KHOA CÔNG NGHỆ THÔNG TIN-ĐHKHTN
CSC11004 - MẠNG MÁY TÍNH NÂNG CAO

QUEUE MANAGEMENT & QUALITY OF SERVICES

Lê Ngọc Sơn

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Agenda

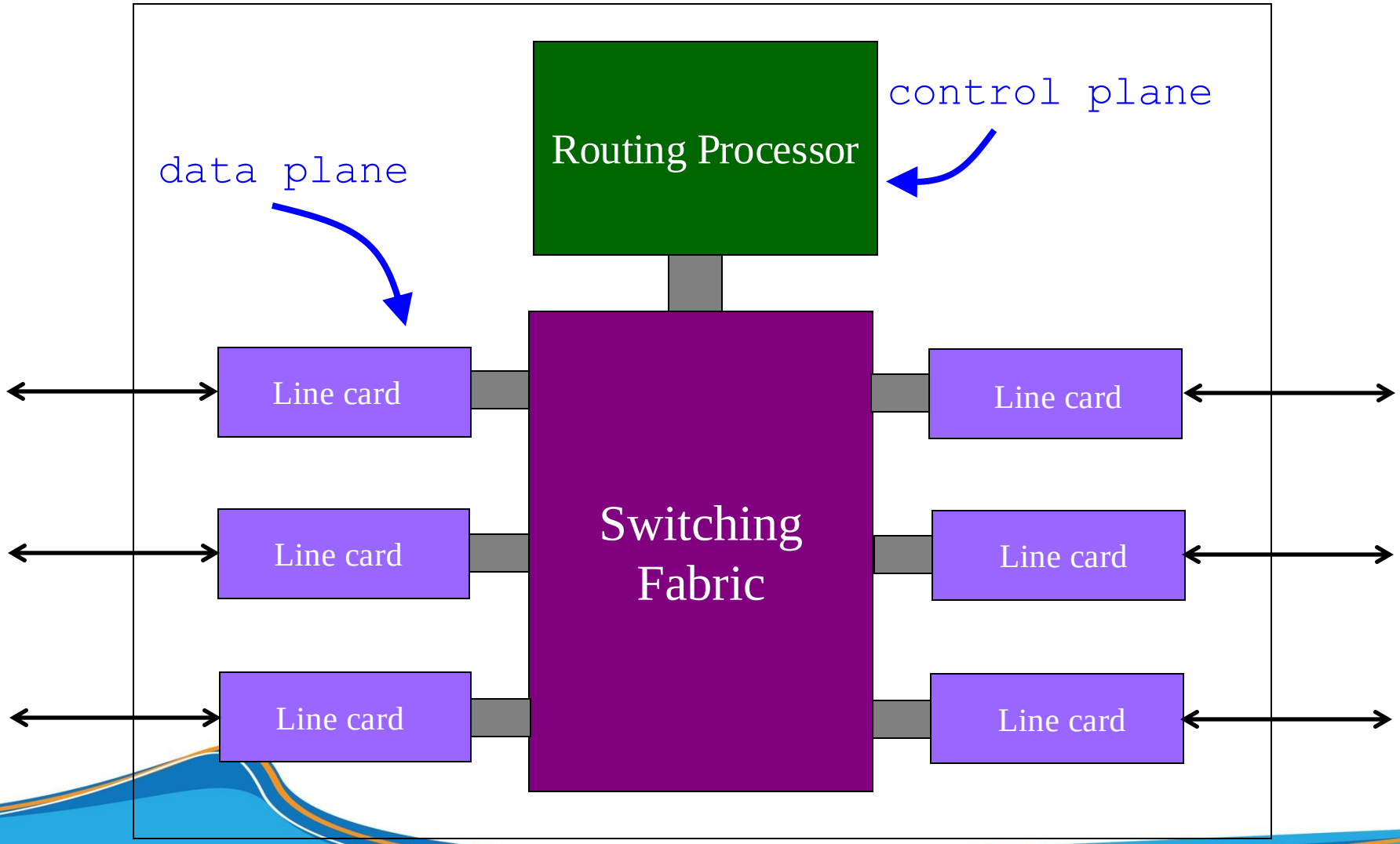
- Queue Management
- Drop Policy
- Scheduling Discipline
- Quality of Service
- IntServ and DiffServ

Queue Management

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

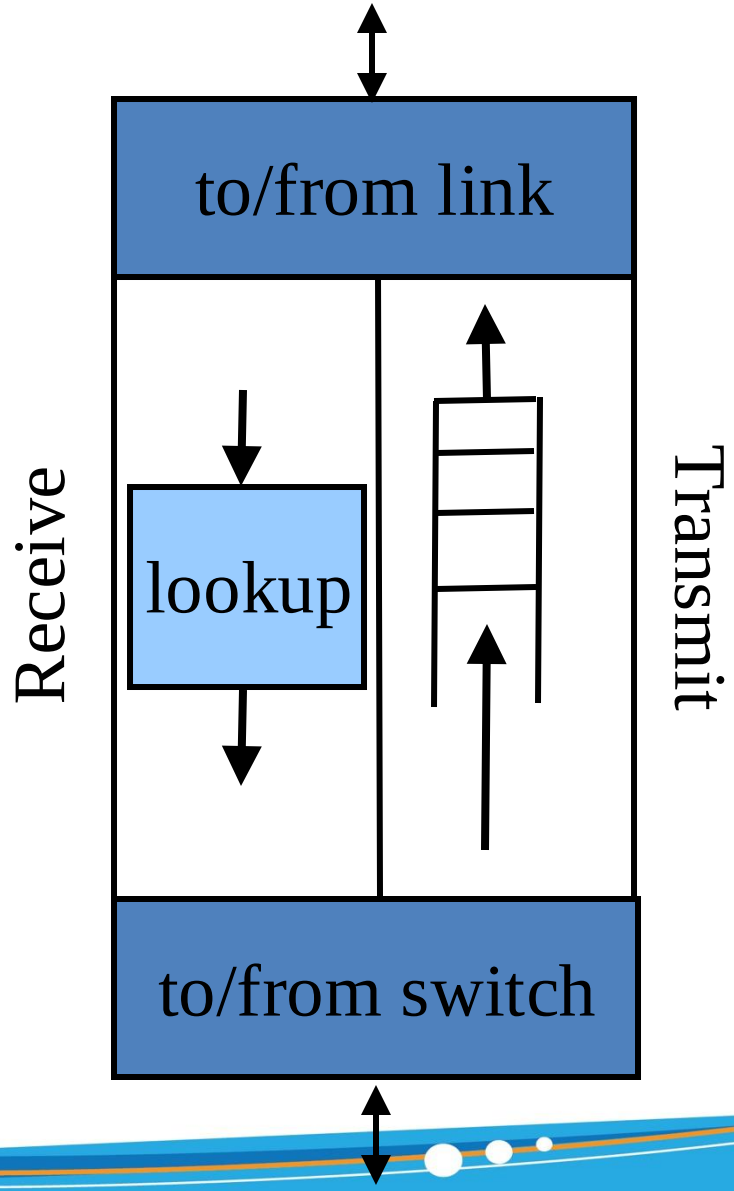
fit@hcmus

Router


Routing Processor
control plane
data plane
Line card
Line card
Line card
Switching Fabric
Line card
Line card
Line card

Line Cards (Interface Cards, Adaptors)

- Packet handling
  - ✓ Packet forwarding
  - ✓ Buffer management
  - ✓ Packet filtering
  - ✓ Rate limiting
  - ✓ Packet marking



Buffer Size

- Why not use infinite buffers?
  - ✓ no packet drops!
- Small buffers:
  - ✓ often drop packets due to bursts
  - ✓ but have small delays
- Large buffers:
  - ✓ reduce number of packet drops (due to bursts)
  - ✓ but increase delays
- Can we have the best of both worlds?

Queue Management

- What is Queuing Management?
  - ✓ Managing how packets are queued and transmitted in networking devices like routers and switches.
  - ✓ Goal: Efficiently handle network traffic and ensure quality of service (QoS).
- Queuing allocates both bandwidth and buffer space:
  - ✓ Bandwidth: which packet to serve (transmit) next
  - ✓ Buffer space: which packet to drop next (when required)
- Importance of Queuing Management:
  - ✓ Prevents congestion.
  - ✓ Optimizes bandwidth usage.
  - ✓ Improves user experience.

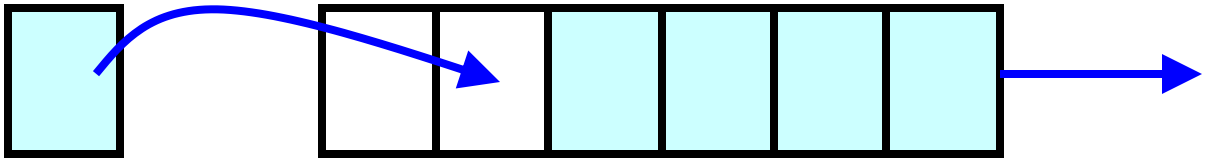
Queue Management Issues

- Drop policy
  - ✓ When should you discard a packet?
  - ✓ Which packet to discard?
- Scheduling discipline
  - ✓ Which packet to send?
  - ✓ Some notion of fairness? Priority?
- Goal: balance throughput and delay
  - ✓ Huge buffers minimize drops, but add to queuing delay (thus higher RTT, longer slow start, …)

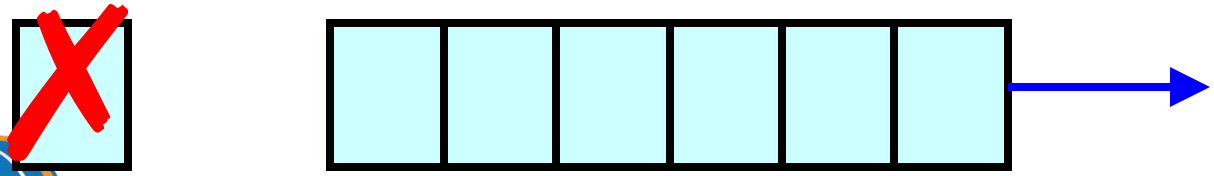
Drop Policies

FIFO Scheduling and Drop-Tail

- Access to the bandwidth: first-in first-out queue
  - ✓ Packets only differentiated when they arrive

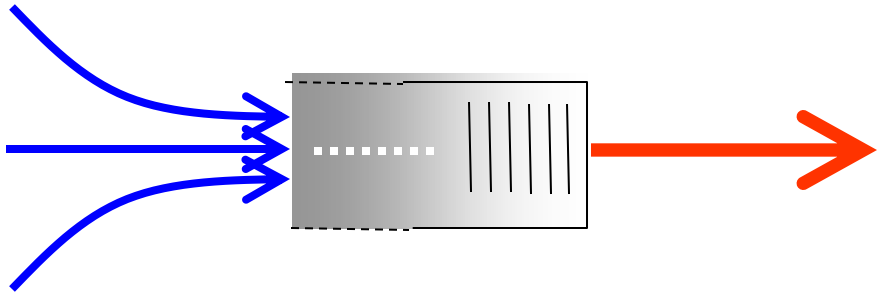


- Access to the buffer space: drop-tail queuing
  - ✓ If the queue is full, drop the incoming packet



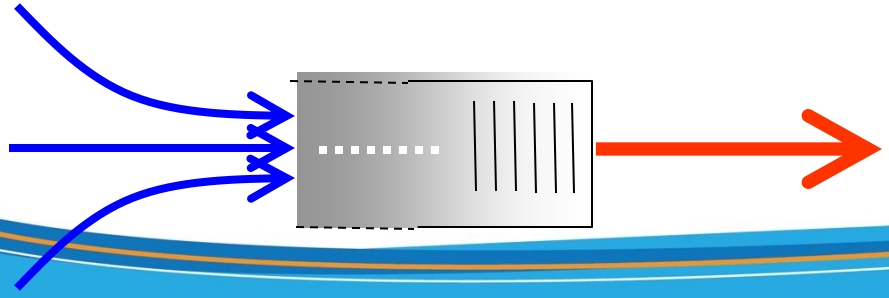
Bursty Loss From Drop-Tail Queuing

- TCP depends on packet loss
  - ✓ Packet loss is indication of congestion
  - ✓ TCP additive increase drives network into loss
- Drop-tail leads to bursty loss
  - ✓ Congested link: many packets encounter full queue
  - ✓ Synchronization: many connections lose packets at once



Slow Feedback from Drop Tail

- Feedback comes when buffer is completely full
  - ✓ … even though the buffer has been filling for a while
- Plus, the filling buffer is increasing RTT
  - ✓ … making detection even slower
- Better to give early feedback
  - ✓ Get 1-2 connections to slow down before it's too late!



Active Queue Management (AQM)

- AQM refers to techniques used in networking to control queue lengths before they overflow, aiming to minimize packet loss and congestion.
- Unlike **Drop-Tail**, AQM actively drops packets before the queue becomes full, reducing the risk of bursty loss and global synchronization.

AQM Benefits

- Prevents Congestion: Drops packets early to avoid buffer overflow.
- Reduces Latency: Keeps queue lengths shorter, minimizing packet waiting times.
- Improves Throughput: Prevents link underutilization by avoiding TCP global synchronization.
- Enhances Fairness: Helps in fair bandwidth distribution across multiple flows.

