

OBEX Performance Evaluation and Parameter Optimization for High Speed IrDA Links

Pi Huang, A.C. Boucouvalas

Multimedia Communications Research Group

School of Design, Engineering and Computing

Bournemouth University, Fern Barrow, Poole, BH12 5BB, U.K.

Email: {phuang, tboucouv}@bournemouth.ac.uk

Abstract

OBEX is a session protocol designed to transmit data objects between resource-limited devices. It has been adopted as the framework for object exchange for wireless transports including IrDA and Bluetooth. A mathematical analysis for OBEX throughput over the IrDA protocol stacks is carried out in this paper. OBEX maximum packet size and turnaround time are optimized for the system throughput. Numerical results show that OBEX throughput improves significantly by the optimization. We also offer suitable OBEX parameters selection guidelines for different data rates by considering the trade off between system performance and hardware requirements.

Index Terms

OBEX, IrDA, Bluetooth, Optimization

I. INTRODUCTION

Object Exchange protocol (OBEX), originally developed by IrDA—Infrared Data Association, is a session protocol, and can best be described as a binary HTTP protocol [1]. It is optimized for ad-hoc wireless links and can be used to exchange objects such as files, pictures, calendar entries (vCal) and business cards (vCard). OBEX does not specify the top or bottom API making it very flexible and can run over most transports like TCP/IP and Bluetooth [2] [3]. Today, OBEX is built-in for devices like PDAs, mobile phones and printers. Microsoft Windows2000 and the recent Linux distributions have also built-in OBEX support.

Significant research has been conducted to the physical or link layer of different wireless technologies. However, it is also important to ensure the upper layer working compatibly with the lower layer. In this paper a mathematical analysis is developed for OBEX throughput over the IrDA protocol stacks. The stacks include protocols from the physical to the transport layer: IrPHY [4] [5], IrLAP [6], IrLMP [7] and TinyTP [8]. An equation for OBEX throughput is derived in the presence of bit error rate (BER).

Protocol parameters are then optimized for maximum throughput. Although the maximum data rate for IrDA links is 16Mbps [5], the optimization study is also carried out for data rates up to 100Mbps in order to give guidelines for future systems. Due to space limitation, only OBEX packet size and turnaround time are studied in this paper. In fact, optimizing the IrDA link layer parameters can further improve the system throughput [9].

The paper is organised as follows: First, we briefly describe the IrDA protocol stacks and OBEX in section II. In section III, we carry out a systematic analysis and derive the throughput formula for OBEX. An optimization study for OBEX parameters is given in section IV to maximize the system throughput and offer suitable parameters selection guidelines.

II. IRDA PROTOCOL STACKS

IrDA featured devices are widespread for the business and mobile environments. The IrDA protocol stack is the layered set of protocols particularly aimed at point-to-point infrared communications and the applications needed in that environment.

A. IrPHY (IrDA Physical layer):

The IrDA Physical layer defines a directed half duplex serial infrared communications links through free space to facilitate the point-to-point communication. Framing data such as begin and end of frame flags (BOFs and EOFs) and cyclic redundancy check (CRCs) are also considered to be part of the physical layer. Transceivers with data rates of 4 and 16Mbps employ 32 bit CRC [4] and [5].

B. IrLAP (IrDA Link Access Protocol):

IrLAP is the link access layer and it is based on the HDLC (High-level Data Link Control) protocol, which is extensively used in data communication networks. By using mechanisms including retransmission, low-level flow control and error detection, IrLAP provides reliable data transfer. It transmits data in the form of frames of length l and

organizes the transmission in a manner of go-back-N (GBN) error recovery. When the transmitter completes the transmission of a window of length N —number of information frames (I) that can be sent before link turnaround, it then sets the P bit in the last I-frame to signal link turnaround and request an acknowledgement from the receiver. Referring to standards [5] and [6], the window size and frame size range from 1-127 and 128bit-16384bit respectively.

C. IrLMP (IrDA Link Management Protocol):

IrLMP provides support for multiple software applications or entities to operate independently and concurrently, sharing the single link provided by IrLAP between the transceivers [7]. It also offers higher level discovery which includes address conflict resolution on IrLAP discovery. In this paper, we consider IrLMP has only one application. After the connection initial negotiation, IrLMP adds two bytes overhead to the upper layer packet as the link management information.

