

A Study of the Suitability of IrOBEX for High-Speed Exchange of Large Data Objects

Casey T. Deccio, Joseph Ekstrom, D. Ryan Partridge, Kevin B. Tew, and Charles D. Knutson
 {cdeccio, jce, rpartrid, tewk, knutson}@cs.byu.edu

Abstract—This paper demonstrates that careful tuning of the OBEX and IrLAP negotiated parameters allows OBEX to scale well for use with large data objects and high transmission rates. Due to the substantial time overhead inherent in link turnarounds, minimizing turnarounds during the transmission of a large object helps to maximize link efficiency. The IrLAP window size and OBEX packet size significantly impact the number of required turnarounds during the transmission of a large object. When these parameters are properly tuned, maximum throughput can be achieved, and OBEX performs efficiently at high data rates.

Index Terms—Object Exchange, OBEX, IrDA, IrLAP

I. INTRODUCTION

OBJECT EXCHANGE (OBEX) [1] is a binary session layer protocol designed to transmit data objects between resource-limited devices. It is based on HTTP, which has become a worldwide standard for object exchange across the Internet. Since it is binary rather than text-based, OBEX is an attractive option for resource-constrained wireless devices because processing overhead is much lower than for traditional text-based protocols. OBEX was originally developed by the Infrared Data Association (IrDA) for infrared wireless devices, and has since been adopted by the Bluetooth SIG as a standard for object exchange.

The wireless technology market has experienced tremendous growth. Users today expect to be able to communicate and access data anytime, anywhere, using almost any device. Traditional mobile devices such as cell phones and PDAs must now handle complex jobs involving large files, such as MP3 audio and MPEG video. The demand for efficient transmission of large data and media objects between wireless devices at increasingly higher speeds calls into question the scalability of the OBEX protocol.

In [2] it was shown that the IrDA Link Access Protocol (IrLAP) [3] scales to a data rate of 100 Mbps without modification. Building on that work, this paper shows that careful tuning of the OBEX and IrLAP negotiated parameters allows OBEX to scale well for use with large data objects and high transmission rates.

The rest of the paper is organized as follows: In section II we discuss the IrLAP and OBEX protocols and their key parameters. In section III we introduce the mathematical model on which we have based our study. Section IV presents our experimental results. In Section V we present our conclusions.

II. BACKGROUND

In order to properly tune OBEX for maximum performance, it is necessary to understand the underlying link layer. IrLMP

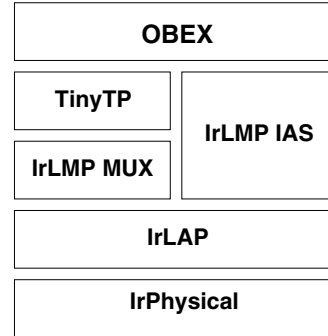


Figure 1. OBEX and the IrDA protocol stack. [1] [3] [4] [5].

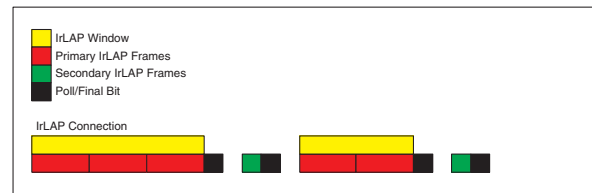


Figure 2. Example IrLAP windows.

[4] and TinyTP [5] add a constant 3 byte overhead but do not influence the IrLAP or OBEX parameters in any other way. This section is a brief overview of the IrLAP and OBEX protocols, and highlights key parameters that impact performance.

A. IrLAP

The Link Access Protocol (IrLAP) layer [3] establishes and manages the link between a primary and a secondary infrared device (see Figure 1). Because point-to-point infrared communication is inherently half-duplex, the receiving device may not transmit until it receives a “clear to send” from the transmitting device. A “clear to send” is signaled when the poll/final bit of an IrLAP frame is set, forcing a link turnaround. A device

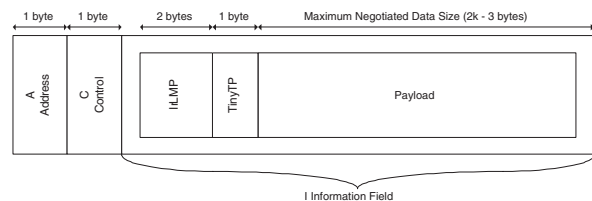


Figure 3. IrLAP Frame.

may send a *window* of multiple unacknowledged *frames* before setting the poll/final bit (see Figure 2).

