OBEX and high speed IrDA links

Pi Huang and 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 higher layer protocol adopted as the framework for wireless object exchange for wireless transports including IrDA and Bluetooth. In this paper, we develop a model which leads to derivation of the OBEX throughput over the IrDA protocol stack for various data rates in the presence of bit errors. We then optimize the OBEX packet size and turnaround for maximum OBEX throughput. The results show significant improvement on OBEX performance using optimized parameters.

1. Introduction

IrDA (Infrared Data Association) and Bluetooth are two of the most widely adopted indoor wireless technologies. A large number of portable devices on the market today have been equipped with IrDA ports and more recently Bluetooth chipsets for their wireless communication needs. These devices range widely from mobile phones and digital cameras to portable computers and printers [1] and [2]. In recent years, there have been many studies on the design and performance of physical and link layers of IrDA and Bluetooth systems. In contrast there have not been enough studies to examine the important interaction between the lower and higher layers. OBEX is a compact, efficient, binary session layer protocol that enables a wide range of devices to exchange data in a simple and spontaneous manner. OBEX is defined by members of IrDA for the purpose of interconnecting a wide range of devices that support IrDA protocols [3]. It is not, however, limited in use to only the IrDA environment. OBEX has been adopted by other wireless technology transports, including Bluetooth, as the framework for wireless object exchange.

Many link layer performance improvements and evaluations have been undertaken recently to address different infrared link issues including the impact on throughput of future increase in data rates and design issues, minimum link turnaround time [4] and processing speed [5]. In [6], optimum link parameters used to maximize the link throughput are also presented. However, no study has been carried out for the upper layers. It is of interest to examine the compatibility and performance of OBEX over the IrDA stacks. In this paper, we first briefly describe the IrDA protocol stacks and OBEX. We then carry out a mathematical model for OBEX which allows derivation of the OBEX throughput taking into account the lower IrDA protocol stack. Based on this model, the impact of the OBEX packet size and turnaround time on the throughput is studied in a wireless environment of varying BER. In an effort to improve the performance of OBEX when line BER is variable, we then study the optimum OBEX parameters which allow maximum throughput for any BER at different data rates.

2. 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 checks (CRCs) are also considered to be part of the physical layer. Transceivers with data rates of 4 and 16Mbps employ 32 bit CRC [7] and [8].

B. IrLAP (IrDA Link Access Protocol)

IrLAP is the IrDA link layer protocol. It is based on High-Level Data Link Control (HDLC) and Synchronous Data Link Control (SDLC) with extensions for some unique characteristics of infrared communications [9].

IrLAP provides simplex reliable data transfer using the following mechanisms:

- Retransmission.
- Low-level flow control.
- Error detection.

By dealing with reliable data transfer at low level, upper layers are free from this concern and can be assured correct delivery. IrLAP transmits data as frames and windows subject to go-back-N (GBN) error recovery, as shown in Fig.1. When the transmitter completes a full window transmission, it sets the P bit in the last data frame to signal a request for an acknowledgement from the receiver. Referring to standards [8] and [9], the window size and frame size range from 0-127 and 128bit-16384bit respectively.

