PERFORMANCE OF LAPD FRAME-RELAY NETWORKS: TRANSMISSION ERROR EFFECTS AND CONGESTION CONTROL

K.-J. Chen, B.T. Doshi, H.Q. Nguyen and K.M. Rege
AT&T Bell Laboratories
Holmdel, NJ 07733, U.S.A.

In this paper we describe some models which were developed to study the effects of transmission errors and various congestion control schemes on the performance of LAPD Frame Relay networks. These models are used to provide results for representative error characteristics and traffic mixes. It is shown that link efficiency remains high at representative values of transmission error parameters and that effective congestion controls are possible for such networks.

1. INTRODUCTION

Over the last few years there has been significant interest in using LAPD as the level 2 protocol on the bearer channels for packet switching in an ISDN environment. LAPD brings switching and multiplexing functions to level 2, the lowest possible level with packet switching. This reduces complexity in the protocol and makes for a higher frame handling capacity in the protocol handlers and easier hardware implementation. The functions performed by LAPD are switching, multiplexing, error detection, window rotation and retransmissions (in response to rejects and timeouts). When a node performs only the first three functions, it is called a Frame Relay (FR) node. When all network nodes are FR nodes with only the end systems terminating full LAPD, we have a frame relay (FR) network. If LAPD is fully terminated at the end systems as well as the network edges, we have an edge terminated frame relay network. In our discussion, we will focus on a connection involving two LAPD end points separated by \( M-1 \) frame relay nodes (Figure 1). We will also assume that the level 3 protocol rides end-to-end. Our discussion and results are applicable to both FR and edge terminated networks.

By eliminating the need for state machines, FR further simplifies the protocol and thus can add significantly to the frame handling capacity and ease of hardware implementation. This is crucial at high link speeds and small frame sizes. For example, the transmission times of a 128 byte frame at 64 kb/s and 1.544 Mb/s are 16 msec and 0.67 msec, respectively. A 10 byte frame (character, acknowledgment and other control frame) takes 1.25 and 0.06 msec at the above speeds. In order for the protocol handler to not become the bottleneck and/or add significant latency, it should process a frame in less time than the ones stated above. Protocol simplicity and ease of hardware implementation offered by frame relaying is thus very important at high speeds and with short frames.

While frame relaying offers protocol simplicity and hence higher capacity and lower latency, it does so at the expense of some LAPD functionality. Two important ones are the following: (i) since the window rotation and error induced retransmissions are performed only at the end systems terminating LAPD, a transmission error on any of the \( M \) links in the FR connection causes retransmissions over all \( M \) links, which raises the question as to what is the effect on the efficiency of transmission of FR as compared to link by link termination of LAPD; (ii) since neither level 3 nor level 2 is fully terminated at a FR node, it cannot delay or withhold acknowledgements (as would be available at level 3), or use RR/RNR messages (part of LAPD level 2) for congestion control and the question that naturally arises is about the feasibility, ease and effectiveness of alternative congestion controls.

In Section 2 we present simple models and results on the effects of transmission errors on the efficiency of frame relay connections under a wide range of parameters. Typical numbers
suggest that the effects on the efficiency are minimal. In Section 3 we compare performance of a variety of congestion control mechanisms for frame relay connections. The results indicate that effective control is possible in frame relay networks.

2. EFFECTS OF TRANSMISSION ERRORS IN FRAME RELAY CONNECTIONS

When transmission errors cause a frame to be received in error or get dropped at some intermediate node, the subsequent reject frame due to an out of sequence frame or expiration of the $T_{rej}$ timer will result in retransmission of all the frames in the pipeline. It is these retransmissions which affect the effective level 2 throughput. The number of retransmissions is bounded by the window size and the window size based computations provide the worst case analysis of the link efficiency (defined as the ratio of the number of useful level 2 bits transmitted to the total number of bits transmitted). Thus we first calculate the window size (in numbers of frames) as a function of network and access line parameters. The parameters of interest are the following: the number of links, $M$, in a single LAPD connection with $M-1$ FR nodes; the maximum throughput desired for the virtual circuit ($C_G$) ($C_G$ is bounded by the access line speed); the network link speed, $C_L$; the data frame size, $N_d$; the acknowledgment frame size, $N_a$; the round trip propagation delay, $T_p$, the baseline processing and other minimal queueing delay on acknowledgment $T_q$.