Several parameters are negotiated when a link is established. Window sizes and minimum turn around time in the VFIR specification are 127 and 100 ms respectively [6]. In [2] the authors base their research parameters on the VFIR specification, but make allowances for the faster 100 Mbit transmission speed. It should be noted that 100 Mbit IrDA is purely theoretical and has no standard specification. Of these parameters, minimum turnaround time, frame length, and window size directly impact link performance. The *minimum turnaround time* is the latency required by a transmitting device after it sends a packet with the poll/final bit set, and before the other device is permitted to begin transmitting. Some factors that determine the minimum turnaround time are photo receptor energy dissipation time, device interrupt latency, data processing overhead, and memory storage latency. *Frame length* is the maximum size of the Information field in an IrLAP frame (see Figure 3) [3]. *Window size* is the maximum number of unacknowledged frames that can be sent before the link must be turned around. Frame length and window size dictate the frequency with which a link must be turned around, and hence are critical to link utilization efficiency. Table I illustrates the impact of these two parameters on aggregate overhead due to link turnaround during transmission of a single (relatively small) data object. In this table, object size is assumed to be 10 MB and turnaround time is assumed to be 2 ms.

TABLE I
IMPACT OF FRAME LENGTH AND WINDOW SIZE ON IRLAP
TURNAROUNDS

| Frame Length | Window Size | Number of Turnarounds | Total Turnaround Time |
|--------------|-------------|-----------------------|-----------------------|
| 512 B | 1 | 20480 | 40.96 s |
| 2 KB | 1 | 5120 | 10.24 s |
| 512 B | 7 | 2926 | 5.85 s |
| 2 KB | 7 | 731 | 1.46 s |
| 512 B | 127 | 161 | 322 ms |
| 2 KB | 127 | 40 | 80 ms |

B. OBEX

All information carried by OBEX is encapsulated in OBEX headers, conceptually similar to headers in HTTP. These headers may contain meta-data about the object (such as name, length, MIME-type, etc.) and may include one or more body headers containing the object itself. Each header is either fixed length or length prefixed [1].

OBEX transmits objects using GET and PUT methods similar to HTTP. The OBEX 1.3 specification defines a 2 byte packet length header field resulting in a maximum OBEX packet length of 64 KB [1]. This packet size limitation requires that GET and PUT operations for large objects be segmented into multiple OBEX packets. The OBEX standard requires that each *request* packet transmitted in either a GET or PUT operation must be acknowledged by a *CONTINUE response* packet, resulting in an effective session layer window size of one [1].

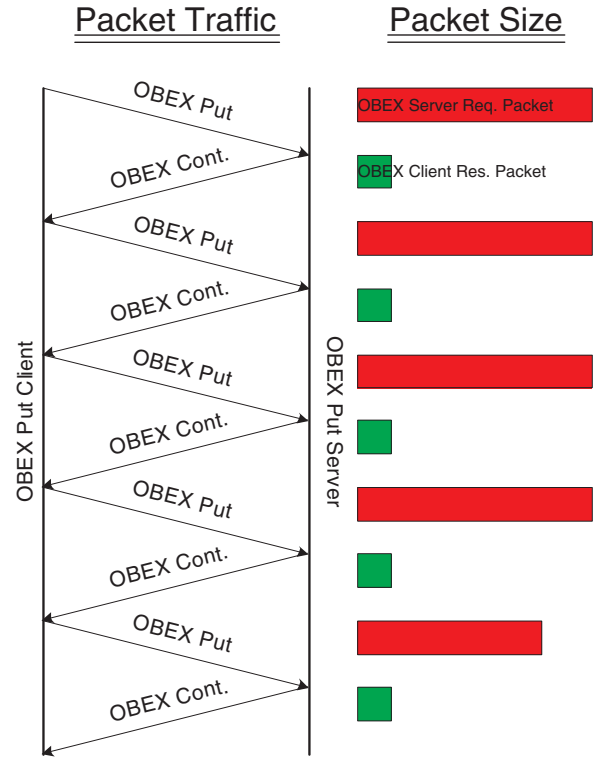


Figure 4. OBEX Conversation.

