EUROCONTROL

ATNP/WG2/
WP/267
19 April 1996

AERONAUTICAL TELECOMMUNICATIONS NETWORK PANEL

WORKING GROUP TWO

Brussels 22.4.96-26.4.96

Report on the Further Investigation of Network Layer Congestion Management in the ATN Internet

Presented By Henk Hof

Prepared by Tony Whyman

SUMMARY

Action 7/9 was accepted by Eurocontrol at the Brisbane WG2 meeting in order to further refine the specification of the Congestion Management Algorithm agreed for the ATN Internet SARPs. This working paper provides a report on work under between the Brisbane Meeting and the subsequent Brussels meeting of WG2 in fulfilment of the action. This work includes further simulation work and analysis of results, and has been undertaken by Eurocontrol, and ESG under a contract from the DFS. Consequently, a detailed proposal can now be made to conclude the specification, and report that a significant proportion of the validation work in this area has been completed. Only the implementation and testing of the proposed algorithms in real implementations is now outstanding.

DOCUMENT CONTROL LOG

SECTION	DATE	REV. NO.	REASON FOR CHANGE OR REFERENCE TO CHANGE
	19-Apr-96	Issue 1.0	

TABLE OF CONTENTS

1. Introduction

1.1 Scope

This paper provides a report on the current validation work draft SARPs for Congestion Management. This work includes an analysis of the proposed Congestion Management algorithm, the development and conduct of validation exercises, and a number of minor change proposals resulting from the validation work.

1.2 References

1.		CNS/ATM-1 Package SARPs and Guidance Material - Sub-Volume 5 Internet Communications Service
2.	WG2/WP255	Simulation Exercises on Congestion Management Techniques
3.	WG2/WP	CNS/ATM-1 Package Internet SARPs Validation Objectives
4.	WG2/WP231	Proposal for Congestion Management Algorithm
5.	WG2/WP254	Comments on the Congestion Avoidance Algorithm proposed for the ATN.
6.	1WF4-BT-005	Scenario Specifications for Simulation Exercises on Congestion Management Techniques

2. Summary

Considerable work has now been performed by Eurocontrol with support from the DFS, in the validation of the Congestion Avoidance algorithm accepted at the Brisbane WG2 meeting. This has taken the form of:

1. Analysis of the algorithm

2. Development and operation of Validation Exercises using simulation models

3. Preparation of the consequential Validation Report and Change Proposals resulting from this activity (i.e. this report).

The analysis has drawn on additional experience on Congestion Avoidance, provided by the DFS, and which has resulted in the work presented in **[5]**. In particular, this analysis has identified problems in and resolutions to:

- When the CE bit is set (i.e. the value of the constant α);

- Transport Layer Interface (NPDU to TPDU relationship);

- The sampling period (in particular, the need to suppress the sampling for a period after a change to the window size);

- Sampling TPDUs other than DT TPDUs (i.e. only DT TPDUs should be sampled);

- The Window decrease factor (i.e. the value of β).

The conclusion of the analysis appeared to be sound and therefore Eurocontrol progressed immediately to the validation of the revised Congestion Avoidance algorithm (i.e. revised after taking into account resolutions to the above). Eurocontrol already had an existing simulation model which had been used during the preparation of **[4]**. This model was updated to include the identified revisions, and a number of simulation exercises developed to test out the Congestion Avoidance specification and, in particular, focusing on the problems identified during the analysis work. The exercise specifications were also developed with DFS support.

The exercise specifications are presented in section 3 of this report, and the analysis of the results in section 4. The exercise specifications also identify which AVOs are covered, with reference to **[3]**.

The first two exercises investigate the operation of the Congestion Avoidance algorithm's effectiveness by simulating several concurrent file transfers between different pairs of End Systems, but with a common "pinch point" in the network. The two exercises provide different connection profiles and are distinguished by the relationship between file transfer start times. The results of these exercises are very encouraging, demonstrating high bandwidth utilisation, good throughput and no retransmissions. The results appear to be independent of connection start times.

The third exercise investigates single file transfers over a limited bandwidth data link to see if the algorithm has a downside effect for short transfers compared with long transfers. The results are compared against the same transfers performed without Congestion Avoidance. The results show that there is a performance penalty to pay for small transfers (e.g 0.5kB), but even so, the result is no worse than when no Congestion Avoidance is employed. For longer file transfers, throughput is almost doubled by the use of the Congestion Avoidance algorithm. Even when only small messages need to be transferred, there still seems good justification for using the Congestion Avoidance algorithm.

The fourth exercise looks at the “fairness” of the algorithm when arbitrating between different types of transfer. The case investigated is when two file transfers share a common “pinch point”, while utilising connections with dissimilar Round Trip Times (RTT). The results of this exercise showed that the algorithm equalised end-to-end transit delay even though the RTT was different. This was not the expected result. The expected result was that Credit Windows would be equalised resulting in a higher throughput for the data transfer with the shorter RTT. This issue was investigated further by changing the parameters of the model to increase the difference in the RTT. This showed that the equalisation of end-to-end transit delay was not a real result and purely a feature of the scenario. However, the data transfer rates were still the reverse of what was expected. This result needs further investigation to see if it is an artefact of the model or scenario, rather than a general result. However, the fact that the algorithm has still successfully arbitrated between the two competing users is a positive result.

The fifth exercise looks at bi-directional traffic. This gave the expected result with poorer throughput resulting from the conflicting demands of DT and AK TPDUs for the same network resources. The loss of throughput is signficant and may be resolved if AK TPDUs are sent at a different priority to DT TPDUs. This would have the advantage of putting them on a separate outgoint queue (hence avoiding often inappropriate congestion experienced reports), and there is also some justification for sending AK TPDUs at a higher priority than DT TPDUs in order to ensure timely responses to changes in the credit window. This will be investigated further.

The sixth exercise looks at how data transfers that do not implement Congestion Avoidance interact with those that do, and is essentially a repeat of the second exercise, but with only one connection using Congestion Avoidance. The result is predictable. Throughput is much reduced and the data transfer rates are very unstable. Not only does this demonstrate the benefits of Congestion Avoidance, but there is clear justification for making it mandatory, as traffic not implementing Congestion Avoidance will have a significant and disruptive effect on the ATN.

The results of these exercises appear to clearly demonstrate the benefits of the Congestion Avoidance algorithm incorporating the modifications proposed in **[5]**. Further investigation will be made into the effects observed in exercise 4 and the value of sending AK TPDUs at a higher priority.

3. Simulation Exercises Specifications

3.1 Simulation Objectives

Assuming the proposals made in **[3]** are accepted then, the following validation objectives are application to this subject.

AVO_448	Evaluate the behaviour of the system in case of congestion (lack of resources at the destination node)	
AVO_449	Verify that the congestion control (congestion avoidance) prevents deadlock in the network.	
AVO_450	Evaluate the impact of congestion on the user service of communication.	
AVO_470	Evaluate the performance of receiver based congestion management over each class of air/ground subnetwork, when an adjacency is supported by more than one class of air/ground subnetwork simultaneously and when no subnetwork preference is given i.e. when an NPDU may go over any of the available subnetwork connections. The evaluation should aim to determine the conditions by which the required QoS is maintained even when congestion occurs.	
AVO_471	Evaluate the impact of congestion, when using receiver based congestion management, on transport connections with different end-to-end path lengths, but which share a congested path segment.	
AVO_472	Validate that when the receiver based congestion management algorithm is used, higher priority transport connections remain unaffected by the network congestion until the congestion reaches the point that the network service is effectively lost to lower priority transport connections.	
AVO_473	Evaluate the importance of an accurate measurement of the round trip delay for effective use of the receiver based congestion management algorithm, and the consequences of mobility i.e. when the round trip time changes significantly due to a change in the point of attachment or air/ground subnetwork used.	

Eurocontrol has developed a simulations model in support of the validation of these objectives. Currently, the model does not support simulation of air/ground data links or priority; these are the subject of later additions. However, the model is still adequate to investigate the impact of the changes proposed in **[5]**, in a ground-ground scenario, and the necessary modifications to the model have been made in order to do this.

A number of exercises have also been defined in order to validate the above AVOs and those appropriate to the current model are specified below. Each is intended to test a particular aspect of a Congestion Management algorithm. In general, each sender transmits data in portions of max. 500 bytes for each DATA-TPDU (this best matches packet sizes measured in the Internet). The overall goal is to achieve a high throughput, low end-to-end delay with a fair sharing of the bottleneck resource among competing users (if such ones do

exist). Finally, all these user related measures should provide good results without too high a load found within the bottleneck device, whose buffer load is thus also observed.

3.1.1 Interpretation of Results

The primary goal of Congestion Management algorithms is to prevent any part of a network from getting congested due to high traffic loads, while simultaneously maintaining a reasonable quality of service offered to the users. The performance of such an algorithm thus can be judged from two different viewpoints:

- from the point of view of a network user (who experiences a certain quality of service when using the network), and

- from the point of view of a network operator, who wishes to preserve a stable operation of the network even during times of high load.

Both aspects are equally important to consider. For both areas, the measures considered to be most interesting when studying a Congestion Management algorithm shall be described.

3.1.1.1 User-Related Performance Measures

From the perspective of a network user, clearly the **throughput** achievable is a very important performance measure. This can best be measured (and compared for different variants of the Congestion Avoidance algorithm, e.g. derived by varying parameter settings) by determining the ***total file transfer time***. The total file transfer time is defined as the interval between the instant at which the first byte of data is available at the transport-level of the sender, and the instant at which the sender receives the acknowledgement that the receiver has the last byte of the file.

While the first performance measure is oriented towards transfer of large amounts of data, a user might also be interested in transmitting short messages to other users within the network. For these short messages, the throughput achievable is not of primary interest, but the **end-to-end delay** a message experiences. This can roughly be determined by measuring the time between the instant when a transport packet is given to the underlying network layer for transmission, and the time when this packet arrives at the receiving transport entity. Since message oriented data transfers normally do not lead to the exchange of huge amount of data, it is assumed that no data is queued within the sending transport layer entity. Thus, the message transfer delay experienced by a user (on top of the transport layer) will only be increased by the processing time within the transport layer, which is assumed to be significantly smaller than the total time required to forward the packet to its destination. So the approximation proposed here is assumed to be valid, and only a single kind of data source (namely file transfer) is required to determine both kind of performance measures. For the end-to-end delay, both the average, minimum, maximum, standard-deviation, and 99% quantile should be gathered.

Finally, a user expects to be treated fairly by the underlying network service. That means, given several users competing for a bottleneck resource (e.g. a router or a link), each one will receive the same share of this resource, independent of his actual operating conditions (e.g. application packet size used, path length between sender and receiver, and so on). Thus the third important performance measure from the user's point of view is **Fairness**. The degree of fairness achieved can be determined by simulating the transfer of files of identical size by a number of sources, and comparing both the throughput achieved and the end-to-end delay experienced by each of them (which approximately should be identical).

3.1.1.2 Network-Related Performance Measures

Beside user satisfaction, a network operator is primarily interested in a good utilisation of all network resources, and a stable operation of the network as a whole even in the presence of high traffic loads.

The most important measure to be used in this area is the **buffer load found within the bottleneck device**. A high buffer load requires large amount of buffers, increasing the cost for a particular device. It furthermore increases the risk of packet loss due to buffer shortage, which will trigger retransmission of packets. The latter will both put an additional load on the network, increase the time required to complete e.g. a file transfer, and reduce the overall throughput experienced by network users. In the worst case, repeated packet losses due to buffer shortage might lead to a condition known as "Congestion Collapse", where almost all data is transmitted several times before it eventually reaches its destination. The throughput experienced by a user might be less than 0.1% of the optimum throughput achievable when the network has entered such a condition. To allow a detailed insight into the operation of the network, both the average and maximum buffer load, its standard deviation, and the 99% quantile should be gathered during the simulation run.

Finally, to determine how efficient the network is operated, the ratio of the number of successful TPDUs transmitted to the total number of TPDUs transmitted should be computed for each file transfer. If there are no packet losses within the network, both will be identical. A packet loss, however, requires retransmission of some TPDUs, thus the total number of TPDUs transmitted will become larger than the number of successfully transmitted TPDUs (i.e. TPDUs that do contain 'new' data portions, i.e. data not yet seen by the receiving transport entity).

3.1.1.3 Summary

In summary, the following results are to be provided by each exercise:

1. Total File Transfer Time
2. A Range of transfer times for different size files (where appropriate)
3. The end-to-end packet delay (average, min, max, std dev, 99%)
4. IS Buffer Load (average, min, max, std dev, 99%)
5. Packet Loss Ratio.

3.2 Exercise 1: Slowly Changing Cross Traffic

3.2.1 AVOs Covered

448, 449

3.2.2 Scenario Description

Due to the need to observe network behaviour after the network load has been modified, the fundamental time constant for every Congestion Management algorithm to react on changes in network state is one round trip time (RTT for short). Figure 3-1 illustrates this.

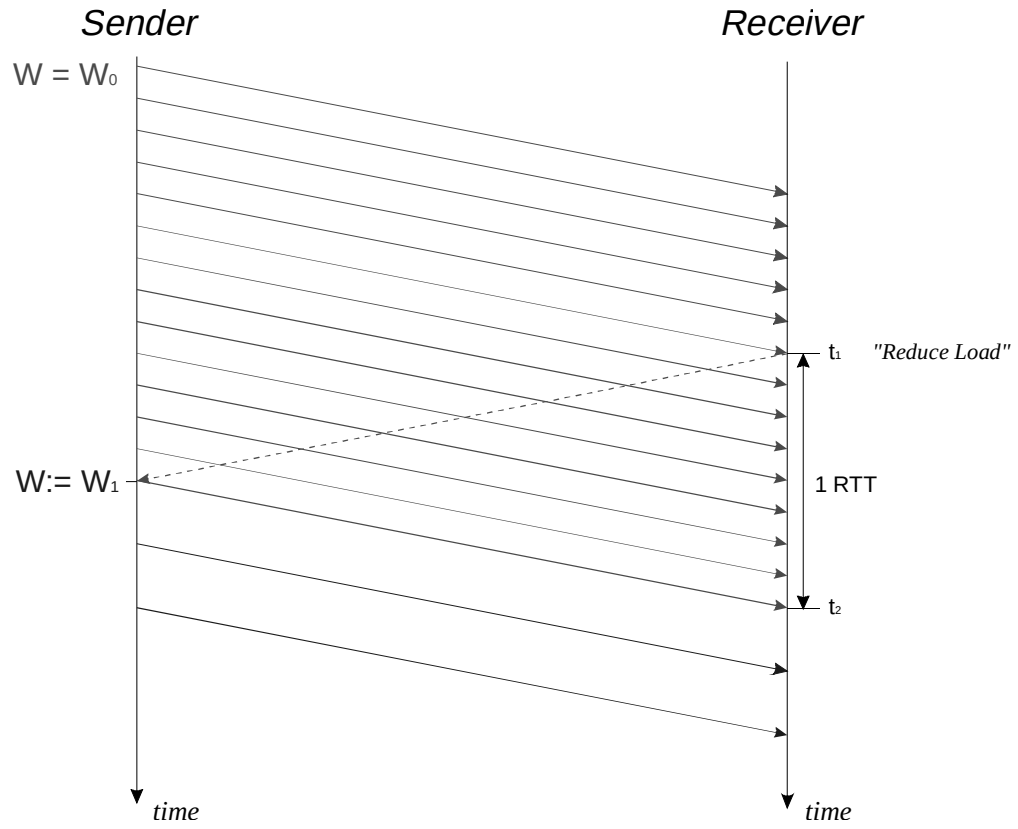

Figure 3-1 Window Adaptation Frequency

The figure depicts a sender transmitting data packets to a receiver, using a window of size W_0. It takes a certain time, until such a packet arrives at the receiver (time is proceeding from top to bottom). Now, assume at time t_1, the receiver decides that the load has to be reduced in order not to overload the network (i.e. it has a sufficient number of Congestion Experienced flags received). It will thus transmit an appropriate signal to the sender (indicated by the dashed arrow going from right to left), who in turn will reduce its load upon reception of the signal by reducing its window size to $W_1 < W_0$. However, it will take a certain time until the first packet transmitted with a smaller load (indicated by a larger spacing of the arrows) will reach the sender. As can be seen from the figure, the receiver will continue to receive packets transmitted with the "old" load level until time t_2, i.e. for approximately one round trip time. Thus even if the sender immediately knew after reception of the first packet transmitted with a smaller load whether to ask the sender to increase or decrease its load, it will not be able to make such a decision prior to one RTT after the last such request has been transmitted to the sender. In practice, however, the receiver will just start to evaluate new Congestion Experienced flags after it receives useful new information (i.e. at time t_2, or approximately after the reception of as many packets as were contained in the old window size that was advertised to the sender prior to asking for a load reduction at time t_1, i.e. W_0).

