EURASIP Journal on Applied Signal Processing 2005:1, 45–56
© 2005 Hindawi Publishing Corporation

Modelling and Optimising TinyTP over IrDA Stacks

A. C. Boucouvalas
Microelectronics and Multimedia Communications Research Centre, School of Design, Engineering and Computing, Bournemouth University, Fern Barrow, Poole, Dorset BH12 5BB, UK
Email: tboucouv@bournemouth.ac.uk

Pi Huang
Microelectronics and Multimedia Communications Research Centre, School of Design, Engineering and Computing, Bournemouth University, Fern Barrow, Poole, Dorset BH12 5BB, UK
Email: phuang@bournemouth.ac.uk

Received 29 March 2004; Revised 20 August 2004

TinyTP is the IrDA transport layer protocol for indoor infrared communications. For the first time, this paper presents a mathematical model for TinyTP over the IrDA protocol stacks taking into account the presence of bit errors. Based on this model, we carry out a comprehensive optimisation study to improve system performance at the transport layer. Four major parameters are optimised for maximum throughput including TinyTP receiver window, IrLAP window and frame size, as well as IrLAP turnaround time. Equations are derived for the optimum IrLAP window and frame sizes. Numerical results show that the system throughput is significantly improved by implementing the optimised parameters. The major contribution of this work is the modelling of TinyTP including the low-layer protocols and optimisation of the overall throughput by appropriate parameter selection.

Keywords and phrases: IrDA, TinyTP, IrLAP, transport layer protocol, optimisation.

1. INTRODUCTION

Indoor infrared data communications, based on the Infrared Data Association (IrDA) standards, have become widely available on a large number of portable devices ranging from mobile phones and digital cameras to laptops and printers [1]. Infrared communication is an excellent choice for effective, inexpensive and high-speed short-range wireless communications. The low-level IrDA protocols including physical (IrPHY) [2, 3], link access (IrLAP) [4], and link management (IrLMP) protocols [5] are adopted as industry standards and implemented on the products. Tiny transport protocol (TinyTP) is an optional IrDA layer, whereas it is so important and widely implemented that it is generally considered a required layer [6].

In [7], an IrLAP model is presented as the first significant work on the IrDA link layer. Subsequently, many link layer performance evaluations and improvements have also been undertaken recently to address different infrared link issues including the impact on link throughput of device processing speed [8] and future increase in data rates [9]. All the previous publications focus on link layer performance by assuming data always available of infinite size and a single application ready to transmit. However, upper layers (e.g., TinyTP) in practice offer finite-size packets to the link layer at specific time periods due to protocol behaviour and limited buffer size. TinyTP also allows multiple applications to operate the IrDA link concurrently. It is therefore of interest to examine the system throughput at TinyTP level.

The rest of this paper is organised as follows. First, we briefly describe the IrDA protocol stacks. Then, the details of TinyTP functionality are described. We subsequently develop a mathematical model for TinyTP which allows derivation of throughput taking into account the lower IrDA protocol stack. The TinyTP receiver window size and the IrLAP window and frame sizes are optimised for the maximum system throughput for any given bit error rate (BER). Finally, the suitable IrLAP turnaround time is investigated for 16 Mbps links.

2. IrDA PROTOCOL STACKS

The IrDA protocol stack illustrated in Figure 1 is the layered set of protocols particularly aimed at point-to-point infrared communications and the applications needed in that environment. A brief description of the IrDA protocol stack is as follows.

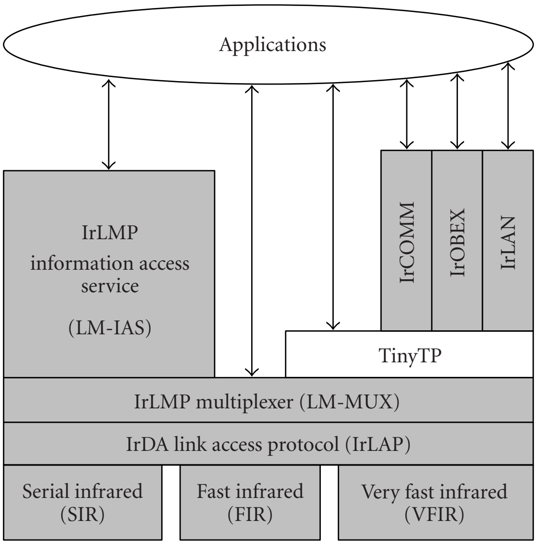

Applications
IrLMP information access service (LM-IAS)
IrCOMM
IrOBEX
IrLAN
TinyTP
IrLMP multiplexer (LM-MUX)
IrDA link access protocol (IrLAP)
Serial infrared (SIR)
Fast infrared (FIR)
Very fast infrared (VFIR)

Figure 1: IrDA protocol stacks.

2.1. IrDA physical layer

The IrDA physical layer defines a directed half-duplex serial infrared communications link established through free space to facilitate point-to-point communication. Framing data such as beginning- 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 16 Mbps have a 6-byte physical header for each IrLAP frame [2, 3].

2.2. IrDA link access protocol

IrLAP is the link access layer and it is based on the high-level data link control (HDLC) protocol. By using mechanisms including retransmission, low-level flow control and error detection, IrLAP provides reliable data transfer. IrLAP transmits data in the form of frames with maximum length of l_{LAP} and organises the transmission using go-back-N (GBN) error recovery. As the physical layer defines a half duplex link, IrLAP manages the transmission by assigning primary and secondary stations. The primary station initiates transfers to the secondary station and manages the link. When the primary completes the transmission of a window size N, information (I-) frames that can be sent before link turnaround, it then sets the poll (P) bit in the last I-frame to signal link turnaround and request the acknowledgement from the secondary. Once P bit is set, the secondary can start sending data. It changes P bit to 0 to turnaround the link when it finishes transmission. Referring to standards [3, 4], the window size and frame size range from 1 to 127 and from 128 bit to 16384 bit, respectively, IrLAP adds a 3-byte header to each frame.

2.3. 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 [5]. To realise the multiplexing, IrLMP assigns each application a unique link service access point (LSAP) address. IrLMP delivers upper-layer data segments based on the first-in first-out (FIFO) queuing [5]. We assume that the multiple application channels equally share the infrared link in this paper. After the connection initialisation, IrLMP adds a 2-byte header to the upper-layer packet providing the LSAP address for the sender and receiver.

3. TinyTP

TinyTP (TTP) is a light transport protocol serving as a flow control mechanism to work with IrLMP [6]. Even though IrLAP provides reliable data transfer, TinyTP is still important to ensure the end-to-end data delivery for the application. This is due to the possible deadlock problem of multiplexed channels introduced by IrLMP multiplexer (LM-MUX). Reliance on IrLAP to provide flow control for a multiplexed channel can result in deadlocks if the consumption of data from one multiplexed channel is dependent on data flowing in an adjacent multiplexed channel. Conversely, if inbound data on a multiplexed channel cannot be consumed and the underlying IrLAP connection cannot be flow controlled due to the possibility of deadlock, inbound data (freshly arrived or buffered) must be discarded in the event of buffer exhaustion. Unfortunately, this reduces the reliable delivery service provided by IrLAP to a best-effort delivery service provided by LM-MUX. To overcome this problem TinyTP provides two functions:

(i) segmentation and reassembly;
(ii) flow control on a per-LMP-connection (per-channel) basis.

For TinyTP, the entire data packet from upper layers can be segmented and reassembled in service data units (SDUs). The maximum SDU size is negotiated at the TinyTP/IrLMP connection establishment. One SDU has to fit within one IrLAP frame. Maximum TinyTP SDU size l_{TTP} therefore has to satisfy the condition $l_{\text{TTP}} \leq l_{\text{LAP}} - l'_{\text{LMP}} - l'_{\text{TTP}}$, where l'_{LMP} and l'_{TTP} stand for the headers of IrLMP and TinyTP. In this paper, we consider the challenge of having large application files to transmit. To make TinyTP efficient, we assume l_{TTP} is set at its maximum value $l_{\text{TTP}} = l_{\text{LAP}} - l'_{\text{LMP}} - l'_{\text{TTP}}$.

