooth and IrDA Stack Diagram

IrDA devices are equipped with a "yellow pages" of services called the Information Access Service (IAS). The IAS contains a listing of the services provided by the IrDA device and provides mechanisms to query the IAS on a remote device. This information tells remote devices how to connect to the services provided by the IrDA device.

Each IAS entry consists of a class name and a set of attributes. Each attribute has a name and a value. An example IAS entry is the entry used for the IrCOMM protocol layer. The class name for this entry is "IrDA:IrCOMM," the attribute name is "IrDA:IrLMP:LsapSel," and the attribute value is the value of the LsapSel service (similar to a port number in the wired world). By defining an attribute class called "Bluetooth" and an attribute name of "Address," the Bluetooth address can be stored in IrDA's IAS database. If a local device queries a remote device's IAS database for "Bluetooth:Address," the Bluetooth address of the remote device is returned. Table 1 shows examples of IAS entries [8].

Class Name	Attribute Name	Attribute Value
IrDA:IrCOMM	IrDA:IrLMP:LsapSel	3
	Parameters	5B:34:26
	IrDA:InstanceName	Device Instance Name
Bluetooth	Address	55:26:5E:36:28:A3
	Name	John Doe's PDA

Table 1. Example IAS Entries

Once the Bluetooth address is retrieved from a remote device's IAS database, the local device can skip the Bluetooth inquiry state and go directly to the paging state, thus reducing the amount of time required to establish the Bluetooth connection.

Pseudocode for the IrDA-assisted Bluetooth connection algorithm is shown in Figure 3. The Local Device section describes the behavior of the client device as it initiates inquiry. The Remote Device section describes the behavior of the server device as it is discovered. Note that the remote device must be in a state of responding to IrDA discovery requests in order to assist in the Bluetooth connection procedure.

```
Let l be the local device
Let r be the remote device
Let BA_l be the Bluetooth address of the local device
Let BA_r be the Bluetooth address of the remote device
Let QR_r be the IAS Query Response from the remote device
Let DD be a list of discovered devices, where DD_n is device n in the list
Let IAS_ba be the IAS entry for the Bluetooth Address

Local Device
Begin
    DD = IrDiscover();
    if(size(DD) == 0)
        return;
    IrConnect(DD_0);
    QR_r = IASQuery(DD_0, “Bluetooth:Address”);
    if(QR_r == null)
        return;
    BA_r = IRIAS_GetUserString(QR_r);
    BTConnect(BA_r);
End.

Remote Device
Begin
    BA_l = RetrieveLocalBluetoothAddress();
    IAS_ba = CreateIASEntry(BA_l);
    while(true) {
        switch(Event) {
            IR_DISCOVER:
            IrDiscoveryResponse();

            IR_CONNECT:
            IrConnectResponse();

            IR_IASQUERY:
            IrIasResponse(IAS_ba);

            BT_CONNECT:
            BtConnectResponse();
        }
    }
End.
```

Figure 3. Pseudocode for IrDA-Assisted Bluetooth Connection Procedure

4.3. Normal Bluetooth Connection Algorithm

In order to measure the performance improvement of IrDA-assisted Bluetooth Device Discovery, we implemented a test application, allowing us to collect empirical data. The following paragraph describes the algorithm used in the test application.

For the normal Bluetooth connection procedure, the application starts a timer and then begins an inquiry of all devices in range. After the inquiry is complete, it establishes a Bluetooth connection to the first device that responded to the inquiry. The device selection procedure is omitted from these empirical results in order to eliminate variation in results caused by user interaction. This device selection step adds a minimum of several seconds to the normal Bluetooth connection establishment procedure, and potentially a great deal more, depending on number of Bluetooth devices in range, user behavior and ease of interface. As soon as the Bluetooth connection is established, the application stops the timer.

Figure 4 below is a screen shot taken of this test application as it establishes a Bluetooth connection using the normal connection establishment procedure. Observe the numerous Bluetooth devices that responded to the inquiry. In a typical application the user would have to choose the desired device from a list of Bluetooth addresses or user-friendly names corresponding to the discovered devices.

