QoS
IntServ and DiffServ

universidade de aveiro

deti.ua.pt

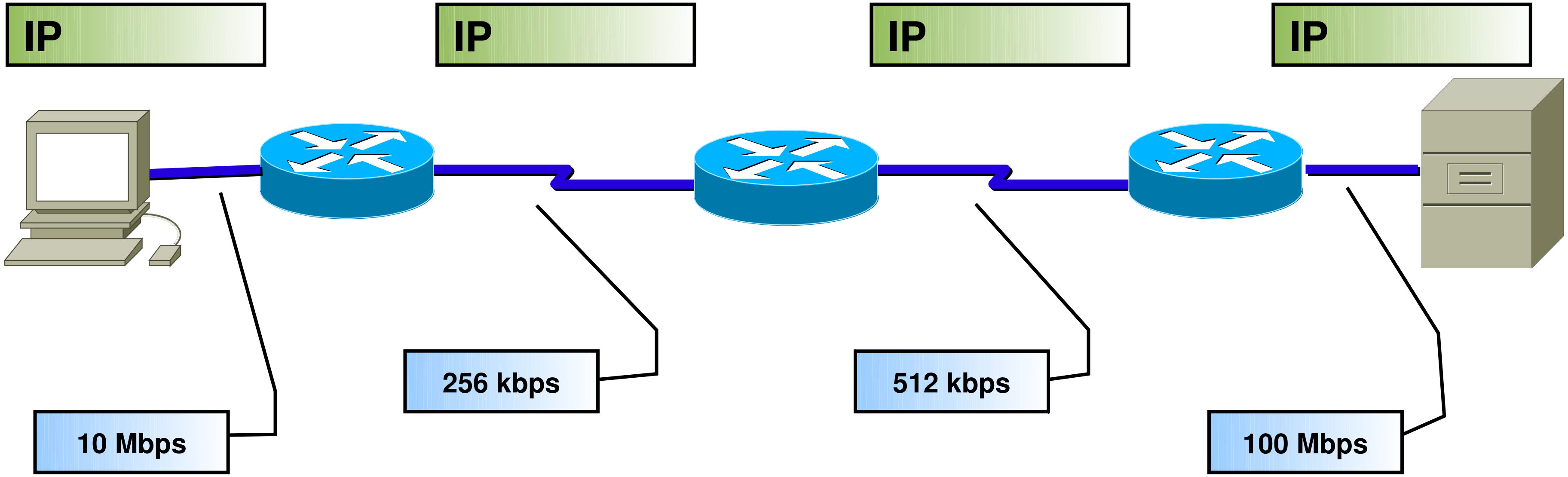
Why IP QoS? Because ...

- Application X is slow! (not enough BANDWIDTH)
- Video broadcast occasionally stalls! (DELAY temporarily increases – JITTER)
- Phone calls over IP are no better than over satellite! (too much DELAY)
- Phone calls have really bad voice quality! (too many phone calls – ADMISSION CONTROL)
- ATM (the money-dispensing-type) are non responsive! (too many DROPs)
- ...

What Causes ...

- Lack of bandwidth – multiple flows are contesting for a limited amount of bandwidth
- Too much delay – packets have to traverse many network devices and links that add up to the overall delay
- Variable delay – sometimes there is a lot of other traffic which results in more delay
- Drops – packets have to be dropped when a link is congested

Available Bandwidth

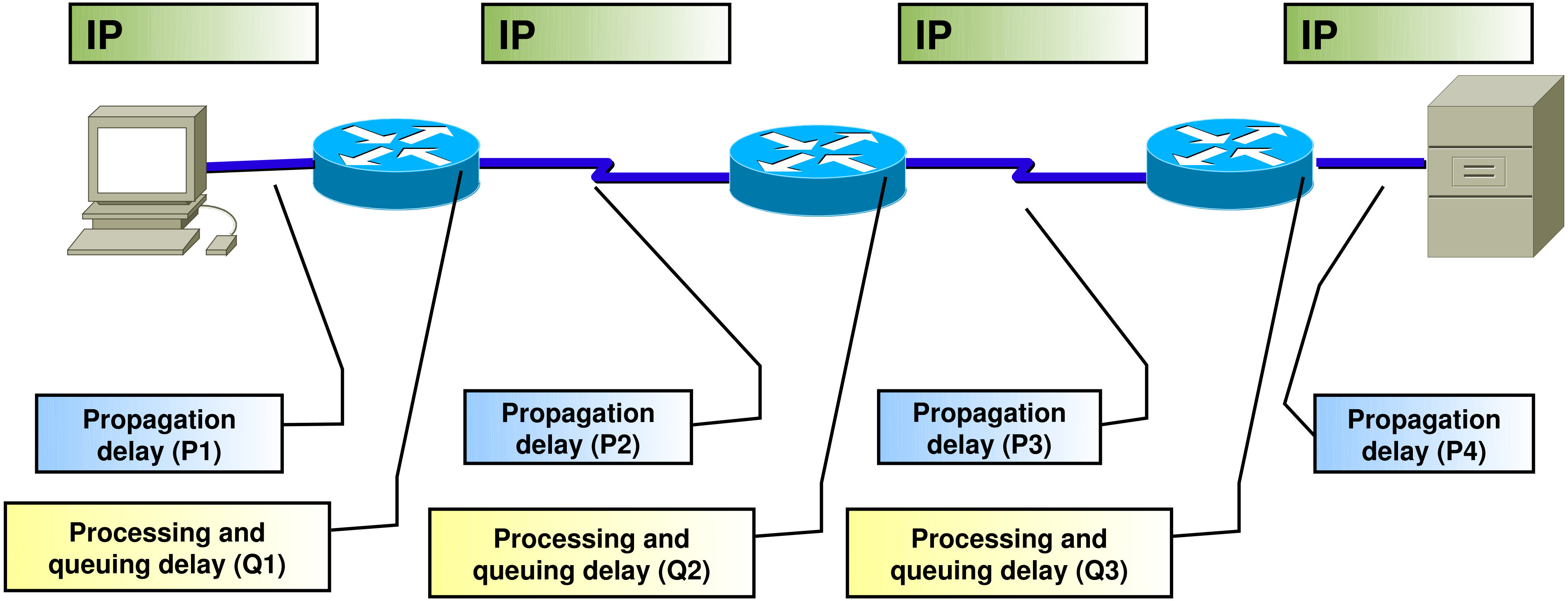


$$BW_{max} = \min(10M, 256k, 512k, 100M) = 256kbps$$

$$BW_{avail} = BW_{max} / Flows$$

- Maximum available bandwidth equals the bandwidth of the weakest link
- Multiple flows are contesting for the same bandwidth resulting in much less bandwidth being available to one single application.

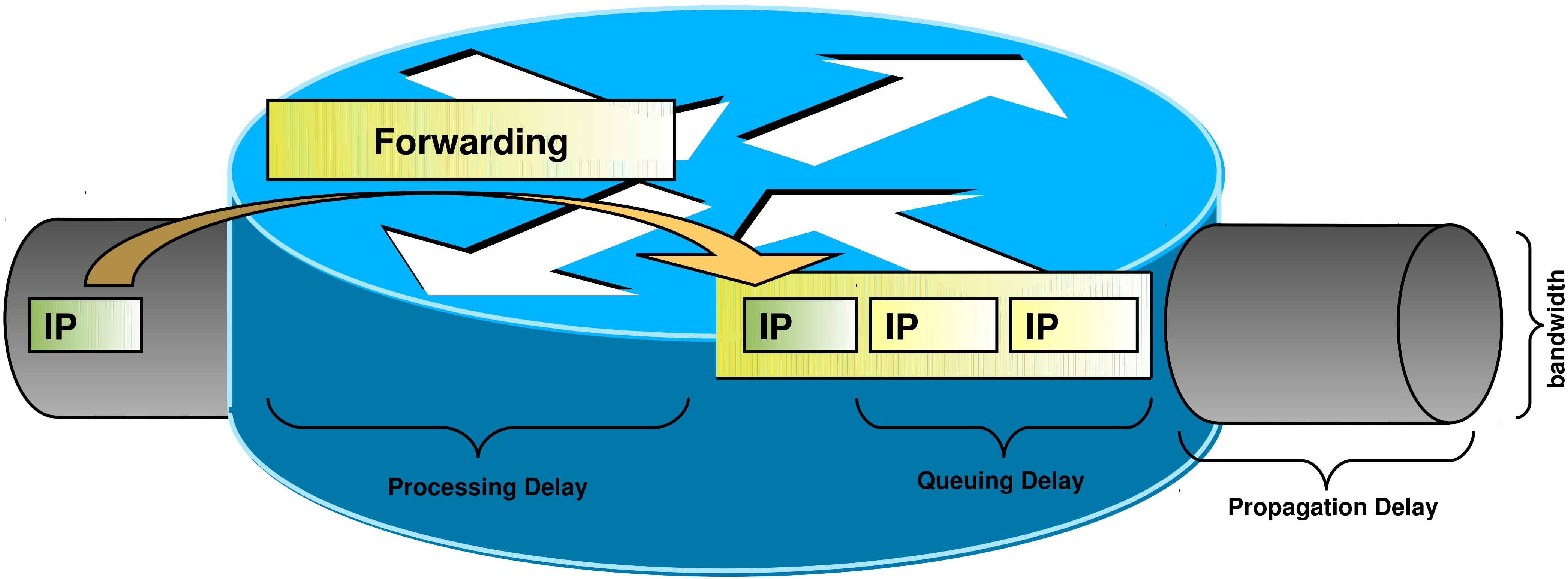
End-to-end Delay



Delay = P1 + Q1 + P2 + Q2 + P3 + Q3 + P4 = X ms

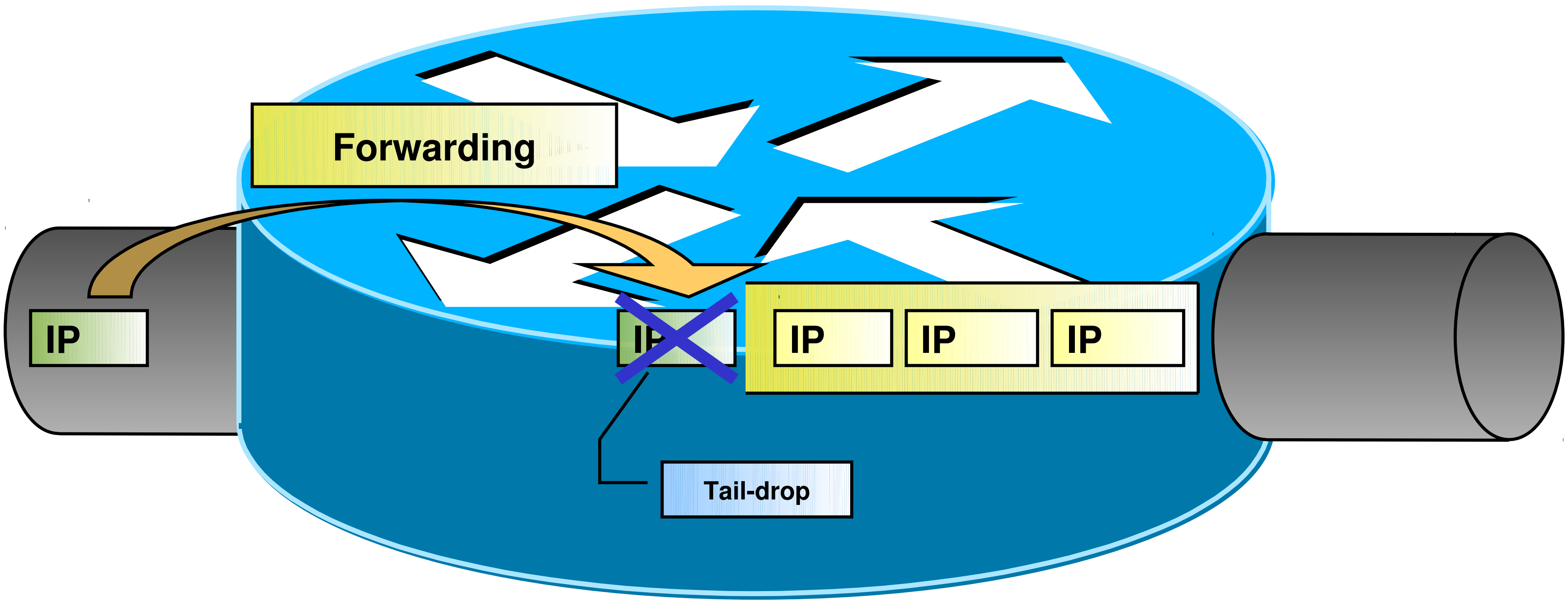
- End-to-end delay equals a sum of all propagation, processing and queuing delays in the path
- Propagation delay is fixed, processing and queuing delays are unpredictable in best-effort networks