To perform flow control, TinyTP maintains a value of receiver window (w) for each TinyTP channel. The value of w is decided by the TinyTP buffer size of the communication peer. The sender will send SDU if $w > 0$ and subtract w by 1. Therefore, each TinyTP application can send maximum w SDU without receiving acknowledgement but it has to stop while $w = 0$. w is updated by the TinyTP acknowledgement from its peer. We assume every TinyTP application has the same value of w in this paper.

Each TinyTP service access point (TTPSAP) is accessible through one and only one LSAP of LM-MUX. A TTPSAP is identified by the address of the LSAP (provided in IrLMP header) through which it is accessed. After TinyTP

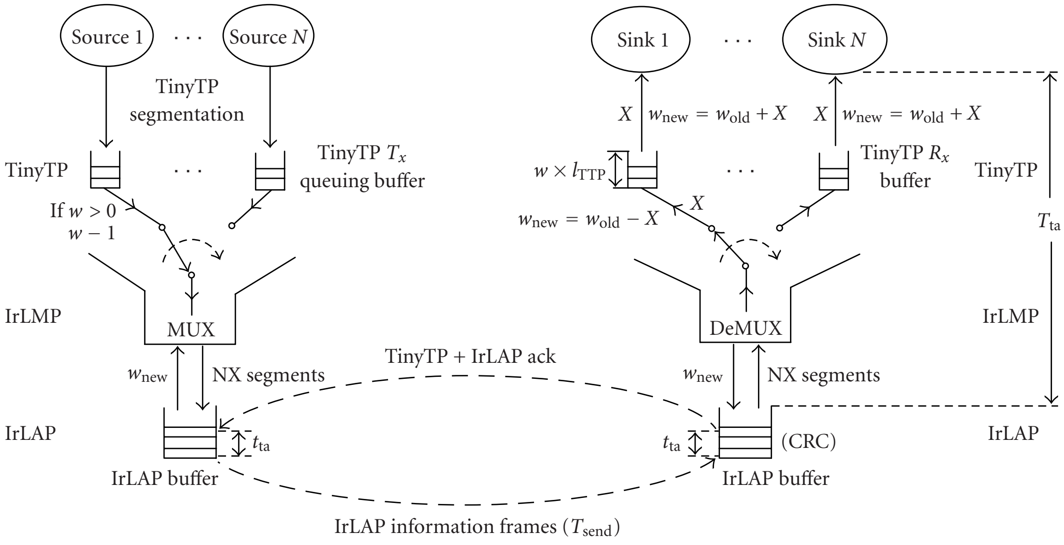


FIGURE 2: TinyTP data transmission.

TABLE 1: Parameters used in the modelling.

Symbol	Parameter description	Unit
C	Link data rate	bps
B	Number of TinyTP connections	—
p_b	Link BER	—
p	Frame error rate	—
N	Maximum IrLAP window size (number of frames)	—
w	TinyTP receiver window	—
l_{LAP}	Maximum IrLAP frame data length	bit
l_{TTP}	Maximum TinyTP segment size, $l_{\text{TTP}} = l_{\text{LAP}} - l'_{\text{LMP}} - l'_{\text{TTP}}$	bit
l'_{PHY}	PHY header: BOF + EOF + CRC	48 bits
l'_{LAP}	IrLAP header	24 bits
l'_{LMP}	IrLMP header	16 bits
l'_{TTP}	TinyTP header	8 bits
t_{I}	Transmission time of an information (I-)frame	s
t_{S}	Transmission time of a supervision (S-)frame	s
t_{ack}	Time to transmit an IrLAP acknowledgement packet	s
t_{ta}	IrLAP minimum turnaround time	s
t_{Fout}	IrLAP F-timer time-out period	s
T_{ta}	Time for TinyTP to process the received segments and prepare the acknowledgement	s

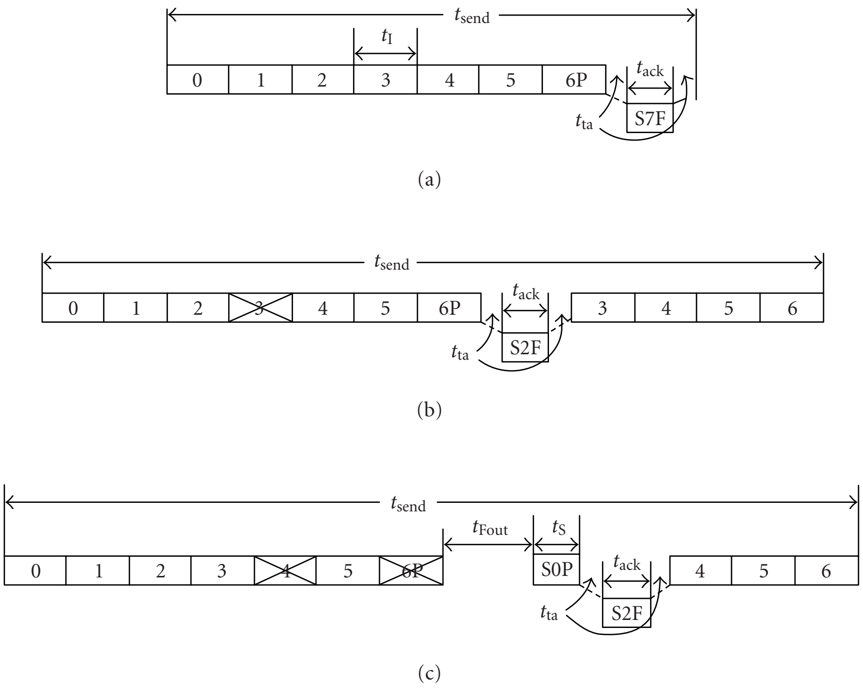
connection initialisation, TinyTP adds 1 byte of header carrying information including its own buffer size and the segmentation status. A flow chart of the data transmission with multiple TinyTP connections is provided in Figure 2.

4. MATHEMATICAL MODELLING

The mathematical model assumes large application files (e.g., mp3, movie clip) to be sent from the primary to the secondary. TinyTP segments therefore are always at the maximum size ($l_{\text{TTP}} = l_{\text{LAP}} - l'_{\text{LMP}} - l'_{\text{TTP}}$) to accommodate the application data. The "connected" TinyTP segments (excluding the connection establishment and termination segments) are considered in the derivation. Therefore, IrLMP and TinyTP have fixed headers of 2 bytes and 1 byte, respectively. We make use of Table 1 for symbol details.

4.1. IrLAP modelling

Correct IrLAP frame transmissions following an erroneous frame transmission in the same IrLAP window are considered out of sequence and have to be retransmitted (GBN).



(a)

(b)

(c)

Figure 3: (a) Error-free transmission of an IrLAP window, (b) retransmission frames due to error frame at frame 3, and (c) retransmitted frames and F-timer delay due to frame error at I=3 and I=7 (P-bit lost).

Based on the IrLAP model given in [7], in this section, we will derive the average time to successfully transmit one IrLAP window at a given BER. Due to the small size of the IrLAP supervision (S-) and acknowledgement (ack-) frames, we consider the IrLAP windows to be transmission error free. According to the IrLAP standard [4], IrLAP link parameters t_s, t_I, t_{ack}, p, and t_{Fout} are defined as follows:

$$t_s = \frac{l'_{\text{LAP}} + l'_{\text{PHY}}}{C}, \qquad t_I = \frac{l_{\text{LAP}} + l'_{\text{LAP}} + l'_{\text{PHY}}}{C}, \qquad t_{\text{ack}} = t_s,$$
$$p = 1 - (1 - p_b)^{l_{\text{LAP}} + l'_{\text{LAP}} + l'_{\text{PHY}}}, \qquad t_{\text{Fout}} = t_I + 2t_{\text{ta}}. \tag{1}$$

Both supervision and ack-frame have the same length which is the same as the physical and IrLAP header. If the last frame of the window is in error which causes P-bit loss, neither primary nor secondary is able to send data. F-timer t_{Fout} is the final bit timer used by the primary to limit the time it waits for a frame from the secondary. After t_{Fout} has expired, the primary will send a supervision frame to the secondary acknowledging the link turnaround. Figure 3 illustrates the IrLAP operation in detail.

