Performance Issues

Xuetao Wei
weixt@sustech.edu.cn

Contents

Performance Metrics

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

quiz重点

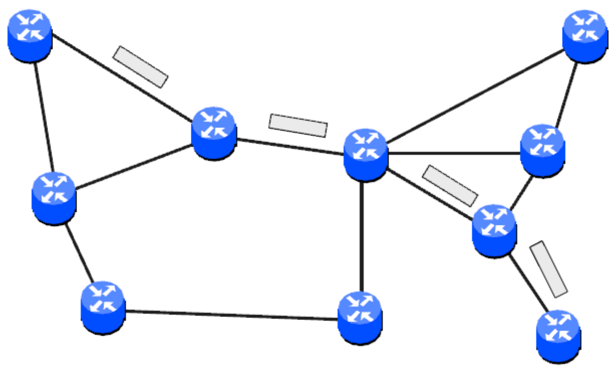
- **Bandwidth**
  - Maximum bits/second that can be transmitted
- **Throughput**
  - Actual rate of data transmission
- **Latency**
- **Jitter**
- **Error rate**

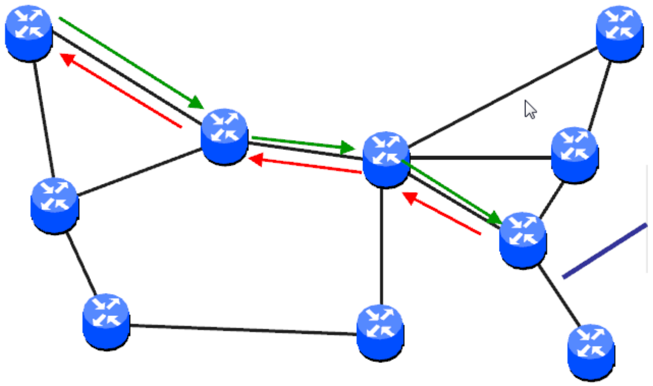
Performance Metrics

- **Bandwidth**
- **Throughput**
- **Latency**
  - Delay between send and receiver (data travel time + processing time at network nodes)
- **Jitter**
  - Variation in packet delay at the receiver
- **Error rate**
  - The number of corrupted bits compared to the total data sent

Quality of Service (QoS)

- The **Holy Grail** of computer networking is to design a network that
  - has the flexibility and low cost of the Internet
  - yet offers the end-to-end quality-of-service guarantee of the telephone network





Service Models

- Networks provide a **service model** to applications

- **Best effort**
  - The network does not provide any performance guarantees for packet delivery
  - All users/flows are treated without differentiation

- **Quality of Service (QoS)**
  - The network provies a certain level of performance guarantee to a data flow
  - Different priorities for different applications, users, or data flows

Two Styles of QoS

- **Average-case**
  - Provide bandwidth/delay/jitter guarantee over many packets
  - Statistical in nature (soft real-time)
  - Multimedia applications (VoIP, Video, Video Conferencing, Online gaming, ...)

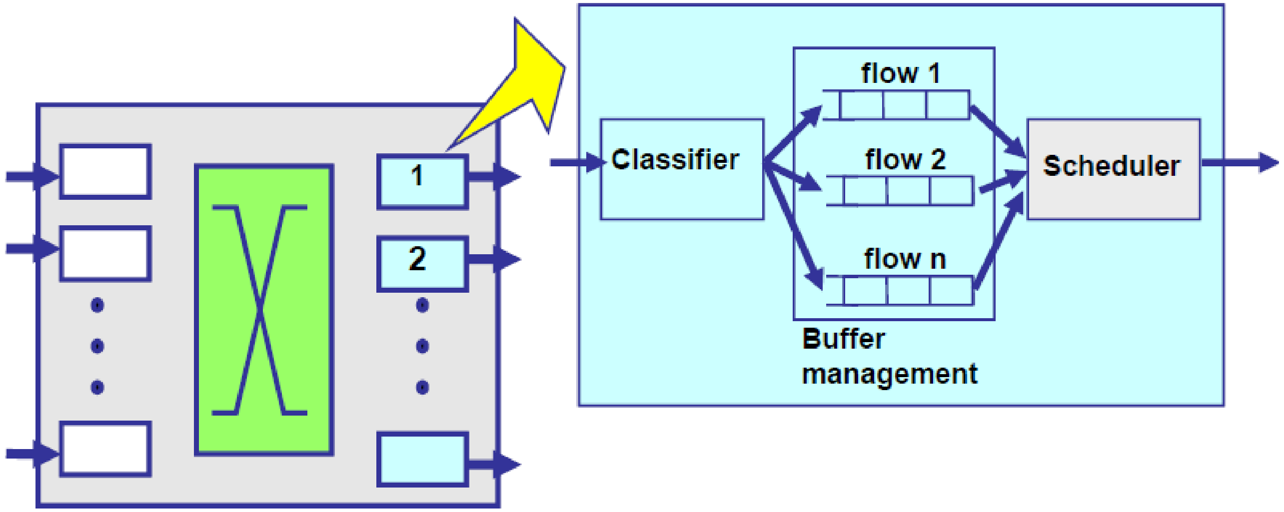
知识点

- **Worst-case**
  - Provide bandwidth/delay/jitter guarantee to every packet
  - Hard real-time
  - Control applications

QoS Enforcement

- **Buffer management**
  - Which packet to drop when buffer/queue is full?

- **Packet scheduling**
  - Which packet to transmit next?



Default Buffer Mgmt/Scheduling

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

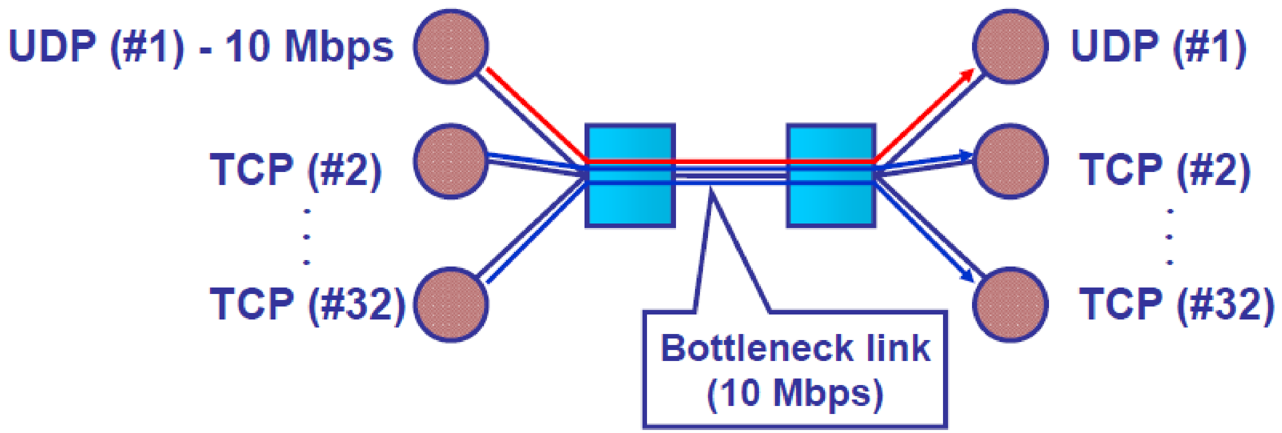
- **FIFO + Drop-tail**
  - Simplest choice and widely used in Internet
- **Functionality**
  - FIFO: scheduling discipline
  - Drop-tail: drop policy
- **FIFO scheduling (first-in-first-out)**
  - Implies single class of traffic
- **Drop-tail buffer management**
  - Arriving packets get dropped when queue is full regardless of flow or importance

FIFO/Drop-Tail Problems

- Leaves responsibility of congestion control completely to hosts (e.g., TCP)
- Does not separate between different flows
- No policing: send more packets => get more service

