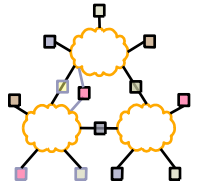
CE693: Adv. Computer Networking

L-5 QoS

Fall 1390

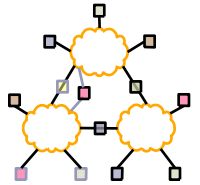
Acknowledgments: Lecture slides are from the graduate level Computer Networks course thought by Srinivasan Seshan at CMU. When slides are obtained from other sources, a a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.

Overview



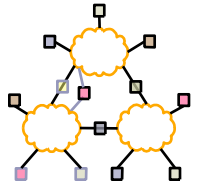
- Why QOS?
- Integrated services
- Adaptive applications
- Differentiated services

Motivation



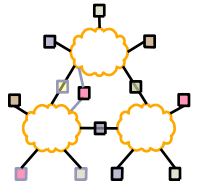
- Internet currently provides one single class of **“best-effort” service**
  - No assurances about delivery
- Existing applications are ***elastic***
  - Tolerate delays and losses
  - Can adapt to congestion
- Future “real-time” applications may be ***inelastic***

Inelastic Applications



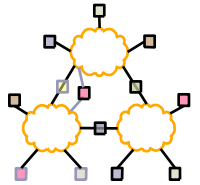
- Continuous media applications
  - **Lower and upper limit** on acceptable performance.
  - BW below which video and audio are not intelligible
  - Internet telephones, teleconferencing with high delay (200 - 300ms) impair human interaction
- Hard real-time applications
  - Require **hard limits on performance**
  - E.g. control applications

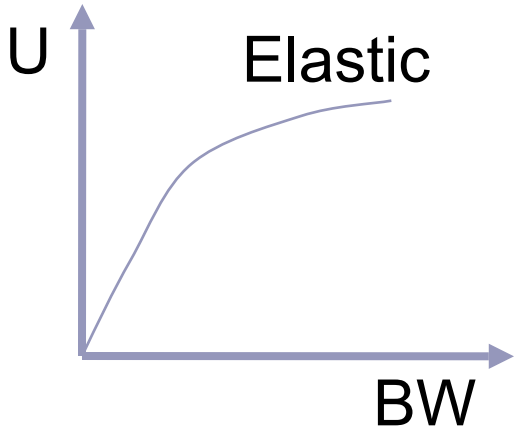
Why a New Service Model?

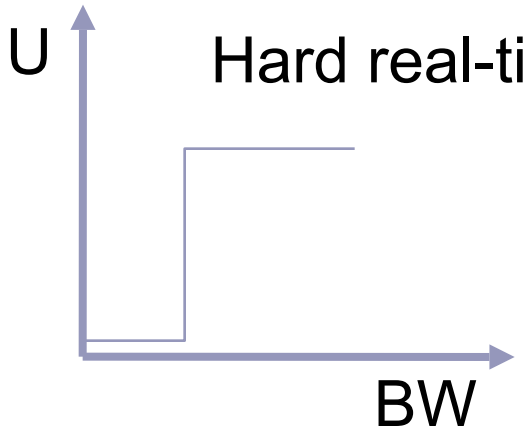


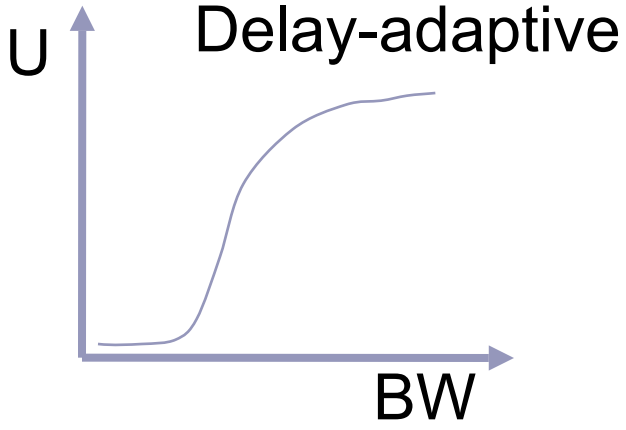
- What is the **basic objective** of network design?
  - Maximize total bandwidth? Minimize latency?
  - **Maximize user satisfaction** – the total **utility** given to users
- What does utility vs. bandwidth look like?
  - Must be non-decreasing function
  - Shape depends on application

Utility Curve Shapes



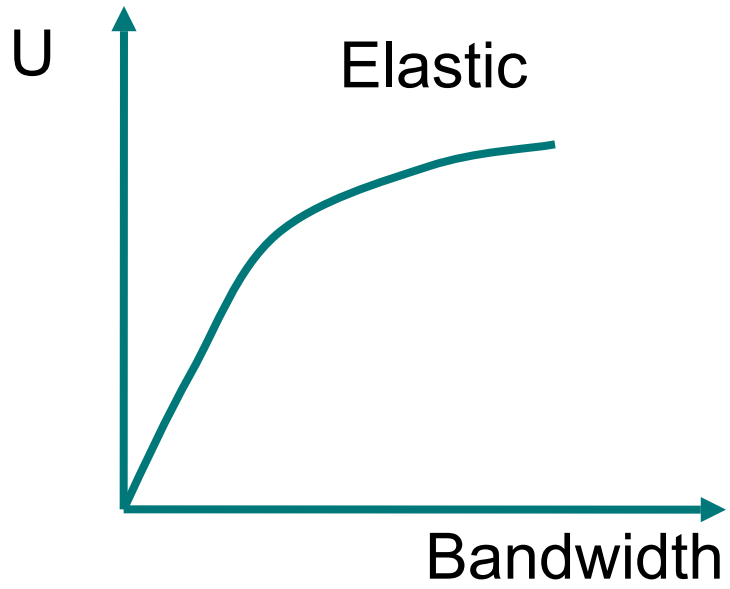






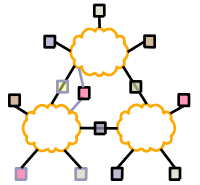
Stay to the right and you are fine for all curves

Utility curve – Elastic traffic

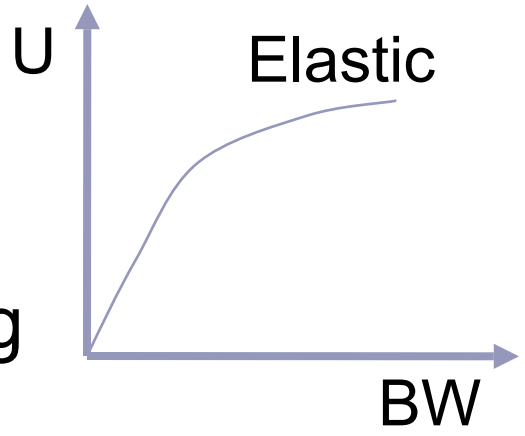


Does equal allocation of bandwidth maximize total utility?