The average time to successfully transmit one IrLAP window consists of the time for frame transmissions, acknowledgements and retransmissions, as well as delays for reversing the link t_{ta} and timer time outs t_{Fout}. As shown in Figure 3, the average time to transmit one IrLAP window with length of A frames is given as follows:

$$t_A = At_I + p(t_{\text{Fout}} + t_s) + t_{\text{ack}} + 2t_{\text{ta}}. \tag{2}$$

The probability of having error/errors in an IrLAP window with A frames is

$$p_1 = 1 - (1 - p)^A. \tag{3}$$

Due to the small value of p, p_1 can be approximated as

$$p_1 = 1 - (1 - p)^A \approx 1 - (1 - Ap) = Ap. \tag{4}$$

While error/errors occur in transmitting the IrLAP window with probability p_1, 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 trigger to recover the error with window length of $0.5A$. If further error/errors occur in the retransmission with probability of $p_2 = p_1(1 - (1 - p)^{0.5A}) \approx 0.5A^2p^2$, another retransmission window is needed with window length of half the previous, that is, $0.25A$, and so on. When the retransmission window is less than 1, we consider the whole window has been successfully transmitted. By including the first window transmission and all the retransmissions, the average time to successfully transmit the IrLAP window is given:

$$\begin{aligned}
T_{\text{send}}(A) &= t_A + p_1\left(\frac{1}{2}At_I + p(t_{\text{Fout}} + t_s) + t_{\text{ack}} + 2t_{\text{ta}}\right)\\
&\quad + \cdots + p_X\left(\frac{1}{2^X}At_I + p(t_{\text{Fout}} + t_s) + t_{\text{ack}} + 2t_{\text{ta}}\right)\\
&= \left(1 + \frac{1}{2}Ap + \cdots + \left(\frac{1}{2}\right)^{(1/2)X(X+1)}A^X p^X\right)At_I\\
&\quad + \left(1 + Ap + \cdots + \left(\frac{1}{2}\right)^{(1/2)X(X-1)}A^X p^X\right)\\
&\quad \times \left(p(t_{\text{Fout}} + t_s) + t_{\text{ack}} + 2t_{\text{ta}}\right)\\
&= \left(1 + \sum_{i=1}^{A}\left(\left(\frac{1}{2}\right)^{(1/2)i(i+1)}(Ap)^i\right)\right)At_I\\
&\quad + \left(1 + Ap + \sum_{i=2}^{X}\left(\left(\frac{1}{2}\right)^{(1/2)i(i-1)}(Ap)^i\right)\right)\\
&\quad \times \left(p(t_{\text{Fout}} + t_s) + t_{\text{ack}} + 2t_{\text{ta}}\right),
\end{aligned} \tag{5}$$

where X is an integer representing the number of retransmissions ($X = \lceil \log_2 A \rceil$). X satisfies the length of the retransmission window to be no less than 1 ($1/2^X \cdot A \leq 1$).

4.2. Derivation of TinyTP throughput

Before deriving TinyTP throughput, we first discuss two TinyTP parameters T_{ack} and T_{ta}. According to the standard [6], TinyTP acknowledgement needs only to provide the updated secondary receiver window size. Therefore, the secondary simply sends the TinyTP header l'_{TTP} as the TinyTP ack. By including the headers of the other layers, the transmission time of the TinyTP acknowledgement is given by

$$T_{\text{ack}} = \frac{l'_{\text{Phy}} + l'_{\text{LAP}} + l'_{\text{LMP}} + l'_{\text{TTP}}}{C}. \tag{6}$$

The time to hold the TinyTP segments in the buffer (T_{ta}) is the time from passing the IrLAP frames to IrLMP to the time the TinyTP gets acknowledgement ready at the secondary. As shown in Figure 2, T_{ta} includes the time to process and strip the headers all the way up to TinyTP, the time to process the TinyTP segments and drain the TinyTP buffer, as well as preparing TinyTP ack and adding the headers of other layers. For different applications, the time to process the TinyTP segments (T_p) is different. In this paper, we assume that the IrDA device uses 8-bit processor and each 8-bit data takes average 2-CPU cycles. As T_p is the major factor of T_{ta}, we assume that $T_{\text{ta}} \approx T_p$:

$$T_{\text{ta}} \approx T_p = \frac{2Al_{\text{TTP}}}{8v} = \frac{Al_{\text{TTP}}}{4v}, \tag{7}$$

where A is the incoming IrLAP window size and v is the processor speed in Hz.

When a TinyTP receiver window size w is allocated for each B TinyTP connections, the IrDA receiver has to assign a TinyTP buffer with size of $B \times w \times l_{\text{TTP}}$. Given the fact that memory is highly constrained for resource-limited wireless device, such devices often cannot afford large memory size for TinyTP. For a given maximum IrLAP window size N, three possible scenarios by implementing different receiver window size are investigated as follows, where B denotes the number of TinyTP connections. We assume the TinyTP connections equally share the link as IrLMP delivers data based on FIFO queuing.

4.2.1. $Bw \leq N$

The TinyTP transmission model is illustrated in Figure 4 by mapping TinyTP segments into IrLAP frames. In Figure 4, parameters $w = 2$, $B = 2$, and $N \geq 4$ are employed which satisfy $Bw \leq N$. The IrLAP window will be always less than four due to the w constraint. As the time to prepare the TinyTP acknowledgement packet T_{ta} depends on the CPU speed of the receiver, it is normally much longer than IrDA link turnaround t_{ta} and the time to transmit the IrLAP ack packet. Thus, it is sufficient to assume $T_{\text{ta}} > t_{\text{ta}} + t_{\text{ack}}$. After IrLAP successfully delivers the IrLAP frames, the secondary has to wait T_{ta} before the TinyTP acks get ready. Since two TinyTP connections are considered, the secondary needs to send two TinyTP acks. Then, following the same routine another window will be sent from the primary. Therefore, we only need to consider one window transmission for the TinyTP throughput derivation.

As shown in Figure 4, and using (5), the average time for one TinyTP window transmission T_1 is given by

$$T_1 = T_{\text{send}}(Bw) + T_{\text{ta}} + BT_{\text{ack}} + t_{\text{ta}}, \tag{8}$$

where w is the receiver window size and B is the number of TinyTP connections.