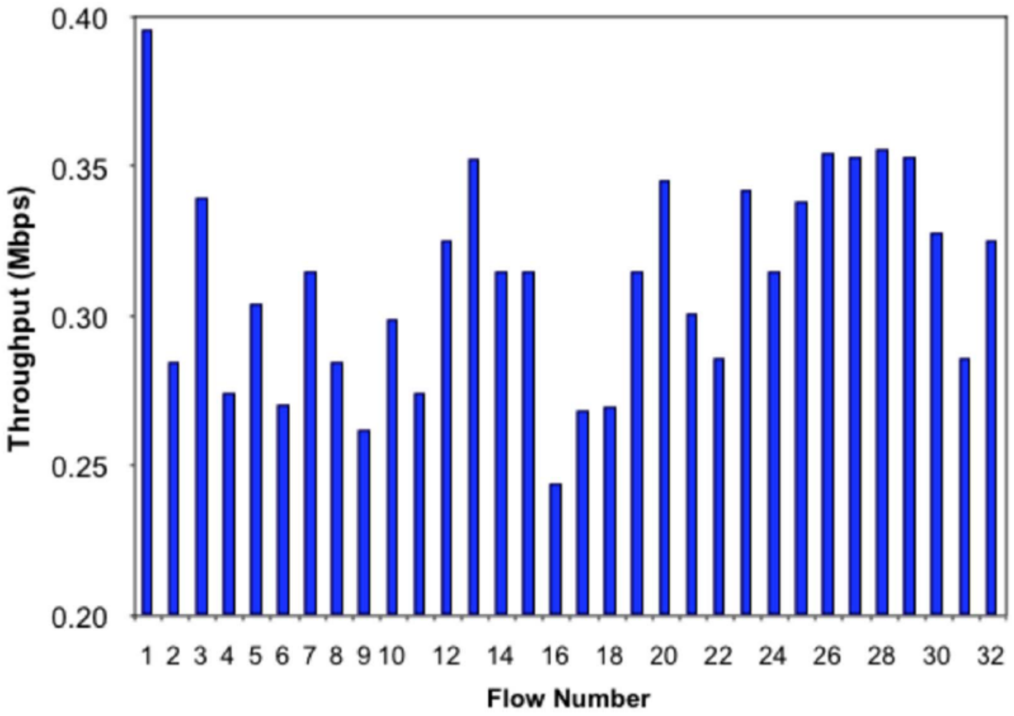
Non-responsive Senders

- 1 UDP (10 Mbps) and 31 TCPs sharing a 10 Mbps line

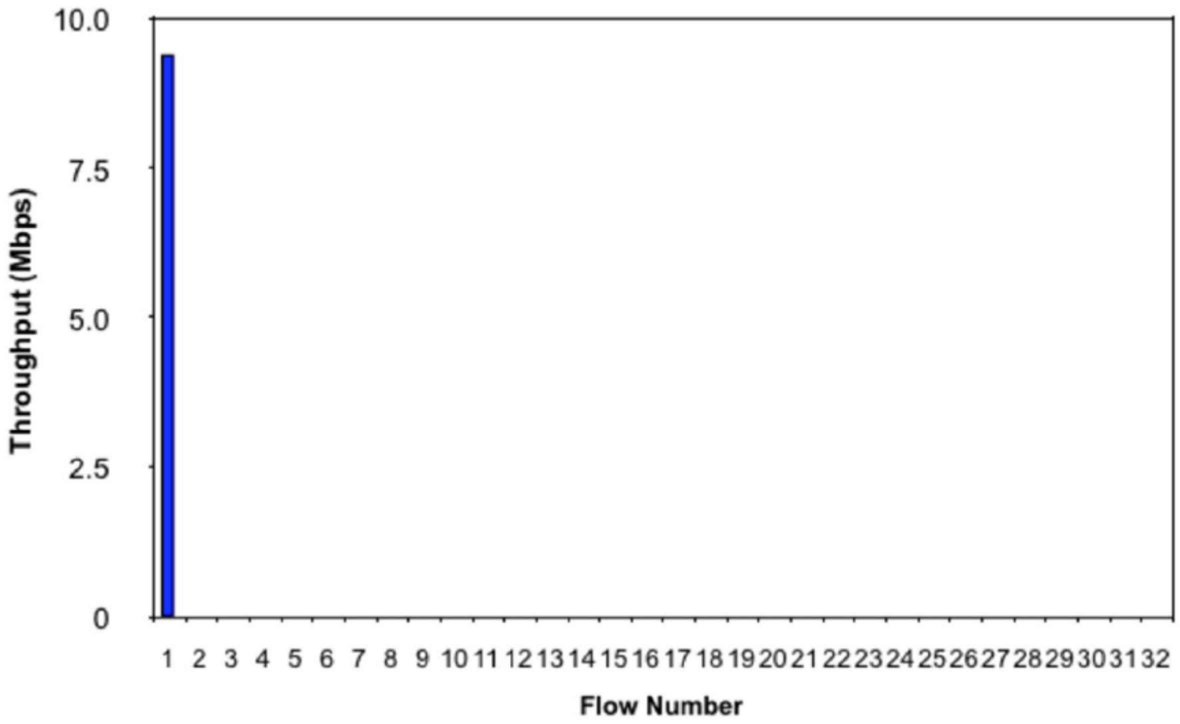


UDP vs. TCP

desired



reality



Service Level Agreement (SLA)

记住这个名字 重要

- SLA: a service contract between a customer and a service provider that specifies the forwarding service a customer should receive
  - The type and nature of service to be provided
  - The expected performance level of the service, which includes two major aspects: reliability and responsiveness
  - Traffic Conditioning Agreement (TCA) which defines the rules used to realize the service

  - Classifier rules: metering, marking, discarding and/or shaping rules which are to apply to the traffic streams selected by the classifier

Contents

- Performance Metrics & QoS
- **Buffer Managment**
- Packet Scheduling
- Network-wide QoS

Traffic Policing & Shaping

重点

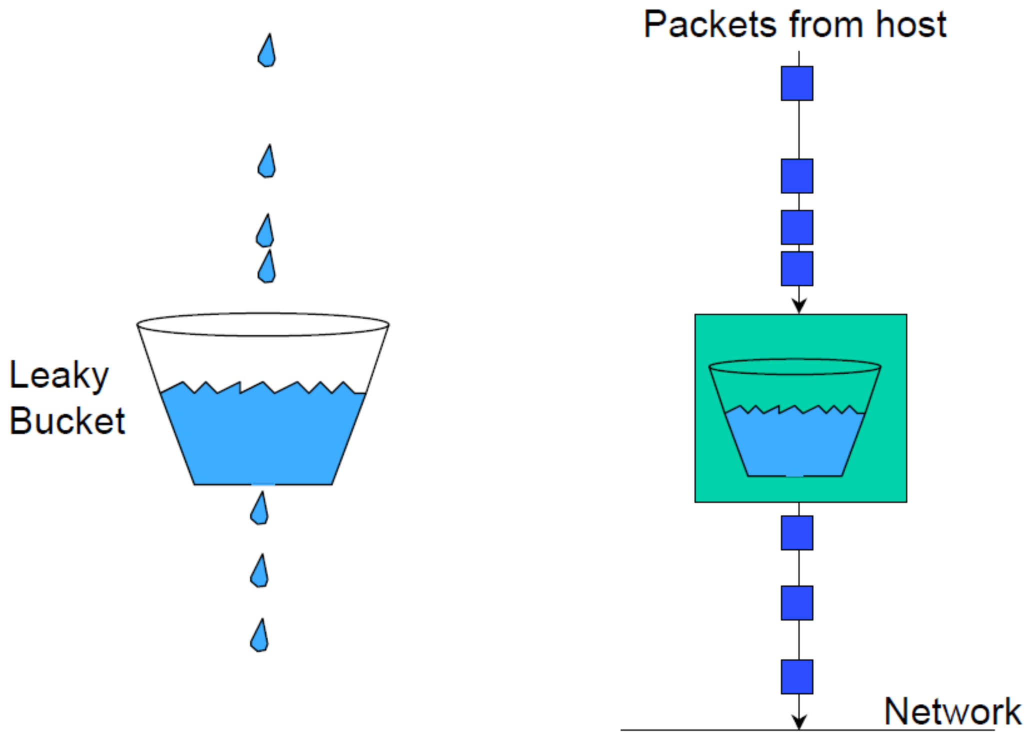
- **Traffic policing**
  - Monitor traffic flows and take corrective actions (mark or drop) when the observed characteristics deviate from a specific SLA.

- **Traffic Shaping**
  - Control burst data or traffic rate to send data streams in order to ensure conformance to a specific SLA.
  - By delaying non-conforming traffic until that conforms to the profile.

- Policer **drops** packets, shaper **delays** packets.

Leaky Bucket

- Packets may be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced



What is the problem?

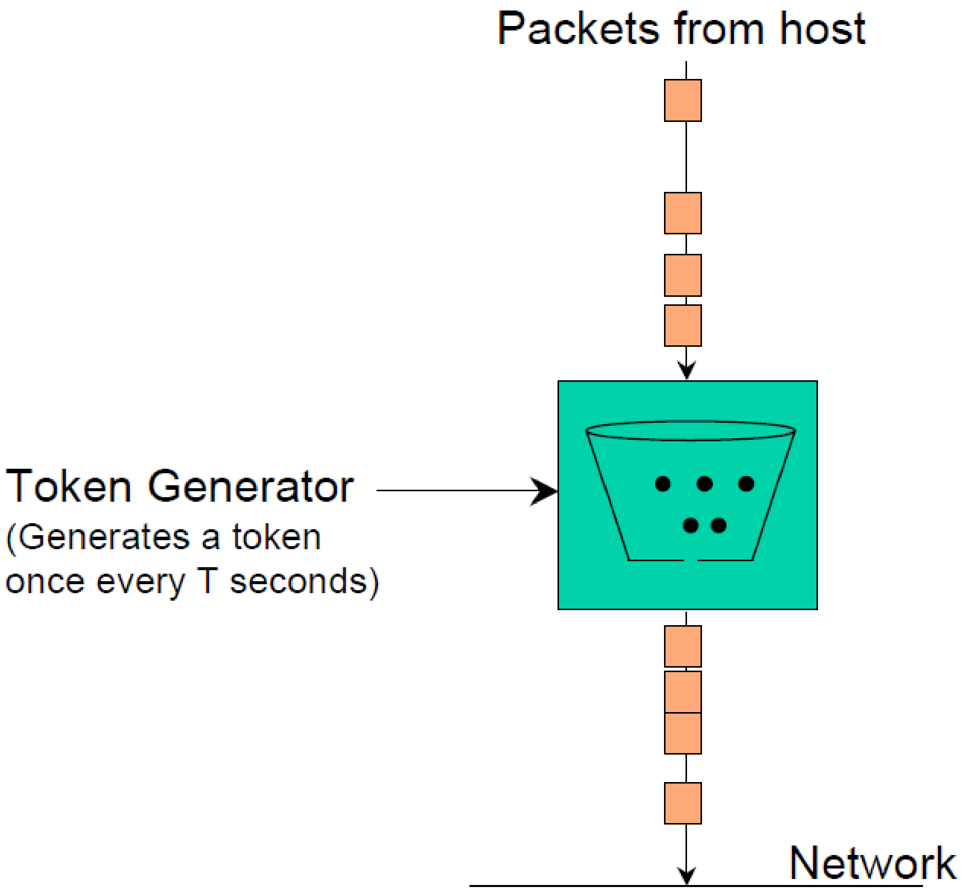
Leaky Bucket

- The leaky bucket is a “traffic shaper”: It changes the characteristics of packet stream
- Traffic shaping makes more manageable and more predictable packet streams
- Some times, we may allow short bursts of packets to enter the network without smoothing them out, but leaky bucket doesn't support bursty traffic.
- Solution: **Token bucket**!

