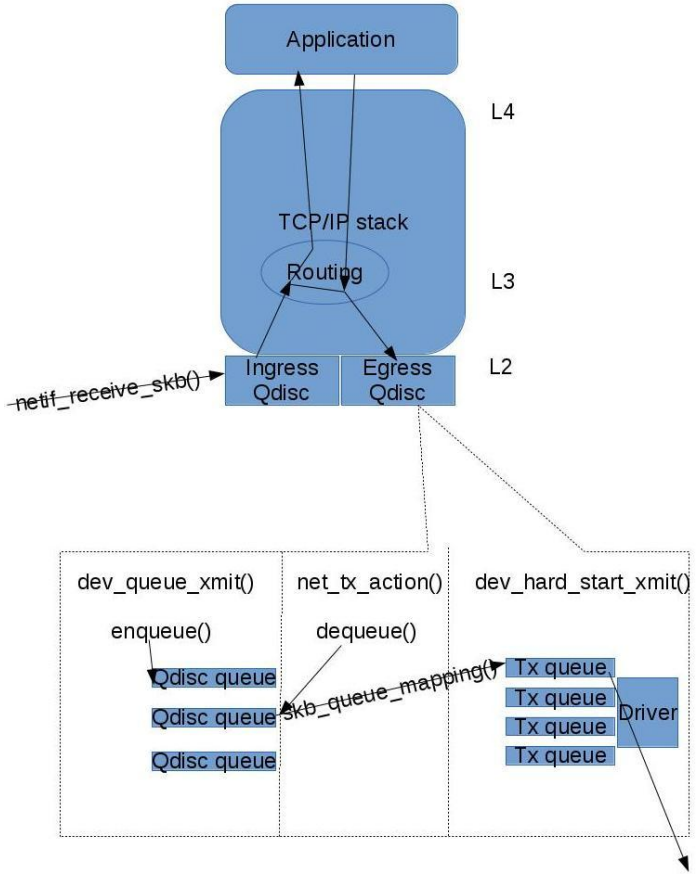
Linux Traffic Control

Cong Wang
Software Engineer
Twitter, Inc.

Network stack



Overview

- Qdisc: how to queue the packets
- Class: tied with qdiscs to form a hierarchy
- Filter: how to classify or filter the packets
- Action: how to deal with the matched packets

```
for_each_packet(pkt, Qdisc):
    for_each_filter(filter, Qdisc):
        if filter(pkt):
            classify(pkt)
            for_each_action(act, filter):
                act(pkt)
```

Source code

- Kernel source code:

net/sched/sch_*.c net/sched/cls_*c net/sched/act_*.c

- iproute2 source code:

tc/q_*c tc/f_*.c tc/m_*.c

TC Filter

- As known as classifier
- Attached to a Qdisc
- The rule to match a packet
- Need qdisc support
- Protocol, priority, handle

Available filters

- cls_u32: 32-bit matching
- cls_basic: ematch
- cls_cgroup: cgroup classification
- cls_bpf: using Berkeley Packet Filter syntax
- cls_fw: using skb marks

TC Action

- Was police
- Attached to a filter
- The action taken after a packet is matched
- Bind or shared
- Index

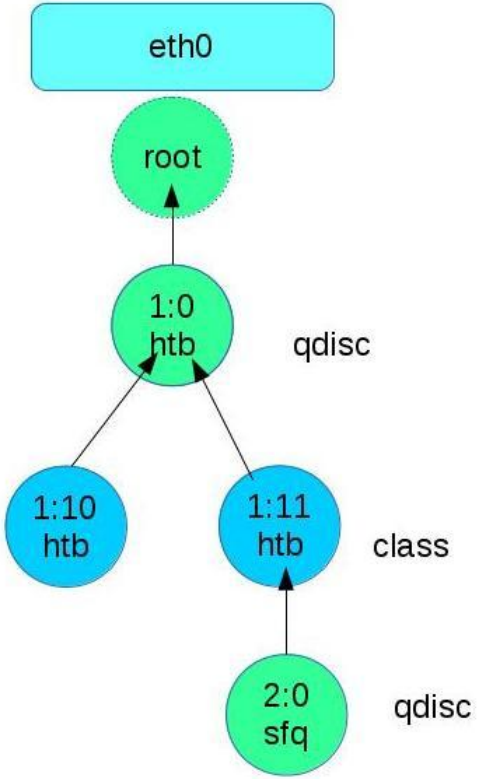
Available actions

- act_mirred: mirror and redirect packets
- act_nat: stateless NAT
- act_police: policing
- act_pedit/act_skbedit: edit packets or skbuff
- act_csum: checksum packets

