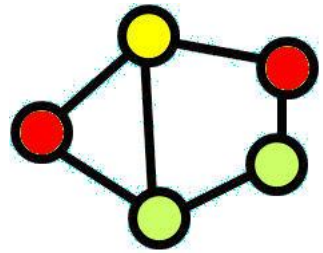
Slide supporting material



Lesson 14: QoS in IP Networks: IntServ and DiffServ

Giovanni Giambene

Queuing Theory and Telecommunications: Networks and Applications
2nd edition, Springer

Introduction

QoS Support in IP Networks

- The introduction of real-time traffic in the Internet (e.g., Voice over IP, VoIP) calls for new approaches to provide Quality of Service (QoS).
- Internet that operates on the basis of **Best Effort (BE)** does not provide QoS support (no bandwidth guarantees, no delay guarantees, no admission control, and no assurances about delivery).
- Real-time traffic (as well as other applications) may require **priority** treatment to achieve good performance.

An Example on the Need of QoS Support in IP Networks

- Let us consider a phone application at 1 Mbit/s and an FTP application sharing a bottleneck link at 1.5 Mbit/s.
- Bursts of FTP can congest the router and cause voice packets to be dropped.
- In this example **we need to give priority to voice over FTP**.
  - **Marking of packets** is needed for the router to distinguish between different classes; and new **router policy** is needed to treat packets accordingly.

QoS Metrics

- Main performance attributes:
  - Bit error rate [%] at PHY layer
  - Outage probability [% of time] at PHY layer
  - Blocking probability [%] at PHY or MAC layer
  - Throughput [bit/s] at MAC or transport layer
  - Packet loss rate [%] at MAC and IP layers (e.g., buffer overflow)
  - Fairness at PHY, MAC or transport layers
  - (Mean) delay [s] at different layers
  - Delay variation or jitter [s] at different layers (especially, application)
- The **Service Level Agreement (SLA) is a contract** between the user and the service provider/operator, which defines suitable bounds for some of the QoS performance attributes above provided that the user traffic fulfills certain characteristics.

I. Stoica, "Stateless Core: A Scalable Approach for Quality of Service in the Internet", *in Lecture Notes in Computer Science*, Vol. 2979, 2001.

IntServ

IntServ & DiffServ

- The key QoS approaches described in this lesson for IP-based networks are:
  - **Integrated Services (IntServ)** in RFC 1633 and RFC 2207.
  - **Differentiated Services (DiffServ)** in RFC 2474 and RFC 2475.
- Note that in both cases CAC schemes are adopted:
  - Traffic flow-based **deterministic CAC with IntServ**,
  - Traffic class-based **statistic CAC with DiffServ**.

IntServ

- The IntServ main concept is to **reserve resources** for **each flow** through the network. **There are per-flow queues at the routers.**
- IntServ adopts an **explicit call set-up mechanism** for the routers in a source-to-destination path. These mechanisms enable each flow to request a specific QoS level.
- **RSVP (Resource reSerVation Protocol) is the most-widely-used set-up mechanism** enabling resource reservation over a specific source-to-destination path (RFC 2205 and RFC 2210). RSVP operates end-to-end.
- RSVP allows a fine bandwidth control. The main drawback of RSVP is the adoption of per-flow state and per-flow processing that cause **scalability issues** for large networks (**heavy processing and signaling loads** at routers).

IntServ (cont'd)

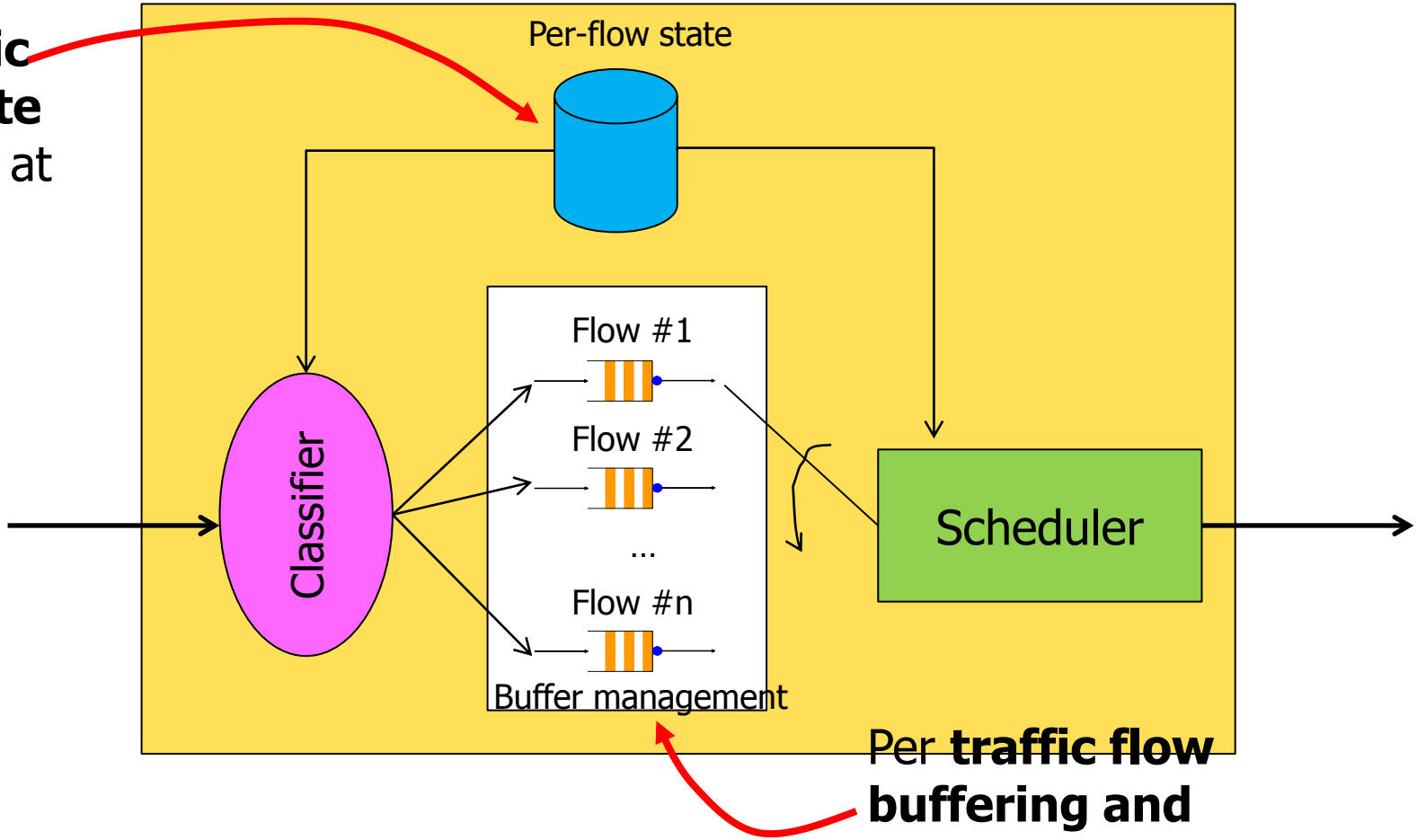
- RSVP uses two types of **FlowSpecs** used by routers to set a path:
  - **Traffic specification (T-Spec)** describing the traffic characteristics of the source according to a **token bucket model** with parameters: bucket depth b, token generation rate r, peak data rate p, etc.).
  - **Request specification (R-Spec)** that describes the required service level and is defined by the receiver.
- **T-Spec is sent from source to destination. R-Spec is sent back from destination to source.**
- **CAC and resource reservation along the source-destination path is performed on a traffic flow basis by RSVP and using both T-Spec and R-Spec.**
  - Routers will admit new flows based on their R-spec and T-spec and based on the current resources allocated at the routers to other flows.

T-Spec and R-Spec in Detail

- **T-Spec specifies the traffic characteristics of sender**
  - Bucket rate and sustainable rate (r) (bits/s)
  - Peak rate (p) (bits/s)
  - Bucket depth (b) (bits)
  - Minimum policed unit (m) (bits) – any packet with size smaller than m will be counted as m bits
  - Maximum packet size (M) (bits) – the maximum packet size that can be accepted.
- **R-Spec defines the resource needed for the flow and requested by receiver (bandwidth requirement)**
  - Service rate (R) (bits/s): bandwidth that is needed for the traffic flow.
  - Slack term (S) (μs): extra amount of delay that a node may tolerate still meeting the end-to-end delay requirement of the traffic flow.

IntServ: Internal Node Structure

Per **traffic flow state** managed at the node

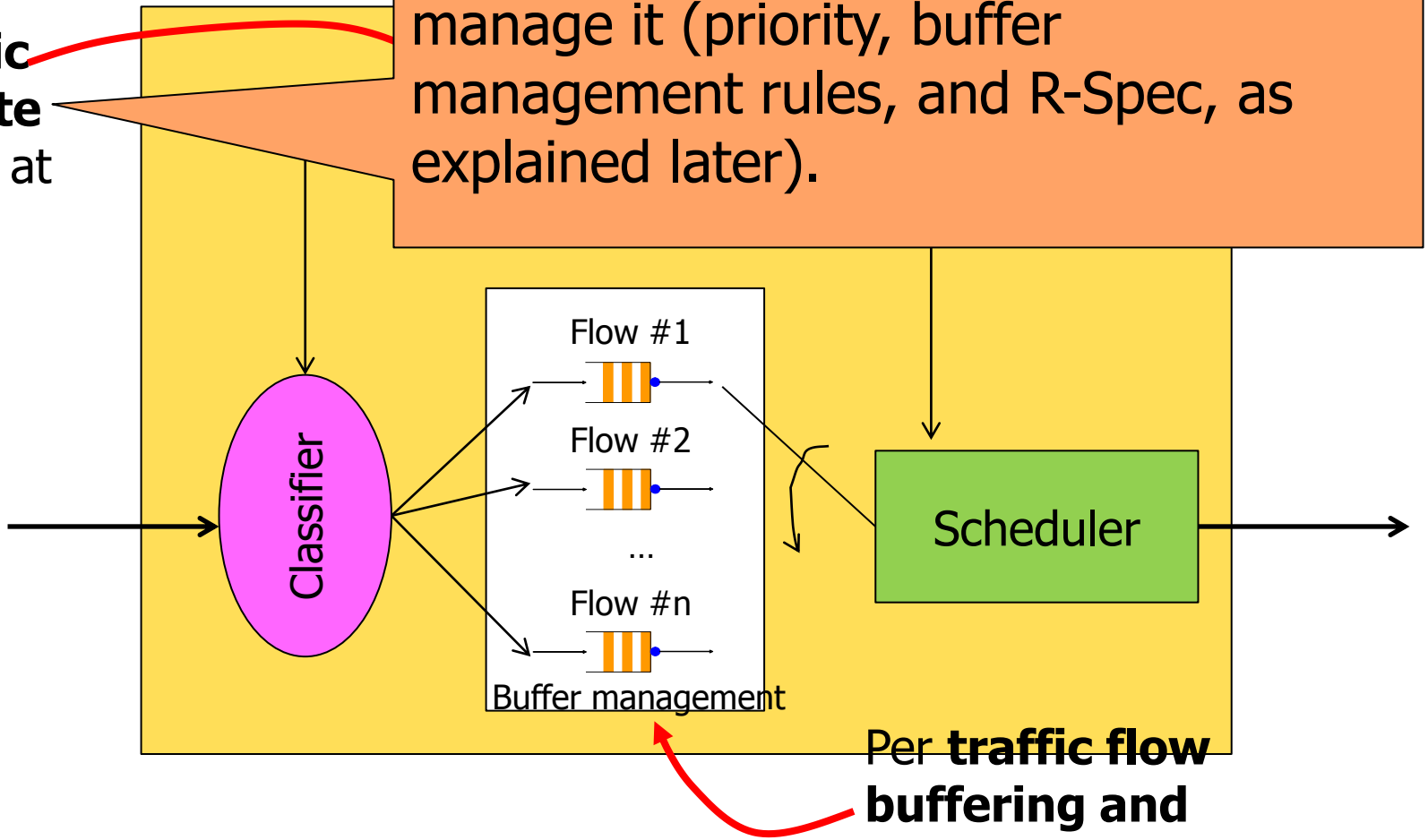


Per **traffic flow buffering and scheduling** at the node.