D. IrTinyTP (IrDA Tiny Transport Protocol):

TinyTP is an optional IrDA layer, although it is so important that it should generally be considered as a required layer [8]. It provides two functions: flow control on a per-LMP-connection basis; Segmentation and reassembly (SAR). SAR is normally disabled by using the default connection parameter. If the communicating peers have large enough buffer size and short propagation delay between them, which is the case for OBEX applications, there is no need to perform flow control. In this case, TinyTP typically adds one byte of information to each upper layer packet.

E. OBEX (Object Exchange):

OBEX is a session protocol and can be resided on top of any reliable transport (e.g. IrTinyTP). It works for many devices that cannot afford the substantial resources required for an HTTP server. OBEX is enough like HTTP to serve as a compact final hop to a device.

OBEX follows a client/server request-response (stop and wait) paradigm for the conversation format [1]. The terms client and server refer to the originator and receiver of the OBEX connection, not necessarily who originated the low level IrLAP connection. Requests are issued by the client (the party that initiates the OBEX connection). Once a request is issued, the client waits for a response from the server before issuing another request. The request/response pair is referred to as an operation. “PUT” and “GET” are the two types of operations used in OBEX. As the name indicates, the “PUT” operation sends one object from the client to the server, while the “GET” operation requests that the server return an object to the client. The maximum and minimum length for both request and response packets are 512K bits and 2048 bits respectively [1].

Fig.1 illustrates OBEX in the process of packetising a large object for transmission when OBEX is in the ‘PUT’ operation. The initial OBEX request packet (first packet) will typically, although not strictly required to do so, have certain headers. We assume that first packet includes the object information of name, length and body header. The connection-oriented session allows capabilities information to be exchanged just once at the start of the connection, and allows state information to be kept. The subsequent

packets therefore only have to give the overhead information of the packet length field and the body header length field.

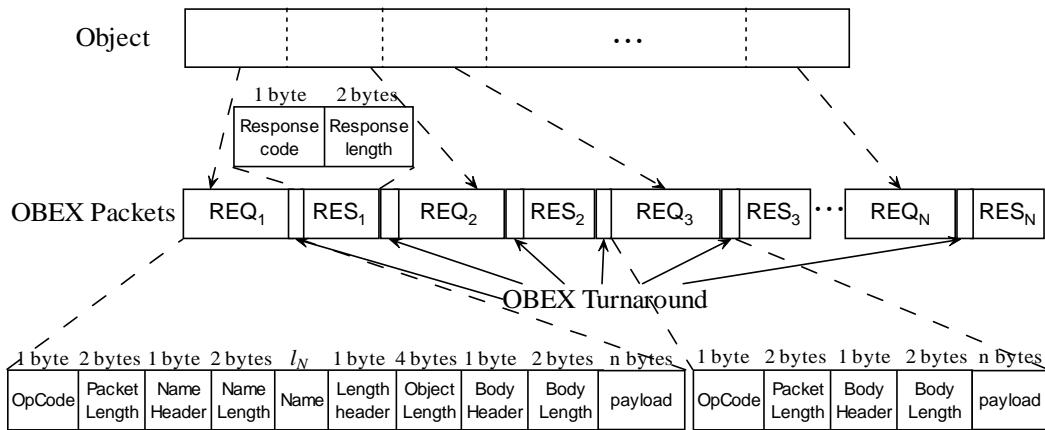


Figure 1: OBEX packetisation, where REQ stands for the request packet, while RES is the response packet.

Fig.2 shows the protocol mapping of an OBEX request/response packet pair down to the link layer of the stacks (IrLAP). Symbols used in the modelling are listed in Table 1. Since OBEX uses ‘stop and wait’ as its transmission scheme, the transmitter has to wait for the acknowledgement before transmits the next packet. If the IrLAP parameters $l \cdot N < P_{send}$, one OBEX packet requires fragmentation in order to fit in IrLAP frames and requires more than one IrLAP window for its transmission. If $l \cdot N \geq P_{send}$, the OBEX packet can be accommodated within a single IrLAP window. The transmitter will simply set the ‘P’ bit at the end of each transmitted window and indicate an expected acknowledgement request from the other side. Thus, here we only need to consider the case of $l \cdot N \leq P_{send}$. Sending one OBEX request packet therefore needs several full IrLAP

windows and it is likely that there will also be a single incomplete IrLAP window at the end.