Admission Control



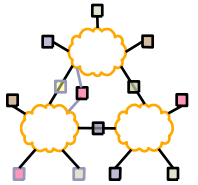
- If U(bandwidth) is concave
  → elastic applications

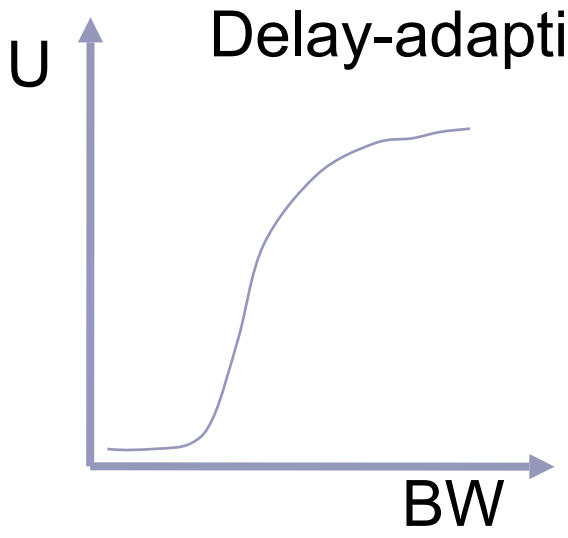


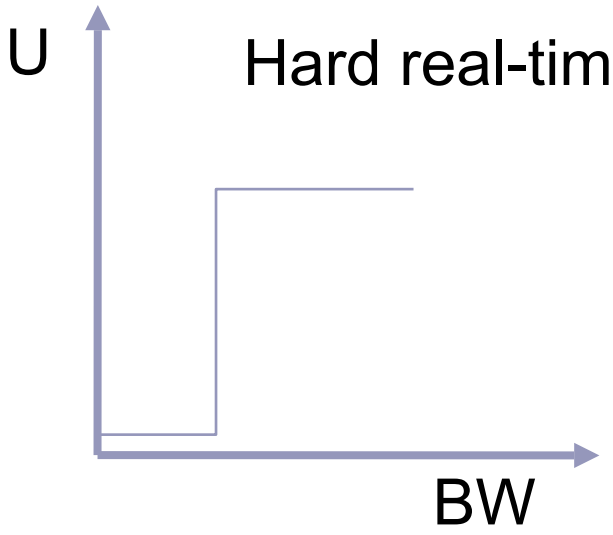
  - Incremental utility is decreasing with increasing bandwidth
  - Is always advantageous to have more flows with lower bandwidth
    - No need of admission control;

This is why the Internet works!

Utility Curves – Inelastic traffic

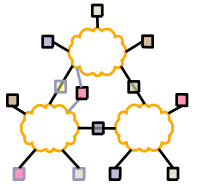




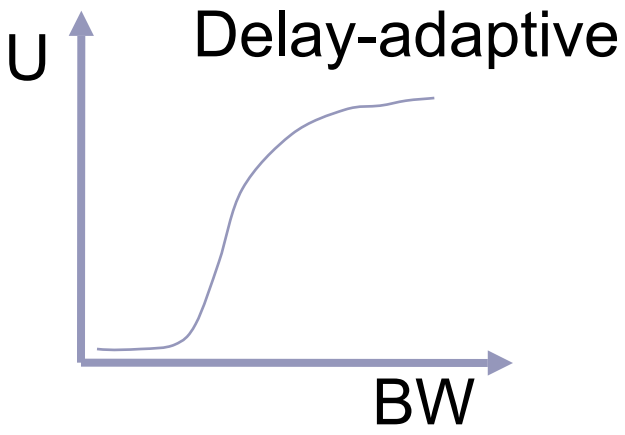


Does equal allocation of bandwidth maximize total utility?

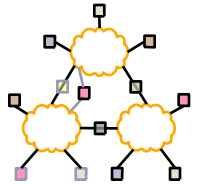
Admission Control



- If U is convex → inelastic applications
  - U(number of flows) is no longer monotonically increasing
  - Need admission control to maximize total utility
- **Admission control** → deciding when the addition of new people would result in reduction of utility
  - Basically avoids overload

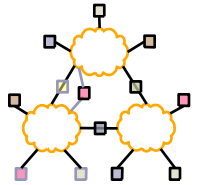


Overview



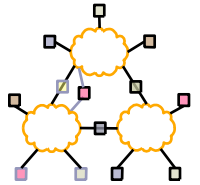
- Why QOS?
- Integrated services
- Adaptive applications
- Differentiated services

Components of Integrated Services

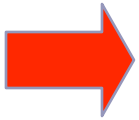


1. **Type of commitment**
   - **What does the network promise?**
2. Packet scheduling
   - How does the network meet promises?
3. Service interface
   - How does the application describe what it wants?
4. Establishing the guarantee
   - How is the promise communicated to/from the network
   - How is admission of new applications controlled?

1. Type of commitment

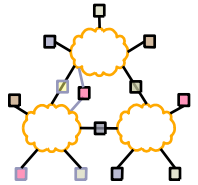


What kind of promises/services should network offer?



Depends on the **characteristics of the applications** that will use the network ….

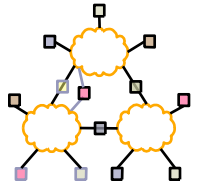
Playback Applications



- Sample signal → packetize → transmit → buffer → playback
  - Fits most multimedia applications

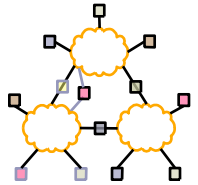
- Performance concern:
  - Jitter – variation in end-to-end delay
    - Delay = fixed + variable = (propagation + packetization) + queuing
- Solution:
  - Playback point – delay introduced by buffer to hide network jitter

Characteristics of Playback Applications



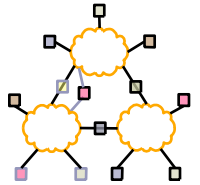
- In general lower delay is preferable.
- Doesn’t matter when packet arrives as long as it is before playback point
- Network guarantees (e.g. bound on jitter) would make it easier to set playback point
- Applications can tolerate some loss

Applications Variations



- Rigid & adaptive applications
  - Rigid – set fixed playback point
  - Adaptive – adapt playback point
    - Gamble that network conditions will be the same as in the past
    - Are prepared to deal with errors in their estimate
    - Will have an earlier playback point than rigid applications
- Tolerant & intolerant applications
  - Tolerance to brief interruptions in service
- 4 combinations

Applications Variations



Really only two classes of applications

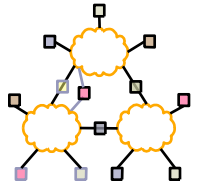
1) **Intolerant and rigid**
2) **Tolerant and adaptive**

Other combinations make little sense

3) Intolerant and adaptive
   - Cannot adapt without interruption
4) Tolerant and rigid
   - Missed opportunity to improve delay

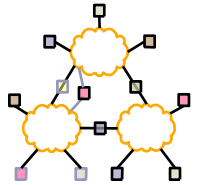
So what service classes should the network offer?

Type of Commitments



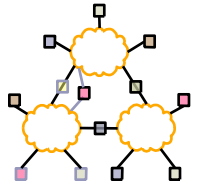
- **Guaranteed** service
  - For **intolerant and rigid** applications
  - Fixed guarantee, network meets commitment as long as clients send at match traffic agreement
- **Predicted** service
  - For **tolerant and adaptive** applications
  - Two components
    - If conditions do not change, commit to current service
    - If conditions change, take steps to deliver consistent performance (help apps minimize playback delay)
    - Implicit assumption – network does not change much over time
- **Datagram/best effort service**

Components of Integrated Services



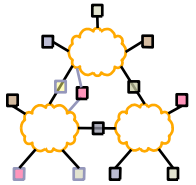
1. Type of commitment
   - What does the network promise?
2. **Packet scheduling**
   - **How does the network meet promises?**
3. **Service interface**
   - **How does the application describe what it wants?**
4. Establishing the guarantee
   - How is the promise communicated to/from the network
   - How is admission of new applications controlled?

