Quality of Service (QoS)

Networking

Pierre-Philipp Braun <pbraun@nethence.com>

Table of contents

Remember about 4 resource types (VMM lecture)

What do if a CPU or RAM bus gets saturated?…

==> resp. cpu schedulers & NUMA

Remember about switching fabrics from HAPPY-L2

What do if all tubes get saturated?…

==> you have two choices

1. upgrade your network devices' hardware —or— in case gw is the bottleneck, upgrade your connection
2. configure and enable QoS

ideally both

another way to look at it

what if your roommate is streaming and doing torrent non-stop,

while you're a gamer and need a good ping?…

even if you upgrade connection, your roomate will just download more

==> QoS is the only choice here

What is QoS

- mainly means prioritize types of traffic
- also includes notion of tracing network reliability

Goals

- latency reduction (like a ping response time)
- jitter reduction (audio flickering…)
- packet loss prevension

What would you like to prioritize?…

What use-cases?…

- VoIP
- video calls / conference calls
- TV / live streaming
- gaming

which one of those is easier to isolate and prioritize?…

==> mostly VoIP as we can dedicate a VLAN for it

Where to put the tag?…

On what devices to configure QoS?…

==> switches / vswitches & routers

notice difference between the places where

- you inject tag (conditioners)
- you interpret tag

Ways to QoS

Coarse-grained

- Type of Service (ToS) – *just a tag*
- DiffServ / DSCP – *a longer tag*

Fine-grained

- IntServ – RSVP (path assessment and bandwidth allocation)

// Questions on QoS essentials?

Class of Service (CoS)

the easiest method around

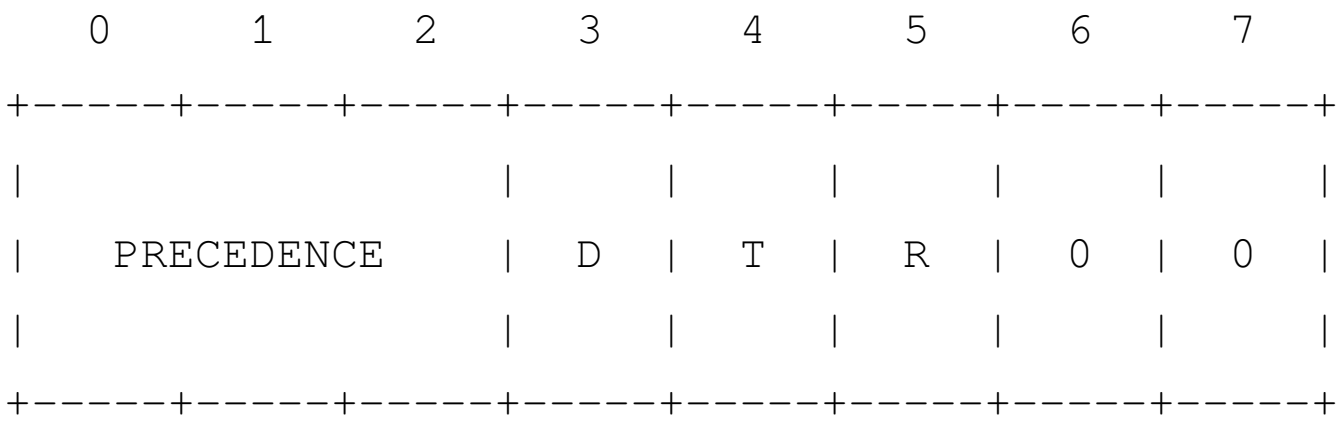
just a layer 2 tag

- 3-bit in Ethernet frame header
- requires Dot1Q (VLAN) tag (P802.1p took some time to be merged)
- CS0 - CS7
- trust mode – receive CoS values from another switch
- –or– rewrite value anyway (like ISPs do)

Type of Service (ToS)

```
IPv4 -- one byte for that purpose
IPv6 -- Traffic Class field
```

Service mappings



- 3-bits – IP Precedence – *never used*
- 3-bits – **DTR: Low Delay - High Throughput - High Reliability**
- 1-bit – lowcost – *breaks DiffServ's ECN*
- least-significant bit – *must be zero*

PRECEDENCE

111 - Network Control

110 - Internetwork Control

101 - CRITIC/ECP

100 - Flash Override

011 - Flash

010 - Immediate

001 - Priority

000 - Routine

DTR

single-bit flip examples from Type of Service RFC

max 2 out of 3-4 bits to flip

```
1000   --   minimize delay
0100   --   maximize throughput
0010   --   maximize reliability
0001   --   minimize monetary cost
0000   --   normal service
```

Where is it best to define the tags?… (apps, systems, network devices?)