Since the window size, $W$, should be large enough to keep the round trip pipeline full when the rest of the network is idle, we have:

$$W \geq 2 \left[ 1 + \frac{N_a}{N_d} \right] + (M-2) \frac{C_a}{C_L} \left[ 1 + \frac{N_a}{N_d} \right] + (T_p + T_q) \frac{C_a}{N_d}.$$

For some typical values of parameters, the values of $W$ are shown in Table 1. We will use these in our analysis of the transmission error effects.

While sophisticated models are available to represent the characteristics of transmission errors in detail, for our purpose a simple, two-state model is adequate. We assume that the transmission facility (access line or a network link) can be in two states: in the normal state there is a random bit error rate denoted by $r$; the second state corresponds to short periods of error bursts during which the bit error rate, $r_b$, is much higher than $r$. The basic assumption is that the error bursts are long enough and separated from one another so that the good and bad periods can be analyzed separately. Let $P_g$ be the fraction of time a link is in the normal state. The overall bit error rate, $r$, is then given by $r = r_g P_g + r_b (1-P_g)$. If we assume that error bursts are long enough to cover at least one full frame, then the frame error rate, $F_e$, is given by

$$F_e = 1 - \left[ P_g (1-r_g) N_d + (1-P_g) (1-r_b) N_d \right]^W.$$

A frame that is being transmitted from the sending endpoint will have to be retransmitted if any of the frames in transit, including this one, is hit by a transmission error. Let $F_{RT}$ be the probability that a frame will have to be retransmitted. If the error bursts are short so that they affect isolated frames, then this retransmission probability, $F_{RTS}$, is given by

$$F_{RTS} = 1 - \left[ P_g (1-r_g) N_d + (1-P_g) (1-r_b) N_d \right]^W.$$

On the other hand, for error bursts long enough to affect an entire window, the retransmission probability, $F_{RTL}$, is given by

$$F_{RTL} = 1 - \left[ P_g (1-r_g) N_d W + (1-P_g) (1-r_b) N_d W \right]^W.$$

Finally, the link efficiency of any link in the connection is given by $L_e = 1 - F_{RT}$. We will denote by $L_{LS}$ and $L_{L}$ the link efficiency corresponding to the short and the long error bursts, respectively. If we put $M=1$ in our equations, we get the link efficiency with link-by-link termination of LAPD. In addition, if we put $C_a = C_L$, then we get the link efficiency for LAPB.
Numerical Results for Typical Parameters: We use some typical mix of access line and network link speeds, and the parameters $P_r$, $r_p$, and $r_l$. The values of $M$, $T_r$, and $T_t$ in our calculations are 6 and 50 msec and 10 msec, respectively. The results for the window size and link efficiency are presented in Table 1. For measurements conducted on AT&T T1 carrier links [6], the background bit error rate was found to be better than $10^{-7}$ for almost all the links and for most facilities greater than 0.999 of the seconds were essentially error free. Thus $P_r = 0.999$ is a very conservative number effectively assuming that the whole errored second is an error burst. With that assumption it is appropriate to use $F_{RTL}$ and $L_{EL}$ as the representative performance numbers. Finally, error bursts of 2.5 seconds and longer were very rare (3-5 per week). The table shows that over a wide range of parameters, the degradation in the link efficiency due to lack of retransmission capability at FR nodes is quite small. When the access line speeds are very high and the propagation delays are large, frame relaying as well as link by link LAPB or LAPD suffers loss of link efficiency for $r_p = 10^{-7}$. The link error characteristics should be better ($r_p < 10^{-8}$ perhaps) to support such applications, or else, selective rejects can be used to control the throughput degradation due to retransmissions with large windows. At much higher values of $C_a$ (and hence $W$), better random error characteristics and/or selective rejects become even more important.