IntServ: Internal Node Structure

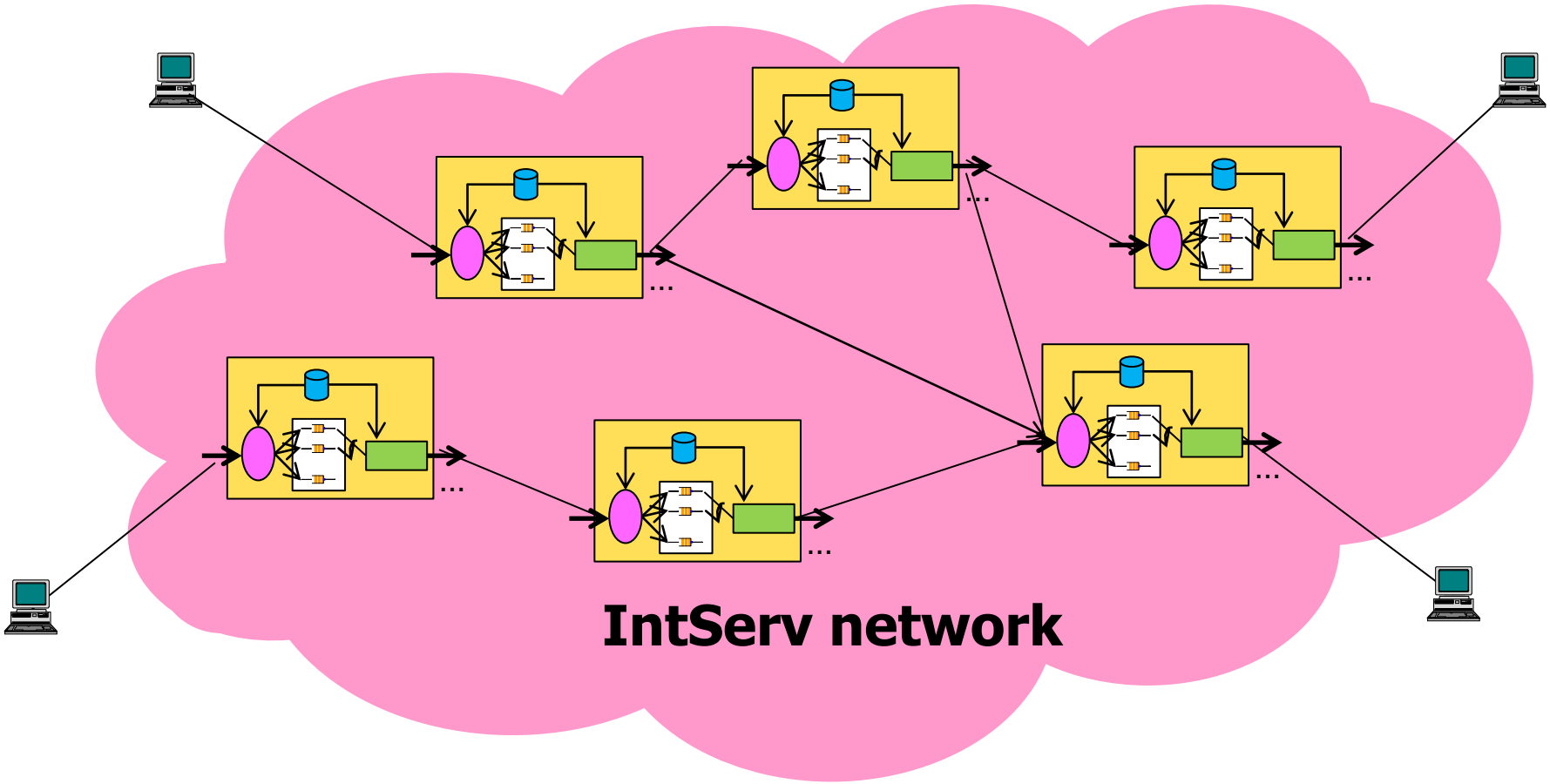
The flow state contains a flow identifier, and instructions on how to manage it (priority, buffer management rules, and R-Spec, as explained later).

Per **traffic flow state** managed at the node



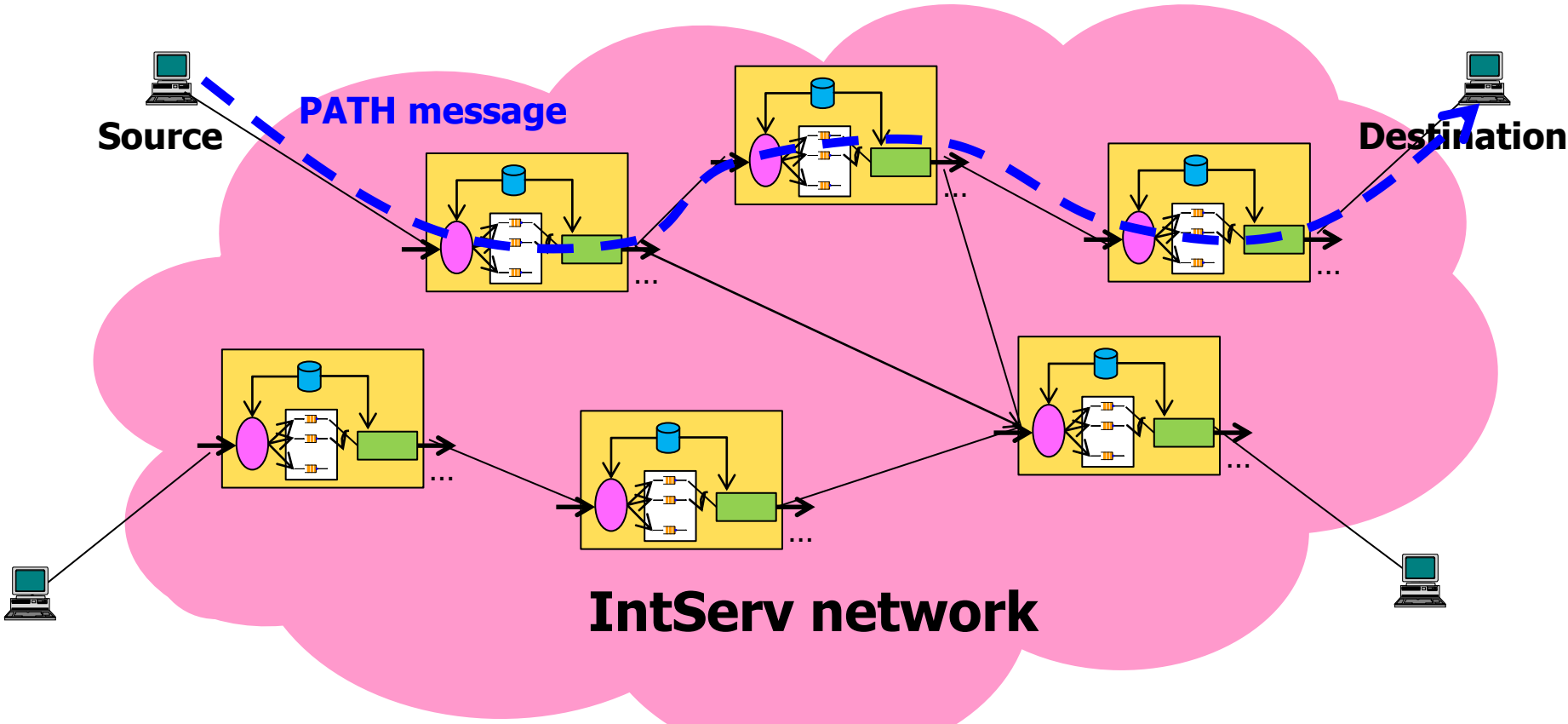
Per **traffic flow buffering and scheduling** at the node.

IntServ Example



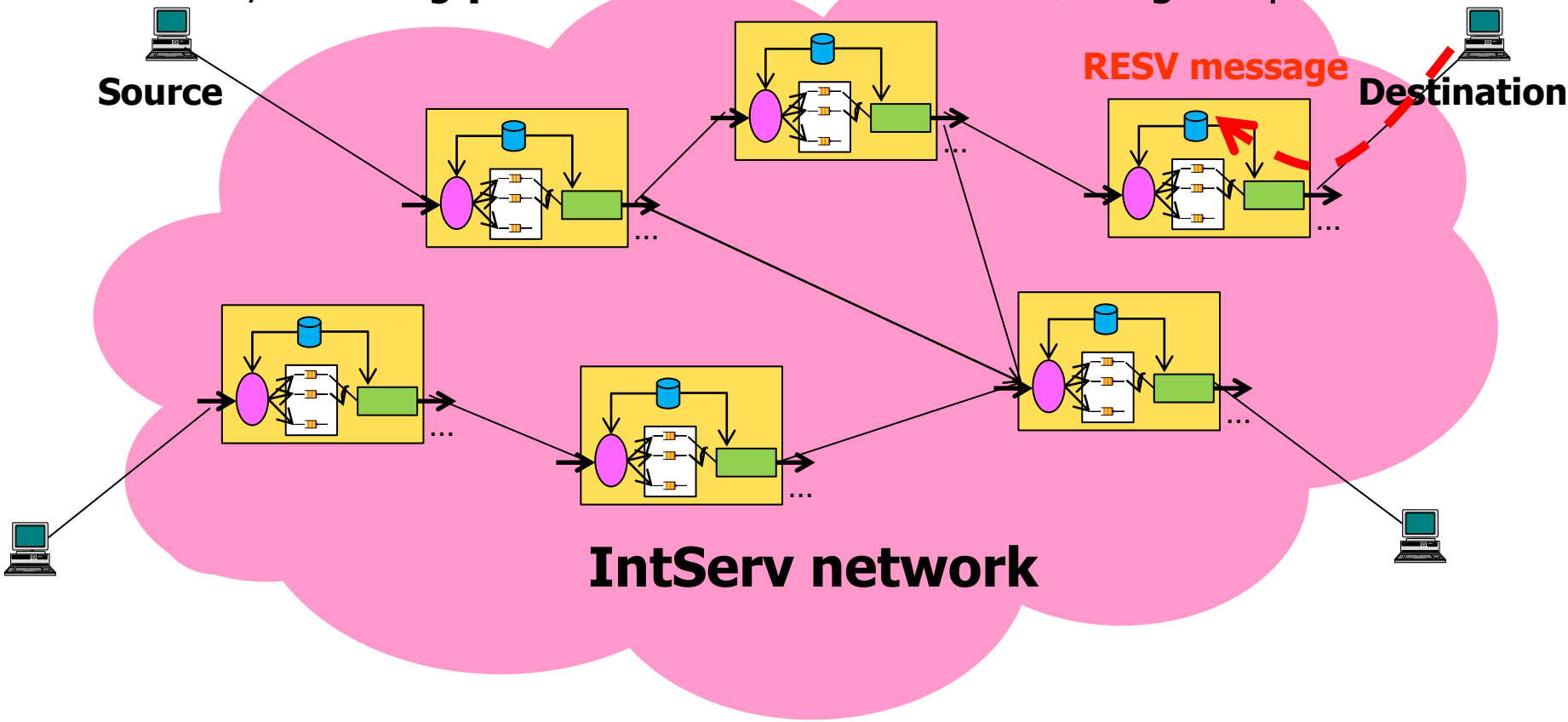
IntServ Example

- Propagating the PATH message with T-Spec from source to destination to establish a path.



