Modeling IrDA Performance: The Effect of IrLAP Negotiation Parameters on Throughput

Michael G. Robertson, Scott V. Hansen, Franklin E. Sorenson, Charles D. Knutson
{michaelr, scott, sorenson, knutson}@cs.byu.edu

Computer Science Department
Brigham Young University
Provo, UT 84602, U.S.A.

ABSTRACT

The Infrared Data Association’s (IrDA) infrared data transmission protocol is a widely used mechanism for short-range wireless data communications. In order to provide flexibility for connections between devices of potentially disparate capabilities, IrDA devices negotiate the values of several transmission parameters based on the capabilities of the devices establishing the connection. This paper describes the design and implementation of a software tool, Irdaperf, to model IrDA performance based on negotiated transmission parameters. Using Irdaperf, we demonstrate that for fast data rates, maximizing window size and data size are key factors for overcoming the negative effects of a relatively long link turnaround time. At slower speeds (especially 115.2 Mbps and below), these factors have a less pronounced effect.

KEYWORDS
IrDA, Infrared Data Communications, Performance Analysis, Wireless Data Communications

1 INTRODUCTION

IrDA infrared devices are widely used for short-range wireless data communication. They require little power and can achieve high transmission rates at close range. IrDA devices are small and robust, which makes them ideal in portable, mobile electronic devices, such as laptop computers, cell phones, and personal digital assistants (PDAs). Despite the simplicity of their design and implementation, IrDA devices can transmit data at up to 4 and 16 Mbps.

The IrDA protocol can be utilized for high-speed data transmission as well as for applications with bi-directional real-time requirements. Real-time data, such as voice or video, must be transmitted at a consistent data rate so it can be received and displayed without the user perceiving delays. For these applications, maintaining a steady data stream is more important than achieving high throughput. Indeed, a small percentage of packets can be lost and not retransmitted without degrading the quality of the transmission. The most common use for IrDA, however, is for data transfer from one device to another. The block of data being transmitted may be small (such as a business card or memo) or large (such as a large document or image). For data transmission, throughput must be maximized to make the transmission time as short as possible.

In order to make data transfers as efficient as possible, IrDA devices initiate transmission by negotiating certain parameters. Some parameters (such as data rate) must be agreed upon by the
two devices, while other parameters (such as packet size) must be respected by the other device. This allows more powerful devices to interoperate with smaller, less powerful ones. Depending on the negotiated values, throughput for a given file transfer can vary greatly. In this paper, we examine the IrDA negotiation parameters in order to determine which have the greatest impact on throughput and to specify the values for those parameters that maximize performance.

We have developed an analytical model of the IrDA protocol and have implemented that model in an application named Irdaperf. We have verified this model and the corresponding application with empirical data obtained from testing infrared-enabled devices. Using Irdaperf, we simulated IrDA data transfers using different values for negotiation parameters and compared the resulting throughput, allowing us to determine the parameters that affect throughput most significantly and the values that are ideal for those parameters.

1.2 Related Work

The most relevant published research in the field of IrDA performance was conducted by Barker and Boucouvalas in [3] and [4]. They created a mathematical model with which they could calculate IrDA performance based on certain parameters. In [3] they focused on data size, bit error rate, and minimum turnaround time. They concluded that the best performance is achieved with a small minimum turnaround time and large window size. They reached similar conclusions in [4].

Ozugar, et. al. investigated the IrDA protocol in [6] by focusing on the Go-Back-N and Selective-Reject Automatic Repeat Request modes in HDLC (High Level Data Link Control), which deal primarily with recovery from error conditions. They also examined how changing certain IrDA parameters and device characteristics, such as data size and processor speed, may affect IrDA performance. They concluded that the device’s processor speed determines the parameters that have the biggest impact on performance. For example, packet error rate is significant for fast processors, but not for slower ones.

Other papers that deal with IrDA provide a summary or history of the protocol or a vision of its future. In [2], Stuart Williams examines the IrDA protocol and its development and comments on recent advancements. Some papers concern themselves with hardware implementation issues or the physics of infrared emission and detection. However, to date, no research has undertaken a comprehensive analysis of the basic IrLAP negotiation parameters and their influence on performance.