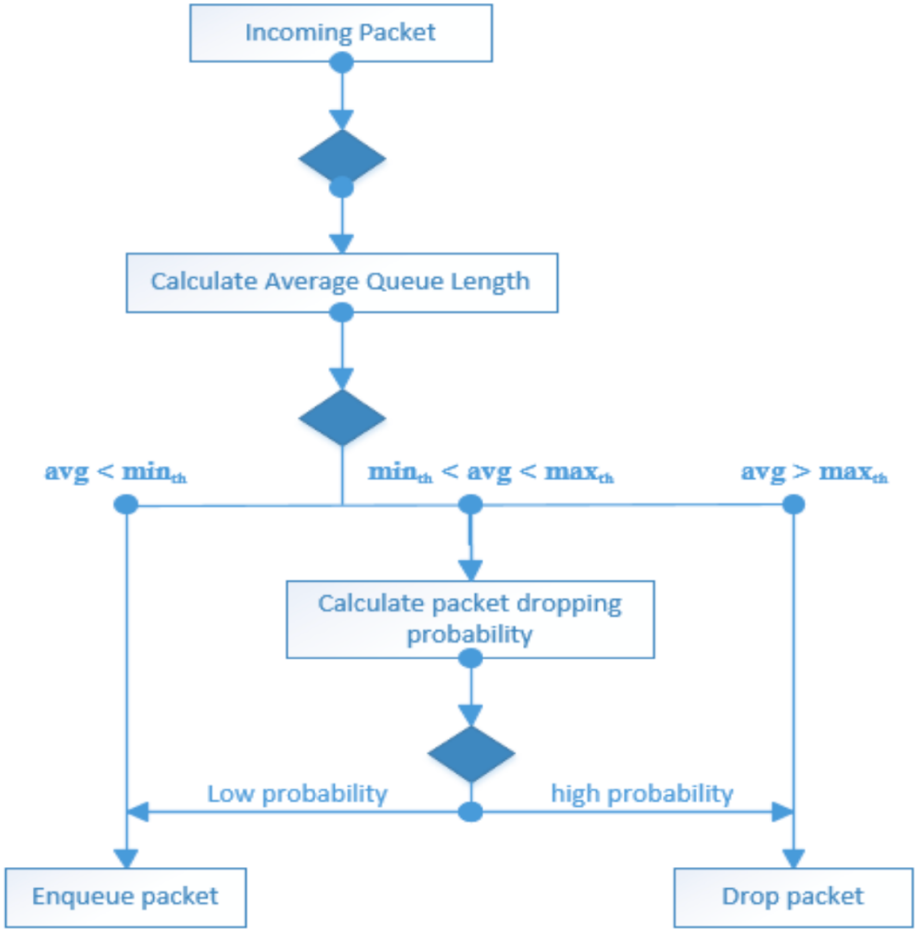
Common AQM Techniques

- Random Early Detection (RED)
- Explicit Congestion Notification (ECN)
- Controlled Delay (CoDel)

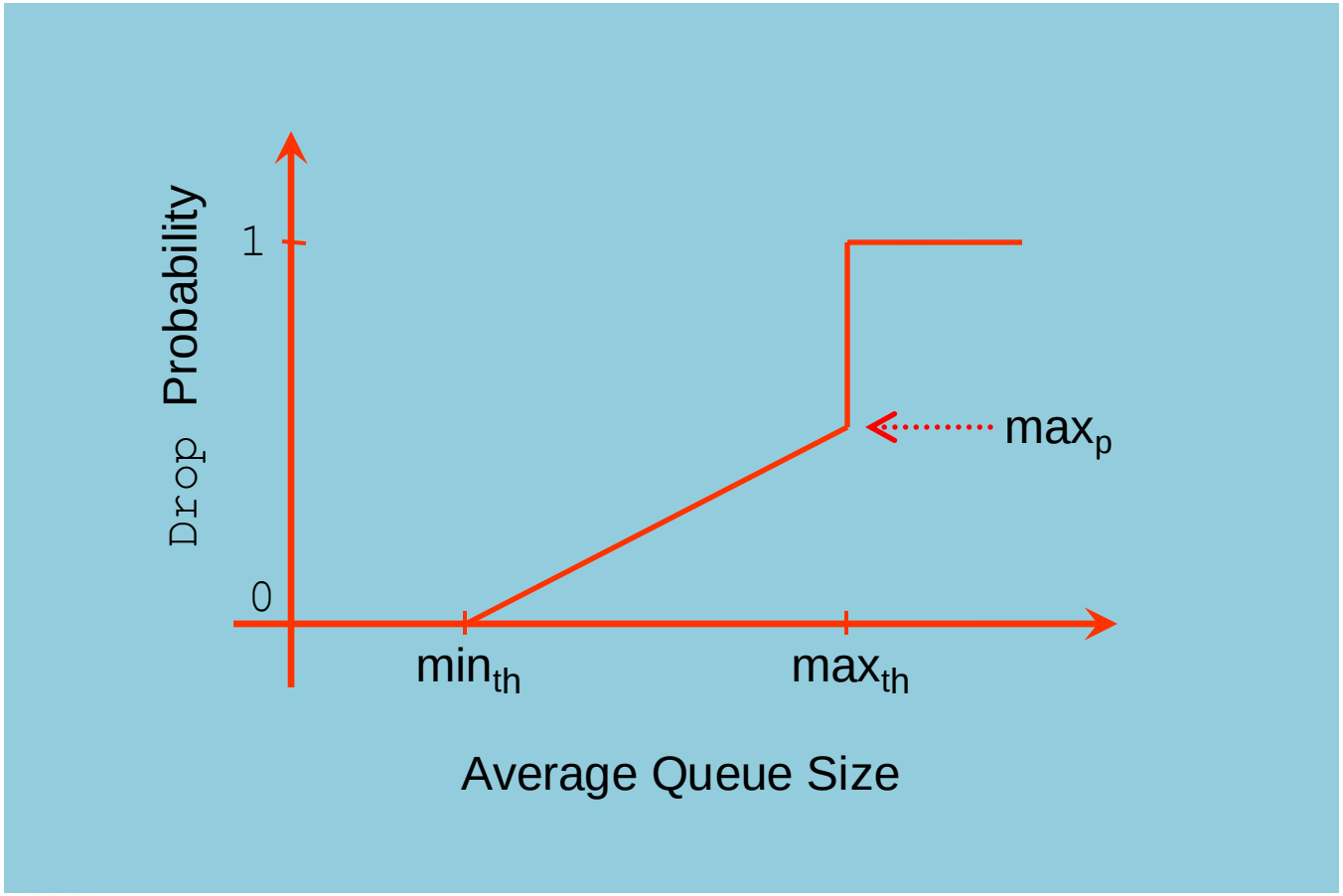
Random Early Detection (RED)

- Router notices that queue is getting full
  - ✓ … and randomly drops packets to signal congestion
- Packet drop probability
  - ✓ Drop probability increases as queue length increases
  - ✓ Else, set drop probability f(avg queue length)

Random Early Detection (RED)



RED Dropping Curve



Properties of RED

- Drops packets before queue is full
  - ✓ In the hope of reducing the rates of some flows
- Drops packet in proportion to each flow's rate
  - ✓ High-rate flows selected more often
- Drops are spaced out in time
  - ✓ Helps desynchronize the TCP senders

Problems With RED

- Hard to get tunable parameters just right
  - ✓ How early to start dropping packets?
  - ✓ What slope for increase in drop probability?
  - ✓ What time scale for averaging queue length?
- RED has mixed adoption in practice
  - ✓ If parameters aren't set right, RED doesn't help
- Many other variations in research community
  - ✓ Names like "Blue" (self-tuning), "FRED"…

Feedback: From loss to notification

- Early dropping of packets
  - ✓ Good: gives early feedback
  - ✓ Bad: has to drop the packet to give the feedback
- Explicit Congestion Notification
  - ✓ Router marks the packet with an ECN bit
  - ✓ Sending host interprets as a sign of congestion

Explicit Congestion Notification

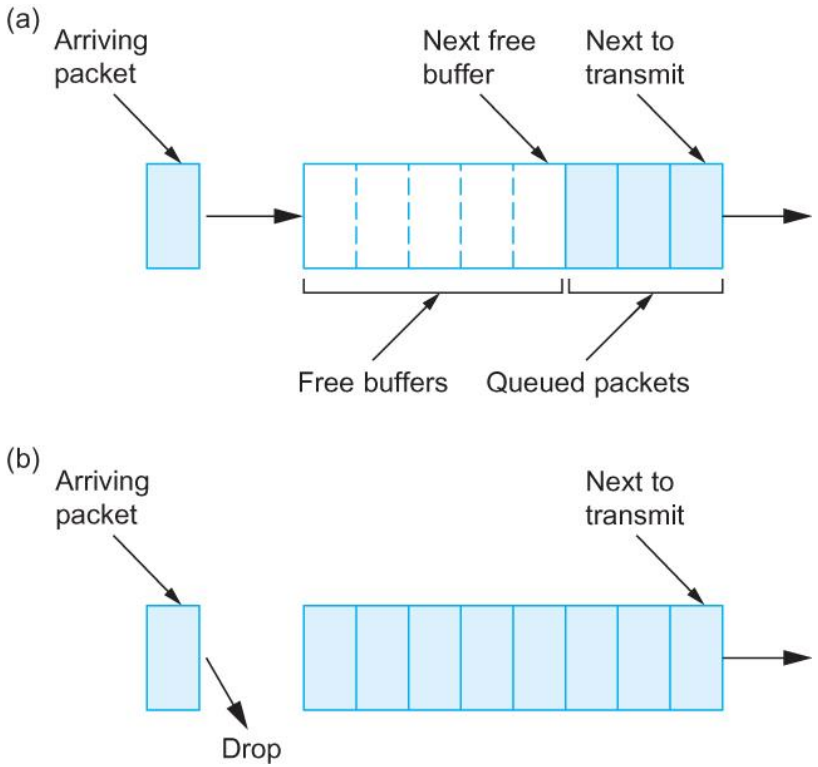
- Must be supported by router, sender, AND receiver
  - ✓ End-hosts determine if ECN-capable during TCP handshake
- ECN involves all three parties (and 4 header bits)
  - ✓ Sender marks “ECN-capable” when sending
  - ✓ If router sees “ECN-capable” and experiencing congestion, router marks packet as “ECN congestion experienced”
  - ✓ If receiver sees “congestion experienced”, marks “ECN echo” flag in responses until congestion ACK’d
  - ✓ If sender sees “ECN echo”, reduces cwnd and marks “congestion window reduced” flag in next TCP packet

Scheduling Discipline

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

First-In First-Out (FIFO)



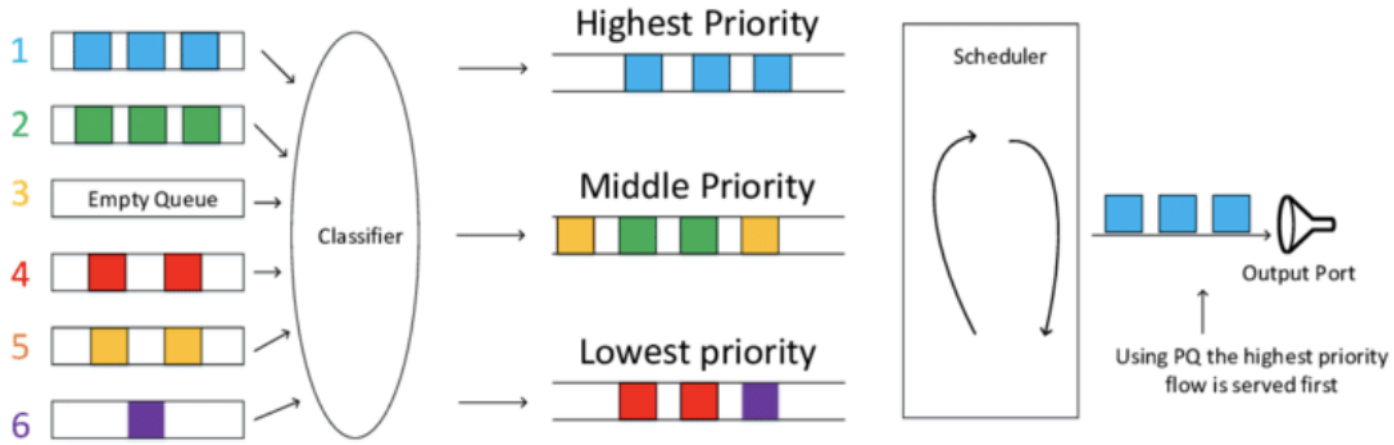
(a) FIFO queuing; (b) tail drop at a FIFO queue.

FIFO Queuing – Priority Queuing

- A simple variation on basic FIFO queuing is priority queuing
  - ✓ Each packet marked with a priority
- The routers then implement multiple FIFO queues, one for each priority class
- Router always transmits packets out of the highest-priority queue if that queue is nonempty before moving on to the next priority queue.
- Within each priority, packets are still managed in a FIFO manner.

