SELECTIVE BACK-PRESSURE IN SWITCHED ETHERNET LANS*

W. Noureddine and F. Tobagi
Computer Systems Laboratory
Stanford University, Stanford, CA 94305

Abstract

Although switched LANs are usually over-provisioned, their characteristics (short RTT, link speed mismatches) lead to increased burstiness, and thus to the occurrence of transient congestion. In order to fully utilize the potential of large switched LANs, a link layer back-pressure mechanism may be used to complement end-to-end flow control by handling the short term congestion. A simple such mechanism, as the one specified in IEEE 802.3x, is shown to improve network performance in some situations, but to lead to poor performance in others. We propose selective back-pressure schemes based on destination MAC address and traffic class information, which overcome the limitations of the simpler scheme. These are shown to provide superior performance for a wide range of situations. The results obtained suggest the need to incorporate these enhancements in the 802.3x standard.

1 Introduction

There have been numerous studies of network congestion control dealing with wide area network congestion, with a focus on TCP congestion control (such as [5, 6, 8, 9, 10, 11]). LAN congestion has received less attention, perhaps due to the fact that, traditionally, LANs have been over-provisioned, relatively small in size and number of users, and congestion was satisfactorily dealt with by TCP.

Today, the picture is different: with the advent of high performance switching, higher speed technologies and new standards for selective multicast and Virtual LANs [1, 2, 3, 4], it is possible to deploy LANs to a large scale (extended LANs). New applications have emerged with large bandwidth and stringent delay requirements, such as multimedia applications. In addition, the mix of link speeds (e.g. 10, 100 and 1,000 Mbps) introduces rate mismatch considerations that were not faced previously. It is well known that, in such a context, congestion is caused by the aggregation of traffic from many sources and the large speed mismatch between links.

While end-to-end congestion control can prevent congestion collapse, it does not always result in optimal network performance. In particular, previous work on TCP performance has shown that TCP's efficiency is limited in the context of switched LANs. The short round trip times in the LAN, which result in fast window opening, lead to increased burstiness and thus buffer loss. Furthermore, the coarse granularity of the timers used to detect packet loss limits the effectiveness of TCP's congestion control mechanism, as the relative impact of the timers is more significant in the context of LANs than in the WAN [5, 6, 7].

Hop-by-hop flow control is a well known congestion control mechanism. Whereas hop-by-hop back-pressure may not be a practical solution in the WAN, it remains a valid option to consider in the LAN context. Indeed, such a scheme has been standardized for (full-duplex Gigabit) switched Ethernet LANs, among others [2]. The design, interactions with end-to-end control, and benefits of such a scheme are the subject of this paper.

Recent work on back-pressure includes studies reported in [13], which focus on back-pressure in the backbone, showing its usefulness in the context of many flows, where TCP's flow control mechanism results in bad performance. Studies reported in [12, 14, 15] and [16], are limited to non-selective back-pressure in LANs, and show the improvement it provides in some situations. In this paper, we provide a more in-depth treatment of the subject and clearly identify situations where such a scheme leads to performance improvements and situations where it leads to *performance degradation.* In addition, we study more sophisticated schemes which use different types of information, such as MAC address and traffic class, and which perform better. We also discuss the requirements for a resilient control scheme. Finally, we give particular

*This work was supported by 3Com Corp.

attention to the support of multimedia.

We use simulation to investigate the issues associated with the use of back-pressure in switched Ethernet LANs, carrying data and video traffic. Our simulator implements *full duplex* Ethernet links of various speeds, and a non-blocking output buffered model for switches. The simulator also implements class-of-service priorities: separate Tail Drop queues for different classes of service exist at the output ports. Queues of different priorities are serviced on a highest priority first basis.

The results presented in this paper are for TCP. Comparable results were obtained for UDP traffic, including self-similar traffic, and similar conclusions are applicable. A BSD Reno version of TCP is used for data traffic (with a maximum window size of 64KB), while real traces are used for video traffic (1.5 Mbps average rate H.261 CQ-VBR). The TCP source model is as follows: the source sends files of fixed sizes[1], after a file is transmitted successfully, the source waits for a random period of time, uniformly distributed between 0 and twice the mean, T, before transmitting the following file[2].

The paper is organized as follows. In section 2 we describe the general structure of a back-pressure mechanism and the various schemes we consider in the study, and identify the possible variants of each. In section 3 we give the results and conclusions of the simulation study of the mechanism. We provide scenarios that illustrate the ideas presented above. We conclude by summarizing the main findings in section 4.

2 Back-Pressure Scheme

We describe here the general structure of a MAC layer congestion control mechanism, as it would be implemented in a LAN device. Although we frequently use the term "switch" to denote such devices, simulation results presented in the following section show that it is critical for congestion control mechanisms to be implemented in end-stations, especially the ones connected to high-speed links, as well as in switches.