Because of the half duplex nature of IrDA and the effective OBEX window size of one, subsequent *request* packets cannot be transmitted until a *CONTINUE* acknowledgment is received (see Figure 4). Therefore, each *request/response* pair generates two turnarounds, one after the *request*, and a second after the *CONTINUE response*.

III. IRDA AND OBEX MODEL

This section presents a mathematical model of the interactions and behavior of OBEX and IrLAP. Variables presented in this section are defined in Table II. The first equation models the number of OBEX layer turnarounds. We build on this equation to derive a model for the number of turnarounds at the IrLAP layer. Finally, based on these two equations, we construct a model for the time necessary to transmit an object of arbitrary size using OBEX over IrLAP.

A. OBEX Turnarounds

Figure 5 illustrates the way in which OBEX packetizes a large object for transmission. The first packet in an OBEX PUT operation typically contains object meta-information (such as name, length, etc.), although it is not strictly required to do so. A body header may or may not be included in this first packet, but if included is normally shorter than in subsequent packets, due to the space occupied by headers for meta-information. While no specific requirements are imposed on the order or content of the first packet, for the sake of our model we assume that the first packet includes name and length headers plus a necessarily shortened body header. The value 10 in the following

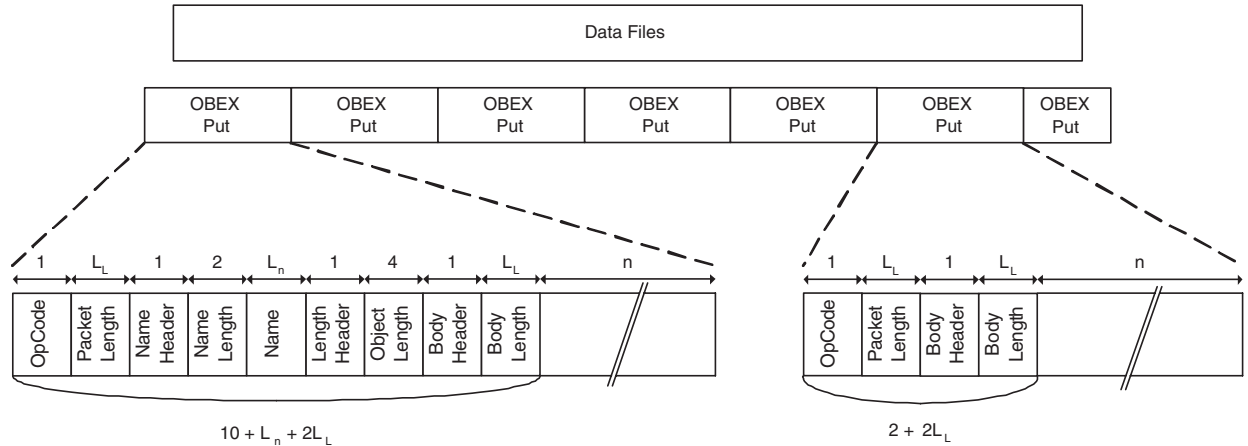


Figure 5. OBEX object packetization. The first packet has additional headers that specify object length and object name. Subsequent packet headers only need to contain the packet and body lengths.

TABLE II
MATHEMATICAL MODEL VARIABLES

| | |
|-----------|--|
| T_O | number of OBEX turnarounds |
| T_L | number of IrLAP turnarounds |
| s_o | object size (bytes) |
| p_0 | payload size of first OBEX packet (bytes) |
| p_n | payload size of all subsequent OBEX packets (bytes) |
| s_p | OBEX maximum packet size (bytes) |
| L_N | length of OBEX name field (bytes) |
| L_L | length of OBEX length field (bytes) |
| B_{rem} | bytes that account for an incomplete OBEX packet (bytes) |
| f | IrLAP frame size (bytes) |
| i | payload size of I field (bytes) |
| w | IrLAP window size |
| r | transmission rate (bits/sec) |
| t | transmission time (sec) |
| t_s | time to send data to receiver |
| t_w | latency imposed by min turnaround time and processing overhead |
| t_r | time for receiver to transmit a response |

equation reflects the constant overhead of the name and length headers. L_N and $2L_L$ reflect the variable length overhead associated with these headers.

$$p_0 = s_p - (10 + L_N + 2L_L) \quad (1)$$

For subsequent packets, we assume a common case in which packets consist entirely of body headers, incurring only the overhead of the packet length field and the body header length field.

$$p_n = s_p - (2 + 2L_L) \quad (2)$$

Although the OBEX specification fixes the length field at two bytes, we have chosen to make it a variable in order to allow for experimentation with packet sizes greater than 64 KB.