If the load found within the network changes exactly at times spaced apart one RTT, Congestion Control algorithms are heavily stressed. So the first scenario will exactly test this situation. Figure 3-2 depicts the scenario to be used.

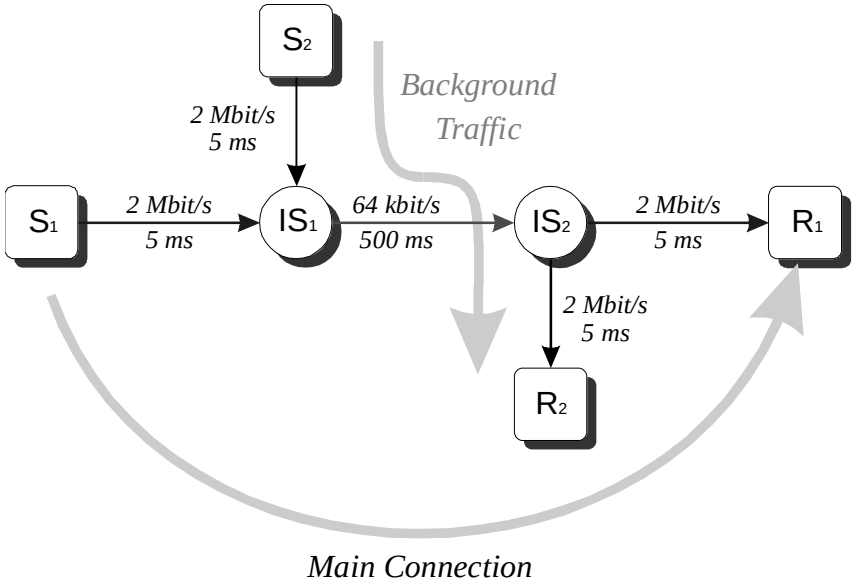

Figure 3-2 Changing Cross-Traffic

The main connection between S1 and R1 starts up at time zero, and has 256 Kbytes of data to transmit. The first cross-traffic stream (between S2 and R2) starts at 2 s, competing with the main connection for the 64 kbit/s bottleneck link. One by one, four more cross-traffic streams (also between S2 and R2) come up at intervals of 1020 ms (which is the RTT for all connections), further reducing the bandwidth available to each of the connections now sharing the bottleneck link.

Each cross-traffic stream has 64 Kbytes of data to transmit. The file sizes are chosen such that the cross-traffic load gradually increases up to a maximum, and then decreases back to zero during the interval in which the main connection is transmitting data.

3.3 Exercise 2: Sudden Jumps in Cross-Traffic

3.3.1 AVOs Covered

448, 449

3.3.2 Scenario Description

If traffic load changes faster than the Congestion Control algorithm can react, packets will have to be buffered within the bottleneck device. This is also an important aspect, since it gives an indication how stable the network can be operated in the presence of heavy load fluctuations, and how much buffers will be required within the intermediate systems.

To evaluate the behaviour of the Congestion Avoidance algorithm under these conditions, essentially the same scenario as shown in Figure 3-2 can be used. However, cross-traffic streams now have to come up at intervals of 20 ms, so the main connection experiences a large drop in available bandwidth within one RTT.

3.4 Exercise 3: Transfer Time for Short Files

3.4.1 AVOs Covered

450

3.4.2 Scenario Description

In order not to overload the bottleneck's buffer space when sudden load changes do occur (e.g. several connections start up within one RTT, so the Congestion Control algorithm will not have sufficient time to react), all Congestion Control algorithms need to perform some kind of "Slow-Start". They start with a small load, gradually increasing the load until the bandwidth available is fully used.

However, if there is only a small file to transmit, its transmission time will adversely be affected since transmission might be finished prior to the time the connection was able to make full use of the underlying bandwidth (i.e. the whole file will be transmitted in the "Slow-Start" phase, where the sender will only transmit with a small rate).

Figure 3-3 depicts the scenario used to evaluate the additional time required due to the "Slow-Start" phase of the Congestion Avoidance algorithm.

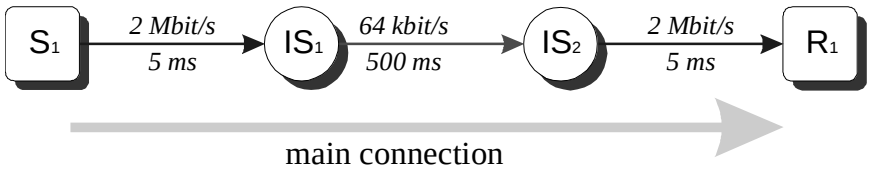

Figure 3-3 Transfer Time for short Files

This scenario is intended to isolate the effects the "Slow-Start" phase of a Congestion Control algorithm has on the transfer time for short files. Therefore, no additional cross-traffic is taken into account. The experiment shall be run for files of sizes

500 byte, 1 Kbytes, 2 Kbytes, 4 Kbytes, 8 Kbytes, 16 Kbytes, 32 Kbytes, 64 Kbytes, 128 Kbytes

The actual time required to transmit the whole frame is compared to the minimum time required, provided the sender would know the bottleneck capacity of 64 kbit/s. To transfer the file without putting too high a burden on the bottleneck system IS_1, and assuming a Round Trip Time of 1040ms, the sender should use a window of size

$$64000\ \text{bit/s} \times 1.040\ \text{s} = 66560\ \text{bit} = 8320\ \text{byte} \approx 17\ [\text{Pkt}]$$

The results obtained using this window size (without the Congestion Avoidance algorithm activated) shall be compared to the ones obtained when the Congestion Avoidance algorithm is in use.

3.5 Exercise 4: Evaluation of "Fairness"

3.5.1 AVOs Covered

450, 471

3.5.2 Scenario Description

This scenario is intended to determine the degree of fairness the Congestion Avoidance algorithm is able to ensure. Fairness here means an equal sharing of the total bandwidth available at the bottleneck device.

It is known from previous research in the area of Congestion Management algorithms, that the adaptation of a window (instead of the transmission rate) is likely to cause problems, if competing users have different path lengths (i.e. round trip times). As the simplest case, two

competing users are considered having different path lengths, that compete for some bottleneck device. The corresponding scenario is shown in Figure 3-4.

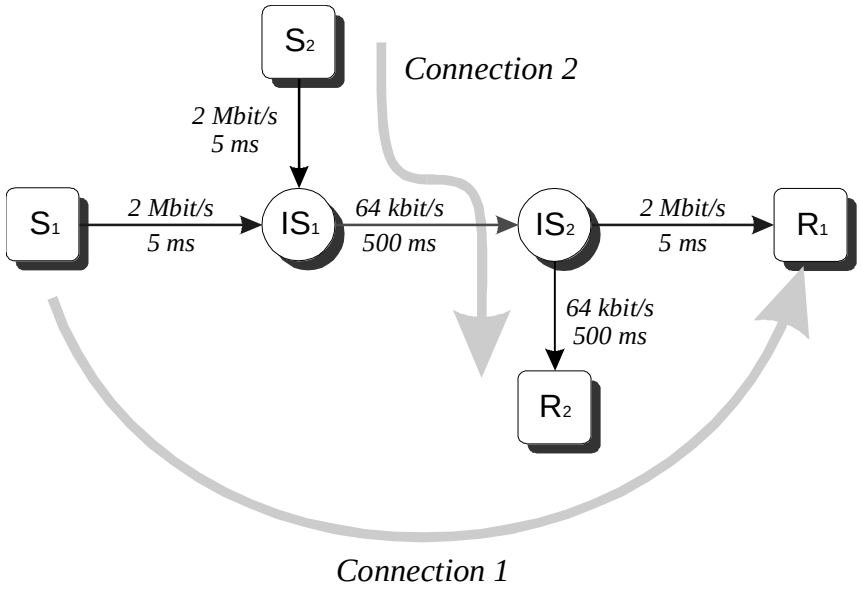

Figure 3-4 Fairness among competing Users

During this exercise, two connections both start at time zero and have 256 Kbytes of data to transmit. Connection 2 uses a path with approximately twice the RTT of that of connection 1. Both are competing for the bottleneck link between IS_1 and IS_2. Of primary interest is the total file transfer time of both senders. If the bottleneck link is shared evenly among the competing connections, both should need the same time to transmit the complete file.

3.6 Exercise 5: Bi-directional Data Transfer

3.6.1 AVOs Covered

448, 449, 450

3.6.2 Scenario Description

Another well-known problem of many Congestion Control algorithms is caused by traffic along the reverse path. If data packets are transmitted along the reverse path, they will keep the intermediate system busy for some time. Acknowledgements arriving during that time will get queued, waiting for the IS to become available again. As soon as the system becomes free, these acknowledgements are transmitted back-to-back. This can have some adverse influence on the operation of the Congestion Control algorithm (e.g. leading to bursts of data packets emitted by one of the senders).

Figure 3-5 depicts the scenario used to perform this exercise.

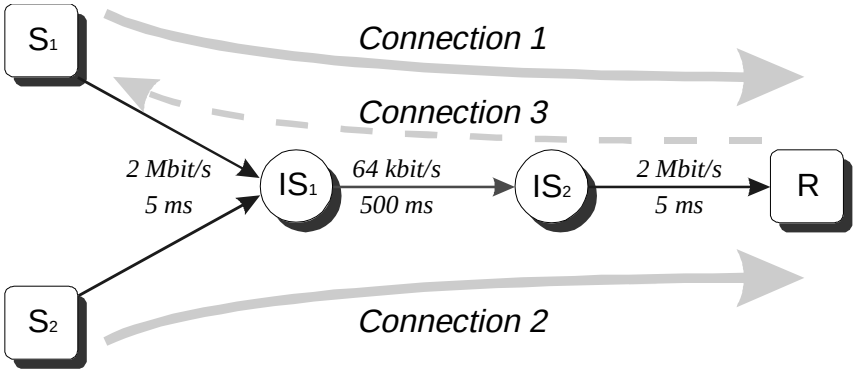

Figure 3-5 Two-way Traffic

Within this exercise, three connections are starting at time zero to transmit. Connection 1 is transmitting a file of size 128 Kbytes to the receiver R, as is connection 2. Both do compete for the shared link between IS_1 and IS_2. At the same time, R is transmitting a file of size 256 Kbytes back to S_1. Note that, for the reverse traffic, the bottleneck is at system IS_2. Data packets coming from R will keep this system busy for a while, forcing acknowledgements for both S_1 and S_2 to be queued behind.

3.7 Exercise 6: Non-compliant Cross Traffic

3.7.1 AVOs Covered

450, 471

3.7.2 Scenario Description

The final exercise considers non-compliant traffic, i.e. what happens within the network if not all senders do behave according to the Congestion Control algorithm (e.g. because one is transmitting voice, which is quite likely not to get controlled in the same way as data traffic).

The configuration used for this exercise is identical to that shown in Figure 3-2. The only difference is that cross-traffic streams are no more controlled using the Congestion Avoidance algorithm, and the packet arrival process for each cross-traffic stream now is Poisson (i.e. exponentially distributed interarrival times between any two packets) such that the aggregate arrival rate of all the cross connections will consume 80% of the bottleneck link bandwidth. As described in 3.2, the main connection will start first at time zero, with one cross-traffic stream switched on every 1020 ms, starting at 2 s.

4. Presentation of Simulation Results

4.1 Exercise 1: Slowly Changing Cross Traffic

This exercise was conducted as specified in 3.2, except that it was not possible within the available timescale to simulate the 500ms delay on the slow data link, or to calculate “99%” figures (mean, min, max. and standard deviation are available where appropriate A 256kbyte file was transferred between two systems (Snd1 and Rcv1) and five 64kbyte files were transferred between systems Snd2 and Rcv2, according to the specified scenario. The measured file transfer times and throughputs are as given below:

Transfer Number	From	To	File Size (KB)	Total Transfer Time (Secs)	Throughput (KB/s)
1	Snd1	Rcv1	256	81.3006	3.149
2	Snd2	Rcv2	64	52.1678	1.227
3	Snd2	Rcv2	64	52.6122	1.216
4	Snd2	Rcv2	64	52.4090	1.221
5	Snd2	Rcv2	64	52.4522	1.220
6	Snd2	Rcv2	64	51.7613	1.236

Table 4-1 File Transfer Times for Slowly Changing Cross-Traffic

The data transfers can be seen graphically in Figure 4-1. As was expected, the transfer rate for the first file rises rapidly until the other file transfers come on stream. It is then slowed down and the algorithm ensures fairness of resource allocation, with all transfers proceeding at the same rate. Once the cross-traffic ceases, then the first connection speeds up to take up the freed resources and completes at a much faster rate. This is born out by the throughput figures given above, where the cross-traffic achieves approximately 1.22kB/s, while the first connection achives an overall higher transfer rate, as it has the data link to itself for part of the exercise. The fairness of the algorithm is born out by looking at Figure 4-2, which shows how the window sizes are equalised across all connections during the period in which cross-traffic is experienced.

During the period of near congestion, the data transfer rate on the first connection may also be assumed to be approximately 1.22kB/s and, from this, it may be concluded that the data link utilisation is:

(6 * 1.22)/(64/8) = 91.5%

A good result.

Figure 4-3 Illustrates the end-to-end delay experienced during the exercise, for data transfers between each pair of End Systems. Note how in both cases an oscillation around a mean figure is seen, and this is continued for the first pair (Snd1, Rcv1), even after the other connections have been terminated. This is expected and illustrates how the algorithm is constantly trying to find the optimal window size. Table 4-2 gives the precise figures during each phase of the exercise.

From	To	Transfer Phase	Mean Delay (Secs)	Min. Delay (Secs)	Max. Delay (Secs)	Std Dev.
Snd1	Rcv1	During Cross Traffic	0.32265	0.24313	0.51254	0.09745
Snd1	Rcv1	Cross Traffic Completed	0.27084	0.22913	0.31405	0.03355
Snd2	Rcv2	During Cross Traffic	0.35781	0.21242	0.53254	0.08811

Table 4-2 End to End Delays for Slowly Changing Cross Traffic

Figure 3-1 shows the buffer load experienced in the congested router. This is as expected, with characteristic peak buffer loads occuring during the period of cross-traffic, and showing up as spikes. It is during such peaks that the CE-bit will be set, forcing a reduction in the traffic load. Such spikes in fact conceal a relatively stable situation in the router, which is illustrated in Figure 4-5. The algorithm ensures that the buffer loading is relatively constant. Small variations are visible during the period of cross-traffic, and note how it does not actually drop by very much, even when only a single connection is operating. The algorithm is always ensuring efficient utilistation of the router. Table 4-3 provides the detailed figures.