1.3 IrDA Negotiation Parameters

The IrLAP (IR Link Access Protocol) layer of the IrDA protocol specifies seven parameters that infrared devices must negotiate before data transfer may commence. The parameters govern the size of the packets, the speed at which they are sent, and the timing of their transmission. The negotiation process frees infrared devices from having to support all possible configurations; instead, two devices may choose the best set of parameters that are mutually supported.
Negotiation parameters are divided into two groups: type 0 and type 1. Two devices must agree on the same value for type 1 parameters but may use different values for type 0 parameters, provided that they respect the other’s chosen value. Baud rate and link disconnect/threshold time are the only type 0 parameters. The type 1 parameters include minimum and maximum turnaround time, data and window size, and the number of additional beginning of frame (XBOF) bytes. Figure 1 shows the seven IrDA negotiation parameters, with their types and permissible values.

<table>
<thead>
<tr>
<th>Negotiation Parameter</th>
<th>Type</th>
<th>Permissible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud Rate</td>
<td>0</td>
<td>9600 bps, 19.2 Kbps, 38.4 Kbps, 57.6 Kbps, 115.2 Kbps, 576 Kbps, 1.152 Mbps, 4 Mbps, 16 Mbps</td>
</tr>
<tr>
<td>Link Disconnect/Threshold Time</td>
<td>0</td>
<td>3 s, 8 s, 12 s, 16 s, 20 s, 25 s, 30 s, 40 s</td>
</tr>
<tr>
<td>Maximum Turnaround Time</td>
<td>1</td>
<td>500 ms, 250 ms, 100 ms, 50 ms</td>
</tr>
<tr>
<td>Data Size</td>
<td>1</td>
<td>64 B, 128 B, 256 B, 512 B, 1024 B, 2048 B</td>
</tr>
<tr>
<td>Window Size</td>
<td>1</td>
<td>1-7 frames</td>
</tr>
<tr>
<td>Additional BOFs</td>
<td>1</td>
<td>Below 115.2 Kbps: some fraction of value below 115.2 Kbps: 48, 24, 12, 5, 3, 2, 1, 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>576 Kbps, 1.152 Mbps: 2, 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 Mbps: 0</td>
</tr>
<tr>
<td>Minimum Turnaround Time</td>
<td>1</td>
<td>10 ms, 5 ms, 1 ms, 0.5 ms, 0.1 ms, 0.05 ms, 0.01 ms, 0 ms</td>
</tr>
</tbody>
</table>

**Figure 1: IrLAP Negotiation Parameters**

The IrDA protocol currently supports data rates from 9600 bps up to 16 Mbps. A simple on-off keying (OOK) modulation scheme is used for baud rates of 1.152 Mbps and below. For each bit duration, a 0 is represented by an infrared pulse while a 1 is represented by no pulse. 576 Kbps and 1.152 Mbps modulation schemes use a 1/4 bit duration pulse, while baud rates of 115.2 Kbps and below use a 3/16 bit duration pulse. Higher baud rates require more complex modulation techniques. Pulse-position modulation (PPM) is used for 4 Mbps and Run-length limited (RLL) encoding is used for 16 Mbps. Both modulation schemes encode data using more bits than OOK but achieve faster data rates by spacing the infrared pulses farther apart and using a shorter pulse duration (125 ns and 41.7 ns, respectively).

The link disconnect/threshold time determines the length of time two devices will maintain their link if invalid data (or no data) is received. Each time a device sends data, it begins a timer that counts down to the link disconnect time. Once it receives a response from the other device, it resets the timer. If a device receives no response, it will continue to send data and wait for a response until the timer reaches the disconnect time. If the two devices are unable to communicate for the duration of the link disconnect/threshold time, both sender and receiver will close the link simultaneously.

The minimum and maximum turnaround times control the timing of IrDA transmissions. IrDA data transmission is half-duplex, meaning that only one device may transmit data at a time. To ensure reliable data transmission, the protocol requires that the device sending data cease transmission to receive acknowledgements from the receiving device. Since no data is transmitted during these breaks, the length of time required to receive the acknowledgement affects throughput significantly. To maximize throughput, devices would ideally send as much
data as possible before pausing, and then wait for as little time as possible. Due to physical
limitations, however, a longer pause is sometimes necessary. While a device is sending data, its
transceiver blinds its own receiver such that it cannot perceive remote infrared pulses. After an
infrared transceiver finishes sending data, it must wait to recover from sending before it can
receive from the other device [5]. Some devices also have limited processing power and need
time to process the data they receive before continuing.

The minimum turnaround time allows a device to specify the length of time the other must wait
before it begins to transmit, thus providing itself sufficient time to recover from data
transmission. Each device specifies its own required minimum turnaround time. Higher quality
transceivers will require less time to recover from data transmission than lower quality
transceivers.

The maximum turnaround time specifies the maximum amount of time a device may transmit
before it must stop and wait to receive an acknowledgement. At baud rates up to 115.2 Kbps, 500
ms is used. At higher baud rates, the maximum turnaround time may be shorter. The maximum
turnaround time does not affect throughput but may be used to provide a real-time constraint for
applications such as voice, where maintaining a constant data rate is more important than
maximizing throughput.