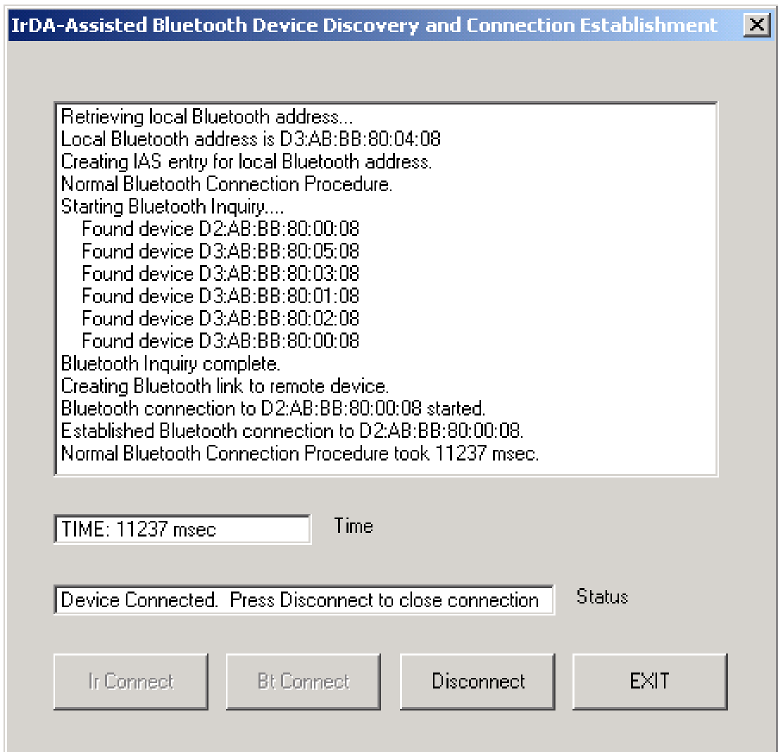

Figure 4. Screenshot of Normal Bluetooth Procedure.

4.4 IrDA-Assisted Connection Algorithm

In the IrDA-assisted device discovery method, the test application retrieves the Bluetooth address from the radio and stores it in an IAS entry for “Bluetooth:Address”. The application then starts the timer, and performs a remote IrDA device discovery. When the IrDA device discovery is completed, the application establishes an IrDA connection with the remote device. In our experiments, only one IrDA device was in range, which is typically the case when using IrDA due to its short range and limited cone angle.

Once an IrDA connection has been established, the application performs an IAS query of the remote device for the "Bluetooth:Address" attribute. If the query succeeds, it extracts the Bluetooth Address from the IAS result and passes it to the Bluetooth connect method. The application stops the timer once the Bluetooth connection has been established.

Figure 5 shows the flow of information that occurs between local and remote devices during IrDA-assisted Bluetooth discovery. Processes internal to the device are shown inside the device structure, while communication between the devices is depicted by arrows.

