TCP and Quality of Service

Lasse Seppänen
Lasse.Seppanen@evitech.fi
Ruutikatu 4 B 19
02600 Espoo
050-5822122

1. Abstract

The Internet has historically offered a single level of service, that of "best effort," where all data packets are treated with equity in the network. The Internet itself does not offer a single level of service quality, and some areas of the network exhibit high levels of congestion and consequently poor quality, while other areas display consistent levels of high quality service.

This is being solved by defining methods of "Quality of Service" (QoS) to TCP. In this document ideas of the concepts for QoS are discussed.

2. Introduction

The material of this presentation is gathered from largely from Paul Ferguson, Cisco Systems Inc, the co-writer of Quality of Service with Geoff Huston. Cisco web pages have been very useful providing easy-to-swallow information. Another basis has been High-Speed Networks by William Stallings.

In order to understand this document some background information of TCP/IP may turn up useful.

3. TCP/IP Principles

In the global Internet TCP/IP protocol suite is the common bearer service. It provides an end-to-end “best effort” service. [6]

3.1. TCP Header Format

TCP uses a single type of protocol unit, TCP segment. The header contains 20 octets. [2]

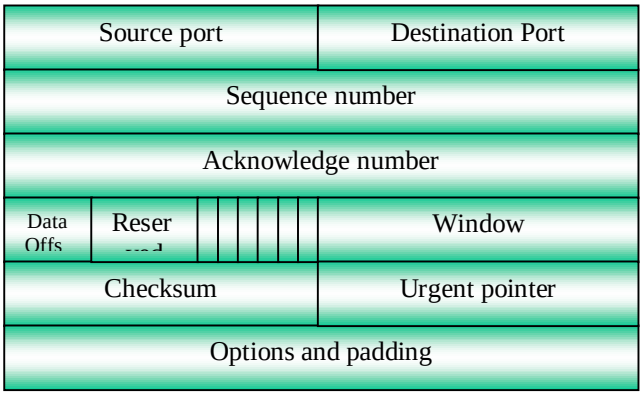

Figure 1: TCP Header

The sequence number points to the first data octet in this segment, unless SYN flag is set. Otherwise it is the initial sequence number.

Acknowledgement number contains the sequence number of the next data octet that TCP entity expects to receive.

Flags are

- URG: Urgent pointer field significant
- ACK: Acknowledgement field significant
- PSH: Push function
- RST: Reset the connection
- FIN: No more data from sender

Window contains the flow control credit allocation. It contains the number of data octets indicated in the acknowledgement field that the sender is willing to accept.

Urgent pointer points to the last octet in a sequence of urgent data to allow the receiver to know how much urgent data is coming.

Push and urgent flags implement two TCP services:

- **Data stream push**. Normally TCP decides when enough data to be sent has accumulated and sends. By push function the TCP user can force transmission anytime.
- **Urgent data signaling** provides means of informing the destination that important data is coming.

A TCP user (an application like FTP) issues a SEND command to pass data to TCP, which places is to a send buffer. TCP may gather data for some time and send it out when convenient, with the exception of possible push request.

At the other end similar actions take place: storing incoming data to a buffer and delivering it when convenient, with the natural exception of possible push request.

Timestamp field provides TCP means to monitor the roundtrip time of the connection.

3.2. TCP Flow Control

TCP has a sliding-window mechanism for flow control. Each individual octet of data is considered to have a sequence number. When TCP sends a segment, it includes the sequence number of the first octet in the segment data field. The receiver acknowledges the incoming segment indicating the number of octets that may be sent and the octets that are received. As the connection is established during the first segments the transmission flow is stabilized so that the ring-trip delay (time between sending a segment and receiving the acknowledgement to it) is known to the sender and segments are sent accordingly.

3.3. Throughput of TCP

The throughput of TCP depends on the sliding window size, propagation delay and data rate. Sliding window size is the distance between the last acknowledged and last sent octet.

W = TCP window size in octets
R = Data rate in bps
D = Propagation delay in seconds

Normalized throughput S

$$S = 1, \quad W > RD/4$$
$$S = 4W/RD, \quad W < RD/4$$

S = 1 means the maximum throughput.

Many TCP connections are multiplexed over the same network interface leaving for each connection only a part of the capacity. This reduces R. Since the connections may be long D is the sum of each delay of the routers on the way. A router delay may be long especially in the case of congestion. In case of retransmission the throughput is reduced.

3.4. TCP Congestion Control

The internet routing algorithms are able to handle congestion with unbalanced loads and brief surges in traffic. In the end the only solution is limiting the load in the network. This is the idea of the congestion control mechanisms.

