RZ 1739 (#62769) 9/1/88
Communications 23 pages

# Research Report

# Throughput Analysis of a Class of Selective Repeat Protocols in High-Speed Environments

Phuoc Tran-Gia, Hamid Ahmadi, and Parviz Kermani

IBM Research Division
Zurich Research Laboratory
8803 Rueschlikon
Switzerland

IBM Research Division
Almaden • Yorktown • Zurich

# THROUGHPUT ANALYSIS OF A CLASS OF SELECTIVE REPEAT PROTOCOLS IN HIGH-SPEED ENVIRONMENTS

Phuoc Tran-Gia, Hamid Ahmadi* and Parviz Kermani*
*IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland*

**ABSTRACT:** To support data link control in communication systems operating with high-speed transmission systems, various selective repeat protocols have been proposed. In this paper the performance of a selective repeat protocol known as Checkpoint Mode protocol and some of its variations are analyzed. The aim of the analysis is to obtain simple closed-form expressions for the throughput efficiency of the protocol. The approximate results provide insight into the impact of the protocol parameters on its performance. The approximation accuracy is checked with simulations. Numerical examples show the significant performance improvement achieved by the CPM protocols over the go-back-n protocol when used over high-speed terrestrial and satellite links.

---

*Permanent address: IBM Research Division, T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA

August 3, 1988

## 1. INTRODUCTION

In a data communication network, the role of Data Link Control (DLC) is to provide an error-free logical link over a potentially error-prone physical circuit. This is usually achieved by grouping the data to be transmitted into blocks or frames, adding redundancy through appending a Frame Check Sequence (FCS), numbering the information frames, and providing a set of protocol rules by which the nodes at either end of the link initialize recovery through retransmissions.

There are three basic classes of protocols which employ error detection with retransmission, commonly called ARQ (Automatic Repeat Request) protocols [1]: *stop and wait*, *go-back-n*, and *selective repeat* protocols.

The stop and wait protocol is the simplest. The basic idea is to transmit one frame at a time and to wait until the correct reception of that frame is acknowledged. This protocol is not very useful for modern data networks because of its very inefficient use of communication links [1].

The go-back-n data link protocol is mostly used in today's data communication networks. Typical examples of go-back-n protocols are IBM's SDLC [2], DEC's DDCMP, ISO's standard HDLC [3], or IEEE's standard 802.2. In go-back-n protocols, successive data frames are transmitted continuously without waiting for an acknowledgement. When a negative acknowledgement for a frame is received, the frame in question and all frames following it are retransmitted. The ideal version of go-back-n protocols has been analyzed extensively in [4-6]. Various performance issues such as throughput or link efficiency, delay distribution and receiver buffer behavior [7] have been studied for this class of protocols. The go-back-n procedure with finite n has also been analyzed in the context of a more practical version such as HDLC [3,8] and ADCCP/SDLC [9].

For communication links whose the propagation delays are comparable to transmission times of data frames, the go-back-n protocol is generally very efficient. However, there are some other communication links such as satellite and terrestrial links with very high transmission rates, e.g., T1, T2, T3, where the number of frames transmitted in a round-trip delay time is very large. As a consequence, the throughput efficiency of a go-back-n protocol is drastically reduced [10,11]. For these environments selective repeat strategies can be used to enhance the performance [12-17]. The basic idea of a selective repeat ARQ procedure is to retransmit only those frames that are negatively acknowledged or whose timeouts have expired. This obviously improves the performance, but since frames may now be received out of sequence, a reordering buffer at the receiving side is needed [18].

Recently, a selective repeat protocol called Checkpoint Mode (CPM) which allows selective retransmissions of lost packets without any restrictions has been defined and proposed [19,20]. This protocol realizes the ideal selective repeat procedure. In [21], the performance of CPM was studied through simulation. The subject of this paper is an analytic study of this protocol. In addition, a few variations of this protocol will also be considered.

The organization of this paper is as follows. In Section 2, we present a brief description of the CPM protocol and a few variations. Then, through analysis and simulation, we present and discuss various performance aspects relevant to these protocols in Section 3. Section 4 includes the numerical results. Finally, in Section 5, we present our concluding remarks.

## 2. PROTOCOL DESCRIPTION AND APPLICATION

First, we briefly describe the generic CPM protocol as defined and suggested in [19,20]. We will refer to this protocol as the basic CPM.

### 2.1 The basic CPM protocol

This protocol employs two types of frames for the information exchange: Information frames (I-frames) and Checkpoint frames (CP-frames), see Fig. 1. I-frames are used for information transfer and are sequentially numbered in the range of 0 to $2^{16}-1$. The $N_s$ field contains the frame sequence number.

The CP-frames are transmitted periodically, and are used to indicate the status of I-frames received up to that interval. That is, a CP-frame is sent from the receiver to the sender to show which I-frames have not been received (indicated by $N_{x1}, N_{x2}, ..., N_{xm}$), and which I-frame is expected next (indicated by $N_r$). Thus the $N_r$ field contains a number whose value is one greater than the value of the $N_s$ of the most recently received new I-frame. No I-frame is retransmitted by the sending station unless specifically requested by the receiving station via a CP-frame.

The CP-frame acknowledges the receipt of all I-frames whose $N_s$ values are

- between the $N_x$ values in the CP frame,
- between the $N_r$ and the $N_{xm}$,
- older than the oldest $N_x$ (i.e. $N_{x1}$).

For the recovery process of I-frames, the sending station keeps a table in which for each transmitted I-frame with send-sequence number $N_s$, it records $V_s$ and the $N_s$ of the *next* I-frame to be transmitted (see Fig. 2).

Upon receipt of a CP-frame, a station will use the following algorithm to determine whether to retransmit a requested I-frame.

For each frame in the table with $N_s$ equal to any of the $N_x$ in the CP-frame, the station checks whether the associated $V_s$ is less than the $N_r$ of the CP-frame. If so, it retransmits the corresponding I-frame. Otherwise, it ignores the request for retransmission because, when $V_s \geq N_r$, the remote station may not yet have received the previous transmission of this frame. This is the mechanism used in the CPM protocol to avoid unnecessary retransmission. Figure 2 illustrates the recovery procedure of the CPM protocol by an example.

Loss of a transmitted I-frame can only be detected through a sequence error, i.e. when an I-frame is received whose $N_s$ is more than one greater than the $N_s$ of the last new I-frame received. Hence, loss of the last I-frame in a sequence of I-frames cannot be detected until a subsequent I-frame has been transmitted. Without an additional mechanism, the very last I-frame sent would not be recovered and long recovery delays would occur for all I-frames which are not immediately followed by further I-frame transmissions. This problem is circumvented by introducing "empty I-frames" and a timer function named T_I.

