Quality of Service (QoS)

Outline

- Quality of Service
- Integrated Services (RSVP)
- Differentiated Services

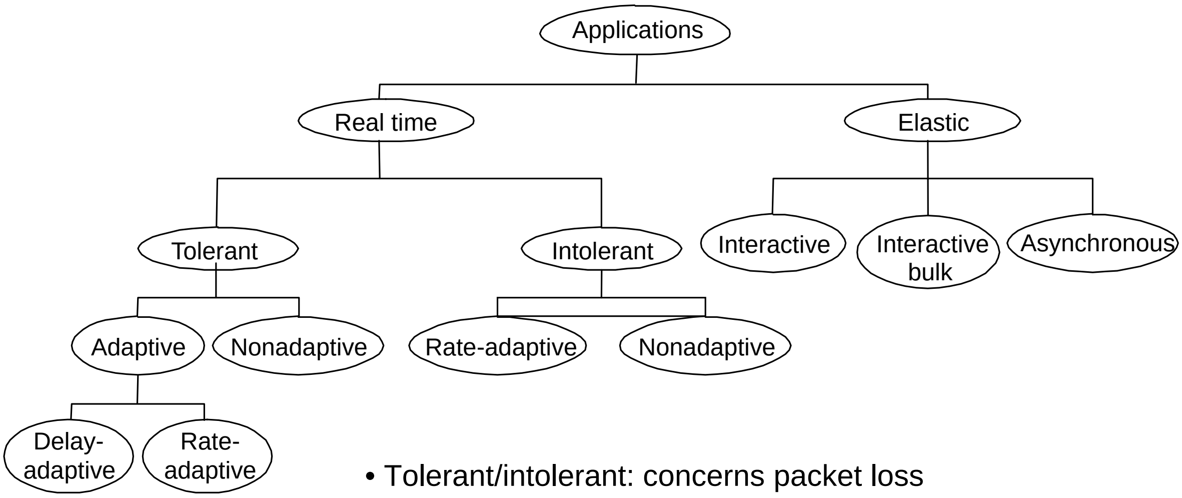
Need for QoS

- Basic assumption: bandwidth is **scarce** also in the future
  - if we can install enough capacity that network can never be overloaded, everyone gets premium service all the time ⇒ best effort service is enough
  - if bandwidth is scarce, mechanisms are needed to control/isolate different traffic types (need a new service model to support QoS)
- Need to understand requirements from different (new) applications
  - traditional data does not (necessarily) need QoS
  - **multimedia** has different/varying requirements on the network
    - need for high-bandwidth links (improved coding helps)
    - **timeliness** of delivery, called real-time application
      - ex. voice, video, industrial control
  - multimedia needs assurance from the network that data arrives on time
  - if bandwidth is scarce, data and multimedia traffic interfere with each other
- Current state of Internet
  - best-effort model: makes no guarantees, leaves cleanup operation to edges
- QoS network = network that can provide different levels of service

Application requirements

- Roughly, two types: real-time and non-real-time
- Non-real-time:
  - “traditional data”
  - applications like Telnet, FTP, email, web browsing
  - relies on lossless delivery (through retransmissions)
  - can work without guarantees of timely delivery of data
  - also called **elastic:** applications adjust to available capacity (TCP)
  - since applications are elastic, no need for QoS (just add more capacity)
- Real-time:
  - telephone, video conference, streaming audio/video
  - requires “deliver on time” assurances
    - large delay prohibits for example phone conversation
    - variations in delay can be smoothened by using application level buffers, but overall delay increases
  - may also need assurances regarding bandwidth (throughput) and loss
  - assurance must come from inside the network ⇒ need QoS mechanisms