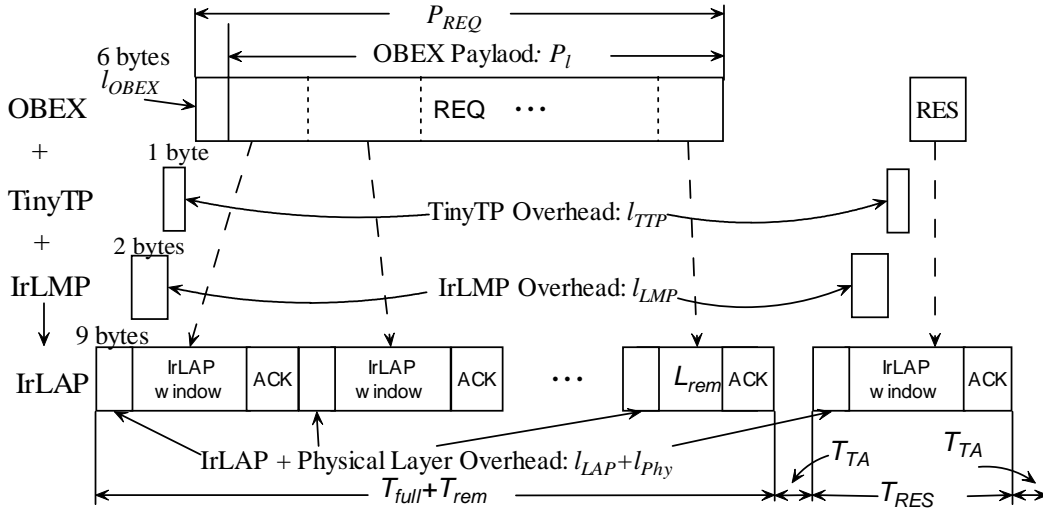


Figure 2: Mapping OBEX, TinyTP, IrLMP to IrLAP frames.

TABLE I:

Symbol.	Parameter Description	Unit
C	Link data rate	bit/s
p_b	Link bit error rate	-
p	Frame error rate	-
N	Number of frames in one IrLAP window	-
l	I-frame message data length	bit
l_{phy}	Physical layer overhead: BOF+EOF+CRC	48bit
l_{LAP}	S-frame length/ I-frame overhead	24bit
l_{LMP}	IrLMP overhead	16bit
l_{TTP}	IrTinyTP overhead	8bit
l_{OBEX}	OBEX request packet overhead	48bit
P_{REQ}	OBEX request packet size	bit
P_{send}	Total packet length for IrLAP to send: $P_{REQ} + l_{LMP} + l_{TTP}$	bit
t_l	Transmission time of an Information (I)-frame	sec
t_s	Transmission time of a Supervision (S)-frame	sec
t_{ack}	Time to transmit an acknowledgement packet	sec
t_{ta}	IrLAP minimum turnaround time	sec
t_{Fout}	IrLAP F-timer time-out period	sec
T_{TA}	OBEX turnaround time	sec

III. MATHEMATICAL MODEL

For the purpose of deriving the mathematical model, we assume that packets are sent in the OBEX ‘PUT’ operation mode. We only consider the ‘connected’ OBEX packets (not the first packet). Therefore, OBEX packet overhead is fixed in length. The length of the packet header is illustrated in Fig.3. The mathematical model and derivation of OBEX throughput for a connection using the IrDA protocol stacks follows next.

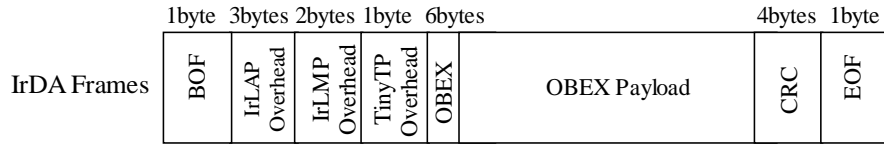


Figure 3: IrDA IrLAP frame structure.