Processing and Queuing Delay



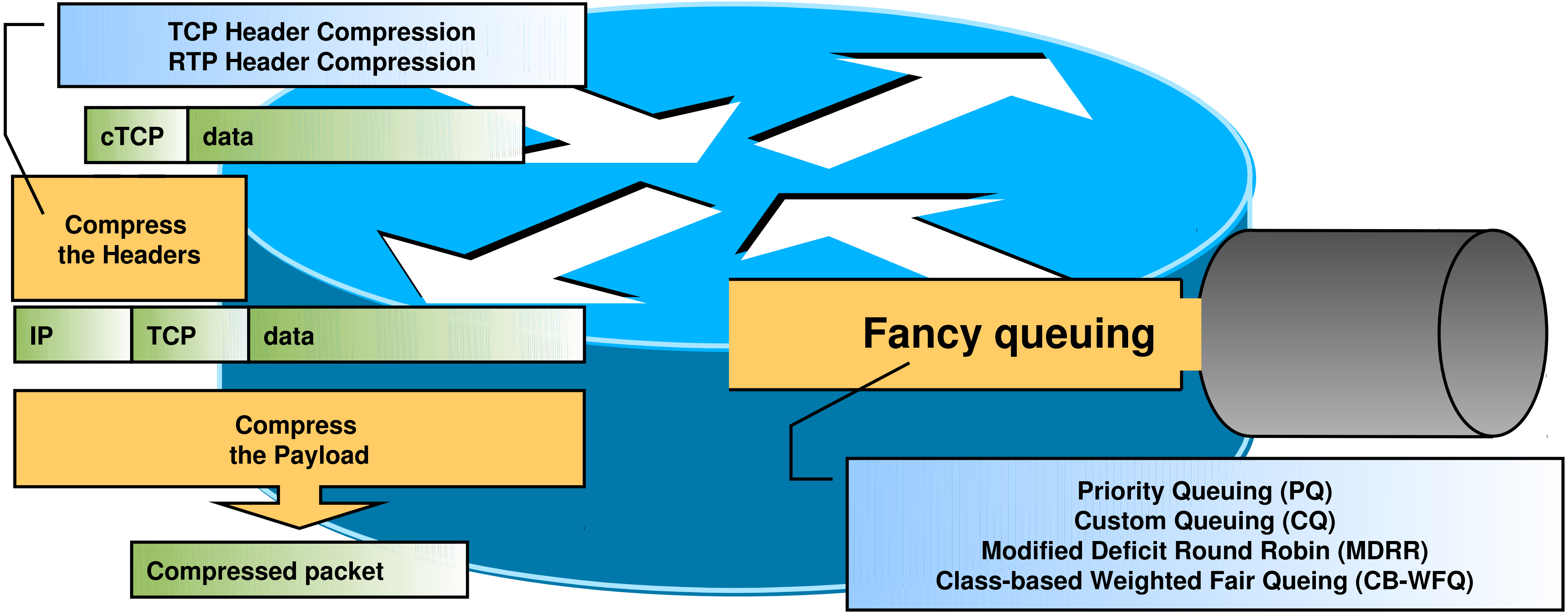
- Processing Delay is the time it takes for a router to take the packet from an input interface and put it into the output queue of the output interface.
- Queuing Delay is the time a packets resides in the output queue of a router.
- Propagation or Serialization Delay is the time it takes to transmit a packet.

Packet Loss



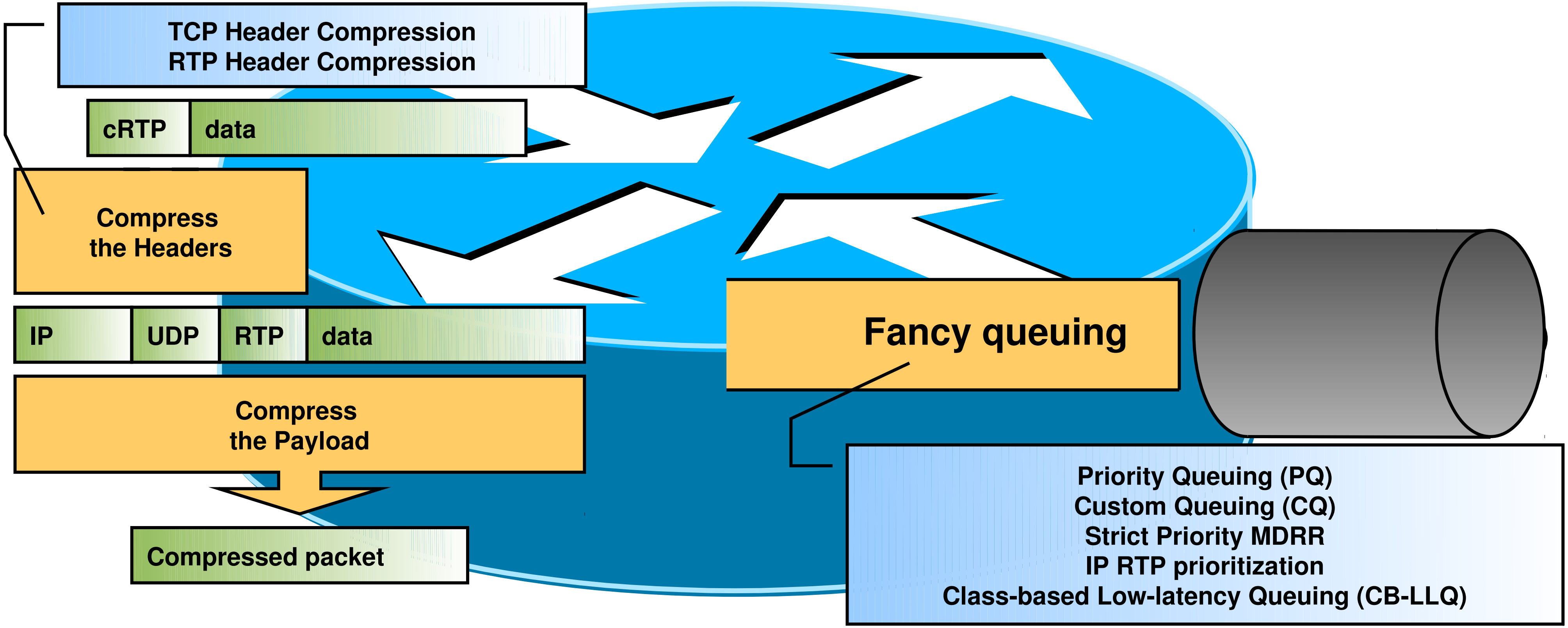
- Tail-drops occur when the output queue is full. These are the most common drops which happen when a link is congested.
- There are also many other types of drops that are not as common and may require a hardware upgrade (input drop, ignore, overrun, no buffer, ...). These drops are usually a result of router congestion.

How to Increase Available Bandwidth?



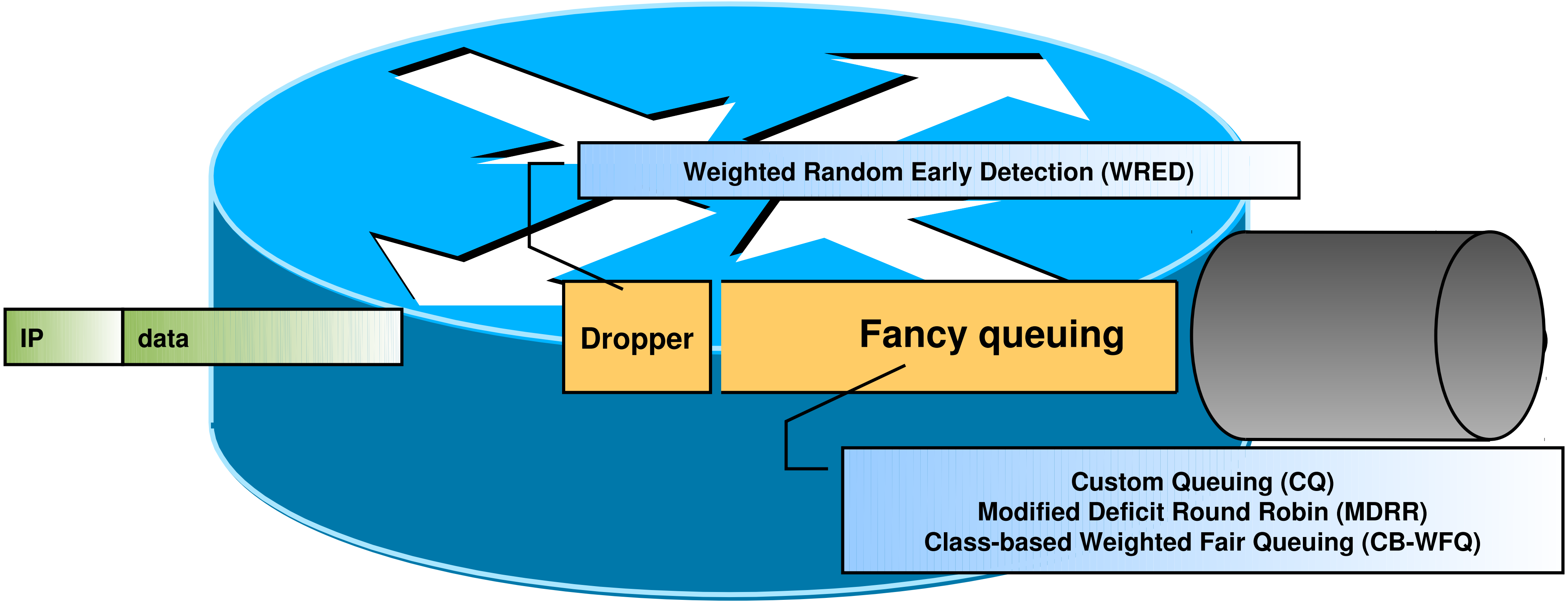
- Upgrade the link. The best solution but also the most expensive.
- Take some bandwidth from less important applications.
- Compress the payload of layer-2 frames.
- Compress the header of IP packets.

How to Reduce Delay?



- Upgrade the link. The best solution but also the most expensive.
- Forward the important packets first.
- Compress the payload of layer-2 frames (it takes time).
- Compress the header of IP packets.

How to Prevent Packet Loss?