The data size specifies the number of bytes that may be contained in one frame. The window size
is the number of contiguous frames that may be transmitted before an acknowledgement must be
received. Smaller window sizes enable devices with limited memory to process incoming data
without overflowing their buffers and losing data. The IrLAP specification dictates that the
maximum turnaround time has priority over the window and frame sizes in determining when
link turnaround must occur. If a device chooses data and window sizes such that the total number
of bytes in a window cannot be transmitted in the time allotted by the maximum turnaround time,
the data and window sizes must be adjusted accordingly.

Figure 2 illustrates an IrDA data transmission between two devices with different window sizes.
In the first diagram, the primary device sends seven consecutive frames before it receives an
acknowledgement. Assuming that the primary is sending 2048 byte packets at 4 Mbps, each
packet requires 4.1 ms to transmit. Assuming a latency of 5 ms, the entire sequence would
require 33.7 ms transmission time. In the second diagram, an acknowledgement must be received
after every frame. The added number of link turnarounds increases total transmission time to
63.7 ms. The same amount of data is sent in both scenarios, but the latter requires almost twice
the time (yielding half the throughput) due to the smaller window size.
Additional beginning of frame markers (XBOFs) are synchronization bytes transmitted before a frame. Their purpose is to allow slow devices to synchronize by giving them more time prior to the transmission of a packet header. No XBOFs are used at 4 Mbps, as the time required to transmit one byte is negligible at that speed. For baud rates of 576 Kbps and 1.152 Mbps, only two XBOFs may be used. At 115.2 Kbps and below, the number of XBOFs may vary from 0 to 48.

2 RESEARCH

This study describes a model of the IrDA protocol stack for use in performance analysis. Our research focuses on the IrLAP layer with data transfer rates between 576 Kbps and 4 Mbps. We have excluded slower transmission speeds because varying negotiation parameters has minimal effect on throughput at those speeds. We have also excluded the 16 Mbps VFIR specification as it is not yet widely available in IrDA devices. Our model provides a means of analyzing the effect of negotiation parameters on throughput and thereby determining those that play the most significant role in improving performance. Using our model, we have discovered several scenarios in which selecting appropriate negotiation parameter values may compensate for other deficiencies that diminish throughput.

2.1 Model

In an IrDA data transfer, a block of data is first passed from the application initiating the transfer to the top level of the IrDA protocol stack. This top layer may be one of several application-layer protocols, depending on the particular usage model being employed. For point and shoot object exchange [8], the appropriate top layer is the IrDA Object Exchange protocol (IrOBEX), which sits on top of the IrLMP layer. For other applications the top layer may be some other high-level protocol where that protocol is the appropriate layer for the chosen usage model. For example,
IrCOMM would be the appropriate high-level protocol for an application that implements legacy serial communications. Applications may also write directly to the IrLMP layer. For the purpose of this study, we have selected the point and shoot object exchange usage model since it is the most commonly used among IrDA devices. As a consequence, our study assumes that applications write directly to IrOBEX as the top layer of the IrDA protocol stack.

The IrDA protocol is comprised of six layers. The Physical layer packages data into individual frames and the Frame Checksum (FCS). The IrLAP layer creates and maintains connections with other devices and controls the negotiation sequence before data transmission commences. The IrLMP (Link Management Protocol) implements service discovery and multiplexing of multiple links on the same channel. The TinyTP (Tiny Transport Protocol) layer is a transport protocol that provides flow control. Finally, the IrOBEX (Object Exchange Protocol) layer provides a simple interface through which applications may send and receive objects.

Each protocol layer receives data from the layer above it, adds relevant header information, and passes it on to the next layer. The lowest layer, the Physical layer, transmits the serialized bytes through the infrared transmitter. Each layer of the IrDA protocol increases the amount of data to be sent by adding its own header. Figure 3 shows the packet formats for each layer of the IrDA protocol.
The Physical and IrLAP layers are the layers most relevant to our study. The Physical layer is responsible for dividing the packets into windows consisting of one to seven frames, depending on the negotiated parameters. The IrLAP layer controls the process through which devices negotiate their parameters.

Our model focuses on two metrics: the number of bytes transmitted and the transfer time. Transfer time is calculated by determining the time required to send bytes at the Physical layer and the time spent waiting for the receiving device to acknowledge received data according to the IrLAP protocol. The number of bytes represents the data to be sent plus the header information added by the protocol layers. Our model accounts for the additional headers added by the upper layers but ignores them when computing transfer time. As our goal is to predict throughput, we focus on only the data transfer time, not the time required for an entire IrDA transmission (including device discovery, link creation, etc.). Figure 4 depicts the flow of information in our model.