Our throughput model is based on [10], which gave a detailed study for IrLAP and derived the IrLAP throughput formula in the presence of BER. It uses the concept of window transmission time t_w [11]. t_w denotes the average time needed for a full window transmission. It is the average time taken from the beginning of the window’s first frame transmission to the beginning of the first frame of the next window. It incorporates time needed for data frame transmissions, acknowledgements, and timer timeouts. Referring to [10], t_w is given by:

$$t_w = Nt_I + p(t_{Fout} + t_s) + t_{ack} \tag{1}$$

Where $t_I = (l + l_{LAP} + l_{phy})/C$, $t_s = (l_{LAP} + l_{phy})/C$, $p = 1 - (1 - p_b)^{l + l_{LAP} + l_{phy}}$, $t_{ack} = 2t_{ta} + t_s$ and

$$t_{Fout} = t_I + 2t_{ta} .$$

The number of frames correctly transmitted in one full window transmission N_{corr} is also given in [10]:

$$N_{corr} = \frac{(1-p)(1-(1-p)^N)}{p} \quad (2)$$

As described in section II, we consider fixed overheads of 2 bytes and 1 byte for IrLMP and TinyTP respectively for each OBEX packet. Therefore, IrLAP has to transmit a packet with length of $P_{send} = P_{REQ} + l_{LMP} + l_{TTP}$ for each OBEX packet, Fig.2. One OBEX packet will be transmitted in several full IrLAP windows and one incomplete IrLAP window. By combining (1) and (2), the average time for sending the full IrLAP windows of one OBEX request packet is given in (3), where *floor* means round down to the nearest integer.

$$T_{full} = \text{floor}\left(\frac{P_{send}}{l \cdot N_{corr}}\right) \cdot t_w \quad (3)$$

The length of the incomplete IrLAP window is given by:

$$L_{rem} = P_{send} \bmod (l \cdot N_{corr}) \quad (4)$$

The number of frames in the incomplete IrLAP window is:

$$N_{in} = \text{ceil}(L_{rem} / l) \quad (5)$$

Where *ceil* means round up to the nearest integer.

The probability of having error/errors in the incomplete IrLAP window is:

$$p_{in1} = 1 - (1-p)^{N_{in}} \quad (6)$$

Due to the small value of p , p_{in1} can be approximated as:

$$p_{in1} = 1 - (1 - p)^{N_{in}} \approx 1 - (1 - N_{in}p) = N_{in} \quad (7)$$

While error/errors occur in transmitting the incomplete IrLAP window with probability p_{in1} , due to the randomness of error occurrence, it is sufficient to assume that on average the error occurs in the middle of the window, and a retransmission will occur to recover the error with window length of $0.5N_{in}$. If further error/errors occur in the retransmission with probability of $p_{in2} = p_{in1}(1 - (1 - p)^{0.5N_{in}}) \approx 0.5N_{in}^2 p^2$, another retransmission window is needed with window length of half the previous, i.e. $0.25N_{in}$, and so on. When the retransmission window is less than 1, we consider the whole window has been successfully transmitted. By including all the retransmissions, the average time for transmitting the incomplete window T_{rem} is derived in (8), where X is an integer with value of $X = \text{ceil}(0.5N_{in})$ which satisfies $\frac{1}{2^X} \cdot N_{in} \leq 1$

$$\begin{aligned} T_{rem} &= N_{in}t_I + p(t_{Fout} + t_s) + t_{ack} + p_{in1}\left(\frac{1}{2}N_{in}t_I + p(t_{Fout} + t_s) + t_{ack}\right) + \dots + p_{inN}\left(\frac{1}{2^X}N_{in}t_I + p(t_{Fout} + t_s) + t_{ack}\right) \\ &= \left(1 + \sum_{i=1}^X \left(\left(\frac{1}{2}\right)^{\frac{1}{2}i(i+1)} (N_{in}p)^i\right)\right) N_{in}t_I + \left(1 + N_{in}p + \sum_{i=2}^X \left(\left(\frac{1}{2}\right)^{\frac{1}{2}i(i-1)} (N_{in}p)^i\right)\right) (p(t_{Fout} + t_s) + t_{ack}) \end{aligned} \quad (8)$$