- Upgrade the link. The best solution but also the most expensive.
- Guarantee enough bandwidth to sensitive packets.
- Prevent congestion by randomly dropping less important packets before congestion occurs

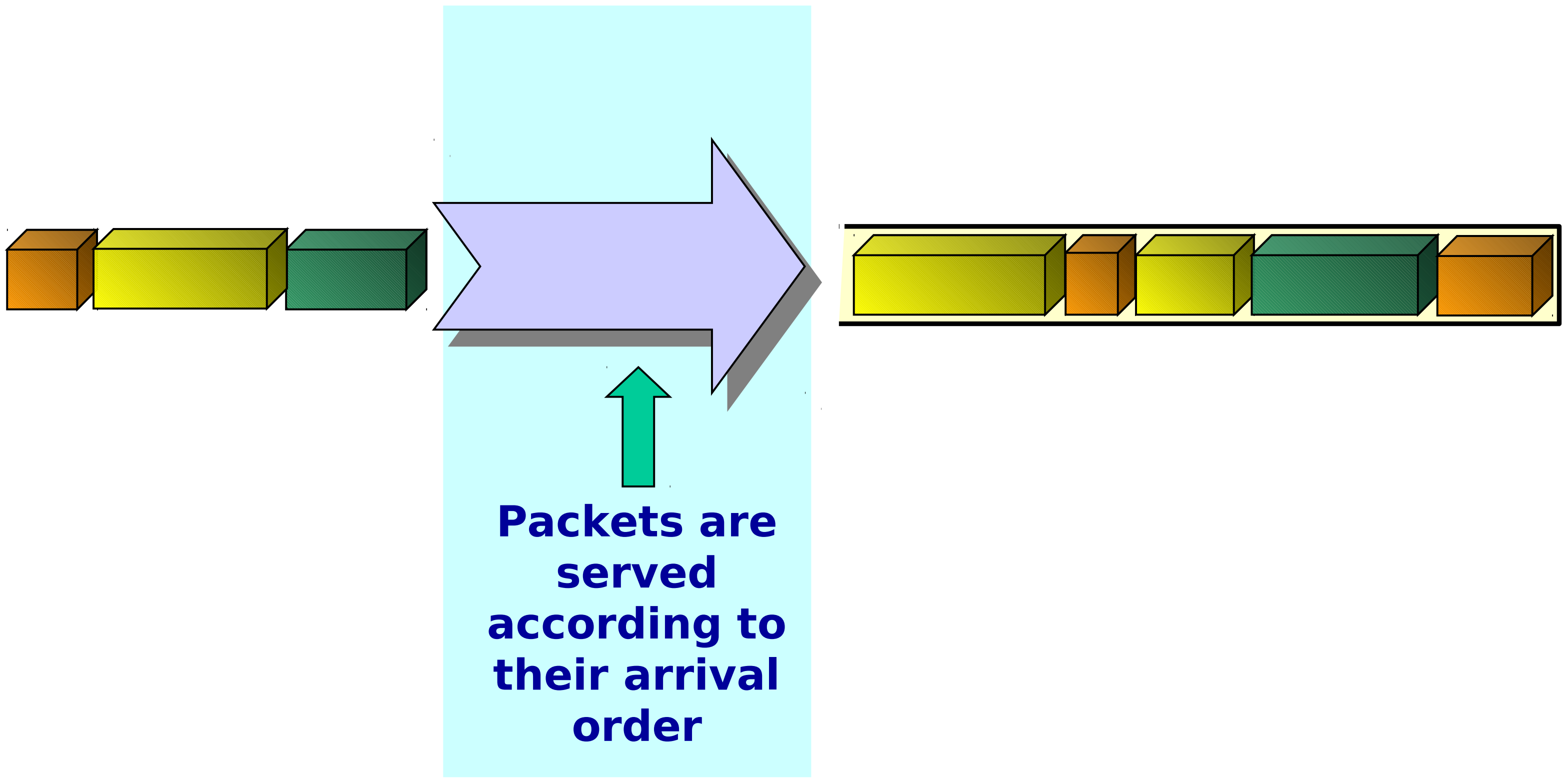
Traffic Terminology

- Flow: a single instance of an application-to-application flow of packets which is identified by source address, source port, destination address, destination port and protocol ID.
- Traffic stream: an administratively significant set of one or more flows which traverse a path segment. A traffic stream may consist of a set of active flows which are selected by a particular classifier.
- Traffic profile: a description of the temporal properties of a traffic stream such as average and peak rate and burst size.

Scheduling/Queuing algorithms

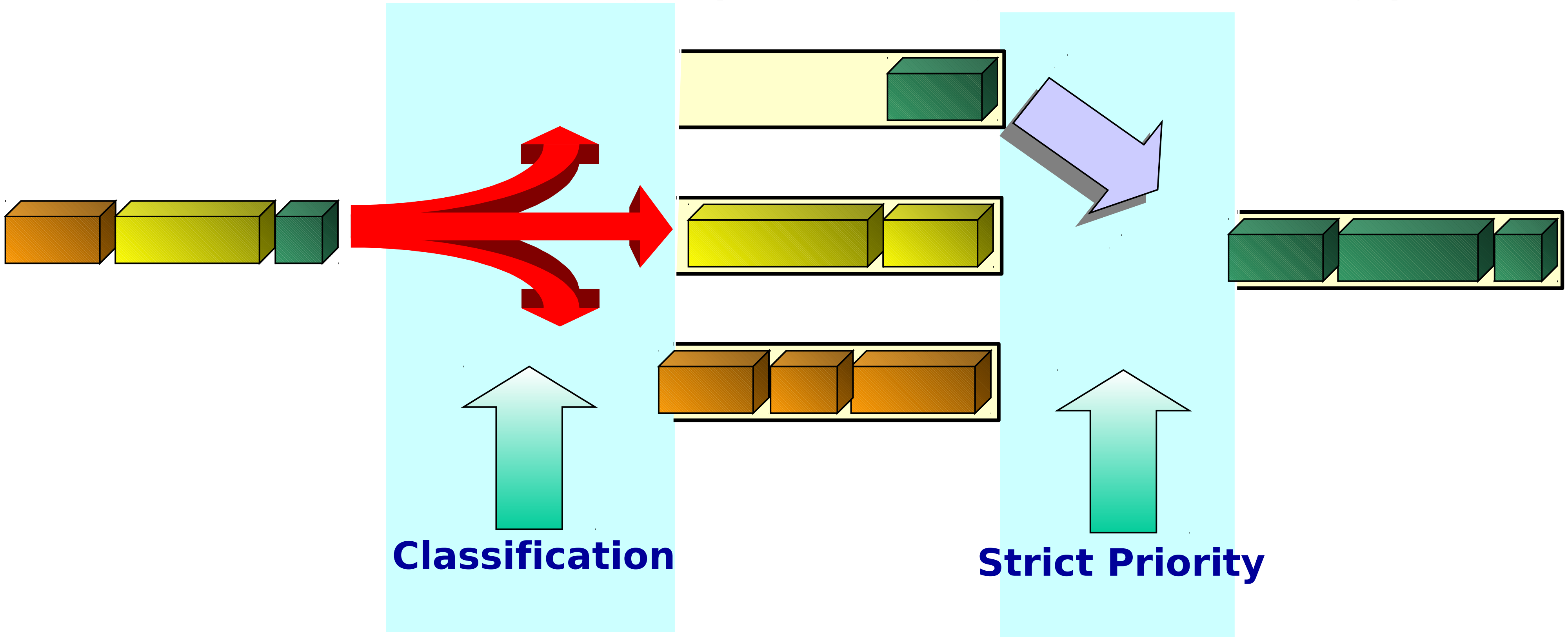
- Scheduling algorithms: decide the order packets from different flows are served in a queue
- Work conserving scheduling algorithms guarantee that the server is not occupied if and only if there is no packets waiting to be served
- Examples of work conserving scheduling algorithms:
  - FIFO
  - Strict priority (priority queuing)
  - Fair Queuing
  - Weighted Fair Queuing

First In First Out (FIFO)



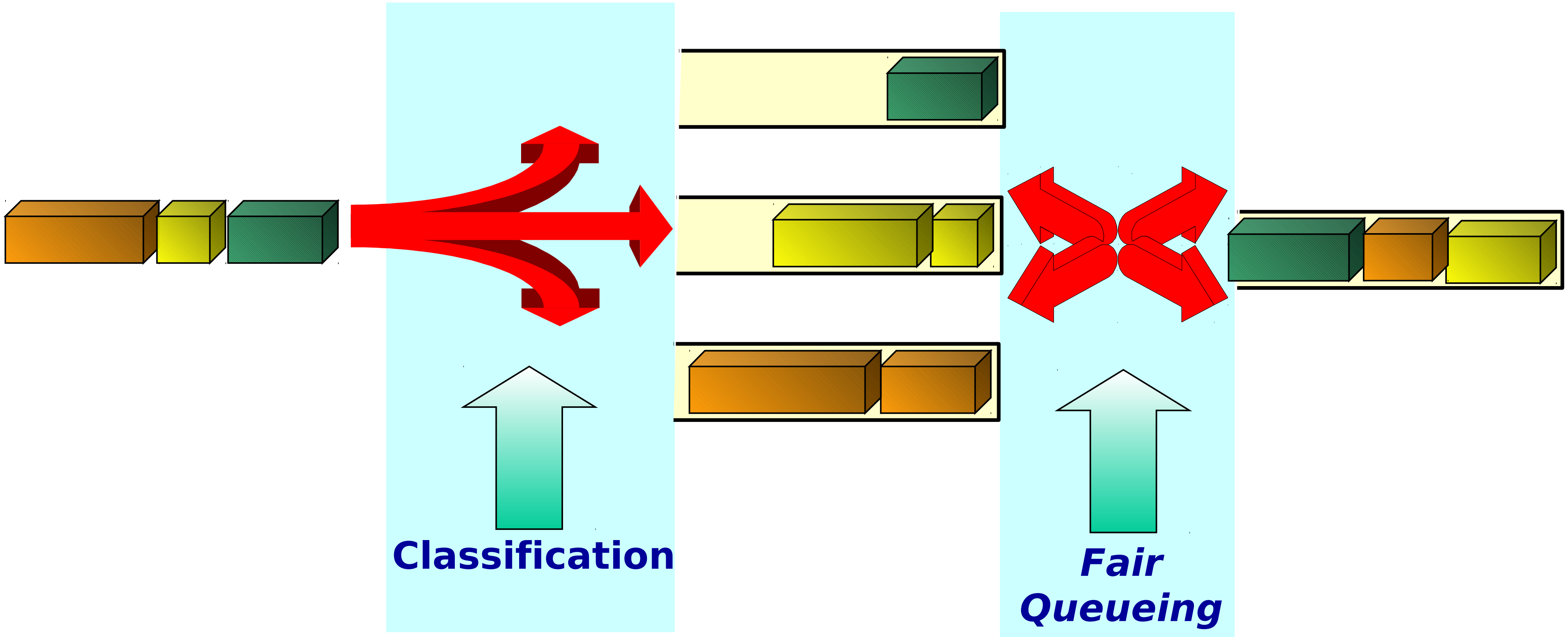
- Does not involve any ordering processing.
- Does not enable QoS differentiation.
- Flows having n times more traffic receive n times more service.
- On finite length queues, flows having smaller size packets receive more service.

Strict priority (Priority Queuing)



- Involves traffic classification according to priority.
- Higher priority traffic is always served before lower priority traffic.
- Enables QoS differentiation.
- Higher priority flows can prevent lower priority flows from receiving any service.

Fair Queueing (FQ)



- Involves traffic classification on different queues.
- Transmission bandwidth is equally distributed over non-empty queues.
- Enables QoS assignment.

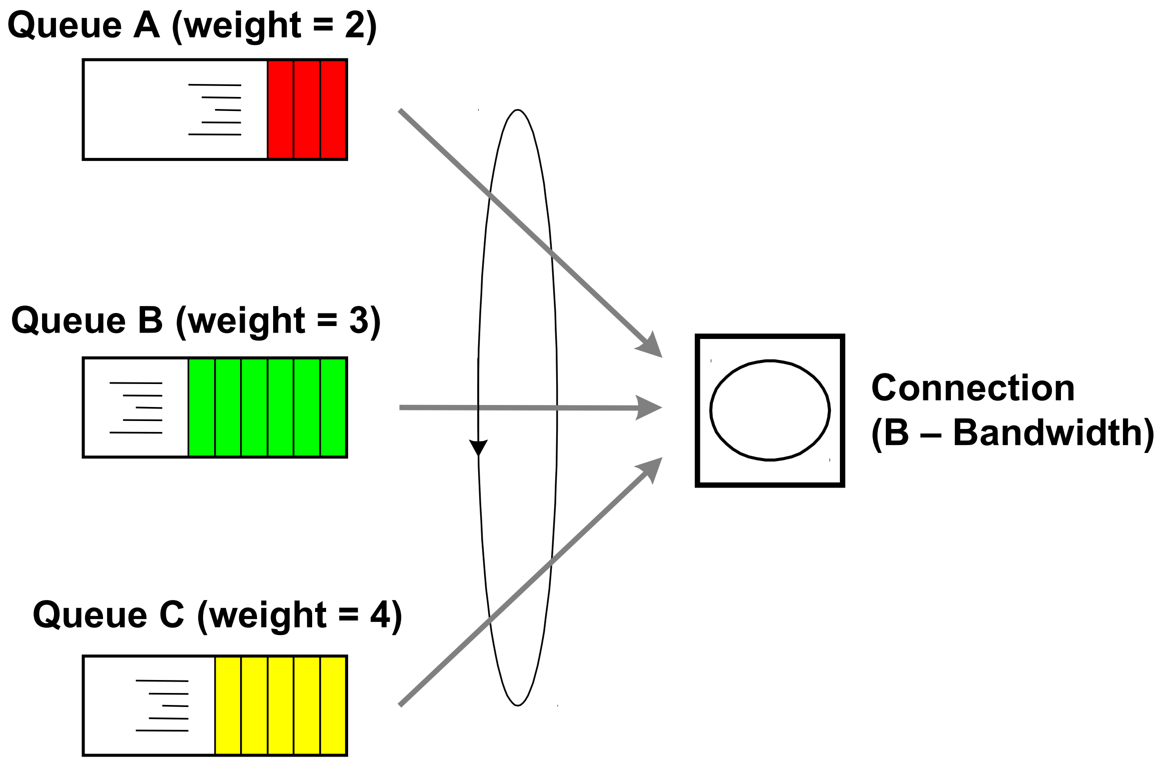
Weighted Fair Queuing (WFQ)