A back-pressure mechanism has three components: detection, notification and action. We describe these below, identifying the choices associated with each.

2.1 Congestion Detection

The LAN device in question must implement a monitoring mechanism, which detects and signals the occurrence as well as the end of congestion. Such a mechanism could be based on the buffer occupancy at output ports of the device.

The congestion detection mechanism we use has the advantage of simplicity and effectiveness. Since we are trying to rapidly tackle congestion, we use an instantaneous measure of congestion as opposed to computing an average over some period of time (e.g., such as in RED[3]). We assume that output buffering is used in the switches and the congestion detection is performed at the switch output buffers. With every buffer are associated a high threshold and a low threshold. When buffer occupancy exceeds the high threshold, congestion is considered to be occurring. This threshold needs to be set low enough for the buffer to be able to handle the packets that are received before the control actions take effect. Congestion is considered to be relieved when buffer occupancy falls below the low threshold. This threshold needs to be high enough to prevent starvation before control actions are reversed. As will be shown in the following section, the threshold margin defined as the difference between the high threshold and low threshold, plays a significant role in the performance of the scheme. It determines the frequency of control messages, as well as the time span of the control actions. A small threshold margin would result in a large number of control messages being exchanged at times of congestion, while a large margin can be detrimental to time-sensitive traffic.

2.2 Notification

When a device experiences congestion or is no longer congested, it has to notify other devices of the fact, asking for control actions to be performed or cancelled. There are several possibilities to consider concerning the choice of devices to notify. For example, a switch experiencing congestion at some output port may only notify the devices that are currently sending to the congested port (as opposed to notifying all neighboring switches). This choice can clearly influence the performance of the control mechanism.

For the present study, we assume that the output buffered switches do not distinguish between input links. Nevertheless, for any of the scenarios we con-

[1] We use fixed file sizes to simplify the analysis, however this does not affect the validity of our results.

[2] We use 10 msec for T in our simulations, in order to obtain sufficient samples.

[3] If LAN switches were to use RED, they may need to drop many packets to control congestion, limiting the effectiveness of the algorithm [7].

sider, one would be able to infer from the results we present the performance that is achieved for the case where switches have the ability to distinguish between input ports. In our discussion, we provide such comments whenever applicable.

Similarly, several possibilities exist as to what information is included in the notification messages sent, as well as the process of selection of MAC flows that are to be controlled. Possible information that could be used to identify a MAC flow is the class of service of the congested buffer or MAC address information. We present in this paper a comparison of schemes which provide different types of notification information. The schemes we examine are discussed below.

A **"simple"** scheme is one where no specific information is provided to discriminate between the flows that are involved in the congestion and others that are not. In this case, all flows are "selected" and control actions will act upon all traffic sent towards the congested switch. The IEEE 802.3x Xon/Xoff is an example of such a scheme.

A **class-of-service-based** scheme communicates the class of the congested buffer to the neighboring switches. Then, control actions would be restricted to traffic which belongs to the class of service that is experiencing congestion, allowing other traffic to proceed.

A **destination address-based** scheme is such that destination MAC address information is made available by the congested device to the other switches[4]. In this case, the process of selecting the flows could be varied. Once the flows going through the congested buffer are identified, one possibility would be to randomly choose one or several of them to be controlled. This method tends to single out the flows (if any) that have the largest numbers of packets in the buffer and thus are the "main cause" of the congestion. Another method would be to control all the identified flows. We comment on the difference in performance and implementation implications of these two methods in section 3[5]. In our simulations, all flows that are found in a congested buffer are controlled. This scheme could include class of service information as well.

2.3 Control Actions

When a switch is notified of congestion occurring (or ending) at a downstream device it has to perform control actions that would alleviate the congestion (or proceed to reverse the control actions taken previously). Such actions include possibly blocking/unblocking traffic destined to the device, or controlling the transmit rate.

In our study, we consider schemes where the control actions and reverse actions are transmission stopping/resuming, respectively. For such actions, several control message formats are possible. For example, the control messages sent could explicitly indicate the time period (in absolute time or in transmission slots) over which the actions are to be performed. Alternatively, they could implicitly indicate a default time period. Finally, control actions may be in effect until messages that explicitly cancel these actions are sent (we use this last format in our simulations). The different message formats have implications on the performance of the control mechanism, as will be shown in section 3.

3 Scenarios and Results

In this section, divided into three parts, we present simulation results for different back-pressure schemes. First, we consider three common situations that show the benefit of back-pressure in its most simple form, such as the Xon/Xoff mechanism defined in IEEE 802.3x [2], in terms of throughput and fairness. Conclusions for other, structurally similar but more complex topologies, can be inferred from the results shown here.