8. CONGESTION AND CONGESTION CONTROL

We will primarily focus on the bandwidth overload here with the hope that frame relay simplifies the protocol handling to the extent that bandwidth is the main bottleneck. Thus congestion occurs when the total data the users want to send (and can send with their access line speeds) over some period $T$ is larger than $C_L T$, where $C_L$ is the speed of the link in question. Since the data traffic is known to be bursty and does not occur continuously during the holding time of a connection, the network, to utilize its resources efficiently, does some concentration. That is, if $C_a$, is the access line speed of the $i^{th}$ virtual circuit going over a link and $N_L$ is the number of virtual circuits set up over this link, then the sum of the $N_L$ access line speeds will be greater than $C_L$. Some congestion avoidance can be accomplished by call setup control, effectively limiting $N_L$ and the sum of the access line speeds. The engineering issues for call setup control in FR networks are similar to those in X.25 or other networks so we will not discuss them here. The question of interest here is what happens and what can be done when enough of the $N_L$ virtual circuits are active simultaneously so that over a reasonably long period the total intended load exceeds the link capacity.

The main effects of such a congestion period are the following: (i) transmit buffer overflow may result in dropped frames, frames delivered out of sequence to the receiver, rejects or Layer 2 timeouts ($T_{200}$). The resulting retransmission of up to a whole window will degrade the useful level 3 throughput and may result in a throughput crash. (ii) Some overactive users may degrade the throughput-delay performance of all the users. (iii) Retransmissions may cause otherwise noncongested components (links) to experience congestion. (iv) Frequent Layer 2 timeouts may result in session disconnects by the end systems.

Thus the objectives of a congestion control mechanism are the following: (i) Keep useful throughput as high as possible in order to get out of the congestion quickly and prevent a throughput crash. This implies that when frames on a virtual circuit have to be dropped, they should be dropped in bunches rather than in an isolated manner (analogous to the effects of random vs bursty transmission errors). (ii) Do not interfere with the natural protocol in absence of congestion. (iii) Protect well behaved users from the misbehaving ones to the extent possible. In the event of a general congestion, divide the pain equitably. (iv) Prevent session disconnects unless desired by the congestion control needs.

We now discuss and compare a set of congestion control mechanisms with respect to the above objectives with greater focus on (i). Some of these controls have been proposed and evaluated for some network environments earlier [2,3,6]. We will attempt to unify them and provide comparative performance guidelines over a spectrum of parameter values and traffic mixes. While some of the results are obtained by simple analytic models, a detailed simulation of FR network is used where dynamics were important.

5.3B.2.3
9.1 Congestion Control Schemes

Note that, as far as the first objective is concerned, the LAPD protocol and the associated windowing mechanism provides a degree of control. During congestion the delays caused by higher buffer occupancy slow down window rotation resulting in a reduced offered load. If we can allocate enough buffer space to hold the whole window for each virtual circuit that call setup control allows, then, as long as the Layer 2 timers do not run out, no frames will be dropped and the desired load shedding is achieved. Two factors affect the feasibility of full buffer dedication. The first one is the buffer size itself, its cost and ease of management. The buffer size in turn depends on the window size and the number of virtual circuits needed to keep the link busy. The other factor is the delay induced by a large buffer under congestion and the possibility of Layer 2 timers timing out and causing retransmissions. Besides, the delays induced by a large buffer during congestion may violate certain delay service objectives.