Priority Queuing

- Multiple levels of priority
  - ✓ Always transmit high-priority traffic, when present
- But lower priority traffic may starve ☹

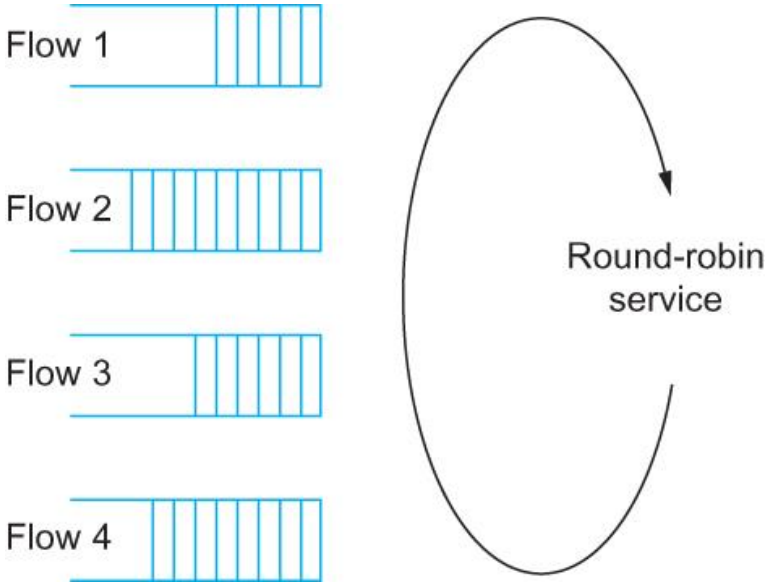


Fair Queuing and Round Robin

- The main problem with FIFO queuing is that it does not discriminate between different traffic sources, or it does not separate packets according to the flow to which they belong.
- Fair queuing (FQ) maintains a separate queue for each flow currently being handled by the router.
  - ✓ The router then services these queues in round-robin algorithm

Fair Queuing - Round-Robin service

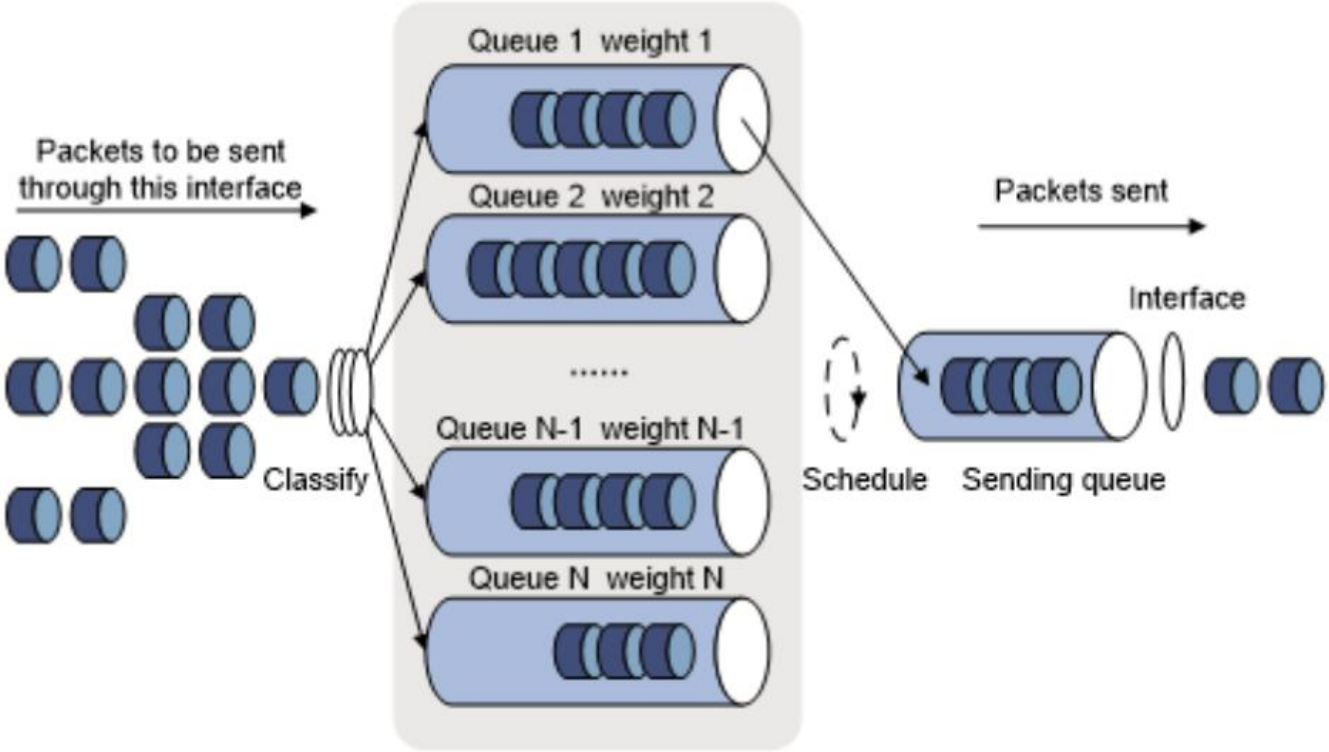
Fair Queuing



Round-robin service of four flows at a router

Weighted Fair Queuing (WFQ)

- allows a weight to be assigned to each flow (queue).

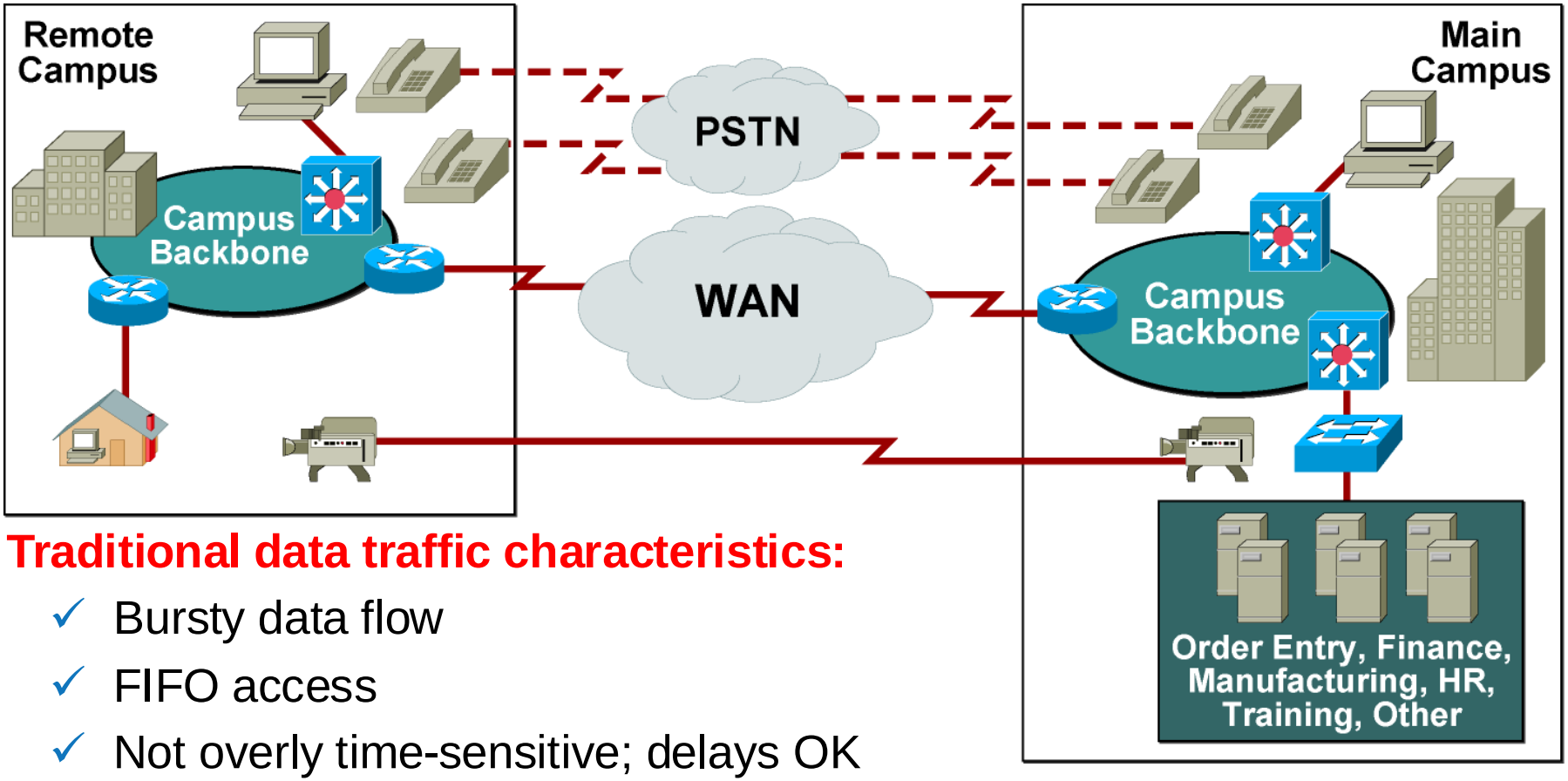


Quality of Service

Related Terms

1. **Bandwidth:** Amount of data transmitted per unit of time (bits per second).
2. **Latency:** Time taken for a packet to travel from source to destination.
3. **Jitter:** Variability in packet delay times (important for real-time applications like voice/video).
4. **Packet Loss:** Percentage of packets that fail to reach their destination.
5. **Reliability:** Consistency in delivering data over the network.

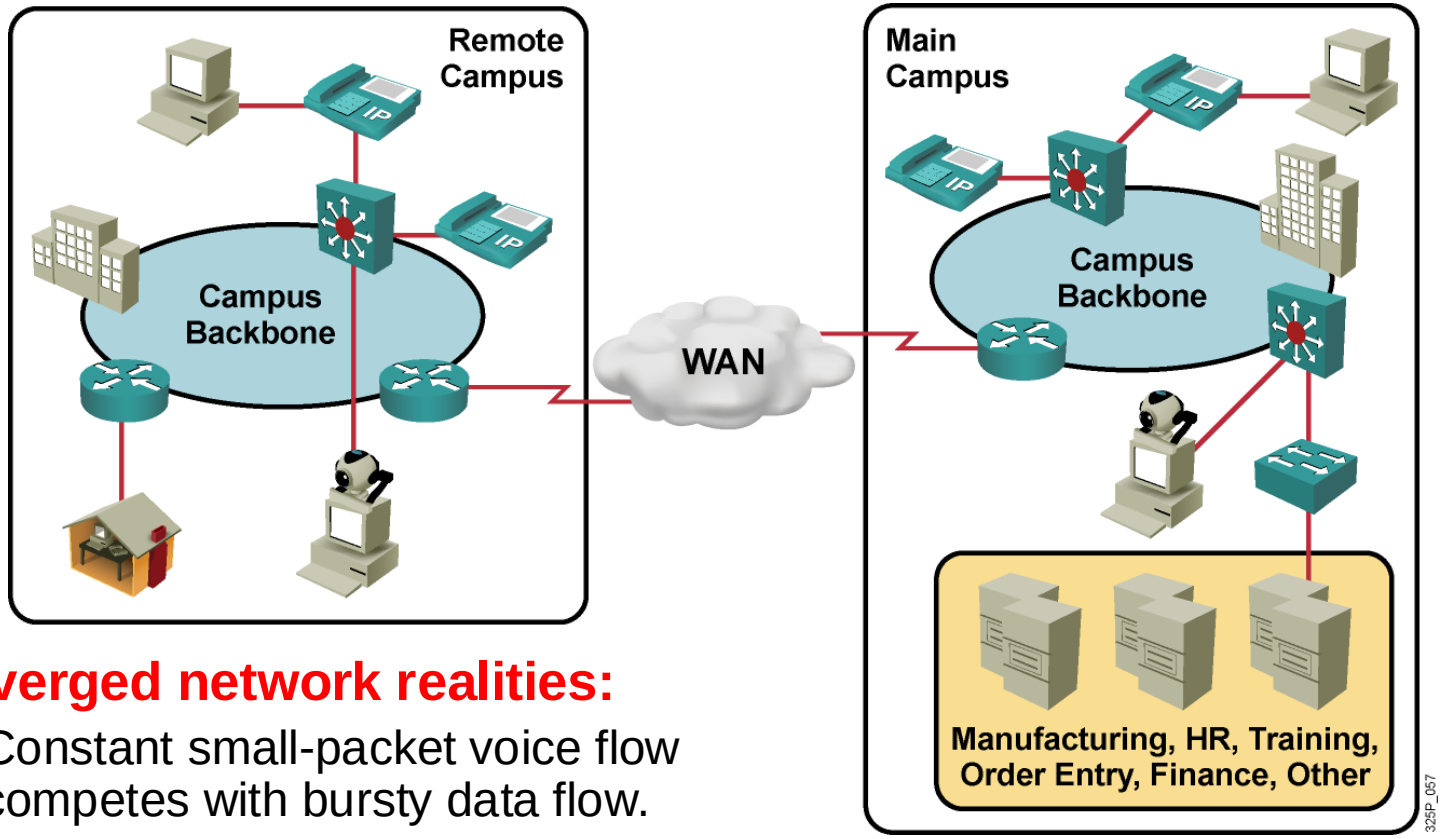
Traditional Non-converged Network



Traditional data traffic characteristics:

- ✓ Bursty data flow
- ✓ FIFO access
- ✓ Not overly time-sensitive; delays OK
- ✓ Brief outages are survivable

Converged Network Realities



Converged network realities:

- ✓ Constant small-packet voice flow competes with bursty data flow.
- ✓ Critical traffic must have priority.
- ✓ Voice and video are time-sensitive.
- ✓ Brief outages are not acceptable.

Converged Network Quality Issues

- Lack of bandwidth: Multiple flows compete for a limited amount of bandwidth.
- End-to-end delay (fixed and variable): Packets have to traverse many network devices and links; this travel adds up to the overall delay.
- Variation of delay (jitter): Sometimes there is a lot of other traffic, which results in varied and increased delay.
- Packet loss: Packets may have to be dropped when a link is congested.

Different Types of Traffic Have Different Needs

Traffic Type	Latency Sensitivity	Jitter Sensitivity	Packet Loss Tolerance	Bandwidth Requirement	Typical Applications	QoS Priority
Real-time (Voice/Video)	Very sensitive (<150ms)	Highly sensitive	Very low (0-1% tolerated)	Moderate to high	VoIP, Video conferencing, Online gaming	High (needs low delay, loss)
Interactive Traffic	Moderate (<300ms)	Moderate	Low (few losses tolerated)	Low to moderate	Web browsing, Email, Instant messaging	Medium (requires responsiveness)
Streaming Media	Moderate	Moderate (can buffer)	Moderate (some losses tolerated)	Moderate to high	Video streaming (YouTube, Netflix), Audio streaming	Medium (can buffer data)
Bulk Data Transfer	Low (delay-tolerant)	Low	High (loss can be retransmitted)	High	File downloads, Backups, Cloud storage, FTP transfers	Low (not time-sensitive)
Background Traffic	Very low	Not sensitive	High (high loss tolerance)	Low to moderate	System updates, Batch processing, Non-urgent data sync	Lowest (best-effort service)

Different Types of Traffic Have Different Needs

- Real-time applications especially sensitive to QoS
  - ✓ Interactive voice
  - ✓ Videoconferencing
- Causes of degraded performance
  - ✓ Congestion losses
  - ✓ Variable queuing delays
- The QoS challenge
  - ✓ Manage bandwidth allocations to deliver the desired application performance
  - ✓ Control delay, jitter, and packet loss

What is Quality of Service ?