Upon transmission of an I-frame, timer T_I, if not already running, will be started. If already running, it will be reset and restarted. If the timer expires, an I-frame with zero length information field (an "empty I-frame") will be transmitted and the timer restarted.

In the CPM protocol, situations may occur where I-frames are received out of sequence at the receiving station. Therefore, I-frames received must be buffered for resequencing. When the resequencing buffer is of finite size, situations will occur where the entire buffer space has been used up and there is a risk of the receiver not being able to buffer received frames. Two alternatives to cope with this problem are 1) to incorporate a flow control scheme in the CPM protocol, and 2) to discard frames that cannot be buffered and rely on the retransmission of these frames. We refer to the former scheme as CPM with flow control and to the latter as CPM with finite resequencing buffer and discard policy.

## 2.2 CPM with flow control

The basic idea in this version of the protocol is to stop sending *new* I-frames when the resequencing buffer is full. A *window flow control* scheme is used to carry out this function (see [21] for a detailed description). The procedure is simply as follows: At link initialization time, the sending station agrees on some *window size* W with the receiving station. The window size W is actually the size of the resequencing buffer (in terms of the number of I-frames) of the receiving side.

If the last CP-frame received contains $N_r = r$ and no $N_x$ (signifying all I-frames up to frame $r - 1$ have been received correctly), then I-frames up to $N_s = r + W - 1$ are allowed to be transmitted. If the last CP-frame received contains $N_r = r$ and in addition some $N_x$'s with $N_{x1} = x$ (note that the $N_x$'s are sorted so that $N_{x1}$ is the *oldest* I-frame not acknowledged), then I-frames up to $N_s = x + W - 1$ are allowed to be transmitted.

This way, it can be assured that at any time not more than $W - 1$ I-frames are outstanding, and hence, no buffer overflow can occur. When the receiver has exactly $W - 1$ out-of-sequence I-frames, the sender does not transmit any new I-frames and only retransmits negatively acknowledged ones until they have been received correctly. It is worth noting that the number of *unacknowledged* I-frames at the sending side is less than W and this control is mainly for the protection of the receiving node buffer. Figure 3 shows an example of the window flow control in operation. In this example the window size is 4 and the assumed CP interval is 5 times the transmission time of an I-frame. For clarity of presentation, only transmission of I-frames from the sender to the receiver and CP-frames from the receiver to the sender are shown in this figure. Also, the resequencing buffer is shown at those instances when the buffer is not empty.

In the window flow control introduced above, the sending station is not allowed to transmit any new I-frame when it has sent $W-1$ frames beyond the oldest unacknowledged I-frame. This condition is referred to as *window being closed.* A certain disadvantage of the window flow control version of the CPM protocol is the idling of the transmission capacity when the window limit has been reached. The proper choice of the window size becomes a very critical issue for this protocol version which we try to study and analyze in this paper.

### 2.3 CPM with finite resequencing buffer and discard policy

In this scheme, the sending station transmits I-frames without any bound, and the receiving station discards them in the event that its resequencing buffer is full. Here, the sizing of the receive buffer becomes a critical parameter. Of course, care has to be taken to avoid deadlocks in such an approach. Our performance studies show the major effect of the finite buffer size on the throughput of the CPM protocol.

## 3. THROUGHPUT EFFICIENCY ANALYSIS

In this section, we present the analysis for the throughput performance of the checkpoint mode class of selective repeat protocols described in the previous section. Due to the complexity of the CPM protocols, we obtain approximate closed-form expressions for the throughput efficiency of the protocol under the window control mode and receiver discard policy mode. The validation of the approximate analysis is achieved by simulations.

In the following analysis, a point-to-point transmission link connecting a sending site A and a receiving site B is considered. The one-way propagation delay from A to B is denoted by $t_p$. To derive the maximum possible throughput using the protocol, we presume the sending site A to be in a saturated state, i.e. there are always frames waiting to be transmitted.

We assume fixed-size information frames of length $(\ell+\ell')$ where $\ell'$ is the overhead part of the frame. The transmission time of information frames $t_I$ is thus $(\ell+\ell')/C$, where C is the link capacity, while the checkpoint frame lengths are considered to be negligible. We also assume an independent bit error rate $p_b$ resulting in a frame error probability $p=1-(1-p_b)^{\ell+\ell'}$.

With these parameter definitions, we derive the throughput efficiency of the three CPM protocols described in Section 2.

### 3.1 Throughput efficiency of the basic CPM protocol

Under saturated load, the throughput efficiency of the basic CPM protocol, i.e. the protocol with infinite window or resequencing buffer size, is exactly the same as the throughput of the *ideal* selective repeat protocol which is given by (cf., Schwartz [1])

$$D/C_{\text{ideal}} = \frac{\ell}{\ell + \ell'} \cdot (1 - p)\,. \tag{1}$$

This is obvious because the only source of inefficiency is the repeated retransmission of the frames in error and the frames overhead.

### 3.2 Throughput efficiency of window-mode CPM

For the analysis of this mode of the protocol, we restrict the checkpoint interval $t_{CP}$ to be an integer multiple of $t_I$. Thus, the time axis is slotted with unit slot lengths $t_I$. There are three types of slots:

- slots with successful transmission of a frame (S-slot)
- slots with transmission of a frame hit by error (R-slot), and
- slots without transmissions, idle slots (I-slot).

The probability of a slot being an S-slot, R-slot or I-slot is $\pi_S$, $\pi_R$ or $\pi_I$, respectively, with

$$\pi_S + \pi_R + \pi_I = 1\,. \tag{2}$$

Note that $\pi_S$ can also be interpreted as the maximum number of error-free frames which can be transmitted per time interval $t_I$. Thus, the throughput efficiency of the protocol is given by

$$D/C = \frac{\ell}{\ell + \ell'} \cdot \pi_S . \tag{3}$$

The main idea of the analysis will be illustrated with a path example as shown in Fig. 4. We consider a fresh test frame which subsequently is hit twice by error after transmission and then successfully received. The process development until the test frame has been successfully transmitted *and acknowledged* is depicted in Fig. 4. As illustrated in this figure, a number of I-slots are induced by the *repeated errors* occurring in the test frame. It is obvious in this example that if the first retransmission of the test frame had been successful, the window size W as depicted would have been large enough to recover the error without inducing idle slots. We define the acknowledgement delay period

$$t_{AD} = 2\,t_P + t_{CP} , \tag{4}$$

as shown in Fig. 4. The acknowledgement delay period is defined as the time from the transmission of a frame to the point in time the transmitting side learns about the status of that frame. In the following, we now choose the window size W to be a multiple integer $\omega$ of $t_{AD}/t_I$

$$\omega = \frac{W\ t_I}{t_{AD}} \tag{5}$$

and assume the test frame to arrive randomly in a checkpoint interval. With this assumption, we estimate the number $n_I(k)$ of I-slots induced by k repeated errors in a test frame before the successful transmission. We refer to $\omega$ as normalized window size; normalized in terms of number of frames that can be transmitted during an acknowledgement delay.