In Table 2 we present the total buffer size required and the worst case delay seen by a frame in that buffer for a set of access line speeds, link speeds and the average activity level on each virtual circuit. The numbers are based on the assumption that enough virtual circuits are set up to keep the link busy 50% of the time. When the activity level on each virtual circuit is high and $C_L/C_s$ is not very large, it seems feasible to provide fully dedicated buffers. At higher values of $C_L/C_s$ and lower activity levels, full dedication is questionable unless the Layer 2 timers are extended (default value is one second), or polling on expiry of these timers is made mandatory, large and inexpensive memories are available and easily manageable at high speeds, and the resulting delays do not violate service objectives.

If the traffic mix consists of a large number of virtual circuits with high values of $C_L/C_s$ and short bursts of data coupled with a small number of virtual circuits with small values of $C_L/C_s$ and longer bursts of data (for example, the former may be interactive sessions and the latter may be file transfers), then the following strategy may be used: For the former type of virtual circuits (type 1) provide a shared buffer sized to keep the probability of overflow small. For the latter type (type II) dedicate the full window worth of buffer. Serve the type I buffer at higher priority than type II. With higher priority type I buffer can be made small without causing a large probability of overflow. Priority also magnifies the natural slow down of the high speed virtual circuits for which the possibility of an overflow has been eliminated. While the type I buffer can overflow, the large number of virtual circuits involved makes the event statistically rare. A mixed (open and closed) network of queue model has been developed for this scheme and an accurate approximation has been derived which will be presented in a future paper.

Table 3 illustrates how this scheme can be used to save buffer space while maintaining a low probability of overflow for the high priority virtual circuits. Note that, if the high priority data bursts are short, congestion caused by statistical fluctuations will be of a short duration and, as in the case of transmission errors, will not have significant detrimental effects if they remain infrequent.

If a sizable fraction of traffic comes from high speed virtual circuits with relatively low activity levels, then the above scheme does not help in reducing the buffer requirement, and some form of additional real time control is necessary. This form of control has two aspects: congestion detection and congestion control actions. Congestion detection can be done implicitly by the LAPD end points by monitoring rejects and timeouts (or acknowledgment delays, although this will not be used in our study) or explicitly by the network components. Similarly, the control action may be taken by the end-systems or by the network. It is also possible to have the network initiate control action which provides a quicker feedback to the end systems enabling them to shed load. Thus the schemes to be described below are generally classified according to whether the detection is implicit or explicit and whether the action is by the end systems, the network or both.

*Implicit Detection:*

C1. Adaptive Window Schemes:

Here, the end system assumes the existence of congestion when a reject is received or when the Layer 2 timer times out and reduces the level 2 window to shed load. When enough consecutive
frames are successfully acknowledged, it is believed that the congestion is over. (The end system assumes a maximum window size of $W_{\text{max}}$ and a minimum window size of $W_{\text{min}}$.) While many schemes are possible within this window adaptation framework [7], we shall discuss the following three schemes: C1.1) On reject or timeout decrease window by 1; on $N$ consecutive successful acknowledgments increase window by 1 up to $W_{\text{max}}$. C1.2) Decrease to $W_{\text{min}}$, increase by 1. C1.3) Decrease to a fraction $\alpha$ of the previous window (suitably rounded), increase by 1. For numerical studies we use $\alpha = 1/2$.

The following factors affect the relative performance of the above schemes: a) Since transmission errors will be perceived as congestion, a fast acting control like (ii) may result in degraded throughput in absence of congestion if the error rate is high. b) When $W_{\text{max}}$ is large, a slow acting control like (i) may be very sluggish. c) For small values of $W_{\text{max}}$ all three should behave similarly. d) Typically $W_{\text{min}}$ is 1. If more is known then $W_{\text{min}} > 1$ can be used. On the other hand, for a very high degree of concentration, even a window size of 1 may not be small enough and $W_{\text{min}} = 0$ may be required. In that case a timer is necessary to raise the window again. Although we have simulated only adaptive window based schemes with implicit detection, for end systems using a non-window based protocol or with no facility to adapt windows, it is possible to shed load by stopping for a duration and then resuming transmission.