- This algorithm guarantees that each queue gets a percentage of the connection bandwith that is, al least, equal to its weight divided by the sum of all queues' weights

$$R_A = \frac{2}{2+3+4} B$$

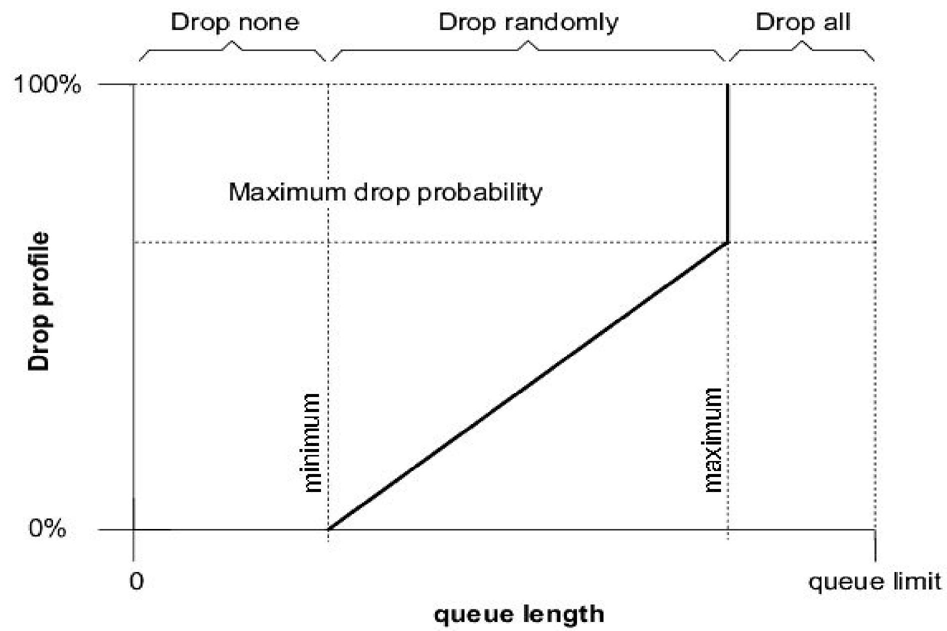
$$R_B = \frac{3}{2+3+4} B$$

$$R_C = \frac{4}{2+3+4} B$$



Dropper Mechanisms

- Random Early Detection (RED)
  - Congestion avoidance mechanism that takes advantage of TCP's congestion control
  - Randomly drops packets prior to periods of high congestion
  - Indirectly (by TCP) tells the packet source to decrease its transmission rate.
- Weighted RED (WRED)
  - Drops packets selectively based on priority classes
  - Higher priority traffic is delivered with a higher probability than lower priority traffic



How can QoS be Applied?

- Best effort – no QoS is applied to packets (default behavior)
- Integrated Services model – applications signal to the network that they require special QoS
- Differentiated Services model – the network recognizes classes that require special QoS

“Integrated *Services*” Architecture

Integrated Services (IntServ) Architecture

- The Integrated Services model (RFC1633) was introduced to guarantee a predictable behavior of the network for these applications
- For flows requiring Quality of Service, it is necessary to perform resource reservation on the flow paths between sources and destinations
  - Reservations are made flow-by-flow
- As opposed to the “best effort” service, the network implements a mechanism to control the admission of reservations (“call admission control”)
  - Flows that were not given any reservation are treated as “best effort” traffic

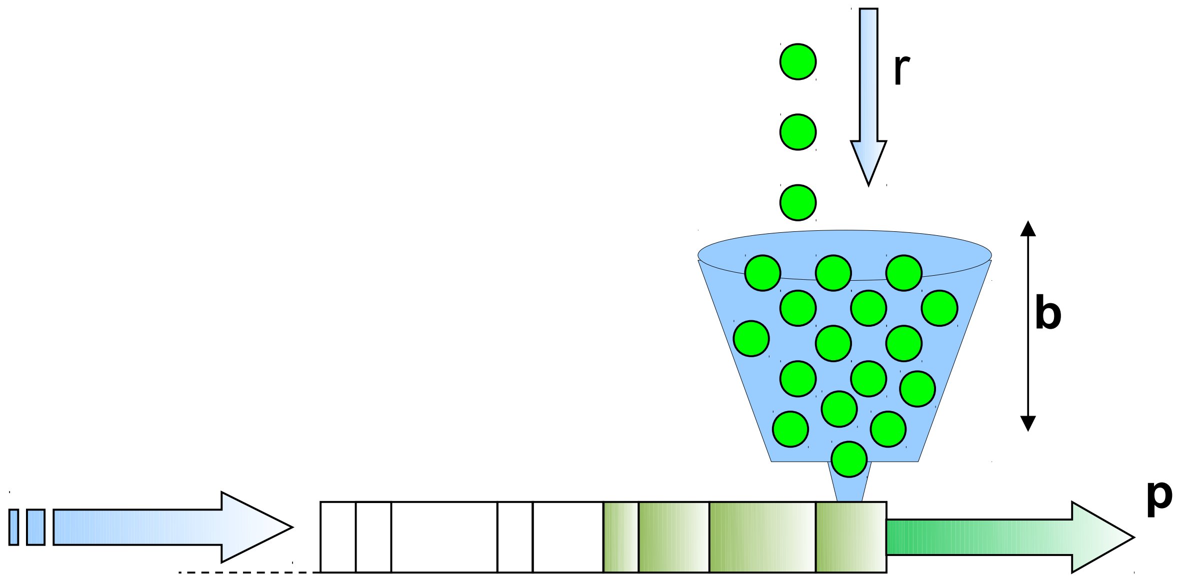
IntServ Classes of Service

- *Controlled Load* (RFC 2211)
  - Provides a service very similar to the "*best effort*" service in a non-congestioned network
  - Terminal stations should feel that a high percentage of their packets is delivered with routers' *queuing delays* very close to zero
- *Guaranteed Service* (RFC 2212)
  - Provides a maximum delay for all IP packets

- Both services demand that the sender condition the packet sending process according to a "*token bucket*" model

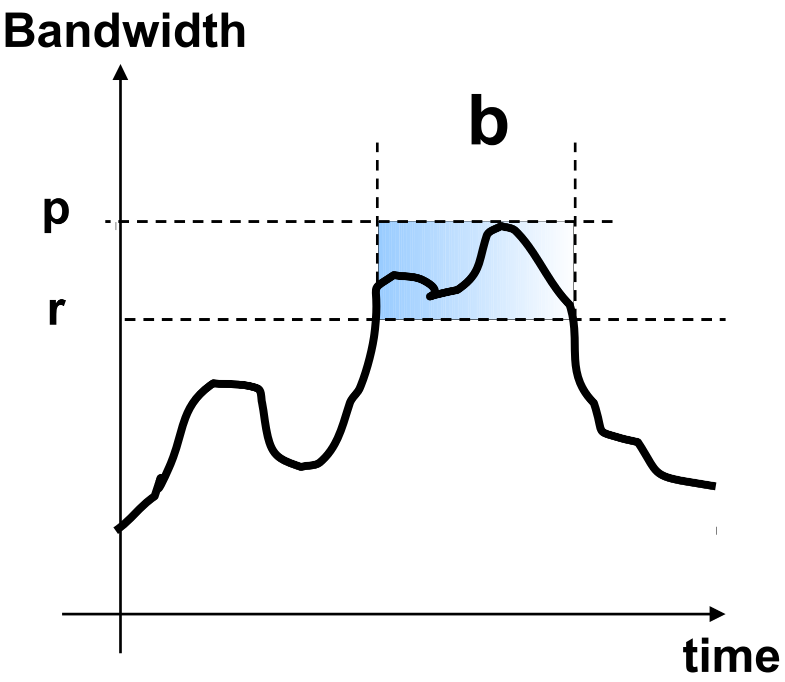
Traffic Characterization at the Sender

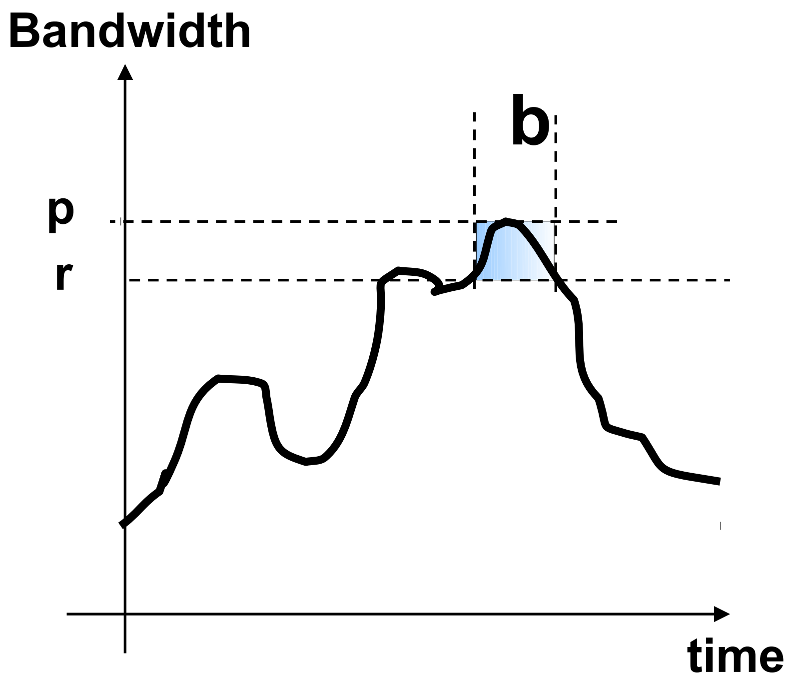
Token Bucket model



- **r = token filling rate (bytes/s)**
- **b = bucket size (bytes)**
- **p = maximum transmission rate (bytes/s)**
- **M = maximum packet size (bytes)**
- **m = minimum packet size (bytes) – each packet having a lower size will be considered as a size *m* packet**