Token Bucket

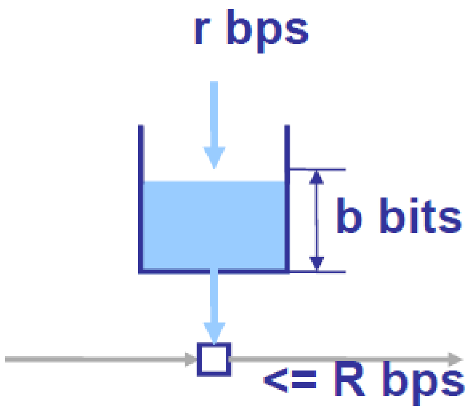
- The bucket holds tokens instead of packets
- Tokens are generated & placed into the token bucket at a constant rate
- When a packet arrives at the token bucket, it is transmitted if there is a token available. Otherwise it is buffered until a token becomes available
- The token bucket has a fixed size, so when it becomes full, subsequently generated tokens are discarded

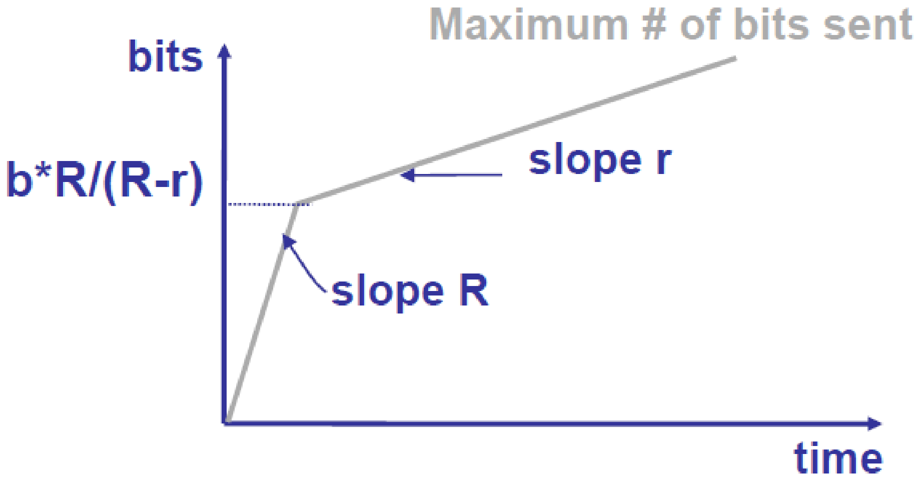
Token Bucket



How Token Bucket Works

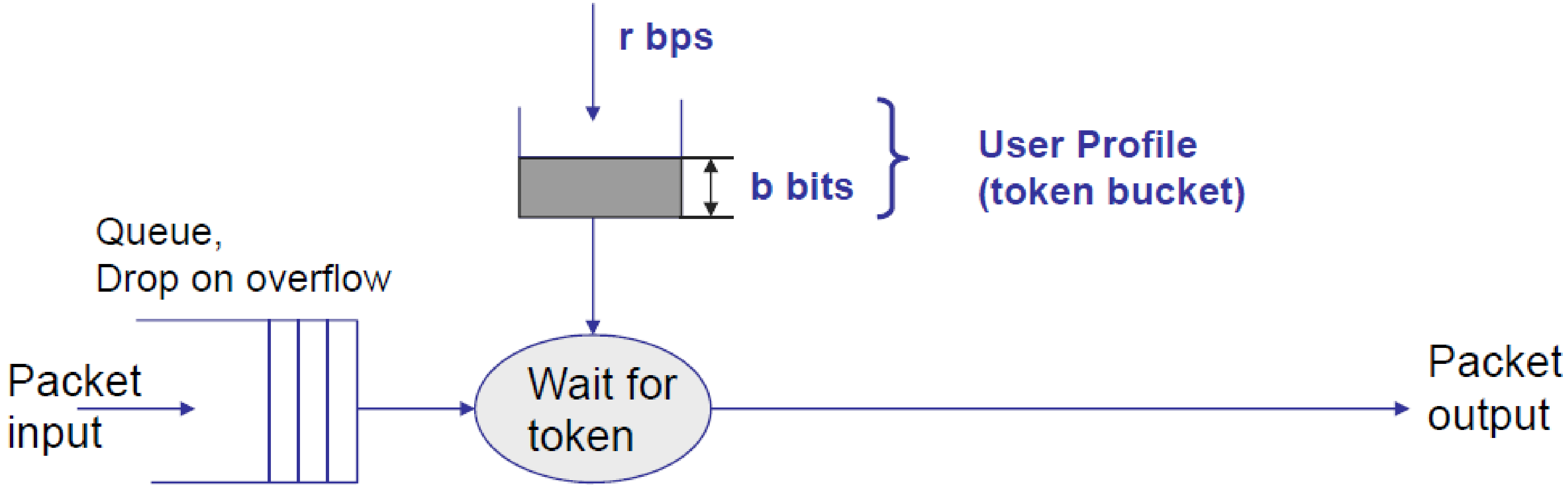
- Parameters
  - r – average rate, i.e., rate at which tokens fill the bucket
  - b – bucket depth
  - R – maximum link capacity or peak rate (optional parameter)
- A bit is transmitted only when there is an available token





Traffic Shaping

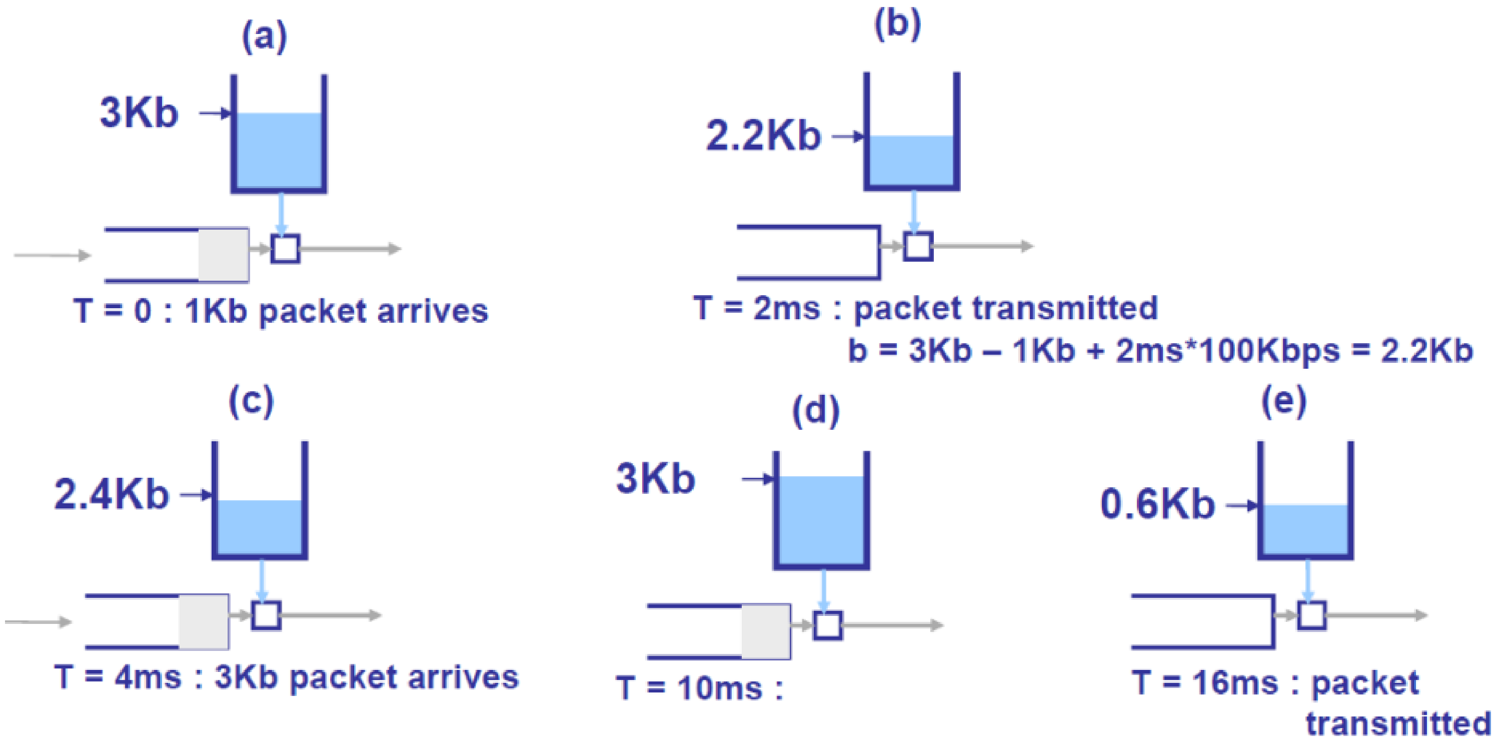
- Shape packets according to user profile or SLA
- Output limited to average of r bps and bursts of b