TC Qdisc

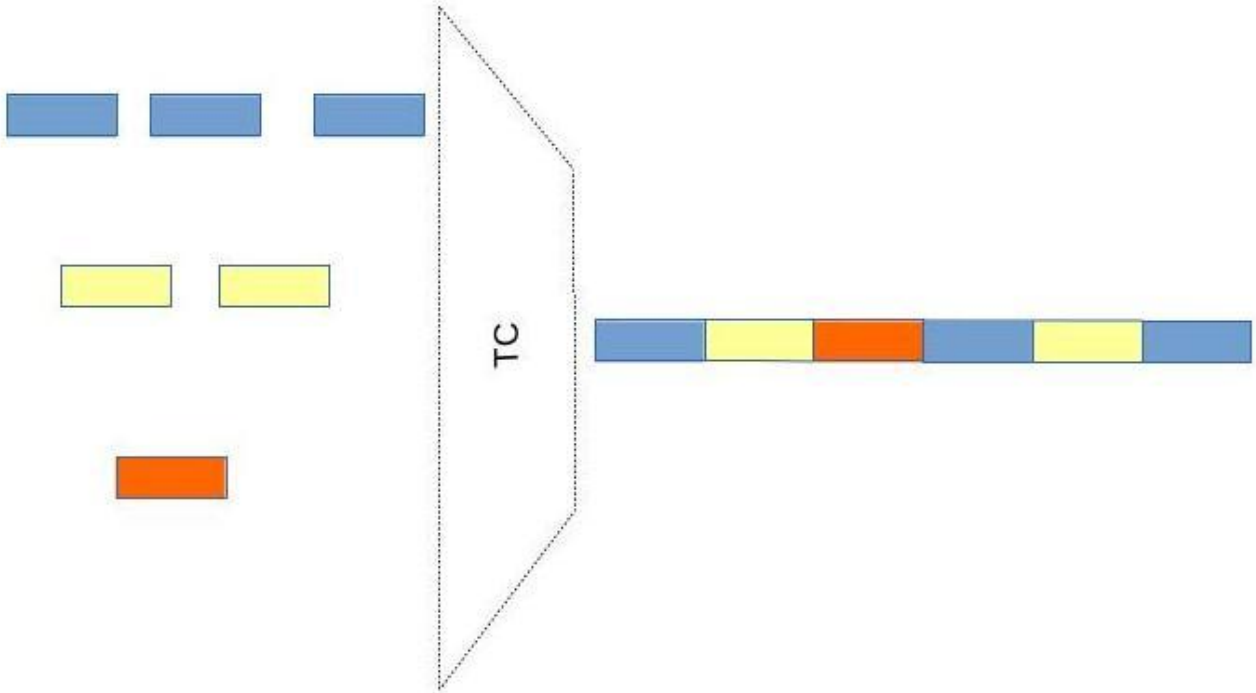
- Attached to a network interface
- Can be organized hierarchically with classes
- Has a unique handle on each interface
- Almost all qdiscs are for egress
- Ingress is a special case