Next, we study situations where the blocking resulting from control actions has a significant negative performance impact. A set of scenarios is presented to illustrate the need for back-pressure based on destination address information, and based on *Class of Service* information in networks that implement traffic class differentiation [3].

For all simulations, unless otherwise noted, we use a high threshold of 80% and a low threshold of 70% for congestion detection, values that appeared to work well in practice for our buffer sizes. We use 1 MB for buffers at 1 Gbps ports, 500 KB for buffers at 100 Mbps ports and 70 KB for buffers at 10 Mbps ports. These are practical values, comparable to the ones used in commercial switches. Buffers at stations are assumed to be very large.

[4] Note that this information is available in the switch. All MAC addresses reachable through a given port can be obtained from the filtering database and, alternatively, the contents of the congested buffer can be examined and the destination addresses extracted from the packets that are in the buffer.

[5] Note that the difference in operation between the two methods depends on the location of the congested buffer. The difference would be minimal close to the edges of the LAN, where switches tend to have one station per port.

3.1 Simple Back-Pressure

Congestion occurs when the demand for network resources exceeds resource availability at some point in the network [10]. In LANs, such situations arise mainly due to the burstiness in traffic. The negative effects of congestion are well known and the scenarios presented here are mainly to illustrate the improvement that can be achieved with MAC layer back-pressure.

Whereas TCP is able to reduce congestion by reacting to packet loss, it suffers from fairness and efficiency problems that are particularly observable in the switched LAN context [5, 6, 7].

TCP's bias against bursty sources results in unfairness in resource usage [5]. Although some of these problems can be addressed at the level of TCP, we do not attempt to do that in this paper.

In addition, the pattern of packet loss seen in the Tail Drop queues of the LAN switches consistently forces TCP to wait for retransmission timers, and the reliance on coarse timers results in relatively large idle time and significant performance loss [5, 6, 7]. Avoiding such losses helps achieving high network utilization: once TCP has increased its window size to the maximum value, its transmission rate corresponds to the maximum rate possible for the connection, since it injects packets in the network at the same rate as packets are being removed on the destination end (this is referred to as the self-clocking property of TCP [9]). Thus, by providing enough buffering resources in the LAN to hold the burst of packets generated by the window mechanism during the initial increase phase "slow start", it is possible to achieve optimal performance. Back-pressure is a form of sharing of buffering resources across multiple devices, allowing the increase of the resources that a congested switch can use. Such sharing is not only more efficient than increasing the sizes of individual buffers, but it has also other advantages, such as addressing fairness issues.

3.1.1 Link Speed Mismatch

Burstiness causes congestion at a point of link speed mismatch. Consider the configuration depicted in Figure 1. Server S is using a number of parallel TCP connections[6] to send files to a client station D through a switch. The graph in Figure 2 shows the achievable throughput on the path between source and destination as a function of the number of TCP connections that are sharing the path, for selected file sizes (solid lines). It is clear that about half the maximum achievable throughput is lost as connections are added on the same path, which is undesirable. The reason for throughput loss is the merging of bursts from multiple connections, resulting in larger bursts, which cannot be accommodated in the 70 KB buffer. Resulting packet loss at the 10 Mbps port is translated into throughput loss by the timer-based reaction of TCP. This loss of network performance would be more severe if larger window sizes were allowed, or if available buffering space was more limited.

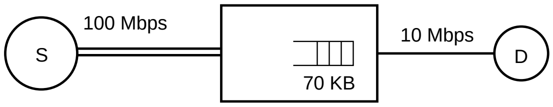

Figure 1: Link Speed Mismatch Scenario.

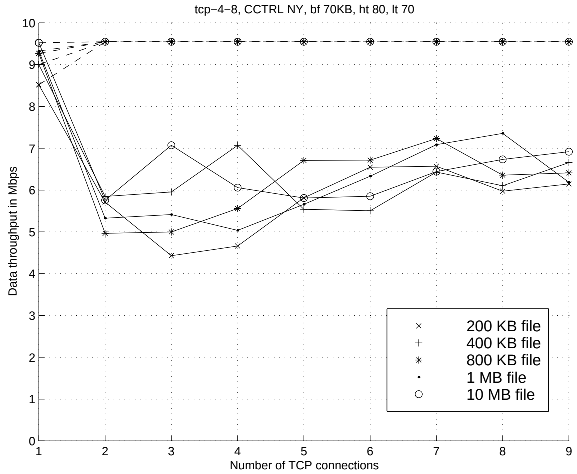

Figure 2: Link Speed Mismatch scenario: achieved throughput versus number of TCP connections.