- Quality of Service (QoS) refers to the technologies and mechanisms used in networks to ensure certain performance levels for critical applications, such as low-latency for VoIP or guaranteed bandwidth for video streaming.

Why is QoS important?

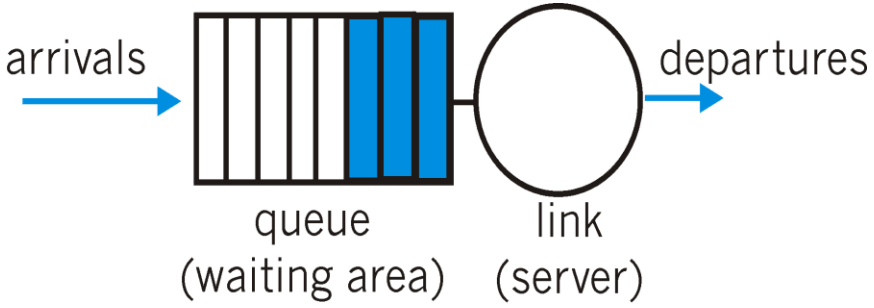
- ✓ Helps in prioritizing critical traffic over less important traffic.
- ✓ Prevents delays, jitter, and packet loss in applications sensitive to such issues.

QoS Mechanisms

- Scheduling
  - ✓ Active Buffer Management
- Traffic Conditioner
  - ✓ Traffic Policing
  - ✓ Traffic Shaping
- Traffic Shaping Algorithms
  - ✓ Leaky Bucket
  - ✓ Token Bucket

Scheduling: How Can Routers Help

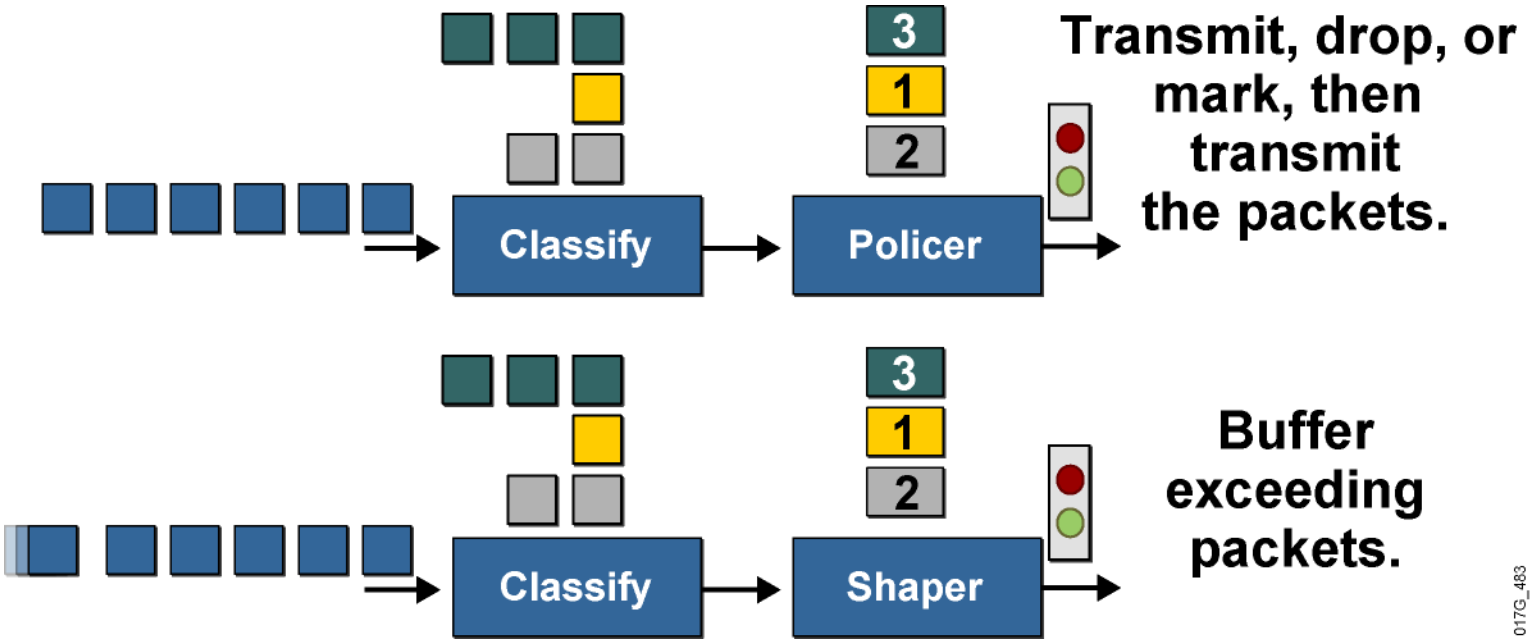
- Scheduling: choosing the next packet for transmission
  - ✓ FIFO/Priority Queue
  - ✓ Round Robin/ DRR
  - ✓ Weighted Fair Queuing
- Packet dropping:
  - ✓ not drop-tail
  - ✓ not only when buffer is full
    - ✓ Active Queue Management
- Congestion signaling
  - ✓ Explicit Congestion Notification (ECN)



Traffic Conditioners

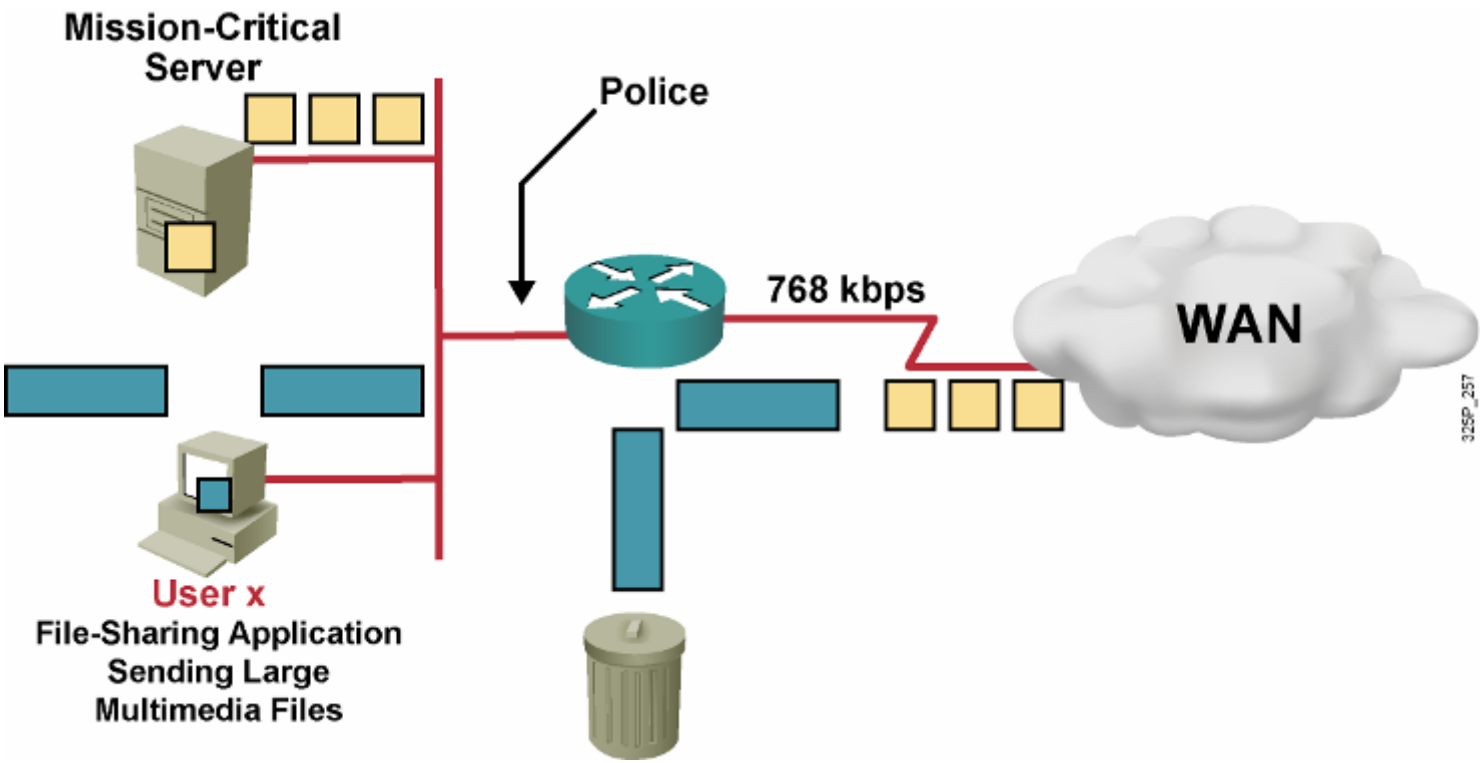
- **Policing**
  - ✓ Limits bandwidth by discarding traffic.
  - ✓ Can re-mark excess traffic and attempt to send.
  - ✓ Should be used on higher-speed interfaces.
  - ✓ Can be applied inbound or outbound.
- **Shaping**
  - ✓ Limits excess traffic by buffering.
  - ✓ Buffering can lead to a delay.
  - ✓ Recommended for slower-speed interfaces.
  - ✓ Cannot re-mark traffic.
  - ✓ Can only be applied in the outbound direction.

Traffic Policing and Shaping



- These mechanisms must classify packets before policing or shaping the traffic rate.
- Traffic policing typically drops or marks excess traffic to stay within a traffic rate limit.
- Traffic shaping queues excess packets to stay within the desired traffic rate.

Traffic Policing Example

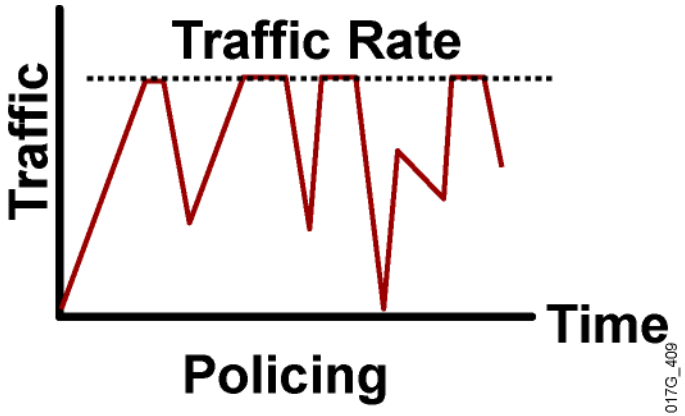


- Do not rate-limit traffic from mission-critical server.
- Rate-limit file-sharing application traffic to 56 kbps.

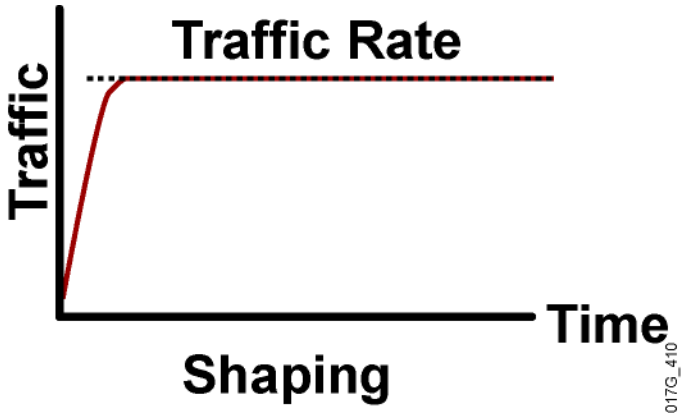
Discussion

Traffic Shaping & Traffic Policing Comparison?

Policing Versus Shaping



- ✓ Incoming and outgoing directions.
- ✓ Out-of-profile packets are dropped.
- ✓ Dropping causes TCP retransmits.
- ✓ Policing supports packet marking or re-marking.



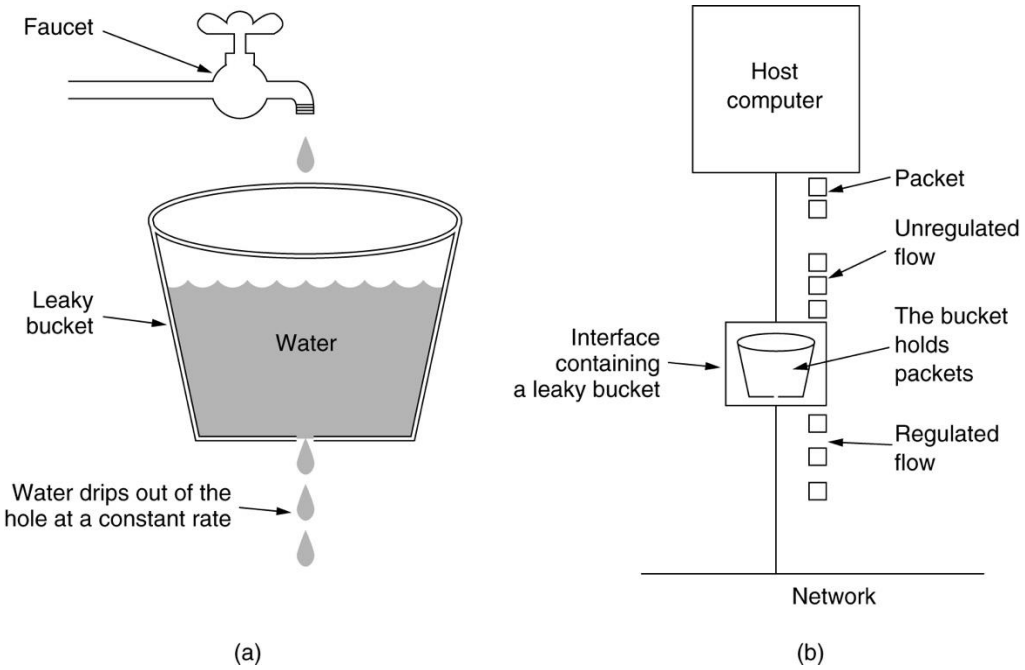
- ✓ Outgoing direction only.
- ✓ Out-of-profile packets are queued until a buffer gets full.
- ✓ Buffering minimizes TCP retransmits.
- ✓ Marking or re-marking not supported.