Class



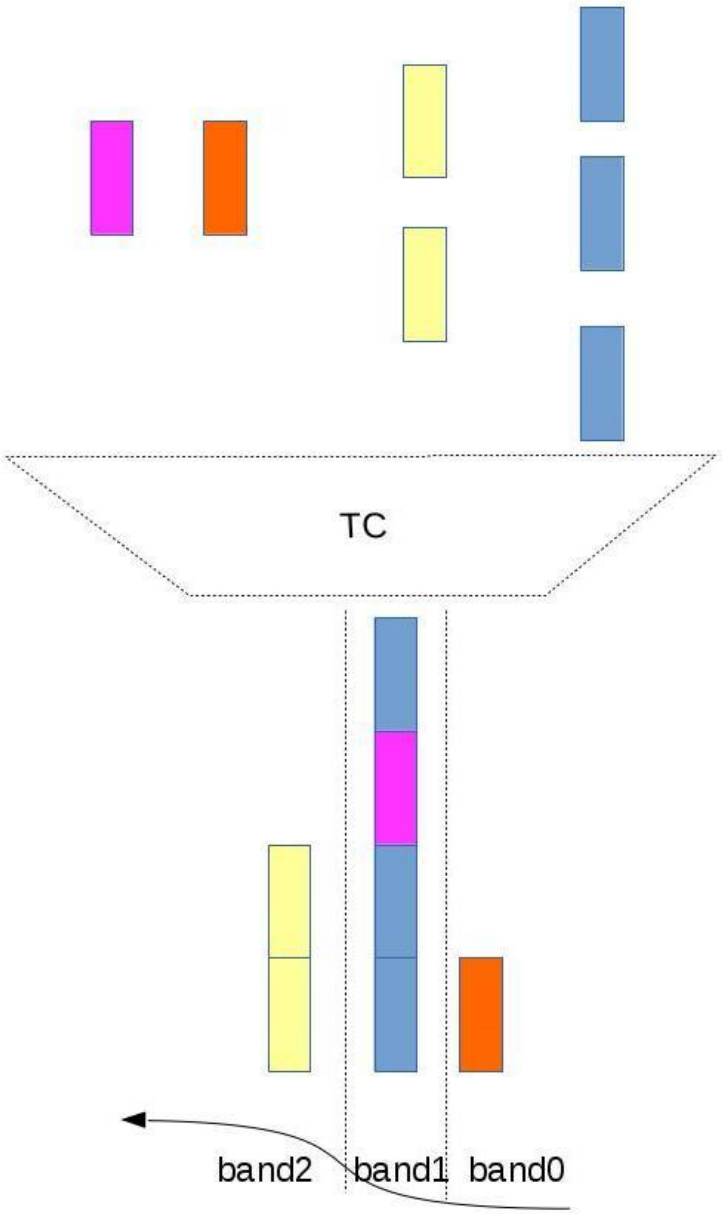
FIFO

- bfifo, pfifo, pfifo_head_drop
- Single queue, simple, fast
- No flow dissection, no fairness
- Either tail or head drop


TC

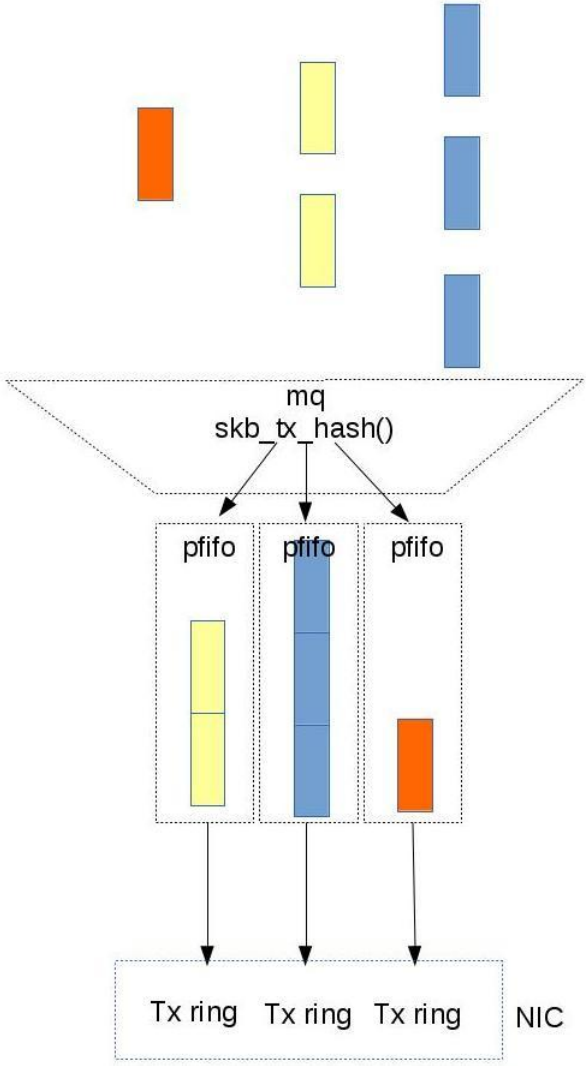
Priority queueing

- pfifo_fast, prio
- Multiple queues
- Serve higher priority queue first
- Use TOS field to prioritize packets