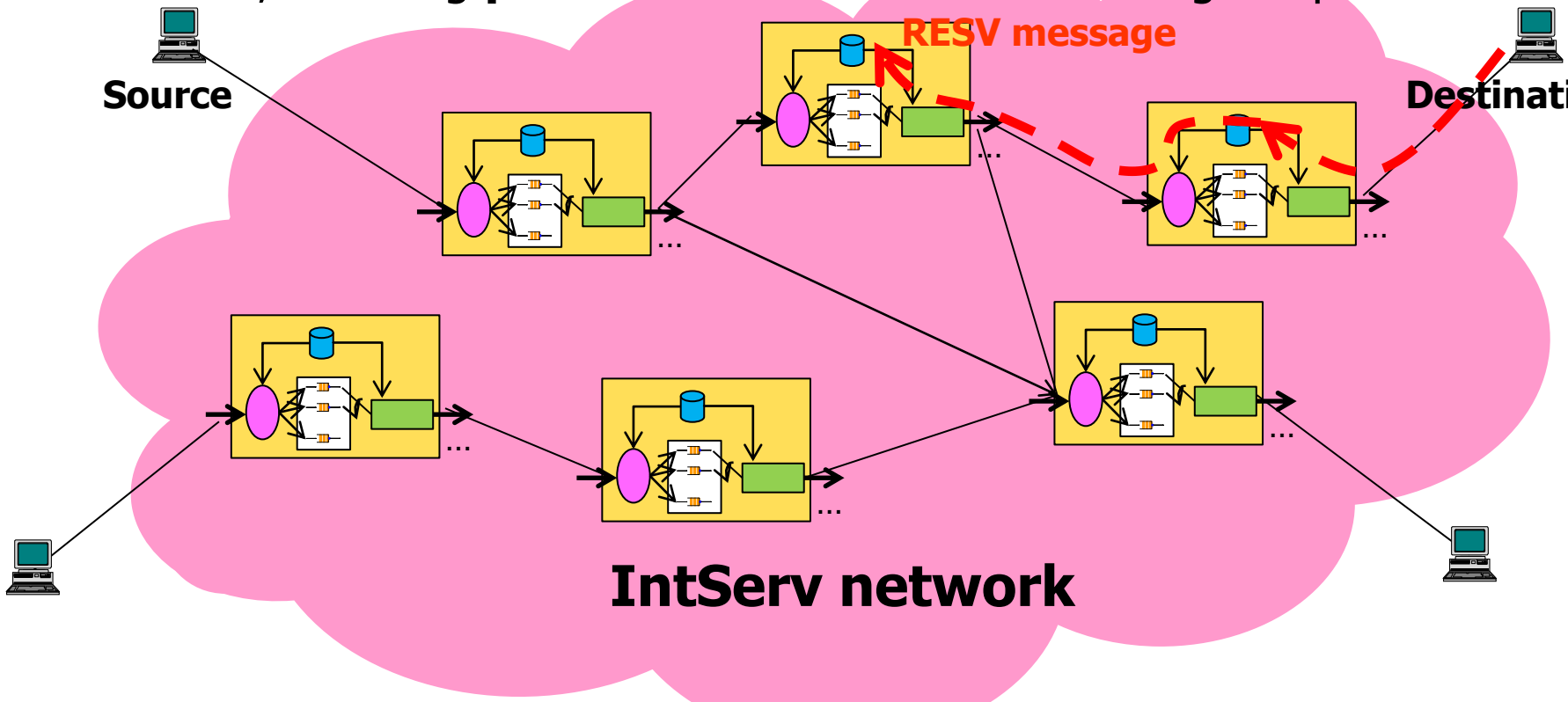
IntServ Example

- RESV message providing back R-Spec to be used by each node along the path for per-flow admission control and resource allocation; installing **per-flow state** in the nodes along the path.



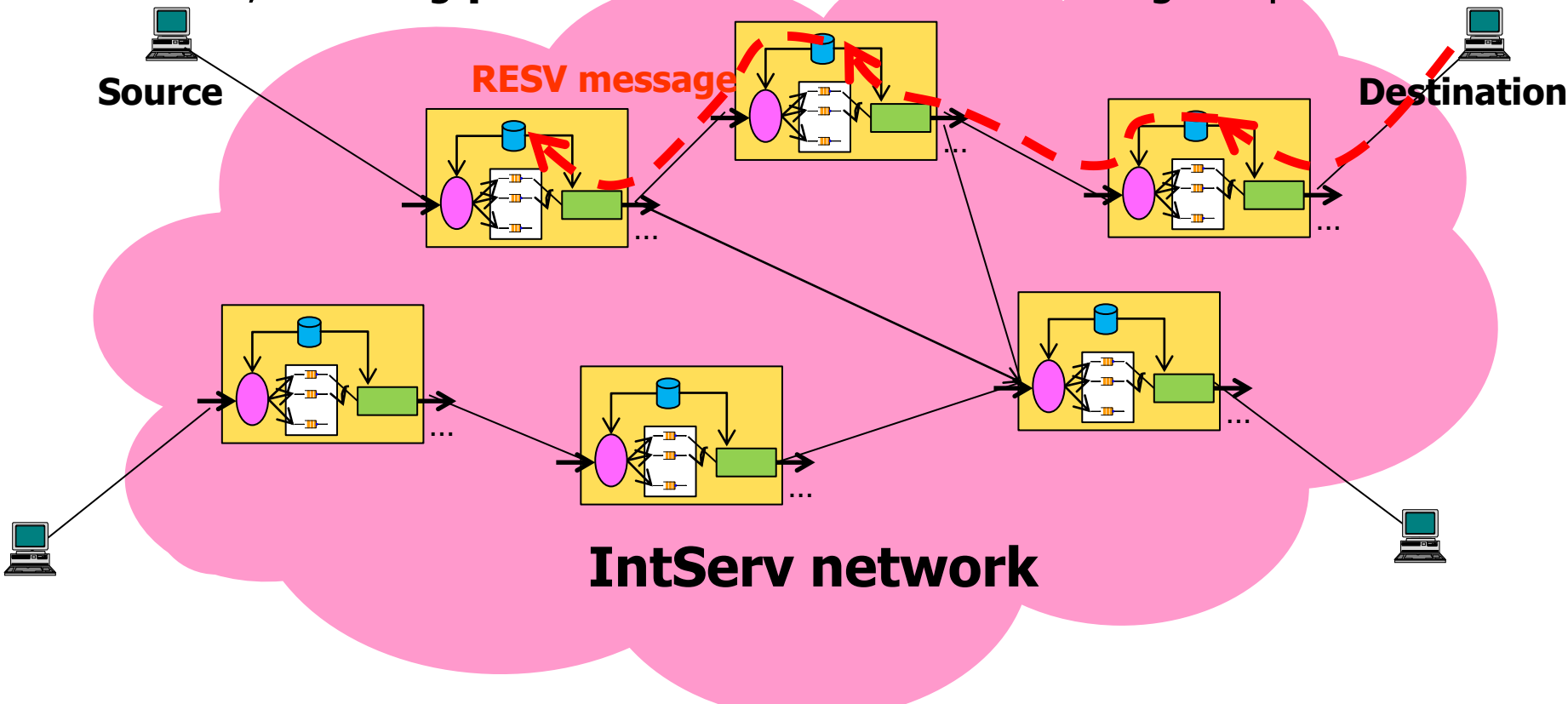
IntServ Example

- RESV message providing back R-Spec to be used by each node along the path for per-flow admission control and resource allocation; installing **per-flow state** in the nodes along the path.



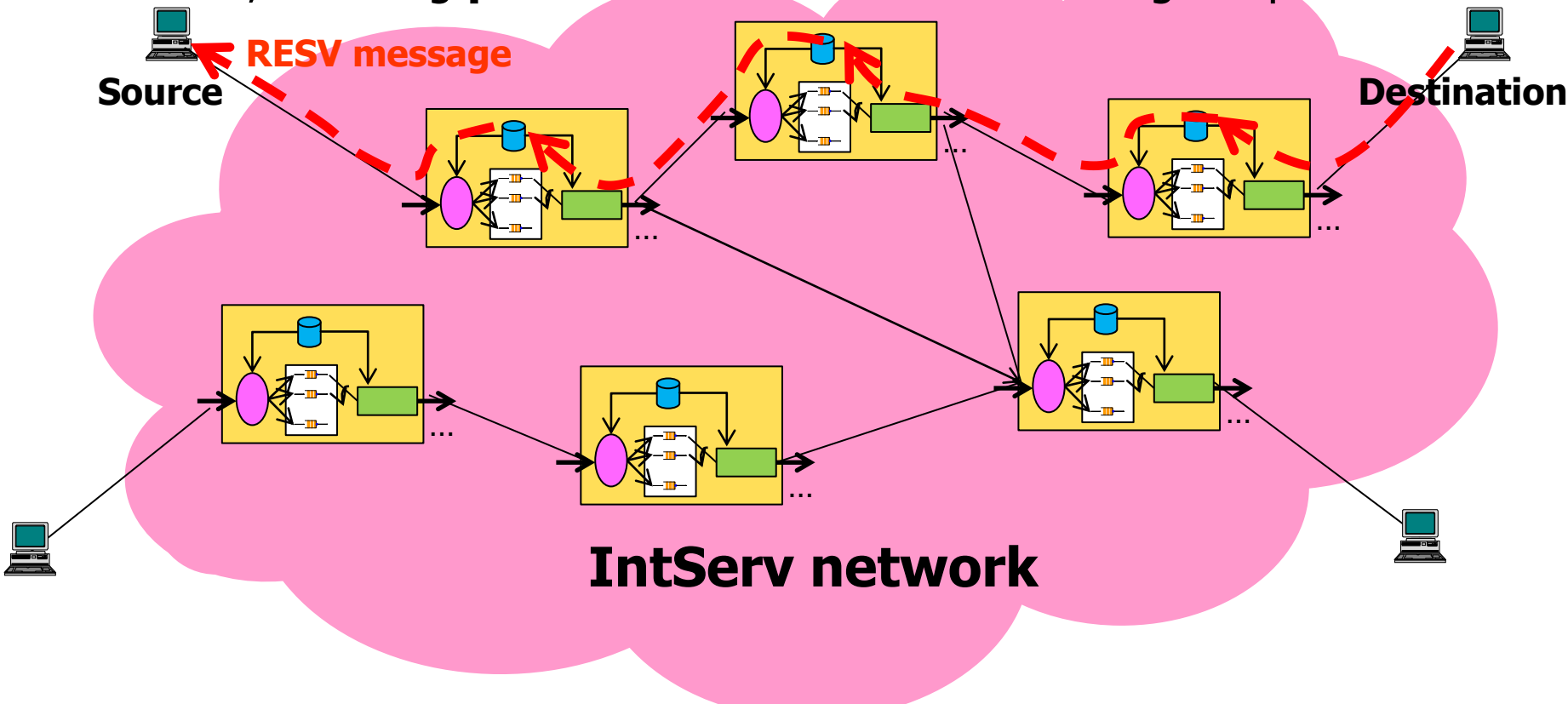
IntServ Example

- RESV message providing back R-Spec to be used by each node along the path for per-flow admission control and resource allocation; installing **per-flow state** in the nodes along the path.