- Apps / systems *as long as switch does not override the tag*
- —or— switches *and eventually by means of CoS*
- ToS domain is not endless…

same goes for DSCP codes (DiffServ)

ToS / DSCP capable products

FOSS

- Linux Netfilter — tc
- Linux Netfilter — MANGLE table directly?
- Linux eBPF
- NetBSD ALTQ
- OpenBSD PF/ALTQ
- another BSD system
- Cumulus Linux, VyOS, …

The competition

- Cisco
- Juniper
- MKT PCQ vs. other means?

LAB // QoS w/ Linux – play ts & PRIQ vs. HTB

LAB // QoS w/ Linux – play with mangle table directly? is that possible?

// Questions on ToS?

Differentiated services

aka DiffServ

- also compatible with both IPv4 and IPv6
- Class Selectors
- Explicit Congestion Notification (ECN)
- DiffServ domain is not endless…

DiffServ vs. ToS

taking over ToS's unused IP Precedence and DTR (tos field)

- DSCP – now using 6-bit! (DS Field)
- DSCP – backwards compatible – 8 Class Selectors reserved
- ECN – skips the lowcost ToS bit and enabling the last bit
- ECN – conflicts with ToS

Per-hop behaviors

- Default Forwarding (DF) PHB – best effort
- Expedited Forwarding (EF) PHB – low-loss & low-latency
- Assured Forwarding (AF) PHB – assurance of delivery
- Class Selector PHBs – ToS compatible

DSCP values description

```
...
CD4 32  real-time interactive
AF43    38  multimedia conferencing
AF42    36  multimedia conferencing
AF41    34  multimedia conferencing
CS5 40  signaling
EF  46  telephony
CS6 48  network control
```

Assured Forwarding PHBs

a very specific case (makes me think of RSVP yet again)

```
10,12,14    Class 1
18,20,22    Class 2
26,28,30    Class 3
34,36,38    Class 4
```

- reliability
- subscribed rates (bandwidth allocation)

LAB // AF PHBs would be happy w/o TCP e.g. got « reliable » UDP as a result? Possibly same orientation for RSVP.

Class Selector PHBs

backward-compatible values

DS Field (Dec)	Description
0	best effort
8,10,12,14	priority
16,18,20,22	immediate
24,26,28,30	(voip signaling)
32,34,36,38	?
40,46	(voip stream)
48	internetwork control
56	network control

CoS - TOS/DSCP mapping?

l2 vs l3 correspondance?

CS0	000000	0
CS1	001000	8
CS2	010000	16
CS3	011000	24
CS4	100000	32
CS5	101000	40
CS6	110000	48
CS7	111000	56

LAB // PoC a network architecture with L2/CoS –> L3/DiffServ convertion

What about WhatsApp and such?…

==> nope, otherwise by ingress `src-ip` & egress `dst-ip`

But the exact IP range is not known publicly.

What about the public network?…

Would ToS or DSCP tags be honored?…

==> not sure, needs to be tested, but here are some hints

RFC 1812 (Jun 1995) - Requirements for IP Version 4 Routers

- many occurences of TOS
- SHOULD behave accordingly
- MUST retain the tag

also mentioned in 5.2.4.3 Next Hop Address

Discard route if

```
route.tos != ip.tos
```

LAB // what about diffserv over the internet? Might there be some bits of the public network with end-user defined QoS?

LAB // PoC TOS over the public network and see if the tag remains. Do the tags remain over the internet, from one end to another? If so, you might have QoS from one enterprise network to another. Can tags pass through to some other enterprise network without IPSEC?

