November 2004

# The price we pay for using TCP

D. King
*Computer Sciences Corporation*

K. Walker
*University of Wollongong*

D. Platt
*University of Wollongong*, dplatt@uow.edu.au

# The price we pay for using TCP


## Abstract

This paper examines a hop-by-hop flow control scheme and compares it with the operation of TCP end-to-end flow control. The individual link control scheme is described. Both schemes are simulated under bursty traffic in a network with 50 nodes and 96 links. It is found that the hop-by-hop scheme results in similar levels of utilization to TCP, but that the average queue size is 30 times smaller.



## Disciplines

Physical Sciences and Mathematics

## Publication Details

This article was originally published as: King, D, Walker K & Platt, D, The price we pay for using TCP, Proceedings 12th IEEE International Conference on Networks ICON 2004, 16-19 November 2004, vol 1, 9-13. Copyright IEEE 2004.

# THE PRICE WE PAY FOR USING TCP

Dale King Karen Walker and Don Platt
Computer Sciences Corporation University of Wollongong

**Abstract - This paper examines a hop-by-hop flow control scheme and compares it with the operation of TCP end-to-end flow control. The individual link control scheme is described. Both schemes are simulated under bursty traffic in a network with 50 nodes and 96 links. It is found that the hop-by-hop scheme results in similar levels of utilization to TCP, but that the average queue size is 30 times smaller**

## 1. INTRODUCTION

Modern backbone networks are characterised by optical links with high bit rates and buffers, which consist of high speed electronic memory. Frequently the links extend over long distances. Within these networks, there is no control of packet flow rates. Flow control has been consigned to an end-to-end control function, which is largely resident at the user terminals.

The decision to remove flow control from the network was taken with the advent of frame relay, and was continued with such technologies as ATM and TCP. Optical signal transmission technology was just beginning to increase the available bit rates for long distance links, but the bit rates were still very low. Frame relay used a maximum bit rate of 2.44 Mbps. Before this time, flow was controlled on a hop-by-hop basis in an earlier system known as X.25 [1].

Once the ATM and TCP standards were in place, the telecommunications system worldwide developed at the extraordinary rate we have seen over the last 25 years. Much effort has been expended to improve the control of both these technologies, but little time has been taken to reconsider the decision to implement end-to-end control.

Nevertheless, there has been ongoing research on the hop-by-hop system. Özveren, Simcoe and Varghese 1995 [2] describe a method for sharing available buffers among flows on a link, allowing the memory requirement to be reduced by an order of magnitude. The paper suggests that some form of buffer sharing technique must be used if a datagram network is ever going to benefit from hop-by-hop congestion control.

Of particular interest, are hop-by-hop controllers for best-effort traffic such as work done on the ATM 'Available Bit Rate' (ABR) service. Since this service provides no guarantees on data rate, it is quite similar to TCP. Zhang and Yang 1997 [3] have proposed a hop-by-hop rate-based flow control scheme for networks where the distances between any pair of neighbouring nodes are not necessarily the same. Using a discrete-time fluid model, it was shown that with their controller design applied, traffic flow and buffer occupancies were asymptotically stable and exhibit good transient behaviours.

The so-called "blocking" problem has been researched by Bohacek [4] and a hop-by-hop scheme was developed so that, with properly chosen feedback gains, a system will be stable and by design will not suffer from the blocking problem.

However, this paper will demonstrate that end-to-end control is inferior to hop-by-hop control, and the price we pay for using this control is higher packet loss, larger buffers throughout the networks, and higher delay time.

## 2. HOP-BY-HOP CONTROL

We assume that the network switches are input queued, although a small output queue may be involved. The essential function of hop-by-hop control is to keep the size of the queue at any input as close as possible to a set point value, while, at the same time, allowing free flow of packets through the queue. The size of each input queue and the processing rate off the front of each queue are monitored regularly and this information is used to determine what the flow rate should be into the queue. This information is fed back to the previous switch, which feeds the queue, and is used to set the flow rate from that switch. This arrangement is sketched in Figure 1.