With back-pressure, the control actions allow the elimination of buffer loss, thus achieving maximum throughput on the 10 Mbps link (dashed lines).

3.1.2 Traffic Merging

Throughput loss due to congestion may also be observed when all link speeds are equal. Consider the scenario shown in Figure 3. N different stations use one TCP connection each, to send files of equal size to the same destination station D. As a result of the burstiness of the traffic sent by TCP and the merging of bursts from several connections, the 70 KB buffer at the 10 Mbps output port to the destination station

[6] Note that having many TCP connections in parallel is a very common situation; for example, some popular commercial web browsers use such parallel TCP connections to download multiple components of a Web page.

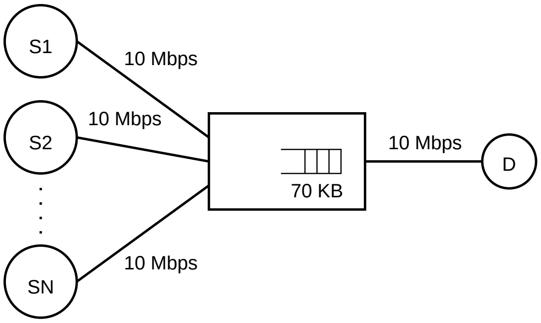

Figure 3: Traffic Merging Scenario.

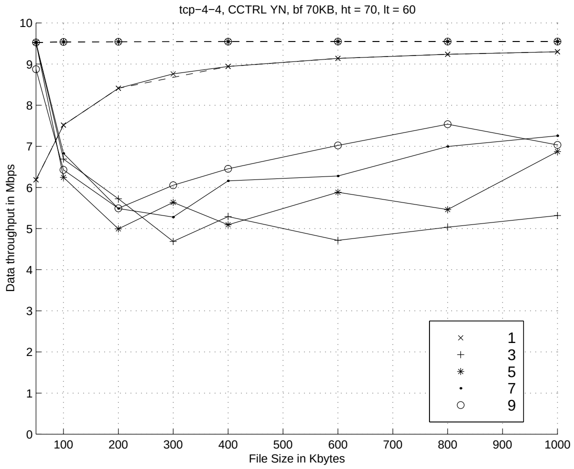

Figure 4: Traffic Merging Scenario: aggregate throughput versus file sizes, for different numbers of sources.

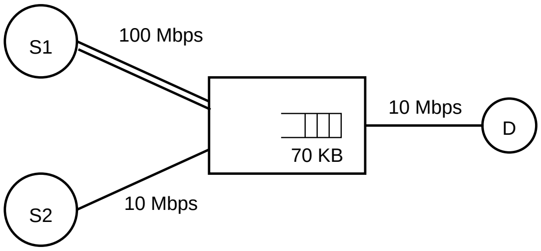

Figure 5: Input Link Speed Mismatch Scenario.

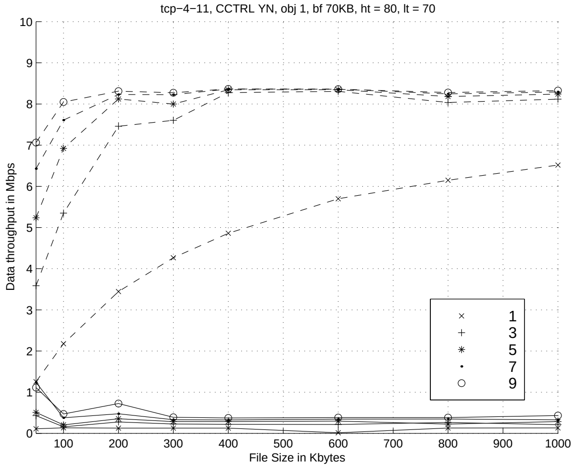

Figure 6: Input Link Speed Mismatch Scenario: aggregate throughput from S1 to D versus the file sizes, for selected numbers of connections (solid lines: without back-pressure, dashed lines: with back-pressure).

may overflow, resulting in packet loss. We show the results in Figure 4.

Again, implementing back-pressure provides a significant improvement in performance (dashed lines). We assume here that flow control is available at the sources (we will return to the issue of source control later in this section). As a result, the aggregate throughput obtained when control is enabled is the maximum achievable throughput. Other results have shown that for a smaller buffer size (e.g., 50 KB), the high threshold has to be set at a lower point (e.g., 70%) for packet loss to be eliminated, since in a topology where packets merge from many input links, as many packets may be sent in parallel before the control actions take effect.