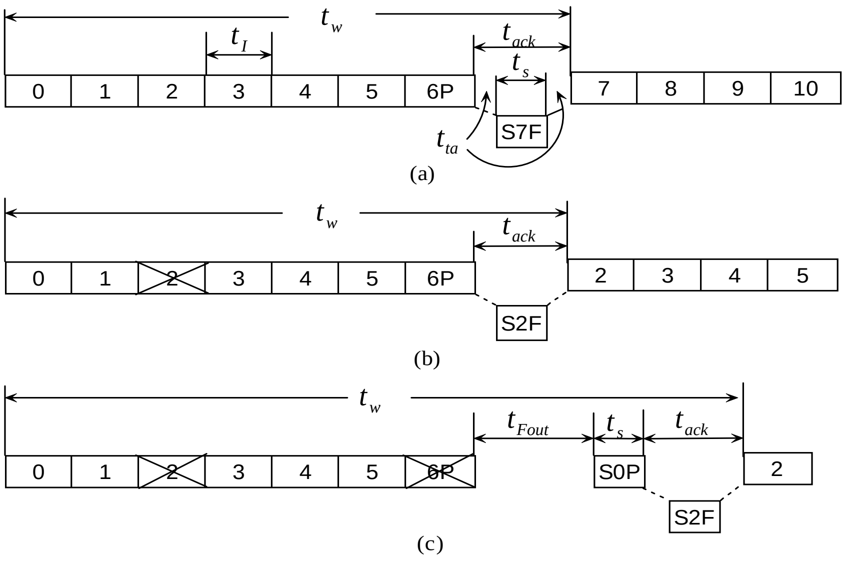

Figure 1: A 7 frame window transmission model. (a) Error free transmission of an IrLAP window (b) Retransmission frames due to error frame at I=3 (c) Retransmitted frames and F-timer delay due frame error at I=3 and I=7.

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 [10]. It also offers higher level discovery which includes address conflict resolution on IrLAP discovery. In this paper, we consider IrLMP has only one application even it allows multiple. 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 [11]. 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 [3]. 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 [3].

Fig.2 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.

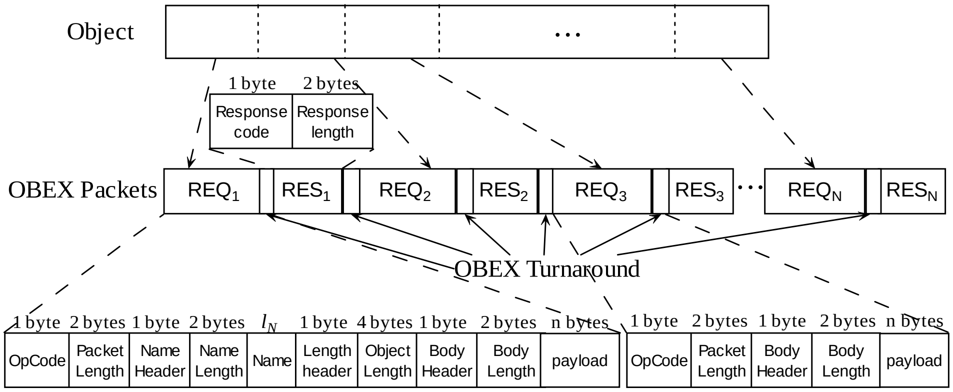

Figure 2: OBEX paketisation, where REQ stands for the request packet, while RES is the response packet.

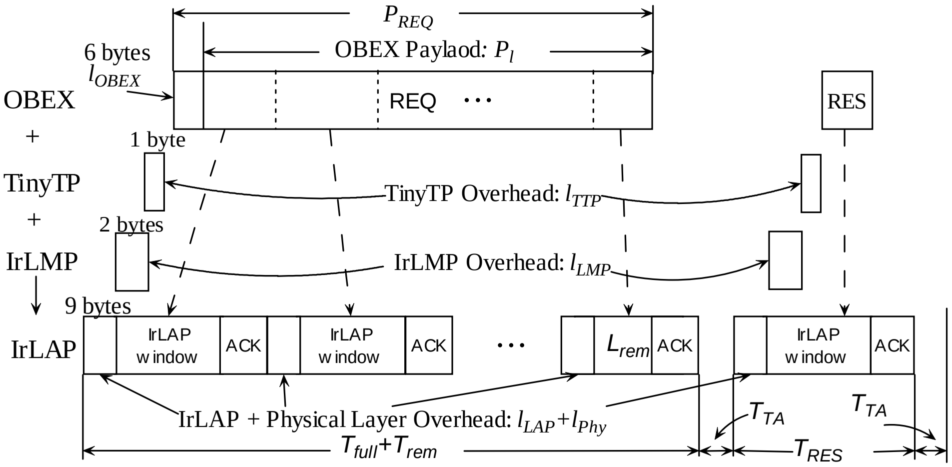

Figure 3: Mapping OBEX, TinyTP, IrLMP to IrLAP frames.