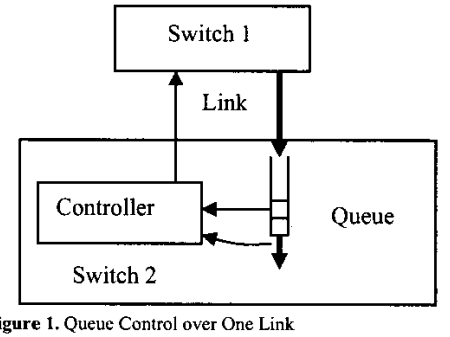


**Figure 1.** Queue Control over One Link

0-7803-8783-X/04/$20.00 © 2004 IEEE

In the simulation and modeling used for this paper, the link controller is a simple linear controller, and has a fixed set point for queue length. The effect of the control system is always to return the queue size to the set point value. The normal operation of the network will continually introduce disturbances to move the queue size away from the set point. Figure 2 shows a block diagram of the control system.

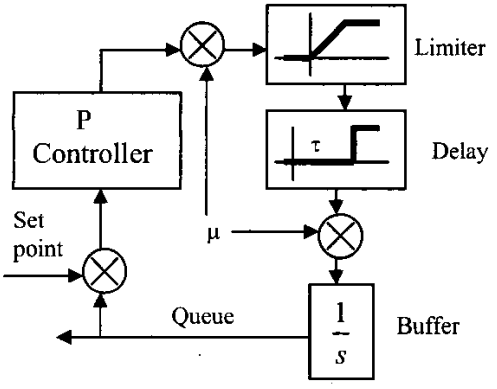


**Figure 2.** Controller Block Diagram

The limiter models the channel capacity. The lowest possible bit rate is zero, and the highest possible is the channel capacity. The delay is the total round trip delay, $\tau$, which includes transport time in both directions and also any processing time at the sending switch. The integrator (buffer) models the queue itself.

The service rate, $\mu$, reduces the queue directly, but is also included in the calculation of the send rate from switch 1, and so is also present in the delayed signal.

This control system is simple to implement and provides quick and accurate control, with response time in the order of the delay, $\tau$, and a small amount of overshoot.

The maximum queue size can be predicted with confidence. It is simply the set point plus the peak flow rate times the round trip time, $\tau$.

The level of the queue set point needs to be considered carefully. If it is set to zero, then the queues are often empty, and the switch cannot respond to a command to send more packets. This results in a reduced utilization of the network. In this paper, we will use a set point equal to the link flow rate times the round trip time for each link.

## 3. MODEL OF A NETWORK NODE

In order to compute the processing rate, it is necessary to specify the scheduling discipline in each network node. We assume that each switch/router has N inputs and N outputs. It is thus called an NxN switch/router. In order to avoid the "head of line" problem, the input queue at each input is divided into N subsidiary queues. This allows a packet intended for any output line to be selected in any packet time slot. In the work reported here, it has been assumed that the scheduling discipline produces a flow proportional to the length of each input queue. This is a reasonable approximation for the operation of the switching fabric, although in practice one encounters several variations in the scheduling strategy. All the modeling assumes only one class of traffic.

Figure 3 shows the model for a 2x2 switch. There are two queues, $q_{11}$, and $q_{12}$, at input 1, one for each of the two outputs. Together, they make up the total queue of all packets coming into input 1 ($q_1 = q_{11} + q_{12}$). It is this total queue which is controlled by the controller of Figures 1 and 2.

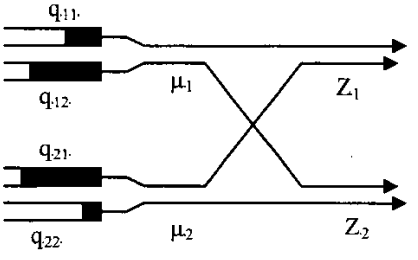


**Figure 3.** Switching for 2x2 Switch

The servicing rate off the front of this queue is $\mu_1$, and this is not controlled by the queue controller, but depends on the line rates of the outputs, $Z_1$ and $Z_2$.

$$Z_1 = k_1 (q_{11} + q_{21}), \quad Z_2 = k_2 (q_{12} + q_{22}) \qquad (1)$$

for some values, $k_1$ and $k_2$, and

$$\mu_1 = k_1 q_{11} + k_2 q_{12}, \quad \mu_2 = k_1 q_{21} + k_2 q_{22} \qquad (2)$$

Equation (1) constrains the flow from any input queue (virtual output queue) to any output line, to be proportional to the input queue size. The output flow rate, $Z_1$, is set by the link controller on the downstream switch. Within the switch, the flow rates from each of the two input queues, $q_{11}$ and $q_{21}$, are determined to be proportional to the queue sizes, and to add up to the output flow rate, $Z_1$. The servicing rate, $\mu_1$, at input queue 1, is the sum of the flow rates from all the queues attached to input line 1.

