Quality of Service

December 12, 2020

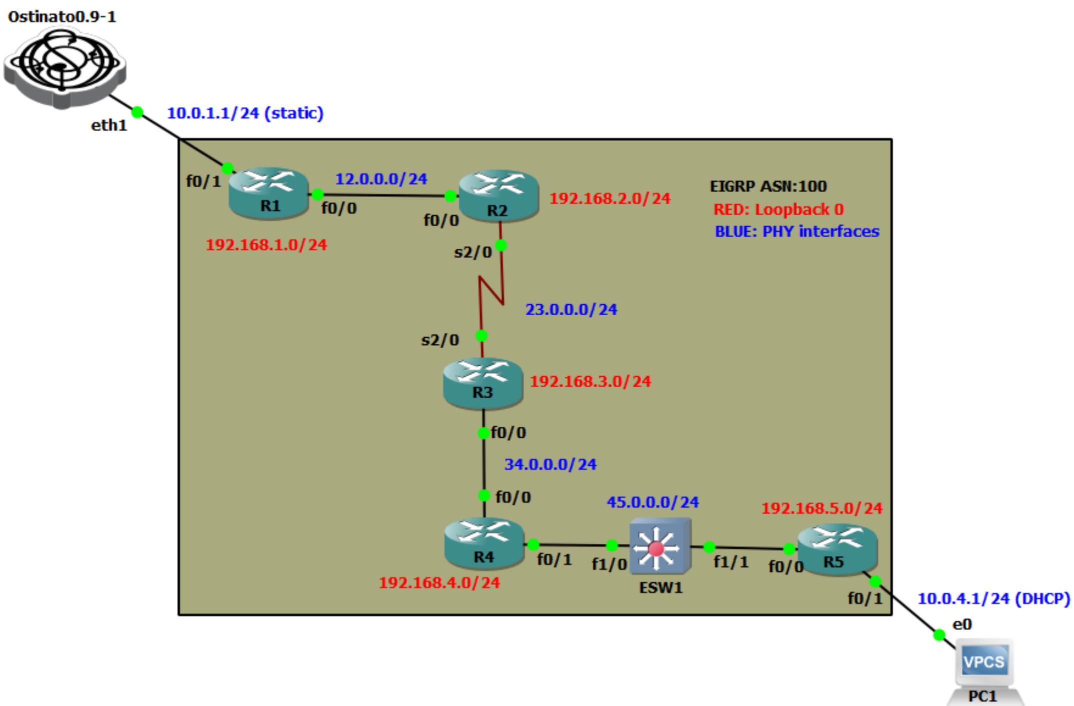
1 Introduction to QoS

Prerequisits * Packets and header structures * Life cycle of a packet/frame * Familiarity with IOS CLI * Basics of Routing and Switching

Agenda * **What is QoS and why we need it? :** What problem does this feature solve * **QoS based tools :** Frameworks such as MQC, HQF * **Classification and Marking :** How to control the priotitization of traffic in a congested network. (IP-Precedence, DSCP, NBAR etc.) * **Congestion Management**: How to control the piling up of data in an interface-buffer * **Queue management** : * **Congestion Avoidance** : Proactively take action before the memory gets filled up (WRED, WTD).

2 Lab Setup

2.1 Topology



2.2 Router Base Config

Router 1

```
!r1
conf t
    int f0/0
        ip add 12.0.0.1 255.255.255.0
        no sh
    int l0
        int l0
            ip add 192.168.1.1 255.255.255.0
    router eigrp 100
        no auto
        pass def
        net 0.0.0.0
        no pass f0/0
end
```

Router 2

```
!r2
```

```
conf t
    int f0/0
        ip add 12.0.0.2 255.255.255.0
        no sh
    int s2/0
        ip add 23.0.0.2 255.255.255.0
        no sh
    int l0
        int l0
            ip add 192.168.2.1 255.255.255.0
    router eigrp 100
        no auto
        pass def
        net 0.0.0.0
        no pass f0/0
        no pass s2/0
end
```

Router 3

```
!r3
conf t
    int f0/0
        ip add 34.0.0.3 255.255.255.0
        no sh
    int s2/0
        ip add 23.0.0.3 255.255.255.0
        no sh
    int l0
        int l0
            ip add 192.168.3.1 255.255.255.0
    router eigrp 100
        no auto
        pass def
        net 0.0.0.0
        no pass f0/0
        no pass s2/0
end
```