Traffic Characterization at the Sender

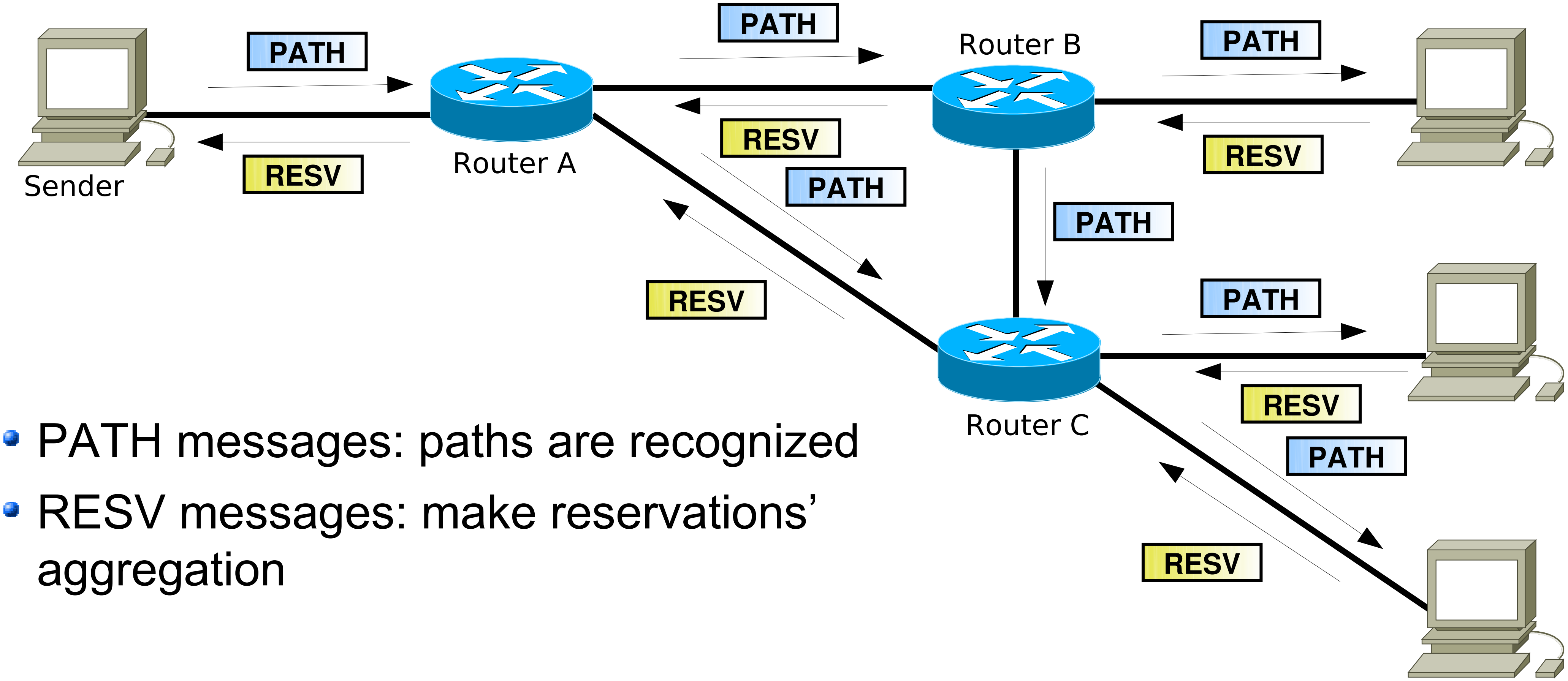




RSVP

- The resource ReSerVation Protocol (RSVP) was developed to communicate resource needs between hosts and network devices (RFC 2205-2215)
- RSVP allows:
  - The source do describe the characteristics of the IP packets flow
  - Destinations to describe the reservation they want
  - Routers to know how to process the packets flow in order to fulfill the requested reservation
- Encapsulated on IP; protocol type = 46 (0x2E)
- Signaling is based on the exchange of PATH and RESV messages
  - PATH announces the traffic characteristics at the sender
  - RESV achieves reservations that were initiated by the receivers
  - If the reservation is not possible, a RESV ERR message is sent
- The routers reservation states have to be periodically refreshed (soft states)
- RSVP is typically used by applications carrying voice or video over IP networks (initiated by a host)
- RSVP with extensions is also used by MPLS Traffic Engineering to establish MPLS/TE tunnels (initiated by a router)

RSVP Signaling



- PATH messages: paths are recognized
- RESV messages: make reservations' aggregation

Format of the RSVP messages

<table>
<tr><td>Version</td><td>iHL</td><td>Service Type</td><td colspan="3">Total Length</td><td rowspan="6">IPv4 Header</td></tr>
<tr><td colspan="3">Identification</td><td>Flag</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol: 46</td><td colspan="3">Header Checksum</td></tr>
<tr><td colspan="6">Source Address</td></tr>
<tr><td colspan="6">Destination Address</td></tr>
<tr><td colspan="5">Options</td><td>Padding</td></tr>
<tr><td>Version</td><td>Flags</td><td>RSVP Type</td><td colspan="3">Checksum</td><td rowspan="2">RSVP Header</td></tr>
<tr><td colspan="2">Send_TTL</td><td>Reserved</td><td colspan="3">Message length</td></tr>
<tr><td colspan="6">Body of the RSVP message</td><td>RSVP objects</td></tr>
</table>

Format of each RSVP object:

<table>
<tr><td>Object Length</td><td>Class Nº</td><td>Class Type</td></tr>
<tr><td colspan="3">Content</td></tr>
</table>

RSVP messages

- PATH (*Type* = 0x01)
  - Tspec (“flow traffic specification”): contains the parameters that describe the traffic source based on the “Token Bucket” model
- RESV (*Type* = 0x02)
  - Tspec: the same that was received on the PATH message
  - FilterSpec (“*filter specification*”): contains the flow descriptor that enables routers to identify packets belonging to this reservation (source address, destination address, protocol type, source port number, destination port number, any combination of these parameters)
  - Rspec (“*flow reservation specification*”): contains the parameters describing the reservation that the receiver wants to become supported
    - Rspec is specified if the receiver wants a service of the “*guaranteed service*” type; when it is not specified, it means that the receiver wants a service of the “*controlled load*” type

RSVP Parameters

RSVP Parameters	Description
TOKENBUCKETRATE (r)	TSpec: Rate of arriving tokens
TOKENBUCKETSIZE (b)	TSpec: Size of bucket
PEAKRATE (p)	TSpec: Maximum bit rate of the flow
MINIMUMPOLICEDUNIT (m)	TSpec: Minimum packet size considered
MAXIMUMPACKETSIZE (M)	TSpec: Maximum packet size
RATE (R)	RSpec*: Reservation rate
DELAYSLACKTERM	RSpec*: Tolerance of the requested delay

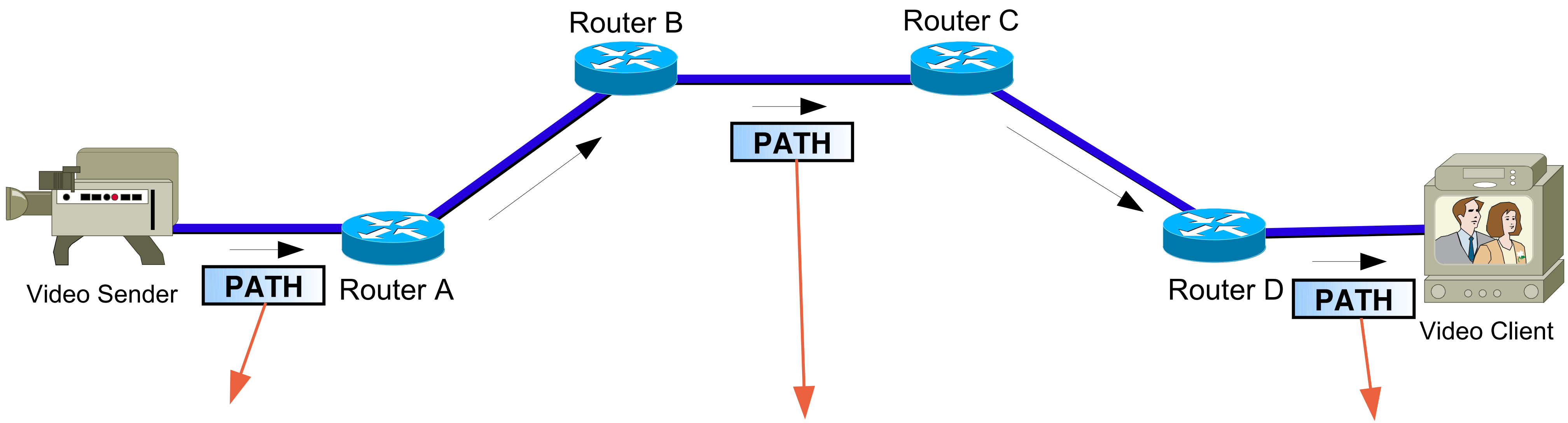
* RSpec is specified only for *Guaranteed Services*

RSVP PATH

- This message includes three mandatory RSVP objects (besides FLOWSPEC):
- SESSION – Identifies the session by the destination IP address, destination port and protocol ID
- RSVP_HOP – Indicates to the next router the sending IP address and port
- TIME_VALUES – Indicates the time period between successive sendings of PATH messages

1	0	RSVP Type: 1	Checksum		PATH header
Send_TTL		0	Message length: 40		
SESSION object length: 12			Class Nº : 1	Class Type: 1	SESSION
Destination Address					
Protocol ID		Flags	Destination port		
RSVP_HOP object length: 12			Class Nº : 3	Class Type: 1	RSVP_HOP
Last Hop Address					
Logical Interface Handle of the last node (LIH)					
TIME_VALUES object length: 8			Class Nº : 5	Class Type: 1	TIME_VALUES
Update period (ms)					

RSVP PATH (Example)



Vs.: 4	iHL: 5	Service	Total Length: 60	
Identification			Flg	Fragment Offset
Time to Live		Protocol: 46	Header Checksum	
Source Address: **Video Server**				
Destination Address: **Video Client**				
1	0	Type: 1	Checksum	
Send_TTL		0	Message Length: 40	
SESSION Length.: 12			Class Nº: 1	Class Type: 1
Destination Address: **Video Client**				
Protocol ID		Flags	Destination port	
RSVP_HOP Length. : 12			Class Nº: 3	Class Type: 1
Last Hop Address: **Video Server**				
Logical Interface Handle of the last node (LIH)				
TIME_VALUES Length: 8			Class Nº: 5	Class Type: 1
Update Period (ms)				

Vs.: 4	iHL: 5	Service	Total Length: 60	
Identification			Flg	Fragment Offset
Time to Live		Protocol: 46	Header Checksum	
Source Address: **Video Server**				
Destination Address: **Video Client**				
1	0	Type: 1	Checksum	
Send_TTL		0	Message Length: 40	
SESSION Length: 12			Class Nº: 1	Class Type: 1
Destination Address: **Video Client**				
Protocol ID		Flags	Destination Port	
RSVP_HOP Length: 12			Class Nº: 3	Class Type: 1
Last Hop Address: **Router B**				
Logical Interface Handle of the last node (LIH)				
TIME_VALUES Length: 8			Class Nº: 5	Class Type: 1
Update Period (ms)				

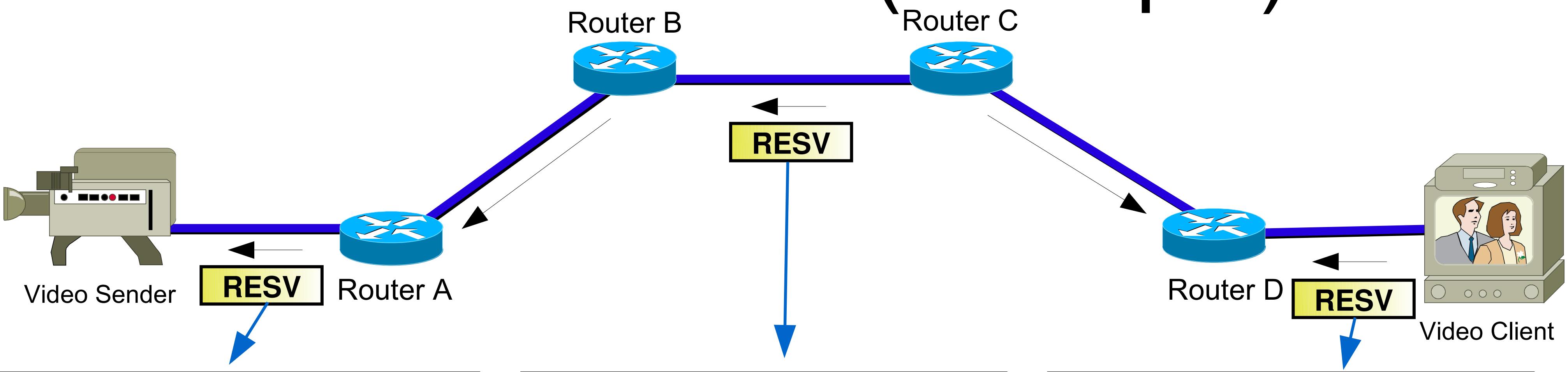
Vs.: 4	iHL: 5	Service	Total Length: 60	
Identification			Flg	Fragment Offset
Time to Live		Protocol: 46	Header Checksum	
Source Address: **Video Server**				
Destination Address: **Video Client**				
1	0	Type: 1	Checksum	
Send_TTL		0	Message Length: 40	
SESSION Length: 12			Class Nº: 1	Class Type: 1
Destination Address: **Video Client**				
Protocol ID		Flags	Destination Port	
RSVP_HOP Length: 12			Class Nº: 3	Class Type: 1
Last Hop Address: **Router D**				
Logical Interface Handle of the last node (LIH)				
TIME_VALUES Length: 8			Class Nº: 5	Class Type: 1
Update Period (ms)				

RSVP RESV

- STYLE – Identifies the style of the reservation
- FLOWSPEC – Includes TSpec and RSpec
- FILTER_SPEC – Indicates the necessary information for the packet classifier

1	0	RSVP Type: 2	Checksum	RESV Header
Send_TTL	0	Message length		
SESSION object length: 12		Class Nº: 1	Class Type: 1	SESSION
Destination Address				
Protocol ID	Flags	Destination Port		
RSVP_HOP object length: 12		Class Nº: 3	Class Type: 1	RSVP_HOP
Address of the last node				
Logical Interface Handle of the last node (LIH)				
TIME_VALUES object length: 8		Class Nº: 5	Class Type: 1	TIME_VALUES
Update period (ms)				
STYLE object length: 8		Class Nº: 8	Class Type: 1	STYLE
Flags	Style Option Vector: 0x00000A (FF)			
FLOWSPEC object length		Class Nº: 9	Class Type	FLOWSPEC
FLOWSPEC object contents				
FILTER_SPEC object length : 24		Class Nº: 10	Class Type: 1	FILTER_SPEC
Source Address				
Reserved	Reserved	Source protocol port		