![Diagram of Irdaperf model]

**Figure 4: Irdaperf model**

### 2.2 Implementation

We implemented our performance model in an application named Irdaperf. Each component of our performance model is represented by a C++ class. Each class has a method, `GetPerformanceResults`, that returns information from the level below the one
represented by the class. For example, the IrOBEX class invokes the GetPerformanceResults method on the IrLMP class. The IrLMP class adds its header information to the data to be sent and calls the IrLAP class’s method. This continues down to the class representing the Physical layer, which then divides the data into frames, adds its header, and computes the time required to transmit the data, taking into account the data rate, window size, and minimum turnaround time.

Irdaperf allows the user to specify values for each device’s negotiation parameters and the size of the block of data to be sent. After modeling the file transfer as described above, Irdaperf returns performance information, including the total number of bytes sent, the number of header bytes, the transfer time, and the throughput attained.

Figure 5: Irdaperf Application

To facilitate repeatability, we wrote a number of Perl scripts to execute Irdaperf repeatedly with different specified negotiation parameters. From the data collected we draw a number of conclusions concerning negotiation parameters and IrDA performance. The results are described in the following sections.

2.3 Results

From a performance standpoint, minimum turnaround time is among the most important of the IrDA negotiation parameters. Boucouvalas and Baker showed in [3] and [4] that turnaround time should be kept as low as possible during IrDA data transfers to achieve the greatest throughput. Furthermore, our research indicates that the negative effect that a high minimum turnaround time has on throughput increases as the data transfer rate increases. To overcome this loss in throughput, it is necessary that data and window sizes be made as large as possible. We will quantify the throughput loss due to a high minimum turnaround time and show the benefit of using large window and data sizes. Tests indicate that increasing window sizes beyond the maximum currently allowed by the IrDA specification would help alleviate the effects of a high turnaround time. Other IrDA negotiation parameters have less significant effect on throughput.

2.3.1 Minimum Turn Around and Data Rate

While a longer minimum turnaround time always degrades throughput, the effect is more pronounced at higher data transfer rates. Figure 6 shows the percentage of maximum throughput for various data rates and minimum turnaround times. For data rates up to 115.2 Kbps, near maximum throughput is achieved for all minimum turnaround times. At higher transfer rates, however, throughput drops dramatically when the minimum turnaround time is greater than 1ms.
At higher data rates, more data can be sent per unit time. This means that time spent idle represents a greater proportional loss in throughput at higher data rates than at slower ones.

As an example, at 9600 bps, 96 bits can be transmitted in 10 ms. At 4 Mbps, however, 40,000 bits can be transmitted in the same amount of time. Likewise, a 9600 bps device requires 6.7 ms to transmit a 64 byte frame while a 4 Mbps device requires only 16 microseconds. Assuming a 5 ms turnaround time, then, the 9600 bps device will spend approximately 43% of its time idle. The 4 Mbps device, however, will spend over 99% of its time idle. Thus, an infrared device capable of transmitting at 4 Mbps will suffer a greater proportional loss in throughput if it is forced to wait 5 ms between packets than will a 9600 bps device.

![Figure 6: Effect of Minimum Turnaround Time on Throughput at Different Data Rates](image)

2.3.2 Data Size and Window Size

Data size and window size can minimize the performance degradation caused by longer minimum turnaround times. The IrDA protocol allows multiple frame and window sizes. Thus, the same amount of data may be sent in a window consisting of one large frame or multiple small frames.
To maximize throughput, a device should send as much data as possible within the negotiated maximum turnaround time (usually 500 ms). At higher data rates, however, it does not matter whether this is accomplished through large windows or large frames. Figure 7 displays two windows containing 512 bytes of data. The first encapsulates all the data in one frame; the second divides it into four frames. The second window consequently contains 24 bytes of header data, as opposed to the first window’s 6 bytes. The net difference of 18 bytes is insignificant at high data rates of 1.152 Mbps and beyond. Thus, at high baud rates, either the data size or the window size may be increased without adversely affecting throughput due to increased header sizes.

![Figure 7: Comparison of Window and Frame Formats](image)

2.3.3 Compensating for High Minimum Turn Around

At higher data rates, minimizing the length of the turnaround period between data transmission is essential for high throughput. If a device requires a long minimum turnaround time, it is best to reduce the number of times link turnaround must occur. This can be accomplished by using large windows or frames, which ensure that as much data as possible is sent before the sender must wait for an acknowledgement.