Explicit Detection Schemes:

Here the network components monitor the usage of critical resources (transmit buffers, for example) and when congestion is perceived, either initiate a control action or send messages to the sending entities. The congestion advisory message may be intercepted by the network edge to take the control action or may be relayed to the end systems where the action is taken. The fastest way to convey congestion status information is to send an inband message in response to every frame which on arrival finds the buffer to be above a threshold. This may be done by generating a separate message or marking a bit in the frames going in the reverse direction. For lower speed networks this may even permit avoidance of buffer overflow due to congestion [3]. At higher speeds the message latency may prohibit congestion avoidance. It may then be easier to put a low pass filter in the measurement over a period and when, congestion is perceived, send inband or out of band messages. The following controls of this general nature can be considered.

C2. Actions at the network edge:

The network edge, on receiving a congestion advisory message (inband message, congestion bit or out of band message) for a given virtual circuit, will drop frames as in the stop-duration scheme discussed below.

C3. Actions in the end system:

It is assumed here that LAPD is enhanced to relay (or generate separate messages from the edge) congestion advisory messages to the sending end system. The messages can then result in load shedding by the end system. Two possible schemes are the following (although we have simulated only the first one so far):

C3.1) Stop-duration: L2 transmitter (or L3-L2 interface) stops transmissions of I frames for a duration. The duration is fixed or selected randomly from a set of values [3]. C3.2) Adaptive window: If the advisory message is in the form of a single bit the action can be the same as in C1.1-C1.3 except that in this scheme error induced actions are avoided. The relaxation algorithm is also the same. When the network sends a more explicit message, the actual window size to be used can be specified based on the buffer size and the number of active virtual circuits. Once again, for reasons of fairness, the advisory messages can be sent selectively to overactive end systems.

Scheme C1.2 has been simulated in a LAN environment [2] with small values for $W_{\text{max}}$ and was found to be very effective. Scheme C3.1 (with small window sizes) has been simulated and was found to give nearly perfect throughput results for a wide range of load [3]. Here, we use the model and workload characterization (a mixture of character and block interactive and file transfer traffic) used in [3] but we study the behavior of these schemes under different window
sizes. The results are presented in Table 4. For the implicit adaptive window scheme, we also studied the effects of transmission errors on the throughput from a single virtual circuit (Table 5).

From these figures we observe the following:

i) Performance of explicit schemes involving actions at the edge or end systems is somewhat sensitive to the buffer size in relation to the window size. When the buffer is large enough so that full windows for a moderate number of virtual circuits can be accommodated in the space between the threshold and the buffer size, stop-duration schemes (C2, C3.1) with different durations achieve excellent throughput characteristics by avoiding buffer overflow. If the gap between the buffer size and the threshold can accommodate only a few full windows, then overflow avoidance is not very good and each overflow causes a significant amount of retransmissions. Adaptive window based schemes will face the same problem during periods of window reduction. However, once the transient period is over, the window will remain relatively small and the steady state throughput will remain robust. This suggests two things: a) Adequate buffers and short latency in congestion advisory message will make explicit stop-duration schemes (C2, C3.1) perform better than the implicit schemes. b) If the buffers are small, adaptive window provides a robust performance. Its transient performance can be improved by combining it with congestion advisory messages. Explicit adaptive window schemes can achieve this. Otherwise, a stop duration message can be interpreted to mean reducing the window to 0 for the indicated duration and then relaxing the control as in the adaptive window scheme. This would eliminate large steady state windows used in the on periods and the resulting throughput loss. Performance of such a scheme should be robust even when the buffer is too small to accommodate one frame for each VC.