Since the OBEX response packets are used only for acknowledgement (no payload), the packet length is equal to the OBEX overhead l_{OBEX} . Due to the small size of the OBEX response packet, it can be accommodated in a single IrLAP window and we assume it is error free. The time required to transmit a response packet T_{RES} is:

$$T_{RES} = \frac{(l_{OBEX} + l_{LMP} + l_{TTP})}{C} + t_{ack} \quad (9)$$

By adding up all the time portions, as shown in Fig.2, the average time for transmission of one OBEX packet is:

$$T = T_{full} + T_{rem} + T_{RES} + 2T_{TA} \quad (10)$$

The OBEX throughput, which is defined as the useful data bit per second, is therefore given by:

$$D = \frac{P_{REQ} - l_{OBEX}}{T} \quad (11)$$

In order to normalize the OBEX throughput, throughput efficiency is given by:

$$TPE = D / C \quad (12)$$

Using this mathematical model, in Fig.4, we compare the system throughput efficiency (TPE) for different data rates by using different OBEX packet sizes and turnaround time. OBEX TPE is plotted against BER. An average implemented IrLAP frame size l of 2048bit, window size N of 20 and minimum turnaround time t_{ta} of 10^{-5} s are used in the simulation. Unless otherwise specified, the same IrLAP parameters are used throughout this paper.

As shown in Fig.4, the throughput efficiency decreases as the BER increases, and the system has larger TPE for low data rates. The results also show that different OBEX packet sizes and turnaround times have different effects on the TPE. P_{REQ} and T_{TA} have significant effects on the high data rate links, i.e. nearly 30% TPE difference between P1 and P2 is shown between Curve5 and Curve6 at $BER=10^{-6}$. Therefore appropriately

adjusting the OBEX parameters may improve the system performance significantly for high data rates.

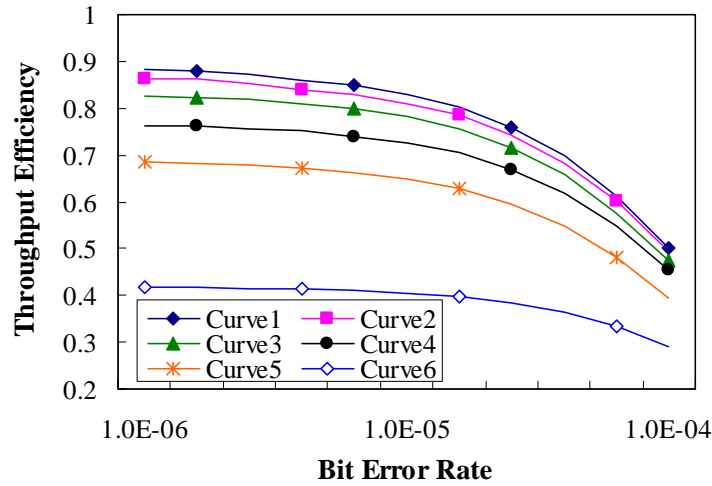


Figure 4: OBEX TPE using different P_{REQ} and T_{TA}
 Curve1: P1, C=1Mb/s; Curve2: P2, C=1Mb/s; Curve3: P1, C=10Mb/s; Curve4: P2, C=10Mb/s;
 Curve5: P1, C=100Mb/s; Curve6: P2, C=100Mb/s.
 where P1: $P_{REQ}=50\text{Kbit}$, $T_{TA}=10^{-4}\text{s}$; P2: $P_{REQ}=500\text{Kbit}$, $T_{TA}=10^{-3}\text{s}$.

IV. OBEX PARAMETERS OPTIMISATION

As shown in the previous section, OBEX packet size and turnaround time have significant effects on the system TPE. However, OBEX performance can be optimized by choosing appropriate OBEX parameters for any data rates and BERs. In this section, we are going to carry out a detailed study for the OBEX packet size and turnaround time by mathematical analysis and simulation results.

First, the OBEX packet size P_{REQ} is a negotiable parameter for the connection. Because its value can be chosen from 2Kbits to 512Kbits [1], it is very important to understand the effect of P_{REQ} on the throughput.