Router 4

```
!r4
conf t
    int f0/0
        ip add 34.0.0.4 255.255.255.0
        no sh
    int f0/1
        ip add 45.0.0.4 255.255.255.0
        no sh
    int l0
```

```
        int l0
            ip add 192.168.4.1 255.255.255.0
    router eigrp 100
        no auto
        pass def
        net 0.0.0.0
        no pass f0/0
        no pass f0/1
end
```

Router 5

```
!r5
conf t
    int f0/0
        ip add 45.0.0.5 255.255.255.0
        no sh
    int f0/1
        ip add 10.0.4.1 255.255.255.0
        no sh
    int l0
        int l0
            ip add 192.168.5.1 255.255.255.0
    exit

    ip dhcp excl 10.0.4.1 10.0.4.100
    ip dhcp pool LOCAL_POOL
        network 10.0.4.0 255.255.255.0
        def 10.0.4.1

    router eigrp 100
        no auto
        pass def
        net 0.0.0.0
        no pass f0/0
end
```

2.3 Ostinato Setup

- Use any interface but **eth0** (Management Interface) for connecting the router.
- Open Ostinato → select generator interface (**eth1**) → file → New Steam → Set the following settings
- **Name** : any, e.g. QoS_TEST
- **Protocol Section**
  - L1 = Mac
  - VLAN = Untagged
  - L2 = Ethernet II
  - L3 = IP

  - L4 = UDP
  - L5 = None
  - Keep the rest as default
- **Protocol Data**
  - MAC
    - Source: Any (default : 00:00:00:00:00:00)
    - destination: Routers interface use `sh int IFACE | in bia` to get the MAC
  - IPv4
    - Source: any (e.g. 10.0.1.2, just keep on the same subnet)
    - Dest: Far end interface IP, 10.0.4.1
    - TOS: override the default TOS (00) to 60 (will be explained later)
- **Stream Control**
  - Rate = 100,000 bits/sec. This will auto calculate the packets/sec
  - After Stream action = Goto First (to make it a indefinite loop)
- **Save Settings**: Save → Apply → Transmit start (play button)
  - By default Ostinato sends streams sequentially if multiple streams are configured. To make is simultainious, select the interface (port) you want to run in parallel → Port configuration → interleaved streams.
- **Verification:** Go to the far end router i.e. R5 and check the input rate of f0/0 using `sh int f0/0 | in input rate` commnad. IOS by default uses 5 miniutes moving average based load calculation, you may change it using `load interval 30` i.e. 30 sec (minimum) load interval in the interface level. Notice the rate in not 100kbps (due to the slow serial link)

```
sh int f0/0 | in input rate
30 second input rate 56000 bits/sec, 112 packets/sec
```

Also, check the CPU throtolling histoty using `sh processes cpu history` command

```
100
 90
 80
 70
 60
 50
 40
 30
 20 **************     *****************************************
 10 ************************************************************
   0....5....1....1....2....2....3....3....4....4....5....5....6
             0    5    0    5    0    5    0    5    0    5    0
               CPU% per second (last 60 seconds)
```

```
100                                                       *
 90                                                       *
 80                                                       *
 70                                                   *   *
 60                                                   *   *
```

```
50                                                     *   *
40                                                     *   *
30                                                     *   *
20 ######*                                             *   *
10 #######                 * *                         # **#
  0....5....1....1....2....2....3....3....4....4....5....5....6
            0    5    0    5    0    5    0    5    0    5    0
             CPU% per minute (last 60 minutes)
            * = maximum CPU%   # = average CPU%
```

3 What is QoS ?

3.1 Basics

- QoS stands for Quality of Service
- What problem does it solve?
  - Provides **predictable management of network resources** during times of congestion.
  - assist in **maximising the end-user experience** of critical sessions. A email traffic may be delayed but not a voice/video traffic.