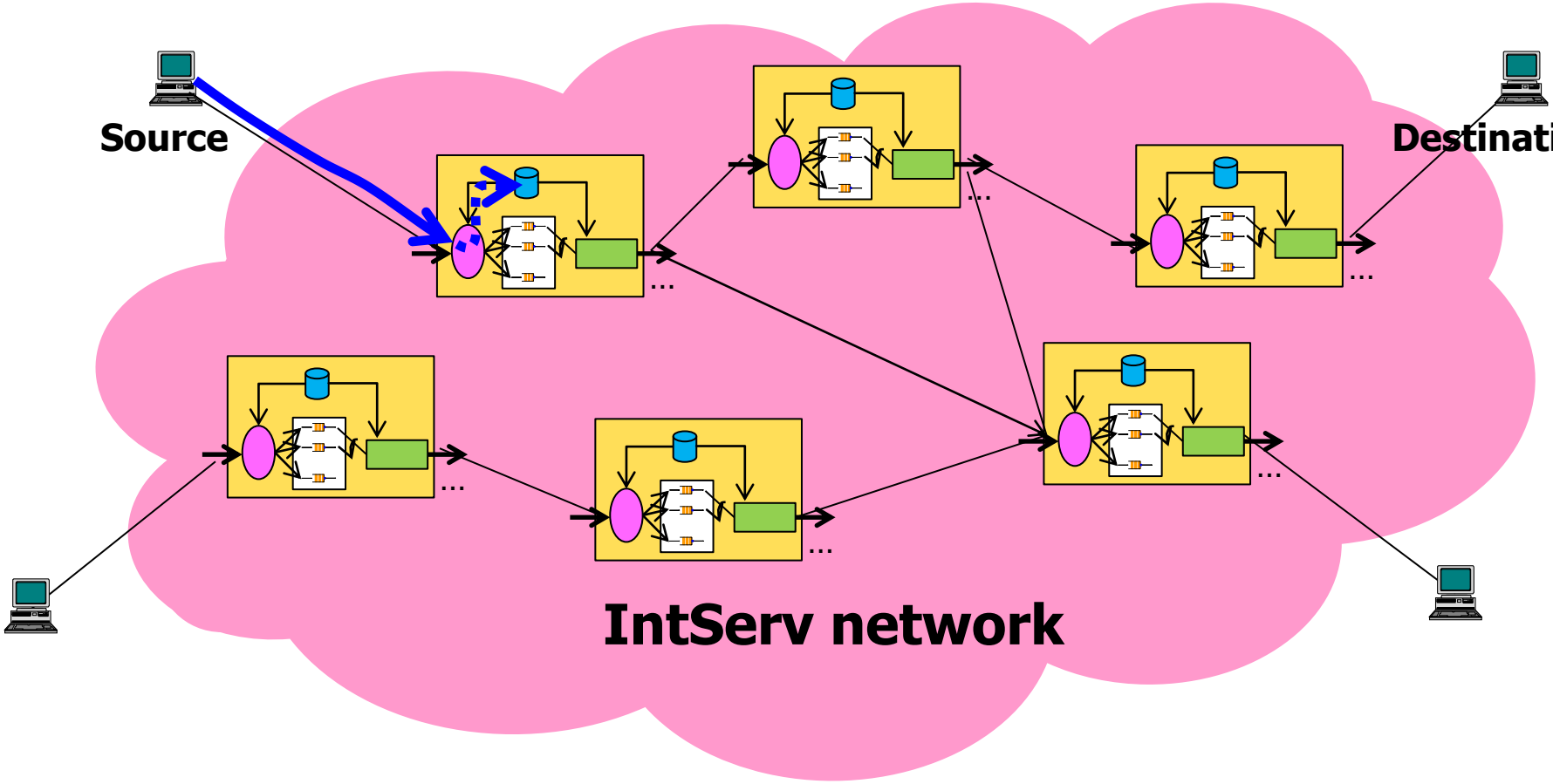
IntServ Example

- RESV message providing back R-Spec to be used by each node along the path for per-flow admission control and resource allocation; installing **per-flow state** in the nodes along the path.



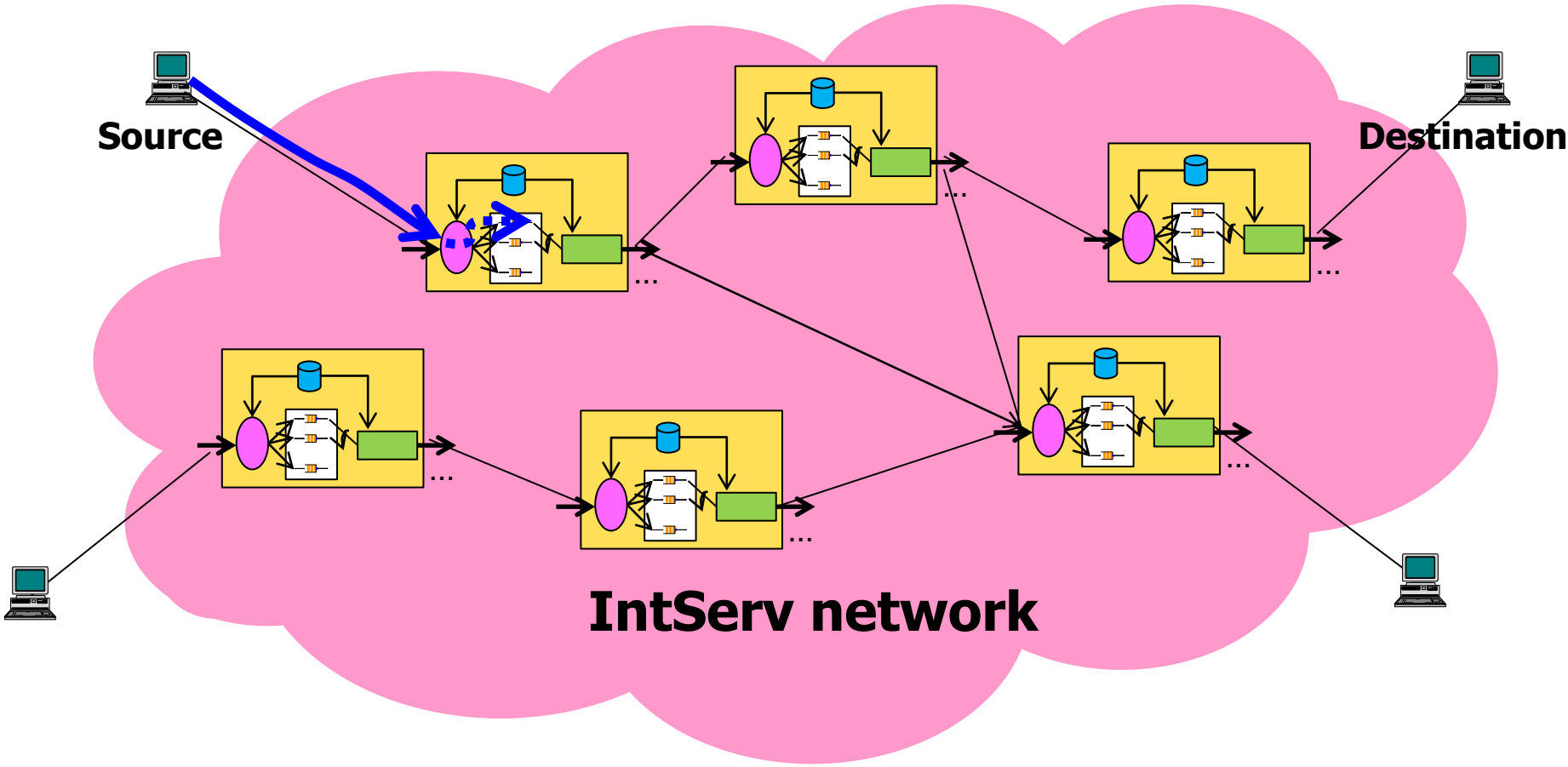
IntServ Example

- Traffic delivery: use of **per-flow classification**.