Considering the large size of P_{REQ} , it is sufficient to assume that P_{REQ} is much larger than the incomplete IrLAP window L_{rem} . By combining equations (3) and (8), the time to transmit P_{REQ} becomes:

$$T_{full} + T_{rem} \approx \frac{(P_{REQ} + l_{LMP} + l_{TTP}) \cdot t_w}{l \cdot N_{corr}} \quad (13)$$

P_{REQ} is also large by comparing to the overheads of IrLMP, IrTinyTP and OBEX, thus, we assume $P_{REQ} - l_{OBEX} \approx P_{REQ}$ and $P_{REQ} + l_{LMP} + l_{TTP} \approx P_{REQ}$. Applying these assumptions to the OBEX throughput equation (11), it becomes:

$$D \approx \frac{P_{REQ}}{\frac{P_{REQ} \cdot p \cdot t_w}{l \cdot (1-p) \cdot (1-(1-p)^N)} + T_{RES} + 2T_{TA}} \quad (14)$$

By using the approximation of equation (7) and setting $A = (T_{RES} + 2T_{TA}) \cdot l \cdot (1-p) \cdot N \cdot p$, we can further simplify equation (14) as:

$$D = \frac{P_{REQ} \cdot N \cdot p \cdot l \cdot (1-p)}{P_{REQ} \cdot p \cdot t_w + A} \quad (15)$$

In equation (15), factor A is independent of P_{REQ} . By considering $N \cdot p \gg p$ and $l \cdot (1-p) \gg t_w$, we then have $l \cdot (1-p) \cdot N \cdot p \gg p \cdot t_w$. Therefore, we can conclude that the throughput D increases with P_{REQ} , which means the system always acquires its maximum throughput at a maximum value of P_{REQ} . According to the standard [1], the upper limit of 512kbit should therefore be used for optimum P_{REQ} .

Second, the OBEX turnaround time T_{TA} in (14), is only presented in the denominator. It leads to the conclusion that throughput D increases when T_{TA} decreases.

As shown in the analysis, larger P_{REQ} should offer better OBEX throughput. For T_{TA} , we have proved that the system throughput will always benefit from smaller T_{TA} . It is interesting to see if the throughput can be improved significantly if we were to use even larger P_{REQ} values and smaller T_{TA} . Using (11) and (12), in Fig.5, the OBEX TPE as a function of OBEX packet size P_{REQ} in the range of $5 \times 10^4 \sim 1 \times 10^6$ bit is examined at BER of 10^{-6} with T_{TA} of 10^{-4} s at three different data rates of 1Mbps, 10Mbps and 100Mbps. In the same figure, the OBEX TPE is also plotted against T_{TA} in the range of $1 \times 10^{-5} \sim 1 \times 10^{-2}$ s at 1Mbps, 10Mbps and 100Mbps. P_{REQ} is set at 500Kbit.

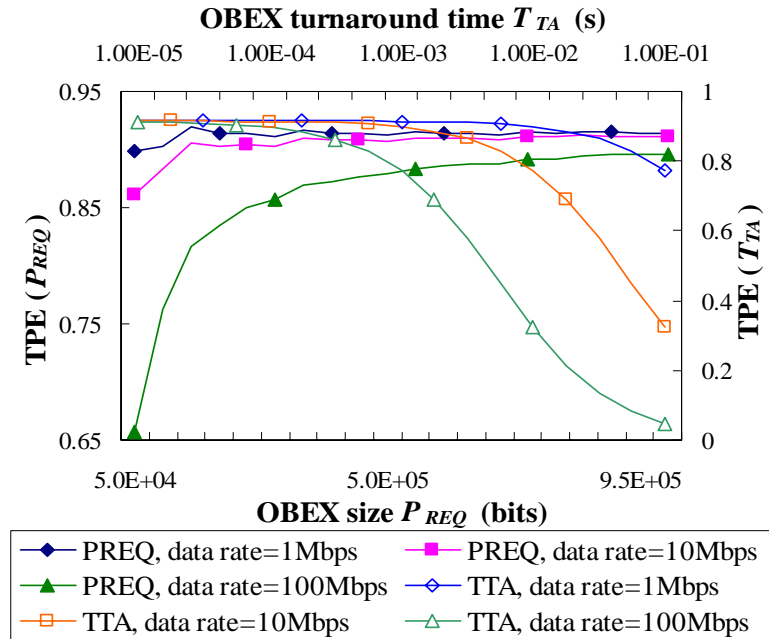


Figure 5: OBEX TPE against P_{REQ} with $T_{TA}=10^{-4}$ s, and OBEX TPE against T_{TA} with $P_{REQ}=500$ Kbit. BER= 10^{-6} .

