TCP over Load-Reactive Links*

Rajesh Krishnan and James P.G. Sterbenz†
BBN Technologies
10 Moulton Street, Cambridge, MA 02138, USA
{krash,jpgs}@bbn.com

Abstract

Emerging network technologies offer new design choices for performing traffic engineering and management, while introducing challenges for existing protocol control mechanisms. There is a tension between new hop-by-hop capabilities and existing end-to-end mechanisms. In this paper we examine one such capability, the Load-Reactive Link (LRL), which has capacity that can be automatically varied in response to the offered load. Before such adaptive links can be considered for deployment, it is essential to understand the behavior of widely deployed end-to-end protocols, in particular TCP. We study the behavior of TCP flows over a particular type of LRL that uses a hysteresis control mechanism for capacity allocation that reacts to current traffic loads. Our results indicate that reasonable performance for a single TCP flow can be achieved with careful choice of control parameters. We also show that poor choice of control parameters make it difficult or impossible for TCP to function. Preliminary simulations have been done with multiple TCP flows, but further investigation is needed to understand the behavior of the aggregation of a large number of flows.

1. Introduction and motivation

Emerging network technologies offer new design choices for performing traffic engineering and management, while introducing challenges for existing protocol control mechanisms. There is a particular tension between new hop-by-hop capabilities and existing end-to-end mechanisms. In general, traffic engineering and management methods can be classified into *static* or *dynamic* approaches based on the time granularity of control, into *control-driven* and *data-driven* approaches based on the locus of control, and into *per-flow* and *aggregate* approaches based on the flow granularity of control.

There is an increasing demand for a more sophisticated traffic engineering capability in the Internet, and mechanisms such as Multiprotocol Label Switching (MPLS) [17] are being considered. Furthermore, data-driven approaches can permit higher rates of change by avoiding signaling overheads. Data-driven approaches can optionally provide explicit feedback to the end terminals (e.g., as in the ATM ABR service class), or they can implicitly perform such operations and expect the end-to-end transport mechanisms to discover and react appropriately. The latter is compatible with the end-to-end flow and congestion control design philosophy of TCP.

Conventional links have a fixed capacity allocation during the lifetime of most traffic flows. The variation in link characteristics observed or inferred by individual flows is due to cross traffic and perhaps occasional routing topology changes. In these classical scenarios, the capacity of the link does not vary in response to the offered load. Emerging switched optical network infrastructure provides new opportunities for dynamic data-driven traffic management. In particular, optical sub-networks can support traffic management mechanisms that permit the capacity allocated to best-effort traffic to be varied frequently in response to the offered load [30]. This paper examines a traffic abstraction we call the *load-reactive link* (LRL) the characteristics of which can be automatically varied in response to the offered load, and examines whether TCP control mechanisms are compatible with the dynamic behavior of an LRL.

With conventional links, an increase in load is reflected to TCP control mechanisms as an increase in packet drops or delays. TCP infers packet losses as congestion and reacts by scaling down the flow rate. Techniques that allow TCP to discern between losses due to error and those due to congestion have also been proposed [10, 27]. With LRLs, however, there is an additional control mechanism at play. When the

---

*This work was supported by DARPA/ITO under Technology Investment Agreement TIA F30602-98-2-0194 with the National Transparent Optical Networks Consortium (NTONC), through a sub-contract issued by Nortel Networks to BBN Technologies.

†The authors wish to thank Craig Partridge (BBN) and the anonymous reviewers for their insightful comments that helped to improve this paper, Protima Chhabra (BBN) for her help with the simulations, and Maged Beshai (Nortel Networks) for encouraging a study of TCP over hysteretically controlled links.

offered load increases, the link will respond by increasing the allocated capacity, whenever possible. This results in a complex interaction between three control mechanisms: (i) the LRL can change capacity allocation in response to congestion, (ii) link queuing and discard disciplines can drop packets in response to increasing queue lengths, and (iii) TCP will scale back its flow in response to congestion inferred from packet delays/drops. These complex interactions can potentially result in control instability or sub-optimal link utilization.

In this paper, we examine the behavior of TCP over links in which the capacity can be changed dynamically in reaction to the offered load. The main focus is to study the behavior of a single TCP connection over a specific instance of an LRL based on hysteresis control. This is the first step to determine whether TCP can be used over such links, understand the performance implications, and determine reasonable LRL hysteresis parameters. Based on positive results for the single TCP connection case, we have begun looking at multiple TCP flows over a single LRL, and present preliminary results.