The TinyTP throughput which is defined as information bits per second is

$$D = \frac{Bw \times l_{\text{TTP}}}{T_1}. \tag{9}$$

4.2.2. $N < Bw < 2N$

The TinyTP transmission model is illustrated in Figure 5. In Figure 5, $w = 3$, $N = 4$, and $B = 2$ are used, which satisfy $N < Bw < 2N$. The first TinyTP window has 4 segments and makes use of maximum IrLAP window length. Since the secondary is fed by 4 TinyTP segments and has no time to process, the secondary will send 2 TinyTP acks to give the information of available buffer size for each application. In this case, the secondary acknowledges the primary with $w1 = w2 = 1$ (available buffer size subtracted from the in-

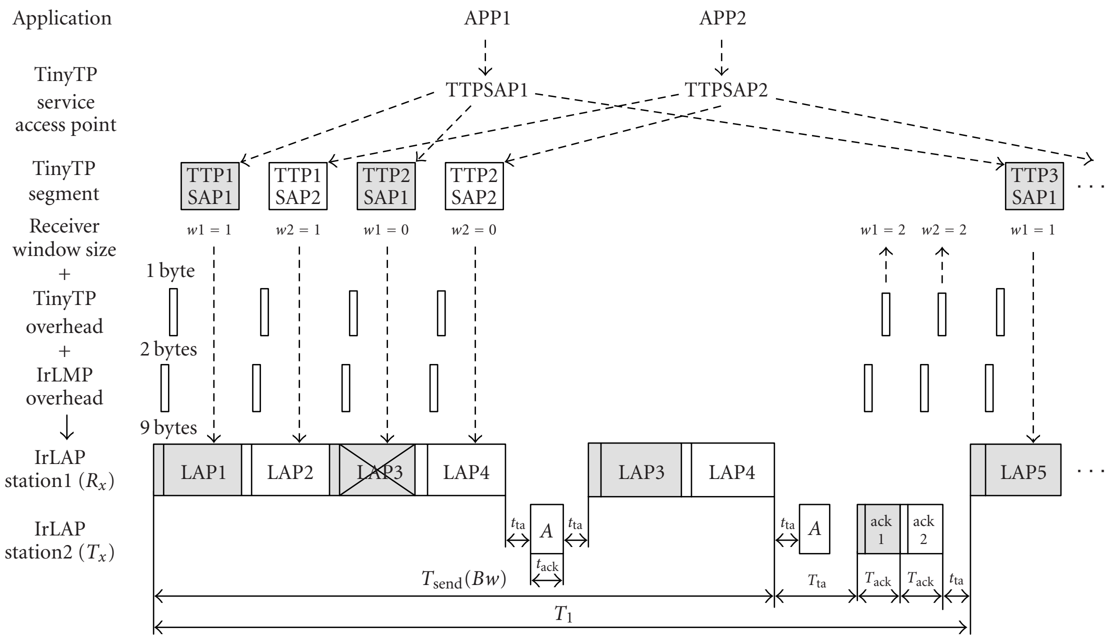


FIGURE 4: TinyTP transmission model when $Bw \leq N$, initial state: $w1 = w2 = 2$, where A is the IrLAP acknowledgement and ack is the TinyTP acknowledgement.

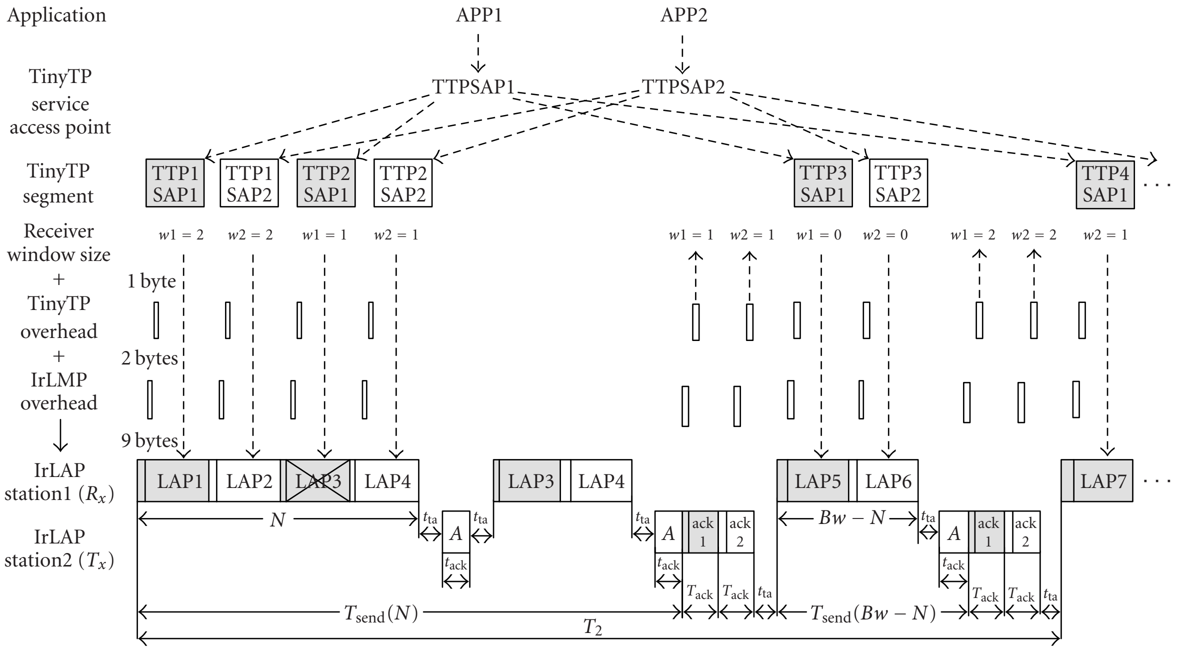


FIGURE 5: TinyTP transmission model when $N < Bw < 2N$, initial state: $w1 = w2 = 3$.

coming segments, $3-2 = 1$). The primary is then able to send 2 segments in the second window. Assuming the 4 TinyTP segments of the last window have been processed and consumed, each of the receiver window equals to 2 ($3 - 1 = 2$). The secondary then acknowledges with $w1 = w2 = 2$. As the same process will be repeated, we only need to consider two window transmissions for deriving the TinyTP throughput.

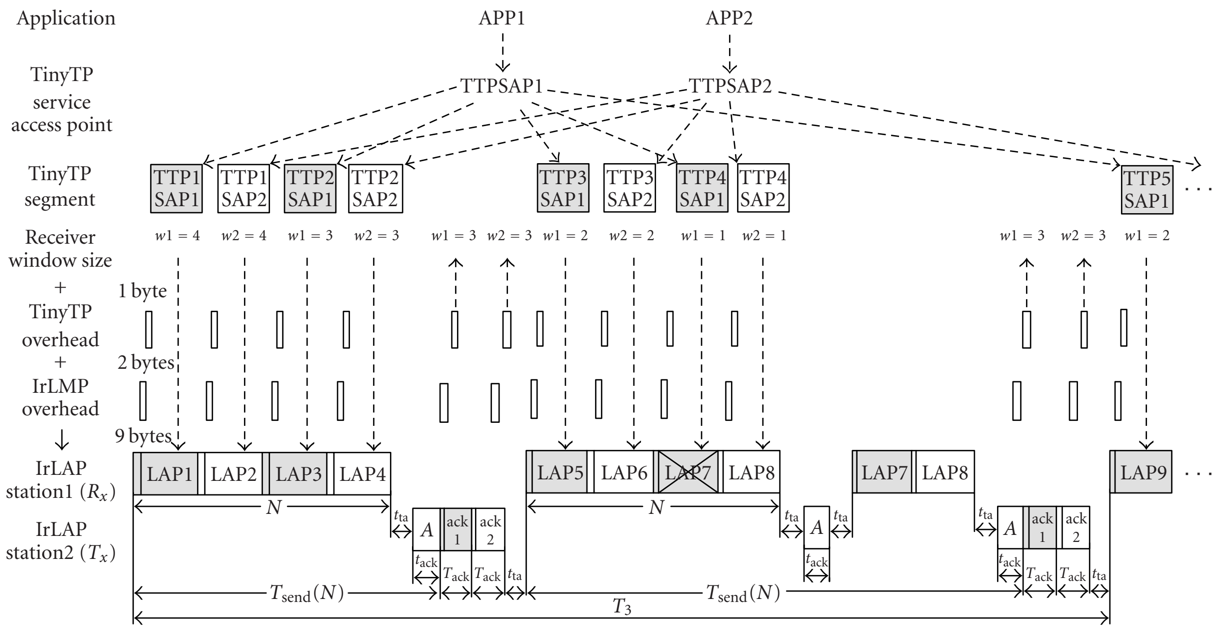


FIGURE 6: TinyTP transmission model when $w \geq 2N$, initial state: $w1 = w2 = 5$.