3.1.3 Fairness Issues

Back-pressure can reduce unfairness that results from the bias of Tail Drop queues against bursty TCP sources [5]. Whereas similar results to the ones presented in 3.1.1 and 3.1.2 above are shown in [12, 14, 15], this section describes a new type of situations where back-pressure proves to be useful. Consider the scenario shown in Figure 5. Server S1 is sending a number of files to D, using parallel TCP connections. S2 is sending one long file (equivalent to an infinite supply of data) using a TCP connection to the same destination. TCP source S1, being on a 100 Mbps link, is "burstier" than S2 and suffers more loss at the 70 KB switch buffer. The results are shown in Figure 6. The aggregate throughput from S1 to D is very low, in the order of a few 100 Kbps for all the file size values and number of connections shown. On the other hand, the connection from S2 is using the remainder of the bandwidth on the 10 Mbps link. Such a division of the bandwidth is clearly unsatisfactory.

The dashed lines in Figure 6 shows the throughput obtained for the aggregate flow from S1 to D when back-pressure is used (the connection from S2 to D gets the remaining bandwidth). In this case, the division of bandwidth is more balanced. For large files and many connections the throughput ratio is about 8 to 1, and is directly related to the threshold margin and the format of control messages. When stations are told to resume transmission, S1 can send packets 10 times faster than S2 and is likely to fill the buffer space corresponding to the threshold margin, while S2 would have sent one packet. The division of bandwidth can thus be altered by using a smaller margin (the share of S1 will decrease), or by using a different format for control messages, which would specify the number of packets that each incoming link can send.

We conclude that congestion can significantly reduce the performance obtained from the network, even in the presence of TCP. A MAC layer mechanism helps attenuate the performance degradation by reducing the number of packets dropped during periods of transient congestion. Moreover, preventing packet loss from happening helps achieving a more efficient use of resources, especially when the lost packets belong to flows coming from the WAN. Such packets would have already used resources along the way and better not be lost in the destinations' LANs. Furthermore, by using back-pressure, congestion can be moved out towards the boundaries of the LAN where it can be dealt with more efficiently. For example, stations can reduce their transmission rate by using their large memory resources to buffer packets, or by notifying higher layers to reduce the data generation rate. Routers can use elaborate techniques for congestion control or intelligent dropping [5, 6].

A similar performance gain as above is expected in situations where the controlled flows have similar congestion levels (e.g., similar file sizes, similar number of connections etc...). As we examine different situations in the following parts of this section, we show how such a non-discriminating mechanism could lead to significant performance loss.

3.2 MAC Address Back-Pressure

In this part, we look at situations where the asymmetry of the topology and/or the traffic conditions result in performance degradation when such a scheme is used. These situations suggest the need for control to be performed based on destination address information.

3.2.1 Unnecessary Control

In this scenario we show how, when control actions are enabled, the most congested path performs well, but degrades the performance of the others. Consider the

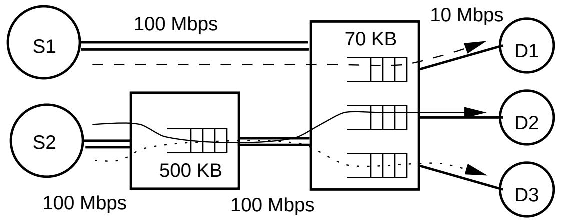

Figure 7: Destination Address-Based Differentiation, Scenario 1.

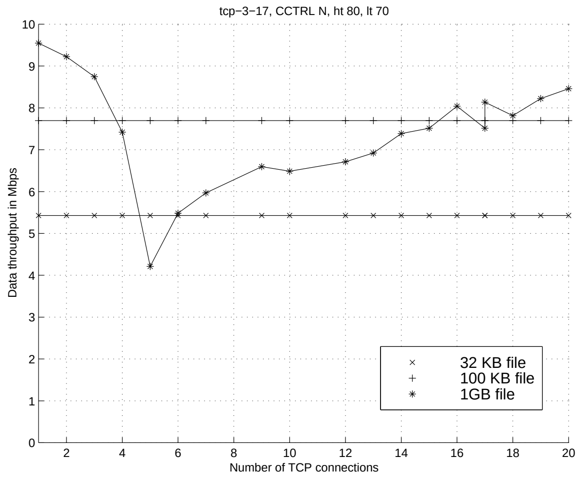

Figure 8: Destination Address-Based Differentiation, Scenario 1. Throughput achieved for each destination, versus the number of TCP connections between S1 and D1, without back-pressure.

scenario shown in Figure 7. Data server S1 is sending a set of large files to a destination station D1, using a varying number of TCP connections. Server S2 is sending files of size 32 KB to destination D2 and 100 KB to D3. The aggregation of many connections between S1 and D1 creates congestion on their path, while the S2-D2 and S2-D3 paths are not congested.