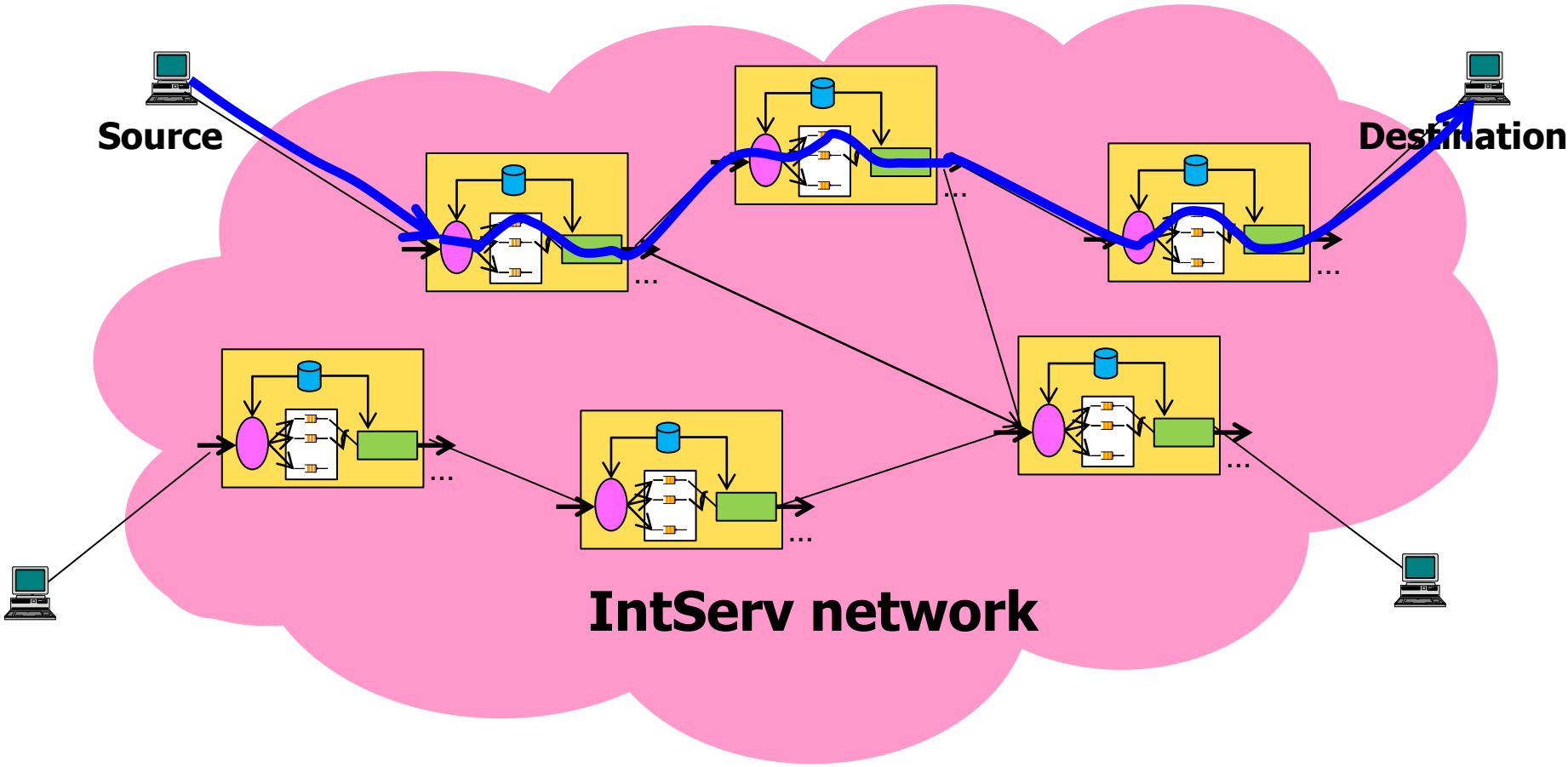
IntServ Example

- Traffic delivery: use of **per-flow buffer management**.



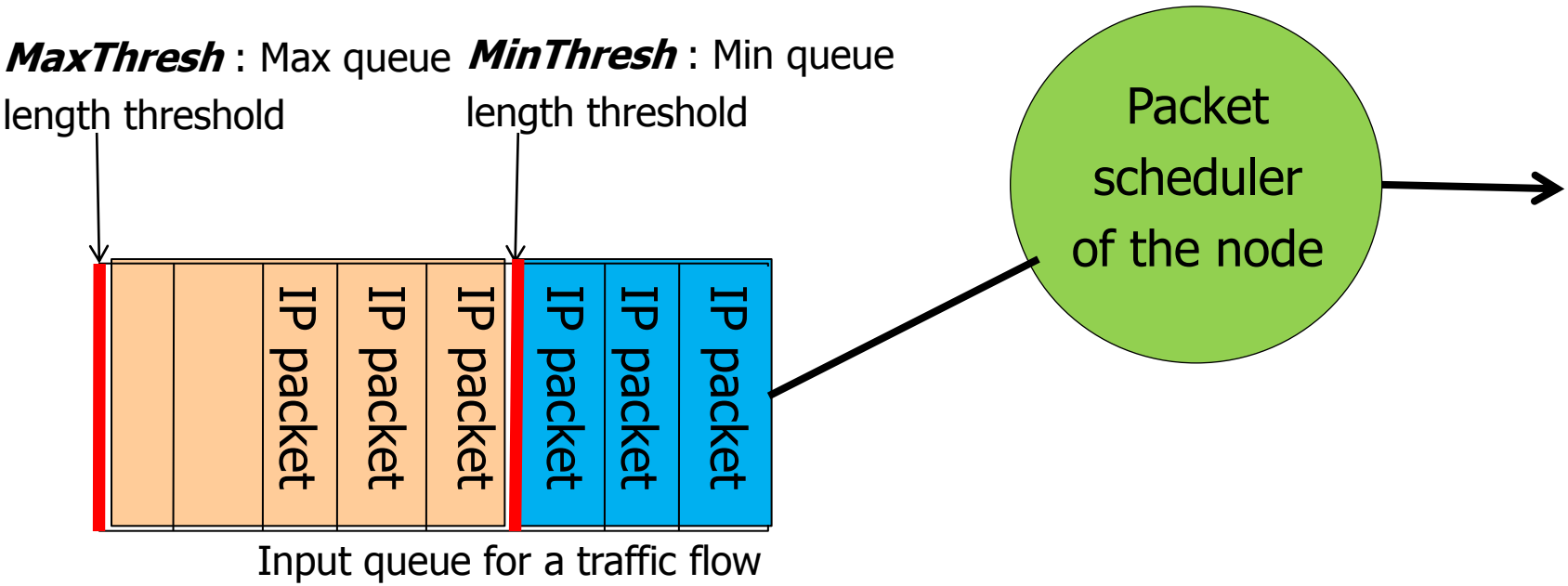
IntServ Example

- Traffic delivery: use of **per-flow traffic scheduling** at the nodes.



IntServ: Buffer Management

- Instead of using a simple drop-tail mechanism, buffer management is adopted by IntServ. Let us consider the following definitions related to the management of traffic at a generic buffer in the IntServ router.



IntServ: Buffer Management (cont'd)

- **Random Early Detection (RED)**

  IP packets are dropped randomly with a given probability when the average queue length exceeds a minimum threshold (MinThresh). If a maximum threshold (MaxThresh) is exceeded, all new IP packets are dropped.

- **Weighted RED (WRED)**

  This technique drops IP packets selectively on the basis of the IP precedence.

IntServ: Class of Service

- IntServ allows two service types:
  - **Guaranteed Service (GS)**
    - For hard real-time applications.
    - The user specifies traffic characteristics.
    - Requires admission control at each router.
    - Can mathematically guarantee bandwidth, delay, and jitter (**deterministic guarantees**).
  - **Controlled-Load Service (CLS)**
    - For applications that can adapt to network conditions within a certain performance window.
    - The user specifies traffic characteristics.
    - Requires admission control at each router.
    - Guarantees are not as strong as with the guaranteed service (**statistical guarantees based on average values**).

IntServ: Guaranteed Service

- GS provides **quantitative QoS guarantee (i.e., guaranteed bandwidth and strict bounds on end-to-end delay) on a flow basis**.
- GS can manage applications with **stringent real-time delivery requirements**, such as audio and video applications.
- With GS, each router **guarantees a bandwidth R and a certain buffer space B for each traffic flow.**
- **The sender sends an RSVP-PATH message** to the receiver specifying the traffic characteristics (T-Spec) and setting up the path. **The receiver computes R and responds with an RESV-message** to request resources for the flow (R-Spec).

IntServ: Guaranteed Service (cont'd)

- A source is characterized according to a **fluid traffic model:** bit-rate as a function of time (no packet arrivals).
- GS uses **a token bucket filter (r, b, p) specified by T-Spec** to shape the traffic.
- **In a perfect fluid model, a flow conformant to a token bucket with rate r and depth b will have its delay bounded by b/R, provided that R ≥ r** [Parekh 1992, Cruz 1988].
- GS uses a **Weighted Fair Queuing (WFQ) scheduling** scheme at the routers to service the queues (one queue per flow).

A. K. J. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Service Networks", MIT Laboratory for Information and Decision Systems, Report LIDS-TH-2089, February 1992.

IntServ: Guaranteed Service (cont'd)

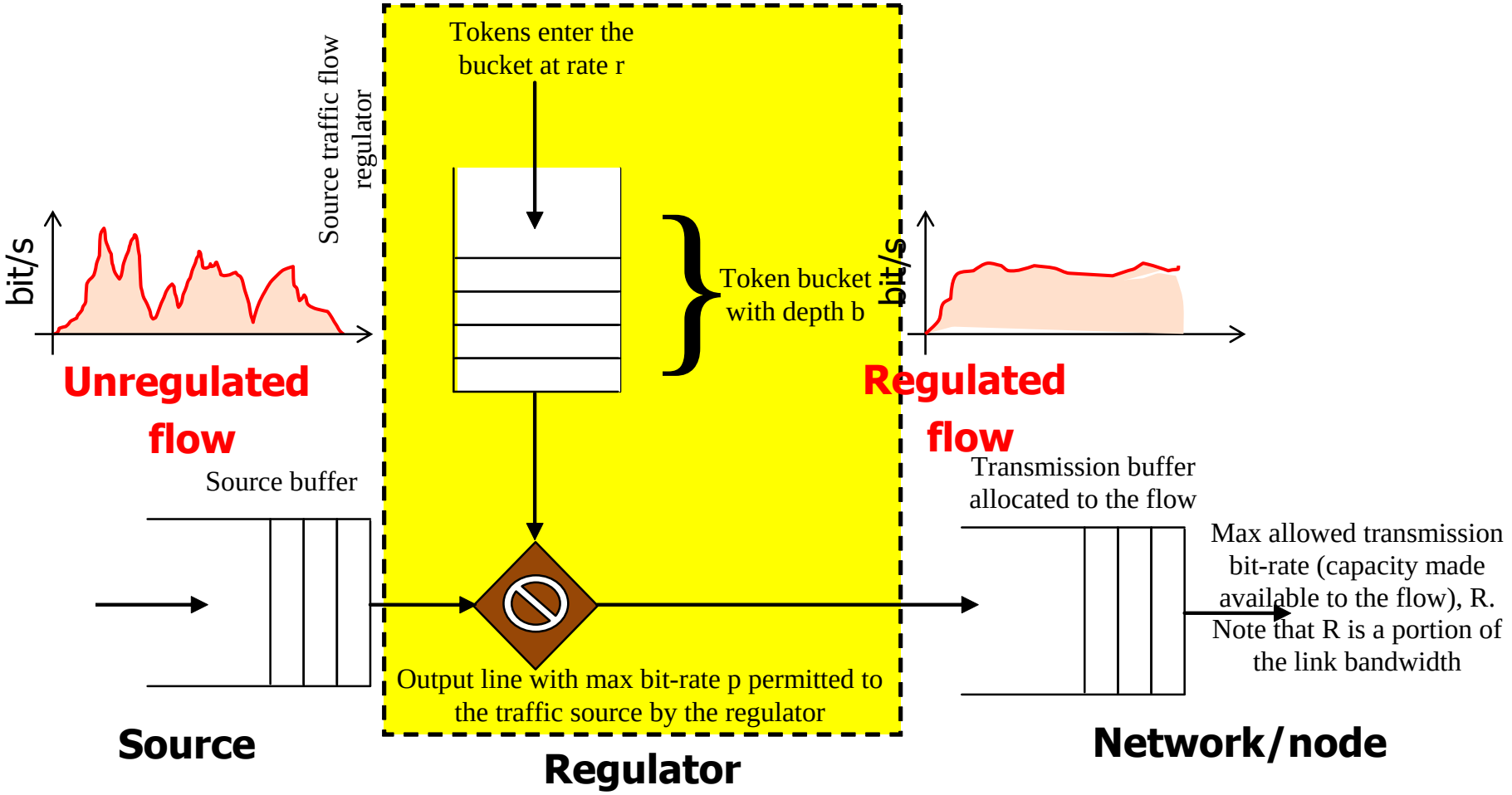
r = regime (mean) bit-rate
p = peak bit-rate
b = bucket depth

- A source is characterized [illegible] bit-rate as a function of time [illegible].
- GS uses **a token bucket filter (r, b, p) specified by T-Spec** to shape the traffic.
- **In a perfect fluid model, a flow conformant to a token bucket with rate r and depth b will have its delay bounded by b/R, provided that R ≥ r** [Parekh 1992, Cruz 1988].
- GS uses a **Weighted Fair Queuing (WFQ) scheduling** scheme at the routers to service the queues (one queue per flow).