Traffic Shaping Algorithms

The Leaky Bucket

- The Leaky Bucket Algorithm
  - ✓ used to control rate in a network.
  - ✓ It is implemented as a single-server queue with constant service time.
  - ✓ If the bucket (buffer) overflows then packets are discarded.
- Leaky Bucket (parameters r and B):
  - ✓ Every r time units: send a packet.
  - ✓ For an arriving packet
    - If queue not full then enqueue
- Note that the output is a “perfect” constant rate.

The Leaky Bucket Algorithm



(a) A leaky bucket with water. (b) a leaky bucket with packets.

Discussion

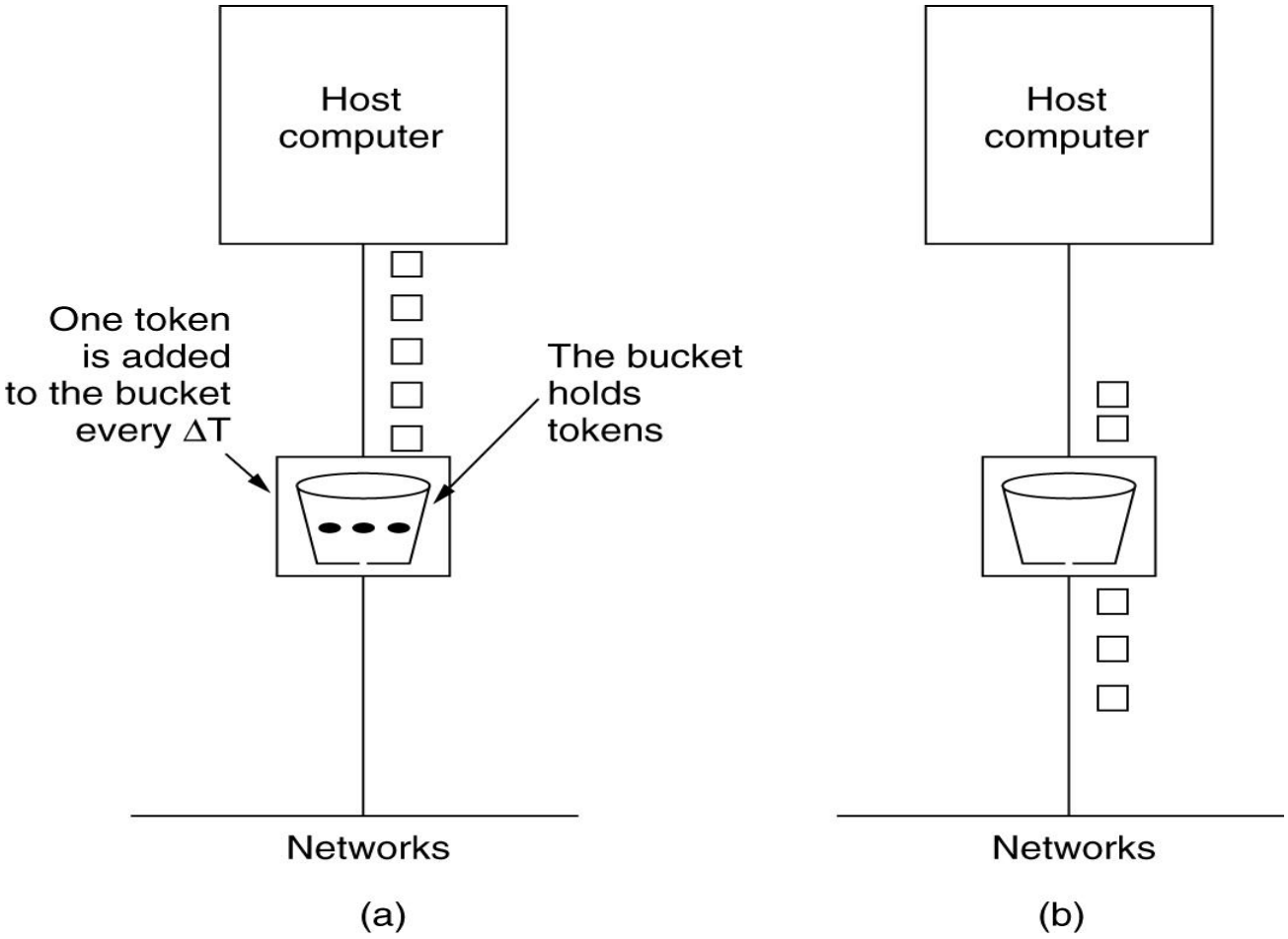
Drawbacks of Leaky Bucket ?

Token Bucket Algorithm

- Highlights:
  - The bucket holds tokens.
  - To transmit a packet, we "use" one token.
- Allows the output rate to vary,
  - depending on the size of the burst.
  - In contrast to the Leaky Bucket
- Granularity
  - Bits or packets

- Token Bucket (r, MaxTokens):
  - Generate r tokens every time unit
    - If number of tokens more than MaxToken, reset to MaxTokens.
  - For an arriving packet: enqueue
  - While buffer not empty and there are tokens:
    - send a packet and discard a token

The Token Bucket Algorithm



(a) Before. (b) After.

Leaky Bucket vs Token Bucket

Leaky Bucket

- Discard:
  - Packets
- Rate:
  - fixed rate (perfect)
- Arriving Burst:
  - Waits in bucket

Token Bucket

- Discard:
  - Tokens
  - Packet management separate
- Rate:
  - Average rate
  - Burst allowed
- Arriving Burst:
  - Can be sent immediately

QoS Models

Three QoS Models

Model	Characteristics
Best effort	No QoS is applied to packets. If it is not important when or how packets arrive, the best-effort model is appropriate.
Integrated Services (IntServ)	Applications signal to the network that the applications require certain QoS parameters.
Differentiated Services (DiffServ)	The network recognizes classes that require QoS.

Best-Effort Model

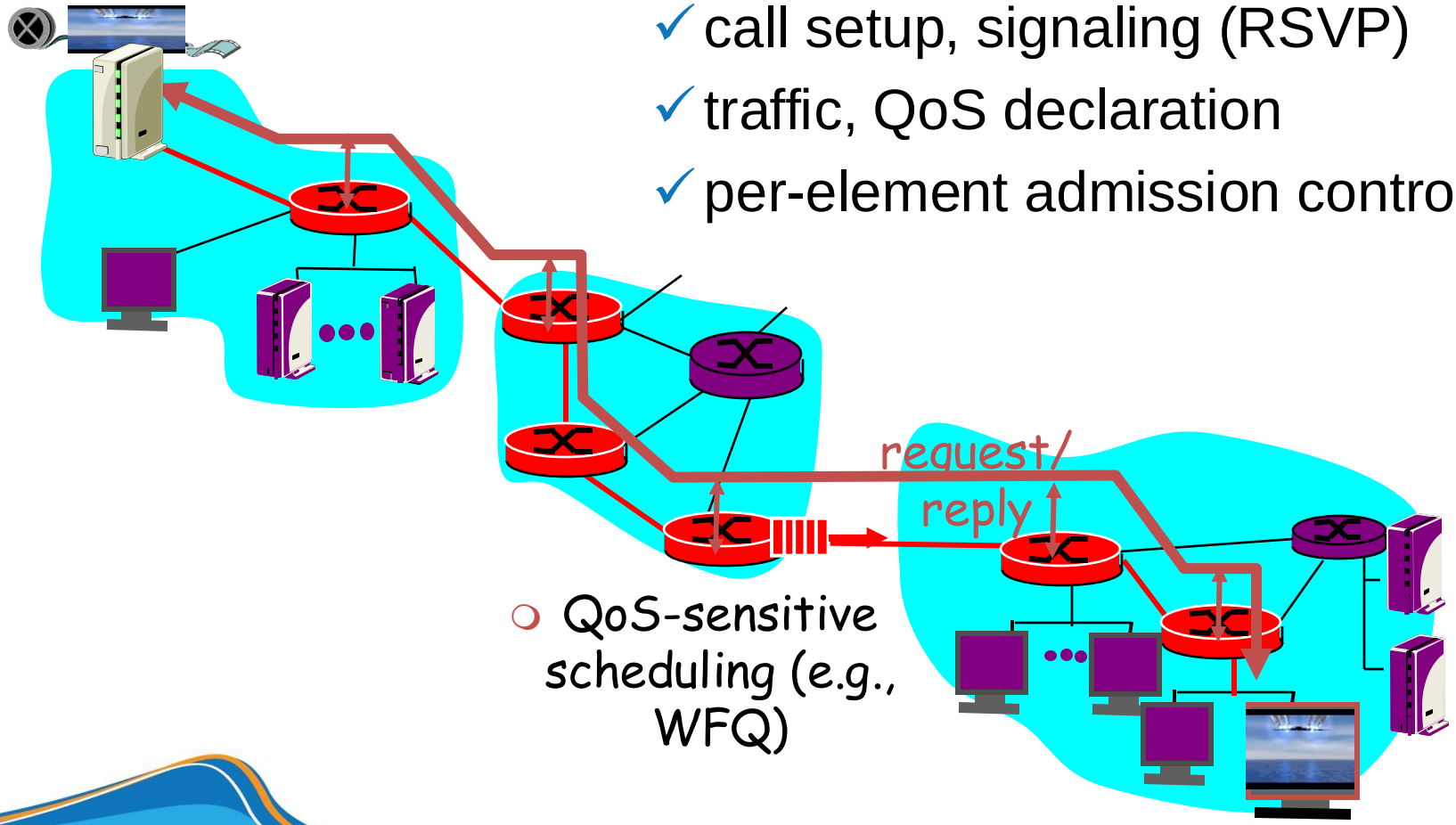
- Internet was initially based on a best-effort packet delivery service.
- Best-effort is the default mode for all traffic.
- There is no differentiation among types of traffic.
- Best-effort model is similar to using standard mail—"The mail will arrive when the mail arrives."
- Benefits:
  - ✓ Highly scalable
  - ✓ No special mechanisms required
- Drawbacks:
  - ✓ No service guarantees
  - ✓ No service differentiation

IETF Integrated Services

- Architecture for providing QOS guarantees in IP networks for individual application sessions
- Resource reservation: routers maintain state info (a la VC) of allocated resources, QoS req's
- Admit/deny new call setup requests

Intserv: QoS guarantee scenario

- Resource reservation
  - ✓ call setup, signaling (RSVP)
  - ✓ traffic, QoS declaration
  - ✓ per-element admission control



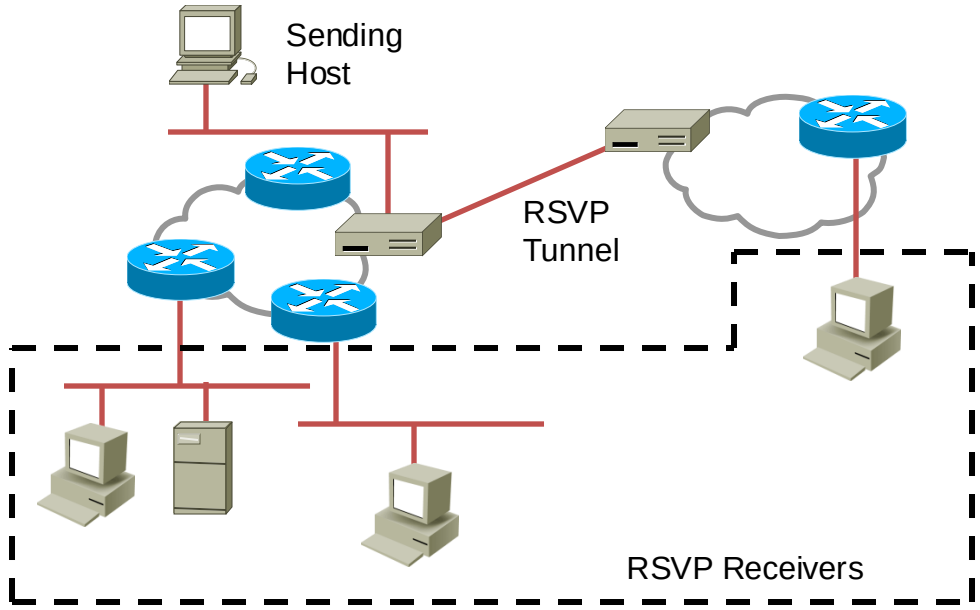
Call Admission

Arriving session must :

- declare its QoS requirement
  - ✓ R-spec: defines the QoS being requested
- characterize traffic it will send into network
  - ✓ T-spec: defines traffic characteristics
  - ✓ signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
  - ✓ RSVP