Since the first OBEX packet has a different payload capacity than subsequent packets, we subtract p_0 from s_o and then divide by p_n to yield the number of packets with payload size p_n required to transmit the object. To this, we add one to account for the first packet (with payload size p_0). Because each OBEX *request* packet must be acknowledged with a *CONTINUE response* packet, the number of packets sent by the sender and receiver is symmetrical, therefore we multiply by two. Since each OBEX packet requires a turnaround, the number of OBEX packets is analogous to the number of required OBEX turnarounds.

This yields the following equation, which models OBEX turnarounds as a function of object size.

$$T_O = 2 \left(\left\lceil \frac{s_o - p_0}{p_n} \right\rceil + 1 \right) \quad (3)$$

B. IrLAP Turnarounds

IrLAP turnarounds are dependent upon IrLAP windows, which are calculated with respect to OBEX packets. The number of *full* OBEX packets, or packets of size s_p , is denoted by $\left\lceil \frac{s_o - p_0}{p_n} + 1 \right\rceil$. The final OBEX packet will normally be less than full, and has size B_{rem} which is given by

$$B_{rem} = ((s_o - p_0) \bmod p_n) + \left(\left\lceil \frac{(s_o - p_0) \bmod p_n}{p_n} \right\rceil \right) * (2 + 2L_L) \quad (4)$$

Each full OBEX packet is transmitted in $\left\lceil \frac{s_p}{i * w} \right\rceil$ IrLAP windows. Similarly, B_{rem} requires $\left\lceil \frac{B_{rem}}{i * w} \right\rceil$ IrLAP windows for transmission. We multiply the number of IrLAP windows for a single OBEX packet by the number of OBEX packets, and add the number of IrLAP windows for the under-filled packet to produce the total number of windows required to transmit an object of size s_o . Each IrLAP window requires an acknowledgment, so the number of windows is identical to the number of turnarounds, and hence must be multiplied by two.

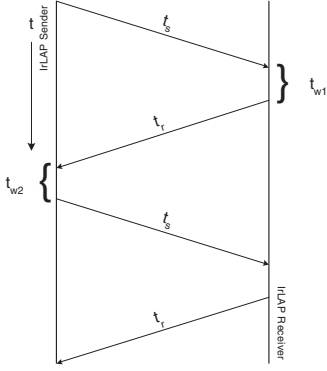


Figure 6. IrLAP transmission time diagram. t_s is sender's transmission time. t_r is receivers transmission time. t_{w1} and t_{w2} are the turnaround time for the receiver and sender respectively

The resulting equation for IrLAP turnarounds is

$$T_L = 2 \left(\left(\left\lceil \frac{s_p}{i * w} \right\rceil * \left\lfloor \frac{s_o - p_0}{p_n} + 1 \right\rfloor \right) + \left\lceil \frac{B_{rem}}{i * w} \right\rceil \right) \quad (5)$$

C. Transmission Time

The general formula for the time required to transmit data, receive a response, and be ready to transmit again is given by

$$t_{round-trip} = t_s + t_{w1} + t_r + t_{w2} \quad (6)$$

A single IrDA round trip is illustrated by Figure 6. We have defined $t_s = \frac{(f+2)*w}{r}$, where 2 is the size of the Address and Control fields of the IrLAP frame (see Figure 3). The time equation is divided into three smaller equations, shown below. t_{full} is the time required to transmit full OBEX packets. $t_{partial}$ represents the time for an under-filled OBEX packet that is larger than one IrLAP window. t_{end} is the time due to any remaining bytes that do not completely fill one IrLAP window.