Fig.3 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, we only need to consider the case of $l \cdot N \le P_{send}$. 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.

TABLE 1:

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	16bits
L_{TTP}	IrTinyTP overhead	8bit
l_{OBEX}	OBEX request packet overhead	48bits
P_{REQ}	OBEX request packet size	bit
P_{send}	Total packet length for IrLAP to send	bit
t_I	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

3. 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.4. The mathematical model and derivation of OBEX throughput for a connection using the IrDA protocol stacks follows next.

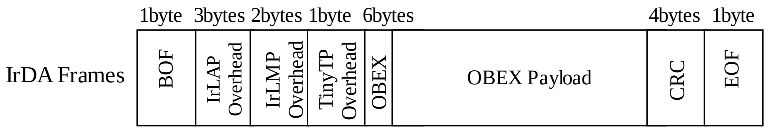

Figure 4: IrDA frame structure.

Our throughput model is based on [12], 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 [13]. 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 [12], 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 $t_{Fout} = t_{\mathrm{I}} + 2t_{ta}$.

The number of frames correctly transmitted in one full window transmission N_{corr} is also given in [12]:

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

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.3. 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.

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

The length of the incomplete IrLAP window is given by:

$$L_{rem} = P_{send} \operatorname{mod}(l \cdot N_{corr}) \tag{4}$$

The number of frames in the incomplete IrLAP window is:

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

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

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

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

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

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

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 $p_{in2} = p_{in1}(1-(1-p)^{0.5N_{in}}) \approx 0.5{N_{in}}^2 p^2$, 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 $X = ceil(0.5N_{in})$ which satisfies ${}^{1}\!/_{2X} \cdot N_{in} \le 1$.

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

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:

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

By adding up all the time portions, as shown in Fig.3, the average time for transmission of one OBEX packet is:

$$T = T_{full} + T_{rem} + T_{RES} + 2T_{TA} \tag{10}$$

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

$$D = \frac{P_{REQ} - l_{OBEX}}{T} \tag{11}$$

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

$$TPE = D/C \tag{12}$$

Using this mathematical model, in Fig.5, 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.

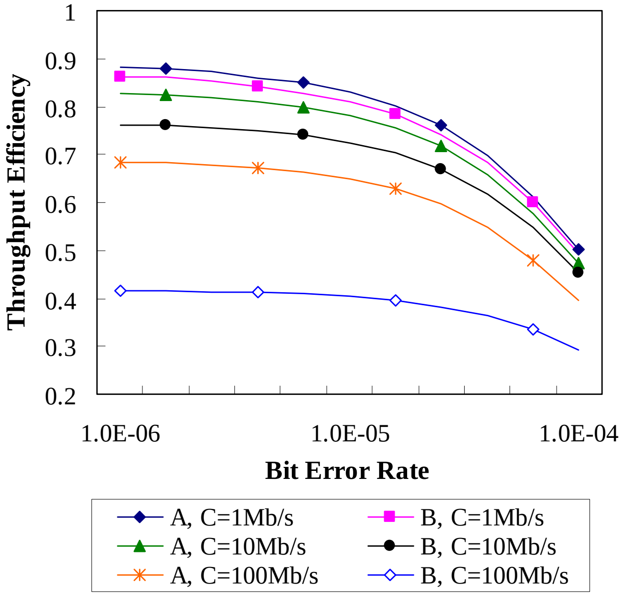

Figure 5: OBEX TPE using different P_{REQ} and T_{TA}.
Where A: P_{REQ}=50Kbit, T_{TA}=10^{-4}s; B: P_{REQ}=500Kbit, T_{TA}=10^{-3}s.