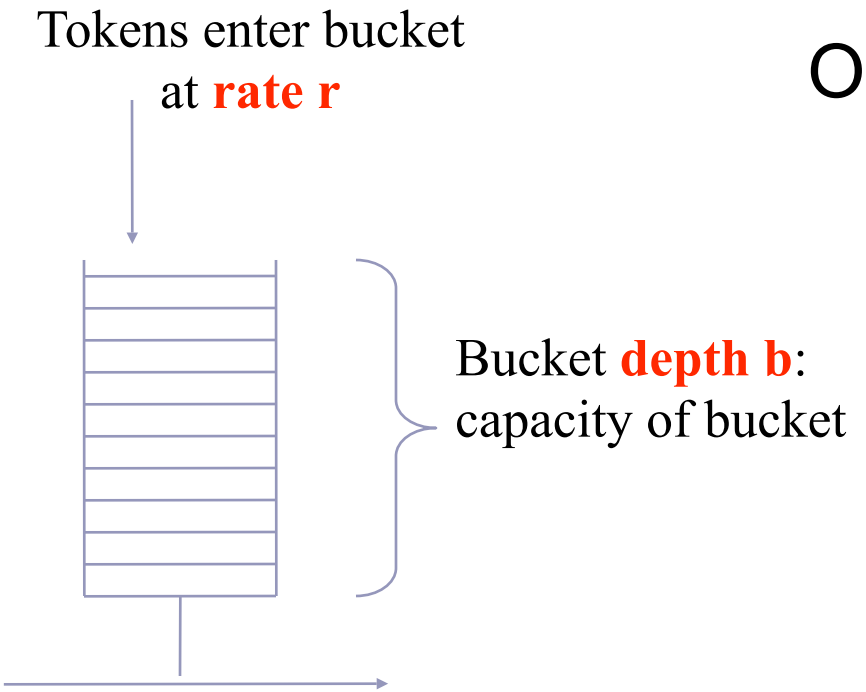
Scheduling for Guaranteed Traffic



- Use **token bucket filter** to characterize traffic
  - Described by rate *r* and bucket depth *b*
- Use **WFQ** at the routers
- Parekh's bound for worst case queuing delay = b/r

Token Bucket Filter

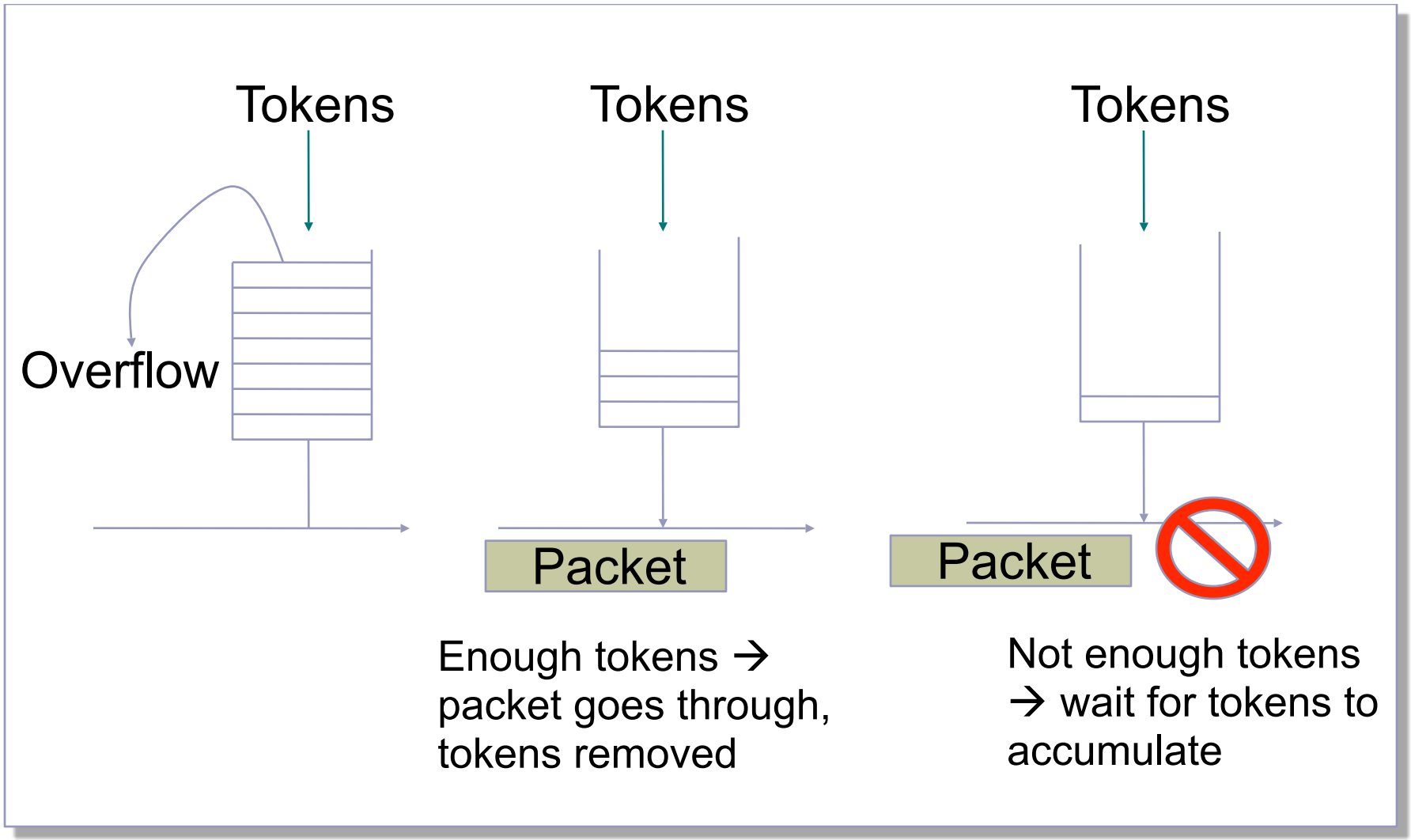




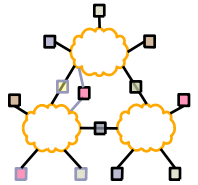
Operation:

- If bucket fills, tokens are discarded
- Sending a packet of size P uses P tokens
- If bucket has P tokens, packet sent at max rate, else must wait for tokens to accumulate

Token Bucket Operation

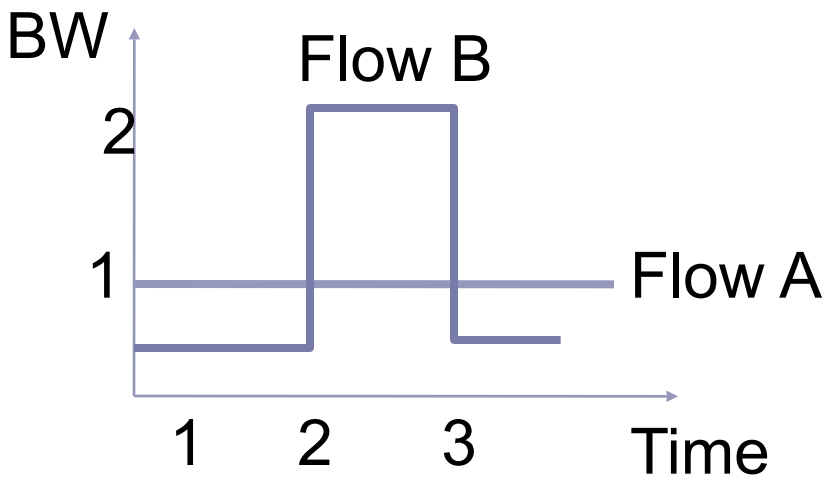


Token Bucket Characteristics



- On the long run, rate is limited to r
- On the short run, a burst of size b can be sent
- Amount of traffic entering at interval T is bounded by:
  - Traffic = b + r*T
- Information useful to admission algorithm

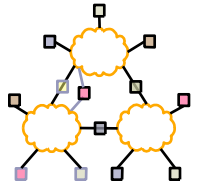
Token Bucket Specs



Flow A: r = 1 MBps, B=1 byte

Flow B: r = 1 MBps, B=1 MB

Predicted Service



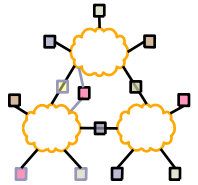
Goals:

- Isolation
  - Isolates well-behaved from misbehaving sources
- Sharing
  - Mixing of different sources in a way beneficial to all

Mechanisms:

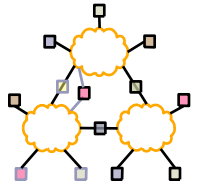
- WFQ
  - Great isolation but no sharing
- FIFO
  - Great sharing but no isolation

Predicted Service



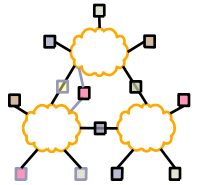
- FIFO jitter increases with the number of hops
  - Use opportunity for sharing across hops
- FIFO+
  - At each hop: measure average delay for class at that router
  - For each packet: compute difference of average delay and delay of that packet in queue
  - Add/subtract difference in packet header
  - Packet inserted into queues expected arrival time instead of actual
    - More complex queue management!
- Slightly decreases mean delay and significantly decreases jitter

Unified Scheduling



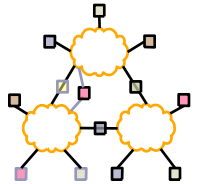
- Assume 3 types of traffic: guaranteed, predictive, best-effort
- Scheduling: use WFQ in routers
- Each guaranteed flow gets its own queue
- All predicted service flows and best effort aggregates in single separate queue
  - Predictive traffic classes
    - Multiple FIFO+ queues
    - Worst case delay for classes separated by order of magnitude
    - When high priority needs extra bandwidth – steals it from lower class
  - Best effort traffic acts as lowest priority class

Service Interfaces



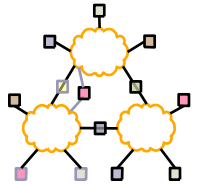
- Guaranteed Traffic
  - Host specifies rate to network
  - Why not bucket size b?
    - If delay not good, ask for higher rate
- Predicted Traffic
  - Specifies (r, b) token bucket parameters
  - Specifies delay D and loss rate L
  - Network assigns priority class
  - Policing at edges to drop or tag packets
    - Needed to provide isolation – why is this not done for guaranteed traffic?
      - WFQ provides this for guaranteed traffic

Overview



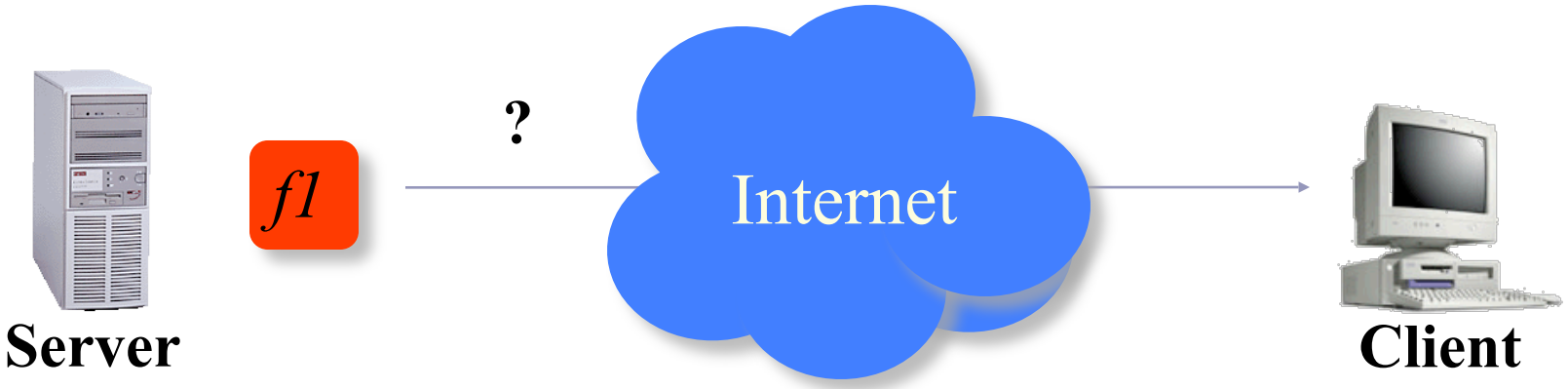
- Why QOS?
- Integrated services
- Adaptive applications
- Differentiated services

Internet Video Today



- Client-server streaming
  - Skype video conferencing
  - Hulu
- DVD transfer
  - BitTorrent → P2P lecture
- Synchronized video (IPTV)
  - Overlay multicast → multicast lecture

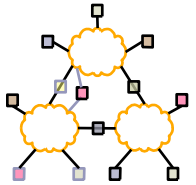
Problems Adapting to Network State

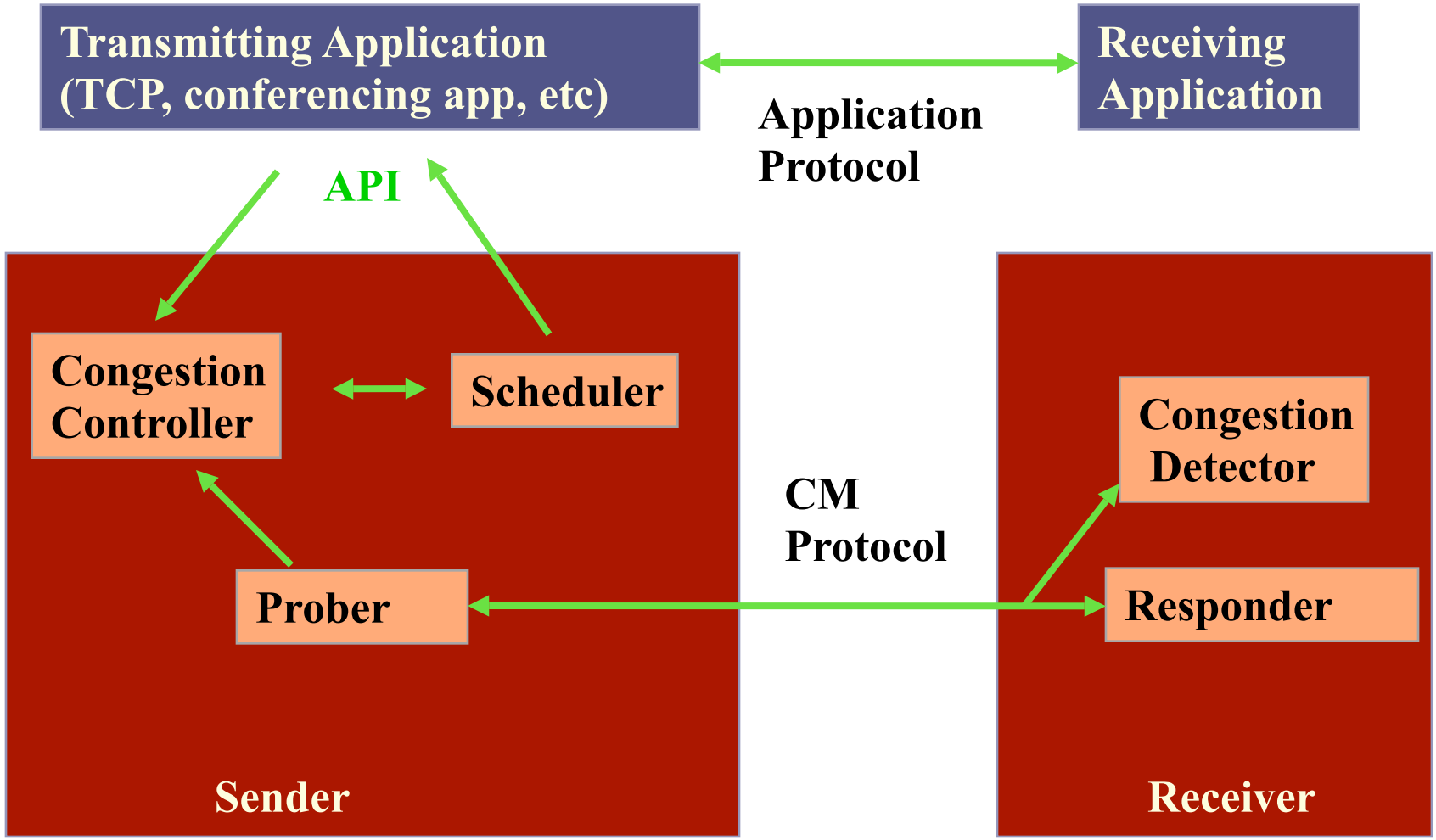


- TCP hides network state
- New applications may not use TCP
  - Often do not adapt to congestion