While we focus on conventional TCP flows in this paper, our work may be applicable to other situations as well. For example, in TCP Trunking [16] proposed by Kung and Wang, a single control TCP connection is used for flow and congestion control of aggregate traffic. This provides further motivation to explore the behavior of a single TCP flow over LRLs.

Dynamic, data-driven link capacity allocation can play a larger role in future internetworks and simplify traffic engineering for network operators, provided it can be shown that TCP/IP can work well over networks containing LRLs.

The rest of this paper is organized as follows. Section 2 describes relevant related work. In Section 3 we introduce a model for load-reactive links. We describe the simulation environment for studying TCP over LRLs in Section 4 and the results of the simulations in Section 5. We present a concluding summary and directions for future work in Section 6, followed by bibliographic references at the end.

2. Related work

The proposed LRL mechanism is motivated by a novel high-speed hybrid optical-electronic architecture called the PetaWeb [30] for which a similar hysteresis control has been proposed. Similar mechanisms have been proposed in the literature in the context of interworking LANs/MANs with B-ISDN [18] [5] [12]. Queues with hysteretically controlled service rates have been modeled in general [8] [7] [4] and applied to ATM networks [29].

Use-It-or-Lose-It policies were proposed in the context of the ATM UBR+ service to dynamically vary the capacity allocated for connectionless traffic. There exists considerable work on architectures and performance analysis related to interworking ATM networks with connectionless networks in general, and TCP/IP networks in particular (for example, see [26] [18] [5] [12]).

The Transmission Control Protocol (TCP) [24] is the reliable transport protocol widely used in the Internet and runs over an unreliable best-effort service provided by the Internet Protocol (IP). Since its introduction, TCP has been a subject of significant research and numerous modifications and variations have been proposed to optimize TCP. For an explanation of TCP congestion avoidance and control mechanisms, see [13] [28]. For TCP performance issues in general, and satellite links in particular, see [23]. For an analytical model of bulk TCP performance see [21] [20], and for an analysis of short TCP transfers, see [3]. The performance of increase and decrease algorithms similar to that of TCP has been modeled in general [6]; general control-theoretic flow control mechanisms applicable to TCP flow and congestion control have also been proposed [14]. The performance of TCP variants over fixed capacity links and wireless links has been studied extensively [9] [2] [25].

However, we do not know of any other work that attempts to model TCP over load-reactive links.

3. Load-reactive links

In this section, we describe an abstract model for a load-reactive link. We present a hysteresis control mechanism as a specific instance of a load-reactive link.

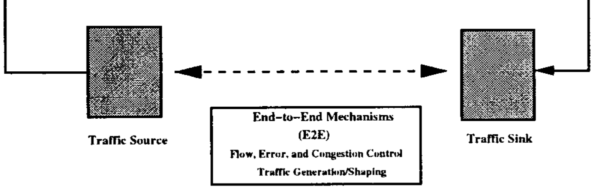

Figure 1. Load-reactive link

3.1. LRL abstract model

We propose a link abstraction called the load-reactive link (LRL) for the purpose of studying TCP behavior over links whose capacity varies dynamically in response to the offered load. Our proposed LRL model consists of a link and a buffer as illustrated in Figure 1. The characteristics of the link such as delay, capacity and error rate are time-varying functions that are controlled by an internal, load-reactive control algorithm (LRCA). The link buffer has an associated queuing and discard discipline (LQDD).

The LRCA has access to the state of the LQDD, which includes information such as the instantaneous and long-term statistics of the queue lengths. Optionally, the buffer length available can also be controlled by the LRCA. In particular, the LRCA can adjust the capacity allocated to connectionless traffic.

The LRCA can be realized using a variety of algorithms ranging from simple hysteresis control of allocated capacity based on instantaneous queue lengths (see Section 3.2) to more sophisticated strategies such as reinforcement learning based on long-term trends and dynamic policy iteration. The LRCA requires a number of algorithm-specific parameters. Even with a simple hysteresis control there are a large number of parameters to consider. These include the minimum capacity, maximum capacity, low threshold, high threshold, whether the thresholds are static or adaptive, capacity increments and decrements, whether the increments/decrements are static or adaptive, the recomputation interval, whether buffer occupancies are computed based on packets, bytes, or a combination thereof, and whether instantaneous values of queue occupancies are considered or a moving average is used.

