Quality of Service

1. Overview

1.1 Introduction

Quality of Service (QoS) is where the data traffic on a network be it a LAN or a WAN, is subject to scrutiny and control. Quality of Service is primarily an IP concept and uses tools that have existed since the early days of IP plus newer tools and protocols that are designed to aid the provision of defined predictable data transfer characteristics. Quality of Service is also relevant at layer 2 within the Ethernet environment and also within the WAN technologies such as Frame Relay and ATM.

The drive for QoS has become very strong in recent years because there has been a growth of multimedia traffic such as voice and video, that mixes it with more traditional data traffic such as FTP, Telnet and SMB. Applications such as Voice over IP and Microsoft's Netmeeting have opened up the possibility of more interactive communications between users not just over the LAN but also over the WAN.

1.2 Sensitive Traffic

Much data traffic has been very tolerant of delays and packet drops, however voice traffic has different characteristics. Voice is low volume but is intolerant of jitter (delay) and packet loss. Video is also intolerant of jitter and packet loss, plus it has the added complication of being very bursty at times. This convergence of multimedia traffic with traditional data traffic is set to grow and therefore requires methods and tools to ensure that providers can deliver networks that give the users confidence to use these tools effectively.

1.3 Where QoS is Applied

You can often get away with introducing high bandwidth multimedia applications to a data network that has a well designed core and access topology, particularly if the core is based on Gigabit Ethernet technology and the access switches are not over-subscribed with respect to the backbone links. Even in this scenario there is the issue of speed mismatches between technologies such as Gigabit Ethernet and Fast Ethernet. This requires buffering and therefore leads to a requirement to queue and prioritise traffic. Layer 2 switches can mark frames using Class of Service (CoS) marking if end devices do not mark the frames themselves. These frames can then be assigned to hardware queues that exist on modern layer 2 switches. Often the concept of Trust Boundaries is used to describe how far out to the endpoints a switch will trust the marking of a frame.
The most significant network bottlenecks exist at the Remote Access points, the WAN access, Internet access and the servers. Once a packet is on the core of the network, it generally deserves to be there, providing that your security is intact. Many of the technologies involved in QoS are to do with how packets are dealt with as they enter and leave a network because merely adding more bandwidth at the edge is only a short term solution that just resolves capacity and perhaps some congestion problems. Adding bandwidth does not resolve jitter or add any traffic prioritisation features.

1.4 Reasons for QoS

Problems faced when attempting to provide QoS include:
A shortage of bandwidth because network links are oversubscribed.
Packets being lost due to congestion at bursty periods. Modern DSPs can fill in for between 20ms to 50ms of lost voice.
End-to-End delay made up by a number of components that occur in the following order:

1. Fixed **Switch Delay** - as packets are layer 2 switched from the call initiator.
2. Fixed **Encoding Delay** - coding PCM to G.729 (5 ms look ahead then 10ms per frame, with G.729 a Frame Relay packet would typically have 2 x 10ms samples), G.726 (0.125us), G.723 (30ms), G.728 (2.5ms) etc.
3. Fixed **Voice Activity Detection (VAD)** - around 5ms.
4. Fixed **Packetisation Delay** - i.e. packetising samples in G.711, G.726, G.729 etc.
5. Variable **Output Queuing Delay** - as voice packets enter an output queue and wait for the preceding frame (voice or data) to be played out.
6. Fixed **Serialisation Delay** - the time it takes to get the bits on to the circuit. The faster the circuit and the smaller the packet, the less this delay.
7. Fixed **Processing Delay** - the time it takes for the packets to be examined, routed, managed etc.
8. Variable **Network Queuing Delay** - due to clocking frames in and out of the different network switches.
9. Fixed **Network Propagation Delay** - normally assumed to be 6 microseconds/km or 10 microseconds/mile (G.114) as the packets
traverse the medium. Across the UK this is typically between 1ms and 3ms.

10. Variable **Input Queuing Delay** - as voice packets enter an input queue and wait for the preceding frame (voice or data) to be played out.

11. Fixed **Dejitter Buffer Delay** - can be up to 50ms.

12. Fixed **Switch Delay** - as packets are layer 2 switched to the endpoint.

13. Fixed **Decoding Delay** - decoding the compressed packets is trivial compared to encoding and so takes less time.

The **Delay Budget** is made up of the sum of the delays. In the UK, the total one-way delay budget is typically less than 100ms.

Given two time sensitive packets e.g. voice packets, the End-to-End delay may be different for each of those packets. The difference between the delays i.e. variable delay, is known as **Jitter**. Both Delay and Jitter have a significant impact on Voice and Video traffic, jitter often being more of a problem than delay. In addition, both technologies can cope with some packet loss, you don't for instance, wish to retransmit a part of the conversation, instead it is better just to lose it or smooth it over. Where video differs from voice is in its bandwidth requirements. Video is bandwidth hungry and bursty compared to voice which is steady and low in its bandwidth consumption.