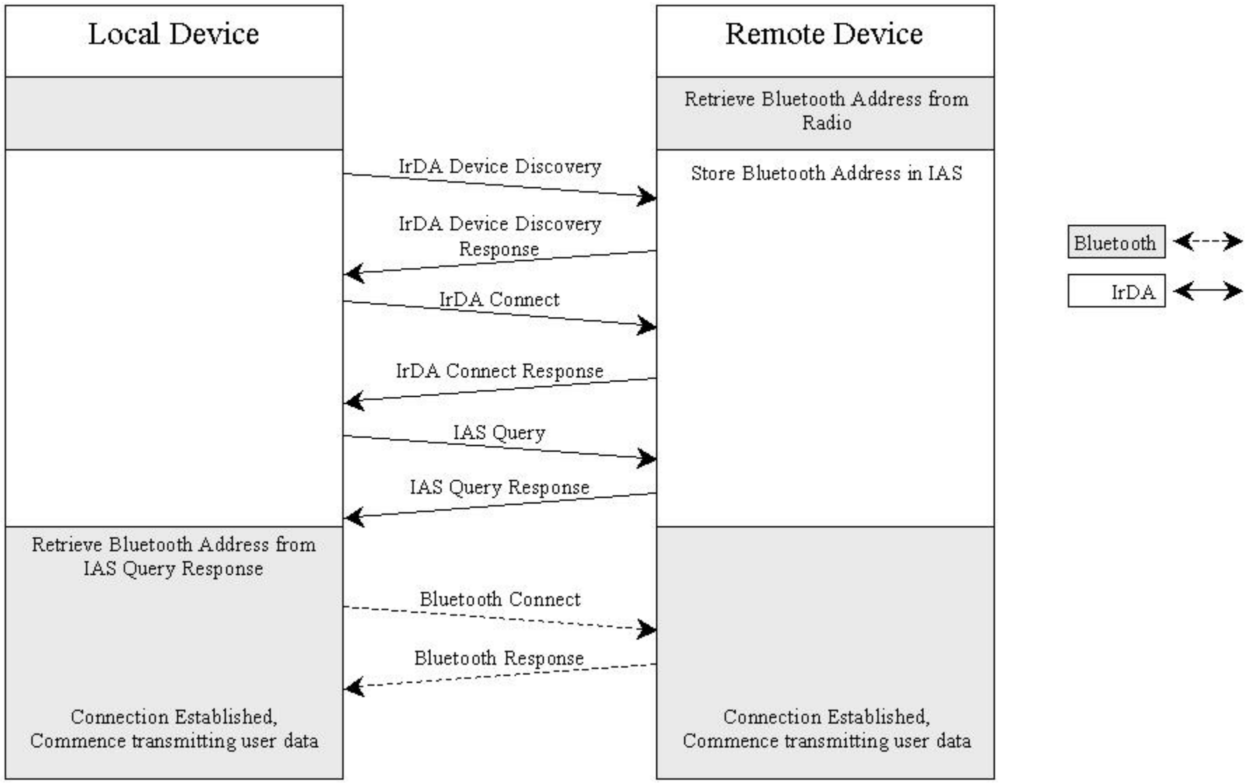

Figure 5. IrDA-Assisted Bluetooth Connection Procedure

Figure 6 shows a screenshot of the test application performing an IrDA-assisted Bluetooth connection. Each step of the algorithm previously described in Figure 5 is reflected in this screenshot. Notice that, while in normal Bluetooth inquiry multiple devices responded, only one device responded to the IrDA device discovery portion of this procedure. IrDA device discovery typically yields fewer device responses than Bluetooth inquiry because of the short-range, directional nature of the infrared signal.

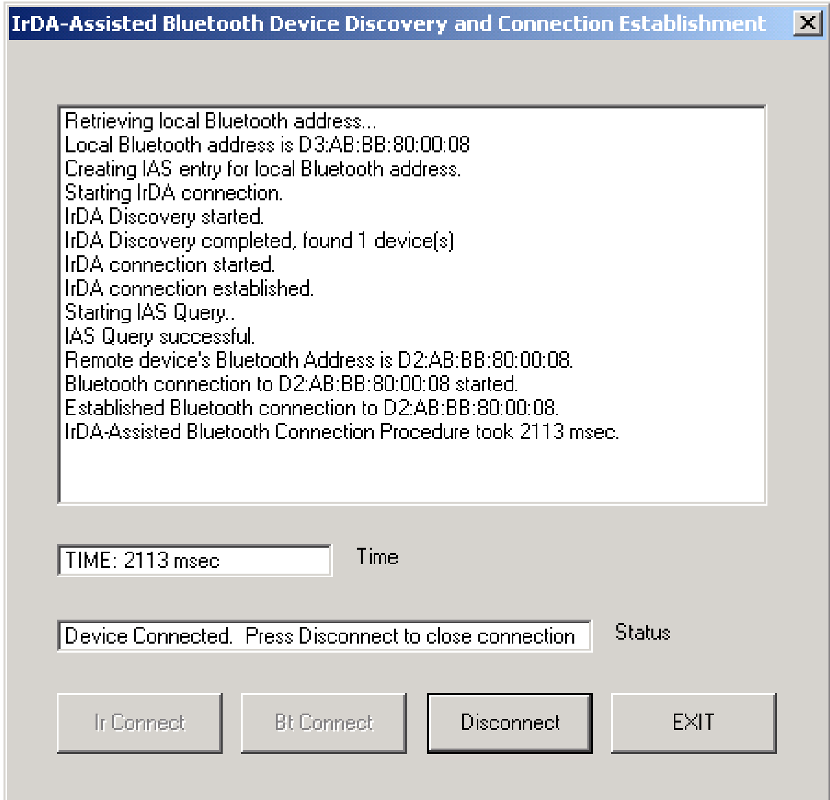

Figure 6. Screenshot of IrDA-Assisted Bluetooth Method

Figure 7 and 8 contain packet traces of the IrDA interactions containing the IAS Query and IAS Query Response. These results were captured using an XTNDAccess IrDA Probe. Observe the IAS Query's class name of "Bluetooth" and attribute name of "Address" in Figure 7. Note the IAS Query Results in Figure 8 which contain the Bluetooth address