The link controller for input 1 then controls the flow rate from the upstream switch. When a network is controlled in this way, it is always the downstream nodes which control the flow rates.

## 4. SIMULATION

For the purposes of demonstration, the 50 node network, shown in Figure 4, was selected for simulation. It has 50 nodes and 96 links. It was generated by placing nodes randomly through an area roughly comparable to the USA. Simulations were carried out using the well known network simulator, NS-2, and a custom program which simulates hop-by-hop control. This custom program uses a fluid flow model with a time step equal to one tenth of the lowest round trip time in the network [5, 6]. NS-2 used TCP as the end-to-end flow controller. Five of the nodes were used as traffic sources, ten as destinations, and the other 35 nodes simply operated as switches. Twenty-two individual traffic flows, crossing the network from edge to edge, were set up (one shown in heavy). For each source/destination (s/d) pair, two single hosts were specified. In order to allow NS-2, with TCP to operate at reasonable utilization levels, it was necessary to specify the link bandwidth at 100 kbps.

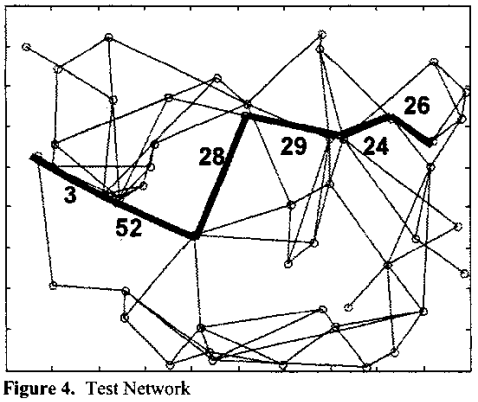


**Figure 4.** Test Network

Over the time of the simulation, the routes between every pair of nodes were fixed. The variables recorded are as follows: queue size at each input, incoming flow rate and queue servicing rate at each input, and average utilisation of each link.

A start-up test was carried out, in which the network started with no packets anywhere in the system, and steady traffic was offered to each s/d pair. This test gives an indication of the time it takes the network to respond to large changes in traffic load under the two different control regimes.

A second test was carried out, where the traffic offered to all s/d pairs over all routes is bursty.

## 5. RESULTS

Figures 5 and 6 show the result for link 3, under start-up conditions for hop-by-hop control. While the incoming rate exceeds the outgoing rate, the queue increases. After a transient time of about 1 second, the utilization reaches a steady state of about 50% and the queue level rises to be equal to the set point, about 12 kbits, after a slight overshoot. The link controller is operating as intended.

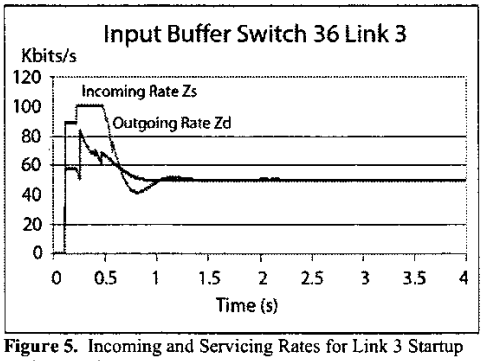


**Figure 5.** Incoming and Servicing Rates for Link 3 Startup Under Hop-by-Hop

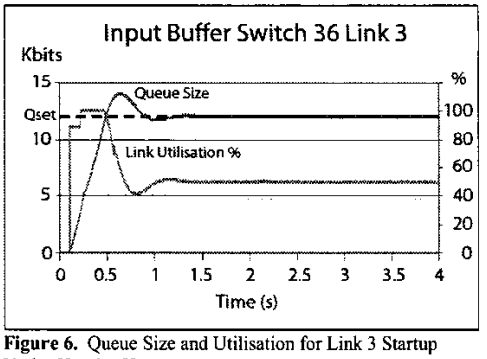


**Figure 6.** Queue Size and Utilisation for Link 3 Startup Under Hop-by-Hop

Figures 7 and 8 show the results for link 52. This is a heavily congested link. In the first part of the transient, the incoming rate exceeds the servicing rate, and the queue increases. At about time = 0.4 seconds, the servicing rate exceeds the incoming rate, and the queue size decreases. During this time, the servicing rate exceeds the link bandwidth, and the incoming rate goes to its limit, where it stays. After the queue disappears, it is impossible for the servicing rate to exceed the bandwidth.

