Dyband

IP Traffic Management: TCP Rate Control vs. Queuing

There are a number of differences between IP traffic management systems, such as hardware vs. software approach, monitoring and reporting capabilities, and support for dynamically-addressed environments, to name a few. The most fundamental difference, however, is in the approach to controlling transfer rates. The various approaches fall into two basic categories: TCP rate control and queuing. This document describes the differences between the two approaches and shows how queuing is the more effective of the two.

Overview

It is important to begin by reviewing the basic characteristics of TCP, TCP rate control, and queuing.

TCP Characteristics

TCP is a reliable connection-oriented delivery service using IP to transport messages between machines. It uses:

- An acknowledgement scheme to determine if messages have arrived (i.e., that no packets have been lost)
- A sliding window scheme to control the rate at which information flows between machines; when there is packet loss, the requesting station reduces its window size

TCP Rate Control

A traffic management system that uses TCP rate controls manages TCP traffic by predicting the round-trip time (RTT) for each stream it is managing and, on the basis of this prediction, changing the TCP semantics mid-stream. The semantics are changed by:

1. Intercepting acknowledgement packets sent by the receiver to the sender and holding them for a period of time (based on the RTT predictions)
2. Modifying the window size advertised in the header of the acknowledgement packets, thus affecting the size of the packets that will be transmitted by the sender

It should be noted that this approach relies upon intrusively modifying packets and spoofing the natural order of TCP rate and window negotiations, with repercussions not only for the network managed by the TCP rate control system but for outside networks as well.

Queuing

IP traffic management systems that use queuing do not modify the packets flowing through them. Instead, they use queues to finely separate the different traffic classifications they are managing and pace the delivery of the packets to the desired speed. The effect from the perspective of the end-stations is that the link between them appears to have precisely the desired speed.

Some queuing system are simple, handling traffic in only a single direction, while others (such as Dyband) manage two sets of queues independently, one for each traffic direction.

It is important to note that every IP packet that traverses the Internet, whether transported over Ethernet, ATM, PPP, or SONET, is queued at each hop along the way. Thus queuing is an integral part of the Internet system, allowing routers to prioritize packet streams as they flow across the Internet.

Limitations of TCP Rate Control Systems

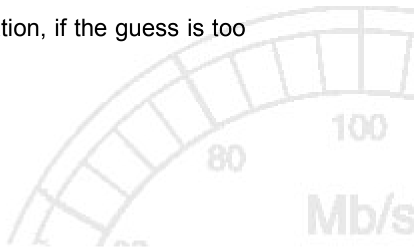
TCP rate control has a number of inherent limitations:

- Inaccuracy
- Lack of prioritization
- Slow response to changing conditions
- Packet loss
- Inefficiency
- Inflexibility
- Poor scalability

Inaccuracy

One of the most significant flaws of TCP rate control is its dependence on accurately predicting the round-trip latency for each stream it is managing. This is particularly difficult because round-trip times can vary greatly. An incorrect guess can result in:

- Bandwidth under-utilization, if the guess is too low
- Bandwidth over-utilization, if the guess is too high



The accumulation of these errors can lead to system-wide under/over-utilization of the aggregate link being managed.

In contrast, a queuing system such as Dyband knows exactly how much bandwidth has been received and how much should be transmitted and will enforce configured rate limits for each. This provides maximum utilization of the aggregate link being managed.

Lack of Prioritization

When a network is at maximum utilization, prioritization of traffic is essential for ensuring that the most important packets are the first to be transmitted and the last to be dropped. With TCP rate control, if over-utilization occurs, packets will be dropped randomly rather than by priority. In contrast, queuing is well-designed to allow the prioritization of traffic.

Slow Response to Changing Conditions

If congestion is detected during a session, TCP rate control will reduce transfer rates, but the reduction will not begin until the current transfer has completed and the next request is initiated. Consequently, there is a delay of at least half the round-trip latency from the time that over-utilization is detected and the rate is actually reduced, by which time it may be too late to prevent queue saturation and packet loss at the upstream router. Likewise, there is a delay of at least half the round-trip latency after congestion is reduced before normal rates can resume.

