Network Performance Technology Whitepaper

Authored by Foursticks CTO Alan Noble

For Equator One Pte Ltd

March 2003

Table of Contents

1 Introduction

As businesses have come to depend on client-server and distributed enterprise applications, the network has become the lifeblood of the company. Business operations demand reliable yet cost-effective networks; as network problems result in lost revenue, increased costs, or both. Solutions have been developed to guarantee and enhance the performance of networks by controlling the traffic they carry. These are known variously as quality of service (QoS), bandwidth management, traffic shaping and, most recently, network performance (NP) solutions; the term we shall use. This whitepaper describes the different technologies used to implement network performance solutions on Internet Protocol (IP) networks, and highlights the advantages of the Hierarchical Credit-based Queuing (HCQ) approach.

2 First Generation Network Performance

Fundamentally, there are only two techniques to control IP network traffic, namely TCP rate control and queuing. In this section we will show the differences between the two approaches. We show that while TCP rate control handles TCP/IP traffic, it is inadequate as an overall traffic control mechanism in that it does not take into account all network traffic. On the other hand, standard queuing techniques, while in general very efficient, have some limitations with respect to real-time traffic.

2.1 TCP Rate Control

In general, rate-based traffic control schemes modify the network transmission rate whenever congestion occurs. The traffic controller tells the network what transmission rate the sending device should use. When the network becomes congested, it reduces the rate, and when the network is no longer congested, it increases the rate.

The TCP protocol has the following properties that are used for rate control:

- TCP senders rely on acknowledgements from the receiver to determine that no packets have been lost during transmission, and
- TCP has a sliding window that specifies the maximum amount of data that the sender can send without receiving an acknowledgement from the receiver. The so-called *window size*, which, is measured in bytes, is the available bandwidth multiplied by the round-trip time[1].

TCP rate control works by modifying packets so as to alter TCP window size as follows:

- Predict the round-trip time for *each stream* of traffic.
- Intercept acknowledgement packets sent by the receiver to the sender and hold them for a period of time, based on the predicted round-trip time.
- Modify the sliding window size advertised in the header of the acknowledgement packets, thus affecting the size of the packets and hence rate that will be transmitted by the sender.

2.2 Queuing

Queuing schemes do not modify the packets flowing through them, but instead use queues to separate the traffic into logical flows and maintain the transmission of the packets at the desired rate. Although simple queues only handle traffic in a single direction, two sets of queues can be used independently to manage traffic in two directions. Queuing is an essential part of IP networking and is used by almost all IP networking devices, such as routers and switches, not just by network performance systems.

[1] Sometimes referred to as the *bandwidth-delay product*.

2.2.1 Classless Queuing

Classless queuing systems are simple schemes that accept data and only reorder, delay or drop packets. The Token Bucket Filter (TBF) is an example of a classless queuing system that only passes packets arriving at a rate which is not exceeding some administratively set rate, with the possibility to allow short bursts in excess of this rate. The accumulation of tokens allows a short burst of overload data to be passed without loss, but any lasting overload will cause packets to be constantly delayed.

2.3 Classful Queuing

Classful queuing systems categorize or *filter* traffic into user-defined "classes" and maintain separate queues for each class.

2.3.1 Class-Based Queuing (CBQ)

CBQ is a simple queuing technique available in open-source form and found in low-end network performance systems. Data is prioritized by processing the queues at specific intervals or in order of priority and sent out according to a given time schedule.

Research has shown that CBQ is unsuitable for fine grained traffic control (see reference CBQ). In particular, CBQ is not well suited when the available bandwidth is dynamic, for example, when controlling traffic over links with variable transmission rates (such as Frame Relay) or on networks where alternative/redundant paths exist with varying rate capacity. CBQ may also result in unacceptable delays in servicing a class (despite its allocated share) due to the need to maintain a backlog of packets for the CBQ scheduler to work effectively.