For P_{REQ} , the corresponding OBEX TPE curves show non-linear shapes in Fig.5. This is due to the change of the incomplete IrLAP window L_{REM} that we ignored in (13).

However, in spite of the slight fluctuation, TPE increases with P_{REQ} at all data rates. This verifies our analysis for the optimum P_{REQ} . The TPE at 100Mbps benefits most as P_{REQ} increases, while at lower speeds, the benefits are small for very large P_{REQ} . For T_{TA} , all of the three OBEX TPE curves decrease with increasing T_{TA} . Links at high data rates are more sensitive and vulnerable to the large T_{TA} than at the low data rates.

Larger P_{REQ} will improve the throughput, however, larger memory buffer size has to be assigned for temporarily storing the unfinished (current transmitting) packet. Given the fact that buffer size is constrained for the resource-limited wireless device, P_{REQ} should not be over size. For T_{TA} , this high layer turnaround time depends on the CPU speed of the communication peers rather than the IrDA transceivers themselves. Therefore, smaller T_{TA} requires faster CPU. Smaller T_{TA} can improve the throughput but very small T_{TA} only leads to trivial improvement on the throughput, Fig.5.

In order to give a suitable parameters selection guideline, we are going to search the suitable P_{REQ} and T_{TA} values for various data rates based on the 'best throughput efficiency'.

By considering 2% sacrifice off the best TPE, we can compromise the throughput and the hardware requirement. For P_{REQ} , the best TPE ($PBTPE$) is obtained by using very large P_{REQ} equal to 4Mbit with $T_{TA}=10^{-4}$ s. OBEX TPE is calculated by increasing P_{REQ} until $TPE = 0.98 \times PBTPE$ for each data rate. The corresponding P_{REQ} values are recorded as the recommended values. Using the same process, the best TPE ($TBTPE$) for T_{TA} is calculated by using very small T_{TA} of 10^{-6} s with $P_{REQ}=500$ Kbit. The recommended T_{TA} for

each data rate is obtained by recording the corresponding T_{TA} value for $TPE = 0.98 \times TBTPE$. By using the suggested P_{REQ} and T_{TA} values, very good throughput (98% of the best TPE) can be obtained for the system with less hardware requirements. In Fig.6, the recommended P_{REQ} and T_{TA} values are plotted against data rate at $BER=10^{-6}$. Note that T_{TA} is shown in logarithmic scale.

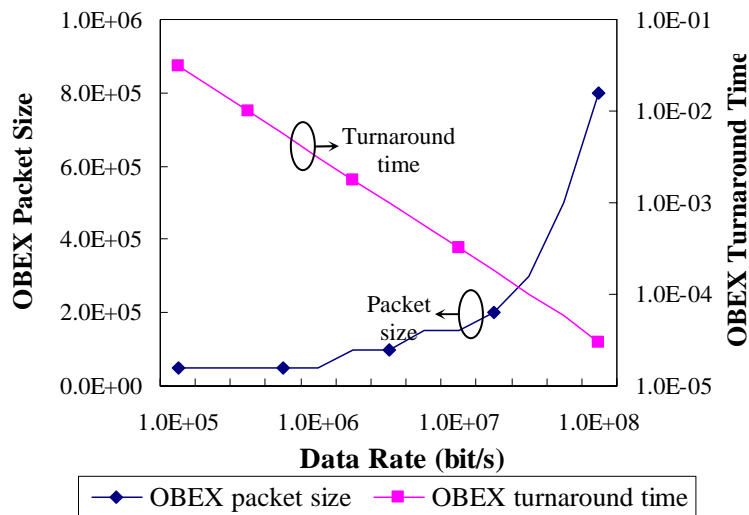


Figure 6: P_{REQ} and T_{TA} selection guidelines

As shown in Fig.6, P_{REQ} increases its size exponentially after the data rate reaches 1Mbps, while T_{TA} decreases linearly with the data rate. For the 100Mbps links, P_{REQ} of 0.8Mbit and T_{TA} of 0.05ms are sufficient to obtain very good TPE for the system.