When the data and window sizes are small, throughput decreases significantly as the turnaround time increases. Reducing data and window sizes helps minimize the loss in throughput. This happens for two reasons. First, as the window size increases, the number of times the link must be turned around decreases, enabling the sending device to spend a higher percentage of its time transmitting data. Second, sending data in larger frames or windows reduces header overhead.

Figures 8, 9, and 10 show throughput for data transfers at 0.576 Mbps, 1.152 Mbps, and 4.0 Mbps, respectively, with various window sizes and minimum turnaround times. In each chart,
when the window size is small, throughput drops as the minimum turnaround time increases. When a large window size is used, however, throughput decreases much more slowly as the minimum turnaround time approaches 10 ms. Furthermore, the drop in throughput is much more pronounced at higher data rates. For example, at 1.152 Mbps, with a window size of 1, throughput with the longest minimum turnaround time is roughly 58% of the throughput with the shortest minimum turnaround time. At 4 Mbps, however, the percentage drops to 17%. When a window size of 7 is used, the percentages are 91% and 59%. For window sizes of both 1 and 7, throughput drops as the data rate increases. The decrease is much less when the larger window size is used, however.

Figure 8: Throughput at 0.576 Mbps Data Rate for Various Window Sizes and Minimum Turnaround Times
Figure 9: Throughput at 1.152 Mbps Data Rate for Various Window Sizes and Minimum Turnaround Times

Figure 10: Throughput at 4.0 Mbps Data Rate for Various Window Sizes and Minimum Turnaround Times
When a short minimum turnaround time is used, throughput remains fairly constant no matter what window size is used. When a longer minimum turnaround time is necessary, however, throughput suffers considerably, especially at higher data rates. The loss in throughput can be alleviated through the use of large window and frame sizes. While throughput still drops at high data rates when a long minimum turnaround time is needed, the loss is not as substantial for large window and frame sizes.

### 2.3.5 Other Negotiation Parameters

At the higher data rates we have examined in this study, generally very few XBOFs are used. Those that are used have a negligible effect on throughput, as they consist of only a few bytes at most. More XBOFs are used at lower baud rates (1.152 Mbps and below), but they do not hurt throughput significantly. Within this lower range of baud rates, the maximum number of XBOFs allowed is higher as the baud rate increases. As an example, 48 XBOFs is the maximum number allowed at 1.152 Mbps, while at 9600 bps, the maximum is 4. These XBOFs represent only 0.7% of the largest packet size for each baud rate and therefore have little effect on throughput.

The maximum turnaround time does not affect performance significantly. It is an upper bound on the time a device may transmit and therefore does not present a problem with throughput. When the maximum turnaround time is sufficiently small, it can restrict the amount of data being sent at one time by a device, which effectively places a limit on the window size and data size. Maximum turnaround time has more bearing on real time applications, in which maximizing throughput is not as important as guaranteeing that data arrives in a timely manner.

The last negotiation parameter, Link Disconnect/Threshold Time, likewise does not affect throughput. It specifies how long a device will wait upon receiving invalid data before it closes the link. It only is relevant in error conditions and thus plays no part in normal data transmission.

### 3 CONCLUSIONS

The key to attaining high throughput in infrared data transfers is to minimize the amount of time a device is not transmitting data. Since IrDA is a half-duplex protocol, by necessity a device must stop transmitting data and wait for an acknowledgement from the receiving device before it can continue. At low data rates, the time lost to turnaround is not significant. At higher data rates, however, the loss in throughput is much greater. Specifying a small minimum turnaround time, then, will increase throughput. In situations where this is not possible, using a large window and data size can alleviate the negative effect of a large minimum turnaround time by increasing the amount of data sent between turnarounds. Doing so also decreases the amount of header data that must be transmitted. High throughput may then be maintained despite the long minimum turnaround time.

### 4 FUTURE WORK

In the future we hope to extend our work to include relevant parameters that are not negotiated by IrDA devices. These include hardware characteristics such as processor power, memory size and access speed, and the time required for a device to convert the light pulses it receives into
data. One of IrDA’s strengths lies in the fact that a wide variety of devices can communicate via infrared. This means, however, that devices of varying computational power must be able to interoperated at a common level. IrDA’s negotiation parameters allow devices to agree on common settings but processor and memory limitations may still limit the degree to which disparate devices may work together. It would also be beneficial to study higher data rates, such as VFIR, which transmits at 16 Mbps, to determine how choosing negotiated link parameters can maximize throughput at those speeds.

7 REFERENCES