The average transmission times for the first and the second IrLAP windows is

$$
\begin{aligned}
T_{\text{send}}(N) = {} & \left(1+\sum_{i=1}^{X}\left(\left(\frac{1}{2}\right)^{(1/2)i(i+1)}(Np)^{i}\right)\right)Nt_{I} \\
& + \left(1+Np+\sum_{i=2}^{X}\left(\left(\frac{1}{2}\right)^{(1/2)i(i-1)}(Np)^{i}\right)\right) \\
& \times \left(p\left(t_{\text{Fout}}+t_{s}\right)+t_{\text{ack}}+2t_{\text{ta}}\right), \\
T_{\text{send}}&(w-N) \\
= {} & \left(1+\sum_{i=1}^{X}\left(\left(\frac{1}{2}\right)^{(1/2)i(i+1)}((w-N)p)^{i}\right)\right)(w-N)t_{I} \\
& + \left(1+(w-N)p+\sum_{i=2}^{X}\left(\left(\frac{1}{2}\right)^{(1/2)i(i-1)}((w-N)p)^{i}\right)\right) \\
& \times \left(p\left(t_{\text{Fout}}+t_{s}\right)+t_{\text{ack}}+2t_{\text{ta}}\right).
\end{aligned}
\tag{10}
$$

With the aid of Figure 5 and by using (5), the average time for one TinyTP window transmission T_2 is

$$
\begin{aligned}
T_2 &= T_{\text{send}}(N) + BT_{\text{ack}} + t_{\text{ta}} + T_{\text{send}}(Bw-N) + BT_{\text{ack}} + t_{\text{ta}} \\
&= T_{\text{send}}(N) + T_{\text{send}}(Bw-N) + 2BT_{\text{ack}} + 2t_{\text{ta}}.
\end{aligned}
\tag{11}
$$

TinyTP throughput is

$$
D = \frac{Bw \times l_{\text{TTP}}}{T_2}. \tag{12}
$$

4.2.3. $Bw \geq 2N$

The TinyTP transmission model in this case is illustrated in Figure 6. In Figure 6, $w = 5$, $N = 4$, and $B = 2$ are used, which satisfy $Bw \geq 2N$. The first TinyTP window has 4 segments which make use of maximum IrLAP window length. The secondary acknowledges with $w1 = w2 = 3$ (available buffer size subtracted from the incoming segments, $5 - 2 = 3$). The primary is then allowed to send another 4 segments in the second window. Assuming the TinyTP segments of last window have been processed and consumed, the secondary then acknowledges with $w1 = w2 = 3$ $(5 - 2 = 3)$, and so on. Therefore, we only need to consider one window transmission for deriving the TinyTP throughput.

From Figure 6, each of the IrLAP windows has a length of N. The average transmission time for the first and the second IrLAP windows is

$$
T_3 = T_{\text{send}}(N) + BT_{\text{ack}} + t_{\text{ta}}. \tag{13}
$$

The TinyTP throughput is given by

$$
D = \frac{Nl_{\text{TTP}}}{T_3}. \tag{14}
$$

TinyTP throughput efficiency (TPE) is given by

$$
\text{TPE} = \frac{D}{C}. \tag{15}
$$

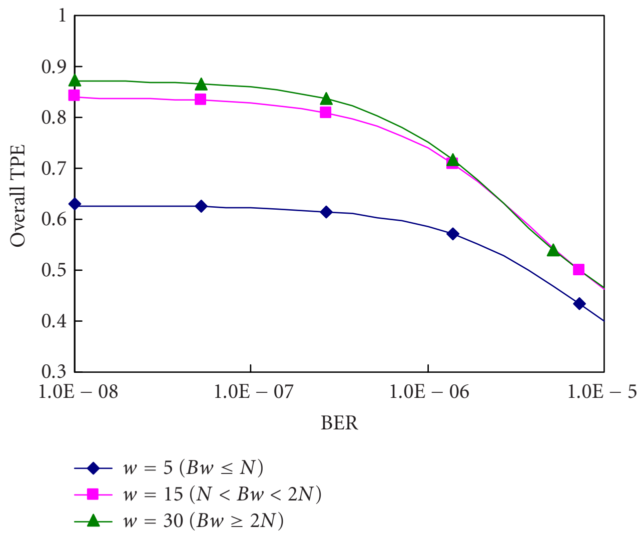


FIGURE 7: Overall TinyTP throughput efficiency comparison using different receiver window sizes w. Overall TinyTP throughput is the aggregate throughput of B channels, and IrLAP window $N = 20$.

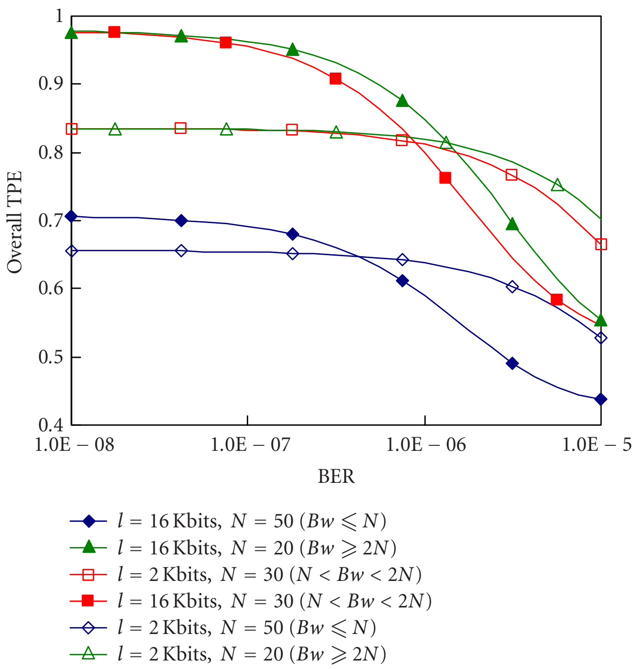


FIGURE 8: TinyTP TPE comparison using different receiver window sizes w ($w = 20$).

5. TinyTP THROUGHPUT ANALYSIS

Equations (9), (12), and (14) reveal the parameters TinyTP throughput depends on. In this section, in order to provide a suitable design guideline for IrDA devices, we carry out an inclusive throughput analysis. We compare the throughput by implementing different TinyTP buffer sizes for various BERs. Subsequently, we examine the effect of IrLAP turnaround time on the throughput. Finally, we investigate the effect of the processor speed.

5.1. *Effect of TinyTP receiver window size* (w)

In Figure 7, TinyTP TPEs are compared by implementing different receiver window sizes w. We fix the maximum IrLAP window and frame size at $N = 20$ and $l = 16$ kbit, respectively. The following parameters are used for this figure: $C = 16$ Mbps, $v = 10$ MHz, $t_{\text{ta}} = 10^{-4}$ second, and $B = 2$. Unless otherwise specified, the same values of C, v, t_{ta}, and B are implemented throughout this paper. The throughput efficiencies are plotted against the BER in the range of 10^{-5} to 10^{-8}.

As shown in Figure 7, all of the three TPE deteriorate with the increase in the BER. In the case of $w = 15$ and 30, the system obtains much better TPEs than when $w = 5$ especially for low BER (TPE > 0.8). The TPE for $w = 30$ is slightly better than for $w = 15$. The graph shows that the system achieves the best throughput for any BER by using a receiver window size at least twice the maximum IrLAP window size ($Bw \geq 2N$). However, a good TinyTP throughput level is also reached by using $w = 15$ ($N < Bw < 2N$). Therefore, a receiver window size in the range of $N < Bw < 2N$ obtains good system performance as well as requiring relatively smaller buffer size.