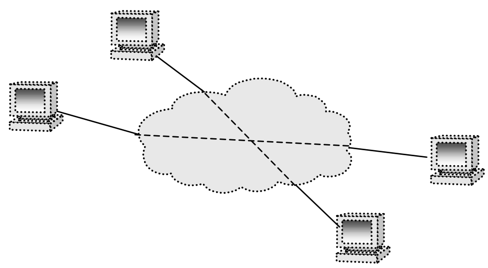
Taxonomy of applications

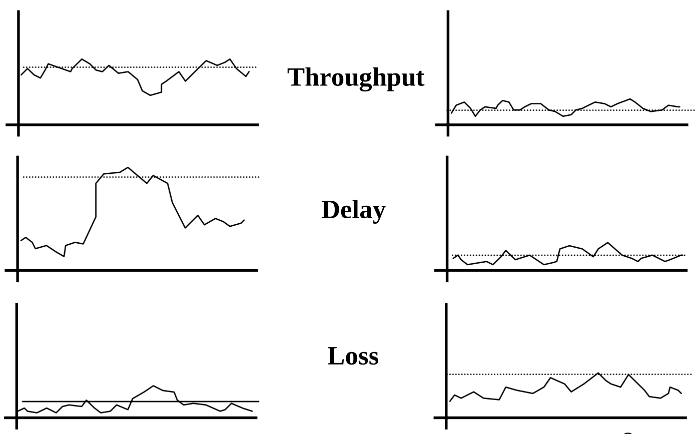


- Tolerant/intolerant: concerns packet loss
- Adaptive/nonadaptive: concerns delay variations
- Delay adaptive: application can adjust amount of buffering
- Rate adaptive: e.g., audio codec can change its bit rate

Internet QoS

- By adding Quality of Service (QoS), we are aiming to provide service differentiation to users
  - with respect to bandwidth, delay and loss characteristics
- Differentiation can be based on different criteria
  - Usage
  - Money
  - Status





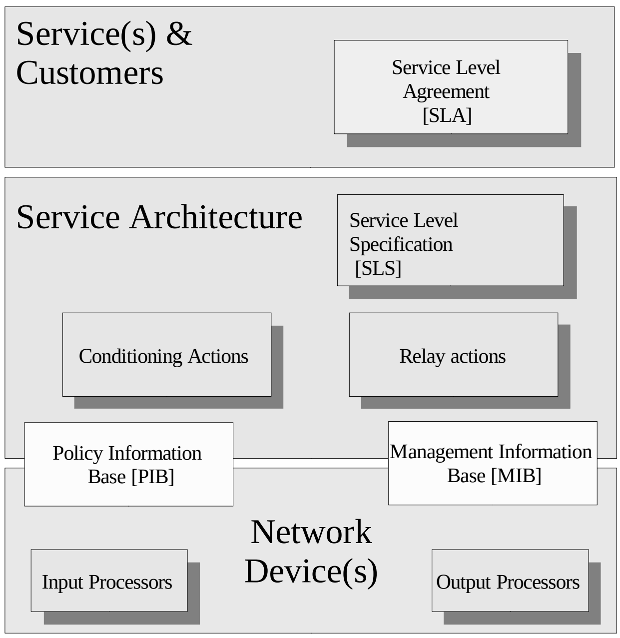
Slide material from Marko Luoma

Terminology

- **Connection**: a dynamically formed reservation of network resources for a period of time.
  - Connection requires a state to be formed inside the network
  - State is a filter defining packets which belong into particular connection and required reservation attributes
- **Flow**: formed from arbitrary packets which fall within predefined filter and temporal behaviour.
  - Packets from one source to the same destination arrive to the investigation point with interarrival time less than t seconds.
  - Local knowledge, no state stored for particular flow
- **Aggregate**: a group of flows which have same forwarding characteristics and share link resources.
- **Class**: a group of connections which share same forwarding characteristics.

Approaches to QoS support

- A complete QoS architecture comprises several layers
  - here we look at basic mechanisms in the “lower layers” (no customer/user relation)
- Fine-grained approach
  - provide QoS to individual applications or flows
  - Integrated Services, RSVP (ATM)
- Coarse-grained approach
  - provide QoS to large classes of data or aggregated traffic
  - Differentiated Services



Outline

- Quality of Service
- Integrated Services (RSVP)
- Differentiated Services

Elements of Integrated Services

- Different functional components of the IntServ architecture
  - Service classes
  - Flowspecs
  - Admission control
  - Reservation protocol
  - Packet classifying and scheduling

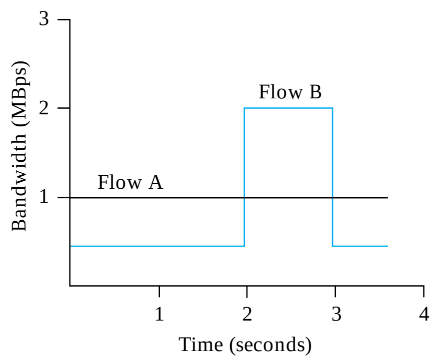
- $\Rightarrow$ Main question: does it scale?