Voice requires steady access where packets are evenly spaced (without Jitter) i.e. an Isochronous data flow in the same way that SDH works where the cells that you want can be demuxed without the flow being stopped. Traditional data networks do not operate this way, we have to make a data network appear to operate in a way similar to SDH. Some hardware manufacturers provide **Dejitter Buffers** that buffer packets as they are received and then play them out as a steady stream thereby eliminating the variable delay. The overall delay is increased slightly as a result of implementing a de-jitter buffer. Packets that arrive so late that they fall outside of the de-jitter buffer are discarded. DSPs may be implemented within the equipment that perform an intelligent interpolation of the missing sound. This minimises the distortion caused by the dropped packets.

**1.5 Functions of QoS**

QoS needs to enable:
Predictable Response times
Management of delay sensitive applications
Management of jitter sensitive applications
Control of packet loss when congestion occurs during a burst (note that continuous congestion means that the link is over-subscribed)
The setting of traffic priorities.
Dedication of bandwidth on a per application basis.
The avoidance of congestion
The management of congestion when it occurs, note that this is different from trying to avoid congestion, sometimes it is not possible to avoid congestion. QoS can be thought of as operating on one or more of three levels:

- **Best Effort**
- **Differentiated Service** - can deal with differing levels of QoS on a packet by packet basis
- **Integrated Service** - requested level of service by an application to a network. Data is only sent after confirmation that the service level is guaranteed to be available.

The order of events with respect to QoS can be described as follows:
Mark and Classify packets according to policies and the behaviour of the traffic. This is carried out with technologies such as IP Precedence and DSCP and is most effective when carried out as far to the edge of the network as possible, even at the device itself (e.g. a VoIP phone).
Congestion management by prioritising traffic based on the marks using queuing technologies that can respond to traffic classes.
Avoid congestion by dropping packets that are not a high priority using technologies such as Random Early Detection where low priority packets are weeded out of a queue so as to prevent indiscriminant 'Tail Drop' of packets.
Policing traffic so that bad packets are kept at bay i.e. limit the ingress or egress traffic depending on the class/markings of the packets. Also perform traffic shaping to maximise the use of WAN bandwidth by specifying peak and average traffic rates.
Fragment and compress packets to maximise the use of bandwidth.

### 2. Marking and Classifying

#### 2.1 Layer 2 Class of Service (CoS)

As detailed in **802.1p**, Layer 2 Class of Service can be provided within the TCI field of the Ethernet frame. The 3 bits give 8 different classes which have the values assigned as follows:

- **000** (0) - Routine
- **001** (1) - Priority
- **010** (2) - Immediate
- **011** (3) - Flash
- **100** (4) - Flash Override
- **101** (5) - Critical
These map to the IP Precedence values in the TOS field of the IP datagram. Layer 2 switches can have two or maybe four queues per port which can be used when there is congestion, to put frames with particular CoS values into the appropriate queues and perform a **Weighted Round Robin (WRR)** approach to servicing these queues.

As illustrated in **ISL**, Cisco’s ISL also has 3 bits allocated to priority values.

### 2.2 IP Precedence and Differentiated Services (DiffServ)

DiffServ is concerned with classifying packets as they enter the local network. This classification then applies to **Flow** of traffic where a Flow is defined by 5 elements; Source IP address, Destination IP, Source port, Destination port and the transport protocol. A flow that has been classified or marked can then be acted upon by other QoS mechanisms. Multiple flows can therefore be dealt with in a multitude of ways depending on the requirements of each flow. Packets are first Classified according to their current DSCP. Then they are separated into queues where one queue may be routed via a marking mechanism and another queue may be examined more closely. After further examination additional packets may be sent for marking or be sent direct to the shaping/dropping mechanisms where all packets end up before leaving the interface.

The IP header has a field called the **Type of Service (TOS)** that sits between the **Header Length** field and the **Total Length** field. (Refer to **IP Datagram** for a view of the Type Of Service field in the IP header.) Traditionally, **IP Precedence** has used the first three bits of the TOS field to give 8 possible precedence values.

- 000 (0) - Routine
- 001 (1) - Priority
- 010 (2) - Immediate
- 011 (3) - Flash
- 100 (4) - Flash Override
- 101 (5) - Critical
- 110 (6) - Internetwork Control
- 111 (7) - Network Control

DiffServ introduces the concept of the **DiffServ Code Point (DSCP)** that uses the first 6 bits of the TOS field thereby giving $2^6 = 64$ different values. **RFC 2474**
describes the Differentiated Services (DS) field and the DiffServ Code Point (DSCP).

With DiffServ each router handles each packet differently. The concept of **Per-Hop Forwarding Behaviour (PHB)** is introduced where classes are developed such as Business, Telecommuter, Residential etc. that can be offered by an ISP as different levels of service. A Per-Hop Behaviour is effectively a way of forwarding a particular flow or group of flows (**Behaviour Aggregate**) of traffic on a DiffServ node. A flow, or flows, of packets marked with a particular DSCP in the DS field will be subject to a particular method of forwarding and rules as encapsulated in the **Behaviour Aggregate**. This Aggregate has three elements to it (or three colours) which determine whether the router interface 1) Drops the datagram, 2) Sends the datagram or 3) reclassifies it. This three-colour marker is detailed in **RFC 2697**.