5.2. *Effect of IrLAP window size* (N) *and frame size* (l)

In Figure 8, TinyTP TPEs are compared by implementing different IrLAP windows and frame sizes. The TinyTP receiver window size is set to $w = 20$. Throughput efficiency is plotted against the BER in the range of 10^{-5} to 10^{-8}. All the TPE curves decrease with the increasing BER. The system achieves better TPE by using large frame size ($l = 16$ Kbit) at low BER, however, at the high BER, the system obtains better TPE by implementing small frame size ($l = 2$ Kbit). For the same window size, the crossing points of the two curves that represent different frame size l implying a better throughput may be achieved by appropriately adjusting of window and frame sizes.

5.3. *Effect of processor speed* (v)

We assume here that the TinyTP segments in the previous window have been processed for the case of $N < Bw < 2N$ and $Bw \geq 2N$. This assumption holds true when the extreme condition $T_{\text{ta}} \leq 3t_{\text{ta}} + 2t_{\text{ack}} + BT_{\text{ack}} + t_I$ is satisfied. To fulfill this condition, processor speed has to be at least as fast as $CNl_{\text{TTP}}/4(3Ct_{\text{ta}} + 2Ct_{\text{ack}} + CBT_{\text{ack}} + l_{\text{LAP}})$. For instance, for a 1– Mbps IrDA link with $N = 4$, $l_{\text{LAP}} = 16$ kbit, $t_{\text{ta}} = 10^{-3}$ s, and $B = 2$, processor speed v has to be at least 0.8 MHz to satisfy the condition. If $v < CNl_{\text{TTP}}/4(3Ct_{\text{ta}} + 2Ct_{\text{ack}} + CBT_{\text{ack}} + l_{\text{LAP}})$, TinyTP throughput will be deteriorated due to the extra time needed to wait for the TinyTP segment processing. Based on our TinyTP model, Figure 9 shows the effect of processing speed when $Bw \leq N$. We obtain the result by using the following parameters: $N = 20$, $l = 16$ Kbit, and $w = 5$. The TPEs are plotted in three different BERs against the processor speed in the range of 10^5 to 10^8 Hz.

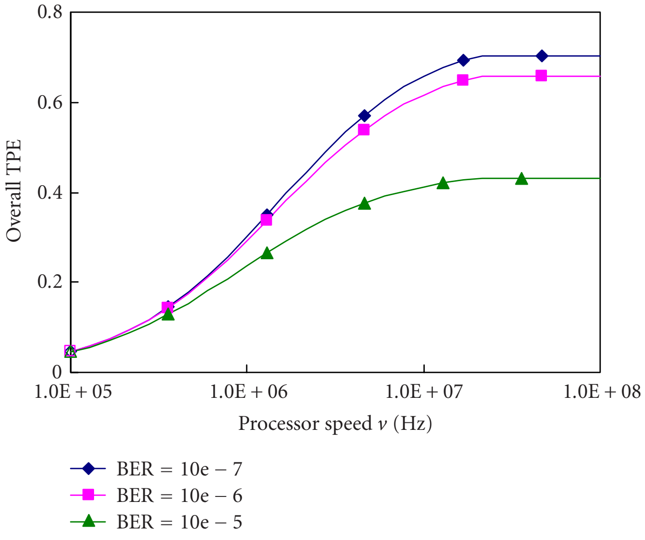


Figure 9: Effect of processor speed on TPE when $Bw \leq N$ in different BER ($w = 5$ and $N = 20$).

All three TPE curves increase with the processor speed until saturation for $v > 10^7$ Hz $= 10$ MHz. Because T_{ta} becomes larger than $t_{\text{ta}} + t_{\text{ack}}$ when $v > 10$ MHz, the secondary will wait for $t_{\text{ta}} + t_{\text{ack}}$ instead of T_{ta} before sending the TinyTP acks. Therefore, the processor speed will not benefit from the system throughput when $v > 10$ MHz. However, the TPE increases significantly with the processor speed up to 10 MHz. Therefore, $v = 10$ MHz is a suitable processing speed for the 16 Mbps IrDA links.

6. THROUGHPUT OPTIMISATIONS

6.1. Optimum TinyTP receiver window size w

As shown in Figure 7, the system achieves its best performance when $Bw \geq 2N$ because it takes full advantage of the IrLAP maximum window size. However, the system will reach the same throughput as $Bw = 2N$ when $Bw > 2N$. Because the throughput is also constrained by the IrLAP window size N, it will not be improved by a receiver window size larger than $2N$. Therefore, a receiver window size of $w = 2N$ can always achieve the best throughput even only one TinyTP connection is running. As shown in Figure 7, good TinyTP throughput is also obtained by using a receiver window size of $N < Bw < 2N$. For the memory scarce devices, in order to improve system performance and resource requirement, TinyTP can use a receiver window size in the range of $N < Bw < 2N$, as this range achieves good throughput as well as requiring relatively small buffer size.

6.2. Optimum IrLAP window size N and frame size l_{LAP}

As shown in Figure 8, if IrLAP window and frame sizes can be optimised, we can achieve better throughput at the TinyTP level. In [10, 11], optimisation equations of the IrLAP parameters are presented to maximise the IrLAP throughput. However, when considering TinyTP performance optimisation, due to the constraint of receiver window size, the optimisations at IrLAP level are not suitable for TinyTP. In order to maximise the TinyTP throughput for any given receiver window size and BER, IrLAP parameters are optimised in this section. In situations where it is convenient to optimise only one variable, either N or l_{LAP}, we obtain maximum TinyTP throughput by using optimum values for l_{LAP} or N, respectively. However the best TinyTP throughput would be obtained when both N and l_{LAP} are simultaneously optimised with BER.

6.2.1. Optimum window or frame size for maximum TinyTP throughput

6.2.1.1. $Bw \leq N$

Because each TinyTP connection cannot send more than w segments before receiving an acknowledgement, IrLAP window size always equals Bw, as shown in Figure 4. Therefore, in this case, we only need to optimise IrLAP frame size l_{LAP} for a fixed N with value of Bw.

By calculating $\partial D/\partial l_{\text{LAP}} = 0$ for (9), which is a function of l_{LAP}, the optimum value of l_{LAP} for any fixed N is derived. After some calculations and careful approximations, the optimum equation for l_{LAP} is derived:

$$l_{\text{opt}} = \frac{1}{Bw}\sqrt{\frac{2(t_{\text{ack}} + BT_{\text{ack}} + 3t_{\text{ta}})C}{p_b}}. \tag{16}$$

6.2.1.2. $N < Bw < 2N$

In this case, both N and l_{LAP} are adjustable. N is limited in the range from N to $2N$. It is possible to maximise throughput by fixing either N or l_{LAP} and optimising the other. By taking $\partial D/\partial N = 0$ for (12), the optimum value of N for any fixed l_{LAP} is derived. Also, for fixed N, optimum l_{LAP} value is derived by taking the derivative of D, $\partial D/\partial l_{\text{LAP}} = 0$. After some calculus and approximations, the optimum equations for N and l_{LAP} are given by

$$N_{\text{opt}} = \frac{Bw}{2}, \tag{17}$$

$$l_{\text{opt}} = 2\sqrt{\frac{(t_{\text{ack}} + 3t_{\text{ta}} + BT_{\text{ack}})C}{(2N^2 - 2BwN + B^2w^2)p_b}}. \tag{18}$$

6.2.1.3. $Bw \geq 2N$

By using the same approach as given above, optimum equations for N and l_{LAP} are given by

$$N_{\text{opt}} = \sqrt{\frac{2(t_{\text{ack}} + 3t_{\text{ta}} + BT_{\text{ack}})C}{(l + l')^2 p_b}}, \tag{19}$$

$$l_{\text{opt}} = \sqrt{\frac{2(Nl' + (t_{\text{ack}} + 3t_{\text{ta}} + BT_{\text{ack}})C)}{N^2 p_b}}, \tag{20}$$

where $l' = l'_{\text{PHY}} + l'_{\text{LAP}}$.