Integrated Services (cont.)

- Service classes
  - guaranteed service:
    - for delay intolerant application, packets never arrive late
    - maximum delay guaranteed
  - controlled load:
    - for adaptive applications that run well if network is not heavily loaded
    - emulate lightly loaded network, even though the network as a whole may be heavily loaded, i.e., use queuing mechanisms to isolate controlled load traffic
    - use admission control to limit controlled load traffic
- Mechanisms
  - telling the network about service requirements, characterizing the data (flowspec) , admission control (can we provide requested service to given data) , signaling / resource reservation (network routers exchange information), packet scheduling (actions of routers to meet the requirements)

Flowspecs

- Two parts: TSpec and RSpec
  - TSpec: flow's traffic characteristics, information about bandwidth used by the flow
  - RSpec: service requested from network (ex. request for controlled load, or delay bound)
- Token bucket: describes the bandwidth characteristics of a source
  - parameters: token rate r and a bucket depth B
  - idea: To send a byte, you need a token. To send packet of length n, you need n tokens. At start no tokens, tokens accumulate at rate r - but never more than B tokens. Whenever you have enough tokens you can spend them in sending data.
  - figure: two flows with the same mean, but different token bucket

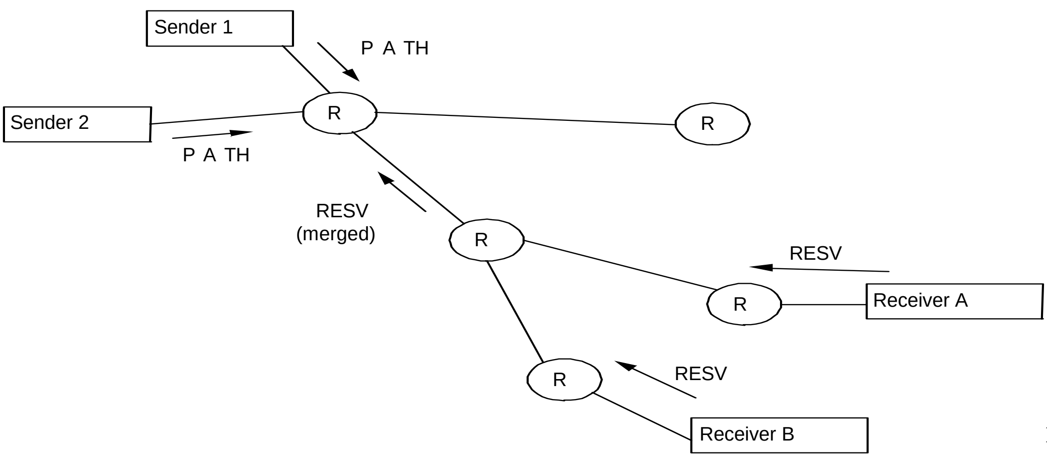


Admission and reservation

- Admission control
  - per flow decision to admit a new flow or not
  - given TSpec and RSpec decide if desired service can be provided with available resources - a difficult task
  - if a new flow is admitted, old flows may not get worse service than what it has requested earlier
  - different from policing = function applied on per-packet basis to make sure that flow conforms to TSpec
- Resource Reservation Protocol (RSVP)
  - key assumption: should not detract from the robustness of today's Internet where routers may crash, links may go down but the end-to-end connectivity survives
  - uses a soft state in routers - soft state need not be explicitly deleted, it times out if not refreshed periodically (30 s refreshment period in IntServ)
  - aims to support also multicast

Path reservation

- Receiver-oriented approach - receiver needs to know sender's TSpec and the path
- Sender sends a message with TSpec to receiver, gets reverse path as a bonus: source transmits PATH, receiver responds with RESV
- If link fails, routing creates a new PATH message and receiver sends RESV along new path, reservations on old path time out and are released $\Rightarrow$ adaptation to changes in topology



Packet classification and scheduling

- Packet classification: associate each packet with appropriate reservation
  - mapping from flow-specific information in the packet header to a single class identifier that determines how the packet is handled in the queue
- Packet scheduling: manage packets in the queues to that they get the service that has been requested
  - not a trivial task...