For instance 5 flows can be treated as a Behaviour Aggregate so they are treated similarly as a group in most respects. Each flow is then distinguished by an additional **Drop Probability** and **Forwarding Behaviour**. Be aware that as the Drop Preference value increases, so the probability of being dropped increases!

The following table illustrates the DSCP values:

<table>
<thead>
<tr>
<th>Per Hop Behaviour (PHB)</th>
<th>DSCP Value</th>
<th>IP Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>000000</td>
<td>0</td>
</tr>
<tr>
<td>Assured Forwarding</td>
<td>001010</td>
<td>001100</td>
</tr>
</tbody>
</table>

With the DiffServ field and the DiffServ Code Point (DSCP) as described, packets of data can be marked with a particular precedence, delay, throughput, reliability, cost or MBZ (Multiple BitZero) as detailed in RFC 2697.
<table>
<thead>
<tr>
<th>Class 2</th>
<th>AF21</th>
<th>AF22</th>
<th>AF23</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>010010</td>
<td>010100</td>
<td>010110</td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td>AF31</td>
<td>AF32</td>
<td>AF33</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>011010</td>
<td>011100</td>
<td>011110</td>
<td></td>
</tr>
<tr>
<td>Class 4</td>
<td>AF41</td>
<td>AF42</td>
<td>AF43</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>100010</td>
<td>100100</td>
<td>100110</td>
<td></td>
</tr>
<tr>
<td>Expedited Forwarding</td>
<td>EF</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>101110</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The values in decimal are given in the following table:

<table>
<thead>
<tr>
<th>DSCP</th>
<th>Binary</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>000000</td>
<td>0</td>
</tr>
<tr>
<td>CS1</td>
<td>001000</td>
<td>8</td>
</tr>
<tr>
<td>AF11</td>
<td>001010</td>
<td>10</td>
</tr>
<tr>
<td>AF12</td>
<td>001100</td>
<td>12</td>
</tr>
<tr>
<td>AF13</td>
<td>001110</td>
<td>14</td>
</tr>
<tr>
<td>CS2</td>
<td>010000</td>
<td>16</td>
</tr>
<tr>
<td>AF21</td>
<td>010010</td>
<td>18</td>
</tr>
<tr>
<td>AF22</td>
<td>010100</td>
<td>20</td>
</tr>
<tr>
<td>AF23</td>
<td>010110</td>
<td>22</td>
</tr>
<tr>
<td>CS3</td>
<td>011000</td>
<td>24</td>
</tr>
<tr>
<td>AF31</td>
<td>011010</td>
<td>26</td>
</tr>
<tr>
<td>AF32</td>
<td>011100</td>
<td>28</td>
</tr>
<tr>
<td>AF33</td>
<td>011110</td>
<td>30</td>
</tr>
<tr>
<td>CS4</td>
<td>100000</td>
<td>32</td>
</tr>
<tr>
<td>AF41</td>
<td>100010</td>
<td>34</td>
</tr>
<tr>
<td>AF42</td>
<td>100100</td>
<td>36</td>
</tr>
<tr>
<td>AF43</td>
<td>100110</td>
<td>38</td>
</tr>
<tr>
<td>CS5</td>
<td>101000</td>
<td>40</td>
</tr>
<tr>
<td>EF</td>
<td>101110</td>
<td>46</td>
</tr>
<tr>
<td>CS6</td>
<td>110000</td>
<td>48</td>
</tr>
<tr>
<td>CS7</td>
<td>111000</td>
<td>56</td>
</tr>
</tbody>
</table>
We can observe the construction of the DSCP values by taking the example for the Per Hop Behaviour AF32. AF32 is derived from the binary 011100. The red section is where the 3 comes from in AF32 and is the Behaviour Aggregate. The blue section is where the 2 comes from in AF32 and is the Drop Probability. The final green section with the 0 is ignored.

The decimal IP Precedence value is derived from the red portion and is also called the Class Selector (CS). Often the DSCPs are configured in decimal. The decimal value is derived from all 6 bits of the DiffServ field as you will see from the table.

Notice how the three Most Significant Bits (MSB) determine the Class and map directly to the IP Precedence bits. The Class Selector Code Points all have the form xxx000. The three LSBs determine the Drop Probabilities and are ignored by IP Precedence-only devices. Also notice that the LSB is always '0'.

Packets from different sources can have the same DSCP value and so can be grouped together as a Behaviour Aggregate and treated in the same manner. A packet with a DSCP not mapped to one of the above PHB i.e. different from the recommendations, will have its DSCP mapped to the Default PHB of 000000. Be aware that the table is recommending the values and different manufacturers could use different ones.

Expedited Forwarding within DiffServ is as close as you can get to IntServ as it provides low-loss, low-latency, low-jitter and guaranteed bandwidth.