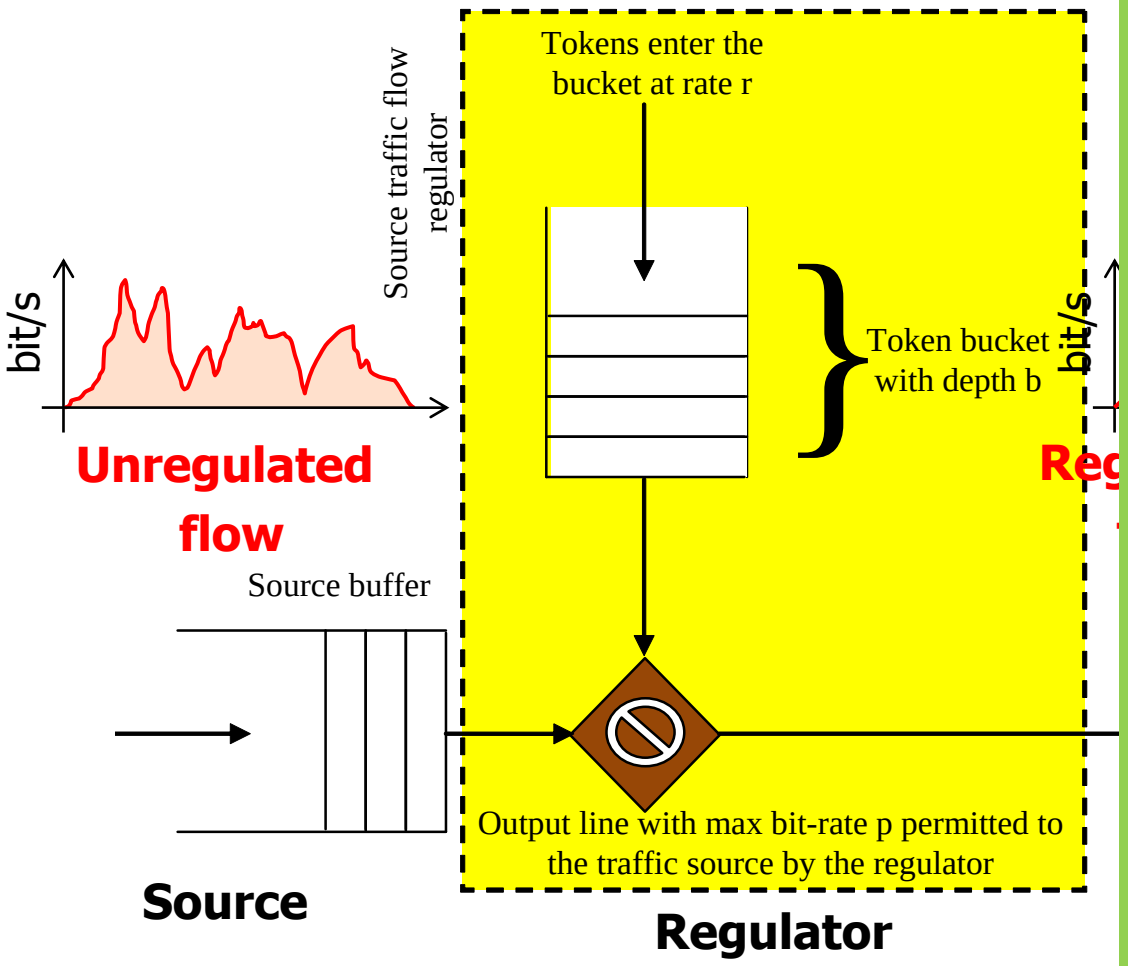
A. K. J. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Service Networks", MIT Laboratory for Information and Decision Systems, Report LIDS-TH-2089, February 1992.

Token Bucket Model and Deterministic Queuing

IntServ: Guaranteed Service - Token Bucket Shaper Model

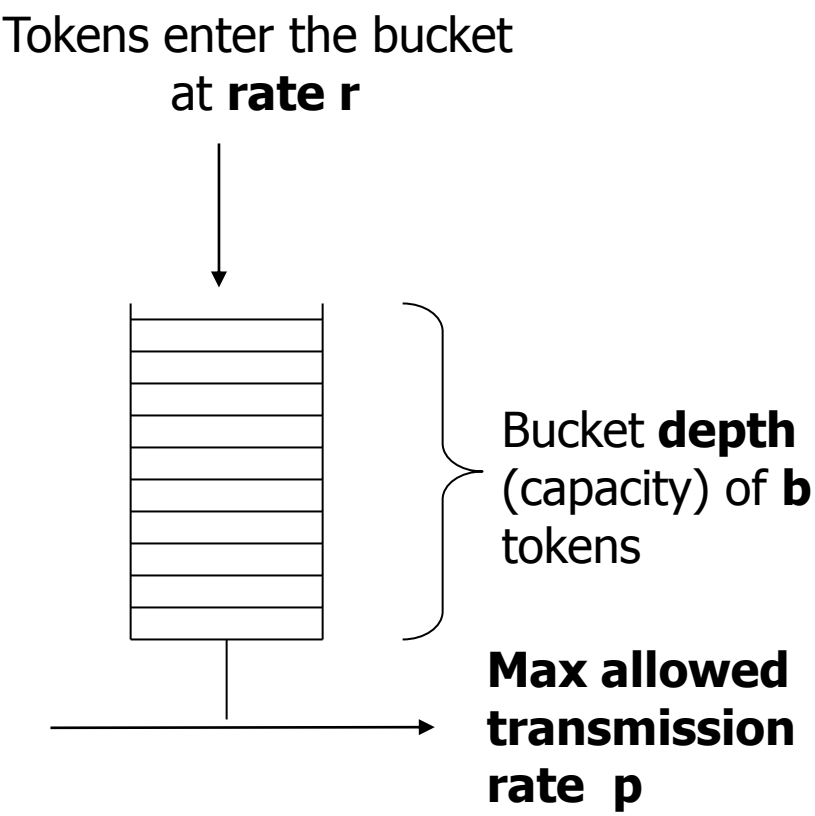


IntServ: Guaranteed Service - Token Bucket Shaper (cont'd)



- If the bucket is full, new tokens are discarded.
- Sending a packet of size L requires L tokens (**1 token for 1 bit**).
- If the bucket contains L tokens, the packet is sent at the maximum rate p, otherwise the packet is sent at a rate controlled by the token rate r.
- In this study we consider a **fluid-flow traffic** model: no packets (M = 0 and m = 0).

IntServ: Guaranteed Service - Token Bucket Shaper (cont'd)



- **We start with an empty buffer and a bucket full with b tokens.**
- The interval for which the token bucket allows sending a burst at the maximum rate p is T_b as:

$B = T_b p = b + r*T_b$ (**max burst size, MBS**)

- Hence, given the token bucket parameters r and b we obtain T_b as:

$$T_b = b /(p–r), \text{ assuming } r < p$$

- The number of bits sent in T_b is:

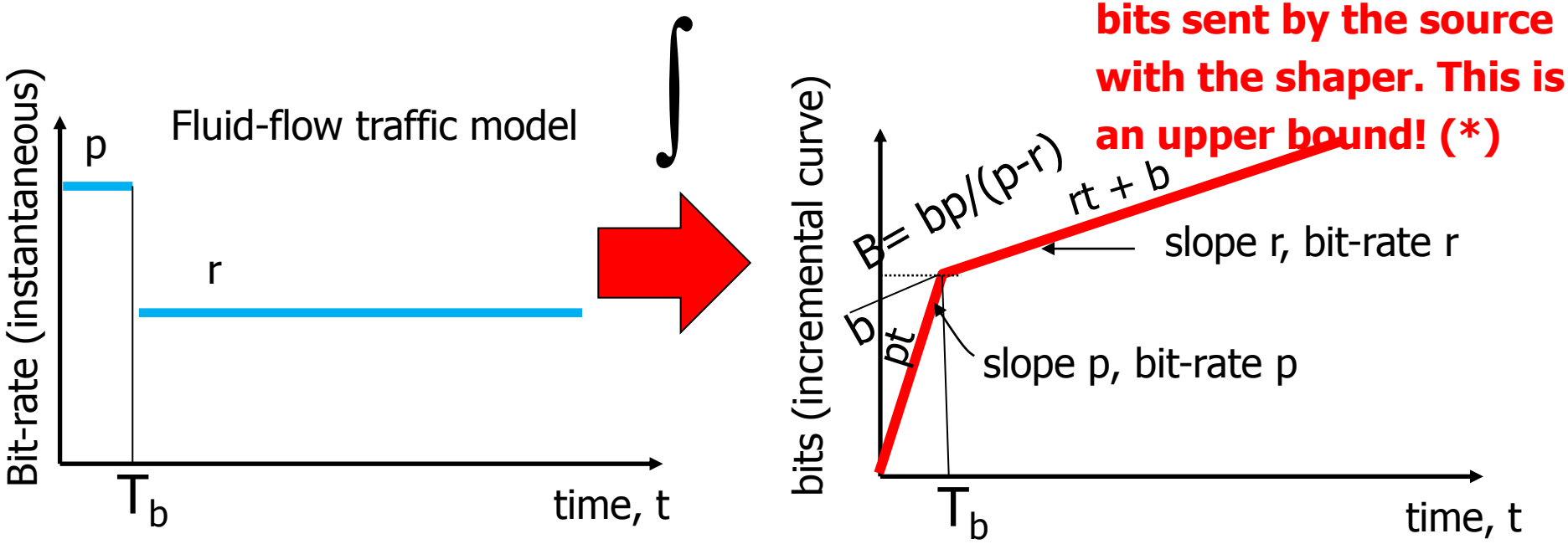
$$B = T_b p = bp /(p–r)$$

- After T_b, the output rate becomes equal to r.