The LQDD can range from simple tail drop strategies to more sophisticated strategies such as Weighted Fair Queuing (WFQ) [22] and Random Early Discard (RED) [11]. Closely related to, but not part of the LRL model are end-to-end (E2E) models for traffic generation, congestion control, and flow-control. In this paper, the specific E2E mechanisms of interest include TCP variants and optimizations.

The E2E sources can come from a number of application sources including HTTP, FTP, TELNET, transaction traffic, and streaming audio and video. They can also use different variations of TCP, or they can use RTP and UDP. Different congestion control mechanisms can be used with multicast traffic. Furthermore, the arrival and departure of flows can be modeled in a number of different ways.

Due to the large number of possible parameters in each of LRCA, LQDD, and E2E, as well as potential network topologies and traffic patterns, we restrict this presentation to a subset.

3.2. Hysteresis control for connectionless traffic

Consider a generic subnetwork architecture that can support both connection-oriented and connectionless traffic. In such a hybrid architecture, capacity can be allocated to connectionless traffic by setting up connections with reserved bandwidth (analogous to CBR, ABR with specified minimum rate, or VBR connections in ATM networks). Connectionless traffic can also be supported on a best-effort basis (analogous to UBR in ATM networks).

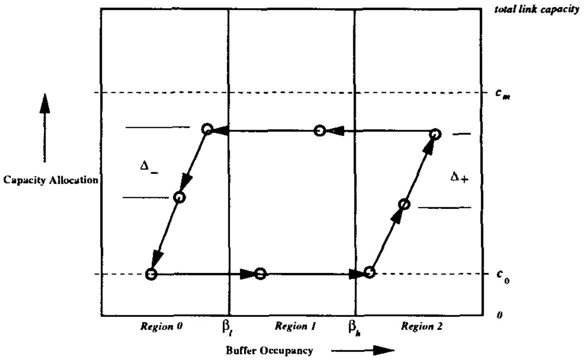

Figure 2. Hysteresis control of connectionless traffic

We can also set up LRL connections to carry best-effort, connectionless traffic. In such connections, capacity will be allocated based on the offered load. When the residual capacity of the link becomes zero, such connections can be optionally pre-empted to support new connections that request a higher grade of service. Limits can be set on the minimum and maximum link capacity available for LRL connections, depending on the specific allocation policy in use.

The capacity allocated to this LRL connection can be varied dynamically using a hysteresis control mechanism in accordance with the offered load. The buffer occupancy (in bytes, for example) can be used as a measure of the offered load. Such a mechanism is illustrated in Figure 2. The space of possible buffer occupancies is divided into three regions: 0,1, and 2. The low threshold mark, β_l separates regions 0 and 1; the high threshold mark, β_h, separates regions 1 and 2.

The initial capacity allocated is denoted by c_0, which can be zero. Initially, when there is no traffic, the buffer occupancy will be zero and this falls in region 0. As the load offered to the link increases, the buffer occupancy will also increase and cross the low threshold into region 1. If the offered load continues to increase further, the buffer occupancy will cross the high threshold into region 2.

The hysteresis control mechanism samples the buffer occupancy periodically. When the buffer occupancy crosses β_h, the hysteresis control mechanism will increment the capacity by a positive delta amount, Δ_+. In every sampling interval that the occupancy is found to be in region 2, the capacity will be incremented by Δ_+. Eventually the capacity will be increased sufficiently for the buffer occupancy to drop back into region 1. The capacity thus allocated is bounded by the residual capacity, c_m, available on the link.

On the other hand, when the offered load decreases, the buffer occupancy will decrease and eventually enter region 0. In every sampling interval that the occupancy is found to be in region 0, the capacity will be adjusted by a negative delta amount, Δ_-, until the buffer occupancy rises back and enters region 1 again or until the capacity allocated falls all the way down to the minimum value c_0.

As long as there is a sustained load being offered, the hysteresis control mechanism tries to keep the buffer occupancy in region 1. In order to do so, it dynamically adjusts the capacity allocated to match the current offered load.

We note that the proposed LRL based on hysteresis control of connectionless traffic does not map directly to any of the ATM traffic classes defined in [1]. However, it is possible to implement such a service over ATM or other packet switched architectures. Also, while other load-reactive link mechanisms are possible, we chose to study this specific hysteresis control mechanism since this work was done in the context of the PetaWeb [30].