2.3.2 Random Early Discard (RED) and Weighted Random Early Discard (WRED)

Random Early Discard (RED) is a queuing technique that detects congestion by computing the average queue size. RED systems keep the average queue size low while allowing occasional bursts of packets in the queue. During congestion, the probability that the system notifies a particular connection is roughly proportional to that connection's share of the bandwidth through the system. RED systems are designed to accompany a transport-layer congestion control protocol, such as TCP.

Weighted Random Early Discard (WRED) introduces a weighting based on the IP precedence to RED. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. For WRED to be successful traffic types must be predetermined and IP precedence tagging must be implemented in the edge and distribution layers of the network. WRED uses this precedence information to determine how it treats different types of traffic.

Both RED and WRED are limited to TCP traffic as they rely on the TCP rate control mechanism and consequently do not apply to UDP or non-IP network traffic. Bursty UDP and non-IP traffic adversely affect WRED systems, as these can overwhelm the rate control mechanism thereby leading to increased latencies.

2.3.3 Weighted Fair Queuing (WFQ)

Weighted Fair Queuing (WFQ) is a common queuing technique that allows several sessions to share the same link, unlike CBQ. WFQ is an approximation of an idealized fluid model called Generalized Processor Sharing (GPS). In GPS each session has a separate FIFO queue. At any given time the N active sessions (the ones with non-empty queues) are serviced simultaneously, each at a rate of $1/N^{th}$ of the link speed. GPS allows different sessions to have different service shares. GPS has several desirable properties. Since each session has its own queue, an ill-behaved session (that is sending more data than its fair share) will only punish itself and not other sessions. Further, GPS allows sessions to have different guaranteed bandwidths allocated to them.

Although WFQ is commonly found in router implementations, research has shown that WFQ does not differentiate traffic unless the output queue length is quite large (even under large traffic loads), which often results in unacceptable latencies (reference WFQ).

2.3.4 Priority Queuing (PQ)

Priority queuing (PQ) is designed to provide a relatively simple method of supporting differentiated service classes. In classic PQ, packets are first classified with each classification placed into different priority queues. Queues are typically classed in order of importance then packets are scheduled from the head of a given queue only if all queues of higher priority are empty. Within each of the priority queues, packets are scheduled in FIFO order.

One major deficiency of PQ is that lower-priority queues may not be serviced when there is high link utilization due to high-priority traffic.

2.3.5 Low Latency Queuing (LLQ)

Low Latency Queuing is a combination of Priority Queuing and Weighted Fair Queuing. Typically, there is one Priority Queue and three Weighted Fair Queues. LLC is an attempt to service real-time traffic in a time efficient manner. Real-time traffic is queued to the priority queue, and all other traffic is allocated to the weighed fair queues. The priority queue is serviced before any of the weighted fair queues, thus allowing real-time traffic to be processed as fast as the network elements allow.

The deficiency of LLQ is that there is no understanding of network topology and the impact of bandwidth variations upon the end-to-end traffic flows, nor is there any consideration to network wide latency. LLC is a simple, best effort queuing technology.

2.4 Advantages of Queuing

Queuing has the following advantages over TCP rate control:

- Scope
- Accuracy
- Responsiveness
- Efficiency
- Scalability
- Compatibility

2.4.1 Scope

TCP rate control systems, as the name suggests, are limited in scope to TCP traffic only. Non-TCP traffic, such as UDP and layer-2 traffic, has no rate control mechanism and does not adapt to the available capacity of the network. Queuing *must* be used for all non-TCP traffic, even if TCP rate control is used for TCP traffic.

2.4.2 Accuracy

TCP rate control depends on accurately predicting the round-trip time for each stream it is managing. This is quite difficult to do because round-trip times vary greatly. An incorrect prediction will result in:

- bandwidth under-utilization, if the predicted value is too low
- bandwidth over-utilization, if the predicted value is too high

These errors accumulate over time, often resulting in system-wide under or over-utilization of the network link being managed. In contrast, queuing systems know exactly how much bandwidth has been received and how much should be transmitted and can easily enforce configured rate limits. This provides maximum utilization of the network link.