IntServ: Guaranteed Service - Token Bucket Shaper (cont'd)

$\alpha(t)$ represents the arrival curve at the output of the shaper, this is the cumulative number of bits generated up to time t: $\alpha(t) = \min\{pt, rt + b\}$

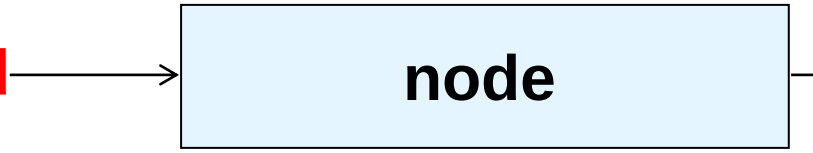
$\alpha(t)$ = Maximum # of bits sent by the source with the shaper. This is an upper bound! (*)



(*) The actual arrival curve coincides with the bound shown here only if the buffer of the traffic source is never empty.

IntServ: The Departure (output) Curve, β(t)

Input bit-rate with related α(t), according to the token bucket model (r, b, p)

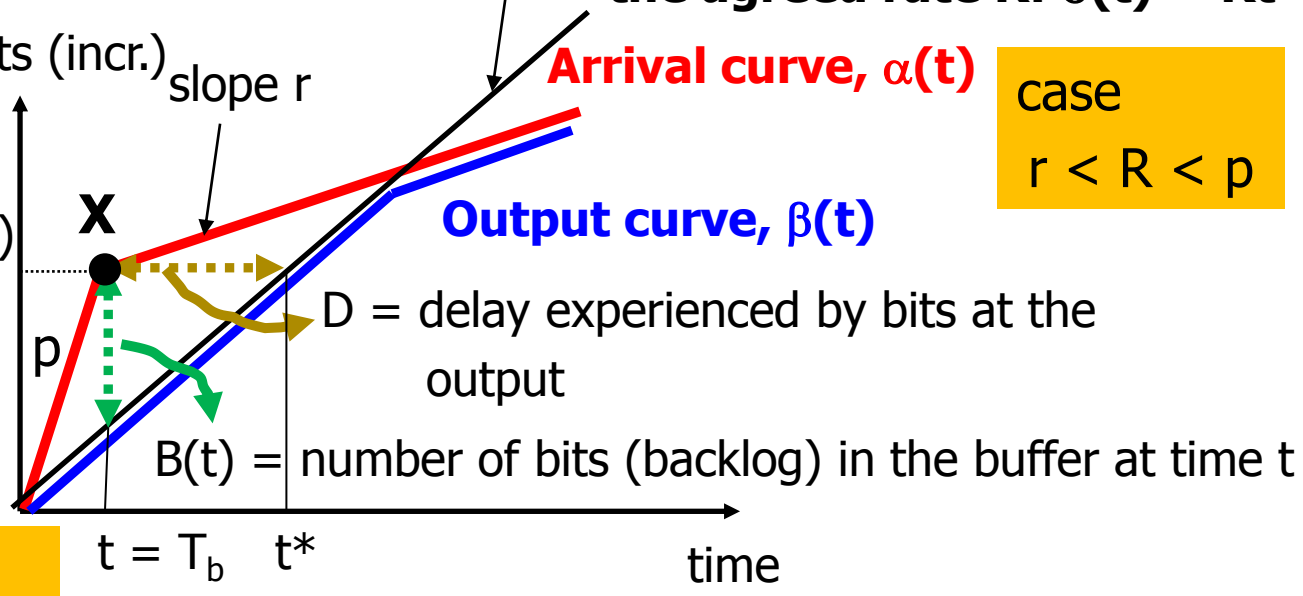


Output bit-rate and related cumulative curve β(t)

Service curve, σ(t) at the agreed rate R: σ(t) = Rt

The departure curve β(t) denotes the number of bits departing the node up to time t.

X is the point of the arrival curve corresponding to the largest buffer occupancy B_{max} and max delay D_{max}.



$\beta(t) = \min\{\sigma(t), \alpha(t)\}, \quad t > 0$

IntServ: QoS Guarantees and Per-hop Reservation

- This system is characterized by bounded delay (D_{max}) and bounded buffer size (maximum buffer occupancy B_{max}) determined as follows:

$$D_{\max} = t^* - T_b = \frac{b}{R} \times \left(\frac{p-R}{p-r} \right) \leq \frac{b}{R}, \quad \text{if } R \geq r$$

$$B_{\max} = pT_b - RT_b = b \times \left(\frac{p-R}{p-r} \right) \leq b, \quad \text{if } R \geq r$$

- **Given a traffic flow characterized by the token bucket model (r, b, p), each router along the path from source to destination has to allocate bandwidth R and a certain buffer B to fulfill the condition that the e2e delay is lower than a certain maximum value, Δ.**

IntServ: QoS Guarantees and Per-hop Reservation

- This system is characterized by bounded delay (D_{max}) and bounded buffer size (maximum buffer occupancy B_{max}) determined as follows:

$$D_{\max} = t^* - T_b = \frac{b}{R} \times \left(\right.$$

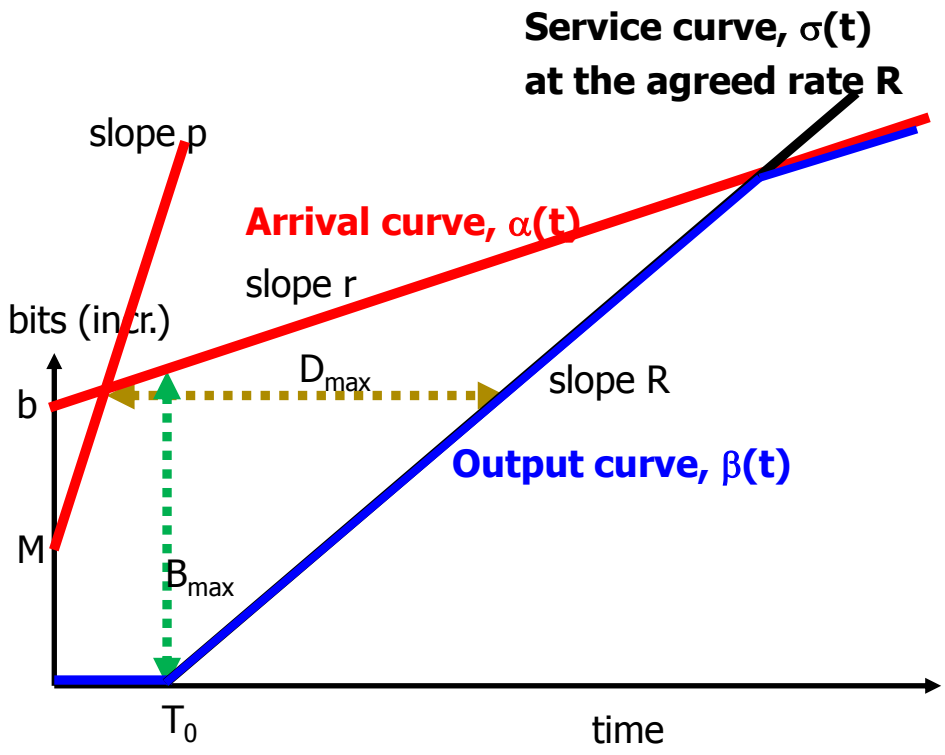
$$B_{\max} = pT_b - RT_b = b$$

This graphical approach to study delay bounds belongs to the discipline called **'network calculus' or 'deterministic queuing systems'**.

- **Given a traffic flow characterized by the token bucket model (r, b, p), each router along the path from source to destination has to allocate bandwidth R and a certain buffer B to fulfill the condition that the e2e delay is lower than a certain maximum value, Δ.**

IntServ: General Arrival-Departure Model

- The generalized model considers both M (maximum packet size) and T_0 (latency due to propagation delay):



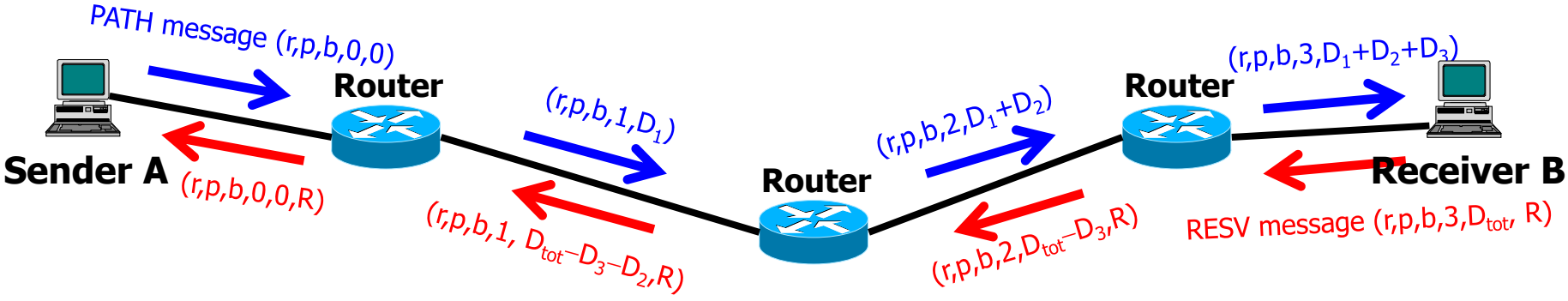
If M < b, at the beginning a packet of size M is soon delivered by the token bucket regulator.

T_0 is responsible to translate the service curve and to increase accordingly the e2e delay.

case
r < R < p

RSVP: Soft-state Receiver-Initiated, e2e Reservation

- Sender A periodically sends (**downstream**) PATH messages with T-Spec (r,p,b) to receiver B. Each router updates the **PATH message** by increasing the hop count and adding its propagation delay.
- When receiver B gets the PATH message, it knows T-Spec (r,p,b), the number of hops and the total propagation delay.
- Receiver B computes the R value and sends back (**upstream**) T-Spec and R-Spec and the propagation delay by means of the **RESV message**
- Each router **allocates bandwidth R and a certain buffer B** to the flow (per-hop delay guarantee) and propagates back the RESV message (with updated delay) to the next router that repeats the reservation process.



IntServ: Controlled Load

- **CLS (RFC 2211) does not provide any quantitative guarantee on delay bounds.**
  - With CLS, the packets of a given flow will experience delays and loss comparable to a network with no load, always assuming compliance with the traffic contract (SLA).
- **The CLS service model provides only statistical guarantees:**
  - A very high percentage of transmitted packets is successfully delivered.
  - Data packets experience small average queuing delays.
- The important difference from the traditional Internet best-effort service is that the **CLS flow does not noticeably deteriorate as the network load increases.**

IntServ: Controlled Load (cont'd)

- CLS uses T-Spec and an estimation of the mean bandwidth requested (R-Spec is not used) that are submitted to the routers along the source-destination path.
- The router has a CAC module to estimate whether the mean bandwidth requested is available for the traffic flow. In the positive case, the new flow is accepted and the related **resources are <u>implicitly</u> reserved. There is not an actual bandwidth reservation with CLS.**
- With the CLS service, there could be packet losses for the flows admitted and no delay bound guarantees.
- **CLS is intended for those applications (e.g., adaptive real-time applications) that can tolerate a certain amount of loss and delay**. CLS is not suited to those applications requiring very low latency.