Shaping Example

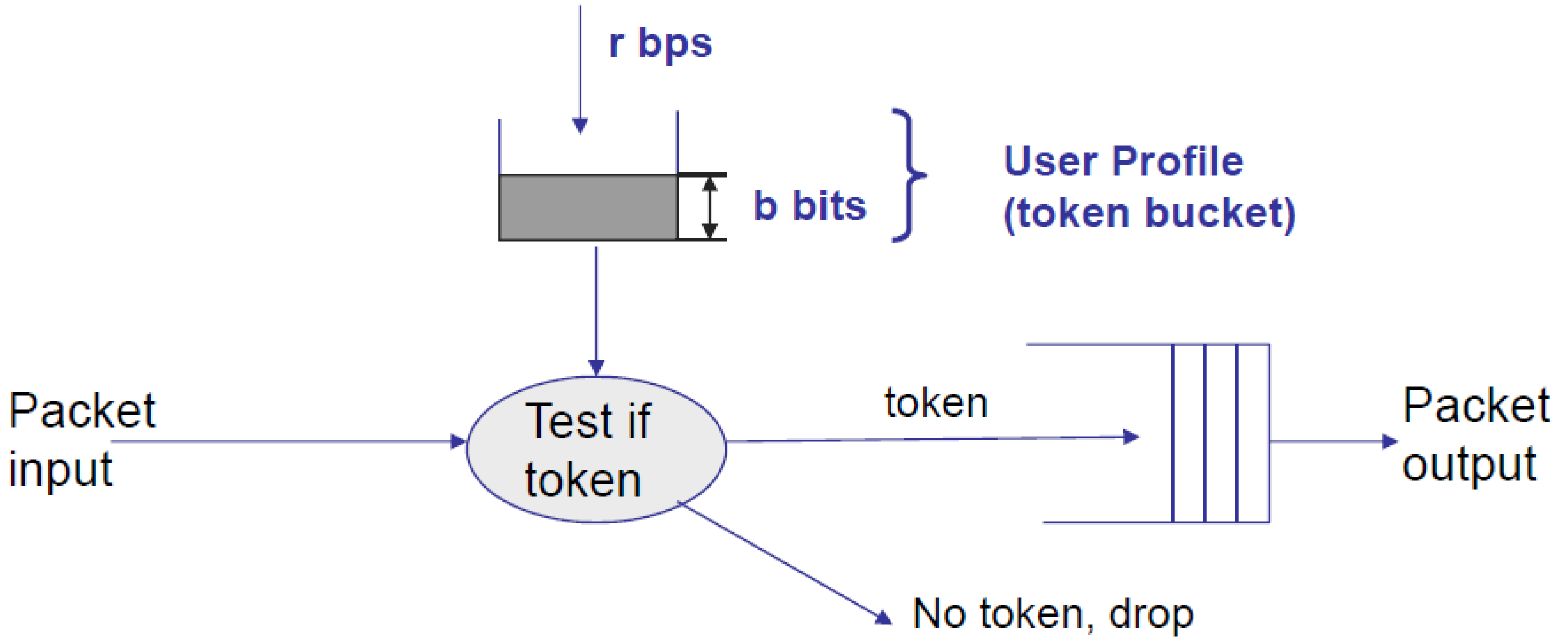
记住这个例子

- r = 100 Kbps; b = 3 Kb; R = 500 Kbps



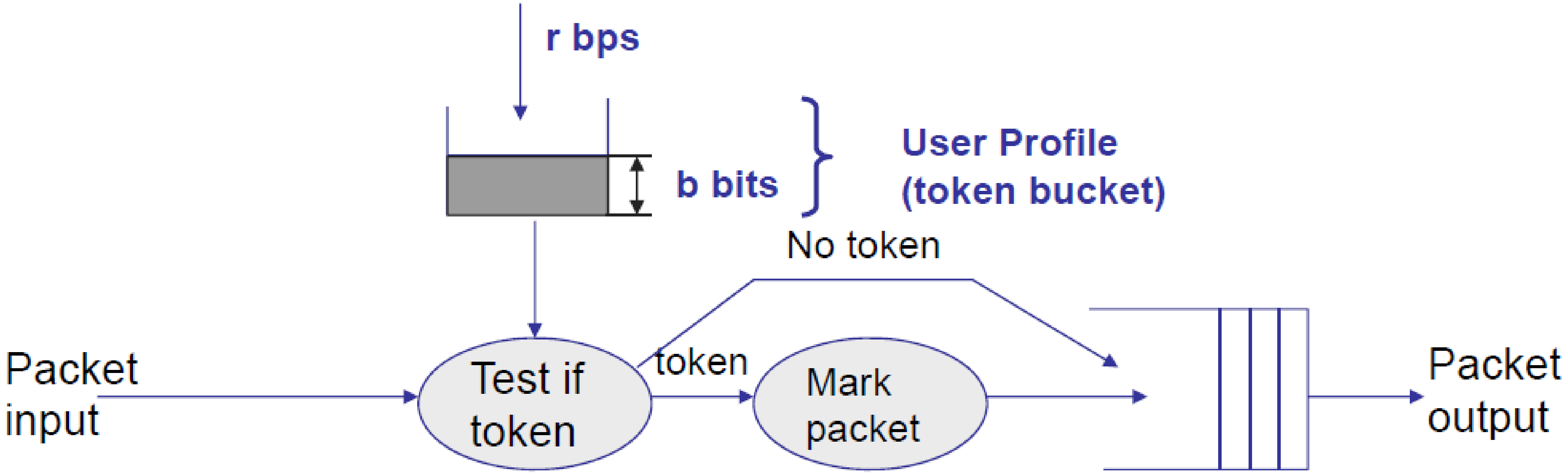
Traffic Policing

- Drop packets that don't meet user profile
- Output limited to average of r bps and bursts of b



Buffer Management

- Mark packets according to the flow's token bucket profile
- During congestion, drop unmarked packets first

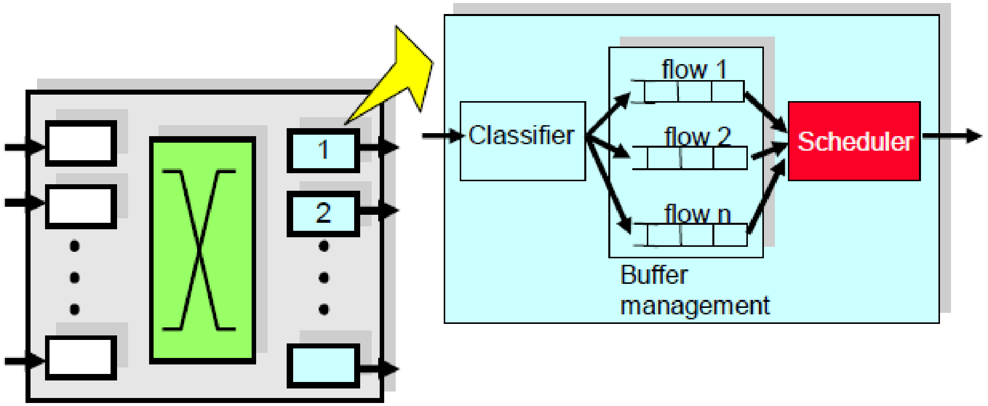


Contents

Scheduling

- **Flow-based traffic policing/shaping**
  - Limit the rate of one flow regardless the load in the network

- **Scheduling: considering multiple flows**
  - Give each “flow” (or traffic class) own queue (logically or physically)
  - Decide when and what packet to send on output link
  - Dynamically allocate resources when multiple flows compete
  - Usually implemented at output interface of a router



Fair Queuing

- Round robin (RR) among queues
- Proportional sharing scheduling (max-min fairness)
- Weighted Fair Queueing (WFT): schedules in proportion to some weight parameter

Round Robin (RR)

- Each flow or service class has its own queue, arriving packets are stored in different queues based on their flow IDs or service classes
- The server polls each queue in a cyclic order and serves a packet from any nonempty queue encountered
- A misbehaving user may overflow its own queue but will not affect others
- The RR scheduler attempts to treat all users equally and provide each of them an equal share of the link capacity
- Problems
  - Not fair when packet sizes are not equal
  - Impractical when the number of queues is large

Max-min Fairness

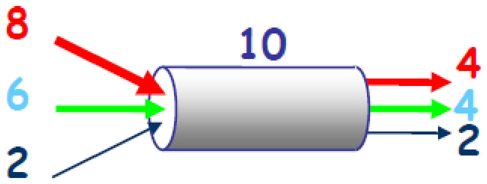
南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Idea: Allocate a user with "small" demand what it wants, evenly divide unused resources to "big" users
- Formally
  - Resources allocated in order of the increasing demand
  - No source gets a resource share larger than its demand
  - Sources with unsatisfied demands get an equal share of resource

Max-min Fairness

- Each flow receives $\min(r_i, f)$, where
  - r_i – flow arrival rate
  - f – link fair rate

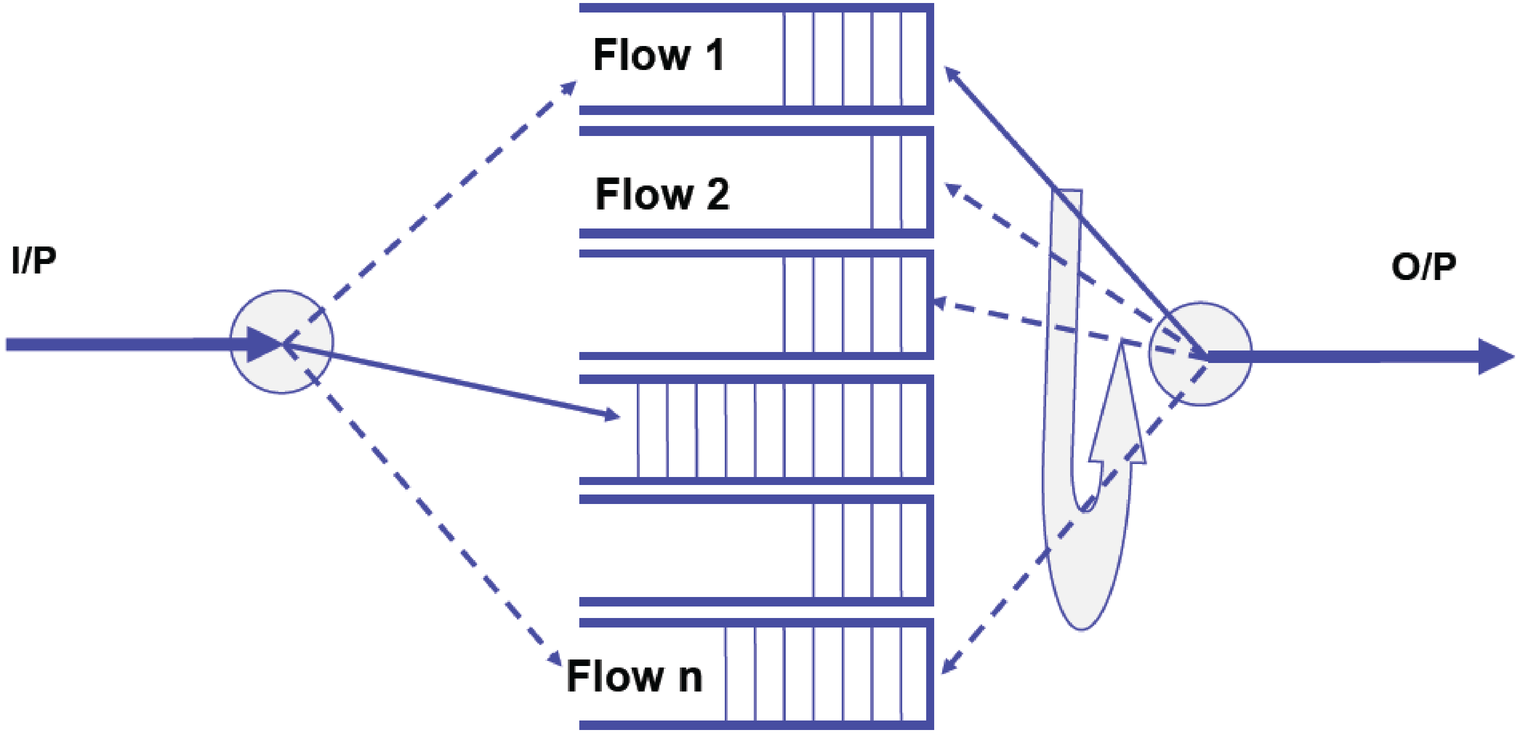
$$\sum_i \min(r_i, f) = C$$



$f = 4$:
min(8, 4) = 4
min(6, 4) = 4
min(2, 4) = 2

Weighted Fair Queueing (WFQ)