$$t_{full} = t_{round-trip} * \left(\left\lceil \frac{s_p}{i * w} \right\rceil * \left\lfloor \frac{s_o - p_0}{p_n} + 1 \right\rfloor \right) \quad (7)$$

$$t_{partial} = t_{round-trip} * \left(\left\lceil \frac{B_{rem}}{i * w} \right\rceil \right) \quad (8)$$

$$t_{end} = \frac{B_{rem} \bmod (i * w)}{r} + t_{w1} + t_r + t_{w2} \quad (9)$$

Equations (7) and (8) are derived directly from the two terms in (5). They represent the time required to send one complete IrLAP window multiplied by the number of IrLAP windows represented by each term. The last equation, (9) is the time to transmit any remaining bytes smaller than one IrLAP window in size. Summing (7), (8), and (9) yields the total amount of time (in seconds) required to transmit an OBEX object of a given size.

$$t = t_{full} + t_{partial} + t_{end} \quad (10)$$

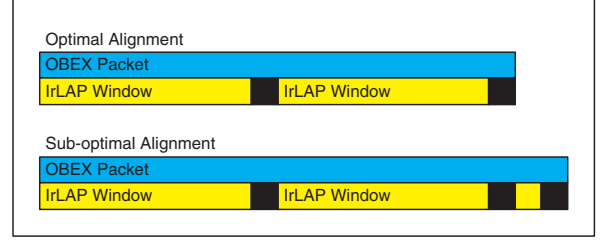


Figure 7. OBEX packet alignment with IrLAP windows.

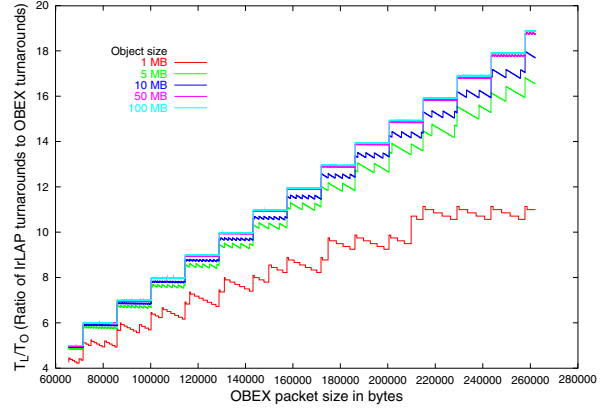


Figure 8. Ratio of IrLAP turnarounds (T_L) to OBEX turnarounds (T_O) for OBEX packet sizes of 64 KB to 256 KB, showing five object sizes (1, 5, 10, 50, and 100 MB).

IV. RESULTS

A. OBEX - IrLAP Turnaround Ratio

In this section we apply our model to a number of scenarios in order to characterize the performance of OBEX as data objects increase in size and as transmission speeds increase. Our first experiment compares the ratio of IrLAP turnarounds to OBEX turnarounds as OBEX maximum packet size increases beyond the specified 64 KB limit. For this experiment, the IrLAP frame size is 2 KB and the window size is 7. As OBEX packet size increases, the number of OBEX turnarounds will obviously decrease. However, of particular interest is the effect that larger OBEX packet sizes have on the IrLAP/OBEX turnaround ratio.

Figure 7 illustrates the manner in which OBEX packets are segmented into underlying IrLAP windows. As OBEX maximum packet size increases, it approaches optimal IrLAP window utilization. Immediately beyond this point, each OBEX packet requires one more nearly empty IrLAP window. Figure 8 shows the effect of IrLAP window utilization on turnaround ratio. Note the series of gradual ratio decreases, each followed by a sharp increase. The local minima, or "sweet spots", occur when OBEX packet size achieves optimal alignment with IrLAP window boundaries. The significance of these "sweet spots" is illustrated when considering total object transmission time.

B. Object Transmission Time

This section explores the effect of varying the IrLAP window size while holding the OBEX packet size constant. This

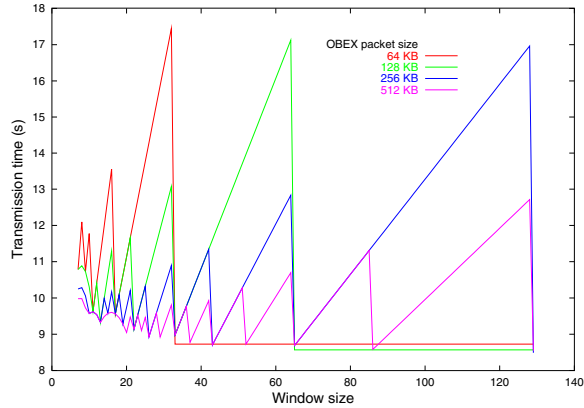


Figure 9. Transmission time of a 100 MB object for increasing IrLAP window sizes, showing OBEX packet sizes from 64 KB to 512 KB.