Assured Forwarding is described in RFC 2597 and Expedited Forwarding is described in RFC 2598.

To manage the policies you need to use a Common Open Policy Service (COPS).

2.3 Integrated Services (IntServ)

Integrated Services (IntServ) is for the internal network which is more easy to control. IntServ can deal with many QoS requirements at one time, the difference with IntServ wrt DiffServ, is that it specifies its traffic requirements and the network sets up the requirements if they are possible, before traffic is sent. If the network is unable to set up the transmission path then the call does not happen, so there is a form of Call Admission Control (CAC) (see later).

RFC 1633 describes IntServ.

Resource Reservation Protocol (RSVP) fits in with the IntServ model and is a signalling protocol that allows applications to request bandwidth and QoS characteristics and provides a mechanism to tell if the network can meet the demands. This mechanism uses a sort of 'scout' that checks out the network ahead of data transmission. Delay and Bandwidth can be reserved and flows can be signalled by the end station or the router.

If a host makes a reservation for a data stream then each network device through the
network attempts to make a reservation to the next hop, this is called the Reservation Path. A successful reservation is made if the user has the permission and if there are enough resources to provide the QoS. Once the destination is reached the same is done for the return data stream so that there is a QoS path back. With traditional RSVP, if just one network node says 'no' to a reservation, then no path is set up at all. As well as accepting a reservation, RSVP also uses a 'Traffic Classifier' that uses DiffServ to instruct the routers how to mark and treat each flow.

Once the data stream has completed, a PATHTEAR and RESVTEAR is triggered to terminate the call and release resources back to the mainstream traffic. The queues are structured such that The IntServ queue (RSVP) has the highest priority, over and above the highest priority DiffServ queues. Applications can use RSVP in combination with **Weighted Fair Queuing (WFQ)** to provide **Guaranteed-Rate Service** so that a VoIP call can for instance reserve bandwidth. In addition, applications can use RSVP in conjunction with a technology such as **Weighted Random Early Detection (WRED)** to provide a **Controlled-Load Service** which provides low delay for video playback for instance, even if there is congestion.

It is now possible to synchronise RSVP call setup with the H.323 call setup between the **Originating Gateway (OGW)** and the **Terminating Gateway (TGW)**. The H.323 setup is suspended just before the destination phone rings. During this period the RSVP reservation is carried out in both directions between the OGW and the TGW because voice calls are two-way and require bandwidth in both directions. Then H.323 kicks in again with either an H.225 Alerting/Connect message (which causes the destination phone to ring) or an H.225 Reject/Release.

802.1p, ToS and RSVP can be set per packet, or per port in a switch.

**RFC 2205** describes RSVP. In addition, Cisco have a proprietary version of RSVP called **Proxy-RSVP** where a best-effort RSVP path can be set up using routers that do not understand RSVP and would ordinarily cause the path to fail.

### 3. Congestion Management

#### 3.1 Queuing

When congestion occurs there needs to be way of sorting the traffic out. Packets that have been marked can be identified and placed in queues. The queues can vary in how much and when they can load up the link with the packets contained within their queue. The shorter the queue length; the lower the latency. Even the fastest of links on the fastest of layer 2 switches can suffer from congestion if the data packets are large. The simplest method of queuing is **Round Robin (RR)** where each queue is serviced one packet from each at a time. Using IP precedence, higher priority traffic can be given a low weight thereby allowing it access to greater bandwidth. This is called **Weighted Round Robin (WRR)**. Adding a **Priority Queue (PQ)** allows you to always serve delay sensitive traffic first. If the priority queue is empty then RR or WRR operates as normal.
Companies such as Cisco have their own methods of dealing with congestion, here we will look at how Cisco have developed their queuing mechanisms.

3.2 First-in, First-out (FIFO)

Whatever the size, the first packet to enter an interface is the first to leave that interface. No queues are involved. This is often the default mechanism for receiving and sending packets on an interface.

3.3 Weighted Fair Queuing (WFQ)

WFQ allocates bandwidth according to IP Precedence using weightings and the number of WFQ queues depend on the number of flows. Routing protocols and LMI traffic bypass the WFQ mechanism. WFQ respects the top priority queues used for RSVP. Any queuing mechanism based on WFQ will preserve 'Reserved Queues' traditionally for RTP Reserve and now RSVP. If RSVP is configured, then RSVP will use this reserved queue. Reserved queue traffic is given a low weight of 4 which means 'High priority'. RSVP is initially given a weight of 128 whilst the reservation is set up, then the RSVP traffic is given a weight of 4.

IP traffic is classified based on source IP address and port, destination IP address and port and IP Precedence. Frame Relay traffic uses the DLCI to classify traffic, IPX uses MAC addresses and socket numbers. WFQ tries to interleave smaller packets between the larger packets. The decision as to which packet in the queue is allowed to go through is based on the first packet to finish entering the queue which therefore favours the shorter packets, so low bandwidth traffic takes priority over high bandwidth traffic.