Clearly the queue controller is unable to take the queue to its set point, because the link is congested, and unable to transport packets into the queue as fast as the demand for them elsewhere in the network.

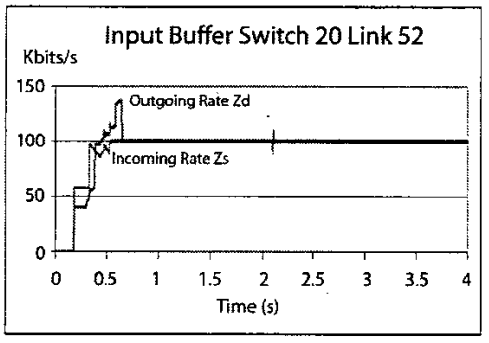


**Figure 7.** Incoming and Servicing Rates for Link 52 Startup Under Hop-by-Hop

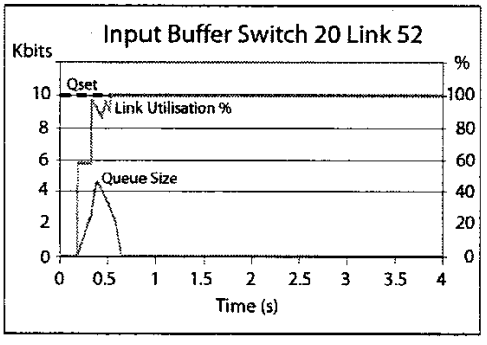


**Figure 8.** Queue Size and Utilisation for Link 52 Startup Under Hop-by-Hop

Figures 9 and 10 show the result for link 3 as reported from NS-2, under TCP control. These figures show the usual slow start pattern. What is notable is that the flow rates have not settled into any steady pattern even after four seconds, and that the queue shows several spikes, up to about 5 kbits. Other links show similar behaviour.

When the traffic is bursty, the situation is more difficult to understand. Figure 11 shows the results for link 3 again. The graph for utilization shows that the link steps back and forth between high utilization, frequently 100%, and zero. The queue remains quite small, always less than 20% of the set point. This is typical of the behaviour of all links in the network, although the queue size shows some variation depending on load patterns and position within the network.

Under this kind of traffic, it is necessary for the switches to have some packets in the queues so as to be able to send them when they are given the opportunity. For significant stretches in Figure 11, for example from time = 2.4 seconds to 2.6 seconds, the incoming flow rate is zero, and the queue is also zero. This stream is therefore unable to contribute to any flow leaving the switch. Under these conditions, the utilization of the outgoing links may be reduced from its possible levels.

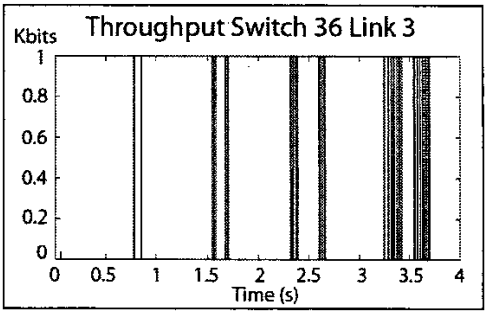


**Figure 9.** Throughput for Link 3 Startup Under TCP

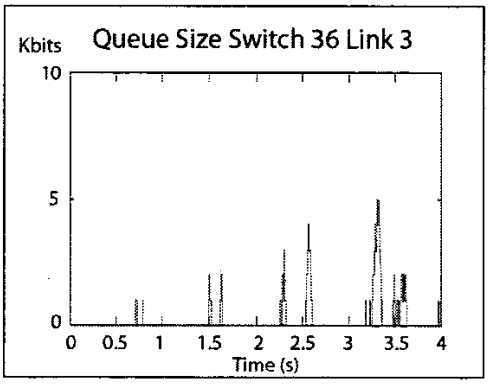


**Figure 10.** Queue Size for Link 3 Startup Under TCP

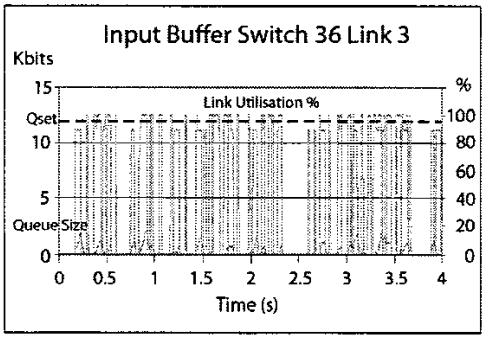


**Figure 11.** Queue Size and Utilisation for Link 3 Bursty Under Hop-by-Hop

By comparison, Figures 12 and 13 show the behaviour of the flow and queue in the same link, when running TCP. The queue reaches levels of 11