RSVP RESV (Example)



Vs.: 4	iHL: 5	Service	Total Length	
Identification			Flg	Fragment Offset
Time to Live		Protocol: 46	Header Checksum	
Source Address: **Router A**				
Destination Address: **Video Server**				
1	0	Type: 2	Checksum	
Send_TTL		0	Message Length	
SESSION Length: 12			Class Nº: 1	Class Type: 1
Destination Address: **Video Client**				
Protocol Id		Flags	Destination protocol port	
RSVP_HOP Length: 12			Class Nº: 3	Class Type: 1
Address of the last node: **Router A**				
Logical Interface Handle of the last node (LIH)				
TIME_VALUES Length: 8			Class Nº: 5	Class Type: 1
Update period (ms)				
STYLE Object Length : 8			Class Nº: 8	Class Type: 1
Flags		Style Option Vector: 0x00000A (FF)		
FLOWSPEC Length			Class Nº: 9	Class Type
FLOWSPEC object contents				
FILTER_SPEC Length: 12			Class Nº: 10	Class Type: 1
Source Address: Video Server				
Reserved		Reserved	Source protocol port	

Vs.: 4	iHL: 5	Service	Total Length	
Identification			Flg	Fragment Offset
Time to Live		Protocol: 46	Header Checksum	
Source Address: **Router C**				
Destination Address: **Router B**				
1	0	Type: 2	Checksum	
Send_TTL		0	Message Length	
SESSION Length: 12			Class Nº: 1	Class Type: 1
Destination Address: **Video Client**				
Protocol Id		Flags	Destination protocol port	
RSVP_HOP Length: 12			Class Nº: 3	Class Type: 1
Address of the last node: **Router C**				
Logical Interface Handle of the last node (LIH)				
TIME_VALUES Length: 8			Class Nº: 5	Class Type: 1
Update period (ms)				
STYLE Object Length : 8			Class Nº: 8	Class Type: 1
Flags		Style Option Vector: 0x00000A (FF)		
FLOWSPEC Length			Class Nº: 9	Class Type
FLOWSPEC object contents				
FILTER_SPEC Length: 12			Class Nº: 10	Class Type: 1
Source Address: Video Server				
Reserved		Reserved	Source protocol port	

Vs.: 4	iHL: 5	Service	Total Length	
Identification			Flg	Fragment Offset
Time to Live		Protocol: 46	Header Checksum	
Source Address: **Video Client**				
Destination Address: **R outer D**				
1	0	Type: 2	Checksum	
Send_TTL		0	Message Length	
SESSION Length: 12			Class Nº: 1	Class Type: 1
Destination Address: **Video Client**				
Protocol Id		Flags	Destination protocol port	
RSVP_HOP Length: 12			Class Nº: 3	Class Type: 1
Address of the last node: **Video Client**				
Logical Interface Handle of the last node (LIH)				
TIME_VALUES Length: 8			Class Nº: 5	Class Type: 1
Update period (ms)				
STYLE Object Length : 8			Class Nº: 8	Class Type: 1
Flags		Style Option Vector: 0x00000A (FF)		
FLOWSPEC Length			Class Nº: 9	Class Type
FLOWSPEC object contents				
FILTER_SPEC Length: 12			Class Nº: 10	Class Type: 1
Source Address: Video Server				
Reserved		Reserved	Source protocol port	

RSVP Reservation Styles

- “Fixed Filter” (Style Option Vector = 0x00000A)
  - The receiver specifies a reservation value for each sender
- “Wildcard Filter” (Style Option Vector = 0x000011)
  - The receiver specifies a unique reservation value to receive traffic from any sender
- “Explicit Filter” (Style Option Vector = 0x000012)
  - The receiver specifies a list of senders from which it wants to receive information and a unique reservation value to receive traffic from the specified senders

- On RSVP RESV messages:
  - The reservation style is declared by the STYLE object
  - Senders are declared on the FILTER_SPEC object

Other RSVP messages

- PATH ERR (*Type* = 0x03):
  - Sent by routers in error situations
- RESV ERR (*Type* = 0x04):
  - Sent by routers when a reservation cannot be supported
- PATH TEAR (*Type* = 0x05):
  - Sent by senders when information they finish transmitting information
- RESV TEAR (*Type* = 0x06):
  - Sent by receivers when they do not want a reservation anymore
- RESV CONFIRMATION (*Type* = 0x07):
  - Sent by routers to confirm the establishment of a reservation

RSVP characteristics

- Multipoint-multipoint model (simplex)
- Reservations initiated by the receivers
- Temporized reservations (soft state)
- Separation between reservation and routing
- Separation between reservation and packet filtering
- Different reservation styles
- Aggregation of reservations

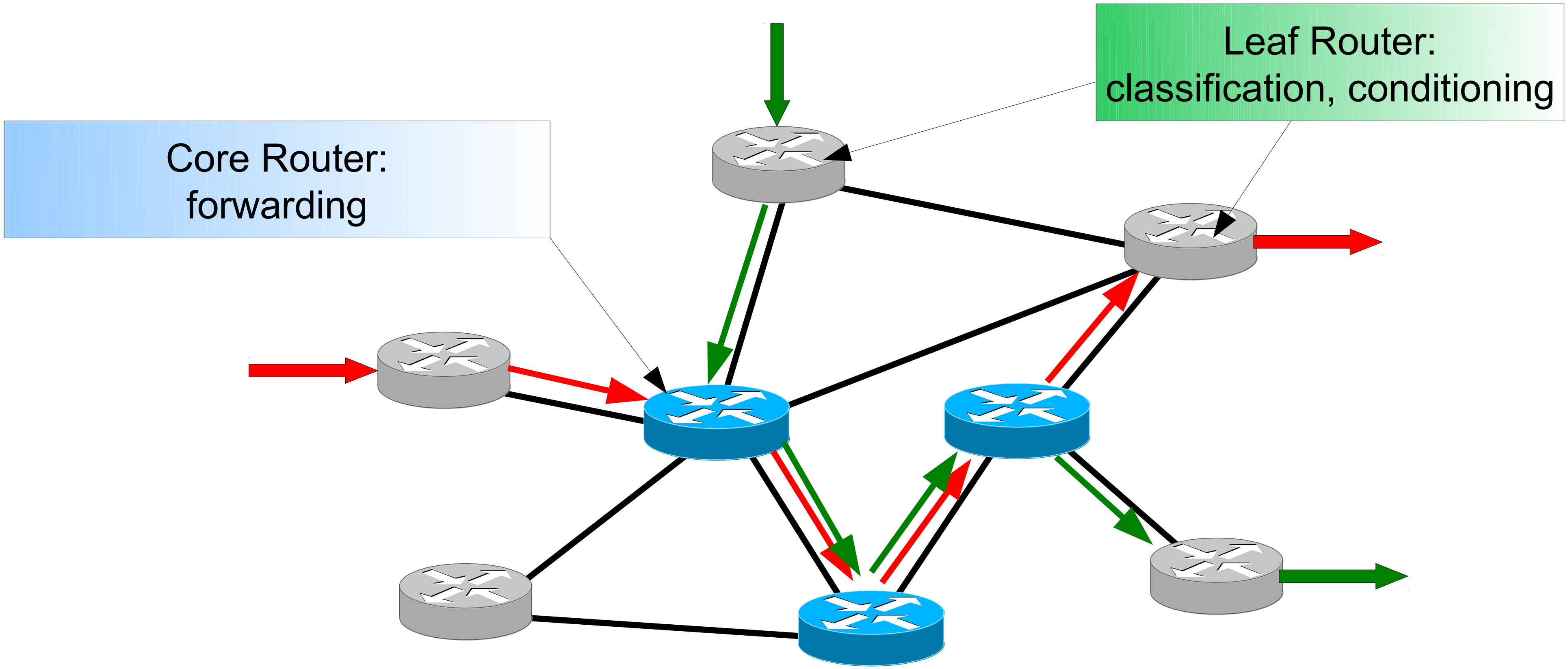
“*Differentiated Services*” Architecture

Differentiated Services (*DiffServ*) Architecture

- Problems of the *Integrated Services* architecture:
  - Routers maintain information about the state of the end-to-end reservations
    - Poor scalability
  - Routers determine attendance order based on multiple fields (source and destination addresses, protocol, source and destination port)
    - Penalizes performance
  - Supports only two service classes: “*controlled load*” and “*guaranteed service*”
    - Low flexibility
  - Demands end-to-end RSVP signaling
    - High reservation establishment times
- Thus, DiffServ architecture:
  - By contract, the traffic flow from each client is classified as belonging to a particular class
  - Treats flow classes that demand the same Quality of Service
  - At the network entrance, packets are marked as belonging to the contracted class and packet scheduling is based on the packet mark

Basic ideas

- Implement simple routing operations on the network *core routers* and leave complex operations to the network *edge routers*.
- It only defines functional elements that enable the support of any service class.



Differentiated Services Model

- Differentiated Services model describes services associated with traffic classes
- Complex traffic classification and conditioning is performed at network edge resulting in a per-packet Differentiated Services Code Point (DSCP).
- No per-flow/per-application state in the core
- Core only performs simple ‘per-hop behavior's’ on traffic aggregates
- Goal is scalability

DiffServ Elements

- The service defines QoS requirements and guarantees provided to a traffic aggregate;
- The conditioning functions and per-hop behaviors are used to realize services;
- The DS field value (DS Code Point) is used to mark packets to select a per-hop behavior
- Per-hop Behavior (PHB) is realized using a particular QoS mechanism
- Provisioning is used to allocate resources to traffic classes

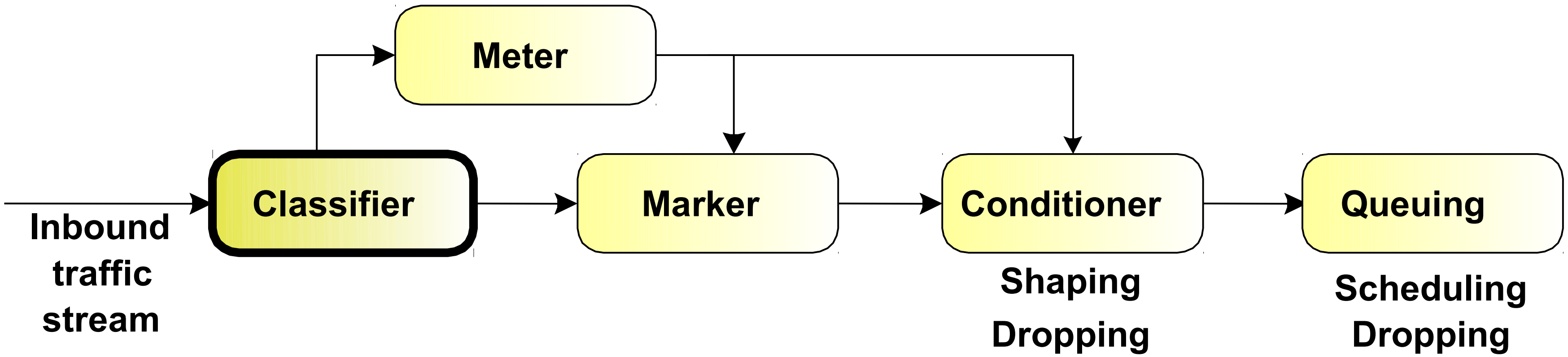
Traffic Terminology

- Behavior Aggregate (BA) is a collection of packets with the same DS code point crossing a link in a particular direction.
- Per-Hop Behavior (queuing in a node) externally observable forwarding behavior applied at a DS-compliant node to a DS behavior aggregate.
- PHB Mechanism: a specific algorithm or operation (e.g., queuing discipline) that is implemented in a node to realize a set of one or more per-hop behaviors.