2.4.3 Responsiveness

While TCP rate control may be effective for steady traffic, it does not handle changing traffic conditions well, since any rate change will not take effect until the current transfer has completed and the next request is initiated. Consequently, there is a delay of at least half the round-trip time from the instant that over-utilization is detected until the rate is actually reduced; by which time it may be too late to prevent queue saturation and packet losses. Similarly, there is a delay of at least half the round-trip time after congestion is reduced before the rate increases. TCP rate control is especially poor on networks with a large bandwidth-delay product, such as satellite links, especially for short-lived traffic flows and traffic bursts.

The following output shows the round-trip time for the popular web site *Google*:

--- www.google.com ping statistics ---

4 packets transmitted, 4 packets received,
0% packet loss round-trip min/avg/max = 335.8/336.7/338.1 ms

An average of 337ms, implies at that TCP rate, control will take on average 169ms to adjust, which is a long time in networking terms.

2.4.4 Efficiency

As noted above, TCP rate control responds to congestion by reducing the window size. For a given round-trip time, this is accomplished by reducing the packet size. This in turn increases the number of packets, not only because of the smaller packet sizes but because of the increased number of acknowledgement packets.

Because of the overhead associated with TCP packet headers, an increase in the number of smaller packets forces network devices to perform more work for the same amount of data throughput, which increases CPU load and latency. Further, a larger number of smaller packets also impacts "innocent bystanders" on the same subnet as the destination server.

For example, if the window size is reduced from, say, 20kb to 2kb, the device must perform roughly 10 times as much work to process the same amount of data. Alternatively, a non-rate control system can handle 10 times the load.

TCP rate control systems are also inefficient because they must monitor each TCP connection, modify the packets, and keep track of when acknowledgements need to be sent. For example, web servers typically have lots of relatively short-lived connections of around 10-100 packets. With rate control, you can only control each of these flows which are typically short lived and may never reach steady state. TCP rate control, by virtue of managing individual flows, does not lend it itself to efficiently managing web traffic as a whole as a *class* of traffic.

Queuing, on the other hand, never alters the number of packets, therefore does not reduce the effective capacity of network devices. Queuing also permits traffic to be managed efficiently as a class, not just per flow.

2.4.5 Scalability

Because of the above-mentioned inefficiencies, TCP rate control systems do not scale well. It is time consuming and computationally expensive for a TCP rate control system to monitor each TCP connection, modify the packets, and keep track of when acknowledgements need to be sent.

Queuing, in contrast, is quite efficient and

- does not reduce packet size and therefore does not reduce the effective capacity of router queues

- does not increase the number of packets and therefore does not increase the CPU load on routers and network hosts

Consequently, queuing systems can run on slower, less costly hardware.

2.4.6 Compatibility

In addition, changing and tampering with the TCP window size on the fly can cause unpredictable behavior on different TCP stacks. For example, certain network applications such as Internet Information Server (IIS), implement proprietary window sizing schemes and do not honor TCP rate control requests (see reference MS-TCP/IP).

2.5 Disadvantages of Queuing

While queuing is superior to TCP rate control overall, first generation queuing schemes have the following limitations.

2.5.1 Rate Sensitive

To differentiate traffic, i.e., make a sensible decision as to which packet to send next, there must be more than one packet buffered in the system. If not, first generation queuing systems will merely take in a packet and immediately forward that packet whenever the outgoing rate is faster than the incoming rate. This is known as work-conserving behavior. As a result, first generation queuing systems are ineffective under variable rate conditions that result in periods of non-congestion.

2.5.2 Increased Latency

To improve traffic differentiation, first generation queuing systems must increase the length of their queues to ensure that multiple packets are present at once. Unfortunately, increasing the length of the queues often introduces unbounded packet delays through the system.

2.5.3 Lack of Consistency Checking

First generation queuing implementations, such as those found in commercial routers, often combine multiple queuing algorithms, for example, RED, WRED, CBQ, WFQ, etc. Unfortunately, it is quite possible to inadvertently program such systems so as to yield unexpected or inconsistent behavior.

2.5.4 Per-hop Control

First generation queuing systems only control traffic per hop. There is no coordination between upstream and downstream queues.

Note, that this is also a problem with TCP rate control systems, not just queuing systems.