Scalability problem of Integrated Services

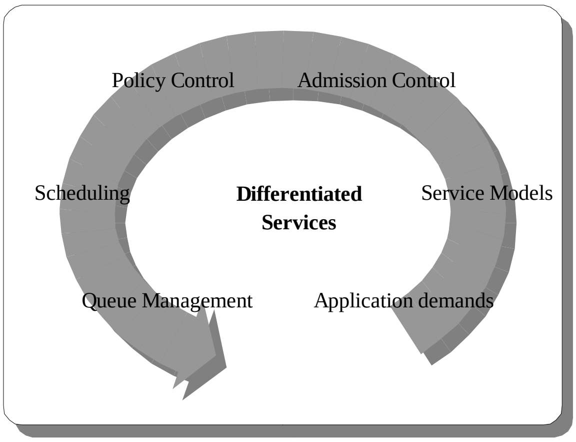
- Integrated services and RSVP enhance best-effort service model, but ISPs find that it is not the right model
- Violates the fundamental design goal of IP: scalability
  - as Internet grows, routers just need to keep up (move bits faster and deal with larger routing tables)
  - with RSVP every flow through router may have a reservation
  - ex. 2.5 Gbps full of 64-Kbps audio streams $\Rightarrow$ $2.5 \times 10^9 / 64 \times 10^3 = 39\ 000$ flows
  - each reservation needs a state that is stored in memory and refreshed periodically
- Need for a solution that does not require so much “per-flow” work

Outline

- Quality of Service
- Integrated Services (RSVP)
- Differentiated Services

Differentiated Services overview

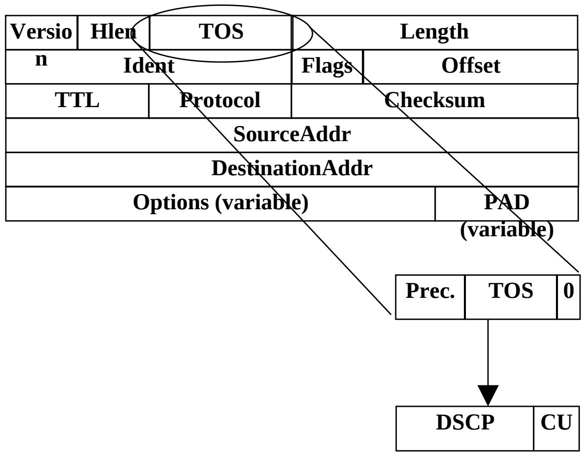
- Physically, nothing more than Best Effort (well, sort of …)
- Logically, number of parallel Best Effort networks
- Packet is destined to one of the parallel networks
  - Packet per packet processed quality of service
  - Connectionless architecture is still preserved
- Each parallel network uses same routing topology (not neccesarily)



Slide material from Marko Luoma

Differentiated Services overview (cont)

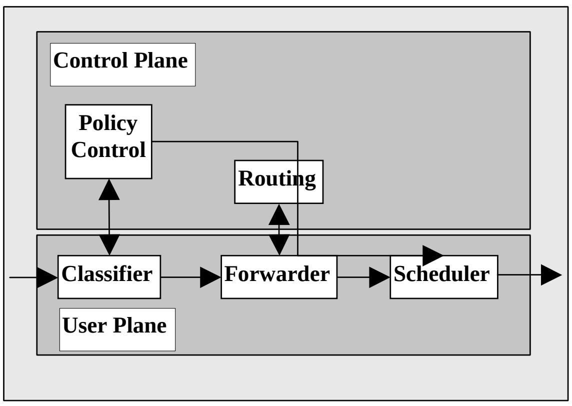
- Identification of which parallel best effort network packet is destined, is coded in each packet
  - IPv4 ToS field is reformatted
  - 6 bits reserved for indicating traffic classes, DSCP (Differentiated Services Code Points) bits
- Questions:
  - Who sets the premium bit, and under what circumstances?
  - What does the router do differently when it sees a packet with the bit set?