Transfer Phase	Mean Loading	Min. Loading	Max. Loading	Std Dev.
During Cross Traffic	5,568.41	0	21,992	6,389.24
No Cross Traffic	3,235.45	0	8,984	3,371.39

Table 4-3 Router Buffer Loading with Slowly Changing Cross-Traffic (in Bits)

Finally, the number of TPDUs transferred and the retransmission ratio can be inspected to see if the algorithm really does minimise packet loss and the need to re-transmit, which is believed to be its crucial advantage over the “Slow-Start” algorithm. There where found to be no re-transmissions due to congestion illustrating one of the key advantages of the algorithm.

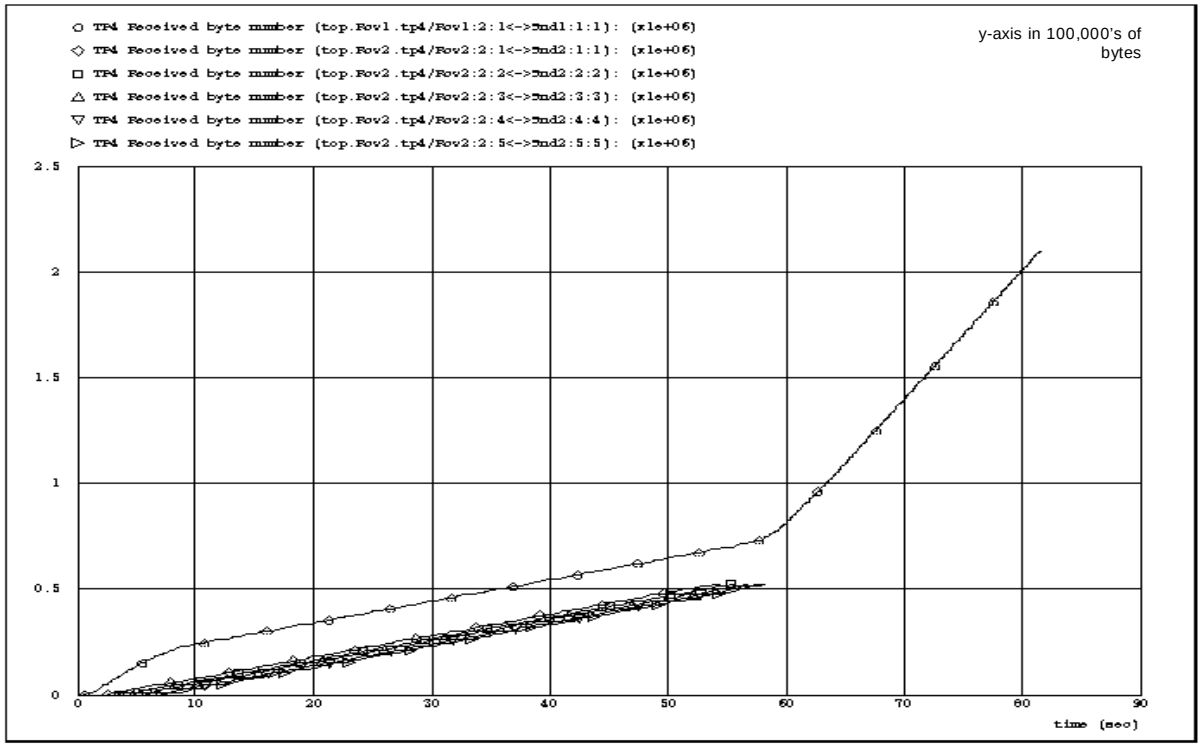

Figure 4-1 Received Byte Counts with Slowly Chaning Cross Traffic

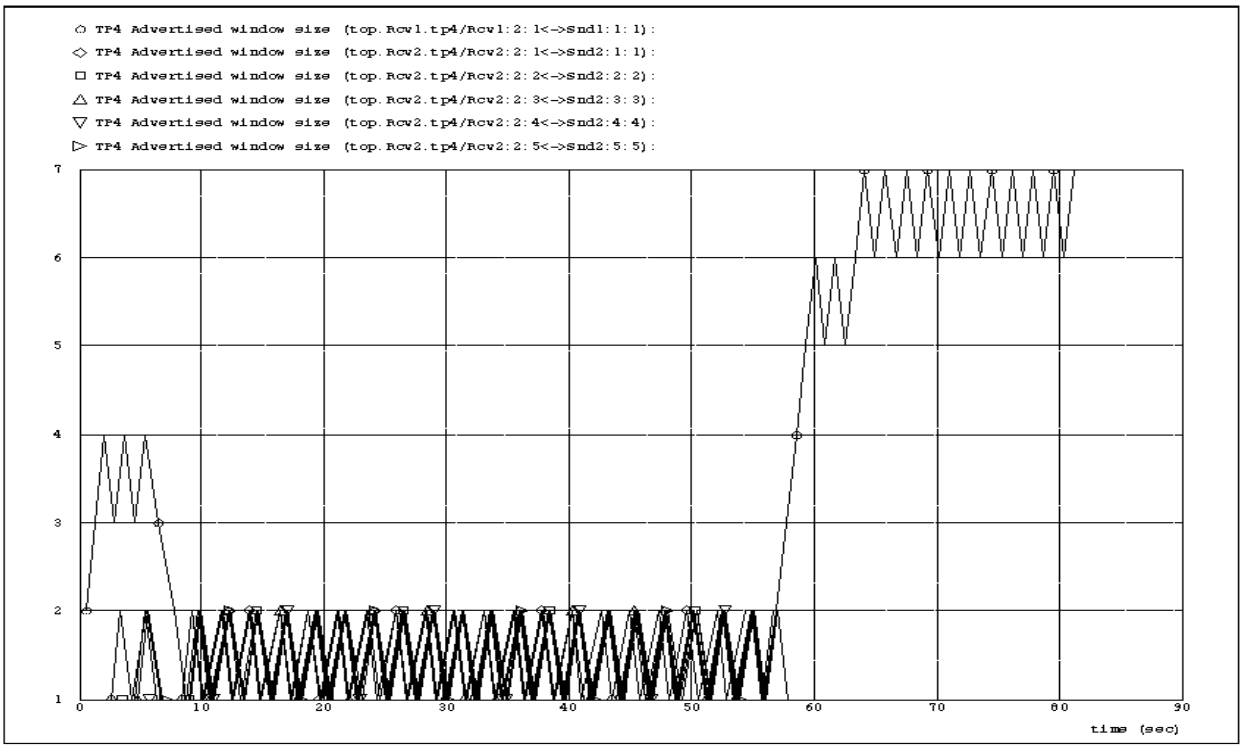

Figure 4-2 Window Sizes for Slowly Changing Cross Traffic

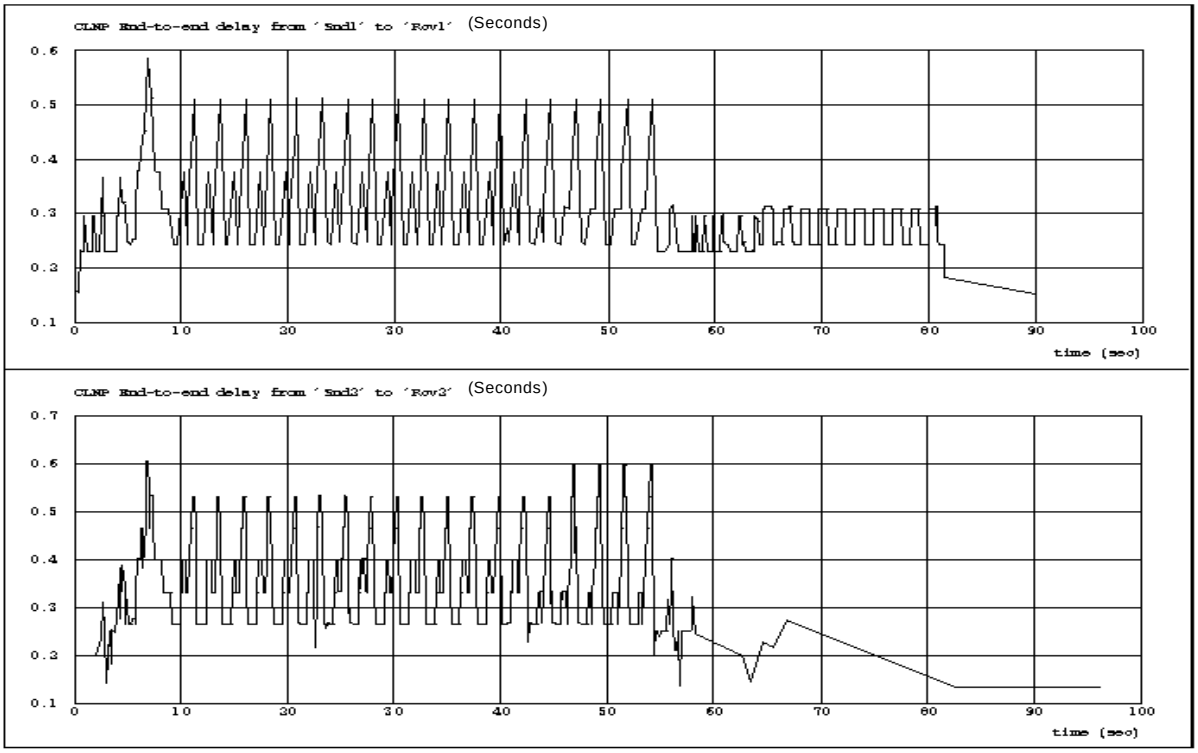

Figure 4-3 End to End Delay with Slowly Changing Cross Traffic

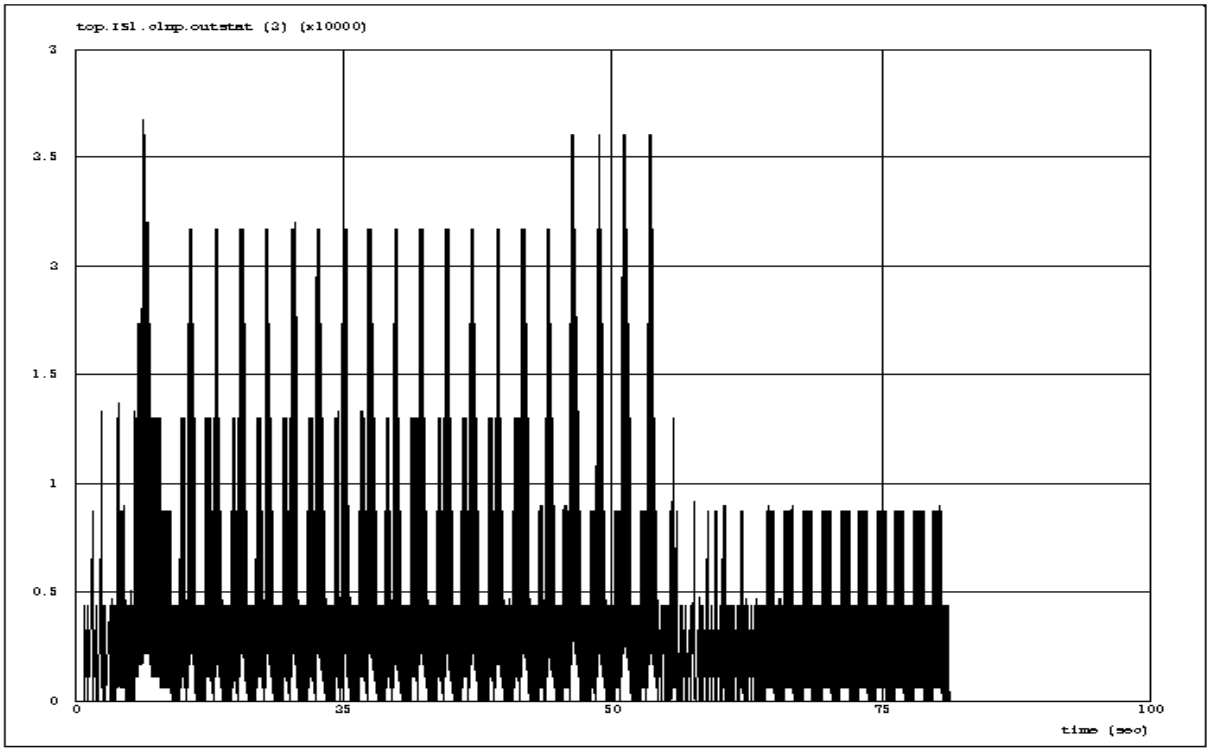

Figure 4-4 Router Buffer Load with Slowly Changing Cross Traffic

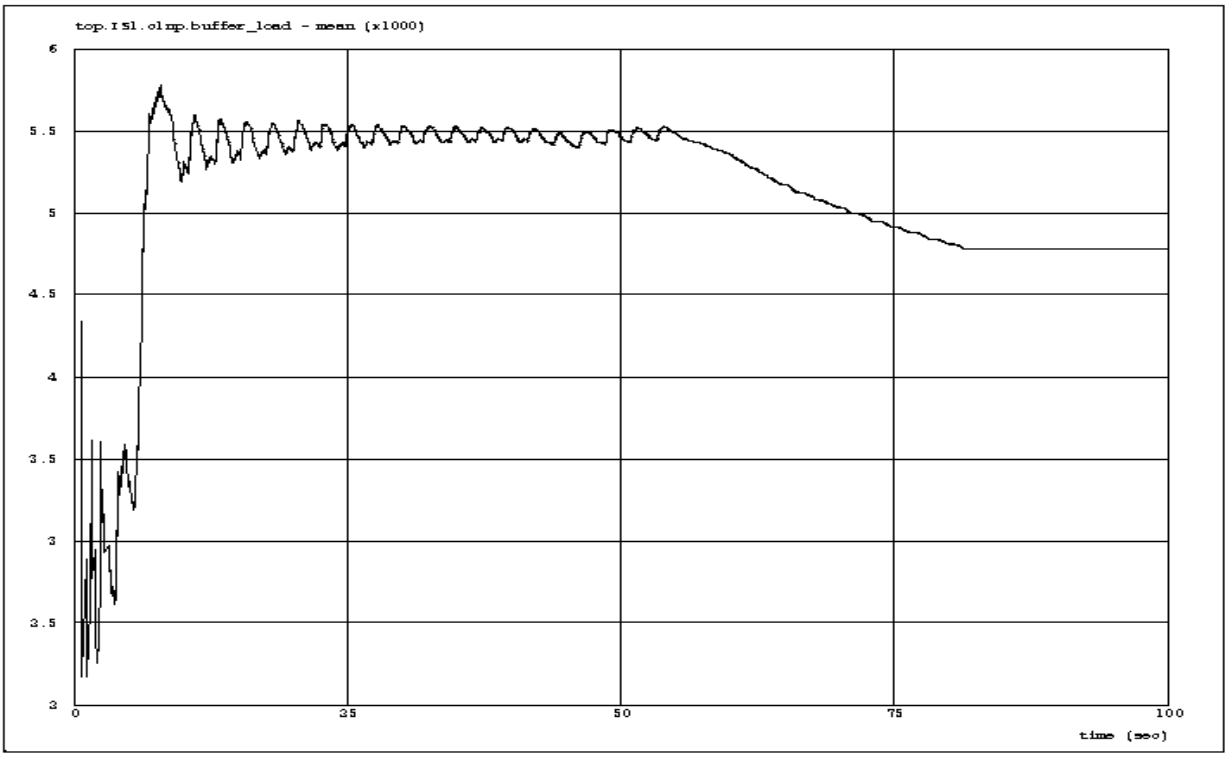

Figure 4-5 Mean Router Buffer Load with Slowly Chaning Cross Traffic

4.2 Exercise 2: Sudden Jumps in Cross-Traffic

This exercise was conducted as specified in 3.3, except that it was not possible within the available timescale to simulate the 500ms delay on the slow data link, or to calculate "99%" figures (mean, min, max. and standard deviation are available where appropriate. The file transfers are now started at intervals of 20ms. A 256kbyte file was transferred between two systems (Snd1 and Rcv1) and five 64kbyte files were transferred between systems Snd2 and Rcv2, according to the specified scenario. The measured file transfer times and throughputs are as given below:

Transfer Number	From	To	File Size (KB)	Total Transfer Time (Secs)	Throughput (KB/s)
1	Snd1	Rcv1	256	80.4010	3.184
2	Snd2	Rcv2	64	51.5893	1.241
3	Snd2	Rcv2	64	51.2597	1.249
4	Snd2	Rcv2	64	51.9535	1.232
5	Snd2	Rcv2	64	51.8667	1.234
6	Snd2	Rcv2	64	51.8493	1.234

Table 4-4 File Transfer Times for Sudden Jumps in Cross-Traffic

The data transfers are shown graphically in Figure 4-6. The result is very similar to the previous exercise, showing that the result is largely independent of when the file transfers start. In this case, all transfers proceed at the same rate until only transfer #1 is left when it completes at the fastest possible rate. Note that the file transfer rates given in Table 4-4, are actually a small improvement over the results from the previous exercise. This is probably due to a more rapid convergence on to equal window sizes, rather than a series of largely separate convergences as each new connection was started. The fairness of the algorithm is again confirmed by inspection of Figure 4-7, which shows that all connections are given the same window sizes.