Once the priority queues have been serviced the other IP traffic is weighted and queued accordingly. The weighting factor is dependent on the IP Precedence. The way this is calculated is as follows:

Say you have three flows with IP precedences of $P_1$, $P_2$ and $P_3$.

Add up $(P_1 + 1)$, $(P_2 + 1)$ and $(P_3 + 1)$.

Each flow will get a proportion of the link bandwidth e.g. the flow with $P_1$ will get the proportion $(P_1 + 1) / ((P_1 + 1) + (P_2 + 1) + (P_3 + 1))$.

As an example, if the three flows had IP Precedences of 1, 3 and 5 then the flow with IP Precedence of 1 will get the proportion $2/12$ whereas the flow with IP Precedence of 5 will get the proportion $6/12$, which is half the bandwidth being significantly more than $1/6$ of the bandwidth.

The problem with WFQ is that as the number of flows increases the higher priority flows have a less significant impact as all the flows get served. For instance, if there are 18 flows with IP Precedence of 1 and one each of IP Precedence 2, 3, 4, 5, 6 and 7, then the flow with IP Precedence 5 will get 6/70 of the link and 18 flows will each get 2/70 of the link. There is now very little difference between the critical traffic and the lower priority traffic.

WFQ across a number of interfaces is called Distributed Weighted Fair Queuing.
3.4 Priority Queuing (PQ)

This utilises 4 queues: High, Medium, Normal and Low. Which queue a packet goes into is determined by whatever classification that packet belongs to. Unclassified traffic is placed into the Normal queue. The higher queues have to be emptied before the lower ones. Potentially this could lead to lower queues being starved of time to empty, this is called Protocol Starvation. High priority traffic would be time critical traffic such as DEC LAT or SNA and the packets in this queue would be forwarded before the packets in the other queues. Low priority traffic may not get forwarded at all. If a queue becomes longer than the specified limit, then subsequent packets trying to enter that queue are dropped indiscriminately (Tail Drop). Priority queuing is a method whereby the engineer can decide which packets are more likely to be thrown away rather than the router making indescriminant decisions.

The top priority queue used for RSVP is not respected by PQ.

3.5 Custom Queuing

Custom Queuing provides the facility for up to 16 queues with varying priorities. This method is used for assigning queue space per protocol. Queue 0 is reserved for router to router communication whilst queues 1 to 16 are for the user. The router deals with each queue in a Weighted Round Robin fashion allowing the configured amount of traffic from each so that no one queue monopolises the bandwidth. Each queue has a default length of 1518 bytes. You need to characterise the average packet sizes for each protocol and then specify the byte count and queue length for each queue. If a particular queue is empty, then the extra space is dynamically shared out amongst the other queues.

The top priority queue used for RSVP is not respected by CQ.

3.6 Class-Based Weighted Fair Queuing (CBWFQ)

With CBWFQ you can manually define up to 64 traffic classes (including one which is the default class queue) that determine the exact bandwidth that is allocated to certain traffic types. With WFQ, a large number of dynamic queues can still cause latency for high-priority traffic. With CBWFQ each class is queued and dealt with exhaustively so you can manually ensure that certain traffic such as voice will not suffer latency. Classifying packets that need to be queued using WFQ does not guarantee that all packets will be serviced.

The class is defined using criteria to match the traffic. These criteria could be access control lists, protocols and interfaces. You then add characteristics to this class of traffic by assigning bandwidth the maximum number of packets (queue limit) and a drop policy. The weight is derived from the bandwidth that you specify and is used by the WFQ mechanism as described above to provide guaranteed delivery.
Unclassified traffic is assumed to be in the Default Class and is acted on by WFQ and is best-effort treated. Non-IP traffic is assigned to flow 0. If the number of packets from a matched flow exceed the queue limit, then by default tail drop occurs whereby the last packets get dropped. For the drop policy, you can instead implement **Weighted Random Early Detect** for a particular class (WRED can also be applied to an interface) so that certain more critical packets (higher precedence) are allowed into the queue and less critical packets dropped. Unused bandwidth for one class is shared amongst the other classes. The recommended maximum bandwidth set aside within the defined classes is 75%. This leaves room for routing protocols and layer 2 keepalives etc.

### 3.7 IP RTP Priority

This has been known as PQ/WFQ, however the PQ is strictly IP RTP priority and no other protocols, so it is not PQ as we have so far understood it so this method is only available for IP RTP traffic. IP Real Time Protocol runs over UDP and gets serviced exhaustively first once the ports have been recognised. The other queues are serviced by WFQ. With IP RTP Priority you get the benefit of still maintaining the top priority queue. The UDP port range (16384 - 32767) used by RTP traffic uses this top priority queue. IP RTP Priority can also use CBWFQ as well as just WFQ which gives PQ/CBWFQ, however the PQ is strictly for RTP traffic only.

The port range 16384 to 32767 is a recommendation. You could use other UDP ports if you wish, video traffic often does.