3 Second Generation Network Performance

3.1 Queuing Without the Disadvantages

Hierarchical Credit-based Queuing (HCQ) is a powerful new queuing method that achieves the efficiency and flexibility of first generation queuing systems, without the disadvantages.

3.1.1 Rate-independent Traffic Differentiation

HCQ dynamically adapts to the incoming rate or the outgoing rate while ensuring that the total number of buffered packets remains constant, even in the face of dynamically varying rates. This is extremely valuable for Frame Relay and other burst-on-demand architectures. Without such technology two scenarios occur: either you are not able to take full advantage of the burstable bandwidth, or you are not able to guarantee performance up to the maximum limit.

3.1.2 Latency Management

HCQ ensures that the number of packets being buffered never grows unacceptably large, nor introduces unacceptable packet delays. In doing so it also enables the explicit and precise setting of interpacket queuing delays, which is essential for real-time traffic.

Guaranteeing the maximum latency of a channel is of course absolutely critical for real-time traffic flow, such as VoIP or video.

3.1.3 Integrated Control and Consistency Checking

HCQ allows the integrated control over all of the elements of network performance, i.e. maximum rate, minimum rate, latency and prioritization. As with other queuing methods, HCQ first classifies or filters traffic into a logical flow of packets, called *channels*. Each HCQ channel can define the following service levels:

- A minimum rate the channel will receive under congestion
- A maximum rate the channel can use
- A maximum latency (delay) between a packet entering the channel and successive packets, under their minimum bandwidth
- Prioritization

A collection of channels and their associated service level is called a *policy*. Before an HCQ policy can be deployed, the requested service levels are checked to ensure that they can be satisfied and therefore admitted into the system[2].

[2] Equator One NP includes a patented consistency checker for this purpose.

HCQ is designed to be used with a unique consistency checker to ensure that only achievable and consistent policies are ever deployed onto a network. This is possible with HCQ because it is a totally integrated algorithm for treating latency, and minimum and maximum bandwidth requirements of a given link, rather than a string of independent algorithms cobbled together.

3.1.4 System-wide Control

HCQ is based on a unique credit-based algorithm, capable of sharing credits system wide between hierarchies of HCQ systems, as well as hierarchically between queues within a given HCQ system. This provides a large number of possibilities not available to first generation techniques, such as decentralized policy management and application within application control.

3.1.5 Dynamic Adaptation

HCQ does not create hard reservations which do not allow other applications and traffic flows to borrow bandwidth or capacity when not in use. Rather, HCQ provides maximum use of available bandwidth and latency by enabling applications to instantaneously take advantage of spare capacity. Such dynamic adaptation can provide up to 60% more performance per bandwidth when compared to traditional provisioning techniques.

3.2 Credit-based Queuing (CQ)

In a credit-based queuing system, each distinct traffic flow, or *channel*, is assigned a number of *credits*, based on the requested service level criteria. The system then ensures that channels only "spend" credits within their allocation. The system periodically replenishes each channel's credits. If the network becomes congested, the channel or system receive fewer credits or they're replenished less often. This forces the transmission of data to slow down. When the congestion clears, the number of credits is either increased or replenished faster. This forces the channel or system to transmit faster. A smoothly running credit-based system eliminates undesirable underflow and overflow in upstream and downstream queues.

A major advantage of credit-based queuing over other mechanisms is that it is computationally efficient and scalable to large bandwidths and applications. This translates to less expensive hardware support and longer in-field life for a given deployment. Whilst straightforward in concept a number of technical obstacles have been solved, problems that have prevented until now the widespread use of credit-based queuing implementation in IP networks[3].

[3] It is interesting to note that the Asynchronous Transfer Mode (ATM) Forum considered using a credit-based flow control mechanism for the ATM specification but abandoned the idea due to concerns that it would be too difficult to implement.