The mean number of slots from the arrival of the test frame until it has been successfully transmitted and acknowledged is (cf. Fig. 4)

$$(k+1)\frac{t_{AD}}{t_I} - \frac{1}{2} \cdot \frac{t_{CP}}{t_I}$$

which includes (k + 1) transmissions of the test frame. This yields

$$n_I(k) = 0 \qquad \text{for } W \geq K$$
$$n_I(k) = K - W \qquad \text{for } W < K$$

where

$$K = (k+1)\frac{t_{AD}}{t_I} - (k+1) - \frac{1}{2}\cdot\frac{t_{CP}}{t_I}$$

or

$$n_I(k) = \max\left[0, (k+1)\frac{t_{AD}}{t_I} - (k+1) - \frac{1}{2}\cdot\frac{t_{CP}}{t_I} - W\right]. \tag{6}$$

By considering an independent bit error rate $p_b$ and the resulting frame error probability $p = 1 - (1 - p_b)^{\ell + \ell'}$, the probability of k successive errors before the successful transmission of a frame is $(1 - p)p^k$. Hence, the mean number $n_I$ of I-slots per successful transmission of a frame is

$$n_I = \sum_{k=\omega}^{\infty} n_I(k)\,(1-p)p^k = \frac{p^{\omega}}{1-p}\left(\left(\frac{t_{AD}}{t_I} - 1\right) - \left(\omega + \frac{1}{2}\cdot\frac{t_{CP}}{t_I}\right)(1-p)\right). \tag{7}$$

We now use an alternative approach to calculate the number $n_I$, by assuming that we know *a priori* the probability $\pi_S$ of a slot to be an S-slot. We observe the transmission process during a time interval T containing M slots; T and M are chosen to be sufficiently large so that the stationary condition can be assumed. It is obvious that the total number of transmissions (S-slots and R-slots) during T is

$$M(\pi_R + \pi_S) = M\,\pi_S + p\cdot M\,\pi_S + p^2\cdot M\,\pi_S + \ldots = M\,\pi_S\cdot\frac{1}{1-p}. \tag{8}$$

Hence, the total mean number of I-slots during T is

$$M\,\pi_I \;=\; M - M(\pi_R + \pi_S) \;=\; M\left(1 - \frac{\pi_S}{1-p}\right). \tag{9}$$

Finally we obtain another expression for the mean number $n_I$ of I-slots per successful transmission of a frame:

$$n_I \;=\; \frac{M\pi_I}{M\pi_S} \;=\; \frac{1}{\pi_S} - \frac{1}{1-p}\;. \tag{10}$$

For an ergodic transmission process the numbers $n_I$ in (7) and (10) are identical. Using this property we obtain

$$\pi_S \;=\; \frac{1-p}{1 + p^{\omega}\left(\left(\frac{t_{AD}}{t_I} - 1\right) - \left(\omega + \frac{1}{2}\cdot\frac{t_{CP}}{t_I}\right)(1-p)\right)}\,. \tag{11}$$

Furthermore, from Eqs. (3) and (11) the throughput efficiency of the CP protocol with window control mode can be written as follows:

$$D/C \;=\; \frac{\ell}{\ell+\ell'}\cdot\frac{1-p}{1 + p^{\omega}\left(\left(\frac{t_{AD}}{t_I} - 1\right) - \left(\omega + \frac{1}{2}\cdot\frac{t_{CP}}{t_I}\right)(1-p)\right)} \tag{12}$$

where

$$\omega = \frac{W\,t_I}{t_{AD}} \qquad \text{and} \qquad t_{AD} \;=\; 2\,t_P + t_{CP}\,.$$

It should be noted here that, in the case of infinite window size, we obtain from Eq. (12) the throughput efficiency of the *ideal* selective repeat protocol as given in Eq. (1).

### 3.3 Throughput efficiency of discard-mode CPM

The derivation of the throughput efficiency of the discard-mode CPM is similar to that of the window flow control mode. For the discard-mode CPM, we devote our attention to the receiving process at the B site with a resequencing buffer of finite capacity Q. It is obvious that here the buffer size Q plays the same role as the window size W did in the previous case.

In the following we briefly describe the analysis steps. A sample path of the transmission process is depicted in Fig. 5. The time axis is again slotted, with the following three types of slots:

- slots with frames received without error (S-slot),
- slots with frames received in error (R-slot), and
- slots with frames received but discarded due to buffer overflow (D-slot).

The probability of a slot to being an S-slot, R-slot or D-slot is $\pi_S$, $\pi_R$ or $\pi_D$, respectively, with

$$\pi_S + \pi_R + \pi_D = 1 .$$

The probability $\pi_S$ is the maximum number of error-free frames which are transmitted per time interval $t_I$ and received by the receiving buffer. Again, the throughput efficiency is

$$D/C = \frac{\ell}{\ell + \ell'} \cdot \pi_S .$$

As illustrated in Fig. 5, we concentrate on a fresh test frame which after transmission is hit subsequently twice by error and then successfully received.

Similar to the analysis in 3.1, the number $n_D$ of D-slots induced by the *repeated errors* of the test frame is determined using two approaches. Then, the throughput efficiency can be obtained by using the identity following from the ergodicity property of the transmission process.

The buffer size Q is chosen to be a multiple integer $\sigma$ of $t_{AD}/t_I$ such that

$$\sigma = \frac{Q\ t_I}{t_{AD}} = \frac{Q\ t_I}{2\,t_P + t_{CP}} \,. \tag{13}$$

We refer to $\sigma$ as the normalized receiver buffer; normalized in terms of number of frames that can be transmitted during an acknowledgement delay.