### 3.8 Frame Relay IP RTP Priority

This scheme uses a Frame Relay PVC to provide a strict priority queuing scheme for RTP traffic.

### 3.9 Low Latency Queuing (LLQ)

LLQ is also known as PQ/CBWFQ. Unlike IP RTP Priority, the single Priority Queue within LLQ can deal with TCP exchanges as well and is therefore the preferred choice to use when queuing all types of traffic. If you configure IP RTP Priority as well as LLQ, then IP RTP Priority will take precedence. You would then need to be careful not to put jitter sensitive traffic through both priority queues.

### 4. Congestion Avoidance

#### 4.1 Global Synchronisation

When an interface becomes busy, queues fill up and you are more likely to suffer **Tail Drop** where packets, however they are marked, are dropped indiscriminently. If there are multiple TCP flows through a congested link, then chances are that many flows will be dropped at the same time. The TCP slow-start mechanism will then kick in at the same time for all the flows that were interrupted. This leads to **Global**
Synchronisation where there are waves of congestion followed by waves of under utilisation of the link which is not an efficient use of the link.

4.2 Random Early Detection (RED)

Rather than have flows dropped simultaneously a technique called Random Early Detection was developed. This uses the concept of queue depth which is the amount that you fill a particular queue. When the average queue depth reaches a Minimum Threshold, packets start to be dropped at a rate that is determined by the Mark Probability Denominator. If the Mark Probability Denominator is 512, then if the average queue depth reaches the Minimum Threshold, 1 packet in every 512 is dropped. As the queue grows even larger, then the probability increases and consequently a larger percentage of the packets are dropped until you reach the Maximum Threshold which is effectively the end of the queue where ALL packets are dropped.

The drops do not occur all at once and a session that was hit once, will not be hit again until it starts responding. This thereby minimises global synchronisation and makes more efficient use the the link bandwidth during congestion. An issue with RED is that TCP sessions become slower and as TCP traffic drops off, the queues begin to fill proportionally more with UDP traffic which means that there is an increased risk of UDP traffic being dropped. UDP does not respond well to drops, particularly voice and video traffic!

4.3 Weighted Random Early Detection (WRED)

WRED is Cisco's implementation of RED and adds the use of DSCP/IP Precedence to enable you to weed out low-priority packets from the queue and to drop them by giving these packets a higher drop probability. When using the DSCP, WRED uses the Drop Preference portion of the PHB to determine the Drop Probability. This thereby minimising the chances of tail-dropping high-priority packets. Using WRED does not preclude the chances of dropping voice packets (Class of Service/IP Precedence of 5) and therefore is not a recommended technology to use within a voice network.

Non-IP packets are treated as if they had an IP Precedence of 0, so making these packets the most likely to be dropped.

WRED does not like non-adaptive flows such as UDP because UDP does not mind if it gets dropped whereas TCP has its own built in mechanisms to throttle back.

4.4 Flow-Based Weighted Random Early Detection (FRED)

A flow is made up of a 5-tuple of source address and port, destination address and port, and IP Precedence. FRED can track flows and penalise those flows that take up more than their fair share of the buffer space. FRED does this by looking at each flow in turn and observing how each one uses the queues. FRED then allows some
bursting space by scaling up the number of packets allowed for each flow in the queues and then waits to see if any flows exceed this number. If a particular flow does exceed the number of packets allowed, FRED increases the probability of packets from that flow being dropped.

5. Link Efficiency

Smaller data packets used for applications such as voice and telnet are susceptible to jitter (delay variation) when large data packets are traversing the same interfaces. **Link Fragmentation and Interleaving (LFI)** at layer 2 is a mechanism whereby large packets are fragmented and the smaller data packets interleaved with the fragments. You need to be able to control the serialisation delay of the data fragments so that there is a maximum time the delay-sensitive packets have to wait if they are caught in a queue behind a data fragment.

**Serialisation Delay** is the time it takes for an interface to push bits out on to the circuit. Only one packet can be dealt with at a time (the nature of serial transmission). If the circuit speed is high e.g. 768 Kbps, then a 1500 byte data frame would take 1500 x 8/768000 = 16ms to be pushed on to the circuit. The same size frame on a 56 Kbps modem link would take 1500 x 8/56000 = 214ms. The maximum acceptable end-to-end delay for delay sensitive traffic such as voice is generally considered to be 150ms (G.114 standard) and the Serialisation delay is only one of the factors involved in the end-to-end delay. On a slow link such as 56 Kbps a small packet such as 128 bytes would only take 128 x 8/56000 = 18ms, however a large packet that is before it will hold up the smaller packet. This shows how damaging large packet sizes are on slow links due to the delay and jitter they incur for small packets trying to get through. Aim for a serialisation delay of not more than 10ms (20ms max). You do this by decreasing the packet size and/or increasing the link speed. The following table is a useful guide of fragment size recommendation compared with link speed assuming a 10ms blocking delay.