DiffServ QoS Actions

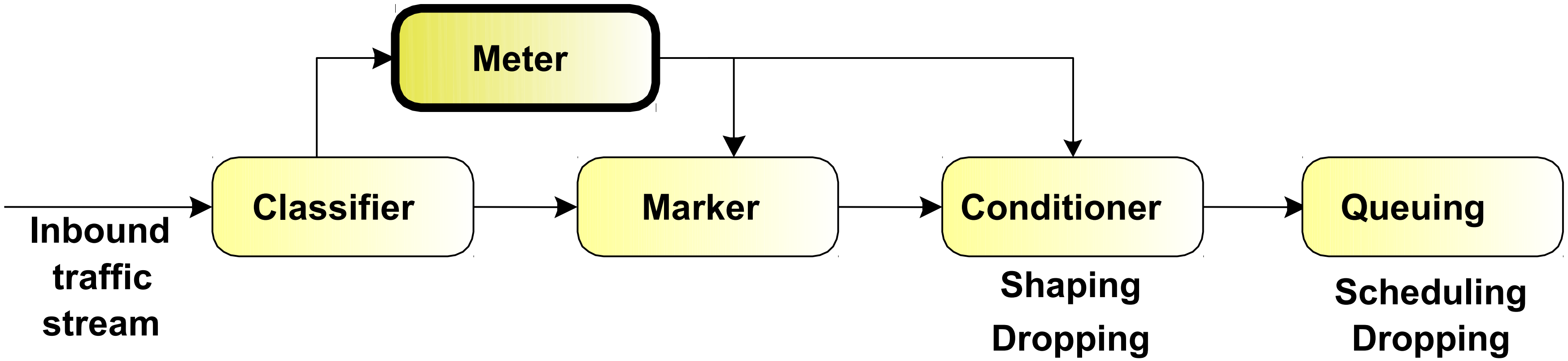
- Classification – Each class-oriented QoS mechanism has to support some type of classification (access lists, route maps, class maps, etc.)
- Metering – Some mechanisms measure the rate of traffic to enforce a certain policy (e.g. rate limiting, shaping, scheduling, etc.)
- Dropping – Some mechanisms are used to drop packets (e.g. random early detection)
- Policing – Some mechanisms are used to enforce a rate limit based on the metering (excess traffic is dropped)
- Shaping – Some mechanisms are used to enforce a rate limit based on the metering (excess traffic is delayed)
- Marking – Some mechanisms have the capability to mark packets based on classification and/or metering (e.g. CAR, class-based marking, etc.)
- Queuing – Each interface has to have a queuing mechanism
- Forwarding – There are several supported forwarding mechanisms (process switching, fast switching, CEF switching, etc.)

DiffServ Mechanisms



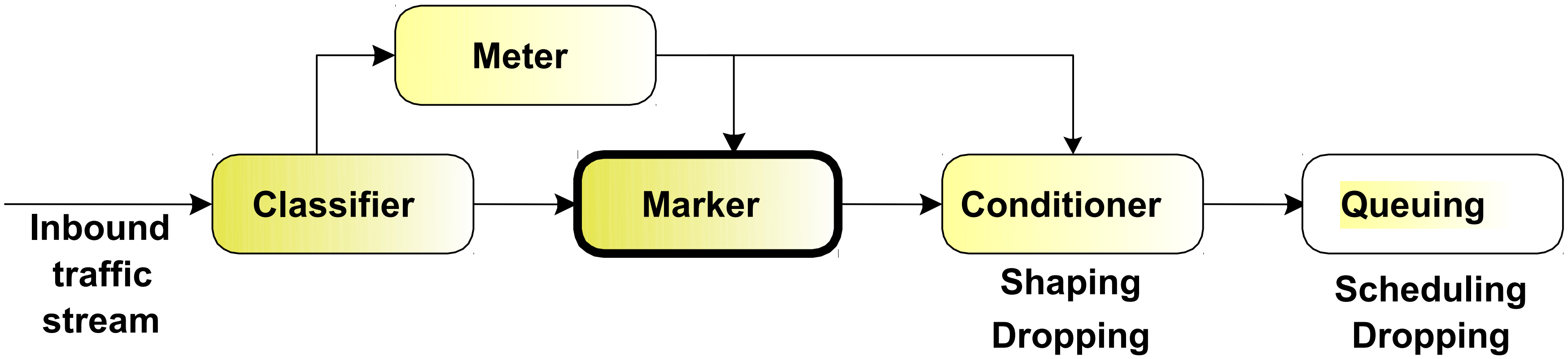
- Most traditional QoS mechanisms include extensive built-in classifiers
  - Committed Access Rate (CAR)
  - QoS Policy Propagation via BGP (QPPB)
  - Route-maps
  - Queuing mechanisms
  - ...

DiffServ Mechanisms



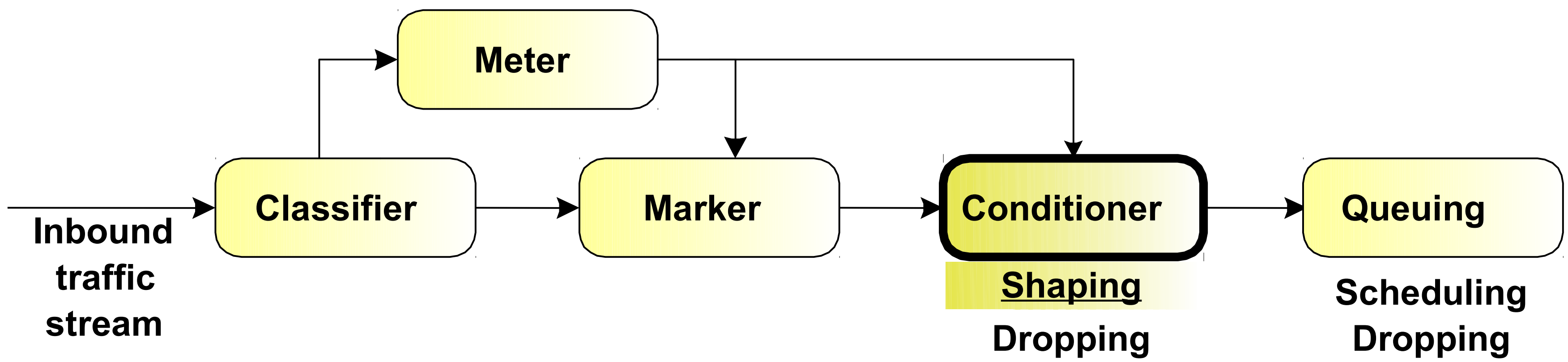
- Token Bucket model is used for metering
  - Committed Access Rate (CAR)
  - Generic Traffic Shaping (GTS)
  - Frame Relay Traffic Shaping (FRTS)
  - Class-based Weighted Fair Queuing (CB-WFQ)
  - Class-based Low Latency Queuing (CB-LLQ)
  - Class-based Policing
  - Class-based Shaping
  - IP RTP Prioritization

DiffServ Mechanisms



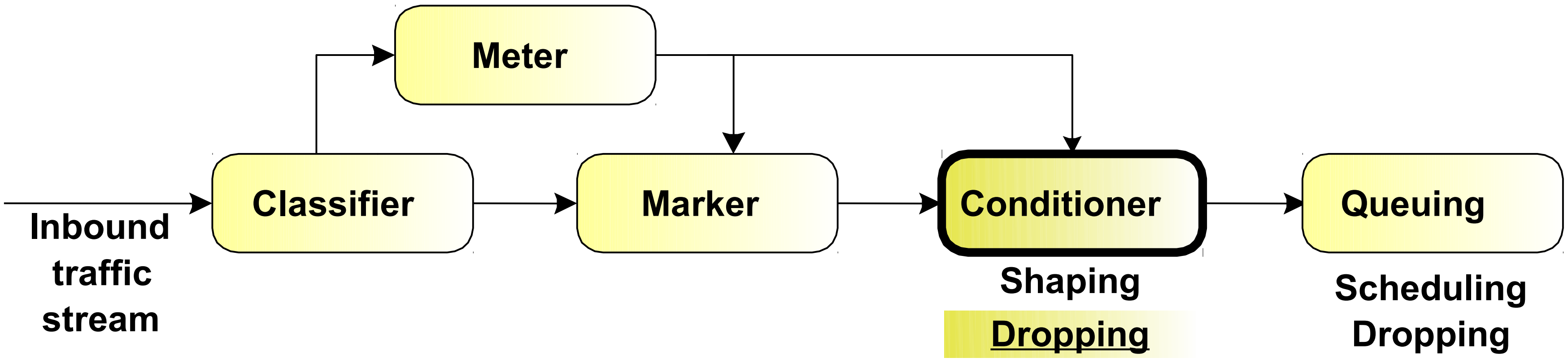
- Marker is used to set:
  - IP precedence
  - DSCP
  - QoS group
  - MPLS experimental bits
  - Frame Relay DE bit
  - ATM CLP bit
  - IEEE 802.1Q or ISL CoS
- Marking mechanisms:
  - Comitted Access Rate (CAR)
  - QoS Policy Propagation through BGP (QPPB)
  - Policy-based Routing (PBR)
  - Class-based Marking

DiffServ Mechanisms



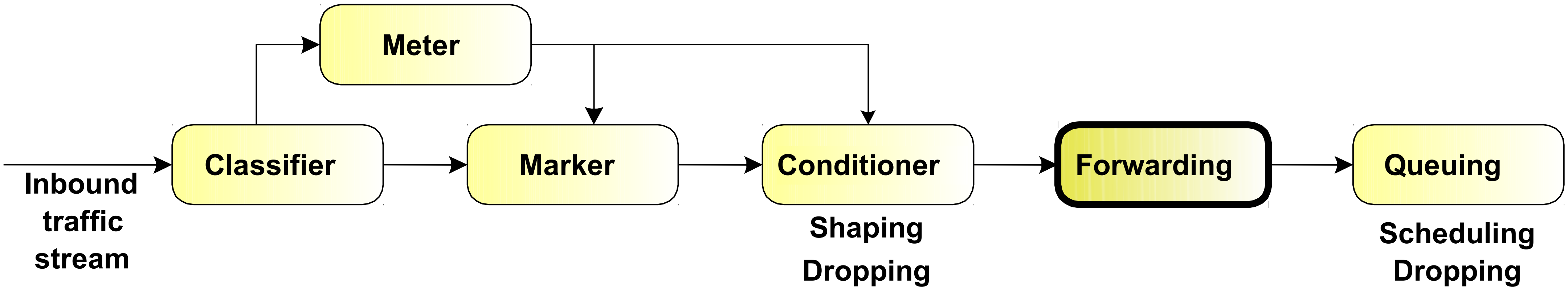
- Shaping mechanisms:
  - Generic Traffic Shaping (GTS)
  - Frame Relay Traffic Shaping (FRTS)
  - Class-based Shaping
  - Hardware shaping on ATM VC

DiffServ Mechanisms



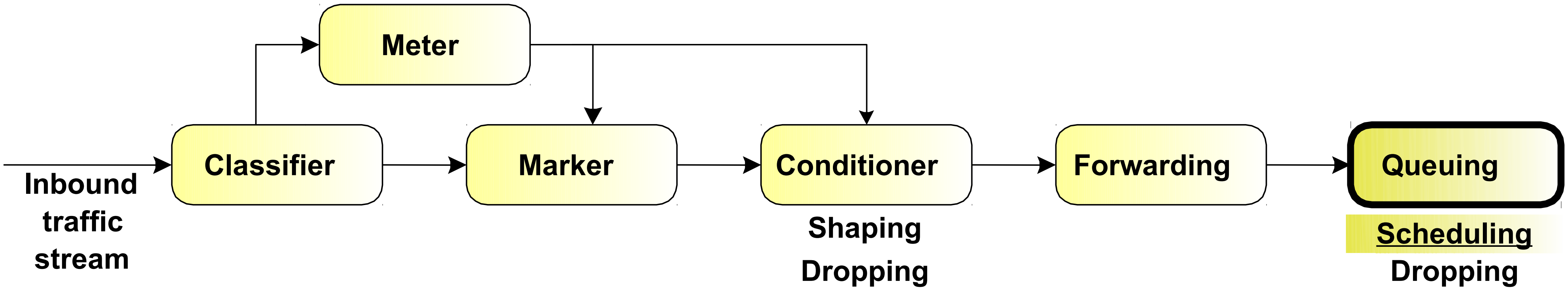
- Dropping mechanisms
  - Committed Access Rate (CAR) and Class-based Policing can drop packets that exceed the contractual rate
  - Weighted Random Early Detection (WRED) can randomly drop packets when an interface is nearing congestion