Figure 4-8 illustrates the end to end delay for each pair of communicating systems. This is very similar to the result achieved by the previous exercise and is indeed smoother during the early phase of the exercise. This confirms the view taken above to explain the slightly faster data transfer rates; there is indeed a more rapid convergence on the optimal window size and a mean end-to-end transit delay. Table 4-5 gives the precise figures during each phases of the connection.

From	To	Transfer Phase	Mean Delay (Secs)	Min. Delay (Secs)	Max. Delay (Secs)	Std Dev.
Snd1	Rcv1	Cross Traffic	0.33267	0.24313	0.50778	0.08123
Snd1	Rcv1	Cross Traffic Completed	0.27248	0.18545	0.30929	0.03384
Snd2	Rcv2	Cross Traffic	0.36317	0.20888	0.59394	0.10288

Table 4-5 End to End Delays for Sudden Changes in Cross Traffic

Figure 4-9 shows the buffer load experienced in the congested router. This is as would be expected. Figure 4-10 illustrates the mean router buffer loading, and this shows the very heavy initial load on the router caused by the near simultaneous start up of each of the connections, followed by the rapid convergence onto a stable situation. Table 4-6 provides the detailed figures.

Transfer Phase	Mean Loading	Min. Loading	Max. Loading	Std Dev.
Cross Traffic	5,850.03	0	26,016	6,836.58
Cross-Traffic Completed	3,252.50	0	8,672	3,391.87

Table 4-6 Router Buffer Loading with Slowly Changing Cross-Traffic

No re-transmissions of DT TPDUs were observed..

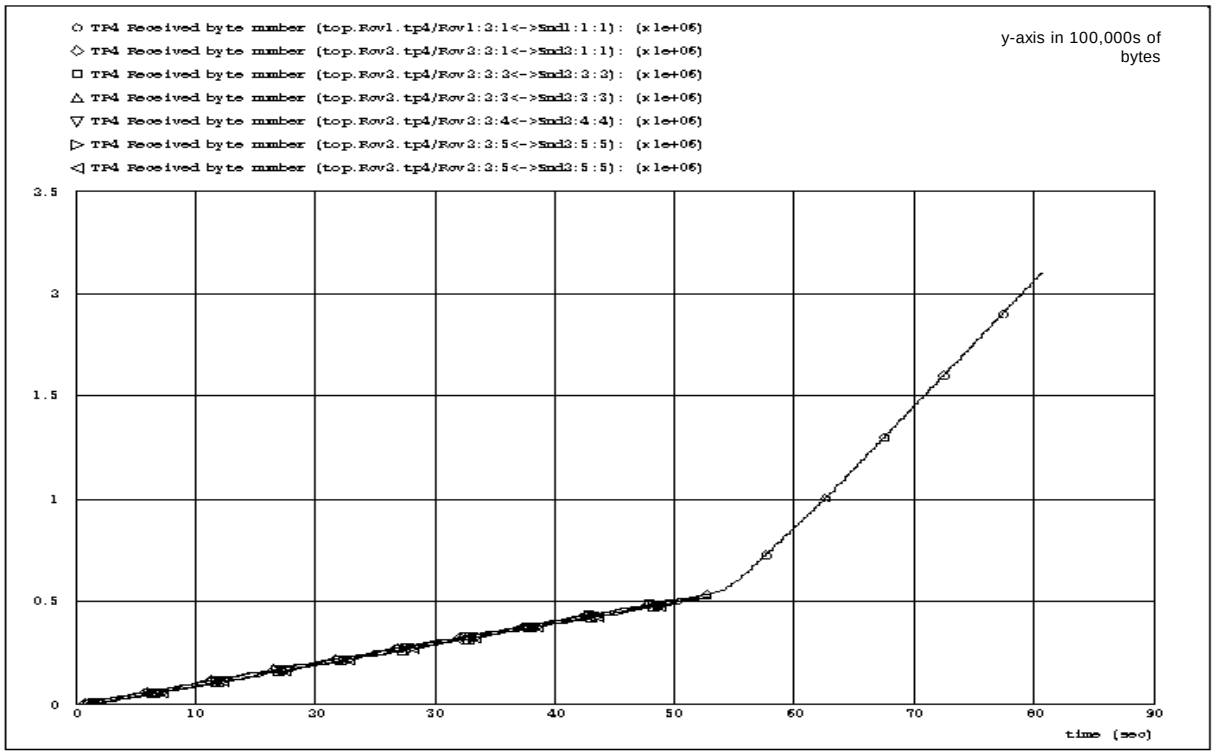

Figure 4-6 Received Byte Counts with Sudden Changes in Cross Traffic

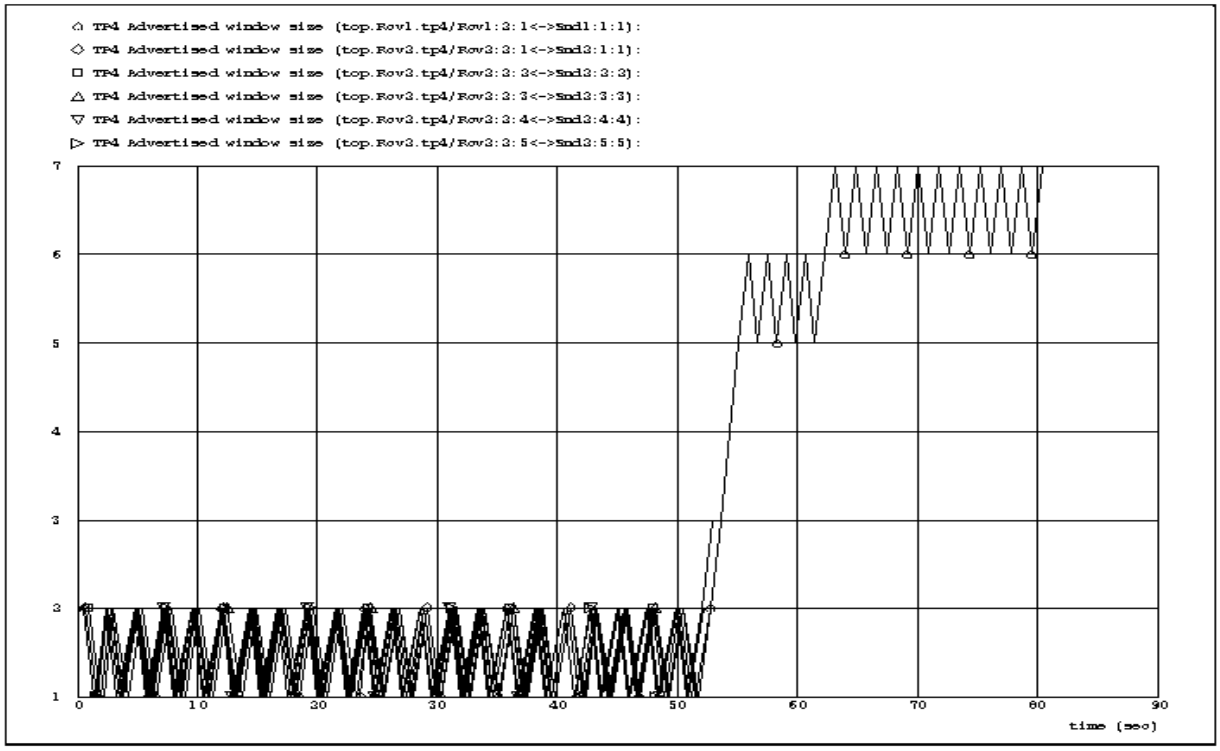

Figure 4-7 Window Sizes for Sudden Changes in Cross-Traffic

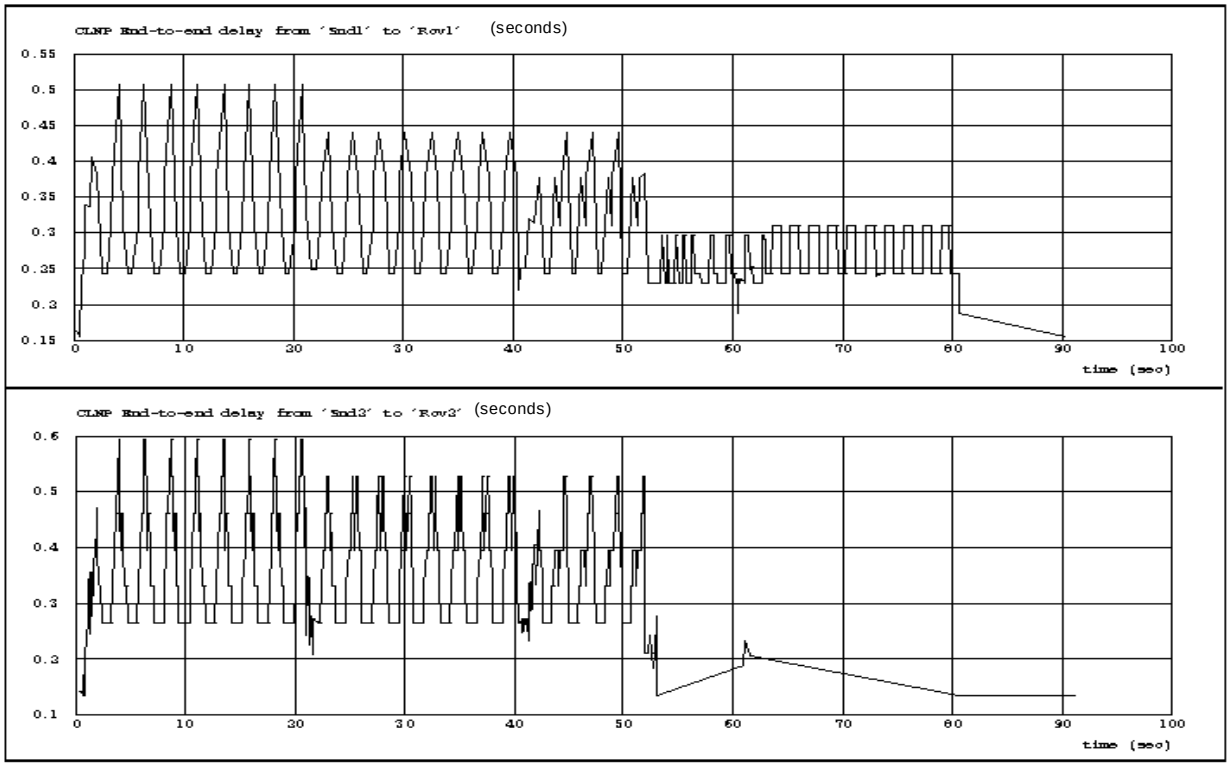

Figure 4-8 End to End Delay with Sudden Changes in Cross-Traffic

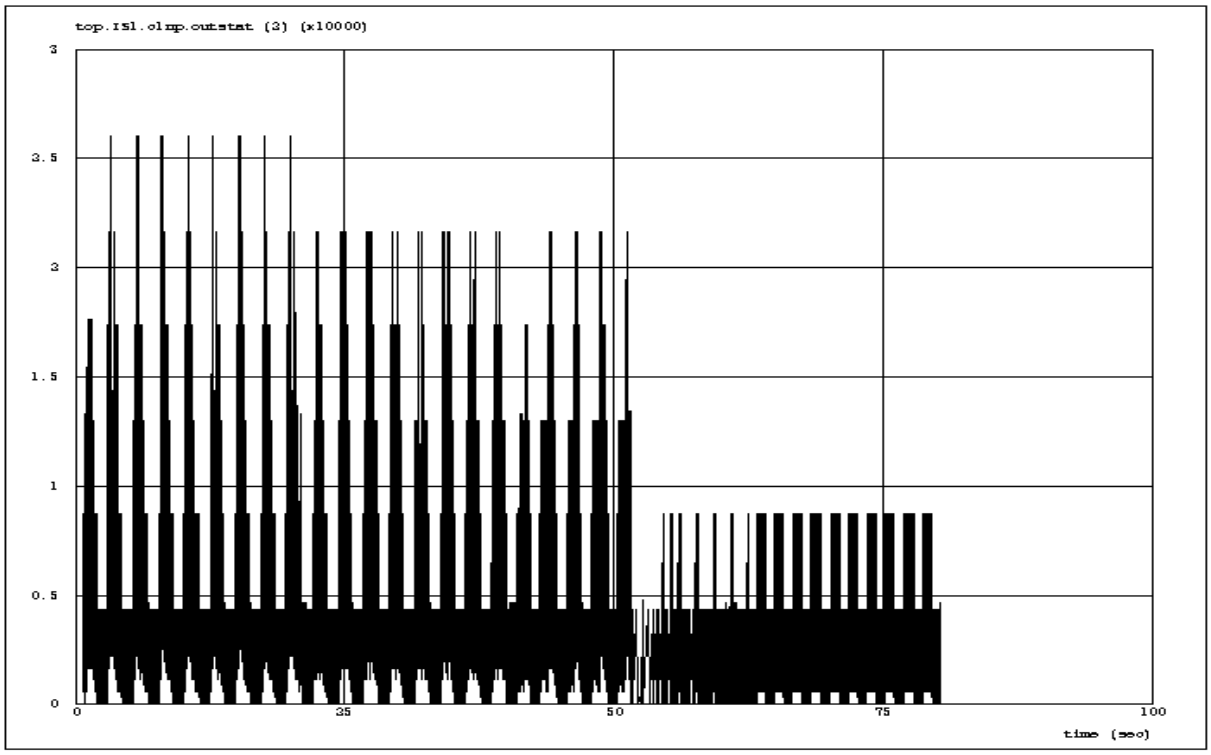

Figure 4-9 Router Buffer Load with Sudden Changes in Cross-Traffic

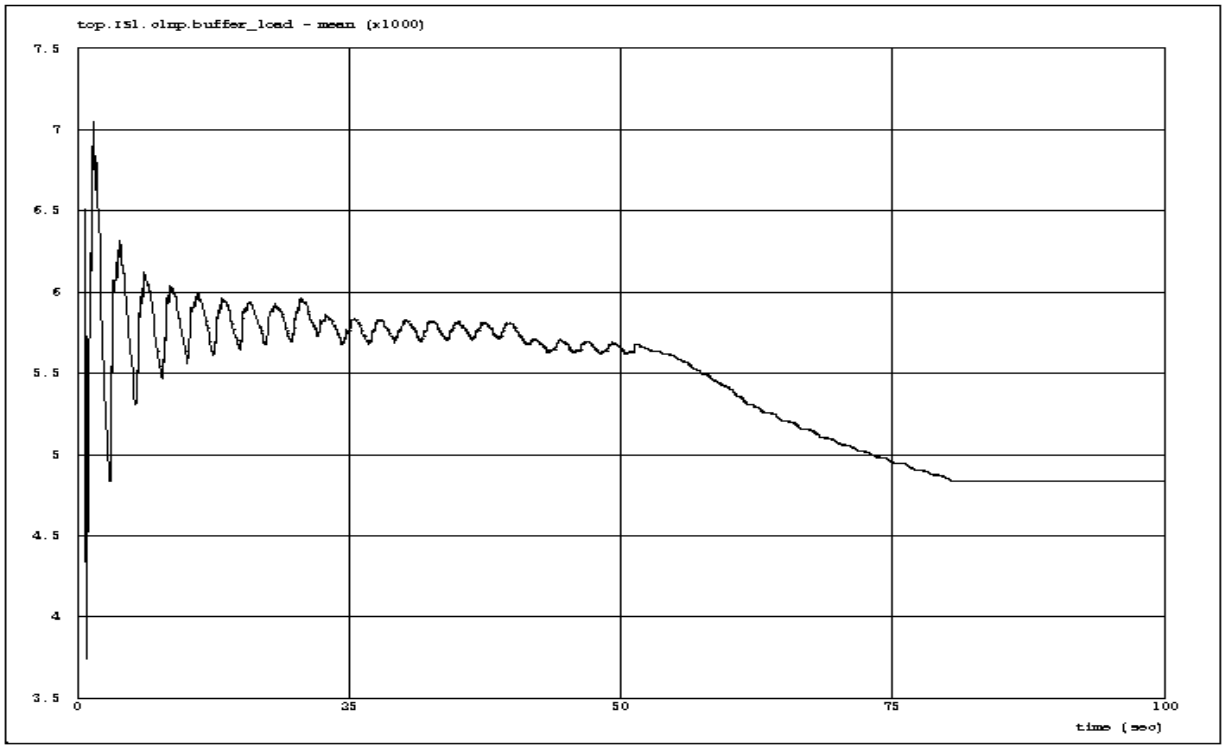

Figure 4-10 Mean Router Buffer Load with Sudden Changes in Cross Traffic

4.3 Exercise 3: Transfer of Short Files

This exercise was conducted as specified in 3.4, except that it was not possible within the available timescale to simulate the 500ms delay on the slow data link, or to calculate "99%" figures (mean, min, max. and standard deviation are available where appropriate. No cross traffic was simulated and the results were obtained for single file transfers. The measured file transfer times and throughputs are given below, with results calculated with Congestion Avoidance Enabled and Disabled.:

				Congestion Avoidance		No Congestion Avoidance	
Transfer Number	From	To	File Size (KB)	Total Transfer Time (Secs)	Throughput (KB/s)	Total Transfer Time (Secs)	Throughput (KB/s)
1	Snd1	Rcv1	0.5	0.38457	1.300	0.3846	1.300147
2	Snd1	Rcv1	1	0.45836	2.182	0.7752	1.290028
3	Snd1	Rcv1	2	0.59077	3.385	1.1582	1.726773
4	Snd1	Rcv1	4	0.85468	4.680	1.8512	2.160761
5	Snd1	Rcv1	8	1.39267	5.744	2.69809	2.965060
6	Snd1	Rcv1	16	2.46866	6.481	4.44587	3.598846
7	Snd1	Rcv1	32	4.62654	6.917	8.31632	3.847856
8	Snd1	Rcv1	64	8.92711	7.169	15.6069	4.100750
9	Snd1	Rcv1	128	17.5283	7.302	28.6443	4.468603

Table 4-7 File Transfer Times with Congestion Avoidance

The results are very clearly illustrated in Figure 4-11. Even with only a 1kB file, there is a significant improvement in throughput with Congestion Avoidance, and the throughput is almost doubled for larger files. The is little doubting the benefits of Congestion Avoidance.

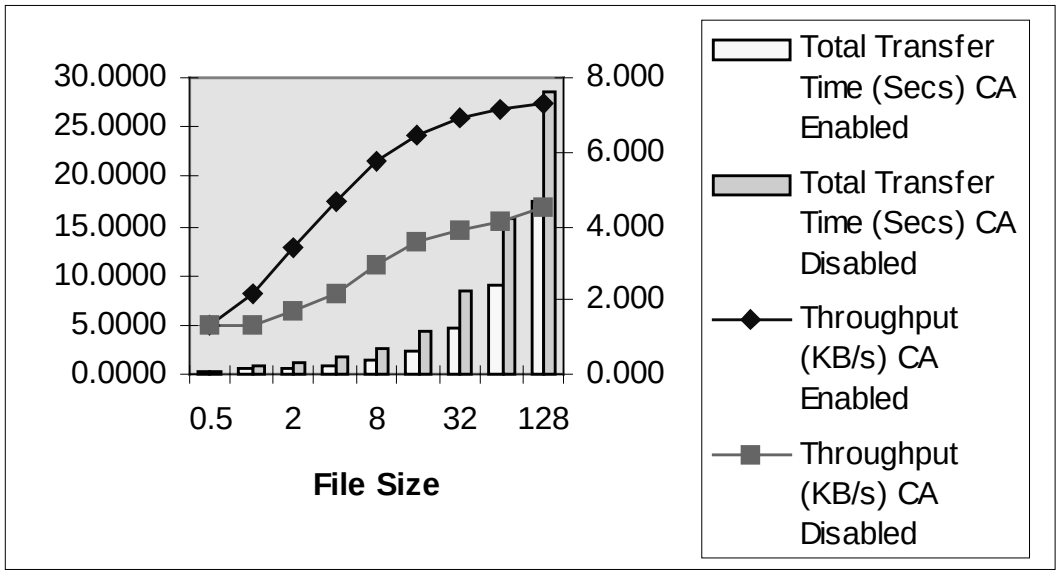

Figure 4-11 Transfer Times for Short Files

4.4 Exercise 4: Evaluation of “Fairness”

The exercise was conducted as specified in 3.5, except that it was not possible within the available timescale to simulate the 500ms delay on the slow data link, or to calculate “99%” figures (mean, min, max. and standard deviation are available where appropriate. Two 256kB files were transferred through a common constriction between two pairs of systems (Snd1 and Rcv1, and Snd2 and Rcv2), but with different length end to end paths, simulated by a slower speed link to Rcv2. The file Transfer Times and Throughputs are as given below:

Transfer Number	From	To	File Size (KB)	Total Transfer Time (Secs)	Throughput (KB/s)
1	Snd1	Rcv1	256	77.12	3.3195
2	Snd2	Rcv2	256	44.73	5.7234

Table 4-8 File Transfer Times for "Fairness" Investigation

It should be noted that these results are not as was first expected. The expected result was that the Window Sizes would be equalised by the algorithm, and hence, transfer #1, with the shorter RTT, would see a higher throughput. Instead (see Table 4-8), Transfer #2 had a significantly better throughput. Figure 4-13 shows that there was no equalisation of window sizes and, instead, Figure 4-14 shows that end to end delays were in fact equalised by the impact of the algorithm. As to why this is true is still under investigation. However, as Table 4-9 demonstrates, the equalisation in end-to-end transit delays is very good.

From	To	Transfer Phase	Mean Delay (Secs)	Min. Delay (Secs)	Max. Delay (Secs)	Std Dev.
Snd1	Rcv1	During Cross-Traffic	0.31754	0.22046	0.44162	0.04854
Snd1	Rcv1	Cross Traffic Ceased	0.24751	0.22913	0.30482	0.02626
Snd2	Rcv2	During Cross-Traffic	0.33631	0.20837	0.45700	0.04644

Table 4-9 End to End Delays for "Fairness" Investigation

Inspection of the buffer loadings for IS1 (the congested router) in Figure 4-15 and Figure 4-16 does not reveal any surprises.

Transfer Phase	Mean Loading	Min. Loading	Max. Loading	Std Dev.
Cross Traffic	46,263.30	0	17,344	3,725.58
Cross Traffic Ceased	2,696.51	0	9,296	2,867.85

Table 4-10 Buffer Loadings for "Fairness" Investigation

No retransmissions were observed during the exercise.