Improving IntServ: Differentiated Services (DiffServ)

- There are the problems below with IntServ and RSVP; this is the reason why a new QoS approach has been proposed for IP networks and called DiffServ.
  - **Scalability**: maintaining per-flow states at the routers in high-speed networks is difficult due to the very large number of flows.
  - **Flexible service models**: IntServ has only two classes (GS and CLS); we should provide more qualitative service classes with 'relative' service differentiation (Platinum, Gold, Silver, …)
  - **Simpler signaling** (than RSVP): many applications and users may only want to specify a more qualitative notion of QoS.

DiffServ

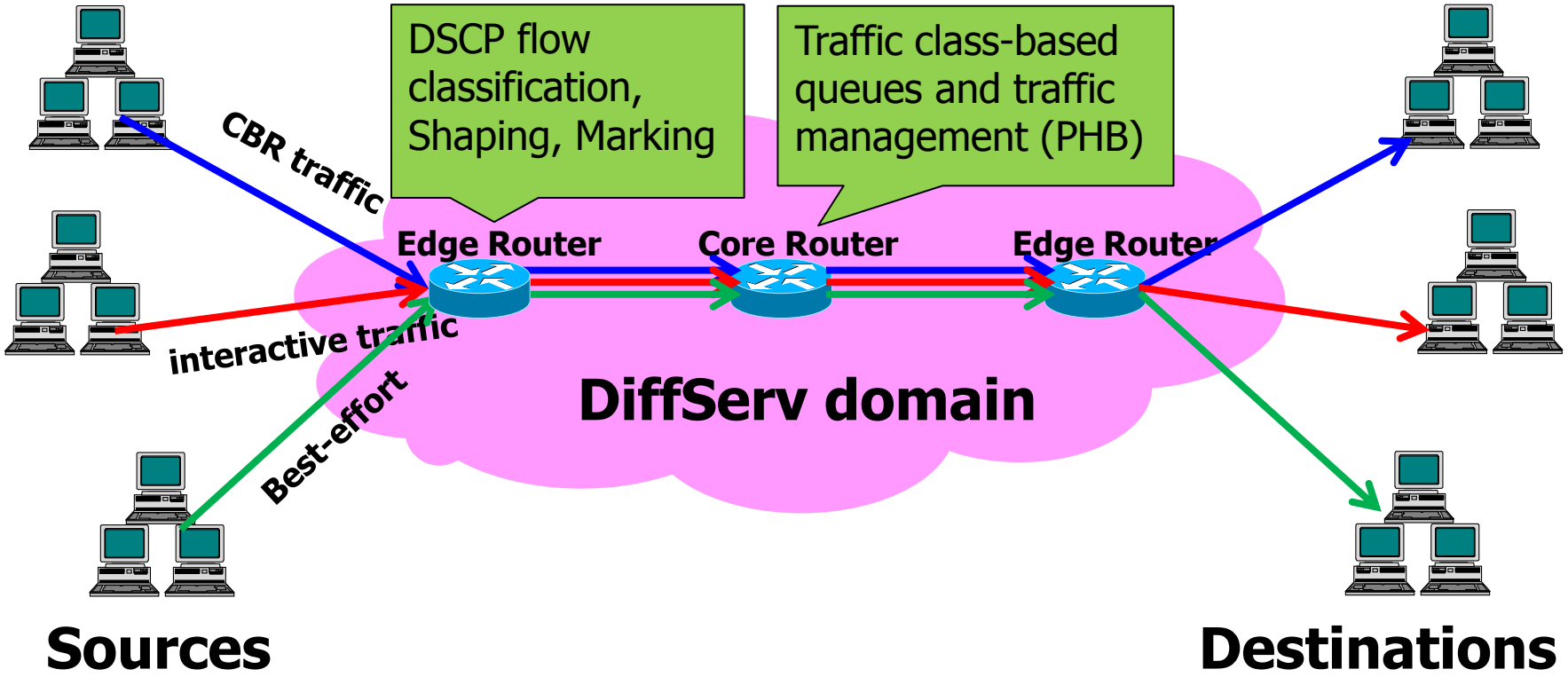
DiffServ

- To achieve scalability, the DiffServ architecture envisages treatment for **aggregated traffic flows** rather than for single flows (as IntServ). **Much of the complexity is out of the core network at edge routers**, which process lower volumes of traffic and lower numbers of flows.
- **DiffServ operates classification for the packets entering the DiffServ domain at edge routers. Instead, core router only perform packet forwarding on the basis of the classification decided at the entrance in the network.**
  - Edge routers classify each packet in a small number of aggregated flows or classes, based on the **DiffServ Code Point (DSCP)** field in the IP packet header.
  - Core routers apply Per-Hop Behavior (PHB) forwarding procedure depending on DSCP.
- No per-flow state has to be maintained at core routers, thus improving scalability.

DiffServ (cont'd)

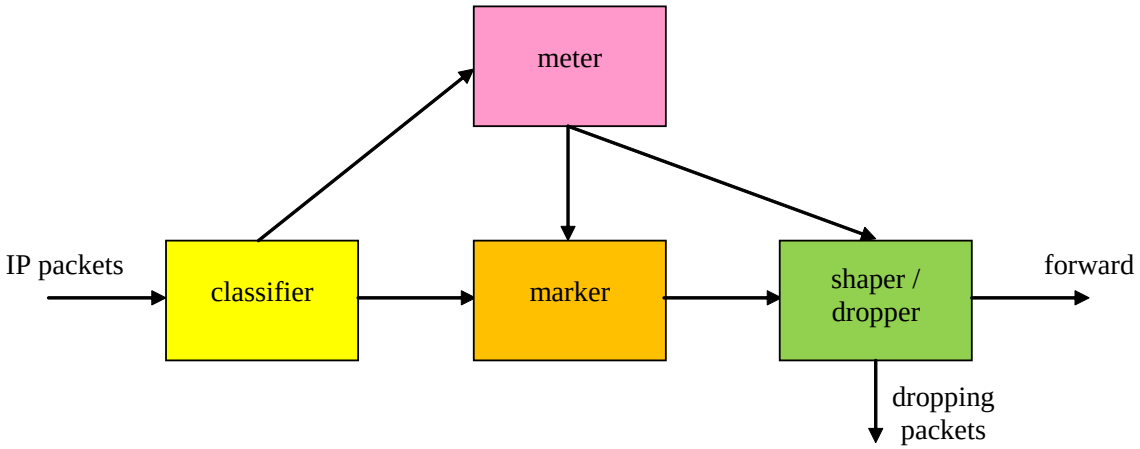
- The main DSCPs of DiffServ are:
  - **Expedited Forwarding (EF)**, RFC 3246, offering some quantitative QoS guarantees for aggregate flows.
  - **Assured Forwarding (AF)**, RFC 2597 and RFC 3260, providing some priority policies for aggregate flows.
- DiffServ traffic management mechanisms include:
  - **At edge routers of the DiffServ domain**: single flows are managed, performing classification (on the basis of the DSCP), marking, policing, and shaping functions.
  - **At core routers within a DiffServ domain**: traffic flows are managed as aggregated flows according to the traffic classes determined by edge routers (PHB). Forwarding and scheduling is based on PHBs.

DiffServ Architecture



DiffServ: Edge Router/Host Functions

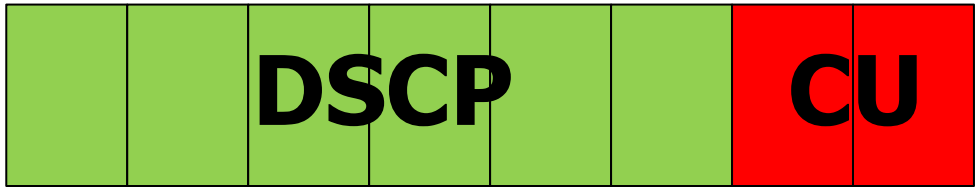
- **Classifier**: It classifies the packets on the basis of different elements (DSCP).
- **Meter:** It checks whether the traffic falls within the negotiated profile (policer).
- **Marker**: It writes/rewrites the DSCP value in the packet header.
- **Shaper/dropper**: It delays some packets and then forwards or discards exceeding packets.



Traffic Conditioner Block (TCB) at edge routers

DiffServ: Classification

- An IP packet is marked in the Type of Service (ToS) byte in the IPv4 header or in the Traffic Class (TC) field in the IPv6 header.
- 6 bits are used for DSCP and determine the PHB that the packet will receive.
- 2 bits are Currently Unused (CU). They can be used for Explicit Congestion Notification (ECN).



ToS byte in IPv4 header or TC byte in IPv6 header

Expedited Forwarding PHB

- **Expedited Forwarding (EF)** - RFC 3246:
  - The EF traffic class is for guaranteed bandwidth, low jitter, low delay, and low packet losses for **aggregate flows.**
  - The EF traffic is supported by **a specific queue at the routers**. The EF traffic is not influenced by the other traffic classes (AF and BE).
  - Non-conformant EF traffic is dropped or shaped.
  - **EF traffic is often strictly controlled by CAC (admission based on peak rate), policing, and other mechanisms**.
  - The recommended DSCP for EF is 101110.

Assured Forwarding PHB

- **Assured Forwarding (AF)** - RFC 2597 and RFC 3260:
  - AF is not a single traffic class, but **4 sub-classes**: AF1, AF2, AF3, and AF4. Hence, we can expect to have **4 AF queues** at the routers. The **service priority** for these queues at the routers is: AF1 > AF2 > AF3 > AF4.
  - **Within each sub-class (i.e., within each queue), there are three drop precedence** values from a low drop level 1 up to a high drop level 3 (with related DSCP coding) to determine which packets will be dropped first in each AF queue if congested: the drop precedence order for the generic queue AFx, $x \in \{1, 2, 3, 4\}$, is AFx3 before AFx2 before AFx1. The packets of a generic AFx class queue are sent in FIFO order.
- Considering EF, AF and BE, **6 IP-layer queues** are needed at the router to support DiffServ.

Assured Forwarding PHB (cont'd)

- AF is used to implement services that differ relatively to each other (e.g., gold, silver, etc.).
- Non-conformant traffic is remarked, but not dropped.
- AF is suitable for services that require a minimum guaranteed bandwidth (additional bandwidth can only be used if available) with possible packet dropping above the agreed data rate in case of congestion.

Priority reduces from top to bottom and from left to right.

	Class 1	Class 2	Class 3	Class 4
Low Drop	AF11 (DSCP 10)	AF21 (DSCP 18)	AF31 (DSCP 26)	AF41 (DSCP 34)
Medium Drop	AF12 (DSCP 12)	AF22 (DSCP 20)	AF32 (DSCP 28)	AF42 (DSCP 36)
High Drop	AF13 (DSCP 14)	AF23 (DSCP 22)	AF33 (DSCP 30)	AF43 (DSCP 38)

Traffic Management and Scheduling at Nodes (DiffServ)

- **Scheduling**: Rather than using strict priority queuing, more balanced **scheduling** algorithms such as fair queuing or weighted fair queuing are used.

- **Buffer Management**: To prevent **problems associated with tail drop events** (i.e., arriving packets are dropped when queue is congested, regardless of flow type or importance), RED or WRED algorithms can be used to drop packets.
  - If congestion occurs, the traffic in the higher class (e.g., class 1) has priority and the packets with the higher drop precedence are discarded first.

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)

Thank you!

giovanni.giambene@gmail.com