LAB // what about IPSEC setup, how much sense does it make to enable it across two zones?

// Questions on DiffServ?

Traffic Schedulers

some kind of a firewall

- Traffic conditioners
- Protected queues — *just an enhancement*
- Queue disciplines — *full-blown algorithms*

Congestion

- Worst case scenario – *some kind of a DoS*
- First TCP implementations re-sent too fast
- Fixed by slowing down the rate of retries until timeout
- BSD implementation became a reference
- Even worse scenario – *waves of congestion vs. idling when everybody has same implementation*

Traffic conditioners

aka profile meters for RIO

- push the tag in IP headers
- works on **inbound traffic**
- RIO – tag packets IN or OUT
- can also be used to drop packets before-hand (before it enters the queue)

What possible methods to identify various traffic types to tag?…

==> LAB

mac addresses?

vlan (CoS)

ip.src & ip.dst?

l4.dst port?

SSL SNI?

DPI -- HTTP, FTP, SMTP, torrent (qos2 lecture)

DPI & SSL -- HTTPS, ... (lbs & proxies lecture)

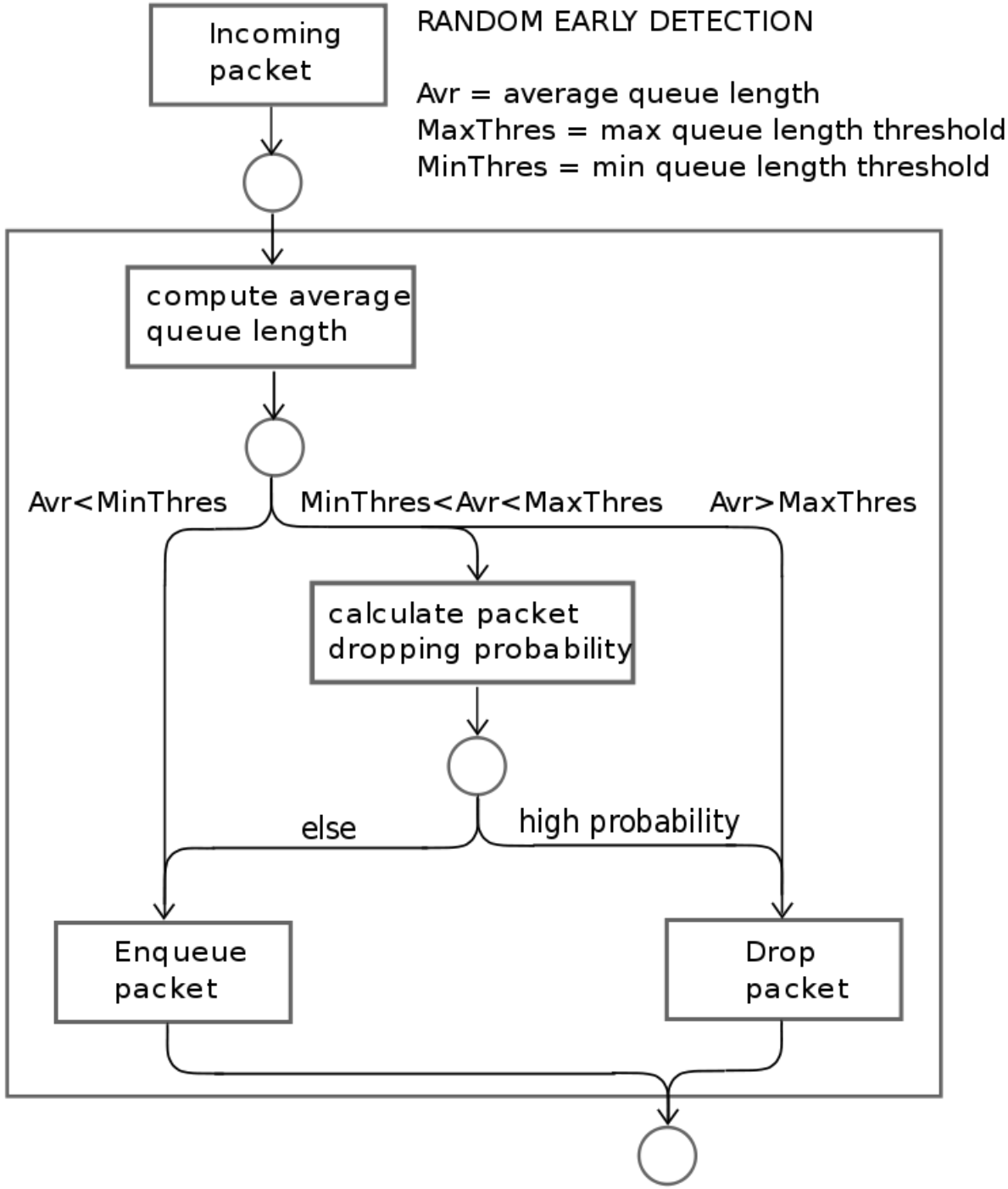
// Are we clear on the conditioners?