4. Simulation environment

We implemented our simulations using ns-2 [19]. The initial focus was limited to a single TCP connection over a hysteresis link. The link MTU was set to 1000 bytes (the simulator default) and the one-way propagation delay of the link was set to 20 ms, representative of a high speed LAN link. The link queueing discipline was set to RED. TCP delayed acknowledgments were used. We assume that the host interface and memory bandwidth are not bottlenecks to TCP performance. The TCP connection duration was set to 58 s, a sufficiently large value relative to the round trip time, that allows the TCP flow to get past the slow start region.

We varied initial (and minimum) link capacity (c_0) values in the range 0–100 kbits/s. The buffer occupancy was counted in either packets or bytes. The byte thresholds were chosen to be multiples of the MTU. The low threshold (β_l) was chosen from the range 0–16000 bytes or 0–16 packets. The high threshold was chosen from the range 0–32000 bytes or 0–32 packets.

For the hysteresis computations, instantaneous samples of the queue length were directly compared against the thresholds. The queue sampling interval and the hysteresis recomputation interval were both set to 100 ms, based on values considered reasonable in [18]. We set both Δ_+ and Δ_- to 50 kbits/s. We set the residual capacity c_m to 100 Mbits/s in order to prevent it from becoming a bottleneck.

5. Results

With traditional fixed capacity links, there are two key aspects of TCP performance: First, can a single TCP connection utilize the available capacity efficiently? Second, does TCP share the link equitably in the presence of other flows? In the case of LRLs based on hysteresis control, there is an additional issue: Over a range of hysteresis control parameters, can TCP open the pipe — both in terms of its own congestion window as well as in terms of the capacity allocated by the link? Furthermore, if hysteresis control is used on a per-flow basis, how can equitable link sharing be achieved? In this section, we present performance results for a single TCP connection over a LRL using hysteresis control.

5.1. Overall behavior

First, we present graphs that illustrate the overall behavior of TCP over hysteresis control with appropriate parameter settings. Figure 3 illustrates the behavior with the low and high hysteresis thresholds set to 2000 and 6000 bytes respectively, and the initial (and minimum) link capacity is set to 20 kbits/s. The instantaneous link capacity, the average buffer occupancy, and throughput over 1 s intervals are plotted in the graph.

After an initial overshoot, we see that the buffer occupancy (lower curve) mostly stays between the thresholds, which is the goal of hysteresis control. We can also see that the link capacity allocated (dotted upper curve) increases from the initial 20 kbit/s to around 1 Mbit/s. This limit is due to the parameter values for this experiment; by increasing the minimum capacity and/or by lowering the high threshold we can increase the resulting throughput. Furthermore, the throughput (upper solid curve) closely tracks the instantaneous capacity indicating a high utilization.

The return link behavior for this case is shown in the graph in Figure 4. While the throughput in this direction still tracks the allocated capacity, more capacity is not allocated. The return link traffic consists only of TCP acknowledgments and the buffer occupancy in this direction stays well below the 6000 bytes. Therefore, the allocated capacity stays at the initial (and minimum) capacity of 20 kbits/s in this case. This suggests that the return link can potentially limit how much the forward link can be opened by a single flow, thereby granting an effective policy control mechanism if hysteresis control is used on a per-flow basis and byte-based buffers are used.