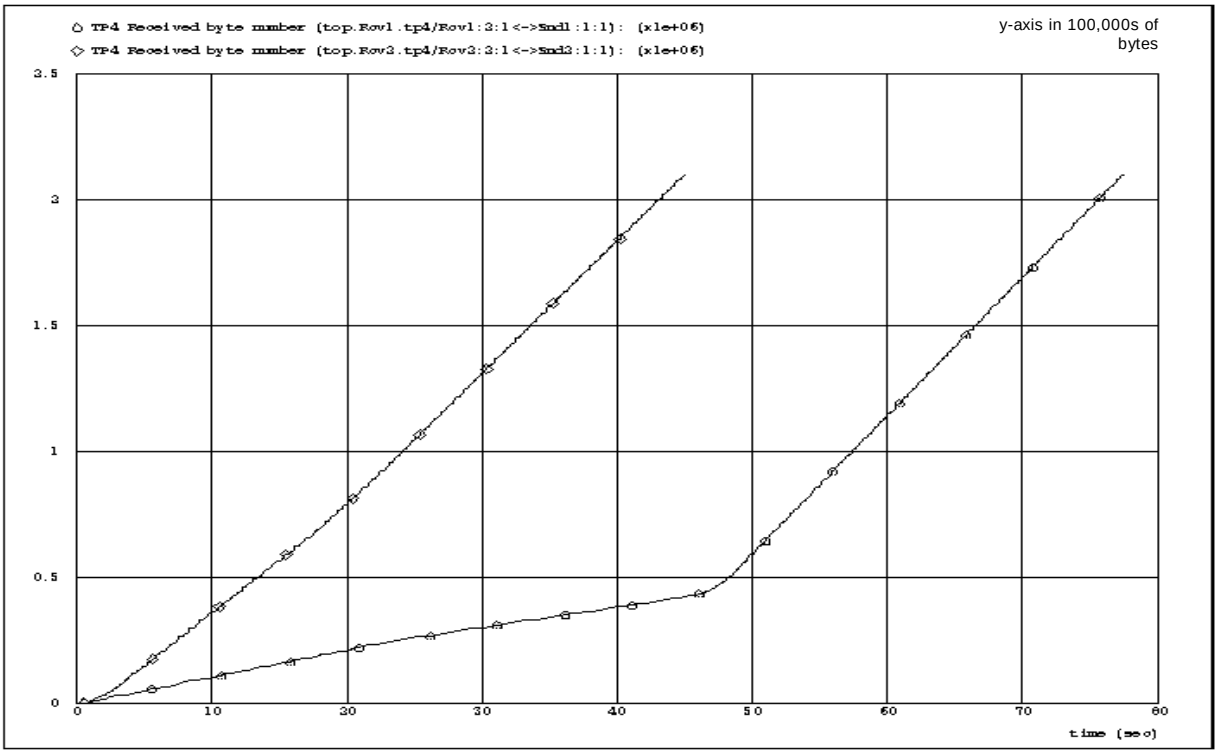

Figure 4-12 Received Byte Counts for "Fairness" investigation

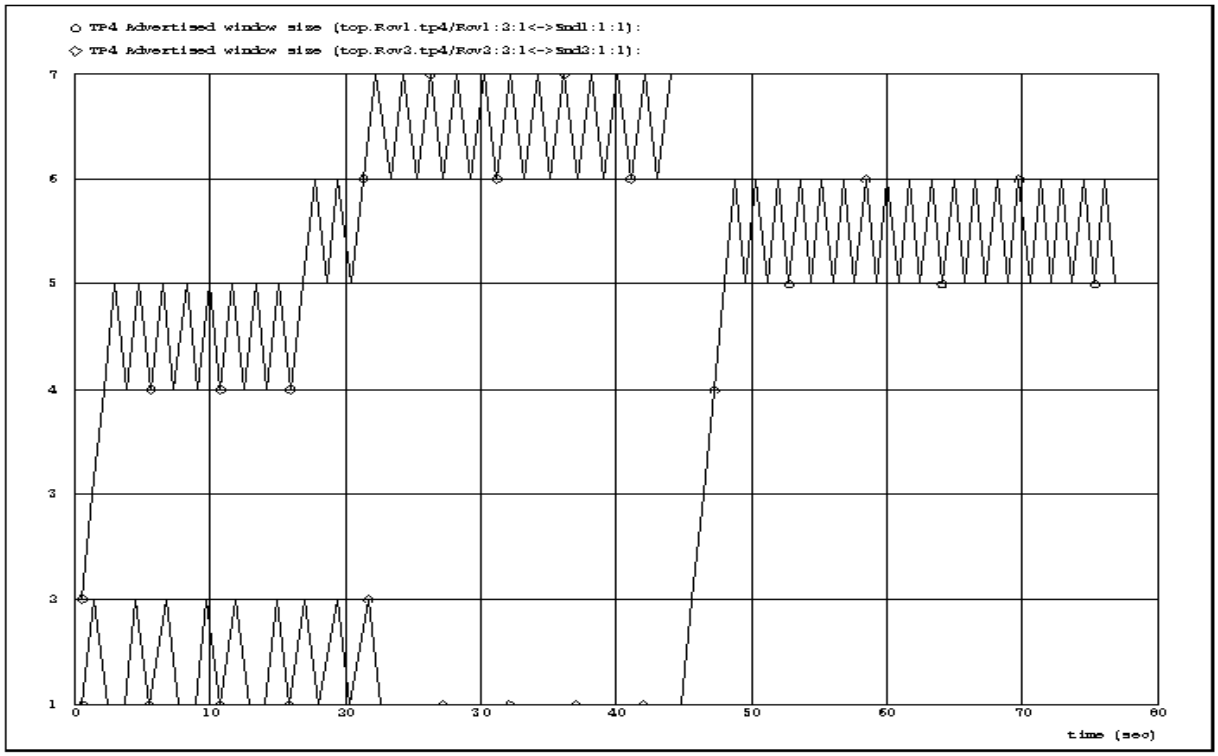

Figure 4-13 Windows Sizes for "Fairness" Investigation

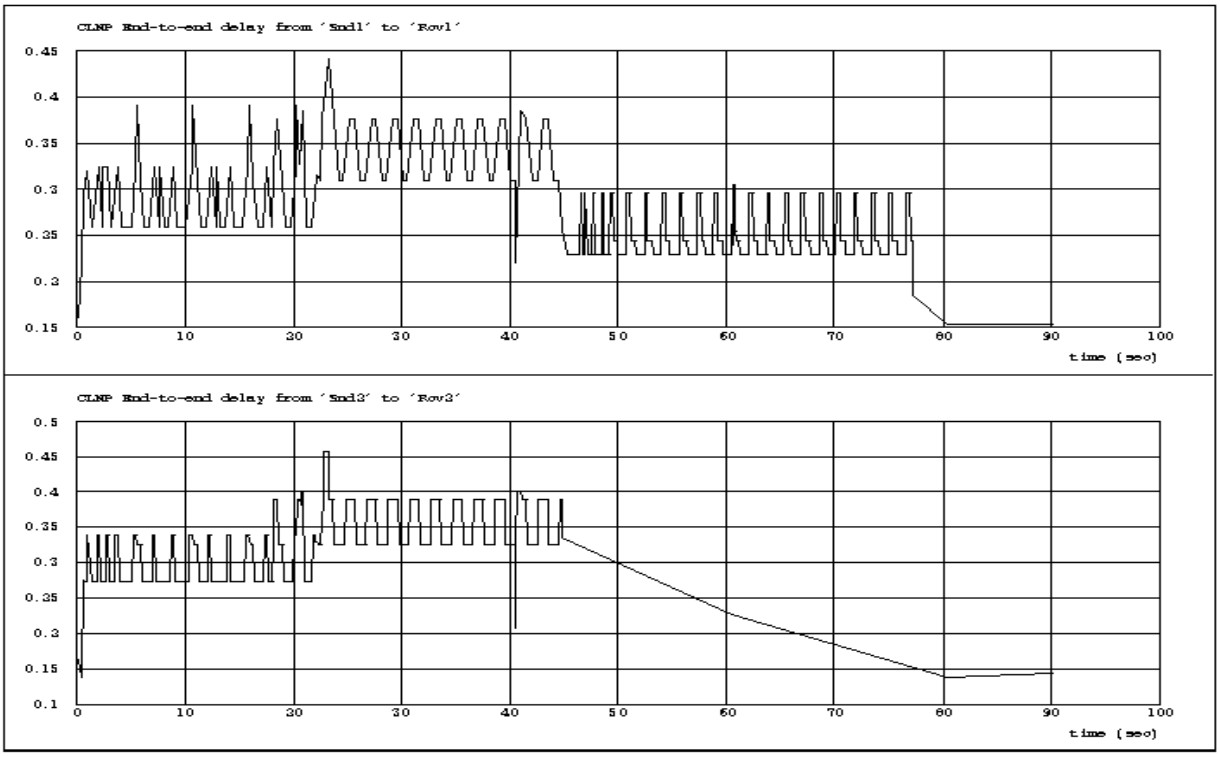

Figure 4-14 End to End Delays for "Fairness" Investigation

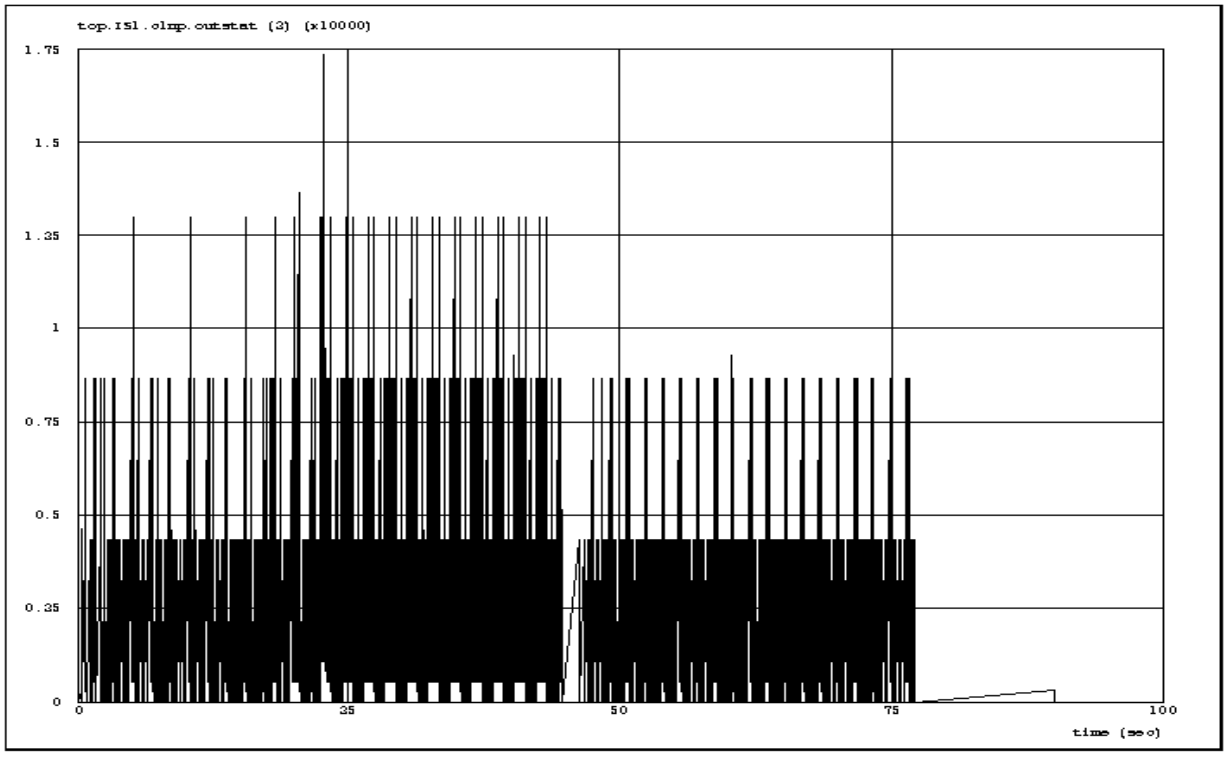

Figure 4-15 Buffer Loadings for "Fairness" Investigation

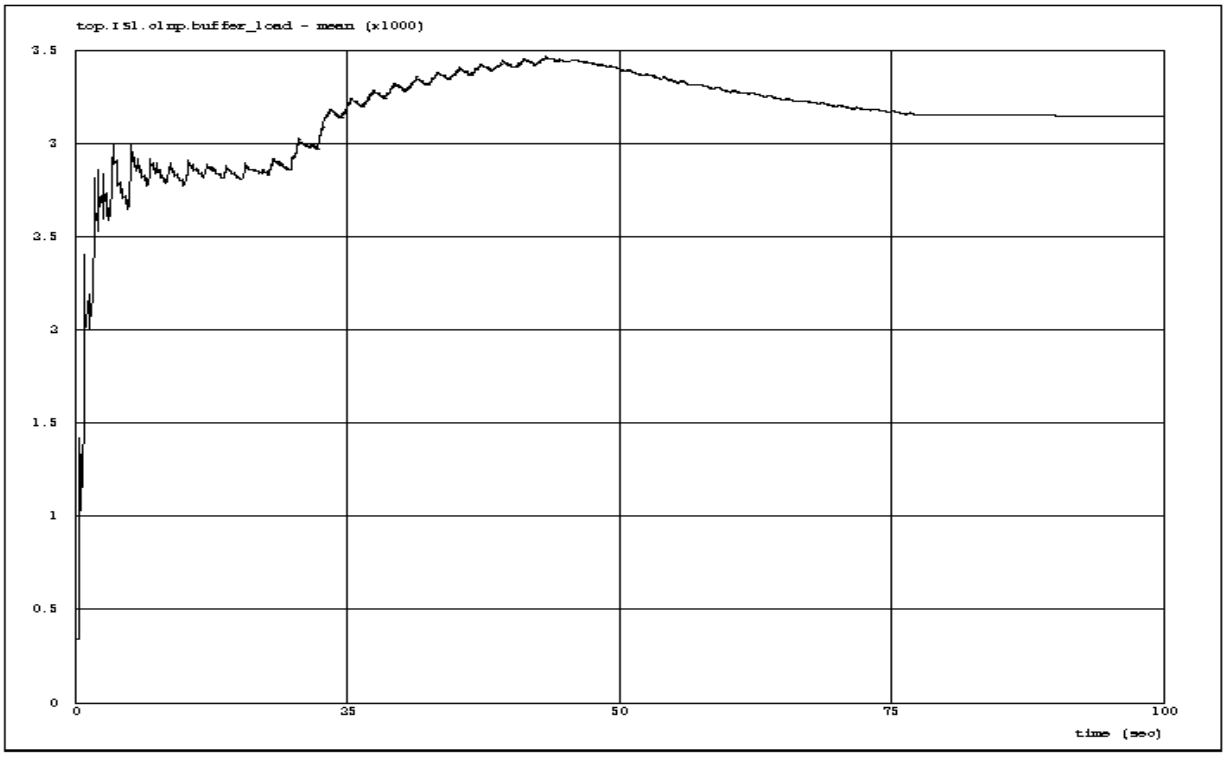

Figure 4-16 Mean Buffer Loadings for "Fairness" Investigation

4.5 Exercise 5: Bi-directional Data Transfer

This exercise was conducted as specified in 3.6, except that it was not possible within the available timescale to simulate the 500ms delay on the slow data link, or to calculate "99%" figures (mean, min, max. and standard deviation are available where appropriate. Two 128kB files are transferred from systems Snd1 and Snd2 to Rcv1, and a single 256KB file is transmitted in the reverse direction from Rcv1 to Snd1. The measured file transfer times and throughputs are as given below:

Transfer Number	From	To	File Size (KB)	Total Transfer Time (Secs)	Throughput (KB/s)
1	Snd1	Rcv1	128	34.8047	3.678
2	Snd2	Rcv1	128	54.0901	2.366
3	Rcv1	Snd1	256	58.4229	4.382

Table 4-11 File Transfer Times for Bi-Directional Data Transfer

The data transfers can be seen graphically in Figure 4-17. As may be seen, the data transfer rate for Transfer #2 is significantly poorer than for Transfer #1, even though they have the same data transfer scenario and the results of the first two exercises would have suggested that they should have the same transfer rates. Figure 4-17 confirms this, and also shows that Transfer #3 has the same transfer rate as Transfer #1, while they are both active. Its slightly higher overall transfer rate is due to it getting sole use of the data link during the latter phase of the experiment.

Figure 4-18 sheds some light on the reason for this difference in transfer rates. Transfer #2 is converging on a smaller credit window than transfer #1, and is only able to increase this, once Transfer #1 completes. Even then, Transfer #3 maintains a larger window size.

The most likely explanation for this is that we are seeing a "slotting" effect occuring. As DT TPDUs on Transfer #1 pass over the same data link from Snd1 to IS1(see Figure 3-5), as do AK TPDUs on Transfer #3, their arrival times at IS1 are well separated and this limits the probablility that there is any competition between them for access to the queue to IS2 (i.e. the restriction point). On the other hand, the arrival times of DT TPDUs on Transfer #2, at IS1, follow a much more random distribution, and there is likely to be competition for access to the queue to IS2 between:

- DT TPDUs on Transfer #2, and
- DT TPDUs on Transfer #1 and AK TPDUs on Transfer #3.

Hence, there is a greater probability that DT TPDUs on Transfer #2 will experience congestion than those on Transfer #1, and this is confirmed by the above results.

Once Transfer #1 has completed, Transfer #2 is able to increase its credit window. However, its DT TPDUs still have to compete with AK TPDUs from Transfer #3 on the queue to Rcv1. Whilst Transfer #3's DT TPDUs have similar competition on the queue from IS2 to IS1 with Transfer #2 AK TPDUs, the slotting effect due to the data link from Rcv1 to IS2 may also be seen here, reducing the probability of conflict. There is therefore a higher probability that Transfer #2 DT TPDUs will experience congestion compared with Transfer #3 DT TPDUs, and this is confirmed by the smaller window size achieved by Transfer #2, even during the later phase of the exercise.

This "unfairness" in the way network resources are apportioned to different connections could be avoided if AK and DT TPDUs were on different output queues in each Router, which may be arranged if they have different priorities. Indeed, giving AK TPDUs higher

priorities than DT TPDUs would aid rapid response to changing congestion levels. This should therefore be investigated.

The competition between AK TPDUs and DT TPDUs also has a downside influence on data link utilisation. From IS2 to IS1, the utilisation is now only 4.382 kb/s, an efficiency of only 54.775%. In the other direction the utilisation is a better 6.044kB/s, achieving an efficiency of 75.55%. In both cases, this is due to interference by AK TPDUs with DT TPDUs, and the greater number of AK TPDUs compared with DT TPDUs in the direction of IS2 to IS1 clearly has an adverse influence.

The end-to-end delay experienced by all three data transfers is shown in Figure 4-19. The overall figures are in line with what would be expected, although the much greater variability of the transfer delays for Transfers #1 and #2 are worth noting, indicating significant interaction between the data flows. Table 4-12 gives the precise figures during each phase.

From	To	Transfer Phase	Mean Delay (Secs)	Min. Delay (Secs)	Max. Delay (Secs)	Std Dev.
Snd1	Rcv1	4.96 - 35.14 seconds	0.22897	0.15383	0.39337	0.06478
Snd2	Rcv1	4.69 - 35.07 seconds	0.30840	0.16804	0.42103	0.05062
Rcv1	Snd1	4.99 - 35.05 seconds	0.22893	0.15383	0.39118	0.06454
Snd2	Rcv1	39.97 - 50.03 seconds	0.26084	0.22913	0.32010	0.03325
Rcv1	Snd1	39.95 - 50.06 seconds	0.26772	0.22484	0.32230	0.035503

Table 4-12 End to End Delays with Bi-Directional Data Transfer

Figure 4-20 shows the buffer loads in the Router at each end of the constricted data link. There are no surprises here, and this is confirmed by Figure 4-21. Figure 4-21 provides the precise figures.

Transfer Phase	Mean Loading	Min. Loading	Max. Loading	Std Dev.
IS1: 4.96 - 36.43 seconds	2,833.58	0	13,008	3,618.07
IS2: 4.99 - 35.18 seconds	2,381.42	0	13,008	3,156.03
IS1: 39.98 - 50.09	2,261.99	0	8,672	2,897.68
IS2: 39.96 - 50.07 seconds	2,473.48	0	8,672	2,473.48

Table 4-13 Router Buffer Loading with Bi-directional Data Transfer (in bits)