Protected queues

no prioritazation just yet

Algorithms to better handle the queue and avoid congestion

- RED & ECN
- RIO & ECN


Incoming packet
RANDOM EARLY DETECTION
Avr = average queue length
MaxThres = max queue length threshold
MinThres = min queue length threshold
compute average queue length
Avr<MinThres
MinThres<Avr<MaxThres
Avr>MaxThres
calculate packet dropping probability
else
high probability
Enqueue packet
Drop packet

Random Early Detection (RED)

some kind of a buffer manager

- *implicit* congestion notification
- evaluates average queue size and probability for packets to be dropped
- dropping/marking depending on average queue length
- ECN compatible – drops **or marks**

RED with In/Out bit (RIO)

welcome back to some kind of RSVP…

- got diffserv edges all around our domain
- IN-PROFILE – contracted QoS
- OUT-PROFILE – normal packets
- good for DiffServ's Assured Forwarding PHBs
- ECN compatible – drops **or marks**

// Are we clear on enhanced queues?

Queue disciplines

- works on **outbound traffic**
- reads the tag and applies the priority accordingly
- drops the less-prioritized packets under congestion

Loads of choices

```
(red, rio)
fifoq, priq
blue, cbq, hfsc, jobs, wfq
```

First-In First-Out Queueing (FIFOQ)

some kind of an ECB (crypto)…

Priority Queueing (PRIQ)

the queue we were expecting

- higher priority gets served first
- up to 16 priority classes

LAB // what if I got more tag differentiators than 16 classes?

BLUE & SFB

some easier kind of RED

- no tuning required – adaptive learning based on packet drops
- ECN compatible – drops **or marks**

Stochastic Fair Blue (SFB)

- calculates probabilities per traffic flow
- –> fair share for the flows as long as there are no hash collisions
- bloom filter faster than hash table, as with Stochastic Fairness Queuing (SFQ)

Class Based Queueing (CBQ)

simple and fair split

- equal shares of the bandwidth among traffic classes
- hierarchical

LAB // does a class bandwidth overlap the other if idling by default?

Per Connection Queue (PCQ)

brutal caps (does NOT overlap when idling)

- split **per connection**
- example: 100Mbit/s per user on a 1Gbit/s connection
- hierarchical
- no priority?

Hierarchical Fair Service Curve (HFSC)

- based on CBQ
- fairness for traffic classes
- service-curve-based scheduler??
- allocation // delay decoupled
- hierarchies (of classes?) on top of that…

Joint Buffer Management and Scheduling (JoBS)

- loss and delay differentiation independently at each node
- (not network capacity / bandwidth)

Weighted Fair Queueing (WFQ)

- round-robin against a set of queues
- weight –> different proportion of RR
- hash the flow –> map it to a set of queues

// Are we clear on queue algorithms?

Ehm, which one of those algos need some manual queue definitions?…

==> those need a class definition

`PRIQ`

`CBQ`

`HFSC`

`JoBS`

- Need to define packet scheduling classes manually
- Additional setting: clear DSCP

Does not need any tag?

`WFQ`

// Are we clear on the queue disciplines?

// Questions on traffic schedulers?