Without back-pressure, the 32 KB and the 100 KB connections[7] perform well (Figure 8), while the aggregate throughput for the S1-D1 connections is well below the maximum.

However, when back-pressure is enabled, the S1-D1 connections achieve maximum throughput while the control actions due to congestion on their path results in loss of throughput for the other two connections (solid lines, Figure 9). When control actions differ-

[7] These connections cannot reach the maximum achievable throughput given the small file sizes, the inter-file delays, the averaging procedure for calculating the throughput, as well as the startup procedure of TCP.

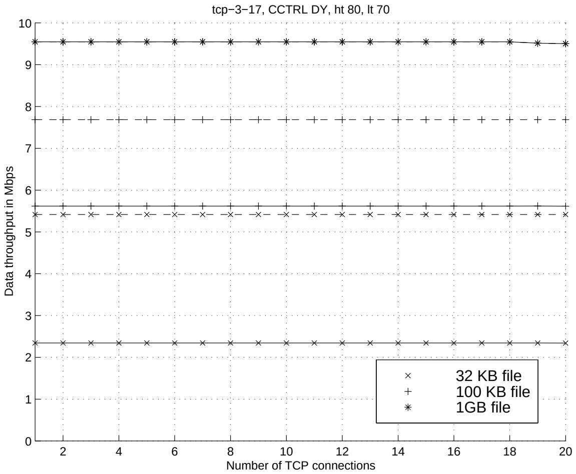

Figure 9: Destination Address-Based Differentiation, Scenario 1. Throughput achieved for each destination, versus the number of TCP connections between S1 and D1, with "simple" back-pressure (solid lines) and destination-address-based back-pressure (dashed lines).

entiate between destination addresses, the effects are eliminated: the connections between S1 and D1 achieve maximum throughput while the S2-D2 and S2-D3 connections achieve the same throughput as in no control case (dashed lines, Figure 9).

Note that if the switch could distinguish between input ports, and thus would just control the link from which the congestion causing connections are incoming, the control actions would not have negative effects in this particular case. However, since it is probable that some packets (e.g. low bandwidth broadcast traffic) would arrive from all links to the congested buffer, the differentiation between input links is not always possible.

A more serious effect of using simple control, is the added vulnerability to anomalous behavior of LAN devices. The effects on LAN performance of malfunctioning or non-conforming devices are amplified and propagated by non-discriminating control. To illustrate this idea, suppose that S1 did not respond to control messages. Then, when congestion is detected, the source keeps sending packets, while S2, which is behind a switch, has its connections blocked in that switch. The result is the starvation of the queues used by connections from S2 and the associated loss in throughput.

Therefore, some resilience to such behavior has to be built into the control scheme. There must be a way to reverse control actions even when congestion is not relieved if it appears that no response is obtained on some flow. It is also possible to single out such a flow by

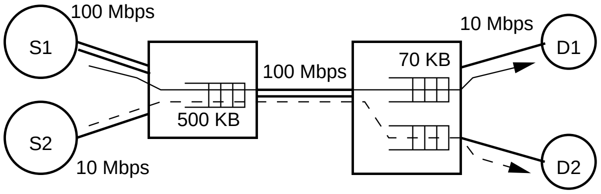

Figure 10: Destination Address-Based Differentiation, Scenario 2.

performing control based on destination address, thus reducing the harmful effects. However, this does not eliminate the effect of the flow on other traffic destined to the same address.

3.2.2 Sharing of Upstream Resources

In this scenario we study the effects of different levels of congestion on one flow path on other flows if non-selective control is used, and show how the sharing of upstream resources can lead to performance loss for these flows. Finally, consider the following scenario (Figure 10). Server S1 is sending a number of files to D1, using parallel TCP connections. S2 is sending one long file (equivalent to an infinite supply of data) using a TCP connection to D2.

Here, the path from S1 to D1 has a link speed mismatch while that between S2 and D2 does not. Therefore, The former is congested when connections are made in parallel, while the latter is not.

The aggregate throughput achieved between S1 and D1, when no flow control is performed is similar to the one shown in Figure 2. Clearly, the connection between S2 and D2 achieves the maximum throughput possible.

When back-pressure is enabled, the non-selective actions allow the connections from S1 to D1 to achieve the maximum throughput possible on the 10 Mbps link (results not shown). However, the unnecessary control of the connection from S2 to D2, due to its sharing of the 500 KB buffer with the S1-D1 connections, results in a loss of throughput, which is a function of the number of connections and the file sizes used (solid lines, Figure 11). The throughput loss increases with the number of connections and the file sizes sent from S1 to D1, and can be very severe, as S2-D2's share of the buffer space decreases. This effect can be eliminated if destination address-based control is used (dashed lines, 11), allowing both flows to achieve the maximum link utilization.