  - provides **differentiated services** to packets based upon pre-defined user criteria. User may define which traffic is more critical than other.
- How does QoS provide those services?
  - There are many QoS features. Some features only performs single job, others do multiple jobs.
  - The core tasks that a QoS feature may accomplish can be as follows. Some features performs one one, some does multiple.
    - **Classification of Data**: Segrigation os streams from an unified traffic.
    - **Queue Management**: Size, placement of packets (which packet goes to which queue), Scheduling order, Transmission Rate (Traffic Shapping).
    - **Preemptive drops**: rate limit multiple ingress interfaces to prevent an egress interface to get overloaded.

    - **Marking of packets**: Mark packets based on its class to tell the router to treat them with different priority.

- **Quality of Experience (QoE)**: The result an user experience as an end-to-end consequece of QoS policies.

3.1.1 Packet life cycle

Packer life cycle in a Router

- packet arrived on the ingress interface (RX-RING). Interface has a DMA access to certain part of the memory called RX-RING.

- Packet queued in the memory buffer. Memory sends HW-interrupt to the CPU and relinquish control of the memory location in RX-RING to the CPU. Based on different switching algorithm (Process, Fast, CEF) router processes it.
- RIB/FIB lookup is done and forwording decission is made.
- Packet header is manipulated and it is placed on the Tx-RING.
- CPU relinqished the control of the TX-RING memory location to the Egress interface and the interface puts it on the line.

The avobe process is mainly used in a roter. However in a switch, the process is different. Memory architecture of a switch could be of two types * **Shared Memory**: In smaller switches e.g. Catalist 3500/4500, the interfaces and the ASIC (Forwarding engine) share a central TCAM memory block via a memory manager (MMU). A schedular makes a predictable schedule of the process. CPU does not see any of the process, everything is handled by the ASIC. CPU performs tasks like Spanning-Tree etc. * **Distributed Memory**: In a larger switches e.g. Catalist 6500 series, there are multiple line card (Supervisor engine or Control plane and Interface Cards or Data Plane). Each DP line card has multiple interfaces with deficated ASIC that mainly performs simple jobs like FCS chek. Each interface-ASIC has their own memory (devided in Tx and Rx memory). All these individual ASIC has an access to the upstream Supervisor. ASICs in the Supervisor Engine decides the destination address and signals the appropriate eggress ASIC to send it. The mode of communication between Supervisor and Line ASICs could be via a Bus or a Ring.

- **Packet Life Cycle in a Shared Memory system**
  - packer arives on the ingress interface
  - interface ASIC immidiately forwards the packet into common shared memory pool.
  - forward decission is made by the forwarding ASIC
  - memory ownership is transfered to the interface ASIC
- **Packet Life Cycle in a Distributed Memory system**
  - Packet arrives on an ingress interface.
  - Interface/port ASIC places it into a local Queue/Buffer.
  - Forwards the packet to the Forworking engine via a BUS or a RING.
  - Forwrding ASIC makes a forwarding decission.
  - packet transmitted onto shared ring/bus to the egress buffer.
  - egrees interface ASIC puts it into the line.

As a core takeaway, it is clear that during the process the data must be stored temporarily somewhere in the device, typically in a buffer. The router manages this buffer with software, a switch does the same in hardware level. If there is no congestion, QoS is not needed. Commonly the Egress buffer suffers from congestion as the ingress interfaces are typically rate-limited by the ISP.

3.1.2 Buffers and Queues

- **Buffer**
  - physical memory used to store packets before and after the forwarding decisions are made.
  - On routers this same memory can be allocated to interface as ingress/egress queues.
  - Shared memeory (of which, part is allocated as buffers) is also used by lots of other CPU processes.
- **Queues**

  - On routers a queue is a logical part of the shared memory buffers.
  - On switches, individual interfaces (or line-cards) have their own memroy which is used as interface queues.
- Buffer-Queue Configuration
  - Configuration os buffer is not normally part of QoS
  - Buffer configuration would involve modifying the quantity of buffes allowed for particuar sized packet
  - **Buffer-Tuning** in cisco IOS is not generally recomanded.
  - QoS configuration is only applicable to queeues.
  - QoS does not alter the physical buffer size, rather it controls how packets are treated inside a queue.

3.1.3 What is Congestion?

- During times of no congestion, QoS is not needed. Packets forwarding follows a FIFO mode in Rx-Tx ring.
- **Egresss Congestion**: Packet forwarded to egress interface faster than Tx-RING can handle them.
- **Ingress Congestion**: Packets arrive in multiple ingress interface faster than the forwarding engine can process them (not a typical scenario). In modern routers and switches the lookup processes are doen in hardware level, therefore it looksup millions of packets per seconds, i.e. 10~20 times faster than the line rate.