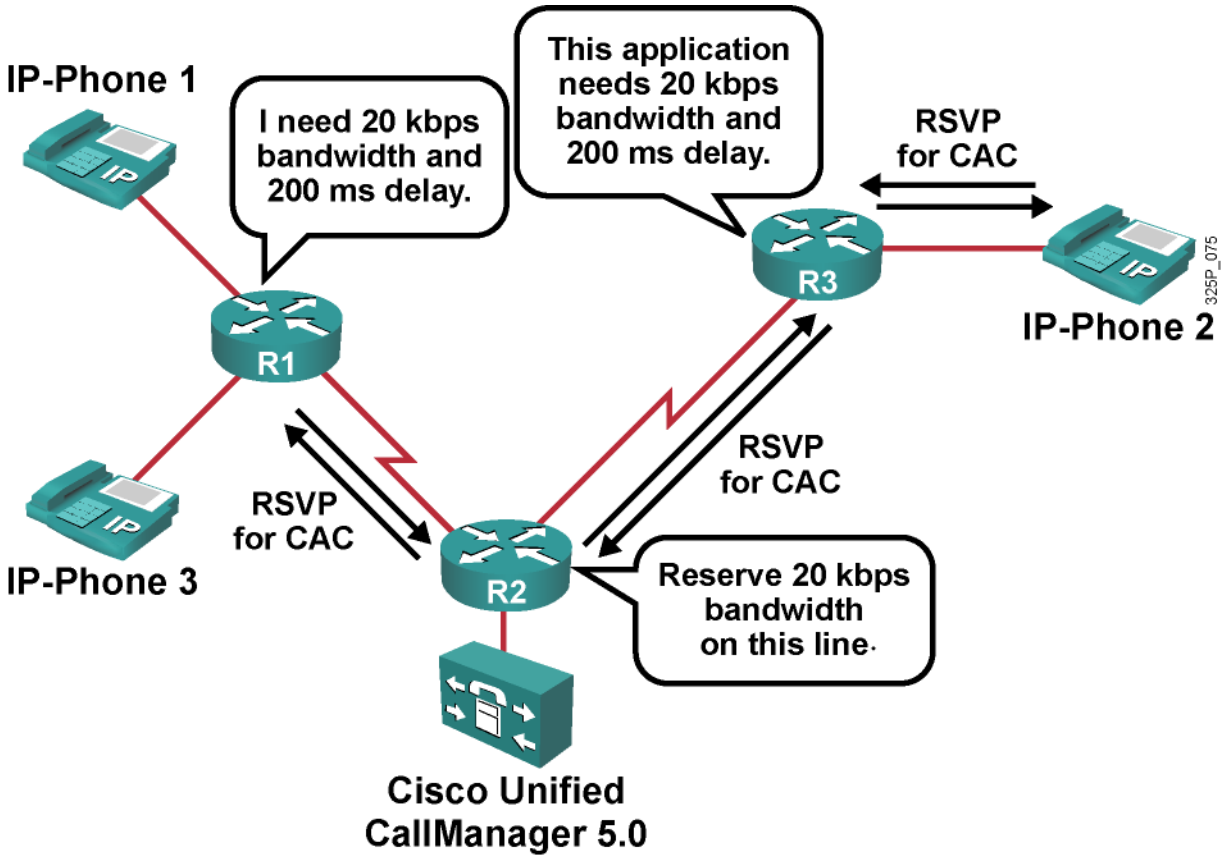
Resource Reservation Protocol (RSVP)

- Is carried in IP—protocol ID 46
- Can use both TCP and UDP port 3455
- Is a signaling protocol and works with existing routing protocols
- Requests QoS parameters from all devices between the source and destination



- Provides divergent performance requirements for multimedia applications:
  - ✓ Rate-sensitive traffic
  - ✓ Delay-sensitive traffic

RSVP in Action



- RSVP sets up a path through the network with the requested QoS.
- RSVP is used for CAC in Cisco Unified CallManager 5.0.

Benefits and Drawbacks

- Benefits:
  - Explicit resource admission control (end to end)
  - Per-request policy admission control (authorization object, policy object)
- Drawbacks:
  - Continuous signaling because of stateful architecture
  - Flow-based approach not scalable to large implementations, such as the public Internet

IETF Differentiated Services

Concerns with Intserv:

- Scalability: signaling, maintaining per-flow router state difficult with large number of flows
- Flexible Service Models: Intserv has only two classes. Also want “qualitative” service classes

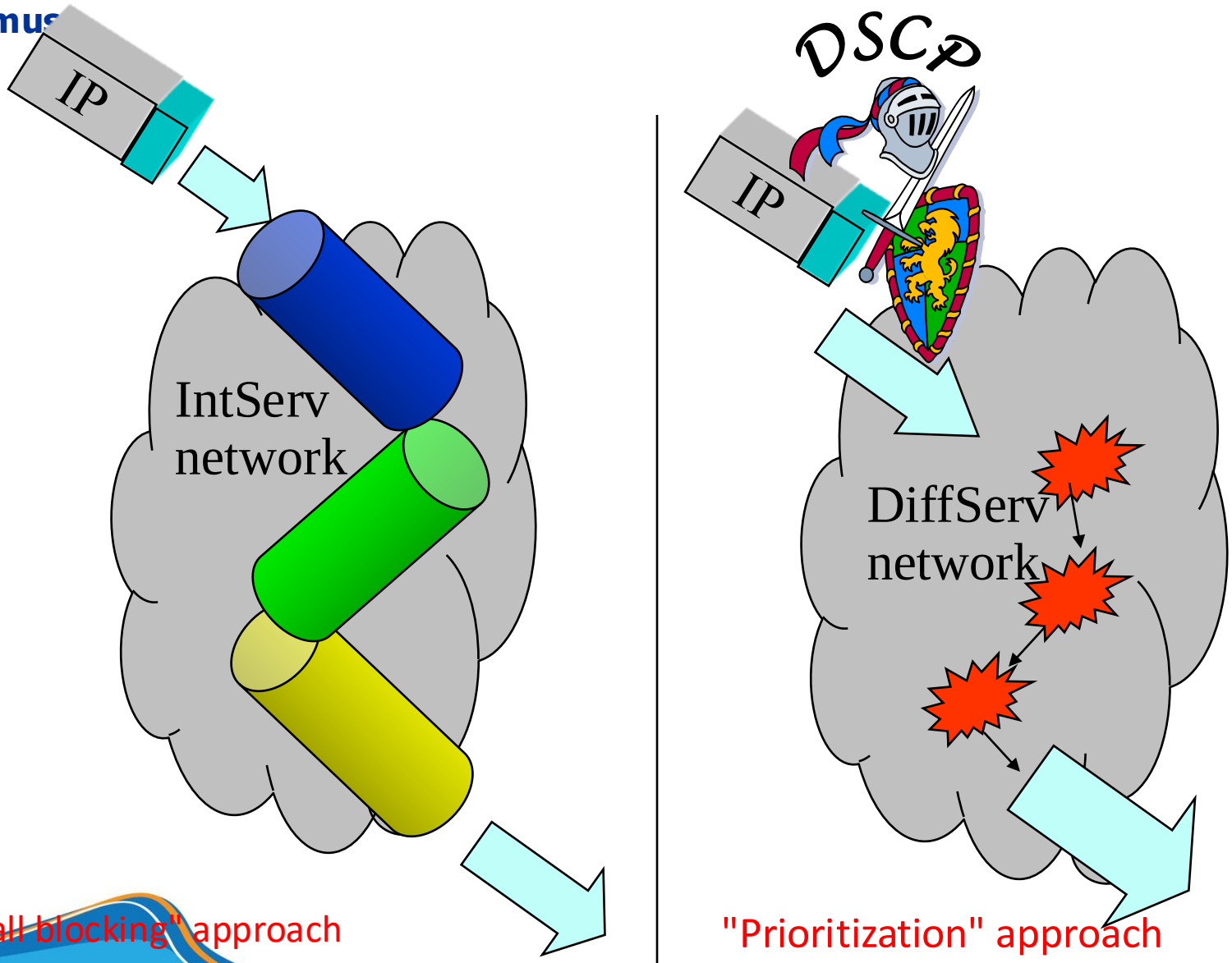
Diffserv approach:

- Simple functions in network core, relatively complex functions at edge routers (or hosts)
- Don’t define service classes, provide functional components to build service classes

DiffServ Model

- Describes services associated with traffic classes, rather than traffic flows.
- Complex traffic classification and conditioning is performed at the network edge.
- No per-flow state in the core.
- The goal of the DiffServ model is scalability.
- Interoperability with non-DiffServ-compliant nodes

IntServ vs. DiffServ

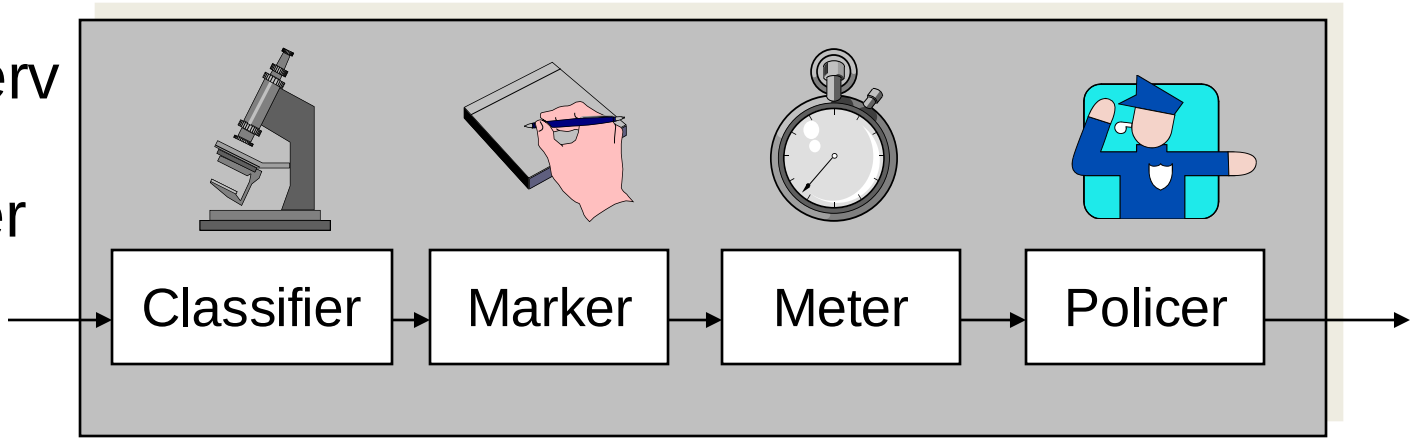


"Call blocking" approach

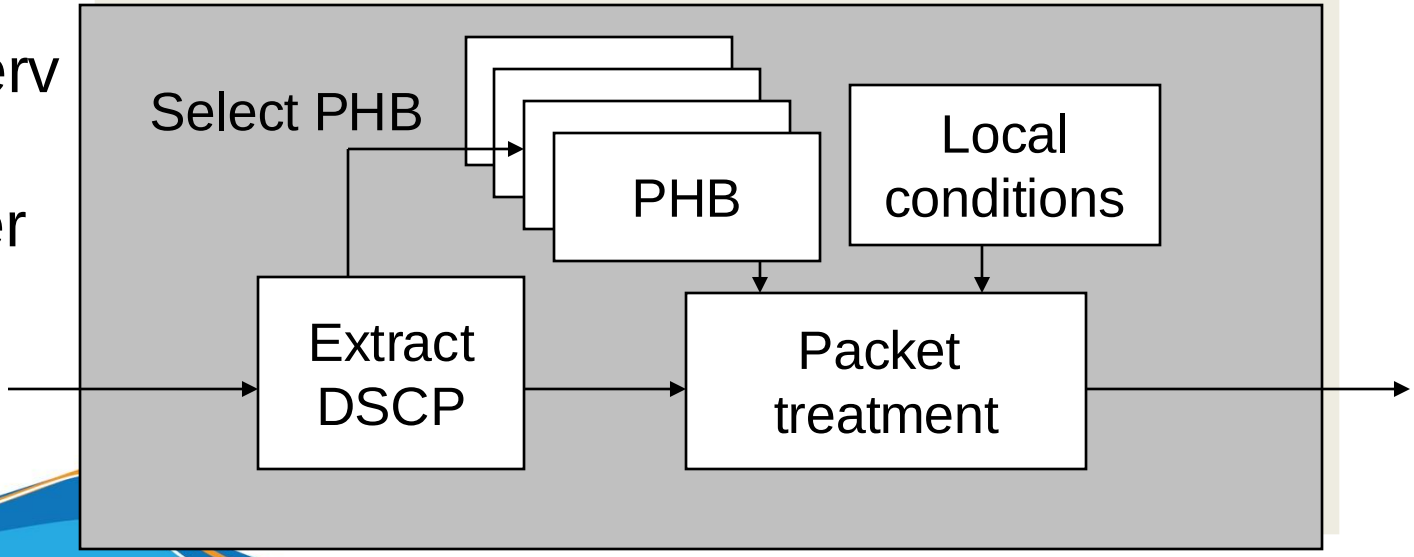
"Prioritization" approach

DiffServ Routers

DiffServ Edge Router



DiffServ Core Router



Classification

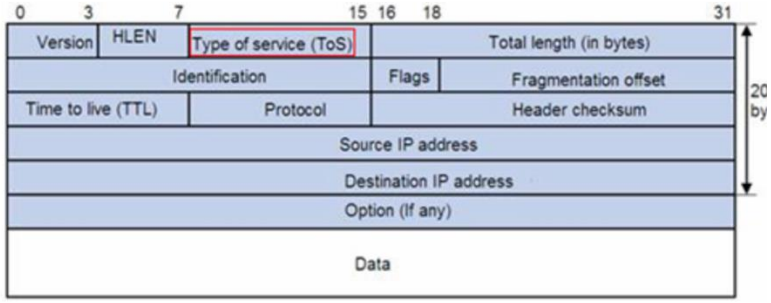
- Classification is the process of identifying and categorizing traffic into classes, typically based upon:
  - ✓ Incoming interface
  - ✓ IP precedence
  - ✓ DSCP
  - ✓ Source or destination address
  - ✓ Application
- Without classification, all packets are treated the same.
- Classification should take place as close to the source as possible.