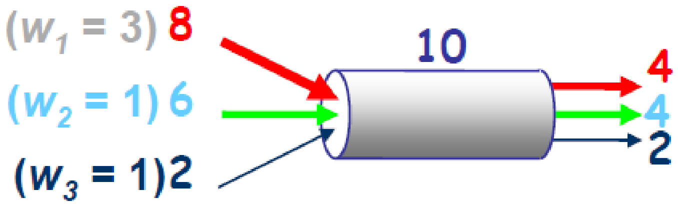
- Associate a weight with each flow



Weighted Fair Queueing (WFQ)

- Associate a weight w_i with each flow f_i,
- If the link congests, compute f such that

$$\sum_i \min(r_i, f \times w_i) = C$$



f = 2:
min(8, 2*3) = 6
min(6, 2*1) = 2
min(2, 2*1) = 2

Flow *i* is guaranteed to be allocated a rate >= w_i*C/(Σ_k w_k)

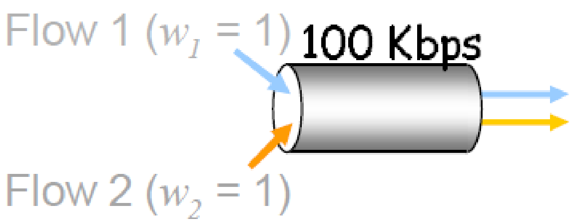
⇓

If Σ_k w_k <= *C*, flow *i* is guaranteed to be allocated a rate >= w_i

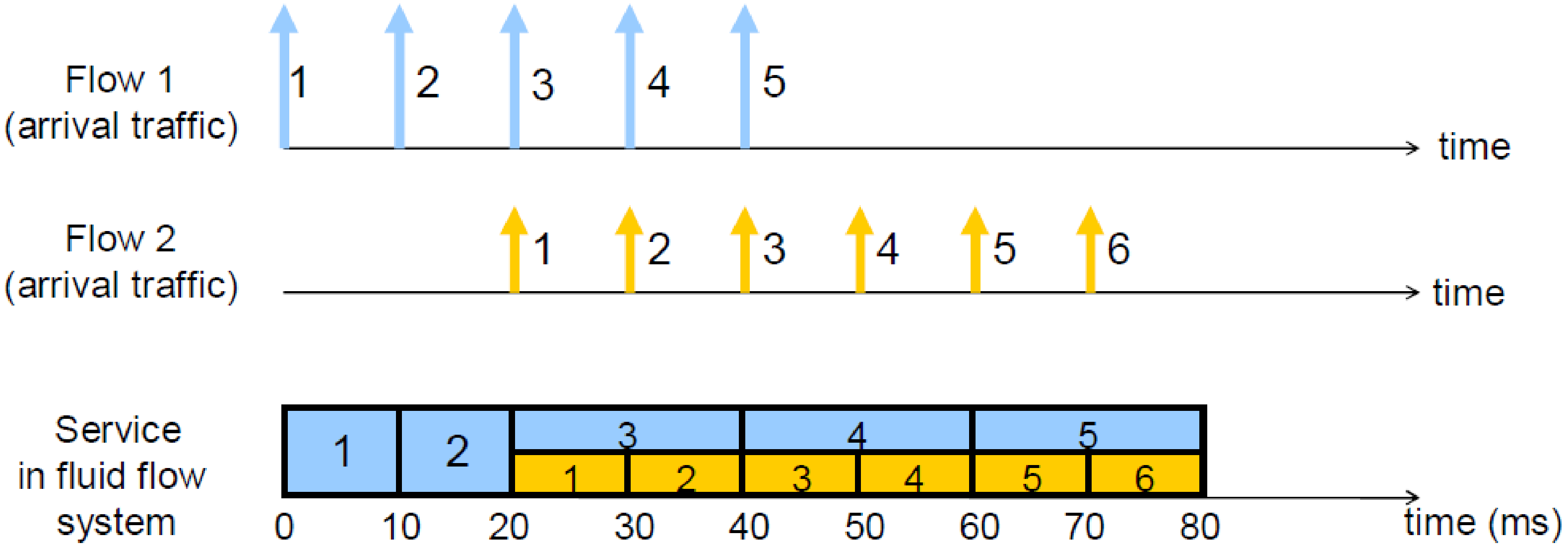
Fluid Flow Model

- How to implement WFQ?
- Flows can be served one bit at a time
- WFQ can be implemented using bit-by-bit weighted round robin
  - During each round from each flow that has data to send, send a number of bits equal to the flow's weight

Fluid Flow System: Example 1

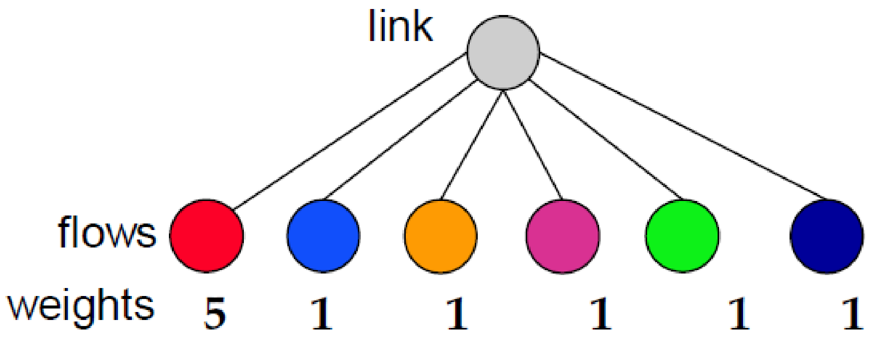


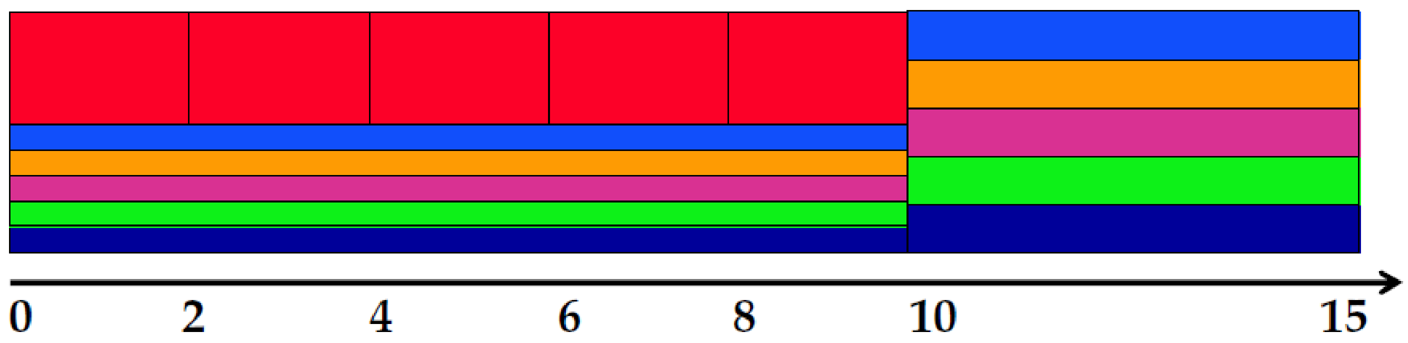
	Packet Size (bits)	Packet inter-arrival time (ms)	Rate (Kbps)
Flow 1	1000	10	100
Flow 2	500	10	50



Fluid Flow System: Example 2

- The red flow has packets backlogged between time 0 and 10
  - Backlogged flow => flow's queue not empty
- Other flows have packets continuously backlogged
- All packets have the same size



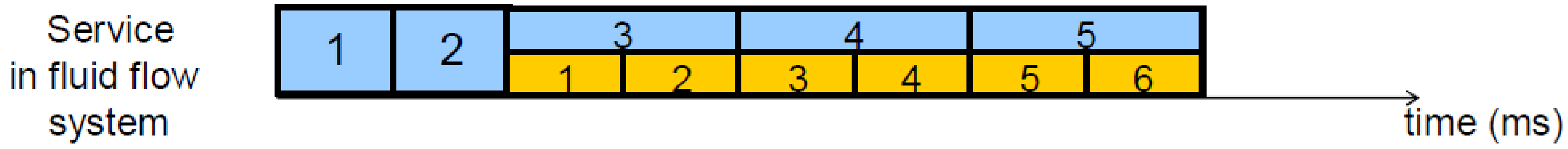


Implementation in Packet System

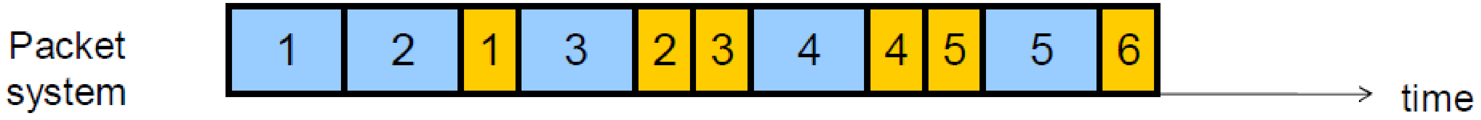
- Packet (Real) system: the packet transmission cannot be preempted. Why?