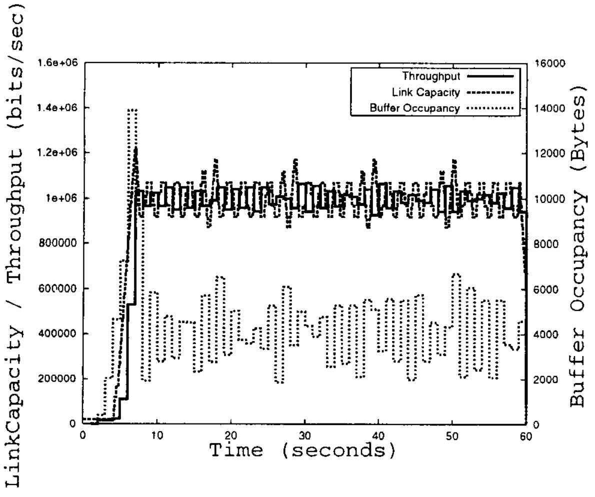

Figure 3. Overall forward link behavior of TCP over LRL

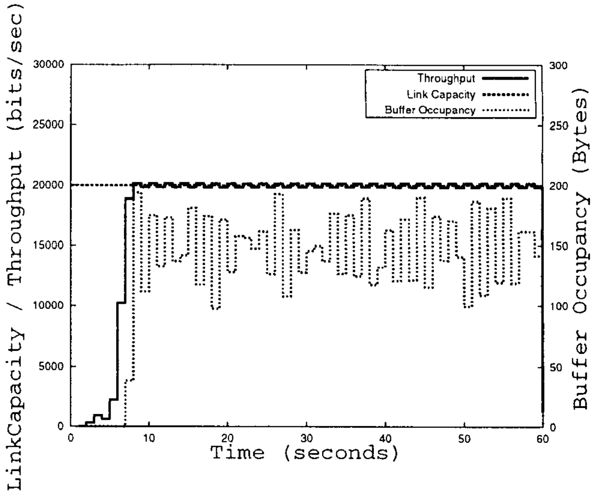

Figure 4. Overall return link behavior of TCP over LRL

5.2. Return path bottlenecks

Next, we address the issue of whether a single TCP connection over a load reactive-link can open the pipe when starting from a zero initial capacity. In Figure 5, we plot TCP throughputs (aggregated over an entire TCP flow) against initial (and minimum) link capacity. The curves are for various low hysteresis thresholds; the high thresholds are set to 4000 bytes above the respective low thresholds. The curves are similar for other high threshold settings that are larger than the low threshold by one MSS (1000 bytes in our case) or more, and these curves are omitted from this paper.

We can observe that over a range of hysteresis thresholds, the overall TCP throughput increases with the initial link capacity allocation. We also observe that allocating zero or low capacities initially can be highly detrimental to TCP performance. For the parameters used in the experiments, an initial capacity of 50 kbits/s or higher is required for good performance. Note that TCP will open its window only to the rate of the link, therefore with low initial capacities, the buffer fills up to a lesser extent. We discuss the effects of the threshold settings later.

With the next set graphs, we will answer why TCP is unable to open the pipe when the initial link capacity allocation is low. So far, the thresholds have been measured as the number of bytes in the buffer. We will now consider whether the performance will improve if thresholds are measured as the number of packets in the buffer.

In Figure 6, we plot TCP throughputs (aggregated over an entire TCP flow) against initial (and minimum) link capacity, when using packet thresholds. The curves are for various low hysteresis thresholds with the high threshold at 4 packets above the corresponding low threshold. With high initial link capacity, the curves are similar to those with byte-based thresholds. However, packet based thresholds yield better throughputs with relatively low initial capacities (2 – 50 kbit/s). The throughput curves for the return link (which carries TCP ACKs) are similar to, but of lower magnitude than the respective forward link counterparts.

The reasons for low throughputs at low initial capacities, and why counting packets is better can be explained by examining how link capacity allocation varies with time for a given TCP flow. Figure 7 contains plots of the instantaneous capacity allocation on the forward link for a single TCP flow, with the initial link capacity allocation being zero.

The curves correspond to various high thresholds, and with the low threshold of 1000 bytes for all cases. When the high threshold is small (solid line), the forward link capacity increases due to queue build up due to packet repeats on time-outs. The corresponding capacity allocations on the return link always stays at zero. This is because TCP slow start will not open the window beyond the initial value unless ACKs are received, and the return link will not open unless enough ACKs are queued. This results in a deadlock-like situation. With the high threshold set to 3000 bytes, the link capacity stays at zero for the length of the simulation. Eventually, enough timeouts will occur, and the reverse queue will build up and the reverse link will open. The forward link behavior with packet-based thresholds is similar. With byte-based thresholds, the return link capacity stays at the initial value. However, with packet-based thresholds, the return link capacity varies, occasionally exceeding 56 kbits/s. These plots are omitted for brevity.

ACKs are typically shorter than the link MTU (40 vs.