6.2.2. *Simultaneous optimum window and frame size for maximum TinyTP throughput*

In this case, both window and frame size can be simultaneously adjusted. The maximum possible throughput performance can be achieved. In order to derive optimum N and l values, first we fix l_{LAP} to derive optimum N by taking $\partial D/\partial N = 0$. Then, the derived optimum N equation (l_{LAP} dependent) is substituted into the throughput equation. Throughput D becomes a function of frame size l_{LAP} for optimum N values. By taking $\partial D/\partial l_{\text{LAP}} = 0$, optimum l_{LAP} equation is derived. This essentially derives the conditions for $\partial D_b/\partial N = \partial D_b/\partial l_{\text{LAP}} = 0$. Finally, by substituting optimum l_{LAP} equation back to optimum N equation (l_{LAP} dependent), we can derive the optimum equation N.

6.2.2.1. *Bw* ≤ *N*

As described in Section 6.2.1.1, IrLAP window size is fixed at Bw. Therefore, only IrLAP frame size l_{LAP} is needed to optimise for throughput which is given in (16).

6.2.2.2. *N* < *Bw* < 2*N*

In 6.2.1.2, optimum N in equation (17) is already l_{LAP} independent. Therefore, we only need to substitute (17) into the throughput equation (12) to derive optimum l_{LAP}. Optimum N and l_{LAP} in this case are given by

$$N_{\text{opt}} = \frac{Bw}{2},$$
$$l_{\text{opt}} = \frac{2}{Bw}\sqrt{\frac{(t_{\text{ack}} + 3t_{\text{ta}} + BT_{\text{ack}})C}{p_b}}. \tag{21}$$

6.2.2.3. *Bw* ≥ 2*N*

Finally, simultaneously optimum N and l_{LAP} equation are given by

$$N_{\text{opt}} = \sqrt{\frac{2(t_{\text{ack}} + 3t_{\text{ta}} + BT_{\text{ack}})CYp_b}{4l' + 4l'p_bY + p_b^2Yl'^2 - p_b^2Y^2l'}}, \tag{22}$$

$$l_{\text{opt}} = \sqrt{\frac{4l' + 4l'p_bY + p_b^2Yl'^2 - p_b^2Y^2l'}{Yp_b^2}}, \tag{23}$$

where $Y = \sqrt{2(t_{\text{ack}} + 3t_{\text{ta}} + BT_{\text{ack}})C/p_b}$.

For low BER, l_{opt} should be very large and takes values larger than 16 kbits, values not allowed by IrDA specification [4]. In practice, therefore, it is restricted to using both approaches for optimum IrLAP parameters. We use approach 6.2.1.3, (20), to obtain N_{opt} in low BER for fixed maximum value $l = 16$ Kbits, until the calculated l_{opt} is less than 16 Kbits ($\sim$ BER $= 5.7 * 10^6$ from (22) using $B = 2$, $v = 10$ Mz and $t_{\text{ta}} = 10^{-4}$ s). Thereafter for higher BER, approach 6.2.2.3, (22) and (23), is implemented to obtain optimum throughput.

In order to examine the accuracy of the optimum equations derived in this section, we compare the results obtained from the equations with the results obtained from exact numerical methods. In Figure 10, the overall TPE is plotted against BER in the range from 10^{-5} to 10^{-8} by implementing the simultaneously optimised N and l. The corresponding optimised IrLAP window and frame size used are plotted in Figure 11.

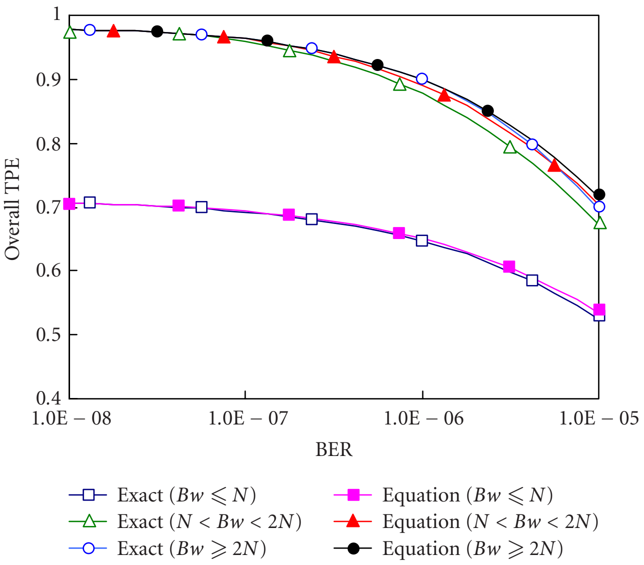


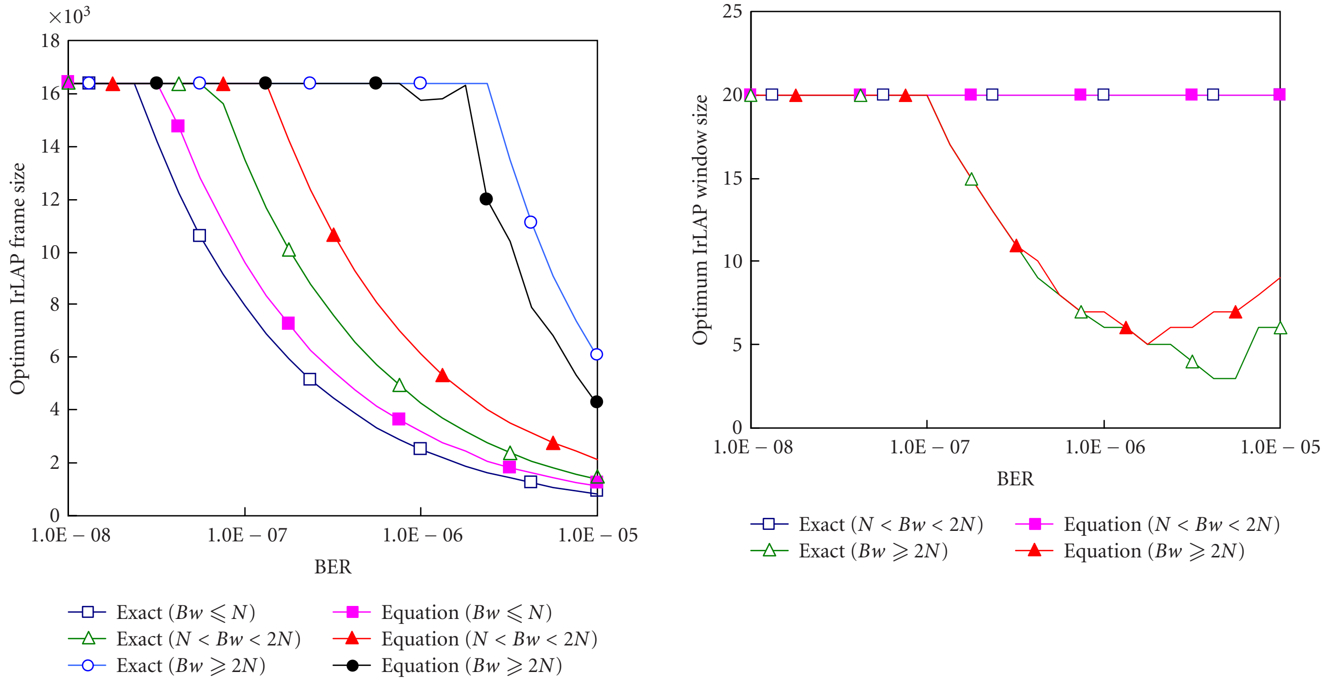
Figure 10: TinyTP TPE using optimum IrLAP window and frame size, and TinyTP receiver window $w = 20$.