Assuming the test frame arrives randomly in a checkpoint interval, we determine the number $n_D(k)$ of D-slots induced by k successive errors of the test frame before its successful reception. The mean total number of slots from the first erroneous reception of the test frame to its successful reception can be written as (cf. Fig. 5)

$$k \cdot \frac{t_{AD}}{t_I} - \frac{1}{2} \cdot \frac{t_{CP}}{t_I} \,. \tag{14}$$

In this number, k erroneous transmissions of the test frame are included. Similar to (6) we obtain

$$n_D(k) = \max\left[0\,, k\frac{t_{AD}}{t_I} - k - \frac{1}{2} \cdot \frac{t_{CP}}{t_I} - Q\right], \tag{15}$$

and after some algebraic manipulations

$$n_D = \sum_{k=\sigma+1}^{\infty} n_D(k)\,(1-p)p^k = \frac{p^{\sigma+1}}{1-p}\left(\left(\frac{t_{AD}}{t_I} - 1\right) - \left(\sigma + \frac{1}{2} \cdot \frac{t_{CP}}{t_I}\right)(1-p)\right). \tag{16}$$

The mean number $n_D$ of D-slots per successful transmission of a frame can be estimated in a different way, assuming that we know the probability $\pi_S$ *a priori*. We observe again the transmission process during a time interval T containing M slots. T and M are chosen to be sufficiently large so that the stationary condition can be assumed for the transmission process. The total number of frames received (S-slots and R-slots) during T is

$$M(\pi_R + \pi_S) = M\,\pi_S + p \cdot M\,\pi_S + p^2 \cdot M\,\pi_S + \ldots = M\,\pi_S \cdot \frac{1}{1-p},$$

and the mean number of D-slots during T is

$$M\,\pi_D = M - M(\pi_R + \pi_S) = M\left(1 - \frac{\pi_S}{1-p}\right). \tag{17}$$

Finally we obtain another expression for the mean number $n_D$ of D-slots per successful reception of a frame

$$n_D = \frac{M\pi_D}{M\pi_S} = \frac{1}{\pi_S} - \frac{1}{1-p}. \tag{18}$$

Assuming the receiving process is ergodic, the numbers $n_D$ in (16) and (18) must be identical. This leads to

$$\pi_S = \frac{1-p}{1 + p^{\sigma+1}\left(\left(\frac{t_{AD}}{t_I} - 1\right) - \left(\sigma + \frac{1}{2}\cdot\frac{t_{CP}}{t_I}\right)(1-p)\right)}. \tag{19}$$

The throughput efficiency of the CPM protocol with discard mode is

$$D/C = \frac{\ell}{\ell + \ell'} \cdot \frac{1-p}{1 + p^{\sigma+1}\left(\left(\frac{t_{AD}}{t_I} - 1\right) - \left(\sigma + \frac{1}{2}\cdot\frac{t_{CP}}{t_I}\right)(1-p)\right)}, \tag{20}$$

$$\sigma = \frac{Q\,t_I}{t_{AD}}, \qquad t_{AD} = 2\,t_P + t_{CP}.$$

For infinite resequencing buffer size, Eq. (20) also yields the throughput efficiency of the ideal selective repeat protocol [cf. Eq. (1)].

A comparison of Eqs. (12) and (20) shows that, in order to obtain the same throughput efficiency, the following relation must hold

$$\omega = \sigma + 1 \qquad \text{or} \qquad W = Q + \frac{2\,t_P + t_{CP}}{t_I}, \tag{21}$$

i.e. the window size (window-mode CPM) must be larger than the resequencing buffer size (discard-mode CPM); the difference required is equivalent to the amount of data transmitted during an acknowledgement delay.

### 3.4 Approximation accuracy

A numerical study has been performed for a wide range of parameters, whereby a validation of the approximation has been performed by computer simulations. An example is given in Fig. 6 for a 2.048 Mbps terrestrial transmission line (T1) with constant packet length of 256 bytes including the header. The checkpoint interval length is chosen to be 10 msec and the one-way propagation delay 20 msec. Figure 6 shows the throughput efficiency of the CPM window flow control mode with a normalized window size equal to twice the acknowledgement delay obtained by the approximate analysis and simulation. Also shown are the curves for ideal selective-repeat and ideal go-back-n protocols. Simulation results are obtained with their 99% confidence intervals. The throughput efficiency of the ideal go-back-n is given by (cf. Schwartz [1]):

$$D/C = \frac{\ell}{\ell + \ell'} \cdot \frac{1 - p}{1 + 2p\left(\frac{t_P + t_I}{t_I}\right)}. \tag{22}$$

It can be seen that the approximation method closely predicts the throughput obtained by the simulation. The accuracy is especially good for bit error rates less than $10^{-4}$.

As discussed above, the main idea of the analysis is based on the estimation of the throughput efficiency reduction which is caused by repeated transmission errors in the same frame. The analysis does not take into account the overlapping effect of successive error processes of the frames. Since the recovery periods (cf. Figs. 4 and

5) of different frames may overlap at higher frame error probabilities, it is obvious that the analysis is less accurate at high bit error rates. Thus, the throughput efficiency obtained by the analysis presented here can be regarded as a *lower-bound* estimation of the protocol performance.

For all the cases tried, the approximation method very closely predicts the falling edge of the throughput efficiency curve, which is very important for dimensioning the system parameters and estimating the good operating range of the protocol.

## 4. NUMERICAL EXAMPLES AND PARAMETER SETTING

In this section, we present results for the throughput efficiency of the CPM with window flow control and the CPM with discard policy for various system parameters. As shown by Eq. (21), the analytical expression obtained for the throughput efficiency of the CPM with discard policy for any value of $\sigma$ always corresponds to the throughput efficiency of the CPM with window flow control with $\omega = \sigma + 1$. Therefore, in what follows we show the results for both protocols with only $\omega$ as an index. We compare the efficiency of these protocols to the efficiency of the ideal go-back-n protocol.

Figures 7 to 9 show the throughput efficiency as a function of packet length for the ideal go-back-n and CPM with window flow control. In all figures, the frame overhead is assumed to be 80 bits. Figure 7 shows the result for a 2.048 Mbps (CEPT-T1) terrestrial link with $t_P = 25$ msec and $t_{CP} = 25$ msec. The normalized window size is $\omega = 2$. For a frame length of e.g. 256 bytes, this corresponds to a window size of $W = 150$. It is apparent from this figure that the throughput efficiency of the CPM protocol with window flow control is far superior to that of the go-back-n protocol. This difference is more pronounced at higher bit error rates, e.g., $p_b = 10^{-5}$. Note also that for this bit error rate the optimum frame length is around 300 bytes for both protocols; however, at the optimum frame length, the improvement in efficiency is about 100%.