TC
band2
band1
band0

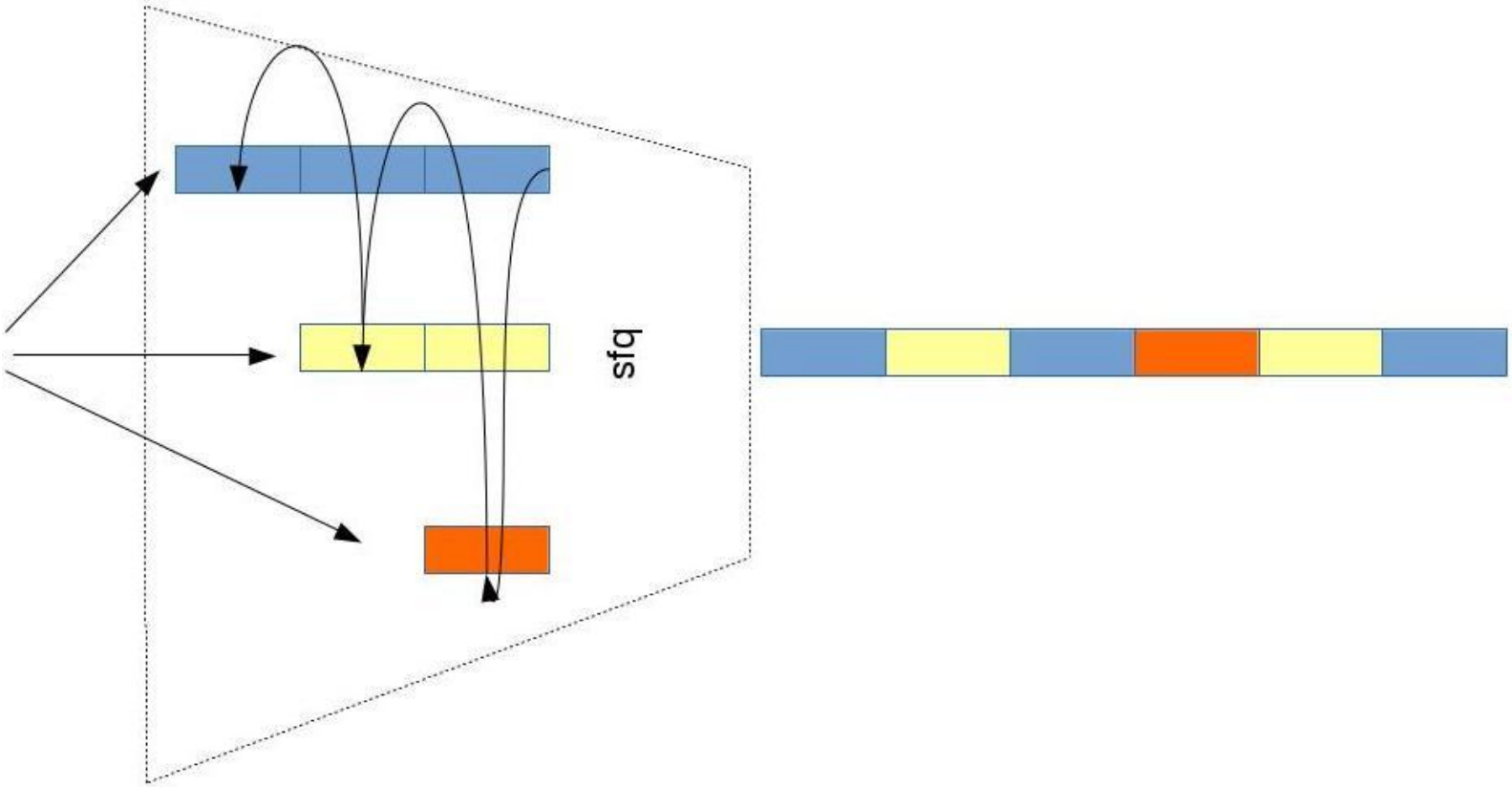
Multiqueue

- mq, multiq
- For multiple hardware TX queues
- Queue mapping with hash, priority or by classifier
- Combine with priority: mq_prio