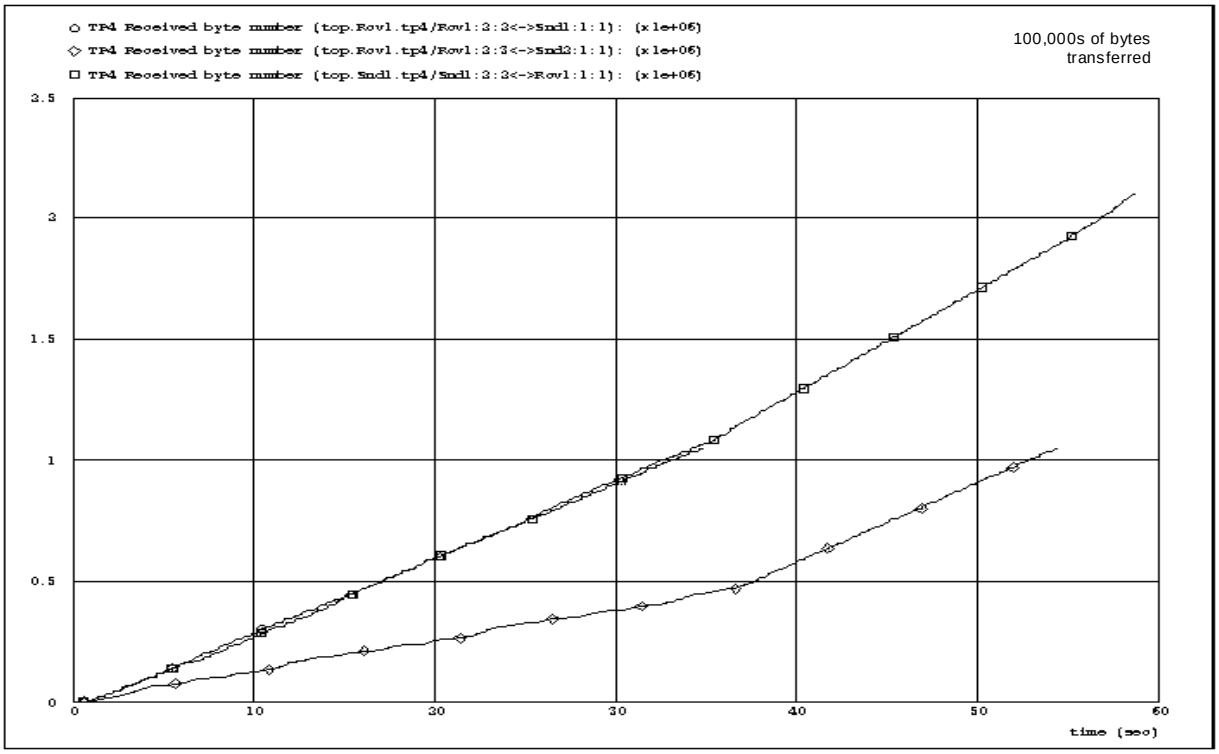

Figure 4-17 Received Byte Counts with Bi-Directional Data Transfer

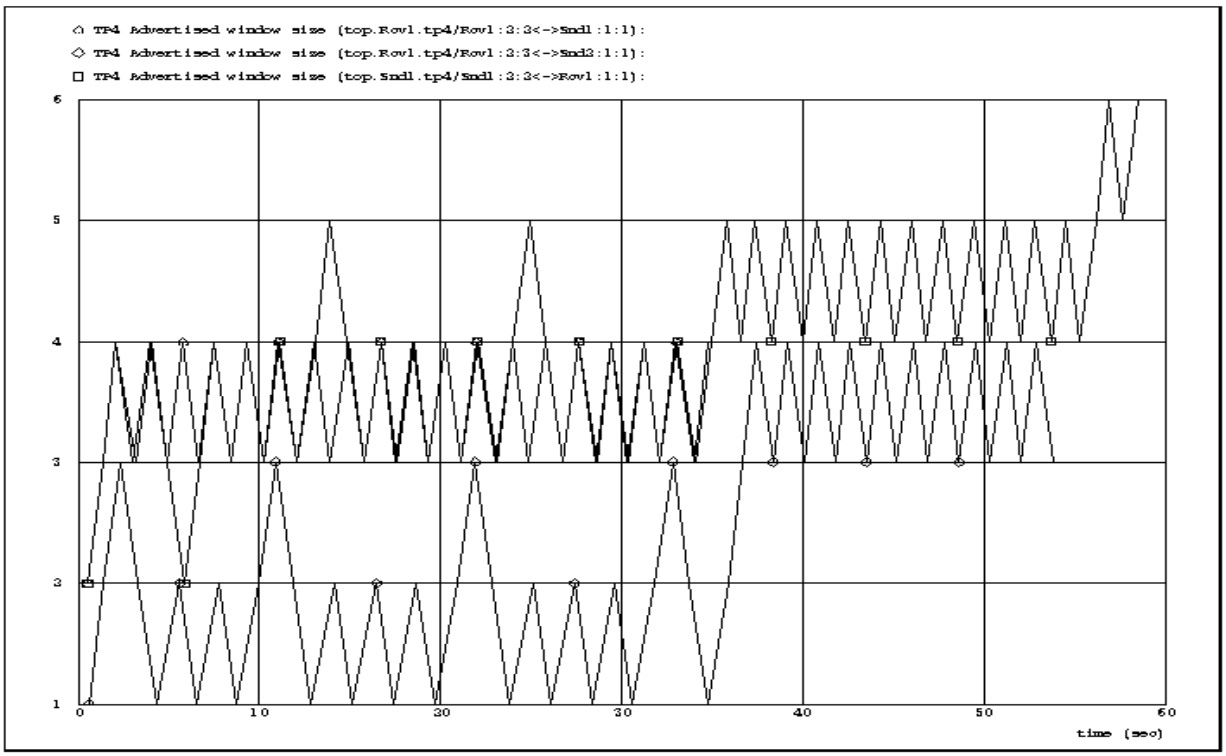

Figure 4-18 Window Sizes with Bi-Directional Data Transfer

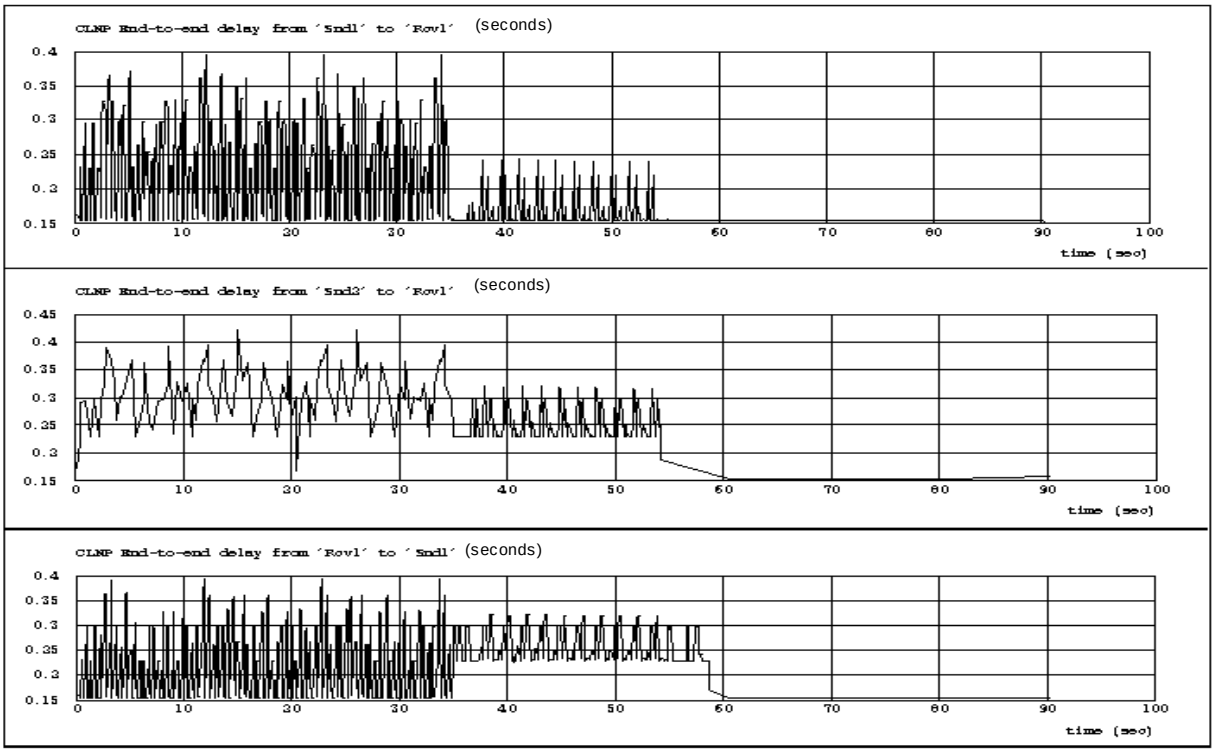

Figure 4-19 End to End Delays with Bi_Directional Data Transfer

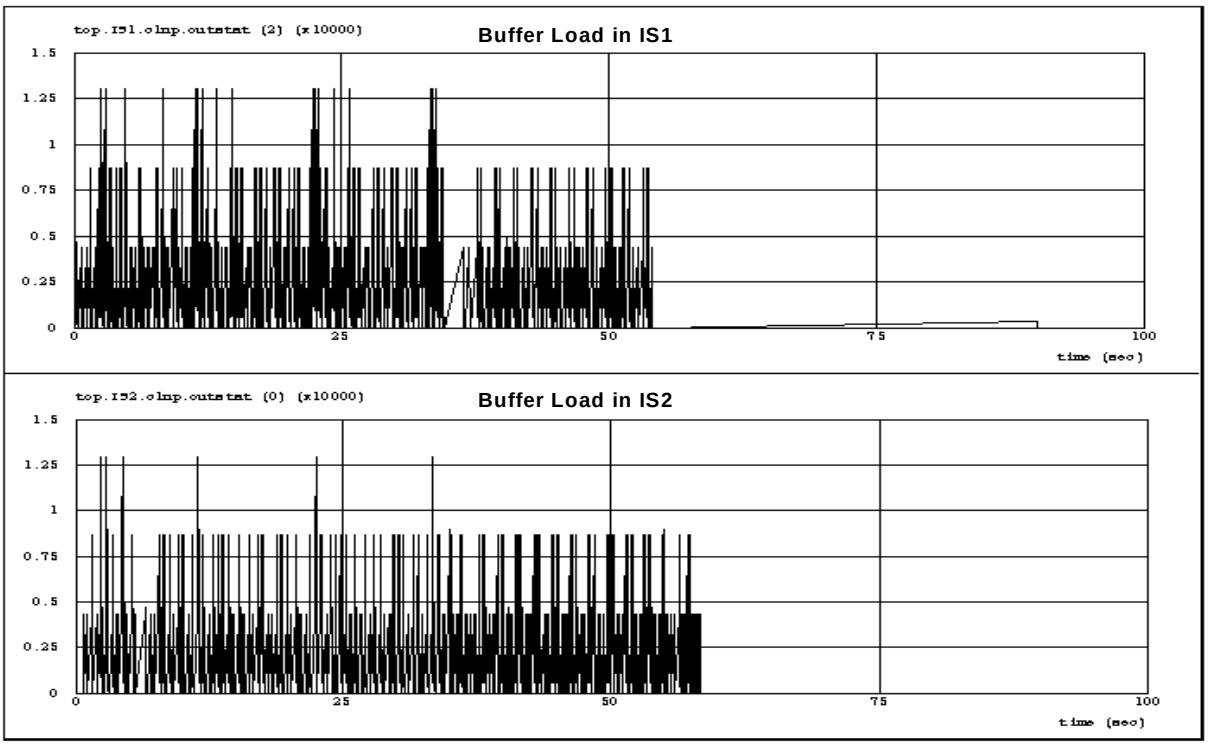

Figure 4-20 Router Buffer Load with Bi-Directional Data Transfer

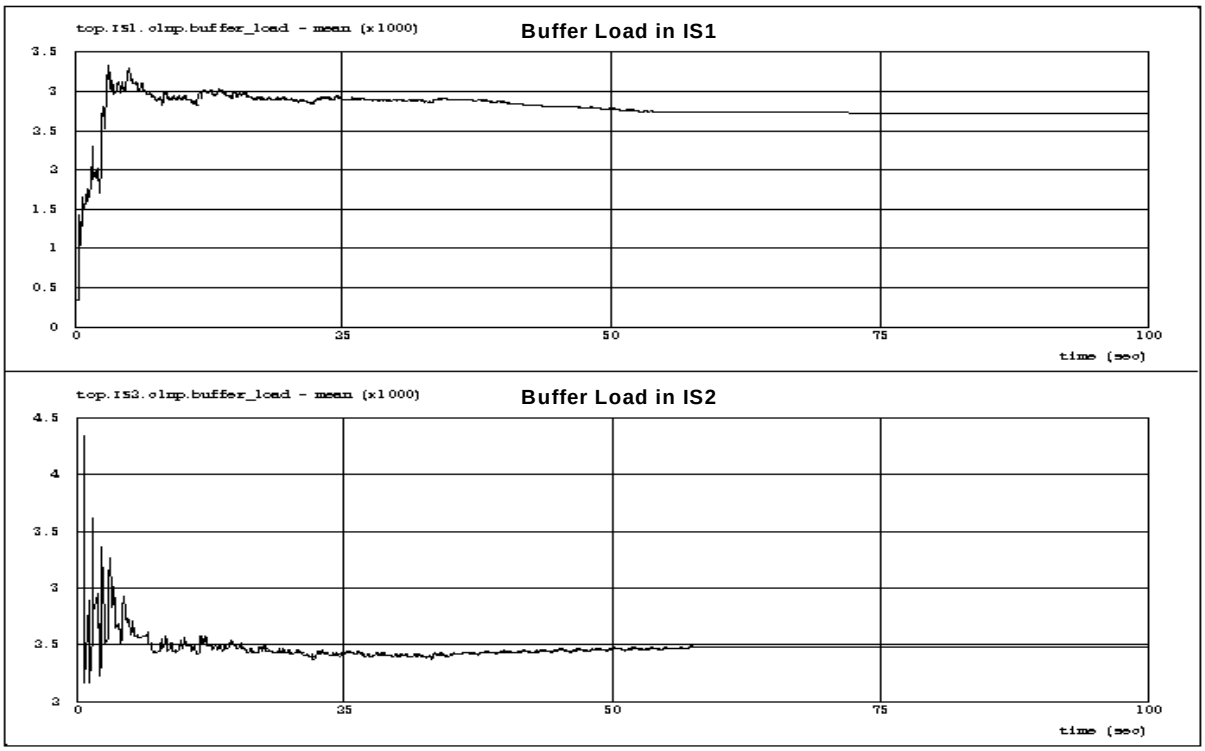

Figure 4-21 Mean Router Buffer Load with Bi-Directional Data Transfer

4.6 Exercise 6: Non-compliant Cross Traffic

The exercise was conducted as specified in 3.7, except that it was not possible within the available timescale to simulate the 500ms delay on the slow data link, or to calculate "99%" figures (mean, min, max. and standard deviation are available where appropriate. A 256kB file transferred using the Congestion Avoidance algorithm, and five 64kB cross-traffic transfers transferred without any congestion avoidance. The simulation was limited to 100 seconds and no file transfer completed in this period.

As may be seen, by inspection of Figure 4-22 there is considerable and almost random variation in data transfer rates. Transfer #1 has a much poorer throughput than in exercise 2. However, at least the Congestion Avoidance algorithm ensures that it has a smooth transfer rate. On the other hand, the other connections have a very unstable transfer rate indicative of the lack of Congestion Avoidance. Overall the throughputs achieved are much lower than was achieved in exercise 2.

Figure 4-24 confirms the poor performance, indicating a much increased end to end transit delay, and with a high variation. Table 4-14 has the detailed figures.

From	To	Transfer Phase	Mean Delay (Secs)	Min. Delay (Secs)	Max. Delay (Secs)	Std Dev.
Snd1	Rcv1	File Transfer	1.15881	0.22913	1.99948	0.53205
Snd2	Rcv2	File Transfer	1.38528	0.13488	0.13488	0.13488

Table 4-14 End to End Delays with non-compliant Traffic

Figure 4-25 shows that the buffer loading is much more variable than before, illustrating an over-stressed router. Table 4-15 provides the supporting figures.

Transfer Phase	Mean Loading	Min. Loading	Max. Loading	Std Dev.
Data Transfer	46,263.30	0	124,144	41,917.00

Table 4-15 Buffer Loading in Router with non-compliant Cross Traffic

Finally, it should be noted that in this scenario, there were a total of 2520 TPDUs transferred and 38 re-transmissions. This shows that the lack of re-transmissions in the previous exercises was due to the operation of the congestion avoidance algorithm.