3.3 Hierarchical Credit-based Queuing (HCQ)

HCQ extends the credit-based queuing concept to define different *types* of credits which form a precedence *hierarchy* as follows:

Type of credit	**Purpose**	**Precedence**
Jitter real-time credits	Guarantee jitter	Highest
Latency real-time credits	Guarantee latency	High
Minimum rate credits	Guarantee minimum rate	Medium
Maximum rate credits	Guarantee maximum rate	Low
No credits (restricted)	Restrict (suppress) traffic	Lowest

Credits at the top of the hierarchy take precedence over credits lower in the hierarchy. For example, *latency real-time credits*, i.e., credits that guarantee an inter-packet latency, take precedence over *minimum rate credits,* but are lower in precedence than *jitter real-time credits*.

3.3.1 Rate Credits

Rate credits are used to ensure that the minimum and maximum rates can be achieved. The HCQ system assigns both types of rate credits to each channel. The *minimum rate credits* ensure that the channel receives its minimum rate during times of congestion. The *maximum rate credits* determine how bandwidth is shared among competing channels during times of non-global congestion.

The system also assigns master credits to a so-called master controller, equal to the sum of the minimum rate credits of each channel. These master credits are used to update the state of the system to ensure that the service levels are met.

HCQ also dynamically adapts to rate changes, and, for example, can detect when bandwidth is available beyond the committed access rate (CAR) on the network link, and permits channels to utilize that available bandwidth subject to their maximum rate credits. This dynamic rate adaptation is particularly useful on Frame Relay and other links that support variable rates. In contrast, TCP rate control and other queuing systems can only adapt to rate changes slowly, if at all.

3.3.2 Real-time Credits

Real-time credits are used to determine that a channel can be admitted into the system and scheduled with guaranteed latency, subject to the channel being at or under its minimum rate. Based on the specified rate at which packets may leave the system and the desired maximum delay, the system issues itself real-time credits equal to the rate multiplied by the maximum delay. Each time a channel requires admittance into the system, it will use a number of credits equal to the Maximum Transmission Unit (MTU) of the channel (the maximum sized packet it can send) multiplied by the maximum delay of the system, divided by the maximum delay this channel requires. If a channel requires more credits than are available, it is not allowed to enter the system. Once admitted into the system, packets are scheduled such that they all meet their latency guarantee. The real-time credit mechanism also ensures that real-time channels do not hog the bandwidth of other channels.

3.3.3 The Hierarchy in HCQ

The per-session approach inherent in TCP rate control prevents the management of hierarchical network, aggregate traffic. For example, a TCP rate control system that places a 1 Mbps limit on bandwidth to a department cannot, in addition, restrict each member of the department to 64kpbs. In contrast, HCQ supports arbitrary hierarchies of channels. Further, HCQ systems themselves can be arranged to form hierarchies and credits can be dynamically shared between such systems.

3.3.4 A Freeway Analogy

What whitepaper would be complete without an analogy?

A first generation queuing system can be compared to a freeway system with ramp meters at each on-ramp but limited to a single lane at each on-ramp. While such ramp meters smoothly regulate the flow of traffic onto the freeway, they do not provide express lanes for important traffic to access the freeway, nor express lanes once travelling along the freeway. Nor, is their any coordination between ramp meters at different on-ramps.

In contrast, a TCP rate control system can be compared to installing a speed delimiter on each vehicle. The system works so long as the speed limit is known ahead of time and no one tampers with the delimiter. In practice, however, both assumptions are often violated.

HCQ can be compared to a smart freeway system with multi-lane ramp meters and express toll lanes[4]. Important traffic, that is traffic with real-time credits, is guaranteed fast access onto the freeway at each ramp meter, and guaranteed an express lane once on the freeway. Further, ramp meters at different on-ramps are coordinated to maximize global throughput and minimize end-to-end delays and collisions.

[4] In highway engineering, both ramp meters and express lanes are well established traffic management techniques designed both to keep traffic flowing on the freeways and to reduce end-to-end delays and collisions.

4 The Future of Network Performance

Network convergence is fast becoming a reality, and while no one can predict what kind of traffic will dominate future IP networks, the following trends are already clear:

- Short-lived traffic flows will be common, driven by increasing network transactions.
- Real-time traffic, such as VoIP and video, will increase dramatically in volume.
- Latency management will become more critical than bandwidth management, as transmission rates increase.