As shown in Figure 11, the optimum frame sizes l_{opt} are fixed at 16 Kbit in the low BER and then drop down significantly with the increasing BER. The exact and equation approaches for the optimum values have only small differences. For all of the three cases, the curves representing two different approaches follow the same shape and are very close to each other. The optimum window sizes N_{opt} have exactly the same values of 20 for either exact or using the equation results when $N < Bw < 2N$. N_{opt} also shows very good agreement for $Bw \geq 2N$, especially in the low BER.

In Figure 10, the throughput efficiencies gradually decrease when the BER increases. Comparing the optimum TPE results obtained from the equations with results obtained from exact numerical methods, they show very good agreement for all three cases. Moreover, the system always acquires its optimum throughput in the case of $Bw > 2N$. Therefore, for a given size of TinyTP receiver window, N_{opt} should always satisfy the condition $Bw > 2N$ and be calculated from (19) and (22) for the corresponding BER. A comparison between Figures 8 and 10 shows that optimisations of IrLAP window and frame size are necessary since the performance is significantly improved at TinyTP level when the optimum values are used.

6.3. *Optimum IrLAP turnaround time*

In order to examine the effect of IrLAP turnaround time, optimum N and l are considered. As shown in the previous section, N_{opt} should always satisfy the condition $Bw > 2N$ for



(a)

(b)

Figure 11: The corresponding optimum IrLAP window and frame size to Figure 10.

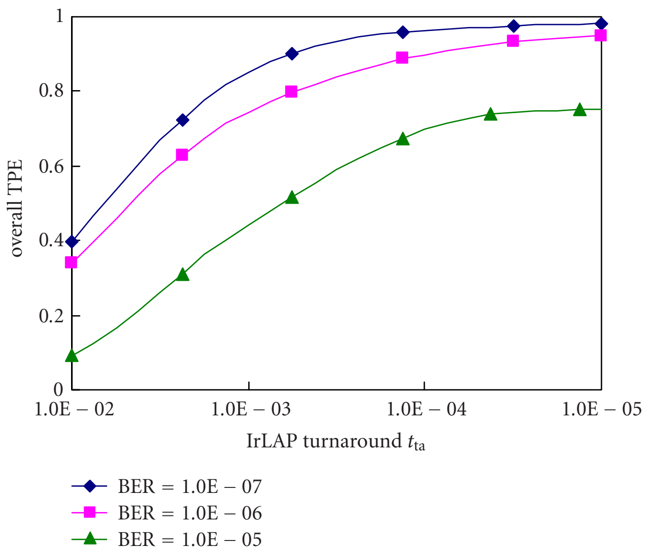


Figure 12: Effect of IrLAP turnaround time on TinyTP TPE, $w = 20$, optimum N and l.

the maximum throughput. Therefore, we only need to consider the case of $Bw > 2N$. In Figure 12, TinyTP TPEs are plotted against IrLAP turnaround time in the range of 10^{-5} to 10^{-2} second. The TPEs are compared by implementing three different BERs. The receiver window size is fixed at 20. As shown in Figure 12, the TPEs for low BER are better than in the high BER for the same IrLAP turnaround time. With t_{ta} in the range of $t_{ta} < 10^{-4}\,\text{s} = 0.1$ millisecond, the three TPEs only increase slightly. However, the TPEs decrease significantly when t_{ta} increases above 0.1 millisecond. By considering the system performance and hardware requirement tradeoff, it can be seen that an IrLAP turnaround time at the level of 10^{-4} second is a suitable parameter for the 16- Mbit IrDA links.

7. CONCLUSIONS

In this paper, we derive a comprehensive model for the IrDA TinyTP performance in the presence of BER by considering multiple IrLMP connections and taking the underlying IrDA protocol stacks into account. Based on the model, the throughput efficiencies are compared by implementing different receiver window size, and IrLAP window and frame sizes. The results show that the system always achieves its best performance when $Bw \geq 2N$ and can be maximised by optimising IrLAP window and frame sizes for any given BER. Subsequently, an inclusive study for parameter optimisations is carried out for the system. Several optimisation equations for IrLAP window and frame sizes are derived and later validated in the simulations. Finally, the effect of IrLAP turnaround time on the throughput is examined. An IrLAP turnaround time of the order in 10^{-4} second is a suitable parameter for 16− Mbps links. By comparing the results, we can see that significant improvements on the throughput are achieved by applying the optimised parameters.

REFERENCES

[1] S. Williams, "IrDA: past, present and future," *IEEE Pers. Commun.*, vol. 7, no. 1, pp. 11–19, 2000.

[2] IrDA, "Serial infrared physical layer specification," version 1.1, October 1995.

[3] IrDA, "Serial infrared physical layer specification for 16 Mb/s addition (VFIR)," errata to version 1.3, January 1999.

[4] IrDA, "Serial infrared link access protocol," version 1.1, June 1996.

[5] IrDA, "Serial infrared link management protocol," version 1.1, January 1996.

[6] IrDA, "Tiny transport protocol," version 1.1, October 1996.

[7] P. Barker, A. C. Boucouvalas, and V. Vitsas, "Performance modelling of the IrDA infrared wireless communications protocol," *International Journal of Communications Systems*, vol. 13, no. 7-8, pp. 589–604, 2000.

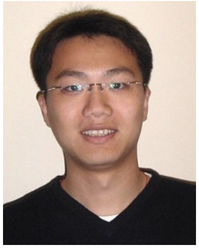
[8] P. Chatzimisios and A. C. Boucouvalas, "IrLAP IrDA protocol throughput dependence on processor speed," in *Proc. International Symposium on CSNDSP*, pp. 272–275, Staffordshire University, Staffordshire, UK, July 2002.

[9] A. C. Boucouvalas and V. Vitsas, "100 Mb/s IrDA protocol performance evaluation," in *Proc. International Conference on Wireless and Optical Communications (WOC '01)*, pp. 49–57, Banff, Canada, June 2001.

[10] V. Vitsas and A. C. Boucouvalas, "Simultaneous optimisation of window and frame size for maximum throughput IrDA links," *Electronics Letters*, vol. 37, no. 16, pp. 1042–1043, 2001.

[11] V. Vitsas and A. C. Boucouvalas, "Optimization of IrDA IrLAP link access protocol," *IEEE Transactions on Wireless Communications*, vol. 2, no. 5, pp. 926–938, 2003.

A. C. Boucouvalas has worked at GEC Hirst Research Centre, and became a Group Leader and a Divisional Chief Scientist until 1987 when he joined Hewlett Packard (HP) Laboratories as a Project Manager. At HP Labs, he worked in the areas of optical communication systems, optical networks, and instrumentation, until 1994, when he joined Bournemouth University. In 1996, he became a Professor in multimedia communications, and in 1999 the became the Director of the Microelectronics and Multimedia Research Centre. His current research interests span the fields of wireless communications, optical fibre communications and components, multimedia communications, and human-computer interfaces, where he has published over 200 papers. He has contributed to the formation of IrDA as an industry standard and he is now a Member of the IrDA Architectures Council. He is a Fellow of the Royal Society for the encouragement of Arts, Manufacturers and Commerce (FRSA) and a Fellow of IEE (FIEE). In 2002 he became a Fellow of the Institute of Electrical and Electronic Engineers (FIEEE) for contributions to optical fibre components and optical wireless communications. He is a Member of the New York Academy of Sciences, and the Association for Computing Machinery (ACM). He is an Editor of numerous journals and in the organising committees of many conferences.



Pi Huang received the B.S. degree in electrical and electronic engineering from the University of Central Lancashire, UK, in 2001, and the M.S. degree in telecommunications from the University College London, UK, in 2002. He is currently pursuing his Ph.D. degree in wireless personal area communication network with the Microelectronics and Multimedia Communications Research Centre (M^{2}C), Bournemouth University, UK. His research focuses on performance modelling and analysis as well as discrete-event simulation of wireless communication protocols and wireless communication networks. He has published over 10 papers in the areas of wireless communications. Mr. Huang is a Student Member of IEEE and IEE.