DiffServ Mechanisms



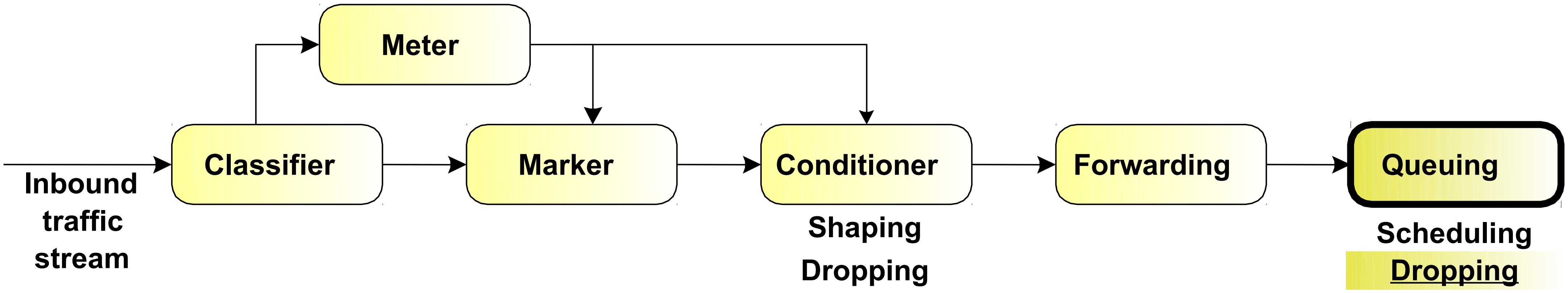
- Forwarding mechanisms
  - Routing
  - e.g. Cisco Express Forwarding (CEF)

DiffServ Mechanisms



- Traditional queuing mechanisms
  - FIFO, Priority Queuing (PQ), Custom Queuing (CQ)
- Weighted Fair Queuing (WFQ) family
  - WFQ, dWFQ, CoS-based dWFQ, QoS-group dWFQ
- Advanced queuing mechanisms
  - Class-based WFQ, Class-based LLQ

DiffServ Mechanisms

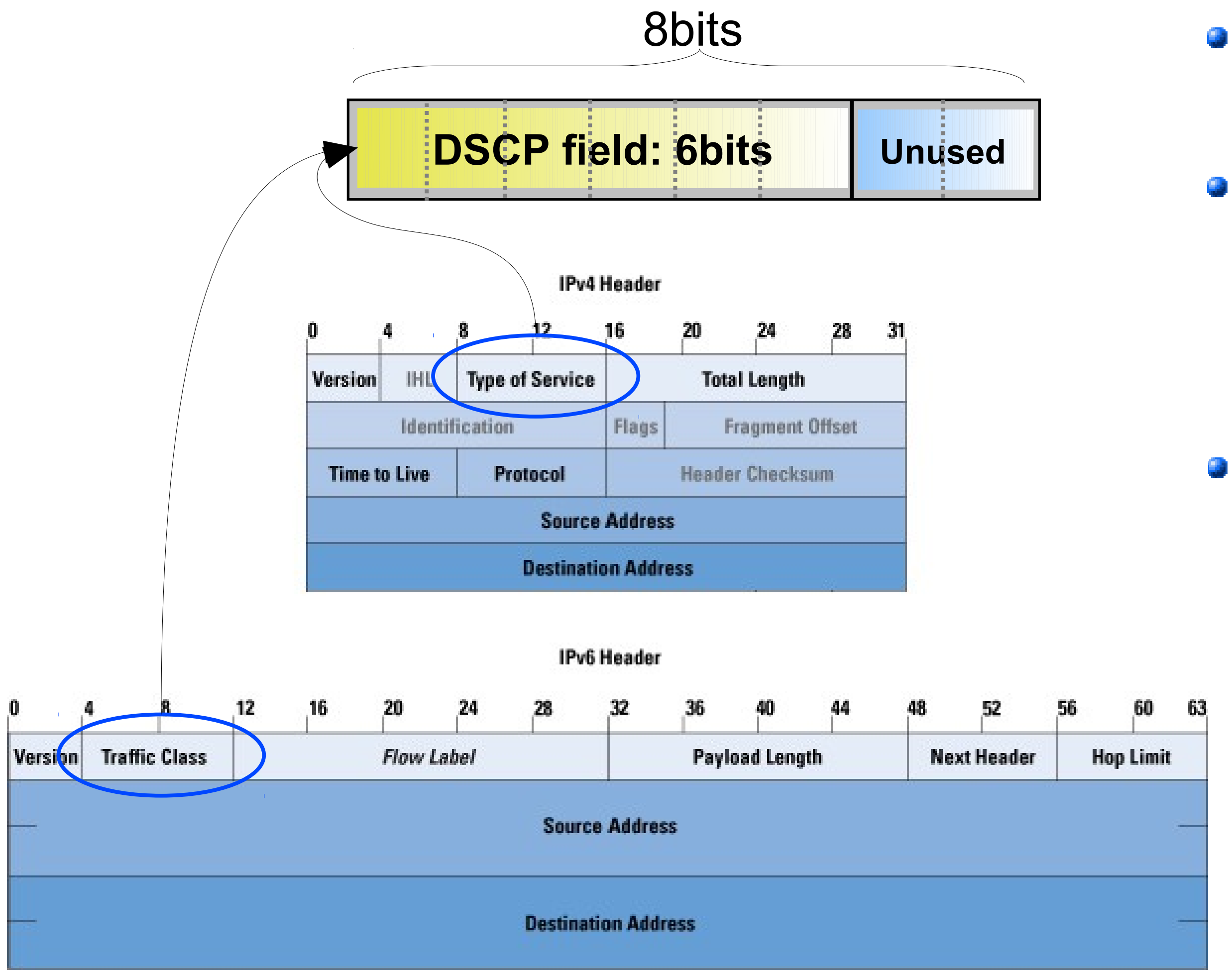


- Dropping mechanisms
  - Tail drop on queue congestion
  - WFQ has an improved tail-drop scheme
  - WRED randomly drops packets when nearing congestion

Functional elements

- The functional elements of the DiffServ architecture are:
- Edge Routers:
  - Classify packets: they mark each packet on the Type of Service field of the IPv4 header or Traffic Class field of the IPv6 header
  - Condition traffic: for example, they use a “Token Bucket” to verify if the incoming traffic is conforming to the contracted traffic and, if not,
    - delay excess traffic or
    - drop excess traffic
- Core Routers:
  - Identify treatment that should be given to packets based on their mark and according to a Per-Hop-Behavior (PHB)

Edge Routers: Traffic classification



- All classification and QoS revolves around the DSCP field
- Format
  - DSCP – Differentiated Service Code Point (6bits)
  - CU – Currently Unused (2bits)
- Packets are marked on the
  - Type of Service (TOS) field of the IPv4 header
  - Traffic Class field of the IPv6 header

Routing on *Core Routers*

- Different PHBs (*Per-Hop-Behaviors*) result on different network performances
- PHBs do not specify which queuing mechanisms should be used
- PHBs examples:
  - x% of the physical bandwidth is attributed to packets from Class A during any time interval of a given specified length
  - Packets from Class A are always served before Class B packets
  - Packets from Class A are served with twice the service bandwidth that is attributed to Class B packets

Per Hop Behaviors

- The Default PHB
  - Traditional best effort service
    - DSCP value (recommended) of “000000”
- Class-Selector PHBs
  - To preserve backward compatibility with the IP-precedence scheme
    - DSCP values of the form “xxx000”
  - These PHBs ensure that DS-compliant nodes can co-exist with IP-precedence aware nodes
- Expedited Forwarding (EF) PHB
  - Provides a low-loss, low-latency, low-jitter, and assured bandwidth service
  - Recommended DSCP value for EF is “101110”
- Assured Forwarding PHB
  - Can be provide different forwarding assurances.
    - For example, traffic can be divided into gold, silver, and bronze classes, with gold being allocated 50 percent of the available link bandwidth, silver 30 percent, and bronze 20 percent
  - The AFxy PHB defines four AFx classes: AF1, AF2, AF3, and AF4

Expedited Forwarding

- Expedited Forwarding (EF) PHB:
  - Ensures a minimum departure rate
  - Guarantees bandwidth – the class is guaranteed an amount of bandwidth with prioritized forwarding
  - Polices bandwidth – the class is not allowed to exceed the guaranteed amount (excess traffic is dropped)
- DSCP value: “101110”; looks like IP precedence 5 to non-DS compliant devices

EF PHB Implementations

- Priority Queuing
- IP RTP Prioritization
- Class-based Low-latency Queuing (CB-LLQ)
- Strict Priority queuing within Modified Deficit Round Robin (MDRR)

Assured Forwarding

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

DiffServ Service Classes

- *Default* (DE) $\rightarrow$ DSCP = 000000
  - *best-effort* service with a single FIFO-type queue
- *Expedited Forwarding* (EF) $\rightarrow$ DSCP = 101110
  - Service of the “virtual leased line” type
  - Provides control for losses, delay and delay variance inside a specified maximum bandwidth
- *Assured Forwarding* (AF)
  - Provides a relative Quality of Service (AF*i* is served with more bandwidth than AF*j*, for *i*<*j*)
  - On each class, there are 3 precedence levels for dropping packets in case of congestion

AF Codepoints	**AF1**	**AF2**	**AF3**	**AF4**
Low drop precedence	**001010**	**010010**	**011010**	**100010**
Medium drop precedence	**001100**	**010100**	**011100**	**100100**
High drop precedence	**001110**	**010110**	**011110**	**100110**

AF PHB Definition

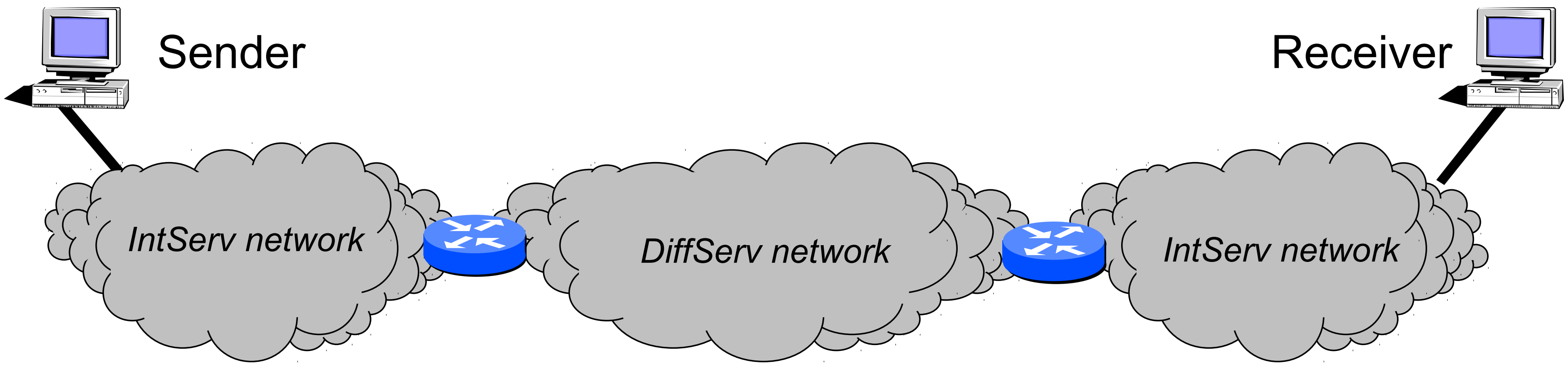
- A DS node MUST allocate a configurable, minimum amount of forwarding resources (buffer space and bandwidth) per AF class
- Excess resources may be allocated between non-idle classes. The manner must be specified.
- Reordering of IP packets of the same flow is not allowed if they belong to the same AF class

AF PHB Implementation

- CBWFQ (4 classes) with WRED within each class
- (M)DRR with WRED within each class
- Optionally Custom Queuing (does not support differentiated dropping)

IntServ and *DiffServ* integration

- Use:
  - *IntServ* architecture (appropriate for small networks) on access networks
  - *DiffServ* architecture (appropriate for large networks) on transit networks
- Border routers of both network types:
  - Classify RSVP requests on the appropriate *DiffServ* service classes
  - If the are no sufficient resources, they refuse the RSVP reservation requests
- Advantages:
  - Provide *IntServ* services on large networks
  - Provide explicit admission control instead of SLAs on *DiffServ networks*



DiffServ Domains and SLAs

- Quality of Service that is provided to a client is configured:
  - By management (traffic conditioning configuration is made on the respective Edge Router)
  - According to the Service Level Agreement (SLA)