Slide material from Marko Luoma

DiffServ router

- Packets are forwarded based on the destination address and class information (DSCP of the packet)
  - scheduling and queueing is done based on the class information
- DiffServ router has two additional elements in datapath compared to basic Best Effort router:
  - Traffic conditioner (TC) (Classifier in figure)
  - Per hop behavior (PHB) (Scheduler in figure)
- Control plane of DiffServ router has one extra element, i.e., policy controller, which is responsible for internal management and configuration of TC and PHB



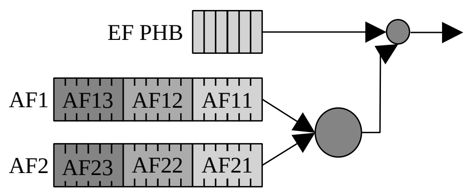
Slide material from Marko Luoma

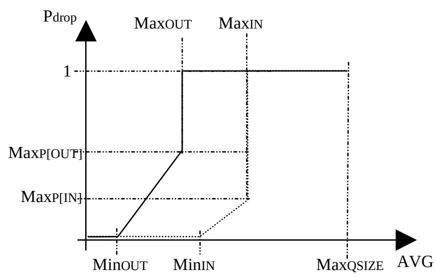
DiffServ conditioner

- Traffic Conditioner consists of
  - Classifiers
    - responsible for logical separation of packet streams
    - inspects DSCP bits from packets
  - Meters
    - responsible for rate metering of logical streams
    - done by using for example token buckets
  - Markers
    - responsible for actions based on metering results and predefined thresholds
    - non-conformant packets may be dropped or marked

DiffServ PHB

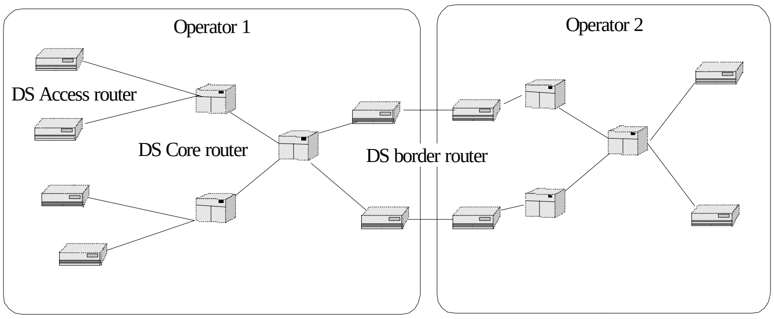
- PHB = Per Hop Behavior
- PHB is a block containing queue management methods required to implement desired service (locally)
  - queues
  - queue space management algorithms
  - schedulers
- PHB defines forwarding actions in a router - no end-to-end specification





DiffServ network

- Workload in DiffServ is divided between two inherently different types of routers
  - edge routes
  - core routers
- Edge routers are on the domain edge and interface
  - customers
  - other ISPs
- Edge routers are responsible for conditioning actions which eventually determine logical network where packet is to be forwarded
  - edge routers set DSCP bits based on service contracts (SLAs) and traffic metering



Slide material from Marko Luoma

DiffServ network (cont)

- Logical network offering differentiated service is a concatenation of PHBs which interact together.
- These logical networks have target service called per domain behavior (PDB).
- Target service is a loose definition for the goal of the logical network when it is provisioned and configured in an appropriate way.
- Edge router chooses PDB for each packet which comes from the customer.
  - marks packet with DSCP of PHB used to implement PDB
- 2 PHBs have been standardized
  - EF: Expedited Forwarding
  - AF: Assured Forwarding
    - actually collection of 4 different classes

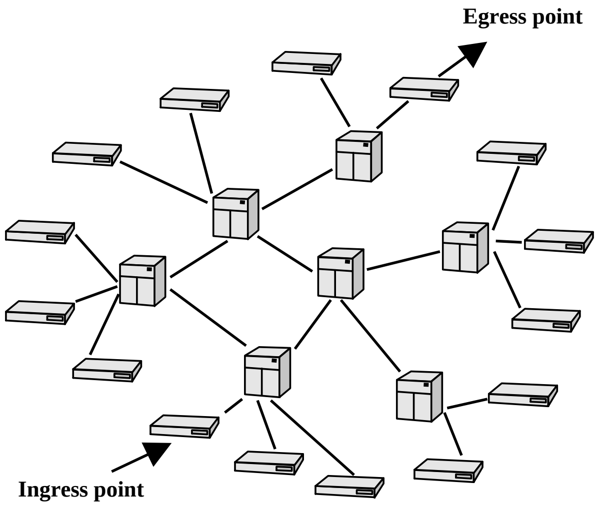
Slide material from Marko Luoma

DiffServ network (cont)