As shown in Fig.5, 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 adopting parameters A and B is shown for 100Mb/s links at BER=10^{-6}. Therefore appropriately adjusting the OBEX parameters may improve the system performance significantly for high data rates.

4. OBEX Parameters Optimization

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.

A. OBEX packet size:

The OBEX packet size P_{REQ} is a negotiable parameter for the connection. Because its value can be chosen from 2Kbits to 512Kbits [3], 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:

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

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:

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

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:

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

In equation (15), factor A is independent of P_{REQ}. By considering $N \cdot p >> p$ and $l \cdot (1-p) >> t_w$, we then have $l \cdot (1-p) \cdot N \cdot p >> p \cdot t_w$. 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}.

B. OBEX turnaround time:

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.6, the OBEX TPE as a function of OBEX packet size P_{REQ} in the range of $5\times10^4 \sim 1\times10^6 bit$ 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 Fig.7, the OBEX TPE is also plotted against T_{TA} in the range of $1\times10^{-5} \sim 1\times10^{-2} s$ at 1Mbps, 10Mbps and 100Mbps. P_{REQ} is set at 500Kbit.

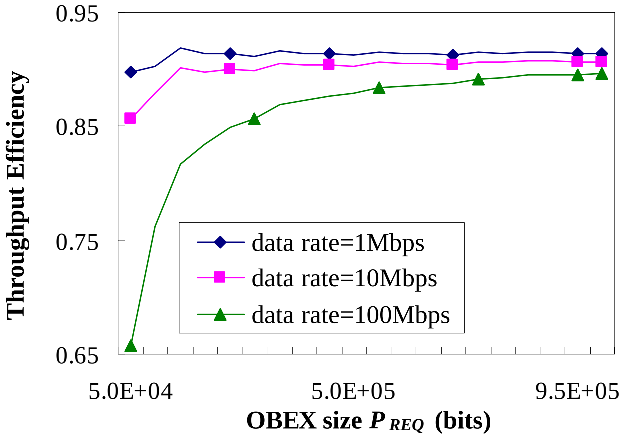

Figure 6: OBEX TPE against P_{REQ} with $T_{TA}=10^{-4}$s at BER=10^{-6}

For P_{REQ}, the corresponding OBEX TPE curves show non-linear shapes in Fig.6. 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}, in Fig.7, 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.

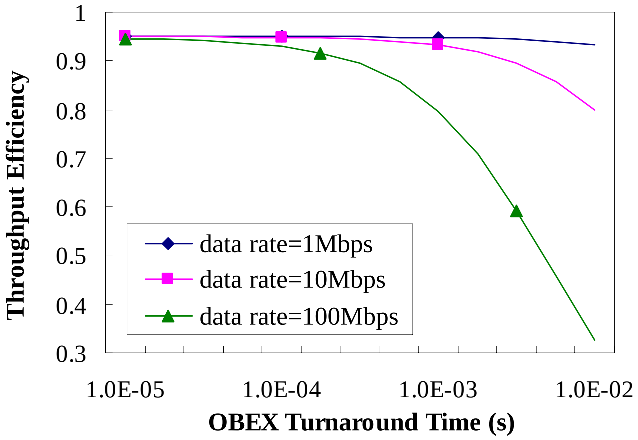

Figure 7: OBEX TPE against T_{TA} with P_{REQ}=500Kbit at BER=10^{-6}

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. As shown in Fig.6, high throughput efficiency over 0.85 is achieved when OBEX packet size P_{REQ} is increased to 500Kbit for 100Mbps links. P_{REQ} size of 200Kbit can satisfy 10Mbps and 1Mbps respectively, larger P_{REQ} is not necessary since it leads to only trivial improvement on the throughput.