<table>
<thead>
<tr>
<th>Link Speed (kbps)</th>
<th>Fragment Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>70</td>
</tr>
<tr>
<td>64</td>
<td>80</td>
</tr>
<tr>
<td>128</td>
<td>160</td>
</tr>
<tr>
<td>256</td>
<td>320</td>
</tr>
<tr>
<td>512</td>
<td>640</td>
</tr>
<tr>
<td>768</td>
<td>1000</td>
</tr>
<tr>
<td>1536</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Multilink PPP (MLP) Interleaving** can be used to give jitter sensitive traffic priority. See the section on **MLP Interleaving**.

**FRF.12** (for VoIP) and **FRF.11 Annex C** (for VoFR) can be used to give jitter sensitive traffic priority. See the section on **Frame Relay Fragmentation**.

PDF created with pdfFactory Pro trial version [www.pdffactory.com](http://www.pdffactory.com)
In a Voice over IP environment Real-Time Transport Protocol is an Internet standard (RFC 1889) used to transport real-time data. Whereas TCP is used for signalling protocols such as H.323, RTP uses UDP for transport because if packets get lost, there is no point in re-sending the data. The RTP header is 12 bytes in length and follows the 8-byte UDP header and the 20-byte IP header.

When using RTP, a technique called Compressed RTP (cRTP) can be utilised whereby the IP header, UDP header and the RTP header can be compressed from the usual 40 bytes down to between 2-4 bytes. This is suitable for point-to-point links, preferably using hardware for the compression.

If speech is sampled every 20ms then you end up with a VoIP payload of 20 bytes. Comparing this to the 40-byte (12 + 8 + 20) VoIP header you can see that the header is twice the size of the payload. Using cRTP compresses this 40-byte header to between 2 and 5 bytes. This is useful for small payload voice protocols such as G.729 where a typical voice call without cRTP would be 26.4Kbps whereas with cRTP this reduces to 11.2Kbps. cRTP is used on an link by link basis and is typically implemented in software rather than hardware so the CPU is loaded heavily when using cRTP. This, together with the benefits being most apparent on slow links, means that cRTP ought only to be used with slow links i.e. less than 768Kbps.

6. Traffic Policing

Conditioning traffic that enters a network can be carried out by controlling bursty traffic and making sure that designated traffic flows get the correct bandwidth. This can often mean 'chopping' off the excess flows, or changing the Precedence of the packets that exceed the bandwidth. Policing ingress traffic causes TCP resends.

The concept of Token Bucket or Leaky Bucket is used when looking at Traffic Policing or Traffic Shaping. Taking concepts from Frame Relay Mean Rate (CIR) = Burst Size (Bc) / Time Interval (Tc). Imagine a bucket which is being filled with tokens at a rate. The bucket has a capacity Bc + Be and each token has permission to send a fixed number of bits. When a packet arrives if it matches the class for that bucket then it is added to the queue and released from the queue according to the specified rate if there are enough tokens available in the bucket. If the bucket fills to capacity then new tokens are discarded ('leaked' out) and cannot be retrieved for use by subsequent packets. If there are not enough tokens in the bucket, then the system may wait for enough tokens before sending the packet (this is traffic shaping) or the packet may be discarded (traffic policing).

The token bucket methodology means that any bursts are limited to the bucket capacity and the transmission rate is governed by the rate at which tokens are added to the bucket.

7. Traffic Shaping

Traffic can also be conditioned by 'Shaping' it rather than use more brutal policing
techniques. Shaping the traffic means that traffic is ‘smoothed’ out using queues to hold up packets just delaying them until there is a trough. The leaky bucket provides a buffer of tokens that can deal with controlled bursts. This is done on egress traffic rather than ingress. **Frame Relay Traffic Shaping (FRTS)** is a good example where the CIR is adhered to by holding back packets during a burst so as to let them go during the next quiet period, maybe even allowing occasional bursting provided that the average utilisation is adhered to over a specified period of time. Cisco use the concept of **Generic Traffic Shaping** when applying traffic shaping principles within a much broader context such as the IP environment. The trouble with Traffic Shaping is that it introduces latency.

Shaping traffic is important if you have mismatched link speeds at each end e.g. a Frame Relay hub-spoke arrangement where you do not want to oversubscribe the hub site and risk packets being dropped in the cloud because you exceed the CIR at one end.

**Generic Traffic Shaping (GTS)** can be applied to any outbound interface and operates with SMDS, ATM, Frame Relay and Ethernet. Outbound traffic can be shaped to meet the requirements of the receiving device(s), DSCPs can be set, BECNs listened to and access lists can be used to determine the traffic to shape.

**Frame Relay Traffic Shaping (FRTS)** is described in the section on FRTS.

### 8. Call Admission Control (CAC)

#### 8.1 Overview

Voice and Video traffic, or calls, can be restricted by way of CAC tools that determine whether or not there is sufficient bandwidth. One too many calls on a link would cause the quality of the other calls to suffer so the extra call is prohibited. CAC’s use is to prevent congestion from voice and video occurring in the first place, before Quality of Service tools are called into play.