- Service decision in edge router can be based on:
  - metering result
    - rate based
    - token buckets
  - predefined set of filters
    - IP address i.e. customer
    - TCP/UDP port, i.e., application
  - user request
    - precoded DSCP
    - RSVP signaling
- Core routers do nothing but forwarding of packets based on the extra information in DSCP field of packets
- Requires
  - Classifier to detect DSCP fields
  - PHB to implement forwarding behaviors

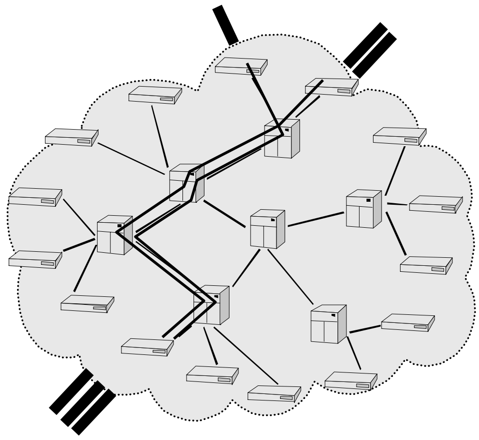
Expedited Forwarding (EF) [RFC2598]

- Leased line emulation
  - from destined ingress point to destined egress point
  - end-to-end service with
    - low loss
    - low latency
    - low jitter
  - “premium service”



EF

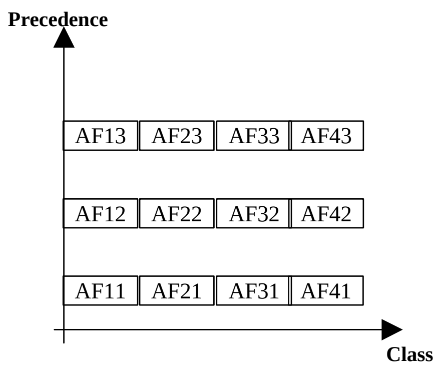
- Service commitment is only assured (not guaranteed)
  - resources inside EF class are shared
    - amount of other EF traffic influences the observed delay, jitter and loss
  - path is freely chosen
    - strict delay constraint can not be held as the delay of paths are inherently different
  - no reservation is done
    - provisioning is in the key role



Slide material from Marko Luoma

Assured Forwarding (AF) [RFC2597]

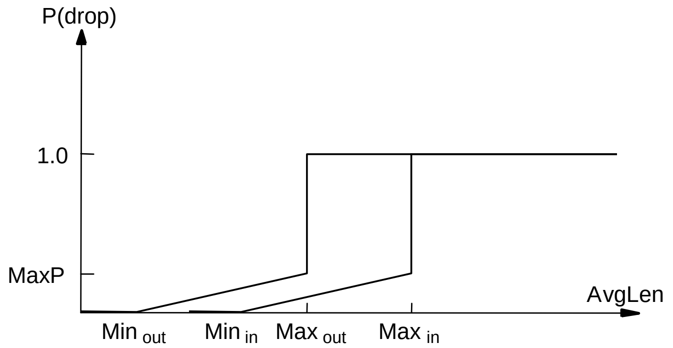
- Four independent service classes
  - all packets of a flow are destined to one of the classes
  - no association of service level between the classes
- Three precedences in each class
  - flow can have packets with different precedences (priorities)
  - order of packets in a flow is not allowed to change
    - precedence can not be used to scheduling decissions inside the class
    - precedence used to give, e.g., drop priorities