mq
skb_tx_hash()
pfifo
pfifo
pfifo
Tx ring
Tx ring
Tx ring
NIC

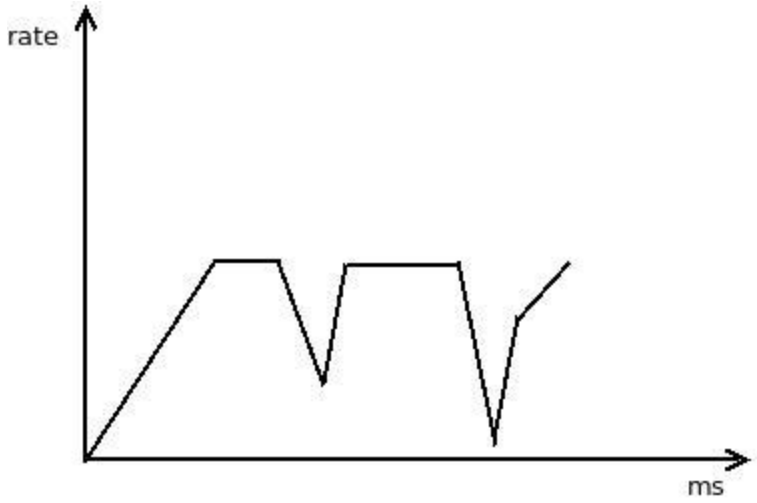
Fair queueing

- Each flow fairly sharing the link
- Round robin, no weights: sfq
- Deficit round robin: drr
- Max-min fairness
- Socket flow dissection + pacing: fq

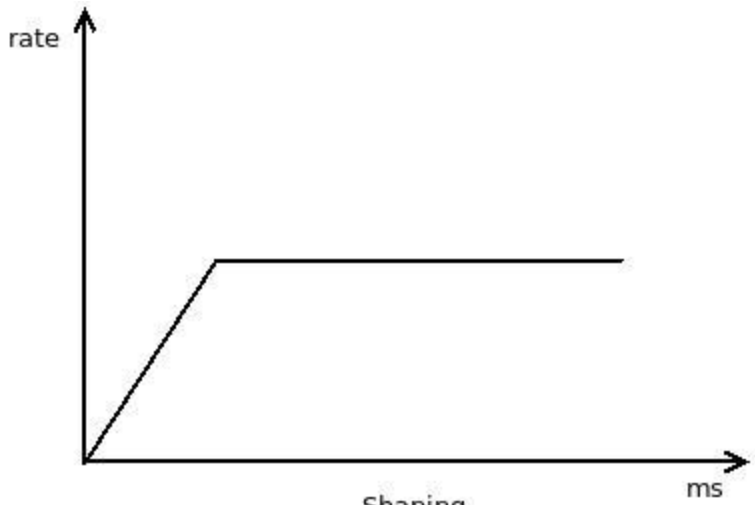

sfq

Traffic shaping

- Shaping buffers and delays packets
- Policing mostly drops packets
- Buffer means latency
- cbq is complex and hard to understand