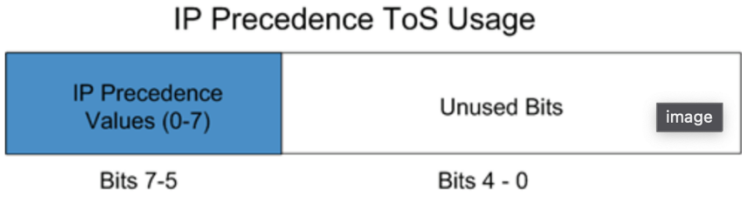
Marking

- Marking is the QoS feature component that “colors” a packet (frame) so it can be identified and distinguished from other packets (frames) in QoS treatment.
- Commonly used markers:
  - ✓ Link layer:
    - CoS (ISL, 802.1p)
    - MPLS EXP bits
    - Frame Relay
  - ✓ Network layer:
    - DSCP
    - IP precedence

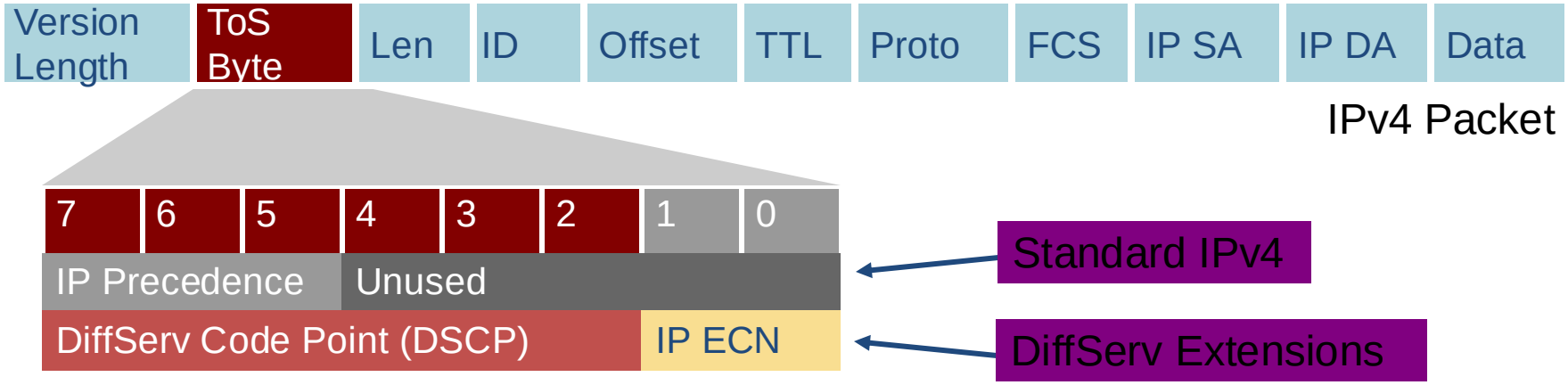
IP Precedence

- An older method used to indicate the priority of a packet in IP networks. It is part of the Type of Service (ToS) field in the IPv4 header.
- The first 3 bits of this field are used for IP Precedence. These 3 bits can define 8 different priority levels, ranging from 0 (best-effort) to 7 (highest priority).
- **Limitations of IP Precedence:**
  - Only define 8 levels of priority, which limits the flexibility and granularity of traffic management.



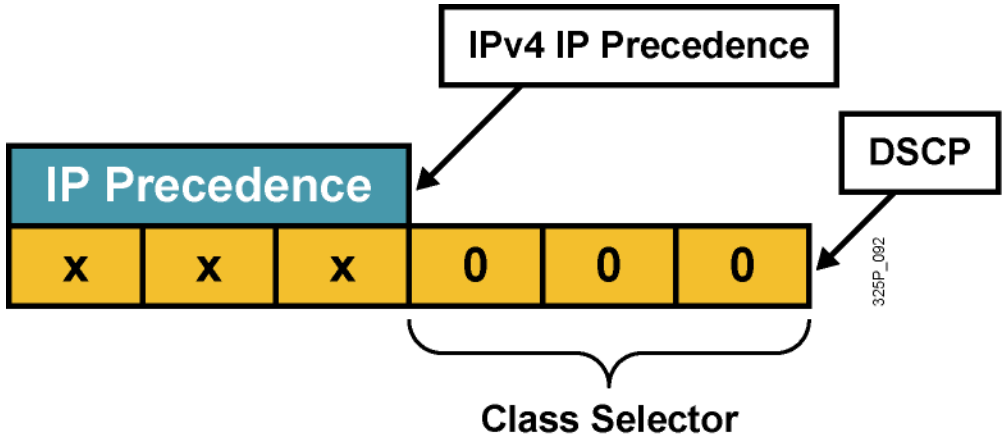


IP Precedence and DiffServ Code Points



- IPv4: three most significant bits of ToS byte are called IP Precedence (IPP)—other bits unused
- DiffServ: six most significant bits of ToS byte are called DiffServ Code Point (DSCP)—remaining two bits used for flow control
- DSCP is backward-compatible with IP precedence

IP **Precedence** and DSCP Compatibility



- **Backward Compatibility:** The first **3 bits of DSCP** are the same as the **IP Precedence** bits, which ensures compatibility.
- Packets marked with IP Precedence can still be understood by DSCP-based networks.
- **Mapping IP Precedence to DSCP:** IP Precedence values are mapped directly to DSCP by setting the last 3 bits to **000**.
  - **Example:** IP Precedence value **101** (priority 5) is equivalent to **DSCP 101000**.

Coexistence in Mixed Environments

•**IP Precedence devices** (legacy) read the first 3 bits.

•**DSCP devices** read the full 6 bits but are backward compatible with IP Precedence.

Both older IP Precedence-based devices and newer DSCP-based devices can operate within the same network, ensuring that traffic is prioritized correctly.

Behavior Aggregator, Per-Hop Behaviors

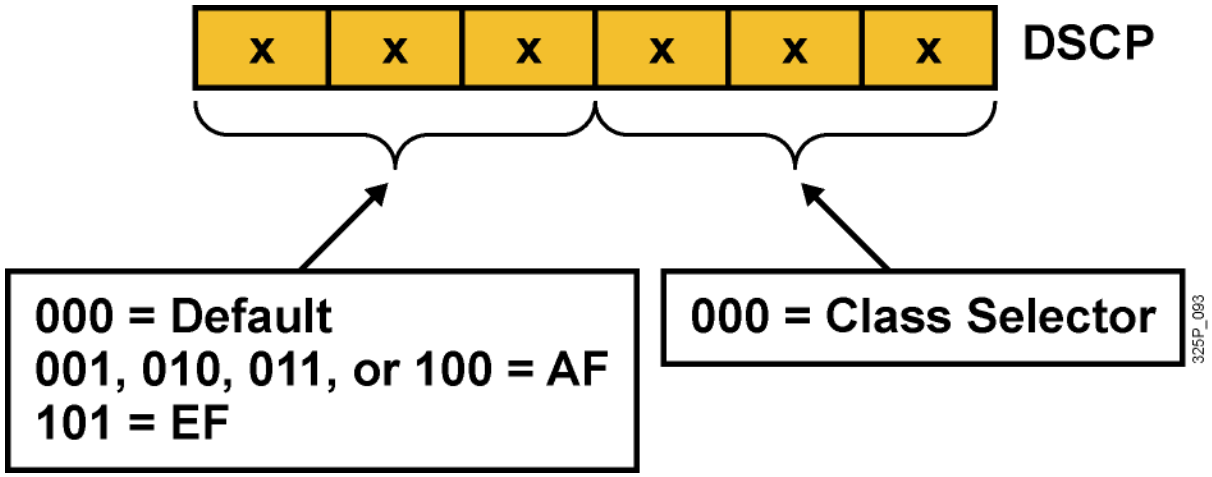
- Behavior Aggregate (BA): the collection of packets that have the same DSCP value (also called a codepoint) and crossing in a particular direction.
- Per Hop Behavior (PHB): the externally observable forwarding behavior applied at a DS-compliant node to a DS BA.

→ *PHB refers to the packet scheduling, queuing, policing, or shaping behavior of a node on any given packet belonging to a BA, and as configured by a Service Level Agreement (SLA) or policy.*

Per-Hop Behaviors

- *Default Forwarding* (DF) PHB — which is typically best-effort traffic
- *Expedited Forwarding* (EF) PHB — dedicated to low-loss, low-latency traffic
- *Assured Forwarding* (AF) PHB — gives assurance of delivery under prescribed conditions
- *Class Selector* PHBs — which maintain backward compatibility with the IP precedence field.

Per-Hop Behaviors



DSCP selects PHB throughout the network:

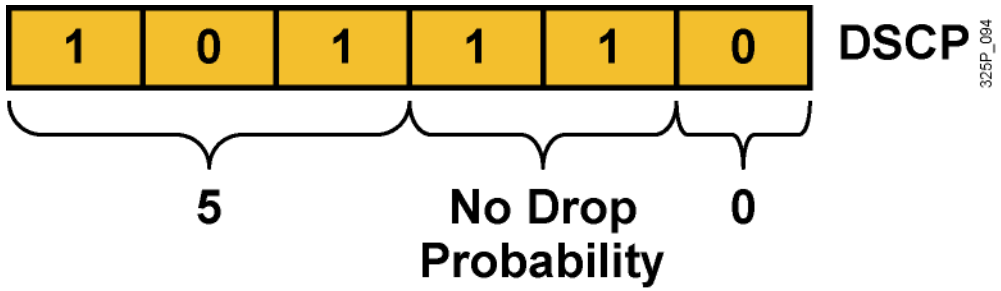
- ✓ Default PHB (FIFO, tail drop)
- ✓ Class-selector PHB (IP precedence)
- ✓ EF PHB
- ✓ AF PHB

Standard PHB Groups

PHB				DSCP			Maps to IP Precedence
Default (Best Effort)				0 000000			0
Scavenger (Less-than-Best-Effort)				8 001000			1
Assured Forwarding	Low Drop Pref.	Med Drop Pref.	High Drop Pref.				
Class 1	AF11	AF12	AF13	10 001010	12 001100	14 001110	1
Class 2	AF21	AF22	AF23	18 010010	20 010100	22 010110	2
Class 3	AF31	AF32	AF33	26 011010	28 011100	30 011110	3
Class 4	AF41	AF42	AF43	34 100010	36 100100	38 100110	4
Expedited Forwarding	EF			46 101110			5

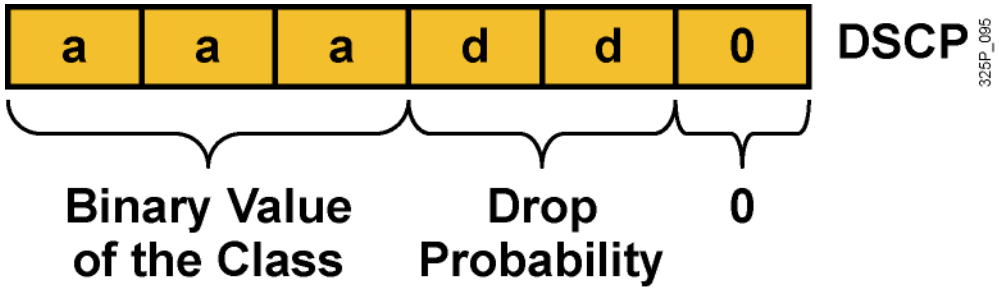
325P_098

Expedited Forwarding (EF) PHB



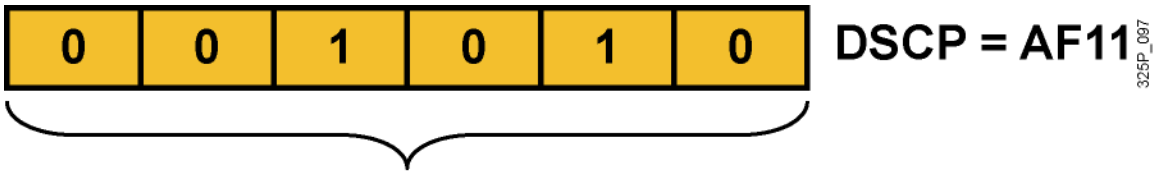
- EF PHB:
  - ✓ Ensures a minimum departure rate
  - ✓ Guarantees bandwidth—class guaranteed an amount of bandwidth with prioritized forwarding
  - ✓ Polices bandwidth—class not allowed to exceed the guaranteed amount (excess traffic is dropped)
- DSCP value of 101110: Looks like IP precedence 5 to non-DiffServ-compliant devices:
  - ✓ Bits 5 to 7: 101 = 5 (same 3 bits are used for IP precedence)
  - ✓ Bits 3 and 4: 11 = No drop probability
  - ✓ Bit 2: Just 0

Assured Forwarding (AF) PHB



- AF PHB:
  - ✓ Guarantees bandwidth
  - ✓ Allows access to extra bandwidth, if available
- Four standard classes: AF1, AF2, AF3, and AF4
- DSCP value range of aaadd0:
  - ✓ aaa is a binary value of the class
  - ✓ dd is drop probability