Network managers will therefore increasingly come to rely on sophisticated queuing techniques, such as HCQ, to manage these traffic demands.

HCQ is a fundamental building block needed to optimize IP traffic for performance and cost in today's distributed networks. Its future is more valuable still, as enterprises and service providers tire of the burden of the limitations of first generation traffic control techniques to meet the necessary performance and cost requirements. HCQ offers an answer for enterprises and service providers wanting not only immediate relief from network congestion but also a path forward for true end-to-end network performance.

For end-to-end network performance to be a reality there will be a shift in the value of the network from the commodity network elements, such as the switches and routers, to the intelligence needed to deliver applications and services seamlessly to enterprises and end users. This intelligence is required to not only to deliver uninterrupted performance to new and innovative applications over diverse networks, devices and topologies, but also to extract the maximum return on investment from infrastructure.

5 Common Questions about HCQ

5.1 Doesn't queuing introduce delays?

First generation queuing algorithms, such as WFQ, do indeed introduce delays, since long queues are required for effective traffic control. HCQ, on the other hand, is effective with very short queues, and therefore does not introduce delays. Further, HCQ dispenses with output queues all together and uses only input queues for filtering purposes.

5.2 I have a router that supports queuing? Do I need HCQ?

Yes. Commercial routers only ship with first generation queuing algorithms, such as WFQ and WRED, which result in poor traffic differentiation and increased latencies.

5.3 I have a router that supports RTP? Do I need HCQ?

Yes. RTP provides a timestamp and a sequence number for each UDP packet and ensures that packets are received in order. Unlike HCQ, however, RTP does not do admission control nor it does it guarantee inter-packet latencies. Also, during times of congestion there is no straightforward way to control how non-RTP traffic is handled and everything else gets dropped by default.

5.4 I have Gigahertz switches. Do I need HCQ?

Yes, for two reasons. Firstly, your WAN will invariably be slower than the LAN, therefore requiring bandwidth management at the junction of the two. Secondly, there is a need for latency management, as latencies dominate bandwidth at very high transmission rates.

5.5 Is HCQ patented?

Yes, HCQ is covered by the following five pending patents in all major countries:

- Credit-based Queuing for Traffic Shaping
- Method for real-time network traffic admission and scheduling
- Rate Limiting through Dynamic Auto Correction
- QoS Consistency Checking
- Dynamic Fair Bandwidth Shaping

For More Information

Contact Equator One for related inquiries.

Singapore

Equator One Pte Ltd

04-01/03 OAC Building

260 Tanjong Pagar Road

Singapore 088542

Phone +65 63960110

Fax +65 63232137

World Wide Web

www.equatorone.com

References

TCP/IP

"TCP/IP Illustrated", Volume 1

By Richard Stevens

Addison-Wesley Publishing Company

TCP-Rate-Control

"The Adverse Impact of the TCP Congestion-Control Mechanism in Heterogeneous Computing Systems"

By Wu-Chung Feng, et al.

Department of Electrical and Computer Engineering, University of Wisconsin-Madison

Proceedings of INFOCOMM 2001

MS-TCP/IP

"TCP/IP and NBT Configuration Parameters for Windows 2000 or Windows NT"

Microsoft Knowledge Base Article 120642

http://support.microsoft.com/default.aspx?scid=kb;EN-US;q120642

CBQ

"Operational and Performance Issues of a CBQ router"

By Fulvio Risso and Panos Gevros

Department of Computer Science, University College London

ACM SIGCOMM *Computer Communication Review*, volume 29, number 5, Oct 1999. ISSN # 0146-4833.

RED

"Random Early Detection systems for Congestion Avoidance"

By Sally Floyd and Van Jacobson

IEEE/ACM Transactions on Networking (TON), v.1 n.4, p.397-413, Aug. 1993

WFQ

"How Unfair can Weighted Fair Queuing be?"

By Goncalo Quadros, et al.

Centro de Informatica e Sistemas da Univeridada de Coimbra (CISUC)

Proceedings of ISCC 2000