- Solution: serve packets in the order in which they would have finished being transmitted in the fluid flow system

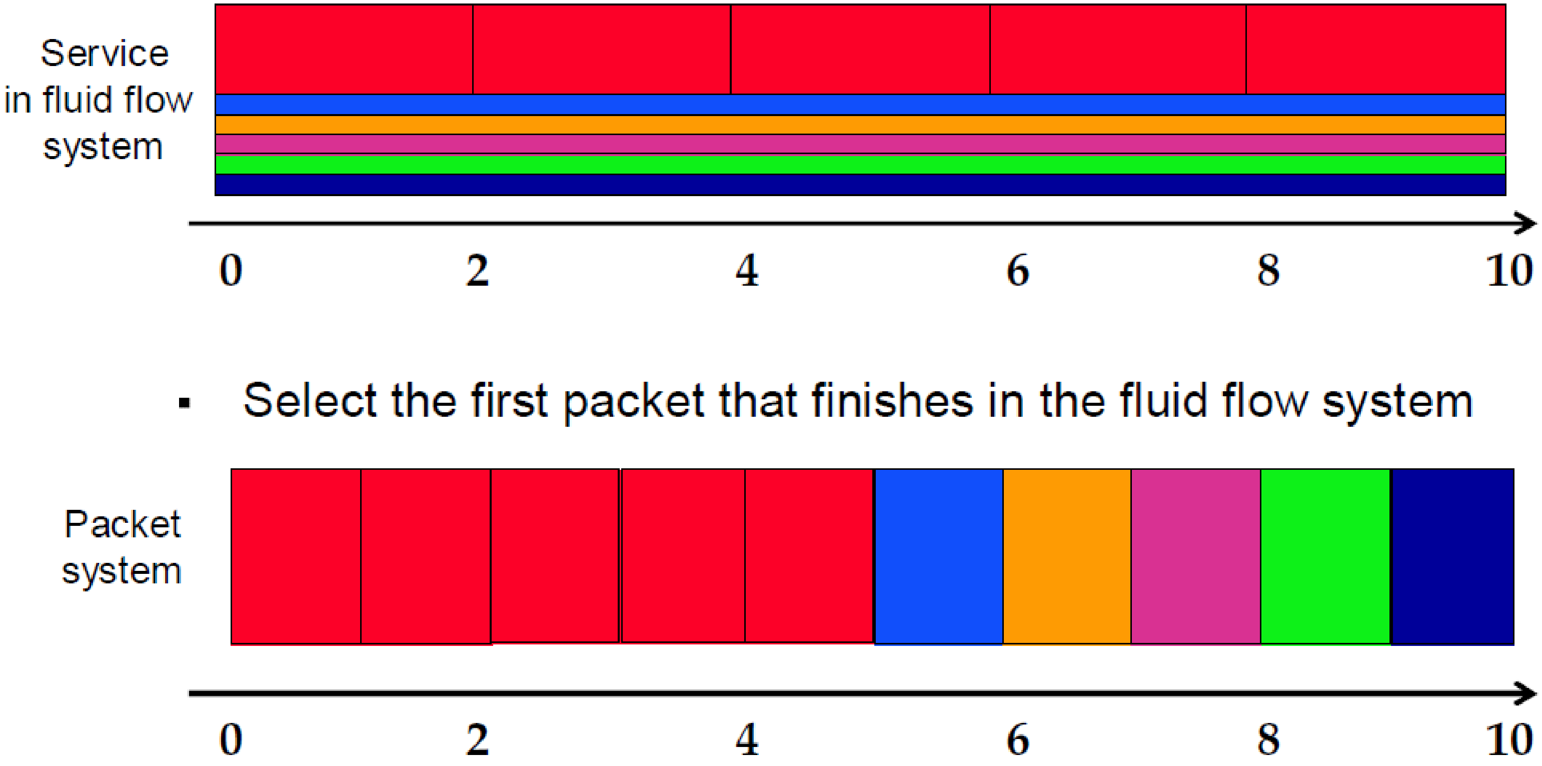
Packet System: Example 1



- Select the first packet that finishes in the fluid flow system



Packet System: Example 2



- Select the first packet that finishes in the fluid flow system

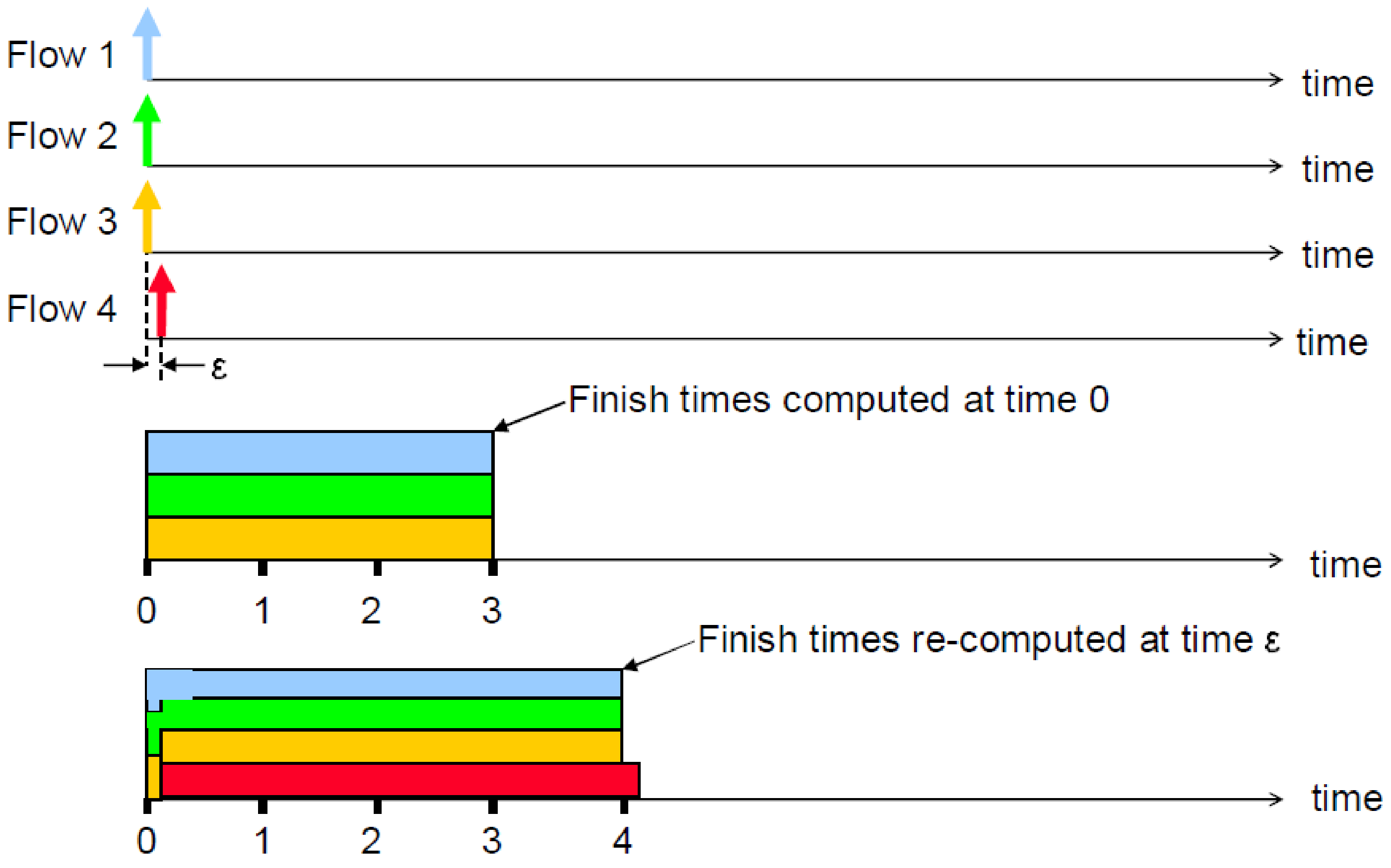
Implementation Challenge

- Need to compute the finish time of a packet in the fluid flow system…
  - … but the finish time may change as new packets arrive!

- Need to update the finish times of all packets that are in service in the fluid flow system when a new packet arrives
  - But this is very expensive; a high speed router may need to handle hundred of thousands of flows!

Example

- Four flows, each with weight 1

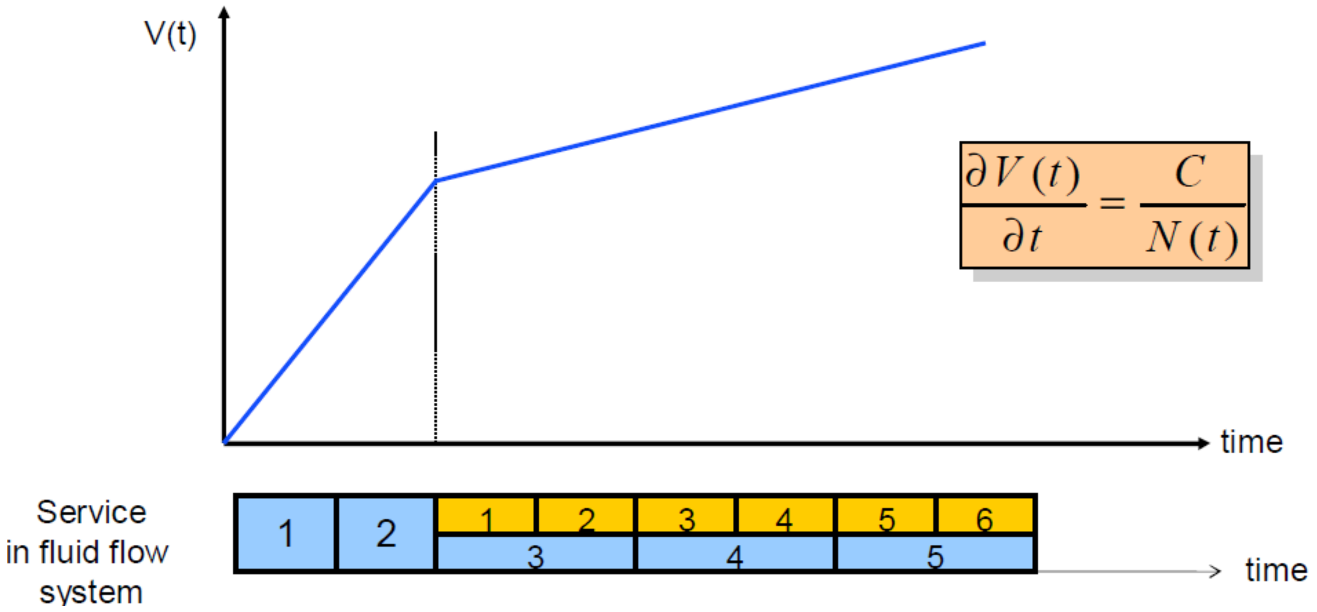


Solution: Virtual Time

- **Key Observation**: while the finish times of packets may change when a new packet arrives, the order in which packets finish doesn't!
  - Only the order is important for scheduling
- **Solution**: instead of the packet finish time, maintain the number of rounds needed to send the remaining bits of the packet (virtual finishing time)
  - Virtual finishing time doesn't change when the packet arrives
- **System virtual time:** index of the round in the bit-by-bit round robin scheme