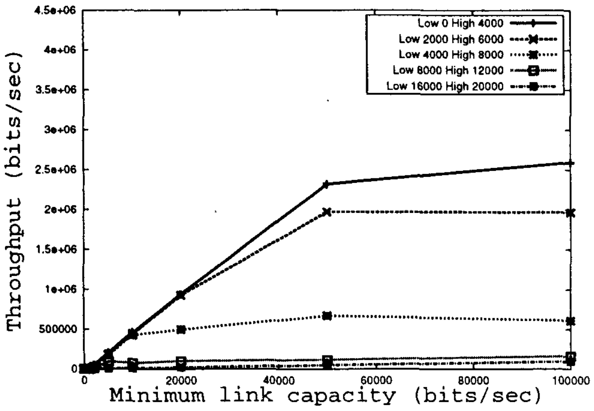

Figure 5. Initial capacity vs. throughput, with byte thresholds

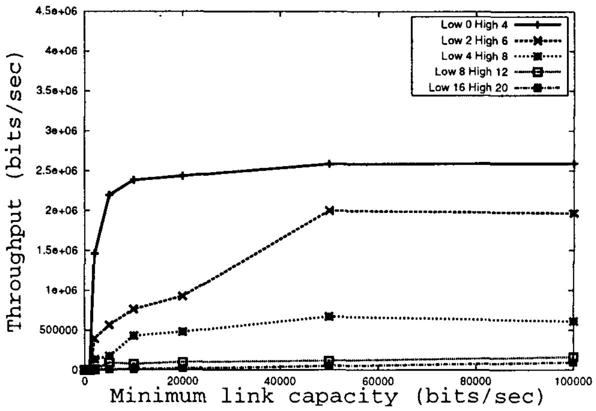

Figure 6. Initial capacity vs. throughput, with packet thresholds

1000 in our experiments). TCP uses MTU sized segments to avoid the silly window syndrome. With byte-based buffer occupancy thresholds, the queue length on the return link does not exceed the threshold. So the return link never opens up and becomes a bottleneck[1]. When counting in packets the return link has a chance to open up. With a high threshold of 2000 bytes or 2 packets, 2 MSS sized packets can meet the high threshold in either case. However on the return link, it will take 50 ACKs to match the high threshold with byte thresholds, whereas it will take only 2 ACKs to do the same with the packet threshold.

However, at very high initial capacities, the return link is not a bottleneck with either byte or packet thresholds. TCP parameters such as the MTU, slow start threshold, and the receiving window, can limit TCP throughput in this case, and high speed TCP options become necessary.

5.3. Instability at very high thresholds

Next, we study the effect of high threshold settings. In Figure 8, we plot the instantaneous link capacity for the length of the run, for different high threshold values. The low threshold value was set to 1000 bytes.

We observe that in general, larger the high threshold, the longer it takes for the queue to build up and for hysteresis control to allocate more capacity. More importantly, we note that setting the high threshold to a very large value (e.g., 16000 bytes) can cause unstable interactions between TCP and hysteresis control. Occasionally, when the queue drops below the low threshold momentarily, the link capacity can rapidly drop to the minimum value. If this happens, it will take a long time once again before the queue can build up sufficiently to allocate more capacity. Thus the link oscillates between periods when it is open and when it is at the minimum capacity, as shown by the curve for the 17000-byte high threshold in Figure 8.

As illustrated in Figure 9, large packet-based high thresholds result in behavior similar to large byte-based high thresholds. However, packet based thresholds can lead to higher throughputs by allowing the return link to open up.

5.4. Multiple TCP flows

To understand how multiple TCP flows share an LRL based on hysteresis control, we performed simulations with 1–16 sessions on the same LRL. Figure 10 is a plot of throughput for different values of c_0, with low threshold β_l = 4000 bytes and high threshold β_h = 8000 bytes for different numbers of TCP flows. The plots are similar over a wide range of thresholds.

When we increase the number of TCP flows (for example from 1 to 4), more load is offered and hence more capacity gets allocated by the LRL. This is reflected in an increase in the aggregate throughput. However, when too many flows are added to the link (for example 8 or 16), the aggregate throughput (and the LRL capacity allocation) does not continue to increase in proportion to the number of flows.

We believe that the aggregate load offered causes temporary queue build-ups causing more packets to get dropped, before the LRL can allocate sufficient capacity. Meanwhile, the TCP connections react by reducing the send rate, and

---

[1] One reviewer observed that it may be possible to exclude control traffic such as ACKs from the scope of the hysteresis control. In this paper, we assume that all packets corresponding to the TCP flow use the same load-reactive link.