Figure 8 shows similar results for a 34.368 Mbps terrestrial (T3) link with $t_P = 25$ msec and $t_{CP} = 25$ msec. The normalized window size is again assumed to be only twice the acknowledgement delay in units of frames. Similar results can also be observed for this case, except that the optimum frame length is smaller. For the bit error rate of $p_b = 10^{-5}$, the optimum frame length of the CPM protocol is about 90 bytes and the throughput efficiency at the optimum frame length is close to 80%, while for the ideal go-back-n it is about 5%. This example shows the inefficiency inherent in the go-back-n protocol in a high-speed environment such as a T3 link at higher bit error rates.

In Fig. 9, we show the comparison of these two protocols for a 2.048 Mbps satellite (T1) link with $t_P = 250$ msec and $t_{CP} = 50$ msec. The window size is twice the acknowledgement delay. This figure also shows the significant improvement that can be obtained by the CPM protocol over a high-speed satellite link.

In Fig. 10, we show the throughput efficiency of the CPM protocol for different window-size values. A terrestrial link of capacity 34.368 Mbps (T3) with $t_P = 25$ msec and $t_{CP} = 25$ msec is considered. The bit error rate is assumed to be $10^{-6}$. These curves are plotted for different values of $\omega$. Recall that the window size W was assumed to be a multiple integer $\omega$ of $t_{AD}/t_I$. That means $W = \omega \cdot t_{AD}/t_I$. The curve for $\omega \rightarrow \infty$ represents the ideal CPM or selective repeat protocol. This figure shows that the range of the optimal frame length is very flat. For a frame length of, e.g., 4000 bytes, which is still within the range of optimum frame lengths, $\omega = 2$ corresponds to a window size of 200 frames. It is obvious from this results that larger window sizes will only improve the throughput unnoticeably at the cost of a significant receiver buffer capacity. In the optimum frame length range, the throughput efficiency is insensitive to a variation in window size. That means a window size twice the acknowledgement delay yields as good a performance as any other window size greater than that.

Figure 11 shows the insensitivity of the throughput efficiency to the checkpoint interval $t_{CP}$. These curves are drawn for a satellite link of capacity 2.048 Mbps with $t_P = 250$ msec and bit error rate of $10^{-5}$. From these curves, it is apparent that the throughput efficiency is almost invariant as the checkpoint interval is varied from 50 msec to 200 msec. It should be noted here that in this figure, while the normalized window size $\omega$ is kept constant, the absolute value W of the window size is dependent on the frame length and the propagation delay. This fact must also be taken into account when sizing the checkpoint interval.

## 5. CONCLUSIONS

In this paper we have described and analyzed the performance of the checkpoint mode protocol and a few of its variants. Closed-form expressions have been obtained to approximate the protocol performance under saturated conditions. The throughput efficiency given by the analysis presented here can be considered as a lower-bound estimation of the protocol performance. The approximation is simple to evaluate numerically and is very useful in dimensioning of the major protocol parameters such as frame length, window and receiver buffer size.

Results comparing the throughput efficiency of the CPM protocol with that of a go-back-n protocol demonstrate that the CPM protocol yields a significant performance improvement on both high-speed terrestrial and satellite links environments.

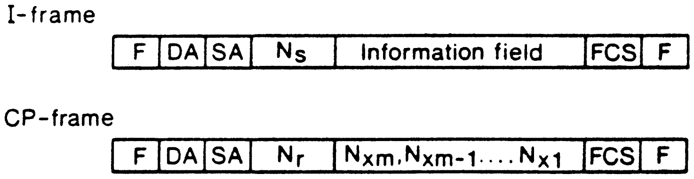
## ACKNOWLEDGEMENTS

The authors would like to thank W. Bux and J. Field for reviewing the manuscript and for their helpful suggestions.

## REFERENCES

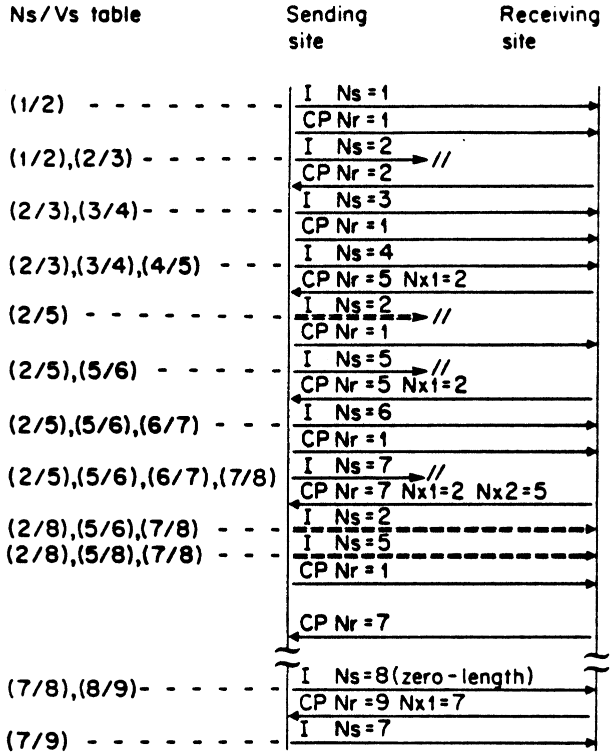
1. Schwartz M., "Telecommunication Networks, Protocol. Modeling and Analysis," Addison-Wesley, Reading, Massachusetts, 1987, Chapter 4.
2. Systems Network Architecture - Technical Overview, IBM Order No. GC30-3073-1.
3. Bux W., Kümmerle K., Truong H.L., "Balanced HDLC Procedures: A Performance Analysis," IEEE Trans. Commun., Vol. COM-28, No. 11, pp. 1889-1898, 1980.
4. Towsley D., Wolf J.K., "On the Statistical Analysis of Queue Lengths and Waiting Times for Statistical Multiplexers with ARQ Retransmission Schemes," IEEE Trans. Commun., Vol. COM-27, No. 4, pp. 693-702, 1979.
5. Konheim A., "A Queueing Analysis of Two ARQ Protocols," IEEE Trans. Commun., Vol. COM-28, No.7, pp. 1004-1014, 1981.
6. Chitre D. M., "Analysis of Throughput Efficiency and Delay in ARQ Systems," COMSAT Technical Review, Vol. 11, No. 2, pp. 345-368, Fall 1981.
7. Benelli G., "Go-back-N ARQ Scheme with Buffer at the Receiver," IEE Proc., Vol. 133, Pt. F, No. 3, pp. 271-276, 1986.
8. Labetoulle J., Pujolle G., "HDLC Throughput and Response Time for Bidirectional Data Flow with Nonuniform Frame Sizes," IEEE Trans. Comput., Vol. C-30, No. 6, pp. 405-413, 1981.
9. Easton M.C., "Batch Throughput Efficiency of ADCCP/HDLC/SDLC/ Selective Reject Protocol," IEEE Trans. Commun., Vol. COM-28, No. 2, pp. 187-195, 1980.
10. Bechtold M., Pujolle G., Spaniol O., "Throughput of a Satellite Channel Communication," in: Acta Informatica 22, Springer-Verlag, Berlin Heidelberg, 1985, pp. 1-14.
11. Andrews D. W., "Throughput Efficiency of Logical Links over Satellite Channels," in: Satellites and Computer Communication, edited by J. L. Grange, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1983, pp. 231-241.
12. Chitre D. M., "A Selective-Repeat ARQ Scheme and its Throughput Analysis," Proceeding of ICC'82, pp. 6G.4.1-6G.4.6, June 1982.