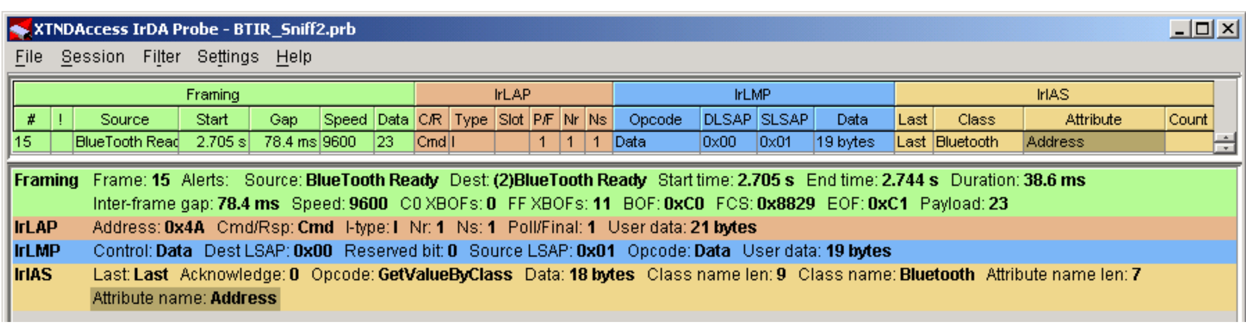

Figure 7. IrDA IAS Query.

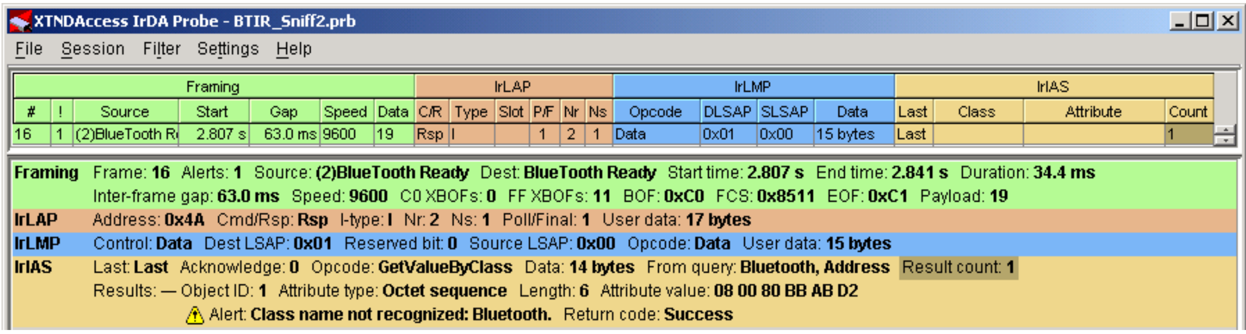

Figure 8. IrDA IAS Query Response.

5. Results

We performed timed repetitions of the Bluetooth device discovery, and confirmed that our test application spent 10.24 +/- 0.04 seconds in Bluetooth inquiry mode. An equivalent number of timed repetitions of IrDA-assisted Bluetooth device discovery showed that this improved method required only 1.05 +/- 0.1 seconds to perform the same device discovery.

Figure 9 shows the time spent performing device discovery and connection establishment in ten trials of each method. The variance between trials is caused by the Bluetooth connection establishment procedure. Bluetooth connection establishment took an average of 12.02 seconds while IrDA-assisted Bluetooth connection establishment took an average of 2.86 seconds. These results show that IrDA-assisted Bluetooth connection establishment is more than 4 times faster than the standard Bluetooth connection establishment approach.