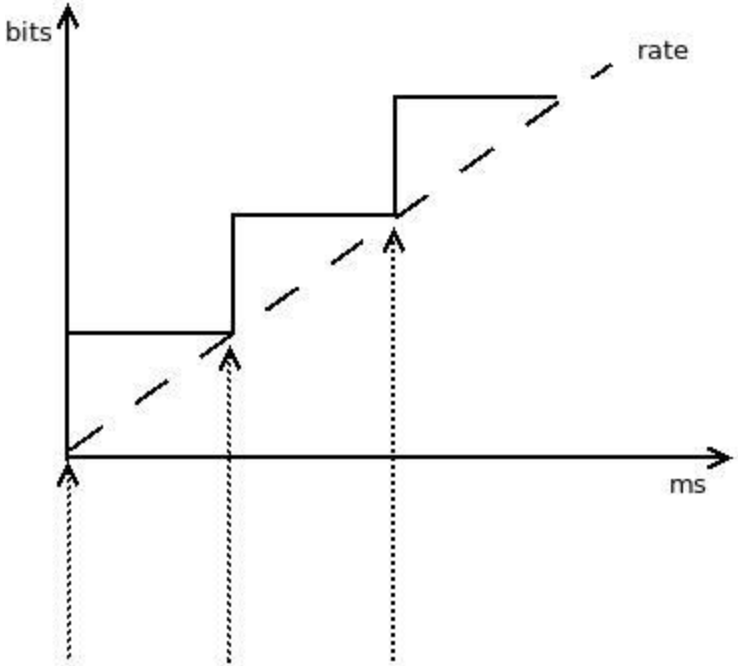
Policing



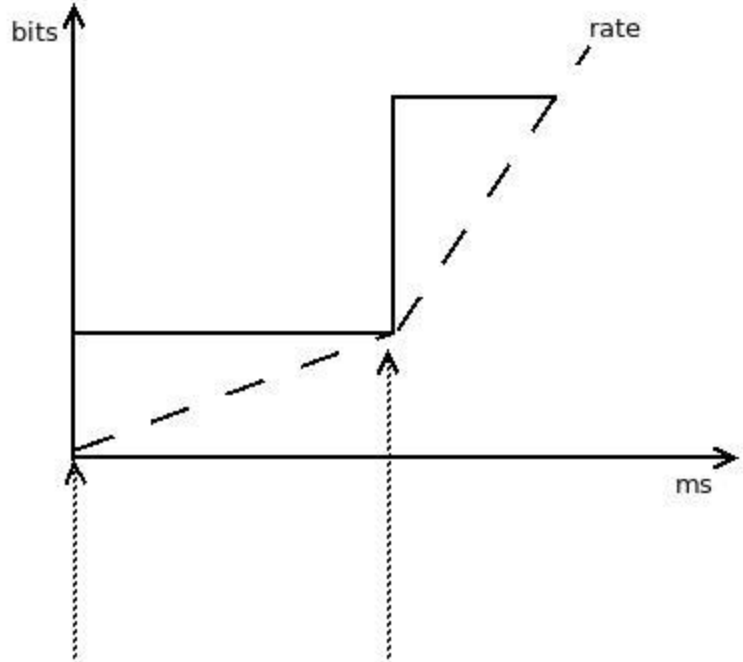
Shaping

Token Bucket Filter

- One token one bit
- Bucket fills up with tokens at a continuous rate
- Send only when enough tokens are in bucket
- Unused tokens are accumulated, bursty
- Still tail drop
- Big packets could block smaller ones