13. Irland M. I., Pujolle G., "Comparison of Two Packet-Retransmission Techniques," IEEE Trans. Inf. Theory, Vol. IT-26, No. 12, pp. 22-97, 1980.
14. Miller M. J., Lin S., "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer", IEEE Trans. Commun., Vol., COM-29, No. 9, pp. 1307-1315, 1981.
15. Weldon E.J., Jr., "An Improved Selective-Repeat ARQ Strategy," IEEE Trans. Commun., Vol. COM-30, No. 3, pp. 480-486, 1982.
16. Metzner J.J., Chang D., "Efficient Selective Repeat ARQ Strategies for Very Noisy Channels," Proceedings of Globecom'83, San Diego, California, pp. 35.2.1-35.2.8, 1983.
17. Anagnostou M.E., Protonotarios E.N., "Performance Analysis of the Selective Repeat ARQ Protocol," IEEE Trans. Commun., Vol. COM-34, No. 2, pp. 127-135, 1986.
18. Rosberg Z., Shacham N., "Re-Sequencing Delay and Buffer Occupancy Under the Selective-Repeat ARQ," Proceedings of Third International Conference on Data Communication Systems and Their Performance, Rio de Janeiro, Brazil, in: Data Communication Systems and Their Performance, edited by L.F.M. de Moraes, E. de Souza e Silva and L.F.G. Soares, Elsevier Science Publishers B.V. (North-Holland), IFIP 1988, pp. 69-82.
19. Brodd W. D., Donnan R.A., "Data Link Control Improvements for Satellite Transmission," in: Satellites and Computer Communication, edited by J. L. Grange, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1983, pp. 201-213.
20. Donnan R.A., "Method and System for Retransmitting Incorrectly Received Numbered Frames in a Data Transmission System," United States Patent 4,439,859, March 27, 1984.
21. Bux W., Kermani P., Kleinöder W., "Performance of an Improved Data Link Control Protocol," Proceedings of 9th International Conference on Computer Communication (ICCC), Tel Aviv, November 1988.



| | |
|---|---|
| F | Flag |
| DA | Destination address |
| SA | Source address |
| $N_s$ | Frame sequence number |
| $N_r$ | Next expected I-frame |
| $N_x$ | Frames in error |
| FCS | Frame check sequence |

Fig. 1: Frame formats of Checkpoint Mode Protocol (CPM)



Ns: Sequence number of the current I-frame
Nr: Sequence number of the next I-frame expected
Nx: Sequence numbers of I-frames in error
Vs: Sequence number of next new I-frame
to be transmitted

Fig. 2: Error recovery example with Checkpoint Mode Protocol (CPM)

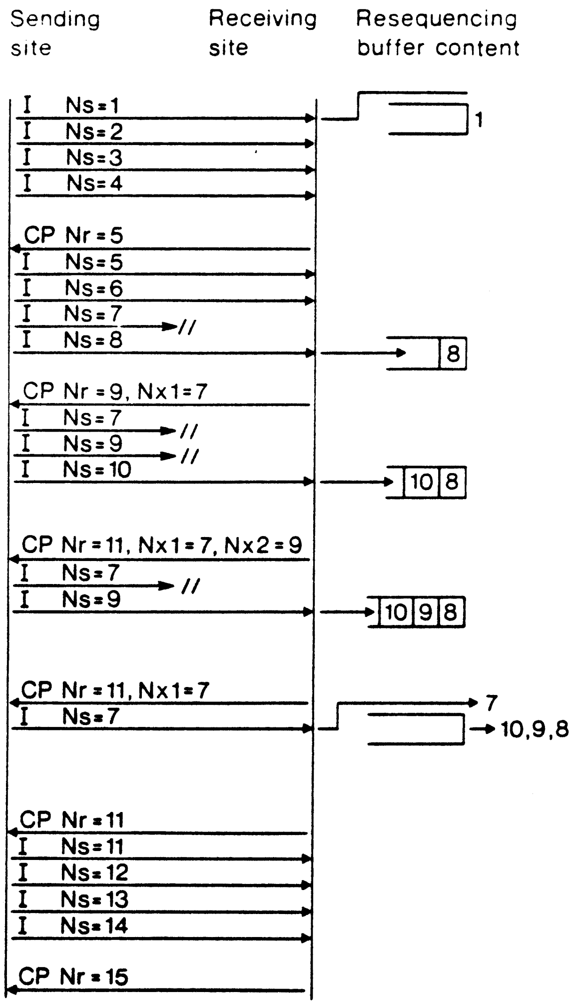


Fig. 3: Example of CPM with window flow control (window size W = 4)