Slide material from Marko Luoma

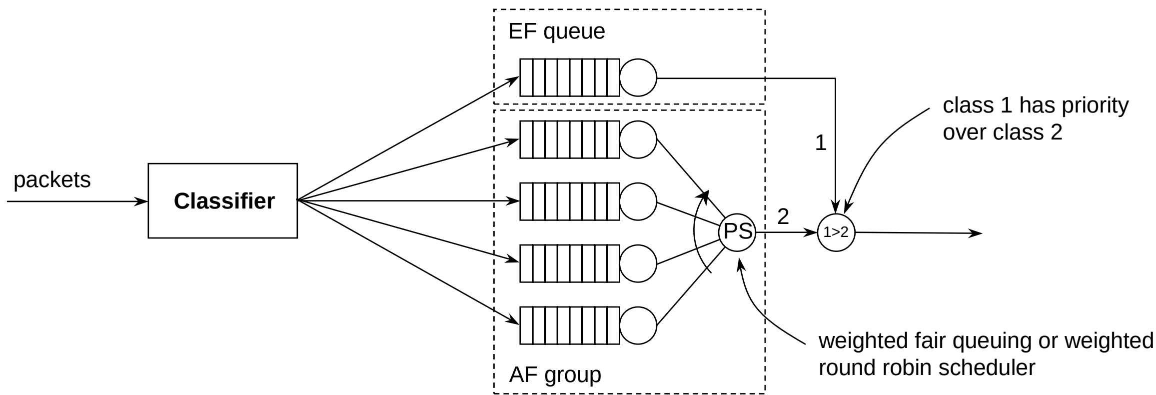
Implementing DiffServ PHBs: RIO

- One possible SIMPLIFIED implementation idea: assume two classes of traffic - “in” and “out”
- Business idea:
  - customer has contracted capacity of X bps, but sends packets with rate Y bps
  - if Y > X, some packets are marked out of profile
  - start to drop “out” packets first if there is congestion
- Two parallel RED algorithms for “in” and “out” packets = RIO
  - more than 2 classes = WRED algorithm

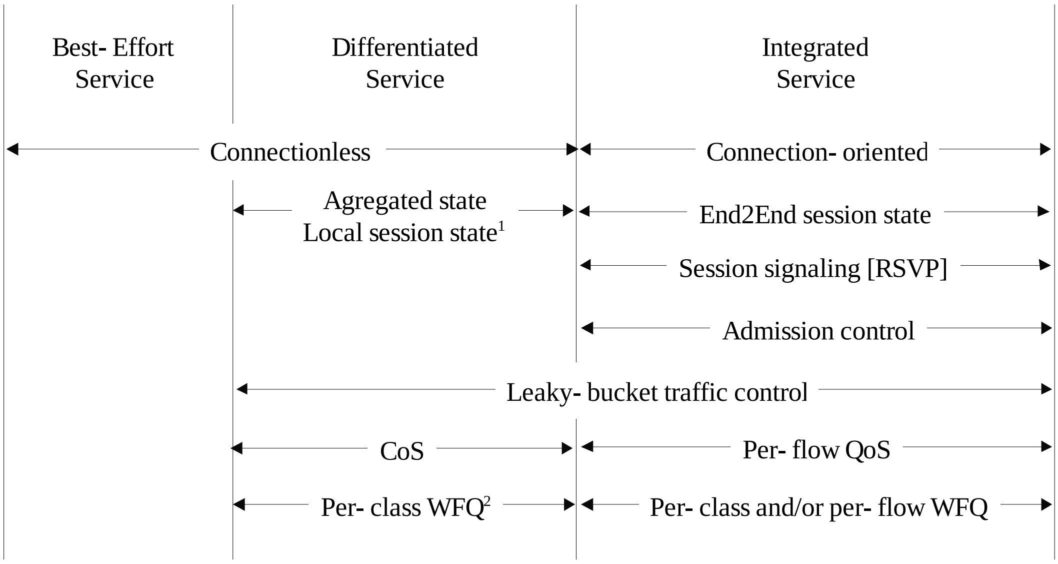


Implementing DiffServ PHBs: more advanced...

- EF packets have absolute priority over AF packets
  - if too much EF traffic, may starve AF queue(s)
  - could be fair queuing, as well
- AF groups separated with fair queuing
  - for each group, to implement drop precedences (3), we could have WRED with 3 classes (instead of 2 as in the previous slide)



Comparison



[1] Border routers may keep track individual sessions if required by policing or multifield classification.
[2] Scheduling depends on per hop behavior [PHB]. Minimum requirement is FIFO with multilevel RED.

Slide material from Marko Luoma

Remarks about Differentiated Services

- The idea of DiffServ is to combine individual flows into aggregates and to provide differentiated services inside the network (i.e., forwarding and discarding) to those.
- Under what conditions does it follow that when you serve an aggregate in a certain way, each individual flow in the aggregate gets some specific service?
  - need fair algorithms
  - open research problem
  - note: if you charge a customer (flow) for a better service, you need to provide that…
- Knowledge of the **offered flow** and careful setting of parameters are important in DiffServ
  - wrong parameters ⇒ your “premium” service is actually worse than your “best-effort” ⇒ careful network planning and provisioning are essential
- How to make sure that system can not be manipulated?