System Virtual Time: V(t)

- Measure service, instead of time
- V(t) slope: normalized rate backlogged flow receiving service
  - C: link capacity
  - N(t): total weight of backlogged flows in fluid flow system at time t

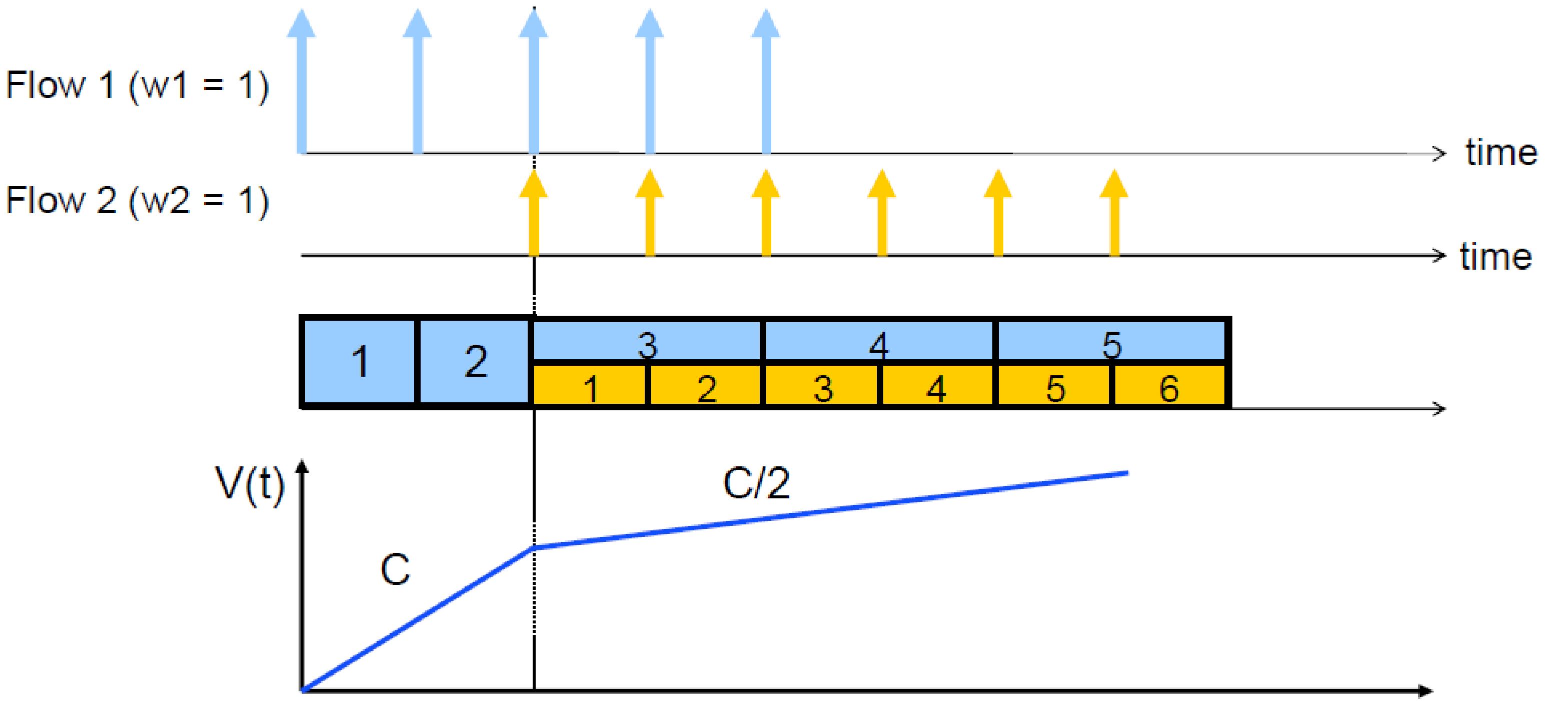


$$\frac{\partial V(t)}{\partial t} = \frac{C}{N(t)}$$

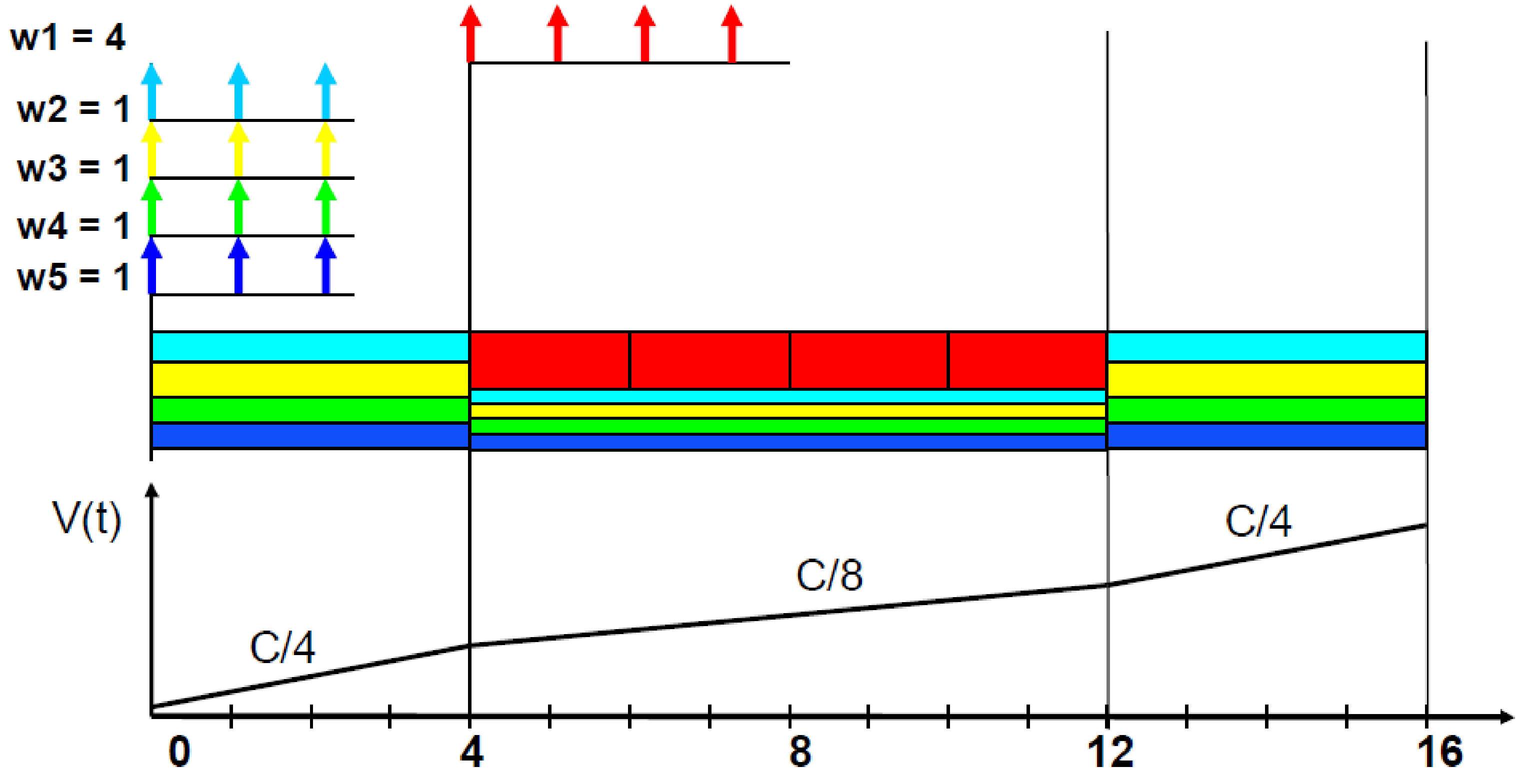
System Virtual Time V(t): Example 1

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- V(t) increases inversely proportionally to the sum of the weights of the backlogged flows



System Virtual Time V(t): Example 2



Fair Queuing Implementation

Define

- F_i^k - virtual finishing time of packet k of flow i
- a_i^k - arrival time of packet k of flow i
- L_i^k - length of packet k of flow i
- w_i – weight of flow i

The finishing time of packet $k+1$ of flow i is

$$F_i^{k+1} = \max(V(a_i^{k+1}), F_i^k) + L_i^{k+1}/w_i$$

Contents

- Performance Metrics & QoS
- Buffer Managment
- Packet Scheduling
- **Network-wide QoS**