3.1.4 Result of a Congestion

- **Delay**: Typically the intr-frame space is fixed. Delay is the **Uniformly scaled IFS**. Delay is irritating for a voice.

- **Jitter**: Jitter is the **Ununiformly scalled IFS**. Jitter kills voice traffic.
- **Drops**: Packets gets dropped (in a voice minor drops are not persivable but not for video).

3.1.5 QoS Protocols

- **Intigrated Services**
  - QoS model in which entire E2E pcket for a packet is **ensured certain minimum QoS charactesistics prior to the packet transmision**.
  - Initial RFC by IETF in 1990s : RFC 1633, 2211, 2212
  - **RSVP** used a primary protocol to setup the path.
    * Requires every node along the path to heed its reservation.
    * Requires every node along the path to keep per-flow state.
  - **Limitation of IS:**
    * In a heterogenious environment over an E2E communication is realistic as different devices perhaps use different protocols (especially for ISP communication)
- **Differentiated Services**
  - Designed to address the challenses of IS
  - RFC : 2474,2597, 2598,3245 and 4594

  - The DS model describes various behavious to be adopted by each complient node (Called Per-Hop Behaviour or **PHB**)
  - No prior reservation but QoS is called when a packet hits am intermediate router.

4 QoS Tools

QoS tools are genrally classified into three high level categories * Classification and Marking * Congestion management framework and * Congestion avoidence

4.1 Classification and Marking

- Traffic must first be segrigated into "Classes"
  - A class of traffic will recive the same QoS treatment. These classes must be pre-determined during the design of the network (Business Decission).
  - Analyse packets to differentiate flows.
- Packets are marked to that the anaylis happens only a limited number of times, usually ingress edge of a network. Markig may happen at the very first stage somewhere down the line. e.g. when some one talks over an IP Phone, the phone marks the voice traffic with a special ToS byte in the IP header. You may also configure an ACL to mark different packets differently.

- Routers perform all the classification and marking in the software level, unlike in a switch it happens in the hardware level. Therefore, Routes can provides DPI however, switch are hardware dependedn. Consequently, QoS in routers are slower than Routers

4.1.1 Policing Shapping and Markdown

- Betwenn ISPs and Customer there are pre-defined contracted rate (Called Commited Information Rate or CIR) which is typically defined in the SLA.
- ISP will police ingress traffic : traffic that is not-conforming is caught by policer and it may drop or markdown (mark it for upstream device but let it go though) those extra packets.
- Customer typically doesn't want any traffic dropped (delay is better than drop). Therefore the customer does **Traffic-Shapping** in the egress/outbound interface that only allows a traffic that complies with the SLA and CIR.

4.1.2 Queuing

- When egreess traffic cannot immediately be trasmitted, it is placed in an egress queue.
- A single egress interface may have multiple, associated queues differentiated by priority.
- QoS features designed for Queuing provide control over which, classified traffic is placed into each of these queues.
- Can also preemptivly drop traffic from within queues to make room for higher-prio traffic (a web traffic can be deprioritzed compared to a Telnet when the queue is about to get congested).

4.1.3 Scheduling

- A schedular orders a packets for processing
- On Routers QoS queuing feature (e.g. WFQ) typically affect queuing and scheduling behaviours
- On a Switch queuing and scheduling can be separate features.
- Traffic Shapping is a Funtion of the scheduling

4.2 Congestion management

The conhestion management feature allows you to controll the congestion by termining the order in which packets are sent out and interface based on priorities assigned to those packets. * Creation of queues * Assignment of packetw to those queues based on the classification and marking * Selectively dropping packets from within queues when those queues reach pre-defined thresholds * Scheduling of the packets ina queueu for transmission

4.3 Congestion avoidance

How to manage a queue so that congestion never happens.