In contrast, Dyband evaluates network traffic every 10 ms and, if congestion is detected, will immediately delay the completion of the current request, causing a corresponding delay in the issuance of the next request and thus reducing packet loss at the upstream router. In addition, when congestion ends, normal rates can resume without delay.

Packet Loss

Packet loss resulting from a slow response to congestion has just been discussed. The potential for packet loss is also increased with TCP rate control because of its manipulation of window size. In order to manage traffic on heavily utilized links, TCP rate control must significantly reduce the window size. This in turn causes packet sizes to be decreased and the number of packets to increase, not only because of the smaller packet payloads but because of the increased number of acknowledgement packets. Routers are rated to process a maximum number of packets per second, and an increase in the number of packets may force them to drop packets. Queuing, by not altering the number of packets, does not add to the risk of packet loss.

Furthermore, queuing, by ensuring that downstream routers are never fed data at a rate that exceeds their link capacity, removes the risk of downstream router packet loss.

Inefficiency

TCP rate control can cause serious inefficiencies in the handling of network traffic, as follows:

1. Network inefficiencies

An increase in the number of smaller packets may not only result in packet loss, as described above; it will also reduce the overall efficiency of the network, affecting the performance of network routers and hosting servers.

- Router queues are generally designed for a fixed number of fixed-size packets. When undersized packets (smaller than the MTU) are generated as a result of TCP window size reduction, the effective capacity of the router queues is greatly reduced.

- Because of the overhead associated with TCP packet headers, an increase in the number of smaller packets forces the routers to perform more work for the same amount of data throughput. This increases router CPU load and latency. Doubling the number of packets doubles the router CPU load and reduces network connection efficiency by at least 3%.

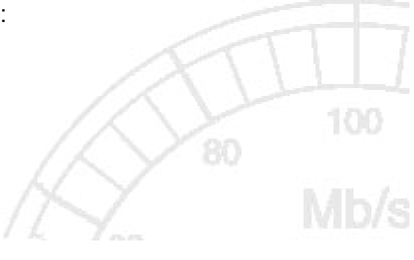
- Similarly, a hosting server must perform the same amount of work handling a request with a small window size as a large one. This means that if the window size is reduced from, say, 12,000 to 500 bytes, the server must perform 24 times as much work to serve the same amount of data. This translates into being able to serve 24 times as many customers with the same server platform if TCP rate controls are *not* used.

  Note, too, that smaller window sizes impact not only internal servers but "innocent bystanders", the servers at the other end of the network connections.

The size of packets that would be required by TCP rate control is determined by the following equation:

$$\text{packet\_size\_bytes} = (\text{rate\_bps} \cdot \text{latency\_sec}) / 8$$

This equation can be used to compute the rate below which a TCP rate control system would need to generate undersized packets (smaller than the standard MTU of 1518 bytes), given a specified round-trip latency:



$$rate\_bps = (1518 \cdot 8) / latency\_sec$$

For example, enforcing a 64 Kbps rate when round-trip latencies average 120 ms requires that the packets be held to 960 bytes, well below the most efficient size of 1518 bytes. The problem is aggravated in higher performance networks, where loading is generally higher and latencies are lower. Enforcing the same 64 Kbps rate on a network with a latency of only 45 ms requires that the packets be held to 360 bytes.

2. Inefficiencies in the TCP rate control system itself

   Users of TCP rate control systems frequently complain that they are unable to reach their maximum rated speed. This is due not only to the extra load on routers and servers described above but also to the CPU performance needed by the TCP rate control system itself. It is a time-consuming process 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:

- It does not reduce packet size and therefore does not reduce the effective capacity of router queues.
- It does not increase the number of packets and therefore does not increase the CPU load on routers and hosting servers.
- The simplicity of the queuing approach allows it to run on slower (and therefore less costly) machines.