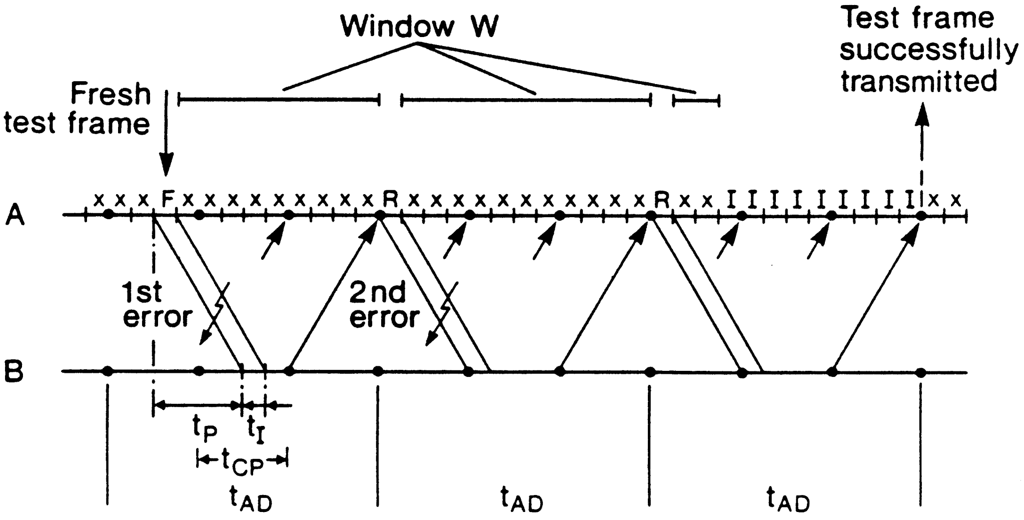


Fig. 4: CPM with window: sample path for throughput analysis

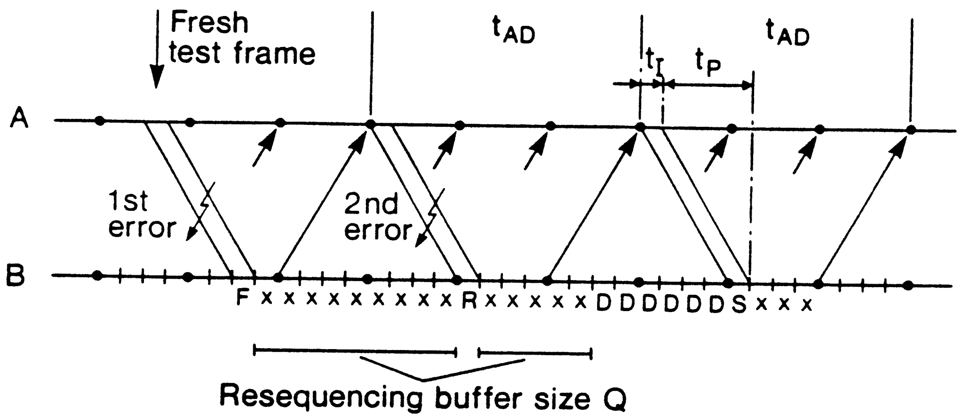


Fig. 5: CPM with discard mode: sample path for throughput analysis

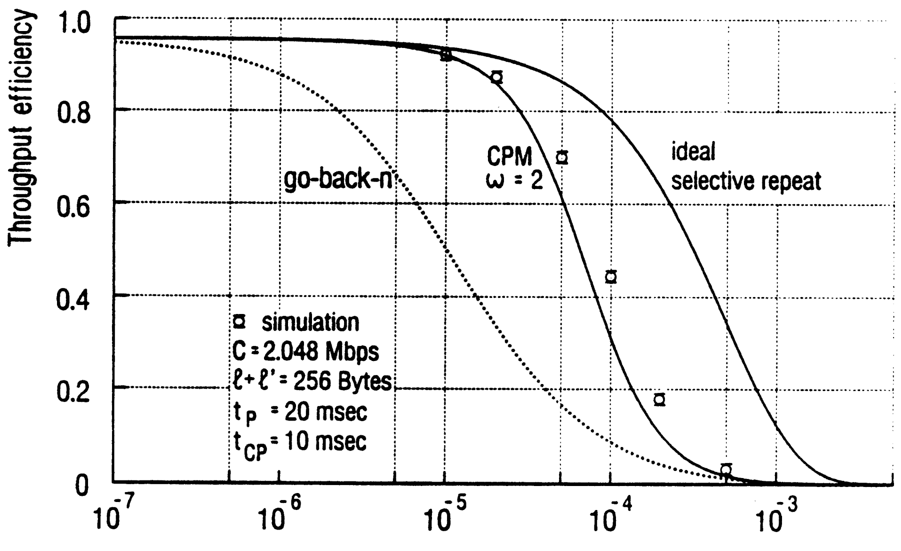


Fig. 6: Approximation accuracy: example of long terrestrial T3 link

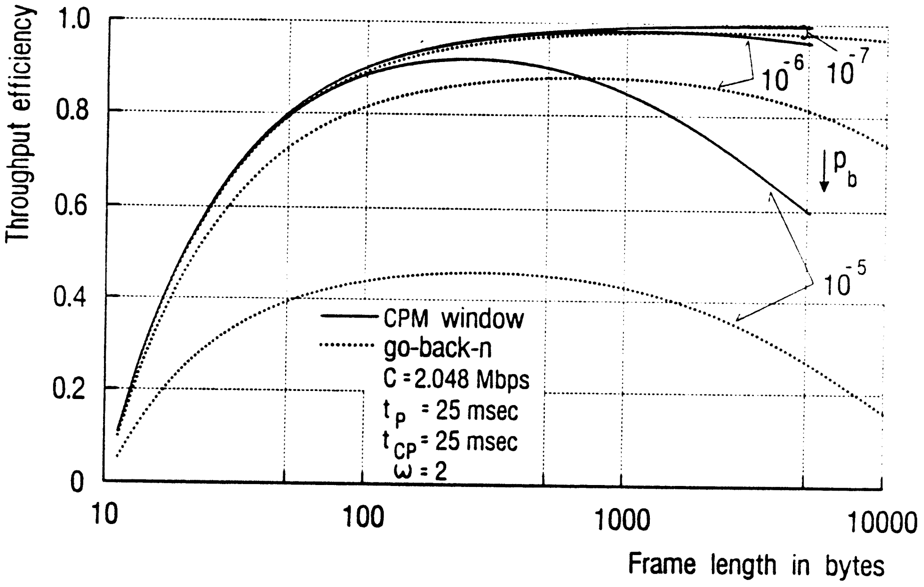


Fig. 7: Throughput efficiency comparison: long terrestrial T1 link