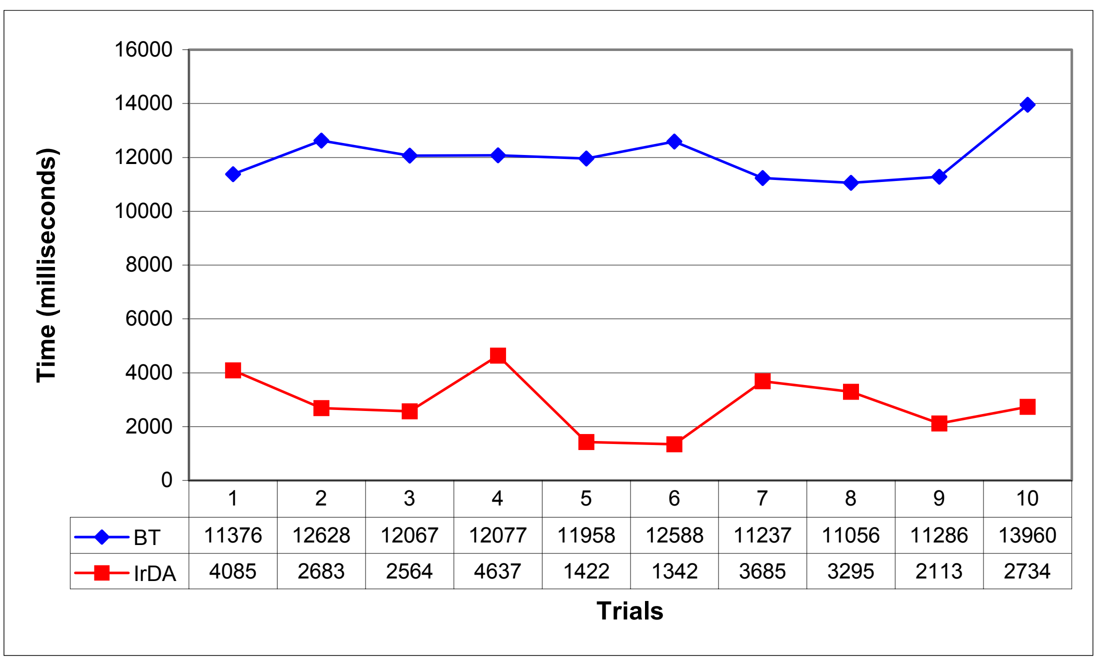

Figure 9. Device Discovery and Connection Establishment.

6. Conclusion

The focus of this research was a case study involving the integration of Bluetooth and IrDA technologies to dramatically improve Bluetooth inquiry and connection time. There continue to be critical questions concerning the means by which these low-cost short-range wireless technologies can be integrated, using the strengths of each technology to overcome the weaknesses of the other. As a case study, integrating Bluetooth and IrDA technologies in the manner described in this paper helps create a more complete solution than either technology can achieve on its own.

Our results show that the integration of Bluetooth and IrDA technologies can greatly improve discovery and connection establishment time between two Bluetooth devices. This is a significant result for certain usage models (such as mCommerce) in which the long inquiry process of Bluetooth negatively affects user experience. By using the rapid device discovery and connection establishment of IrDA to retrieve Bluetooth device information from a remote device, Bluetooth devices are able to connect over four times faster than when using Bluetooth alone. The IrDA-assisted Bluetooth connection establishment procedure can provide an even greater improvement when device selection is considered, since it eliminates the need for user intervention in most situations. This is achieved by IrDA's short range and narrow cone, which perform a type of natural device selection.

The improvements in discovery and connection time that we have demonstrated can be achieved without compromising key strengths of RF solutions such as Bluetooth, namely, longer range, and point-to-multipoint connections that are not limited by line-of-sight obstacles, such as walls and doors.

7. References

[1] "Technical Summary of IrDA Data and IrDA Control." *Infrared Data Association*, Walnut Creek, California. (http://www.irda.org/standards/standards.asp)

[2] Joyce Putscher. "Bluetooth Wireless Chips Catapult to 1.4 Billion Units by 2005." *Cahners In-Stat Group*, Scottsdale, Arizona, July 26, 2000.

[3] Dave Suvak. "Comparing the Benefits of IrDA and Bluetooth." *Wireless Systems Design*, 5(5):31-36, May 2000.

[4] "Specification of the Bluetooth System, Part B: Baseband Specification." *The Bluetooth Special Interest Group*, December 1, 1999.

[5] Theodoros Salonidis, Pravin Bhagwat, and Leandros Tassiulas. "Proximity Awareness and Fast Connection Establishment in Bluetooth." *Proceedings of the First Annual ACM Workshop on Mobile and Ad Hoc Networking and Computing*, Boston, Massachusetts, August 11, 2000.

[6] Charles D. Knutson. "Infrared Communications with IrDA." *Embedded Systems Programming*, 13(6):69-86, June, 2000.

[7] "Serial Infrared Link Access Protocol, IrLAP, Version 1.1." *Infrared Data Association*, Walnut Creek, California, June 16, 1996.

[8] "Link Management Protocol, IrLMP, Version 1.1." *Infrared Data Association*, Walnut Creek, California, January 23, 1996.