Inflexibility

There are a number of contexts in which TCP rate control systems do not perform well or cannot perform at all:

- Where latency varies and is therefore difficult to predict, they are unable to optimize bandwidth utilization.
- Where latency is high (such as satellite systems), they cannot respond quickly to periods of high utilization.
- When handling *any* non-TCP traffic (OSI layer 2 traffic, UDP, ICMP, certain audio and video streams, etc.), they must make use of queuing since there are no acknowledgements to pace and no window sizes to negotiate. In other words, systems using TCP rate controls do use queues for all non-TCP traffic.
- For low-bandwidth connections, TCP rate control has to reduce packet sizes below the MTU, which results in under-utilization of the available bandwidth.

The fact that TCP rate control manages traffic on a per-session basis results in further limitations:

- Difficulty managing traffic by host
  When a host conducts several simultaneous sessions, TCP rate control systems apply the configured controls to each session, rather than to the host's total current usage. Only by using special constructs can a TCP rate control system overcome this handicap, but the number of hosts that can be managed in this way is small compared to Dyband's 50,000-object capability.
- Inability to manage hierarchical networks.
  The per-session approach also prevents the management of multi-level, aggregate traffic. For example, a TCP rate control system that places a 2 Mbps limit on bandwidth to a department cannot, in addition, restrict each member of the department to 500 Kbps. In contrast, Dyband can control up to 32 levels of aggregate traffic.
- Delayed response to configuration changes
  In a TCP rate control system, changes made by the user to rate limits do not take effect until a new session is established.

None of these limitations apply to queuing systems such as Dyband, which is able to

- Handle IP traffic of *any* type, including UDP
- Efficiently control both low- and high-bandwidth connections
- Operate well in environments with low, high, and variable latencies
- Manage traffic by host
- Manage multiple levels of aggregate traffic
- Respond immediately to changes in configured rate limits and priorities

Poor Scalability

Because of the inefficiencies of TCP rate control processing, it will be difficult for these systems to scale well. Queuing systems were able to support 100 Mbps traffic loads long before TCP rate control systems were able to do so and in the future will be able to scale more easily to support higher bandwidth demands.

Conclusion

TCP rate control has a number of limitations, all of which are overcome by queuing. The table below summarizes the comparisons that have been made in this paper..

	TCP Rate Control	**Queuing (Dyband)**
Accuracy	Controls based on guesswork (predictions of round-trip latency)	Controls based on precise data (bandwidth demand vs. capacity for both upstream and downstream traffic)
Prioritization	No prioritization during over-utilization	Strict, dynamic prioritization
Response Time	Responds after a delay of half the round-trip latency	Responds in the next 10-ms shaping cycle
Packet Loss	Increases packet loss because of: ➢ Slow response time ➢ Increase in number of packets during periods of heavy utilization	Minimizes packet loss due to: ➢ Rapid response time ➢ No increase in number of packets
Efficiency	➢ Smaller packet sizes and increased number of packets lowers efficiency of network ➢ Processing of packets is time-consuming, requiring high-end platform	➢ Network not burdened with additional packets ➢ Processing of packets is rapid, requiring only standard platform
Flexibility	➢ Cannot handle non-TCP traffic ➢ Cannot optimize bandwidth utilization when round-trip latency varies ➢ Cannot respond quickly in high-latency networks ➢ Under-utilizes bandwidth in low-bandwidth connections ➢ Has difficulty managing traffic by host ➢ Unable to manage multiple levels of aggregate traffic ➢ Unable to implement changes to configured rate limits until new session established	➢ Handles all IP traffic, including UDP ➢ Operates well in variable-latency networks ➢ Operates well in high-latency networks ➢ Optimizes bandwidth utilization in both low- and high-bandwidth connections ➢ Manages traffic of individual hosts ➢ Manages multi-level aggregate traffic ➢ Implements changes to configured rate limits immediately
Scalability	CPU-intensive approach makes scaling difficult	Efficient approach makes scaling straightforward

For further information on Dyband, and how it can benefit your firm, contact us at
sales@dyband.com
or visit us at
www.dyband.com

Dyband Corporation
215 Stafford Road West, Unit 103
Ottawa, Ontario K2H 9C1
Canada
(613) 820-3677