However, if the amount of buffering needed is such that the 500KB buffer usage exceeds the high threshold, both flows will be controlled, resulting in some

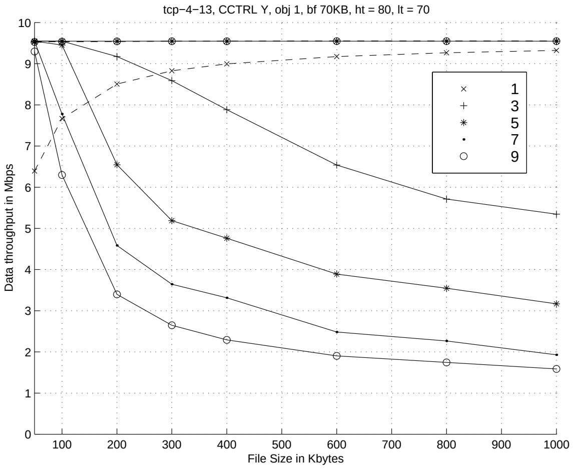

Figure 11: Destination Address-Based Differentiation, Scenario 2. Throughput achieved between S2 and D2 versus the file size used in the connections between S1 and D1, for selected numbers of such connections, solid lines: "simple" back-pressure, dashed lines: destination-address-based back-pressure.

throughput loss for D2-S2. This occurrence can be prevented by not allowing the buffer usage of blocked packets (i.e. which belong to controlled flow) to reach the high congestion threshold, e.g., by limiting it to the low congestion threshold. One would expect that if the flow to control is chosen by random selection out of the flows found in the queue, it would single out the S1-D1 flow, demonstrating the usefulness of such a selection method. This scenario also illustrates the fact that when non-selective control is performed, the most congested path dictates the performance of the others.

In conclusion, it appears that failing to discriminate between groups of flows can result in severely reduced throughput for flows which are unnecessarily controlled. In addition, the method for the selection of flows to control plays an important role in the performance. If all flows going through a buffer are selected when congestion is experienced, then it is important to place a limit on the amount of buffering space that can be used to store blocked packets at a given port. Otherwise, the propagation of congestion can result in reduced performance even when destination address is used to distinguish flows.

3.3 Traffic Class Back-Pressure

In the previous parts, we looked exclusively at scenarios with data traffic, which is relatively delay insensitive. If time sensitive traffic, such as video, is sharing

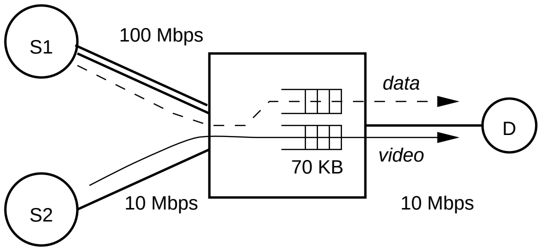

Figure 12: Multimedia Traffic.

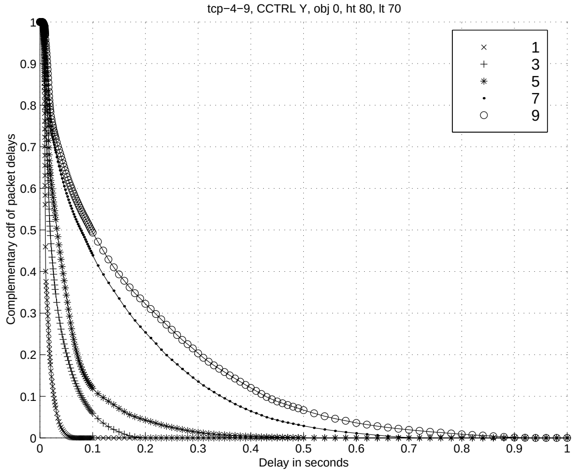

Figure 13: Multimedia Traffic: video packet complementary cdf, with "simple" back-pressure, for selected numbers of data connections.

the same links as data traffic, then the delaying effect of control actions becomes an important factor. Here, we attempt to answer the following questions: What are the situations where implementing back-pressure may be harmful to delay sensitive traffic? Then, is class differentiation in back-pressure necessary?

Figure 12 shows a scenario where the effect of non-discriminating control measures on video packets is particularly severe. S1, a data server, is sending 200KB files to D, which is downloading a video stream from S2. When no back-pressure is performed, the video traffic does not suffer from any delays. However, the data throughput achieved is well below the maximum, as expected[8]. Results are shown in Figure 13. Although video has priority over data and is using a separate buffer it is very severely delayed by the actions resulting from the congestion at the data buffer. In fact, if a 100 msec end to end delay limit is placed, more than 5% of the packets are lost for 3 connections

[8] Results for this case are not shown since similar results were presented in previous sections.

and about 50% for 9 connections. This effect is due to the reduction in the achievable throughput on the S2-D path as the congestion on the S1-D path increases. When this throughput falls below the average rate of the video, the quality of the latter is very severely affected. This scenario clearly demonstrates the need for control actions to differentiate between traffic classes.