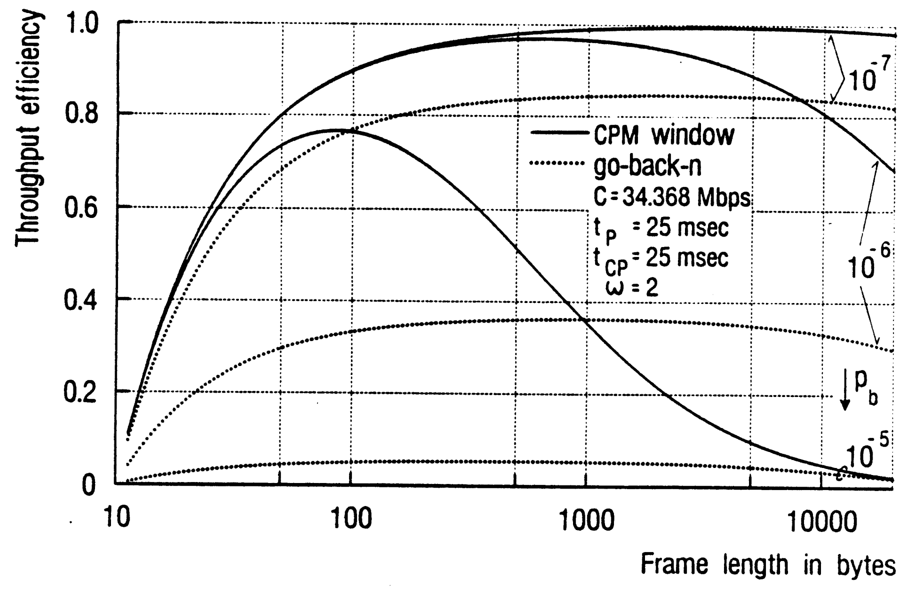


Fig. 8: Throughput efficiency comparison: long terrestrial T3 link

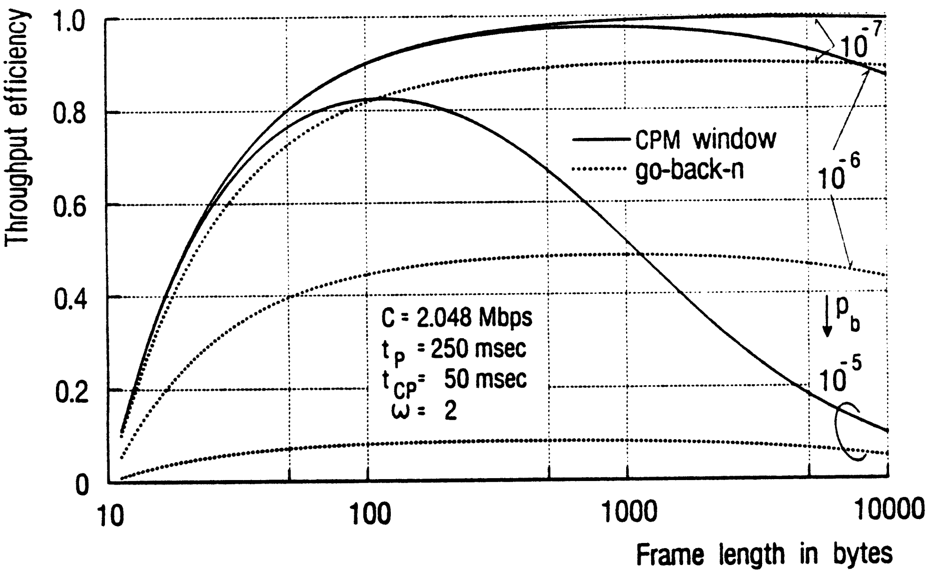


Fig. 9: Throughput efficiency comparison: satellite T1 link

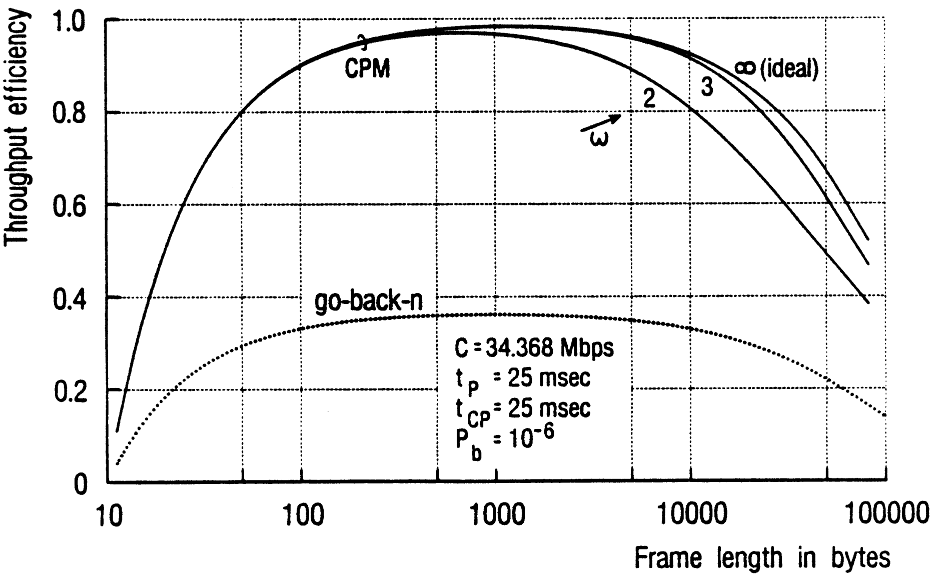


Fig. 10: Impact of window size on throughput efficiency

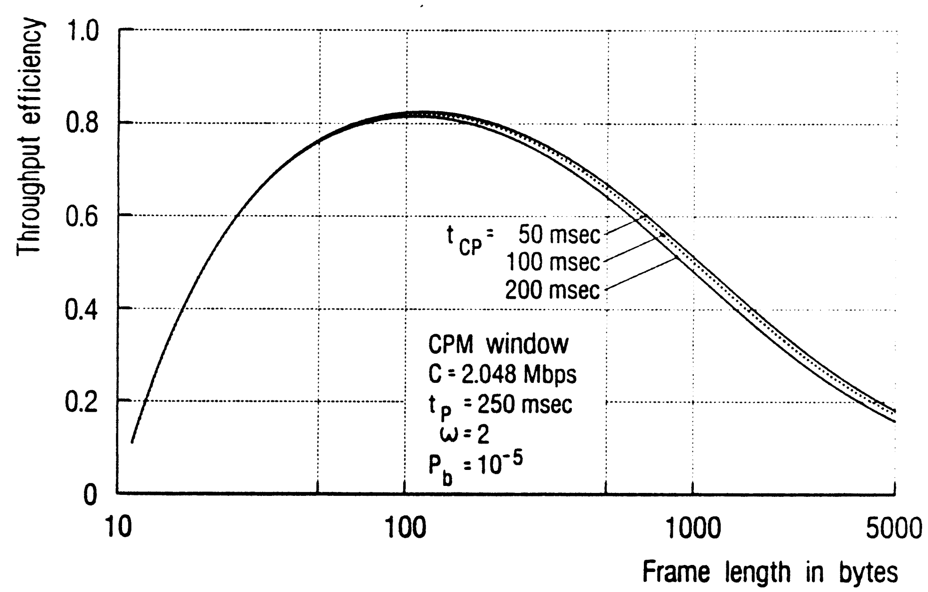


Fig. 11: Impact of the checkpoint interval on throughput efficiency