pacing



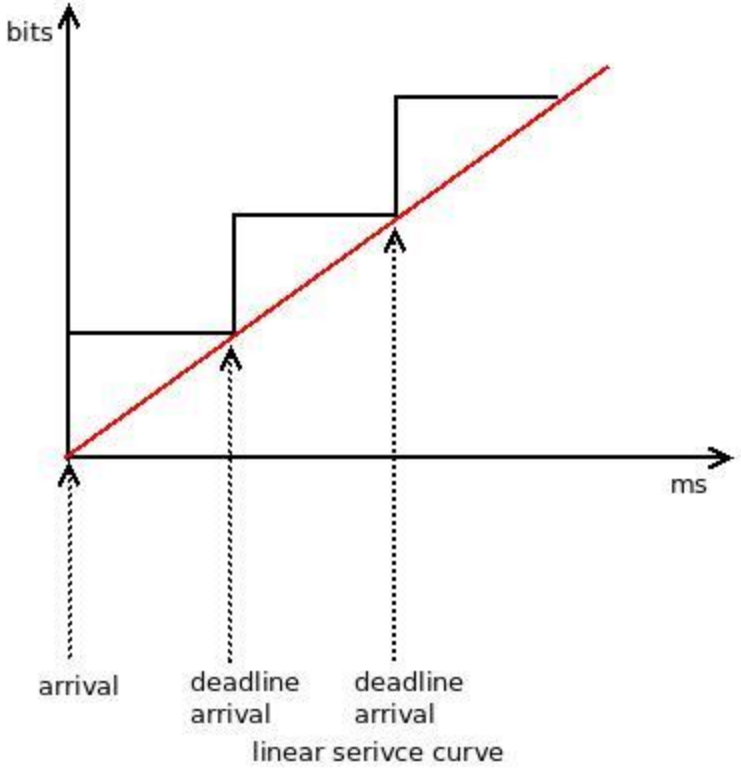
burst

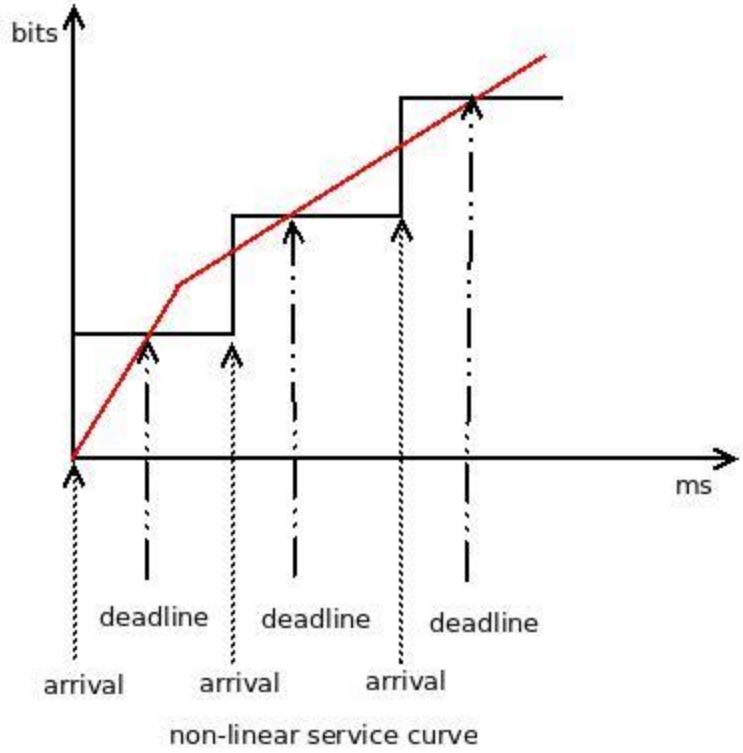
Hierarchical Token Bucket

- Basically classful TBF
- Allow link sharing
- Predetermined bandwidth
- Not easy to control queue limit, latency!

Hierarchical Fair Service Curve

- Proportional distribution of bandwidth
- Leaf: real-time and link-sharing
- Inner-class: link-sharing
- Allow a higher rate for real-time guarantee
- Non-linear service curves decouple delay and bandwidth allocation


bits
ms
arrival
deadline
arrival
deadline
arrival
linear serivce curve


bits
ms
deadline
deadline
deadline
arrival
arrival
arrival
non-linear service curve

Active Queue Management

- Bufferbloat, it’s the latency!
- Manage the latency
- Tail drop hurts TCP (TCP tail loss probe)
- Modern AQM qdiscs are parameterless
- RED, CHOKe, codel, pie, hhf

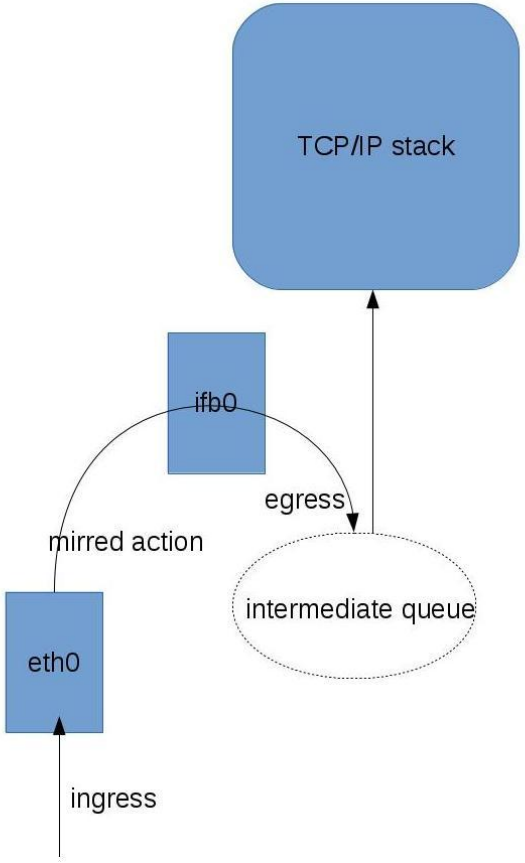
Controlled Delay

- Measure latency directly with time stamps
- Distinguish good queue and bad queue
- Good queue absorbs bursts
- Drop faster when bad queue stays longer
- Head drop

Ingress Traffic Control

- Only ingress qdisc is available
- Classless, only filtering
- Only policing, shaping is essentially hard
- Needs transport layer support: TCP or RSVP

Hack: IFB device



TODO

- Lockless ingress qdisc (WIP)
- TCP rate limiting
- Ingress traffic shaping