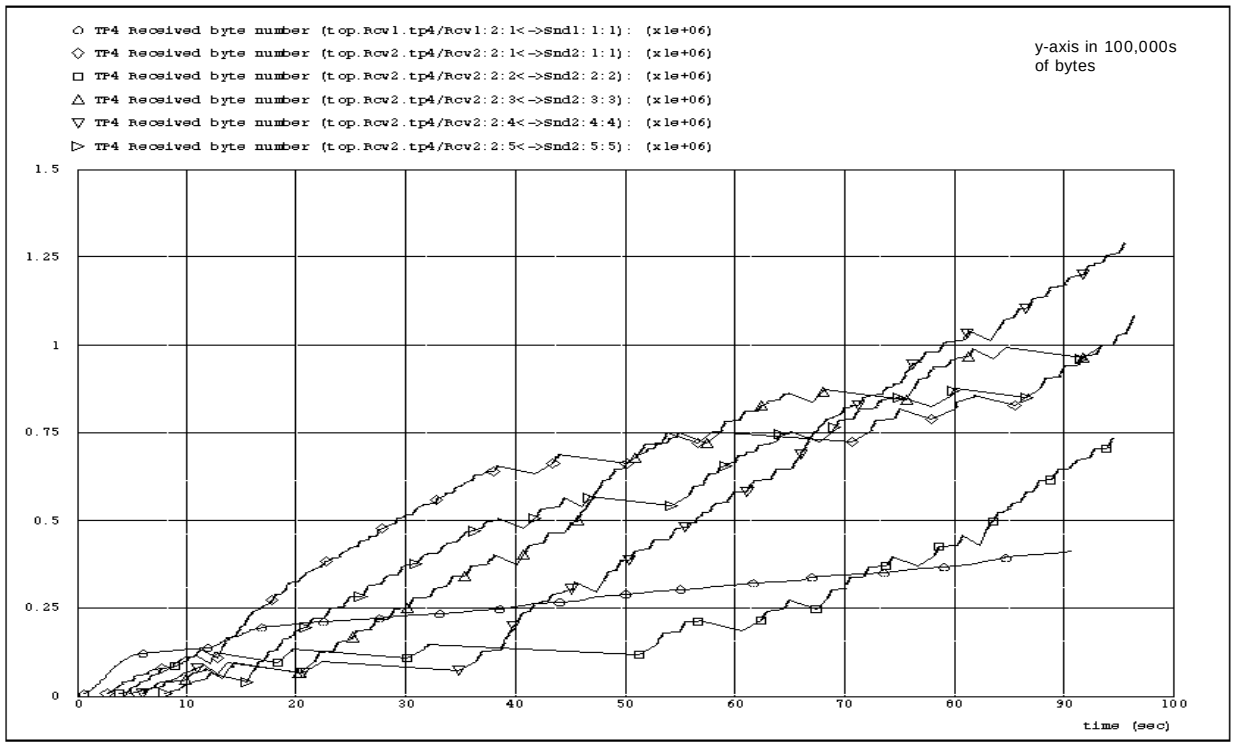

Figure 4-22 Received Byte Counts with Non-compliant Traffic

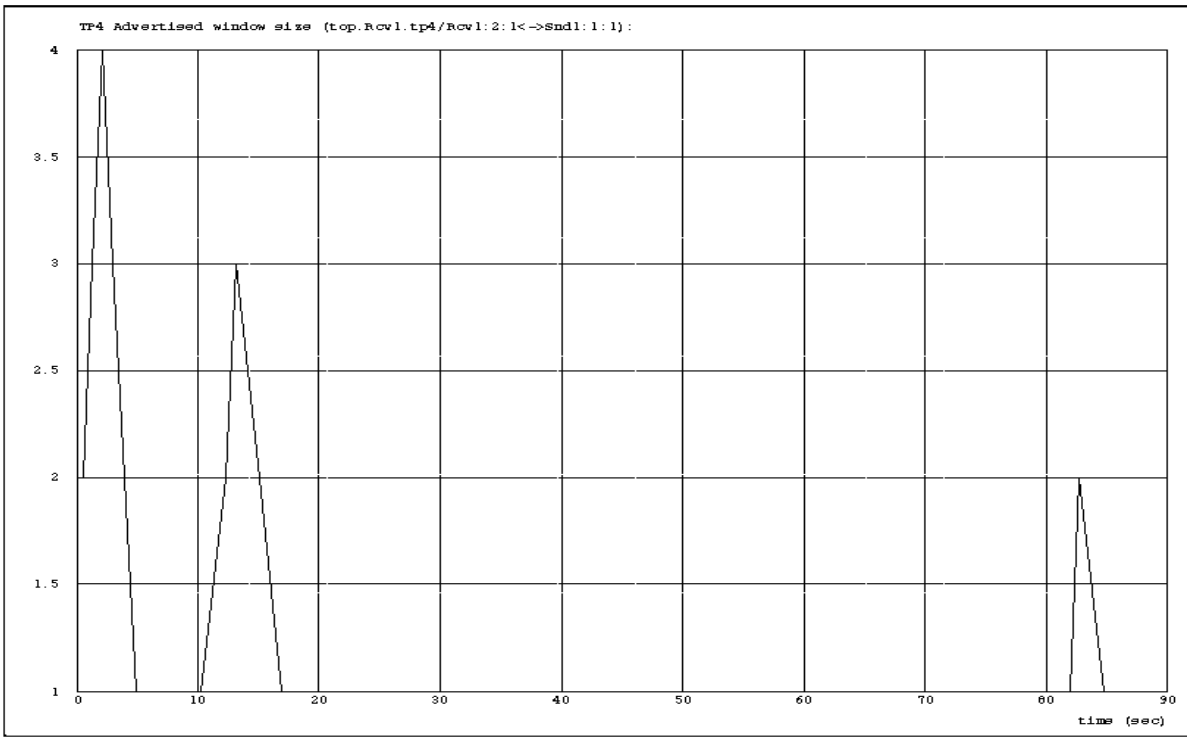

Figure 4-23 Window Sizes with Non-Compliant Cross Traffic

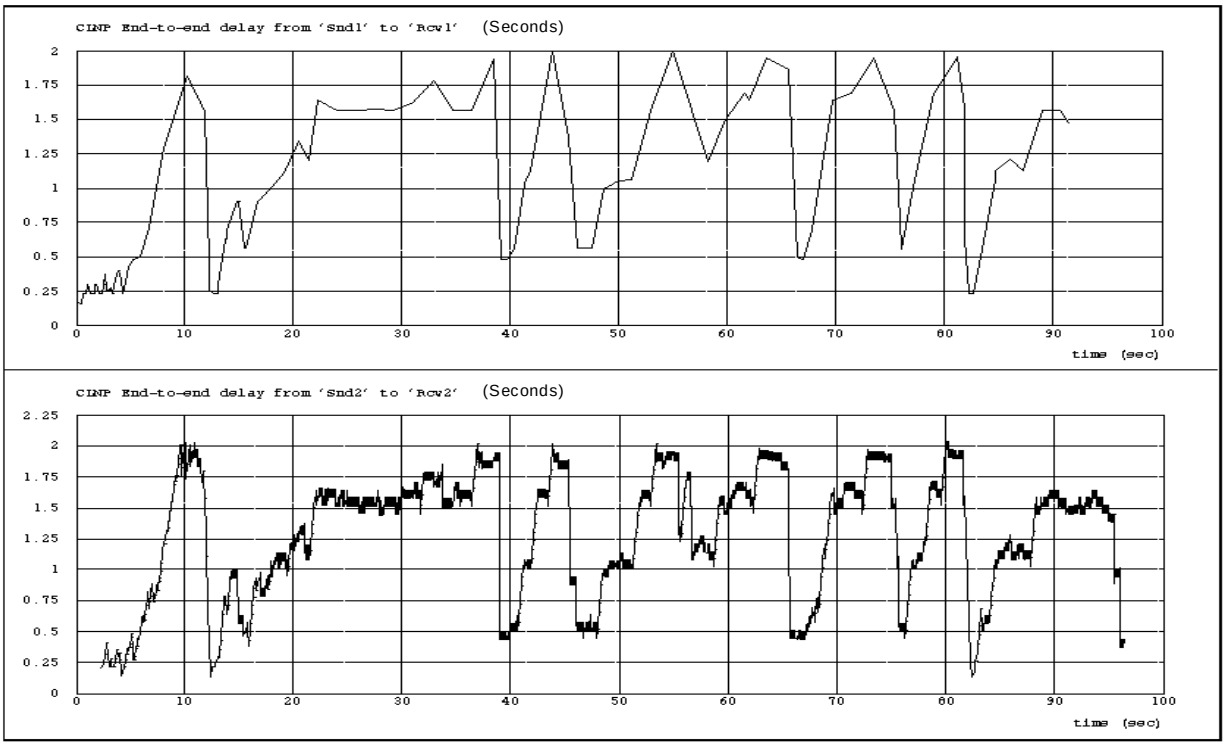

Figure 4-24 End to End Delay with Non-Compliant Cross Traffic

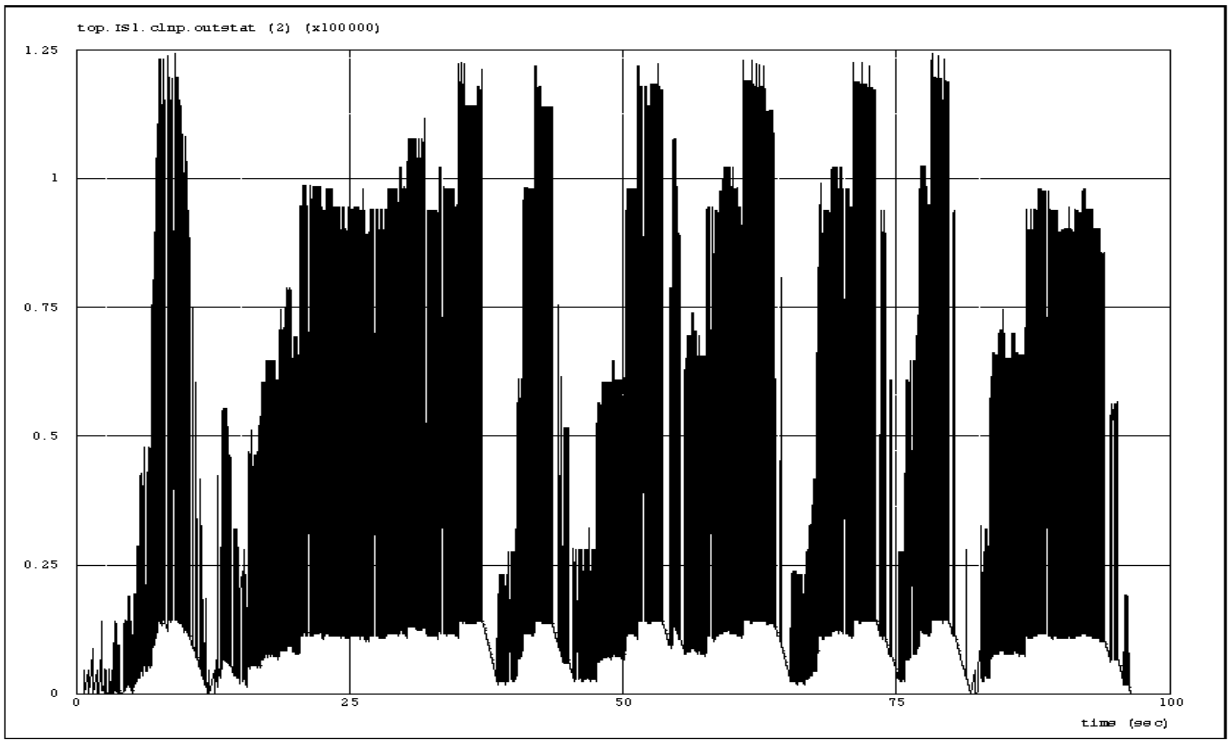

Figure 4-25 Buffer Loading with Non-Compliant Cross Traffic

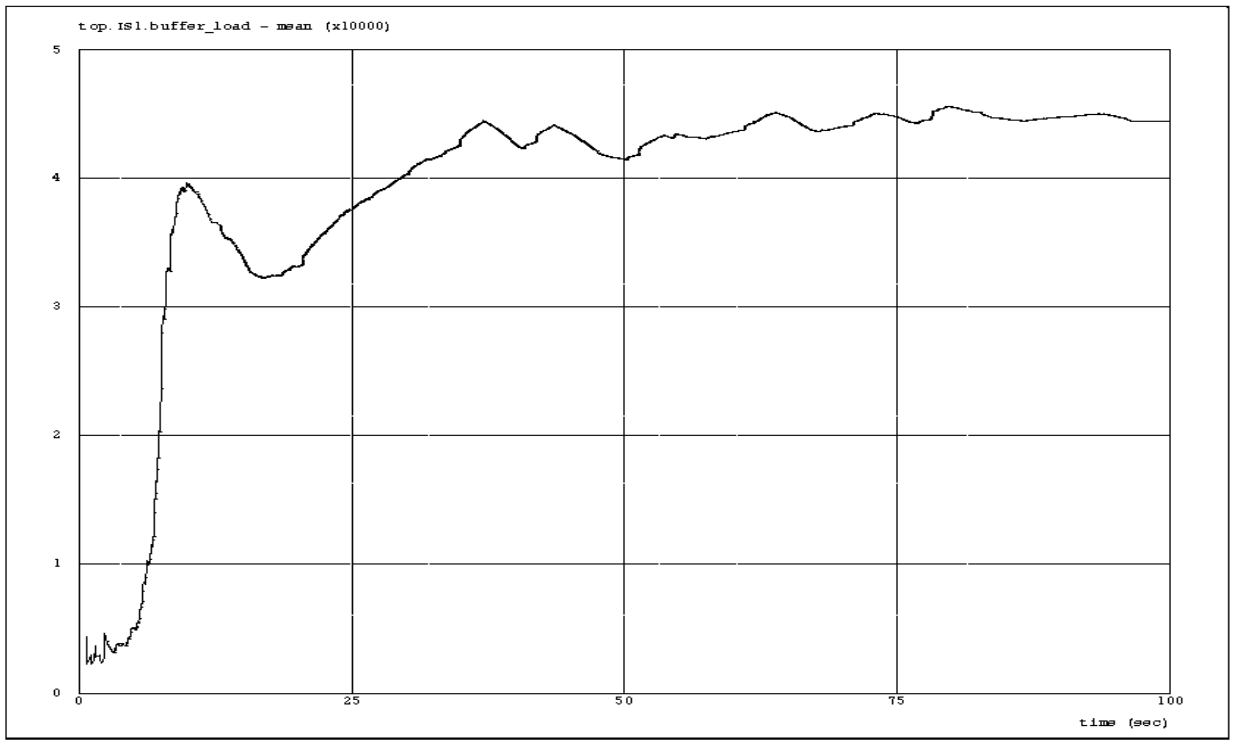

Figure 4-26 Mean Buffer Load with Non-Compliant Cross Traffic

5. Proposed Changes to the Draft ATN Internet SARPs

Referring to **[2]**, the proposed changes can be assumed to have been verified by this work, and the following changes are therefore proposed for the draft ATN SARPs.

5.1 Setting the Congestion Experienced Flag

In paragraph 6.2.4.2 of draft 4.0 of the ATN Internet SARPs, it is proposed that the range of α is set to a single value of one.

5.2 Reporting Congestion Experienced to the Transport Layer

It is proposed to replace paragraph 5.3.5.2.3.2 of draft 4.0 of the ATN Internet SARPs, with:

5.2.5.2.3.2 Reporting Congestion Experience

When Congestion Experienced is indicated in one or more of the NPDUs that conveyed a received NSDU, then a notification of Congestion Experienced for that NSDU shall be reported to the Transport Entity. For each NSDU, the transport entity shall be informed of both the total number of NPDUs into which the received NSDU was segmented, and the total number of those NPDUs which had experienced congestion.

5.3 Determining the Credit Window

It is proposed to insert the follow text at the end of paragraph 5.2.6.2.4 of the ATN Internet SARPs:

If as a result of this procedure, the advertised window size has changed from the previous value of *W*, then the next sampling period shall not start until after a delay equivalent to the estimated Round Trip Time (RTT) has expired.

The note in 5.2.6.2.2 needs also to be changed to reflect the above. The following replacement text is proposed:

Note: Unless the following procedures result in a new value for W, the end of the sampling period determines the beginning of the next sampling period. Re-computation of W and its implications is specified below.

5.4 TPDU Sampling

It is proposed to replace the text of 5.2.6.2.3 of draft 4.0 of the ATN Internet SARPs with the following text:

5.2.6.2.3 Counting of Received DT TPDUs in a Sampling Period

The receiving transport entity shall maintain a count *N*, equal to the total number of NPDUs received that convey DT TPDUs, and a count *NC*, equal to the total number of such NPDUs that report Congestion Experienced.

5.5 Recommended Window Decrease Factor

It is proposed to replace the number “1” in line three of 5.2.6.2.4(1) of draft 4.0 of the ATN Internet SARPs with the symbol δ.

It is also proposed to replace the table in 5.2.6.3 with:

Name	Description	Recommended Value/Range
β	Window decrease factor	0.75..0.95
δ	Window increase factor	1
W_0	Initial Window	1
λ	Congestion Ratio	50%

6. Conclusion and Recommendations

This report is the result of validation activities aimed at validating the appropriateness of the specified algorithm. As a result, there is now considerable confidence in the Congestion Avoidance Algorithm for ground-ground use, with only the implementation in real systems outstanding. Validation work in support of air/ground use and the interaction with network priority is still outstanding. However, no significant problems are foreseen in this area. Work is still required to investigate the benefits of sending AK TPDUs at a different priority to DT TPDUs, and to demonstrate that data transfer at a given priority level is unaffected by Congestion Avoidance at a lower priority level.

It is recommended that WG2 accepts the proposed changes specified in section 5, and records this paper as a contribution to the validation of the draft ATN SARPs.