Congestion Management	Congestion avoidence
Control queuing and scheduling of traffic WFQ, CBWFQ, PQ, LLQ, WRR, SRR, Traffic Shapping	preeptively drops traffic to avoid congestion RED, WRED, WTD, Policing

4.4 Modular QoS Command-Line (MQC)

- Before MQC, QoS was configured in the interface level.
- MQC allows feactures that apply several modules such as ACL, Class-Map etc. to perform Classificatio, Policing, etc to be configured independetly and then linked-togather as needed.
- The 3 main components of the MQC is
  - **Class-Map:** Classifies a traffic (Defalut behaviour **match-all**, others are **match-any** and **match-not**). It does not do anything untill it is referecend

```
conf t
    class-map match-all PREC3   ! creates a class-map named PREC3
        match ip precedence 3   ! matches ip packet with precedence 3
    class-map match-all TELNET  ! creates another class-map named TELNET
        match access-group 101  ! matches ACL 101
end
```

  - **Policy-Map:** Defins action to pursue referencing a classmap

```
conf t
    policy-map TO_ISP             ! creates a policy map
        class PREC3               ! refereces the class-map PREC3
            bandwidth percent 30  ! action: limit bw=30mbps
    policy-map TO_CORE            ! another policy map
```

```
        class prec3               ! matches same class
            set dscp af33         ! action: set the DSCP value to ad33
end
```
  - **Service-Policies:** Applies policy map to an interfacs. Service policy is directional.

```
conf t
    int s0/1                            ! selects an interface
        service-policy in|out TO_ISP  ! applies policy map inbound or outbound
    int f0/0
        service-policy in|out To_CORE
end
```

When a packet arrives on an interface, it checks the policy map then the policy map matches the condition based on the class-map.

Important : In a switch, something that sows up in a CLI in class-map match secion does not nessesarily mean that it is supported on the hardware. You may not get a generic error like **"Fearure not Supported"** until your try to apply a a policy map in the service policy. Therefore, make sure the feature list before a purchase is made.

4.5 Hierachical Queuing Framework (HQF)

Unified configuration of Qos: If a set of platform supports exact same features, they must be configured in exact same ways.

- Consistent queuing behaviour applied with common MQC across all main Cisco IoS software releases.
- Common funtionalities for both distributed and non-distributed implementation, providing consitency of QoS feature behaviour across all software-forwarding hardware.

- Some legacy commands are lost. First implemented in 12.04T onwards.

5 Classification and Marking

Classification is a way to tell the routers about the sevearity of a traffic by indentifying distictive features through classes. Most common ways to classify traffic is by * Marking : tagging packets based on its class * Addressig : Segregating traffic based on its source and destination IP addresses * Application Signature : Application aware classification using Deep packet Inspection (DPI)

5.1 L2 Classification

- There is no priority field in Ethernet II/802.3 header.
- Therefore a L2 frame must be encapsulated by either an ISL or 802.1q header to apply prioritiztion. both supports 3 bits
  - ISL : 3 bits Class of Service (CoS) field
  - Dot1q: 3 bits User-Priority field

  - Dot1p: similar to Dot1q but VLAN ID is set to 0 only.
- In a frame-Relay header there is a Discard-Eligible (DE) bits. If a ingress burst exceeds the CIR the extra bits are marked with DE=1, which will not be QoS guranteed. Therefore, the customer marks un-important traffic (HTTP than VoIP) with DE=True to preserve the discard control within them. DE can be done on DLCI number as well to rate-limit based on the outbound PVC.

5.2 L3 Classification

- Due to limited options, Classification is not done in L2 rather in L3. Both IPv4 and IPv6 has a field called **Type of Service (ToS)** and **Traffic Class (TC)** respectively to manage the prioritization.
- The **Original ToS Byte** : `__P P P D T R 0 0__` as per RFC791
  - P : Precedence
  - D : Delay (0 = normal , 1 = high) : put on a path with as low/high delay
  - T : Throughput (0 = normal , 1 = high) : put on a path with as low/high thoughput
  - R : Reliability (0 = normal , 1 = high) : put on a path with as low/high reliablity
  - Preference combination
    * 000 : Routine (default to normal data)
    * 001 : Priority
    * 010 : Immidiate
    * 011 : Flash
    * 100 : Flash override
    * 101 : CRITIC/ECP (default for VoIP)
    * 110 : Internet-Control (Default for Routing protocol)
    * 111 : Network Control
- However very fiew applications make use the bits other than precedence bits. therefore the rest of the 5 bits are wasted and all traffic must be classified into one of the 8 classes, which is not a feasible solution.