CAC can be implemented in a number of ways:
- Locally - i.e. at the originating gateway, by limiting the DS0s (not for IPT), the maximum number of connections, bandwidth on Voice over Frame Relay, Trunk Conditioning and Local Voice Busyout (LVBO).
- Based on measurement - PSTN Fallback or Advanced Voice Busyout (AVBO)
- Based on available resources - Resource Availability Indicator (RAI), H.323 Gatekeeper zone bandwidth or RSVP.

#### 8.2 Local CAC

Local CAC is not aware of the topology of the network and cannot guarantee QoS. Benefits include no messaging overhead and no delay.

Limiting DS0s (timeslots) between the PBX and the originating gateway is a simple...
method that applies to toll bypass scenarios only.

In a Cisco IPT environment you can limit the number of concurrent VoIP connections on a per dial peer basis so that in total the trunk link will have a total limit. If a particular dial peer has calls directed to 477... then if you limited the number of simultaneous calls to 5, any further calls will be 'hairpinned' back to the PBX and the PSTN would be used for the excess call(s).

In VoFR networks FRF.11 is used to packetize voice. At layer 2 it is therefore possible to distinguish between the Frame Relay data header (FRF.3.1) and the Frame Relay Voice header (FRF.11) and so provide CAC disallowing calls that exceed the bandwidth allocated to voice calls.

On a VoFR, VoATM or VoIP point-to-point connection you can nail down voice connections, this is called **Connection Trunk**. If the trunk link uses **Channel Associated Signalling (CAS)** (a robbed bit technology common in the US), then you use the A, B, C and D bits as keepalives for each channel. On detection of a channel failing or being busy, a **Busy** or **Out of Service (OOS)** signal can be sent back to the PBX in a form that the PBX understands. This is called **Trunk Conditioning** and does not work with Common Channel Signalling (CCS).

**Local Voice Busyout (LVBO)** is used to monitor LAN and WAN ports for failure and then to busy out voice ports or the trunk to the PBX. This is used for analogue or CAS trunks.

### 8.3 Measurement-Based CAC

Measurement-Based CAC is not topology-aware nor does it guarantee QoS.

Cisco have a proprietary Client-Server (Responder) protocol called the **Service Assurance Agent (SAA)** which gives an indication of network conditions by generating traffic and measuring response times, jitter, connect time, packet loss, availability and application performance. These indicators frequently form the basis for Service Level Agreements. Originally designed for SNA traffic it has been extended to simulate VoIP and Web traffic. An RTP/UDP header is used for VoIP simulation and the SAA operates by the client sending a probe to a server that responds.

Two factors **Delay** and **Packet Loss** are used by the SAA to calculate the ITU G.113 **International Calculated Planning Impairment Factor (ICPIF)** for an application on a given network. This value can be used to make CAC decisions. The ITU use the following guideline values:

- 5 - Very good
- 10 - Good
- 20 - Adequate
- 30 - Limiting Case
45 - Exceptional limiting case
55 - Customers likely to react strongly

**Advanced Busy Out (AVBO)** supports H.323, MGCP and SIP environments and can busy out voice ports or trunks based on ICPIF values. The measurements are obviously IP-based. AVBO only works with Analogue and CAS trunks.

**PSTN Fallback** also uses SAA and applies to individual call setups. IP addresses are cached along with their ICPIF values so that SAA probes are only required on an initial call to a new IP address (therefore introducing a postdial delay just for the new entry). PSTN Fallback is useful because it busies out individual calls rather than ports or the trunk, so it is not so brutal. You cannot use PSTN Fallback with RSVP.

### 8.4 Resource-Based CAC

Techniques such as **Resource Availability Indicator (RAI)** and **Gatekeeper Zone Bandwidth** examine the resources at the time of the call request and then make a single CAC decision at that time. The resources examined will include available timeslots, Gateway CPU use, memory use, DSP use and bandwidth availability.

Techniques such as **Resource Reservation Protocol (RSVP)** just look at bandwidth availability and reserves bandwidth and guarantees QoS for the call duration.

RAI is used in H.323 networks with a Gatekeeper. The Gateway informs the Gatekeeper when it is running low on resources. This is determined by configuring a low water mark and a high water mark. The Gateway sends an RAI when the resources fall below the low water mark and when the resources rise above the high water mark. The Gatekeeper then makes routing decisions based on these RAIs.

In an H.323 Gatekeeper environment, gatekeepers can be grouped into zones. The gatekeeper in each zone can limit bandwidth usage within its own zone and also between its own zone and other zones. In a single zone the gatekeeper limits the bandwidth allocation to the smallest link between the gateways within its zone, because the bandwidth limitation is configured per zone. This not very efficient as some links within the zone with larger bandwidths can carry more calls, and this also applies to where you have multiple zones with multiple gateways to each zone and the gatekeepers limit calls to the number that can be carried on the smallest links to each of the zone, even if there are larger links to each zone that could be used. Using Gatekeeper CAC is the only method available in distributed H.323 zone environments.