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.7. For 100 Mbps links an OBEX turnaround time of less than 10^{-4} s is necessary, while for 10Mbps and less data rate links, an OBEX turnaround time of 10^{-3} s is sufficient if we are not prepared to sacrifice more than a few percent of throughput efficiency.

Fig.8 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.8 to the non-optimum cases in Fig.5, 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.

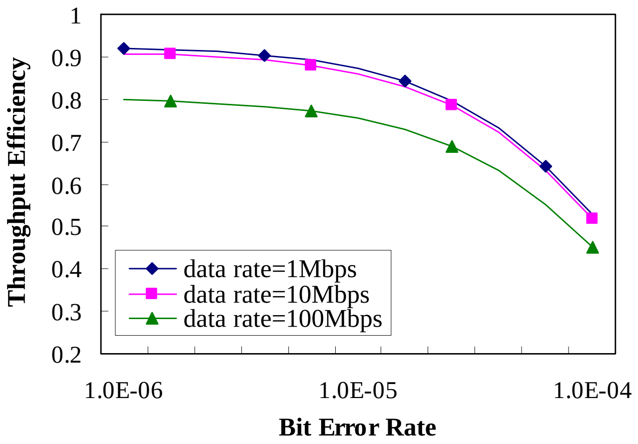

Figure 8: OBEX TPE using the recommended optimum P_{REQ} and T_{TA} values

5. Conclusions

This article examined the performance of OBEX protocol running on top of the IrDA protocol stack. We have used 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 the OBEX throughput efficiency. The non-optimum OBEX parameters show significant detriment on the system performance especially for high data rate links. In order to maximise the throughout and improve the system performance at high BER, the optimum OBEX size and OBEX turnaround time of the model have been studied. The analysis shows the system throughput always benefits by a large OBEX size and a smaller turnaround time. By considering small sacrifice off the system performance, we can compromise the throughput and the hardware requirement. Based on the results, we offer OBEX parameter selection guidelines of suitable OBEX packet size and turnaround time for different data rates. The OBEX throughput shows significant improvements by applying the optimized parameters.

REFERENCES

[1] Williams S., ‘IrDA: Past, Present and Future’, *IEEE Personal Communications*, 7(1), 11-19 (2000)

[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] IrDA, Object Exchange Protocol (IrOBEX), Version 1.3 (Infrared Data Association, March, 2003)

[4] A. C. Boucouvalas, V.Vitsas, '100 Mb/s IrDA Protocol Performance Evaluation", Proceedings of IASTED International Conference on Wireless and Optical Communications (WOC 2001) June 27-29, 2001 Banff, Canada, pp.49-57

[5] P. Chatzimisios and A. C. Boucouvalas "IrLAP IrDA Protocol Throughput Dependence on Processor Speed" International Symposium on CSNDSP 2002, Staffordshire University, July 15-17, pp. 272-275.

[6] A. C. Boucouvalas, V.Vitsas 'Optimum Window and Frame Size for IrDA Links'. IEE. Electronic Letters, Vol.37, No.3, pp194-196, 1st Feb. 2001

[7] IrDA, Serial Infrared Physical Layer Specification – Version 1.1, (Infrared Data Association, 1995)

[8] IrDA, Serial Infrared Physical Layer Specification for 16Mb/s Addition (VFIR)–Errata to version 1.3, Infrared Data Association, (1999)

[9] IrDA, Serial Infrared Link Access Protocol (IrLAP), Version 1.1 (Infrared Data Association, 1996)

[10] IrDA, Link Management Protocol (IrLMP), Version 1.1 (Infrared Data Association, 1996)

[11] IrDA, Tiny Transport Protocol (IrTinyTP), Version 1.1 (Infrared Data Association, 1996)

[12] 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.

[13] Bux W., Kummerle K. & Truong H.L., “Balanced HDLC procedures: A performance Analysis”, IEEE Trans.Com, 1980, com-28, pp.1889-1898.