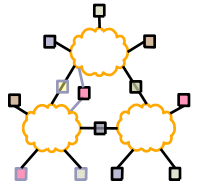
Need system that helps applications learn and adapt to congestion

Congestion Manager Architecture

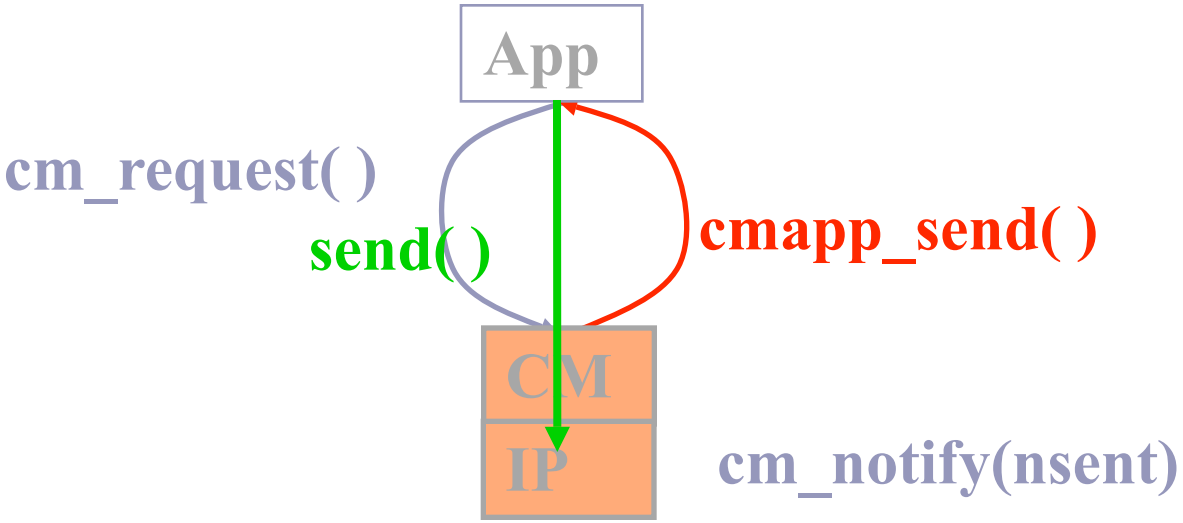




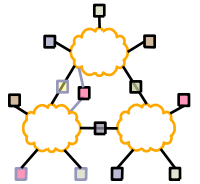
Transmission API



- Buffered send
  - cm_send(data, length)
- Request/callback-based send

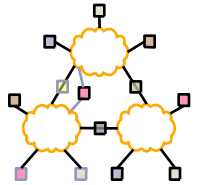


Overview



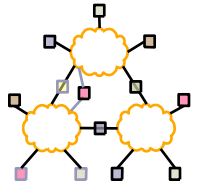
- Why QOS?
- Integrated services
- Adaptive applications
- Differentiated services

DiffServ



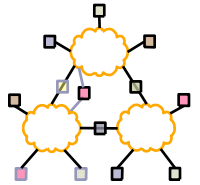
- Analogy:
  - Airline service, first class, coach, various restrictions on coach as a function of payment
- Best-effort expected to make up bulk of traffic, but revenue from first class important to economic base
- Not motivated by real-time! Motivated by economics and assurances

Basic Architecture



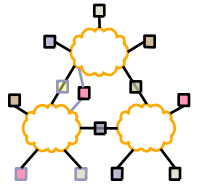
- Agreements/service provided within a domain
  - Service Level Agreement (SLA) with ISP
- Edge routers do traffic conditioning
  - Perform per aggregate shaping and policing
  - Mark packets with a small number of bits; each bit encoding represents a class or subclass
- Core routers
  - Process packets based on packet marking and defined per hop behavior
- More scalable than IntServ
  - No per flow state or signaling

Per-hop Behaviors (PHBs)



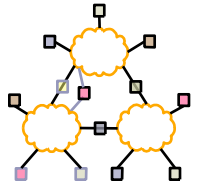
- Define behavior of individual routers rather than end-to-end services – there may be many more services than behaviors
- Multiple behaviors – need more than one bit in the header
- Six bits from IP TOS field are taken for Diffserv code points (DSCP)

Per-hop Behaviors (PHBs)



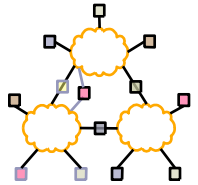
- Two PHBs defined so far
- Expedited forwarding aka premium service (type P)
  - Possible service: providing a virtual wire
  - Admitted based on peak rate
  - Unused premium goes to best effort
- Assured forwarding (type A)
  - Possible service: strong assurance for traffic within profile & allow source to exceed profile
  - Based on expected capacity usage profiles
  - Traffic unlikely to be dropped if user maintains profile
  - Out-of-profile traffic marked