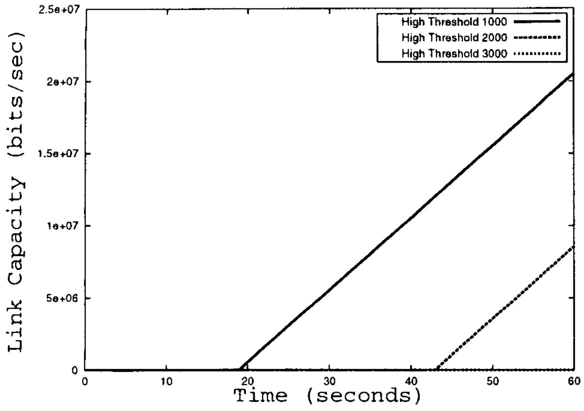

Figure 7. Forward link capacity vs. time, zero initial capacity

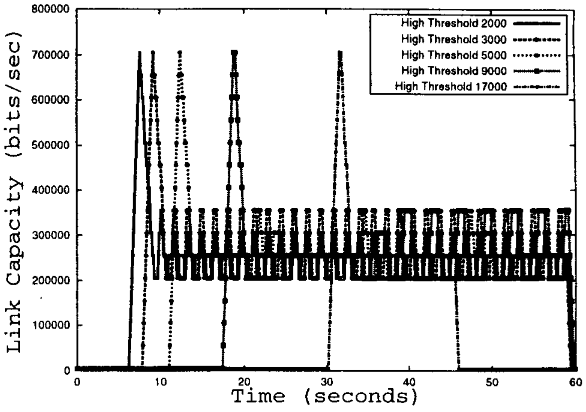

Figure 8. Effect of large byte values of the high threshold

the queus drop below the high threshold; therefore, the LRL perceives a lower offered load and does not allocate aggressively enough. Further investigation is needed to isolate the effects of router buffer sizes (which was set to 50 packets in our experiments) and RED thresholds to clearly understand the aggregation of a large number of TCP flows.

6. Conclusion

In this paper we introduced an abstract traffic management mechanism called the LRL, motivated by the capabilities of emerging networks. We described hysteresis control of capacity allocation of connectionless traffic as a specific instance of an LRL. Given such a mechanism, we studied the performance implications to TCP.

Using simulations, we studied the behavior of a single TCP flow over an LRL based on hysteresis control. Simulations suggest that with appropriate parameter settings, a TCP connection can perform reasonably well over such links. However, we also observed that TCP cannot open the LRL pipe from a zero initial capacity and that the link can become unstable if the high hysteresis threshold settings are too large. We have begun to look at multiple TCP flows over the same LRL, and observe reasonable behavior for a small number of flows, but further investigation is needed to understand the aggregation of a large number of flows, including both TCP and non-TCP flows.

This work is the first step in understanding whether TCP can operate over such a dynamic traffic class. A number of related research areas remain open to exploration. It will be useful to combine analytical models for TCP with those for hysteretic queues. Future work may consider the behavior of multiple flows within an LRL, performance of a network of LRLs, alternative LRL mechanisms, and implementation in a network testbed. For a more detailed discussion of these issues, see [15].

LRL mechanisms have the potential to simplify traffic management in future internetworks. However, additional work is necessary to characterize their performance and to determine their applicability to TCP/IP networks.

References

[1] Traffic Management Specification (version 4.1) AF-TM-0121.000. *ATM Forum Technical Committee*, March 1999.

[2] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, December 1997.

[3] N. Cardwell, S. Savage, and T. Anderson. Modeling TCP Latency. *Proceedings of IEEE INFOCOM 2000, Tel-Aviv, Israel*, pages 1742–1751, March 2000.

[4] S. Chakravarthy. Analysis of the MAP/PH/1/K queue with service control. *Applied Stochastic Models and Data Analysis*, 12(3):179–191, 1996.

[5] E. Chan, V. Lee, and J. Ng. On the Performance of Bandwidth Allocation Strategies for Interconnecting ATM and Connectionless Networks. *ACM Computer Communications Review*, 26(1):29–38, January 1996.

[6] D.-M. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

[7] J. Dshalalow. Queues with Hysteretic Control by Vacation and Post-Vacation Periods. *Queuing Systems*, 29(2/4):231–268, 1998.

[8] J. Dshalalow and E. Dikong. On Generalized Hysteretic Control Queues with Modulated Input and State Dependent