kbits, compared to about 2 kbits for the hop-by-hop scheme.

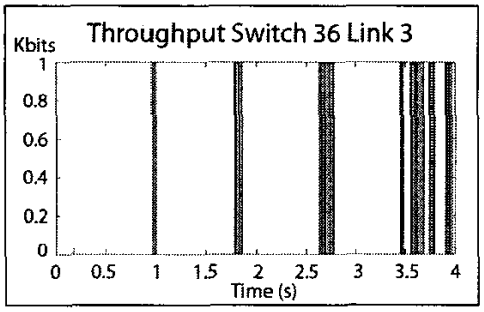


**Figure 12.** Throughput for Link 3 Bursty Traffic Under TCP

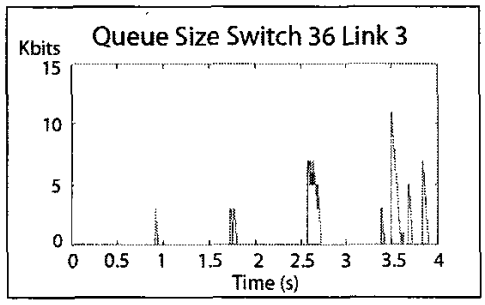


**Figure 13.** Queue Size for Link 3 Bursty Traffic Under TCP

Taking all links into account, the average utilisation across the network is 26.1% using hop-by-hop control and 32.6% using TCP. The average queue size is 0.058 kbit using hop-by-hop, but it is 30 times greater, at 1.73 kbit using TCP.

In order to translate these figures to a modern network, using links with bit rates of 10 Gbps, it is necessary to multiply the queue sizes by $10^5$, making the average queue lengths equal to 580 kbit for hop-by-hop and 173 Mbit for TCP.

## 6. COMMENTS AND CONCLUSIONS

It has been the intention of this paper to keep alive the issue of hop-by-hop vs end-to-end flow control through communications networks. We have described a control system for any network or combination of networks, which is conceptually easy to implement. In our simulations, we have shown that this system responds more quickly to load changes than TCP, provides similar levels of utilization, and requires something like 30 times less buffer space across the whole network.

If it was necessary, it would be a fairly simple matter to adjust the set points used throughout the network and in so doing, to increase both the average utilization, and probably the average queue size.

The simulations presented here involve a network with only 50 nodes. As networks become bigger, the number of nodes increases, and the queues produced using TCP become larger. On the other hand, the queues produced using a hop-by-hop system should not change. They should depend only on their set point, link bandwidth and the round trip time.

## REFERENCES

1. W. Stallings, *'Advances in High-Speed Networking'*, ACM Computing Surveys, Vol. 28, No. 1, March 1996

2. M. Özveren, R. Simcoe and G. Varghese, *'Reliable and Efficient Hop-by-Hop Flow Control'*, IEEE journal on selected areas in communications, Vol. 13, No. 4, May 1995

3. H. Zhang and O. W. Yang, *'The Hop-by-Hop Flow Controller for High-Speed Networks: Single VC Case'*, Global Telecommunications Conference, IEEE, Vol. 2, pp. 785-789, Nov 1997

4. S. Bohacek, *'Stability of Hop-by-Hop Congestion Control'*, Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, December, 2000

5. L. Benmohamed, S. Meerkov. *'Feedback Control of Congestion in Packet Switching Networks: The Case of a Single Congested Node'*, IEEE/ACM Transactions on Networking, Vol. 1, No. 6, pp. 693-708, December 1993.

6. P. Mishra, H. Kanakia and S. Tripathi., *'On hop-by-hop rate-based congestion control'*, IEEE/ACM Transactions on Networking., Vol. 4, No. 2, pp. 224-239, April 1996.