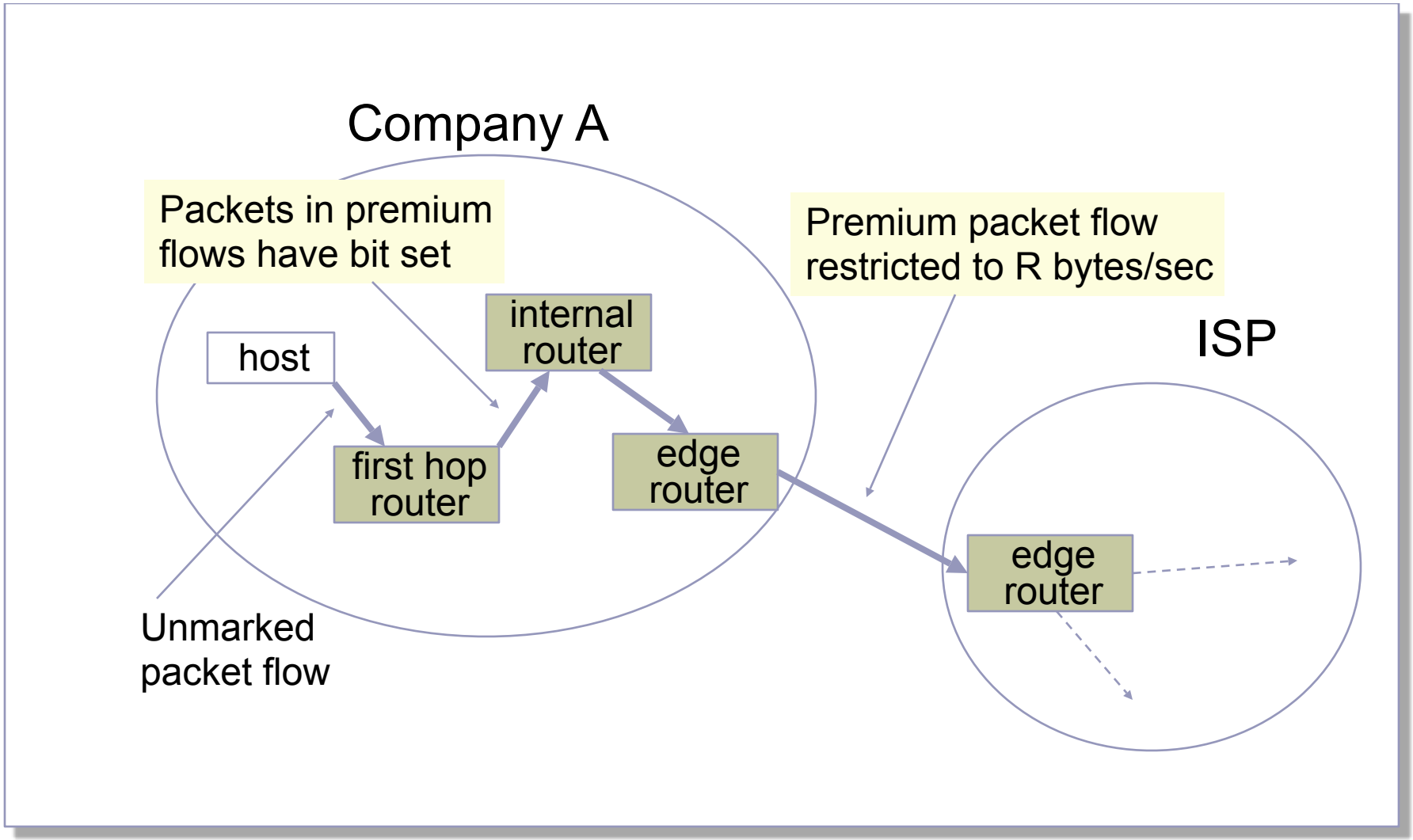
Expedited Forwarding PHB



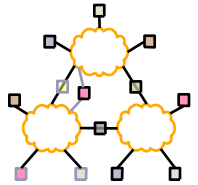
- User sends within profile & network commits to delivery with requested profile
  - Signaling, admission control may get more elaborate in future
- Rate limiting of EF packets at edges only, using token bucket to shape transmission
- Simple forwarding: classify packet in one of two queues, use priority
  - EF packets are forwarded with minimal delay and loss (up to the capacity of the router)

Expedited Forwarding Traffic Flow



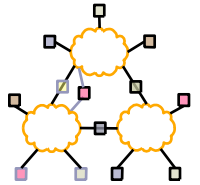


Assured Forwarding PHB



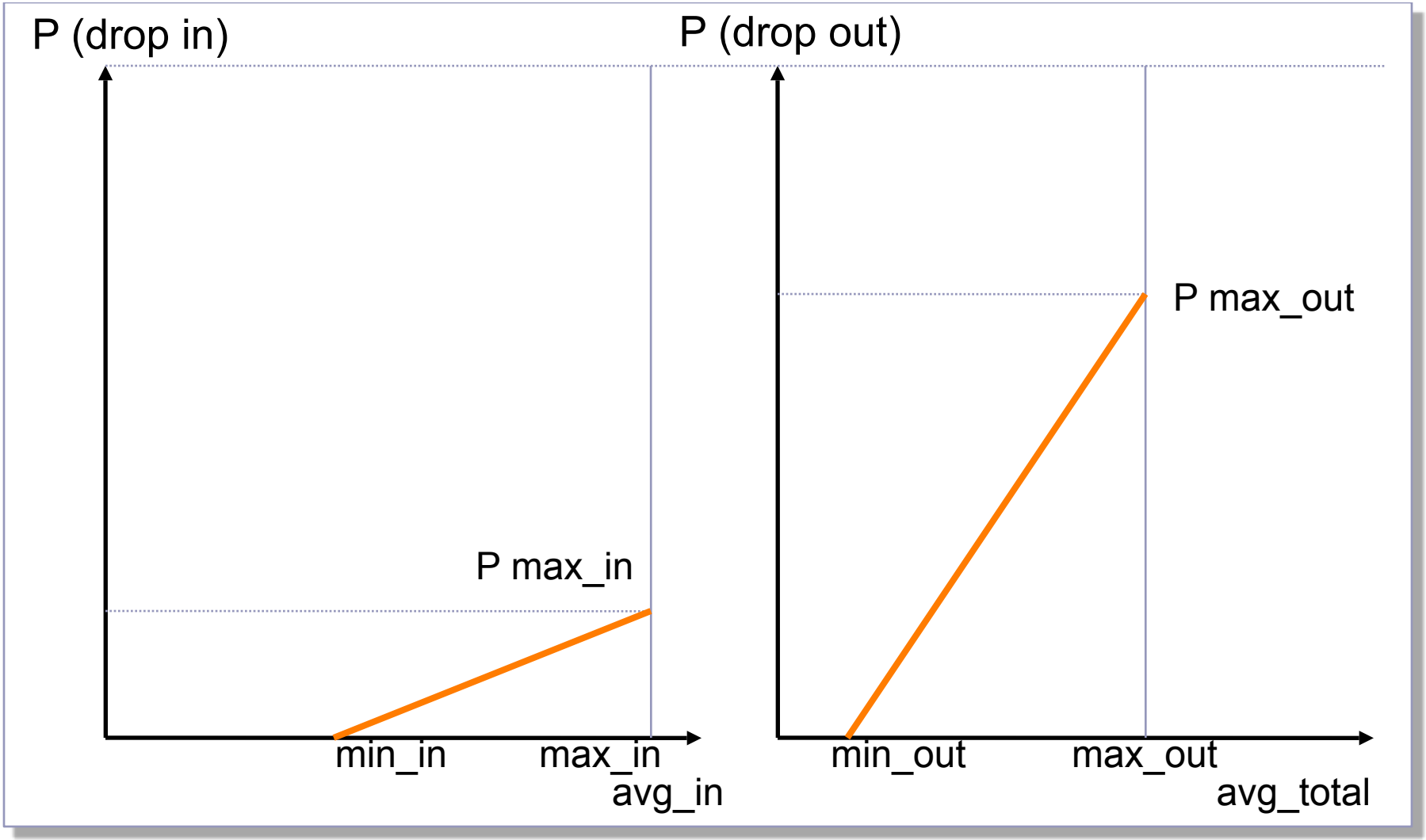
- User and network agree to some traffic profile
  - Edges mark packets up to allowed rate as “in-profile” or low drop precedence
  - Other packets are marked with one of 2 higher drop precedence values
- A congested DS node tries to protect packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value
  - Implemented using RED with In/Out bit

Red with In or Out (RIO)