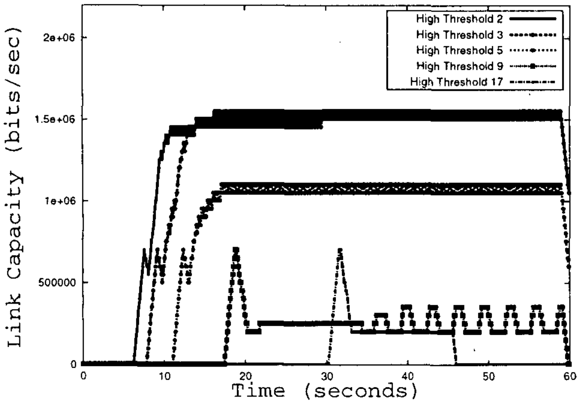

Figure 9. Effect of large packet values of the high threshold

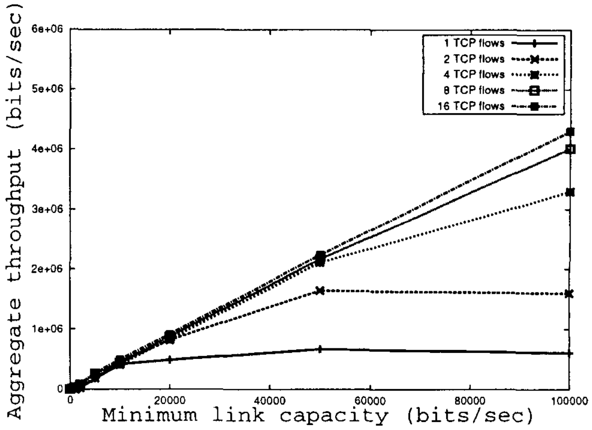

Figure 10. Throughputs for multiple TCP flows on the same LRL

Service. *Stochastic Analysis and Applications*, 17(6):937–961, 1999.

[9] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. *ACM Computer Communications Review*, 26(3):5–21, July 1996.

[10] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communications Review*, 24(5):10–23, October 1994.

[11] S. Floyd and V. Jacobson. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[12] M. Gerla, T.-Y. Tai, and G. Gallasi. Internetting LANs and MANs to B-ISDNs for Connectionless Traffic Support. *IEEE Journal on Selected Areas in Communications*, 11(8):1145–1159, October 1993.

[13] V. Jacobson. Congestion Avoidance and Control. *Proceedings of ACM SIGCOMM '88, Stanford, CA*, pages 314–329, August 1988.

[14] S. Keshav. A Control-Theoretic Approach to Flow Control. *Proceedings of ACM SIGCOMM '91, Zurich, Switzerland*, pages 3–15, September 1991.

[15] R. Krishnan. TCP over Load-Reactive Links. *BBN Technical Memo, BBN-TM-1281, BBN Technologies, Cambridge, MA, USA*, February 2001. http://www.ir.bbn.com/documentation/techmemos/index.html.

[16] H. Kung and S. Wang. TCP Trunking: Design, Implementation, and Performance. *Proceedings of IEEE ICNP '99*, November 1999.

[17] G.-S. Kuo. (Guest Editor). Feature topic on Multiprotocol Label Switching. *IEEE Communications Magazine*, 37(12):36–68, December 1999.

[18] L. Mongiovi, M. Farrell, and V. Trecordi. A Proposal for Interconnecting FDDI Networks Through B-ISDN. *Proceedings of IEEE INFOCOM 1991, Bal Harbour, FL, USA*, pages 10B.4.1–10B.4.8, April 1991.

[19] UCB/LBNL/VINT Network Simulator - ns (version 2). http://www.isi.edu/nsnam/ns/.

[20] T. Ott, J. Kemperman, and M. Mathis. The stationary behavior of ideal tcp congestion avoidance. August 1996. Unpublished manuscript, ftp://bellcore.com/pub/tjo/TCPwindow.ps.

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145, April 2000.

[22] A. Parekh and R. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[23] C. Partridge and T. Shepard. TCP Performance over Satellite Links. *IEEE Network*, 11(5):44–99, September/October 1997.

[24] J. e. Postel. Transmission Control Protocol. *Request for Comments: 793*, September 1981.

[25] L. Qiu, Y. Zhang, and S. Keshav. On Individual and Aggregate TCP Performance. *Cornell CS Technical Report TR99-1744, Cornell University, NY, USA*, May 1999.

[26] A. Romanow and S. Floyd. Dynamics of TCP Traffic over ATM Networks. *IEEE Journal on Selected Areas in Communications*, 13(4):633–641, May 1995.

[27] N. Samaraweera and G. Fairhurst. Explicit loss indication and accurate rto estimation for tcp error recovery using satellite links. *IEE Proceedings on Communications*, 144(1):47–53, February 1997.

[28] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. *Request for Comments: 2001*, January 1997.

[29] S. Thiagarajan and S. Chakravarthy. Analysis of Hysteresis based Priority Queuing Strategy for ATM Networks with batch Markovian Arrivals, - Numerical Results. *Technical Report, Department of IMEB, Kettering University, MI, USA*, 2000.

[30] R. Vickers and M. Beshai. Petaweb Architecture. *Networks 2000, Toronto, Ontario, Canada*, September 2000.