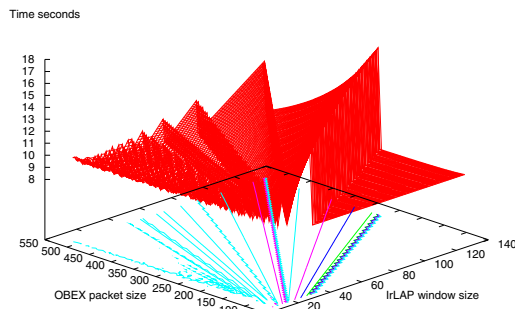


Figure 10. Relationship between window size and packet size.

experiment models the transmission of a 100 MB object where $i = 2$ KB, $r = 100$ Mbps, and w varies from 7 to 127. Figure 9 shows that transmission time reaches a (local) maximum for certain IrLAP window sizes, dropping immediately thereafter to a (local) minimum. These local minima and maxima reflect the effect of the “sweet spots” on transmission time. In other words, transmission time is smallest when OBEX packet size falls near IrLAP window boundaries, avoiding unnecessary additional IrLAP turnaround. Table III summarizes the minimum transmission times for OBEX packet sizes 64, 128, 256, and 512 KB.

TABLE III

TRANSMISSION TIME FOR VARIOUS OBEX PACKET SIZES

| OBEX Packet Size | Optimal Window Size | Minimum Trans. Time | Link Utilization |
|-----------------------|---------------------|---------------------|------------------|
| 64 KB | 33 | 8.73 s | 95.99% |
| 128 KB | 65 | 8.57 s | 97.83% |
| 256 KB | 129 | 8.49 s | 98.70% |
| 512 KB | 257 | 8.40 s | 99.76% |
| Theoretical Min. Time | | 8.38 s | |

$$s_o = 100 \text{ MB}, i = 2 \text{ KB}$$

Figure 10 illustrates the performance surface formed by

OBEX packet size and IrLAP window size parameters. The flat plateau region to the right is where IrLAP window size has met or exceeded the OBEX packet size. The valleys correspond to “sweet spots” as previously described. The ridges depict the worst case combinations of maximum OBEX packet size and IrLAP window size which may reduce throughput by as much as 50%.

V. CONCLUSION

We have shown that the OBEX protocol, when tuned to maximize IrLAP performance, scales to handle high data rates and large objects. While OBEX packet / IrLAP window alignment is the key to OBEX performance tuning, perfect alignment is not necessary. Slightly under-filled IrLAP windows achieve near optimal data throughput. However, slightly over-filled IrLAP frames produce a significantly detrimental effect on throughput, decreasing performance by nearly 50%. Packet sizes that are powers of two (ironically the default specifications for maximum OBEX packet sizes and IrLAP window sizes) result in the worst possible performance due to slightly overfilled IrLAP windows. These negative effects can be mitigated by appropriate tuning of packet sizes within the OBEX session layer.

REFERENCES

- [1] “Irda object exchange protocol obex,” Infrared Data Association, Walnut Creek, California, 2003.
- [2] A. C. Boucouvalas and V. Vitsas, “100 mb/s irda protocol performance evaluation,” in *Proceedings of IASTED International Conference on Wireless and Optical Communications (WOC 2001)*, Banff, Canada, June 2001, pp. 49–57.
- [3] “Serial infrared link access protocol (irlap),” Infrared Data Association, Walnut Creek, California, 1996.
- [4] “Link management protocol,” Infrared Data Association, Walnut Creek, California, 1996.
- [5] “Tiny tp: A flow-control mechanism for use with irlmp,” Infrared Data Association, Walnut Creek, California, 1996.
- [6] “Serial infrared physical layer specification for 16 mb/s addition (v1r),” Infrared Data Association, Walnut Creek, California, 1999.
- [7] “Serial infrared physical layer specification (irphy),” Infrared Data Association, Walnut Creek, California, 1998.