Other experiments have shown that situations where the video traffic and the data connections causing the congestion share the same incoming link, the effect of the control action delay is not very significant due to the prioritization mechanism, unless the time delay before the actions are reversed is large. Therefore, it is necessary to keep the threshold margin as small as possible.

4 Conclusion

In this paper we provided scenarios where congestion occurs in LANs due to traffic burstiness. A hop-by-hop flow control mechanism allows the elimination of packet loss, thus bypassing TCP's costly timer-based flow control mechanism. However, we pointed to the fact that a simple, non-selective back-pressure mechanism could result in control actions affecting connections that are not involved in the congestion, leading to overall performance degradation. In addition, the buffering resources required may be very large, spreading the congestion throughout the network. To avoid this situation, control should use additional information to distinguish between different (aggregate) flows, such as MAC addresses. In addition, particular attention has to be given to the design of the different components of the control scheme, including buffering strategies and selection of the flows to be controlled.

Similarly, if control actions are performed independently of traffic class, congestion at one traffic class may significantly hurt traffic belonging to higher traffic classes, especially by delaying time sensitive traffic.

Finally, situations where the absence of source control results in severe degradation of network performance indicate the need to include in the control mechanism some measures that provide resilience to non-conforming behavior of network devices.

Acknowledgments

The authors wish to thank C. Fraleigh for his help in implementing TCP and the control modules in the simulator.

References

[1] IEEE 802.3z, *Media Access Control Parameters, Physical Layers, Repeater and Management Parameters for 1,000 Mb/s Operation*, in IEEE Standard 802.3, 1998 Edition.

[2] IEEE 802.3x, *Specification for 802.3 Full Duplex Operation*, in IEEE Standard 802.3, 1998 Edition.

[3] IEEE 802.1p, *Traffic Class Expediting and Dynamic Multicast Filtering*, in IEEE Standard 802.1D, 1998 Edition.

[4] IEEE Std 802.3ac-1998, *Frame Extensions for Virtual Bridged Local Area Network (VLAN) Tagging on 802.3 Networks.*

[5] Floyd S., Jacobson V., *Random Early Detection Gateways for Congestion Avoidance*, in IEEE/ACM Transactions on Networking, Volume 1, Number 4, August 1993.

[6] Floyd S., *TCP and Explicit Congestion Notification*, in ACM Computer Communication Review, Volume 24, Number 5, October 1995.

[7] Fall K., Floyd S., *Simulation-based Comparisons of Tahoe, Reno and SACK TCP*, in ACM Computer Communication Review, July 1996.

[8] Floyd S., Fall K., *Promoting the Use of End-to-End Congestion Control in the Internet*, in IEEE/ACM Transactions on Networking (to appear), May 1999

[9] Jacobson V., *Congestion Avoidance and Control.* In Proceedings of ACM SIGCOMM '88, Stanford, CA, Aug 1988.

[10] Jain R., *Congestion Control in Computer Networks, Issues and Trends*, in IEEE Networks, pp. 24-30, May 1990.

[11] Lefelhocz C., Lyles B., Shenker S., *Congestion Control for Best Effort Service: Why We Need a New Paradigm*, in IEEE Network, Volume 10, Number 1, January/February 1996.

[12] Ren J-F., Landry R., *Flow Control and Congestion Avoidance in Switched Ethernet LANs*, in Proceedings of the ICC'97, pp.508-512, June 1997

[13] Pazos C. M., Sanchez Agrelo J.C., Gerla M., *Using Back-Pressure to Improve TCP Performance with Many Flows*, UCLA.

[14] Wechta J., Eberlein A., Halsall F., *The Interaction of TCP Flow Control Procedure in End Nodes on the Proposed Flow Control Mechanism for Use in IEEE 802.3 Switches*, submitted to HPN 98, Vienna, Austria.

[15] Wechta J., Eberlein A., Halsall F., Spratt M., *Simulation Based Analysis of the Interaction of End-to-End and Hop-by-Hop Flow Control Schemes in Packet Switched LANs*, in Proceedings of the 15th UK Teletraffic Symposium on Performance Engineeringin Information Systems, Durham, UK, March 1998.

[16] Wechta J., Eberlein A., Halsall F., *An Investigation into the Performance of Switched LANs*, in Proceedings of the Conference on Networks and Optical Communications, Manchester, UK, 1998.