It is difficult to control congestion in TCP/IP networks because

- IP is a connectionless and stateless protocol that has no means to indicate or control congestion
- TCP has only end-to-end flow control, but can't say anything of the network in between.
- TCP entities can't communicate to keep a certain level of total load.

TCP sliding-window flow and error control mechanisms relate to network congestions, though they are designed for end-to-end traffic. However, TCP cannot distinguish between loss due to packet corruption and loss due to congestion, and packet loss invokes the same congestion avoidance behavior response from the sender, causing the sender's transmit rates to be reduced by invoking congestion avoidance algorithms even though no congestion may have been experienced by the network. [6] Other techniques have been developed for congestion detection, avoidance and recovery.

3.5. TCP Flow and Congestion Control

Sliding-window flow control gives means to the receiver to pace the sender. By the rate of incoming ACKs the sender determines the rate of sent data. If there is a bottleneck in the network it is detected, but not it's location. This adjusting to bottlenecks is called self-clocking.

A number of techniques have been developed to improve congestion control, here are some of them.

Table 1: Implementation of TCP congestion control measures

Measure	*RFC 1122*	*TCP Tahoe*	*TCP Reno*
RTT Variance Estimation	x	x	x
Exponential RTO Backoff	x	x	x
Karn's algorithm	x	x	x
Slow Start	x	x	x
Dynamic Window Sizing on Congestion	x	x	x
Fast Retransmit		x	x
Fast Recovery			x

4. RED – Random Early Detection

By randomly dropping packets this mechanism avoids congestion collapse and global synchronization problem [1]. RED also attempts to create TCP congestion signals using duplicate ACK signaling, rather than through sender timeout. RED monitors the mean queue depth, and as the queue begins to fill, it begins to randomly select individual TCP flows from which to drop packets, in order to signal the receiver to slow down. The threshold at which RED begins to drop packets is generally configurable by the network administrator, like the rate at which drops occur in relation to how quickly the queue fills. The more it fills, the greater the number of flows selected, and the greater the number of packets dropped. This results in signaling a greater number of senders to slow down, thus resulting in a more manageable congestion avoidance.

The RED approach does not possess the same undesirable overhead characteristics as some non-FIFO (First In, First Out) queuing techniques (e.g. simple priority queuing, class based queuing, weighted fair queuing [4]). With RED, it is simply a matter of who gets into the queue in the first place - no packet reordering or queue management takes place. When packets are placed into the outbound queue, they are transmitted in the order in which they are queued.

RED chooses random flows from which to discard traffic in an effort to avoid global synchronization and congestion collapse, maintaining equity in which traffic actually is discarded. Fairness is good, but what is really needed for differentiated QoS structures is a tool that can induce unfairness - a tool that can allow the network administrator to predetermine what traffic is dropped first (or last, as the case may be) when RED starts to select flows from which to discard packets. Services can't be differentiated with fairness.

There are several proposals in the IETF which have suggested using the IP precedence subfield of the TOS (Type of Service) byte contained in the IP packet header to indicate the relative priority, or discard preference, of packets and to indicate how packets marked with these relative priorities should be treated within the network. As precedence is set or policed when traffic enters the network, a weighted congestion avoidance mechanism implemented in the core routers determines which traffic should be discarded first when congestion is anticipated due to queue-depth capacity. The higher the precedence indicated in a packet, the lower the probability of discard. The lower the precedence, the higher the probability of discard. When the congestion avoidance is not actively discarding packets, all traffic is forwarded with equity.

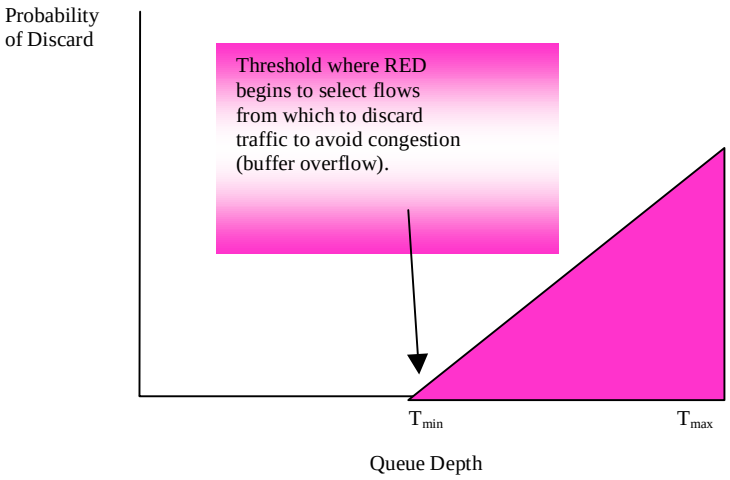

Figure 2: RED throwing away packets

Of course, for this type of operation to work properly, an intelligent congestion-control mechanism must be implemented on each router in the transit path. A least one currently deployed mechanism is available that provides an unfair, or weighted, behavior for RED. This deviation of RED yields the desired result for differentiated traffic discard in times of congestion and is called Weighted Random Early Detection (WRED) or enhanced RED (eRED).