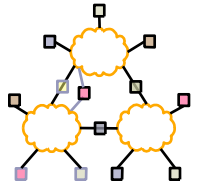


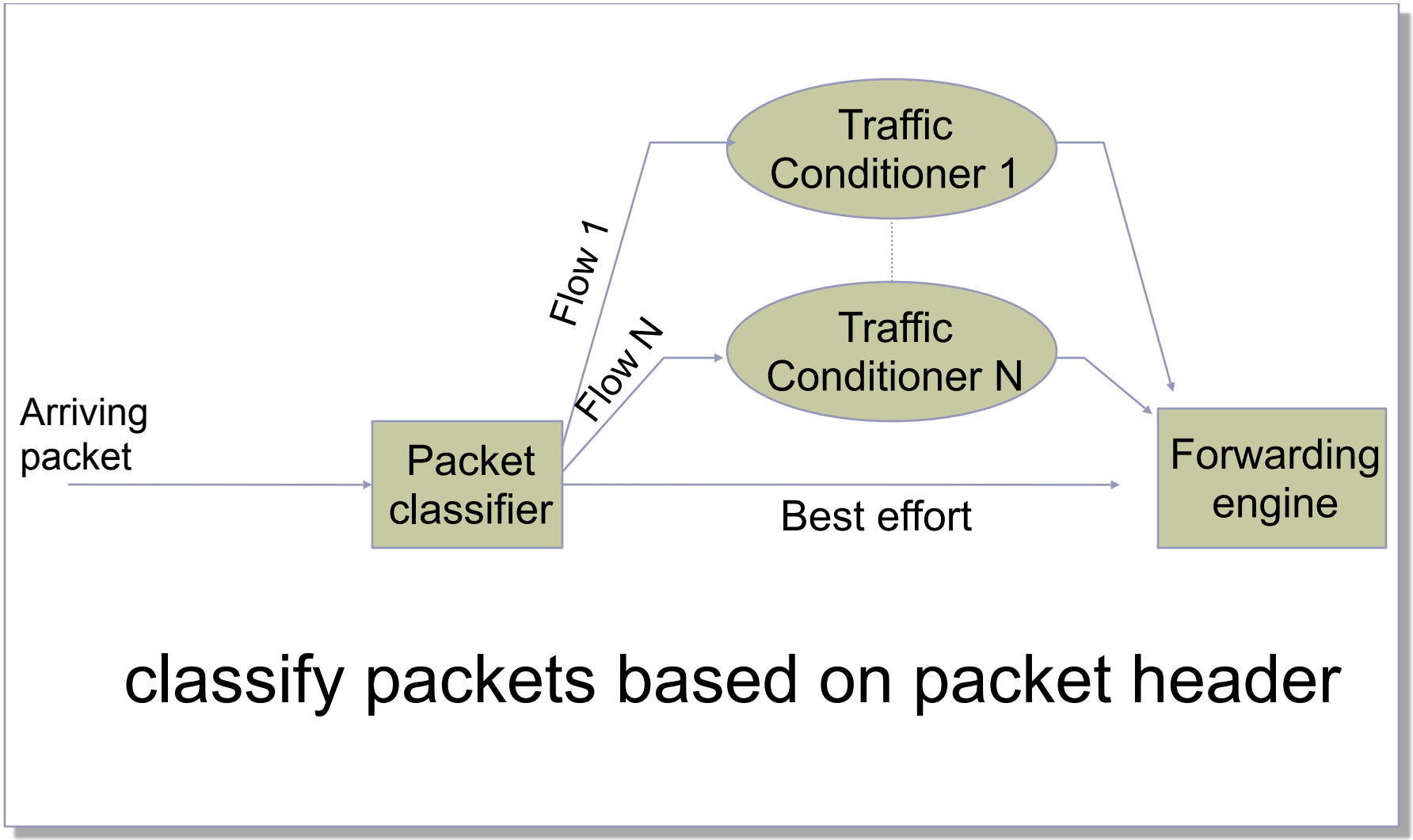
- Similar to RED, but with two separate probability curves
- Has two classes, “In” and “Out” (of profile)
- “Out” class has lower Min_{thresh}, so packets are dropped from this class first
  - Based on queue length of all packets
- As avg queue length increases, “in” packets are also dropped
  - Based on queue length of only “in” packets