AF PHB Values

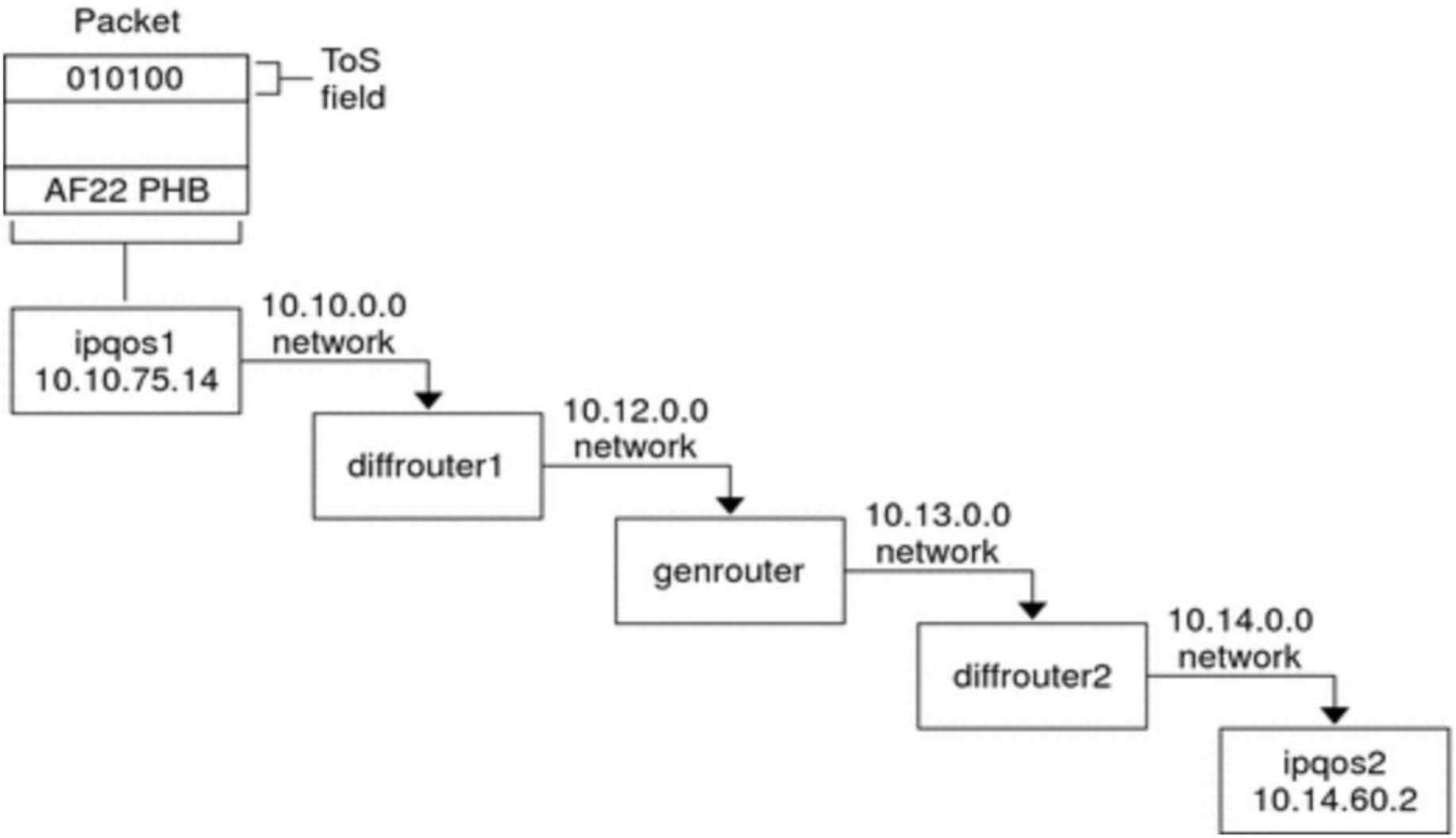


Class	Value		
AF1	001	dd	
AF2	010	dd	0
AF3	011	dd	0
AF4	100	dd	0

Drop Probability (dd)	Value	AF Value
Low	01	AF11
Medium	10	AF12
High	11	AF13

- Each AF class uses three DSCP values.
- Each AF class is independently forwarded with its guaranteed bandwidth.
- Congestion avoidance is used within each class to prevent congestion within the class.

Example


Packet
010100
ToS field
AF22 PHB
ipqos1
10.10.75.14
10.10.0.0
network
diffrouter1
10.12.0.0
network
genrouter
10.13.0.0
network
diffrouter2
10.14.0.0
network
ipqos2
10.14.60.2

Example

1. The user on ipqos1 runs the ftp command to access host ipqos2, which is three hops away.
2. ipqos1 applies its QoS policy to the resulting packet flow. ipqos1 then successfully classifies the ftp traffic.
3. The system administrator has created a class for all outgoing ftp traffic that originates on the local network 10.10.0.0. Traffic for the ftp class is assigned the AF22 per-hop behavior: class two, medium-drop precedence. A traffic flow rate of 2Mb/sec is configured for the ftp class.
4. ipqos-1 meters the ftp flow to determine if the flow exceeds the committed rate of 2 Mbit/sec.
5. The marker on ipqos1 marks the DS fields in the outgoing ftp packets with the 010100 DSCP, corresponding to the AF22 PHB.

Example

6. The router diffrouter1 receives the ftp packets. diffrouter1 then checks the DSCP. If diffrouter1 is congested, packets that are marked with AF22 are dropped.

7. ftp traffic is forwarded to the next hop in agreement with the per-hop behavior that is configured for AF22 in diffrouter1's files.

8. The ftp traffic traverses network 10.12.0.0 to genrouter, which is not Diffserv aware. As a result, the traffic receives “best-effort” forwarding behavior.

9. genrouter passes the ftp traffic to network 10.13.0.0, where the traffic is received by diffrouter2.

10. diffrouter2 is Diffserv aware. Therefore, the router forwards the ftp packets to the network in agreement with the PHB that is defined in the router policy for AF22 packets.

11. ipqos2 receives the ftp traffic. ipqos2 then prompts the user on ipqos1 for a user name and password.

cdio 4.0

fit@hcmus

IntServ and DiffServ

Feature	IntServ (Integrated Services)	DiffServ (Differentiated Services)
QoS Guarantee	Strict, guaranteed per flow	Relative prioritization, no strict guarantees
Traffic Management	Per-flow	Per-class
Resource Reservation	Yes (using RSVP)	No resource reservation
Packet Handling	Individual flow handling	Aggregated traffic class handling
Scalability	Low	High
Complexity	High	Medium
Implementation Cost	High	Lower than IntServ
Use Case	Real-time sensitive applications	Large-scale enterprise/service provider networks
Guarantee of Bandwidth	Yes	No, but higher priority for marked packets
Protocol	Uses RSVP for resource reservation	Uses DSCP for marking and prioritization
Congestion Control	Guaranteed through reservations	Managed through priority levels

Questions ?

Additional Slides

Leaky Bucket

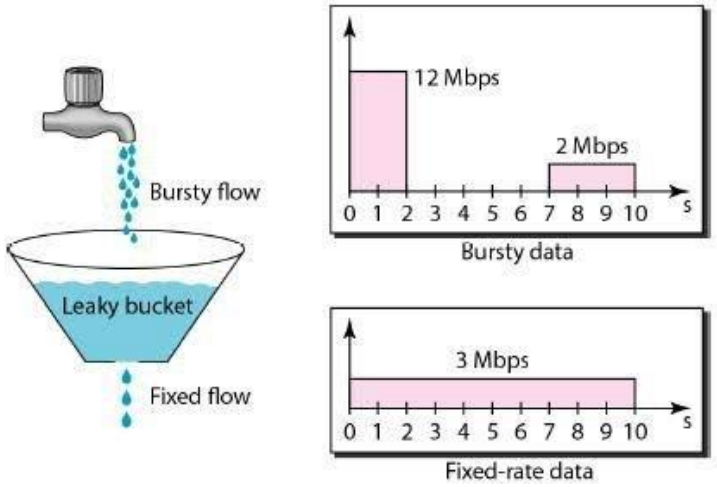
Main working steps

1.When the host has to send a packet , packet is thrown in bucket.

2.Bucket leaks at constant rate.

3.Bursty traffic is converted into uniform traffic by leaky bucket.

4.In practice bucket is a finite queue outputs at finite rate.



Token Bucket :

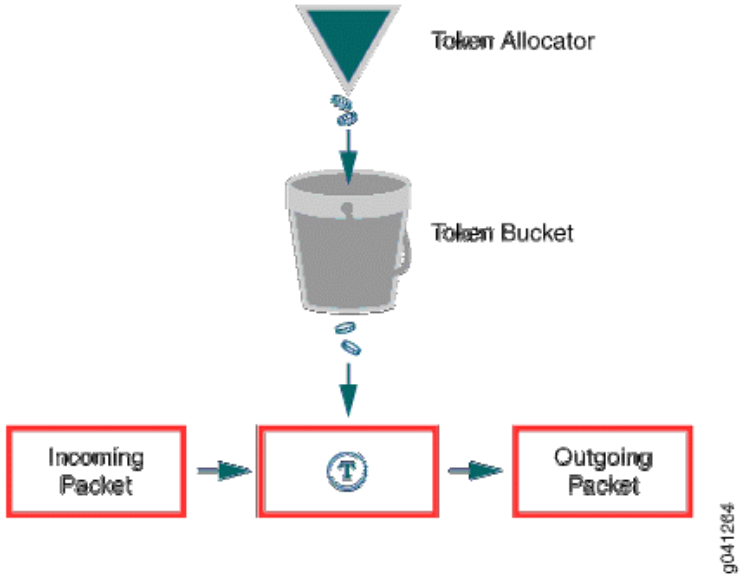
Main working steps

1.In this token bucket holds tokens generated at regular intervals of time.

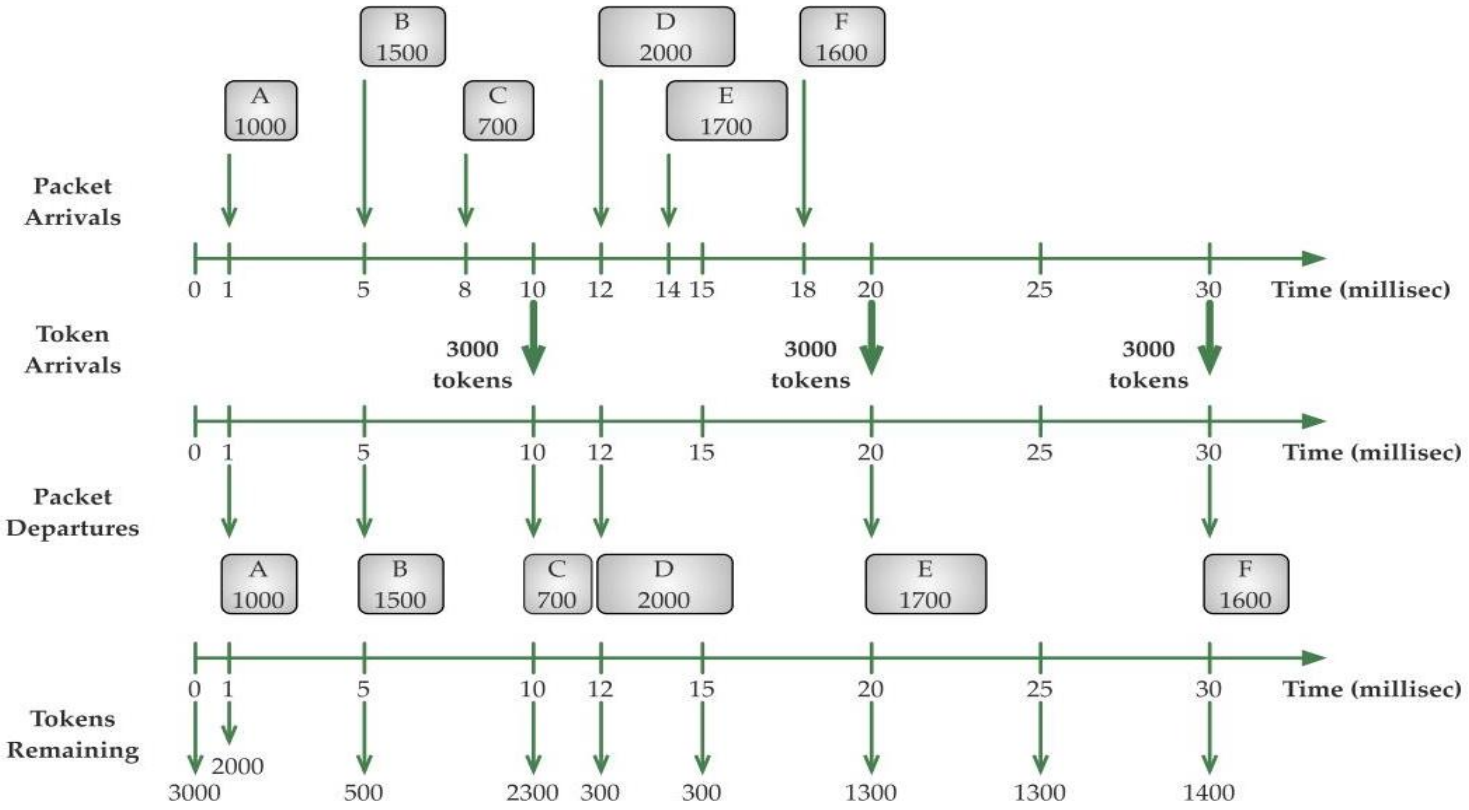
2.Bucket has maximum capacity.

3.If there is a ready packet, a token is removed from bucket and packet is send.

4.If there is a no token in bucket, packet can not be send.



Examples


Packet Arrivals
A 1000
B 1500
C 700
D 2000
E 1700
F 1600
0 1 5 8 10 12 14 15 18 20 25 30 Time (millisec)
Token Arrivals
3000 tokens
3000 tokens
3000 tokens
0 1 5 10 12 15 20 25 30 Time (millisec)
Packet Departures
A 1000
B 1500
C 700
D 2000
E 1700
F 1600
0 1 5 10 12 15 20 25 30 Time (millisec)
Tokens Remaining
3000
2000
500
2300
300
300
1300
1300
1400

Comparision

1.If bucket is full in token Bucket, tokens are discarded not packets.
While in leaky bucket, packets are discarded.

2. Token Bucket can send Large bursts can faster rate while leaky bucket always sends packets at constant rate.