4.1. *WRED – Weighted Random Early Detection*

WRED is useful on any output interface expected to have congestion. WRED is usually used in the core routers of a network, rather than the edge [3]. Edge routers assign IP precedences to packets as they enter the network (Figure 4). WRED uses these precedences to determine how it treats different types of traffic. Standard traffic may be dropped more frequently than premium traffic during periods of congestion.

Admission to network can be controlled actively or passively. Passive means the preset precedence by the end stations. Active means the routers actively changing the policies in the network. With threshold triggering the traffic can be marked so that the some traffic has lower probability of discard in times of congestion.

The precedence is defined in the Type of Service field (8 bits) by changing it to Precedence (3 bits) and Type of Service (4 bits) fields.

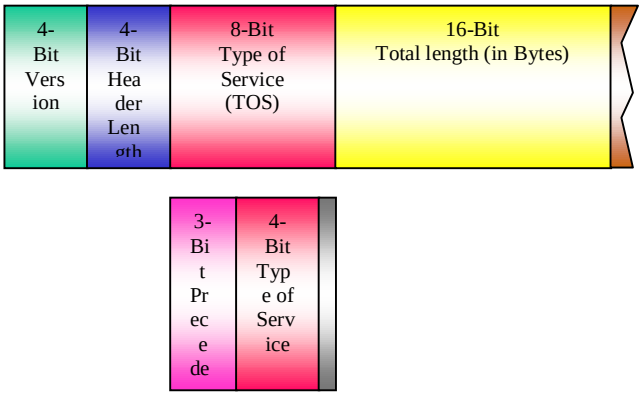

Figure 3: IP Header

4.2. *Precedence Levels with Token Buckets*

The precedence levels can be defined with token bucket control mechanism. Token bucket is a control mechanism that dictates when traffic can be transmitted based on the precedence of tokens in bucket. This provides means to control bursts. Tokens are specified in number of bytes. Thresholds can be set for the bursts of traffic. In case of a burst the token bucket mechanism slows down the traffic.

The precedence levels for different flows are in the next table. HTTP traffic is more important to have fast feedback than FTP and other traffic.

Table 2: Precedences

	Classifi-cation	*Thres-hold*	*Under threshold prece-dence*	*Over threshold prece-dence*
Token bucket 0	HTTP	30	6	5
Token bucket 1	FTP	10	4	0
Token bucket 2	Other traffic	10	3	0

In token bucket mechanism the incoming IP packets are queued for processing per flow basis. In the queue it is checked that their allowed amount / time is not exceeded. If not, then the packets are sent, if exceeded the behavior is not yet standardized. The common possibilities are best effort, discard or marking the packet so that it can be discarded in the future.

5. Conclusion

Because of the basic ideas of TCP/IP it seems to be hard to correct the congestion control problems. A lot is being done affecting many router vendors, who are responsible for implementing these techniques. Yet there is a lot to achieve.

6. References

[1] Ferguson, Paul; Huston, Geoff: Quality of Service, USA, 1998, ISBN 0-471-24358-2

[2] Stallings, William: High-Speed Networks, USA, 1998, ISBN 0-13-525965-7

[3] Cisco: Distributed WRED, http://www.cisco.com/univercd/cc/…ct/software/ios111/cc111/wred.html

[4] Cisco: Distributed Fair Queuing, http://www.cisco.com/univercd/cc/…ct/software/ios111/cc111/dwfq.html

[5] Cisco: Congestion Avoidance Overview, http://www.cisco.com/univercd/cc/…12cgcr/quos_c/qcpart3/qcconavd.html

[6] Ferguson, Paul; Huston, Geoff: Quality of Service in the Internet: Fact, Fiction, or Compromise, 1998 http://www.employees.org/~ferguson/inet_qos.htm

[7] Crowcroft, Jon: IP Traffic Management, http://www.cs.ucl.ac.uk/staff/J.Crowcroft/iptraff/index.html

Figure 4: IP Precedence to Indicate Drop Preference with Congestion Avoidance