ii) Among the implicit adaptive window schemes studied (C1.1-C1.3), C1.2 provided the best throughput performance and C1.1 the worst, with C1.2 and C1.3 being very close. The difference between C1.1 and C1.2 is more significant with large nominal window sizes (e.g. 7) and heavy congestion. Clearly, as discussed above, the performance difference will be even more significant during the transient phase. For the two window sizes considered in Table 4 (3 and 7), \( W_{\text{min}} = 1 \) seems the right choice. However, when high access line speeds (e.g. 1.5Mb/s) with significant propagation delays necessitate a large nominal window (50 or more), the ideal window size even during periods of congestion could be greater than 1 (say, 4 or 5). Without knowledge of the buffer size and congestion status, the end systems are likely to use \( W_{\text{min}} = 1 \). In this case, a fast acting scheme like C1.2 may suffer throughput loss due to its fast downward adaptation followed by a slow recovery procedure. In such cases, a multiplicative scheme like C1.3 may provide a better speed of action.

With explicit schemes, the network can use additional information (e.g. buffer sizes, number of active virtual circuits, etc.) to provide a better transient and steady state performance by adapting the speed of action to suit the prevailing conditions. In addition, explicit schemes can avoid buffer overflow and maintain low delays by requesting load shedding before heavy congestion sets in and then keeping the load within the network's capacity.

iii) As Table 5 indicates, C1.1 is better than C1.2 at avoiding false alarms caused by transmission errors. On the other hand, even for a large window size such as 16, the false alarm rate and its effect on the throughput is negligible for all schemes once the bit error rate becomes \( 10^{-7} \) or lower.

Other Remarks

We have concentrated on the throughput performance under congestion for a variety of congestion control schemes. As we mentioned several times, in each scheme it is possible to enforce fairness by suitable enhancements. Some of these are briefly discussed below.

(i) Fairness is an issue with or without congestion and is only magnified under congestion. If buffers are allocated to individual or groups of virtual circuits (not necessarily the whole window), then round robin service of frames from these buffers will ensure fairness in delay in absence of congestion and dropping of frames selectively from overactive virtual circuits during congestion. An adaptive window scheme can then force the affected virtual circuits to slow
down. Further discrimination can be obtained by explicitly monitoring the bandwidth used by each virtual circuit in the recent past and serving overactive virtual circuits at lower priority. On the other hand, delay sensitive applications can be served at higher priority. If enough buffer space is not available, these overactive virtual circuits will see a greater fraction of their frames dropped. In explicit advisory message based schemes, the messages can be sent selectively based on the bandwidth consumption. All but the simple round robin type mechanism for throughput enforcement involve additional complexity in measurements. (ii) Bandwidth usage can be monitored at the network edge and leaky bucket type of schemes can be used to identify and blacklist traffic above the allocated values. This blacklisted traffic may be dropped at the edge of the network if congestion is known to exist or may be treated differently by each node in the network (low priority, smaller threshold for entrance, etc.). One such scheme coupled with adaptive window is analyzed in [5].

For the parameters and congestion range we have studied, the results indicate that it is possible to slow down each virtual circuit without causing prolonged periods of no useful transmission. Thus, with these parameters, it seems unlikely that the sessions will complete the whole LAPD recovery attempt procedure and eventually terminate the session. On the other hand, the network may decide that it is better to provide excellent service to a fraction of end systems rather than degraded service to all. In this case, the network may decide to force a selected set of virtual circuits to disconnect.

Also note that all except C3.1 and C3.2 are confined to action in the network or implicit detection and action by the end systems thus not requiring any transfer of congestion information at the interface. For C3.1 and C3.2 LAPD needs to be enhanced to provide such information transfer. If this can be done easily and with small latency compared to the network speed, some avoidance of frame dropping can be achieved in addition to load shedding after the frames are dropped.