Fig.7 shows the OBEX TPE by using the recommended optimum P_{REQ} and T_{TA} values. Results are plotted against BER in three different data rates of 1Mbps, 10Mbps and 100Mbps.

Comparing Fig.7 to the non-optimum cases in Fig.4, considerable improvement on OBEX TPE is shown for the same BER. The system is benefited significantly by using the optimum P_{REQ} and T_{TA} values especially for high data rates.

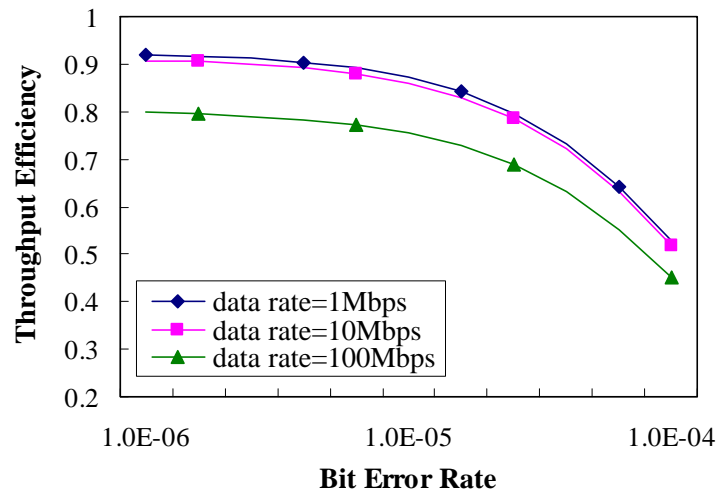


Figure 7: OBEX TPE using the recommended P_{REQ} and T_{TA} values

V. CONCLUSION

This article examined the performance of OBEX protocol running on top of the IrDA protocol stack. We carried out an analytical model to derive the OBEX throughput equation which depends on BER. Based on the mathematical model, we examined the impact of OBEX packet size and turnaround time on OBEX TPE. In order to maximise throughput, optimum OBEX packet size and turnaround time have been studied. The analysis has showed that the throughput always benefits by a large OBEX packet size and a small turnaround time. A suitable OBEX parameter selection guideline is given for

different data rates. The system performance shows significant improvement by applying the optimized values.

REFERENCES

- [1] IrDA, Object Exchange Protocol (IrOBEX), Version 1.3, March, 2003
- [2] Sairam, K.V.S.S.S.S.; Gunasekaran, N.; Redd, S.R.; 'Bluetooth in wireless communication', IEEE Communications Magazine, Volume: 40 Issue: 6, Jun 2002 Page(s): 90 -96
- [3] Srivastava, A.; Friday, R.J.; Ritter, M.W.; Filippo, W.S.; 'A study of TCP performance over wireless data networks' VTC 2001 Spring. IEEE VTS 53rd, Volume: 3, 2001 Page(s): 2265 - 2269 vol.3
- [4] IrDA, Serial Infrared Physical Layer Specification, Version 1.1, 1995
- [5] IrDA, Serial Infrared Physical Layer Specification for 16Mb/s Addition (VFIR)–Errata to version 1.3, 1999
- [6] IrDA, Serial Infrared Link Access Protocol (IrLAP), Version 1.1 1996
- [7] IrDA, Link Management Protocol (IrLMP), Version 1.1, 1996
- [8] IrDA, Tiny Transport Protocol (IrTinyTP), Version 1.1, 1996
- [9] A. C. Boucouvalas, V.Vitsas 'Optimum Window and Frame Size for IrDA Links'.I.E.E.Electronic Letters, Vol.37, No.3, pp194-196, 2001.
- [10] P. Barker, A.C. Boucouvalas, "Performance modelling of the IrDA protocol for Infrared Wireless Communications" IEEE Communications magazine, Vol.36, No.12, pp.113-117, December 1998.
- [11] Bux W., Kummerle K. & Truong H.L., "Balanced HDLC procedures: A performance Analysis", IEEE Trans.Com, 1980, com-28, pp.1889-1898.