RIO Drop Probabilities



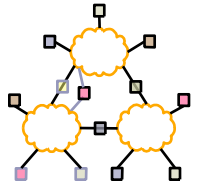
Edge Router Input Functionality

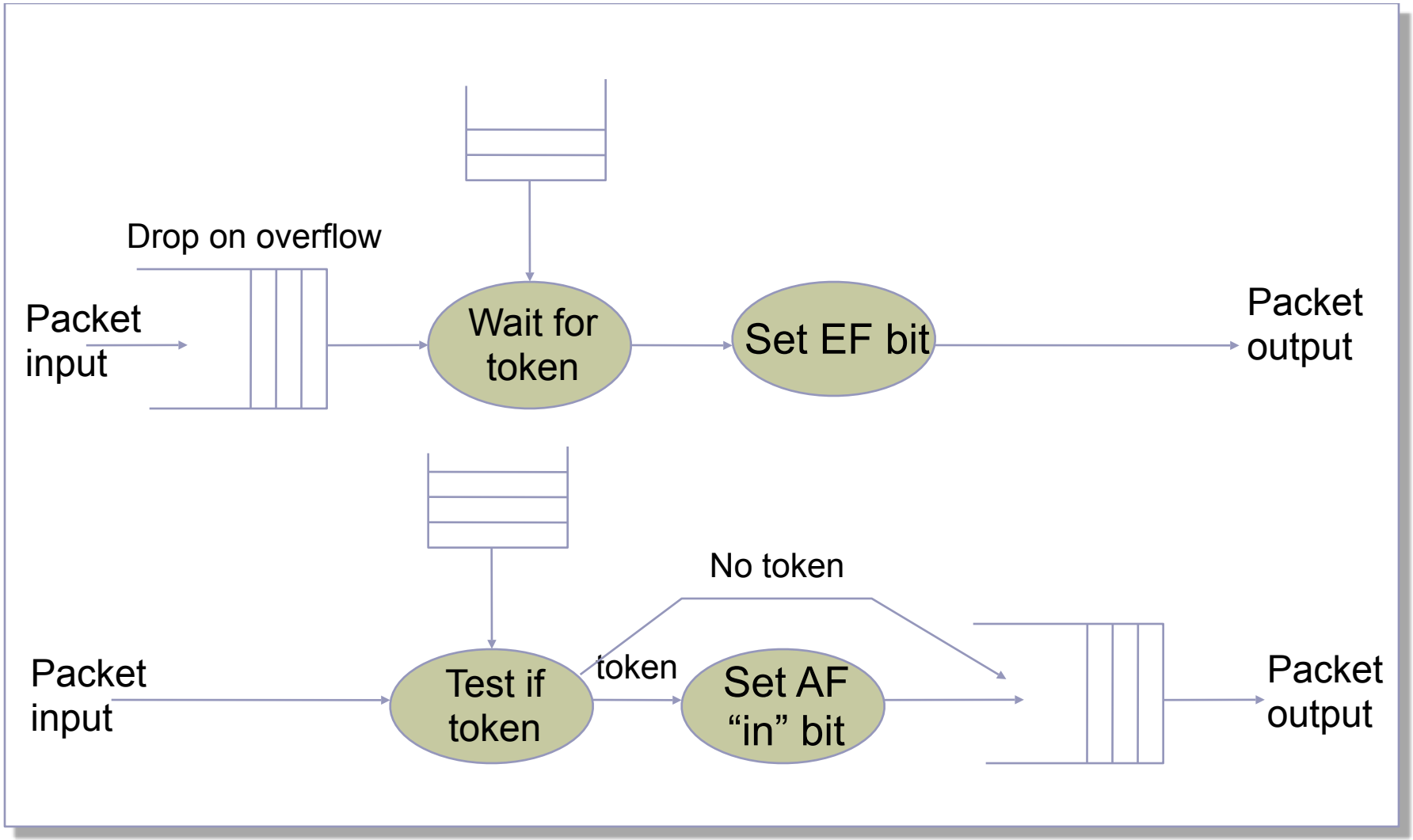




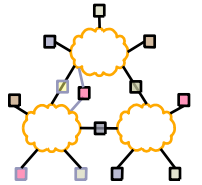
classify packets based on packet header

Traffic Conditioning

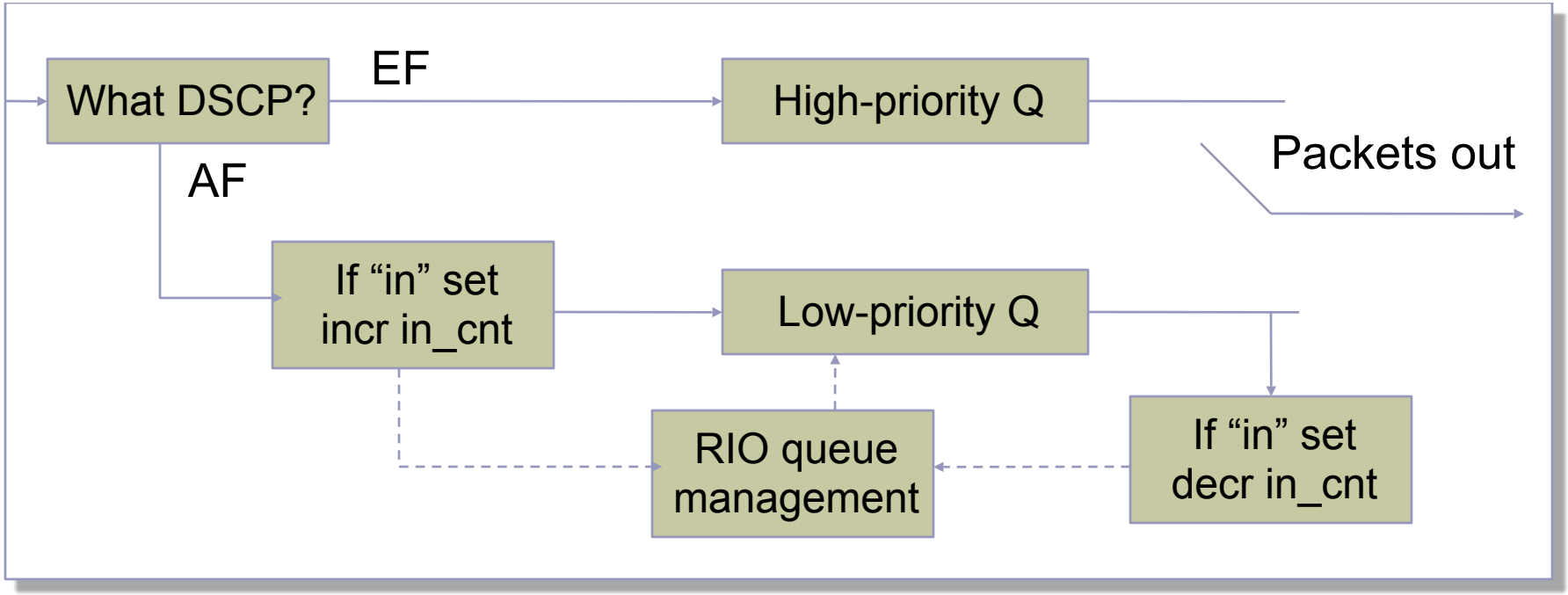




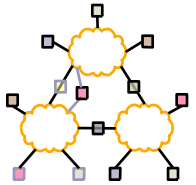
Router Output Processing

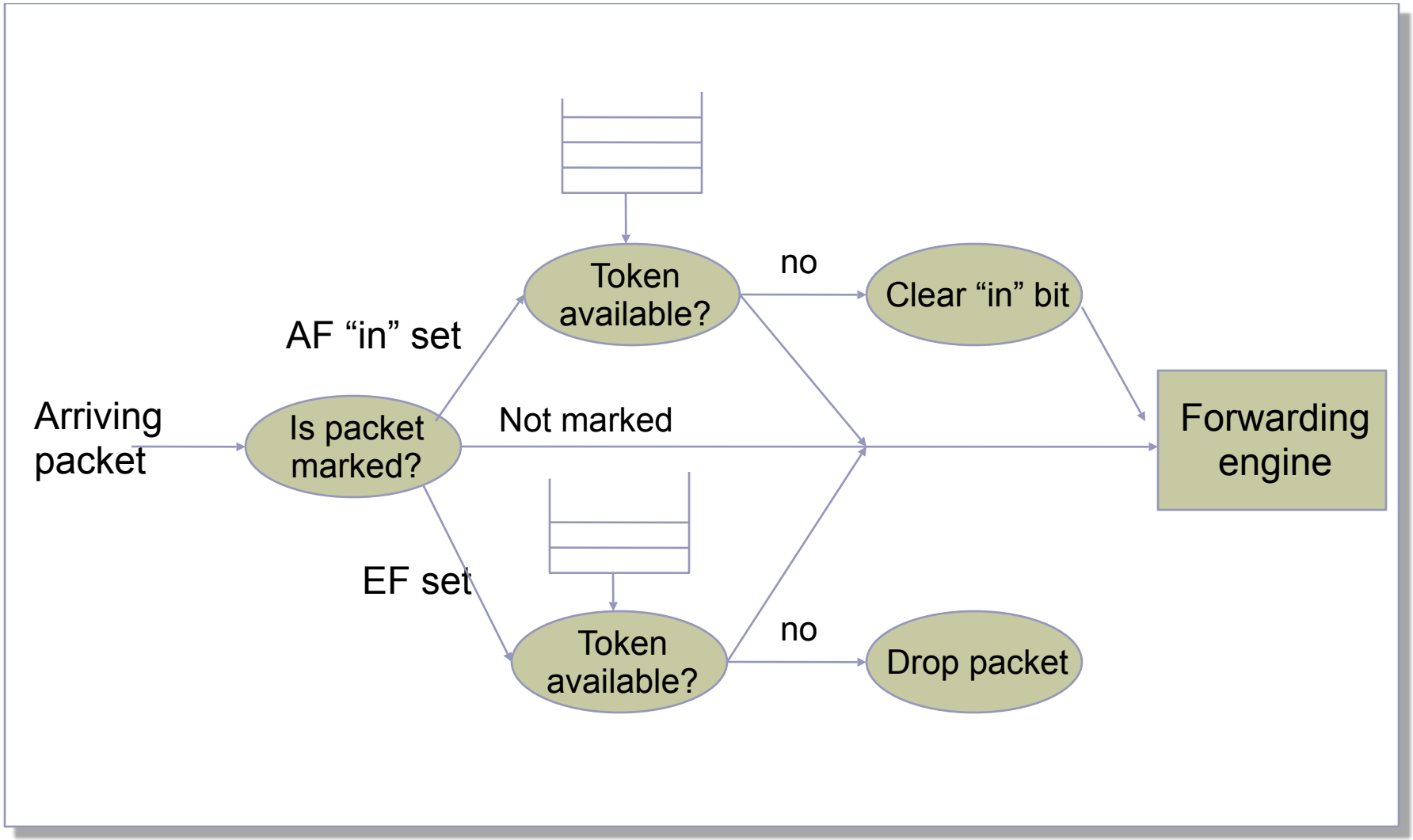


- 2 queues: EF packets on higher priority queue
- Lower priority queue implements RED “In or Out” scheme (RIO)

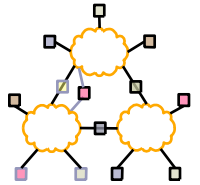


Edge Router Policing



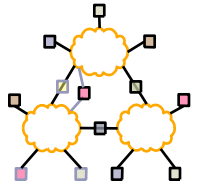


Comparison



	Best-Effort	Diffserv	Intserv
Service	• Connectivity • No isolation • No guarantees	• Per aggregation isolation • Per aggregation guarantee	• Per flow isolation • Per flow guarantee
Service Scope	• End-to-end	• Domain	• End-to-end
Complexity	• No set-up	• Long term setup	• Per flow setup
Scalability	• Highly scalable • (nodes maintain only routing state)	• Scalable (edge routers maintains per aggregate state; core routers per class state)	• Not scalable (each router maintains per flow state)

Next Lecture: Router Design



- Forwarding
- IP lookup
- High-speed router architecture
- Readings
  - [McK97] A Fast Switched Backplane for a Gigabit Switched Router
  - [KCY03] Scaling Internet Routers Using Optics
  - Know RIP/OSPF