REFERENCES

Figure 1: A simple M-link LAPD frame-relay connection

Table 1: Transmission Error Effects

\[ T_p = 50 \text{ msec}, \ T_e = 10 \text{ msec}, \ N_d = 1096, \ N_s = 80, \ r_b = 10^{-10}, \ P_f = 0.999, \ r = 10^{-6}, \ M = 6 \] (end-to-end LAPD), \ M = 1 \ (LAPB link by link).

\[ L_{EL} \]

<table>
<thead>
<tr>
<th>Case</th>
<th>Case</th>
<th>Buffer</th>
<th>Activity Level per Virtual Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Buffer</td>
<td>0.01</td>
</tr>
<tr>
<td>1</td>
<td>113</td>
<td>14.1</td>
<td>22.5</td>
</tr>
<tr>
<td>2</td>
<td>2020</td>
<td>10.5</td>
<td>404</td>
</tr>
<tr>
<td>3</td>
<td>960</td>
<td>5.0</td>
<td>192</td>
</tr>
<tr>
<td>4</td>
<td>68</td>
<td>8.5</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>620</td>
<td>3.23</td>
<td>124</td>
</tr>
<tr>
<td>6</td>
<td>14207</td>
<td>3.08</td>
<td>2841</td>
</tr>
</tbody>
</table>
Table 3: Comparison of the full buffer dedication vs priority to low speed short burst traffic

<table>
<thead>
<tr>
<th>$C_L$ (Kb/s)</th>
<th>$C_s$ (Kb/s)</th>
<th>Buffer Size</th>
<th>LP Fraction</th>
<th>Offered Load</th>
<th>Buffer Size</th>
<th>$P(HP\text{ Overflow})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HP LP</td>
<td>Normal Load</td>
<td>Overload</td>
<td>HP LP</td>
<td>Normal Overload</td>
</tr>
<tr>
<td>1544</td>
<td>16</td>
<td>1544</td>
<td>1716K</td>
<td>0.1</td>
<td>0.6</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1544</td>
<td>64</td>
<td>1544</td>
<td>887K</td>
<td>0.1</td>
<td>0.6</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Throughput Performance of Control Schemes

All adaptive window schemes use $N=8$. Window size 3 is for a three link connection and 7 for a five link connection. Table 4 gives the values of trunk utilization due to packets correctly delivered to Level 3. Buffer size is 4 KBytes [3].

<table>
<thead>
<tr>
<th>Control Scheme</th>
<th>$W=3$ Nominal Offered Load</th>
<th>$W=7$ Nominal Offered Load</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.5 3.0 4.5</td>
<td>1.5 3.0 4.5</td>
</tr>
<tr>
<td>C1.1</td>
<td>0.681 0.624 0.58</td>
<td>0.677 0.611 0.520</td>
</tr>
<tr>
<td>C1.2</td>
<td>0.695 0.641 0.595</td>
<td>0.695 0.628 0.575</td>
</tr>
<tr>
<td>C1.3</td>
<td>0.689 0.633 0.59</td>
<td>0.678 0.621 0.568</td>
</tr>
<tr>
<td>C2</td>
<td>0.670 0.671 0.672</td>
<td>0.55 0.48 0.45</td>
</tr>
<tr>
<td>C3.1</td>
<td>0.691 0.714 0.714</td>
<td>0.63 0.55 0.50</td>
</tr>
<tr>
<td>No control (polling at $T_{200}$ expiry)</td>
<td>0.652 0.558 0.49</td>
<td>0.50 0.43 0.37</td>
</tr>
</tbody>
</table>

Table 5: Effects of Transmission Errors on Implicit Adaptive Window Schemes

$C_s = 64 \text{ kb/s}, W = 16$

<table>
<thead>
<tr>
<th>Control Scheme</th>
<th>Virtual Circuit Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r=0$</td>
</tr>
<tr>
<td>C1.1</td>
<td>0.823</td>
</tr>
<tr>
<td>C1.2</td>
<td>0.823</td>
</tr>
<tr>
<td>C1.3</td>
<td>0.823</td>
</tr>
</tbody>
</table>