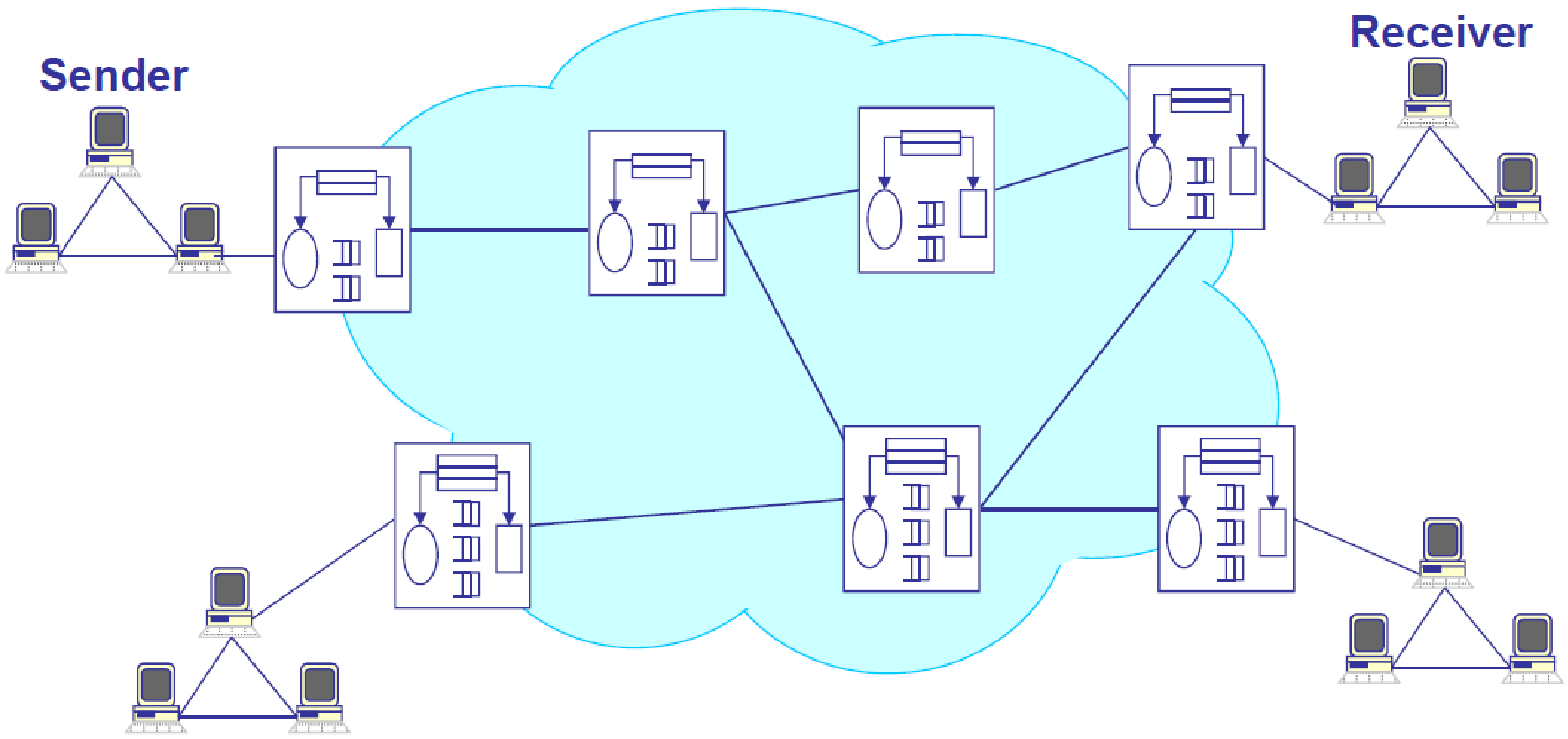
Network-wide QoS

- **Integrated services**
  - Motivated by the need for end-to-end guarantees
  - On-line negotiation of per-flow requirements
  - End-to-end per-router negotiation of resources
  - Complex

- **Differentiated services**
  - Motivated by economics (multi-tier pricing)
  - No per-flow state
  - Not end-to-end and not guaranteed services
  - Simple

Integrated Services

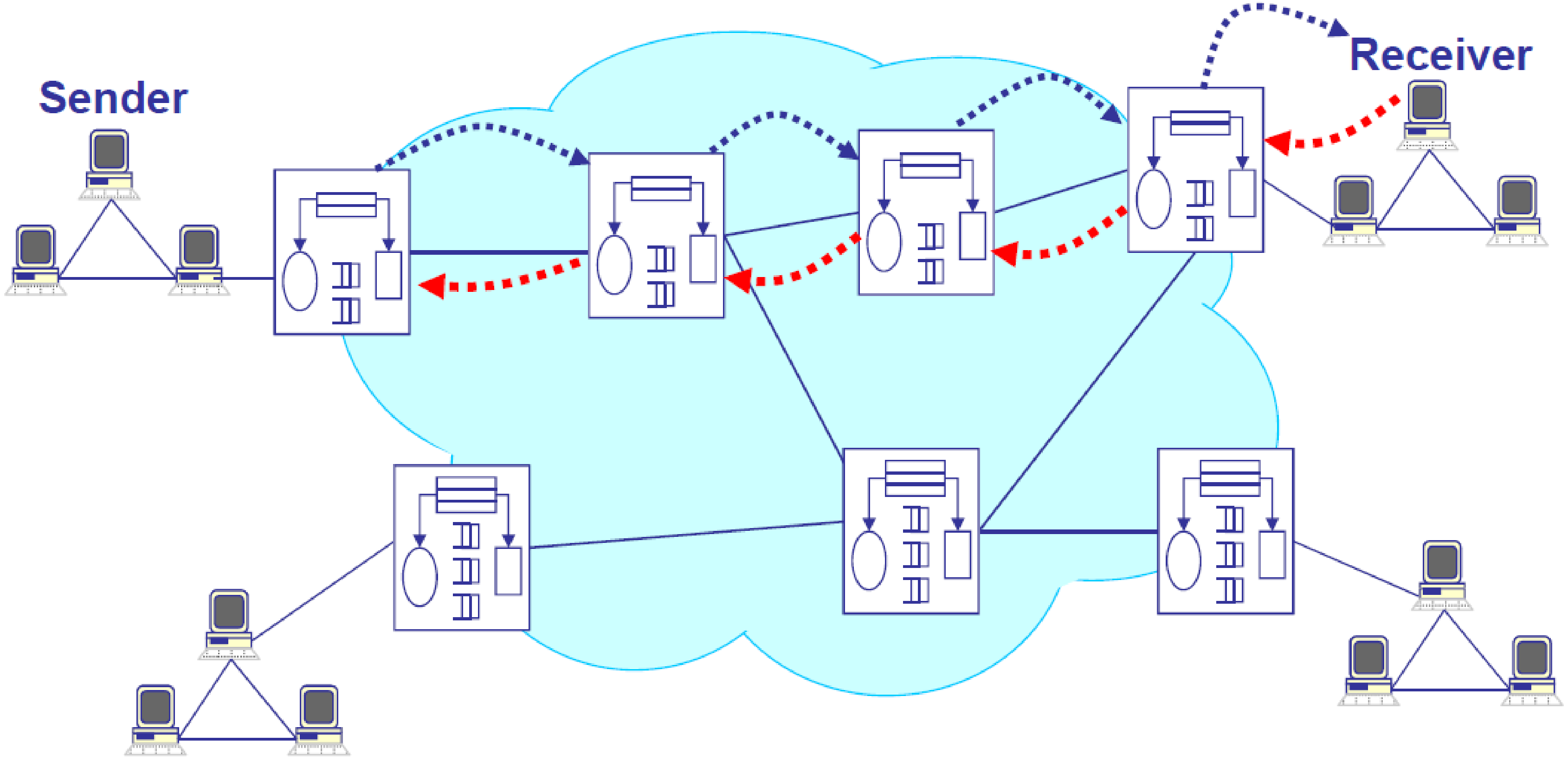
- Example: guarantee 1MBps and < 100 ms delay to a flow



Integrated Services

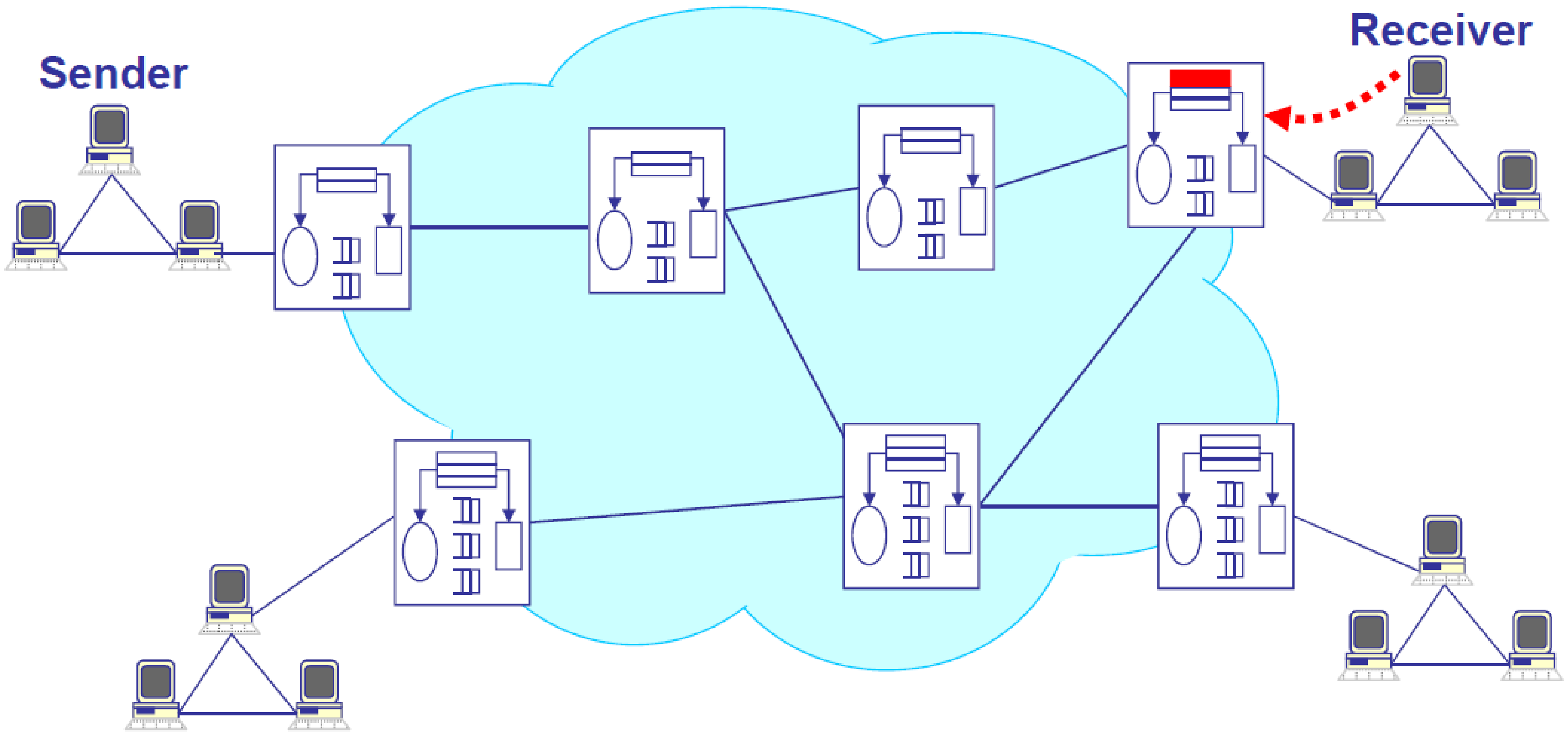
南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Allocate resources - perform per-flow admission control



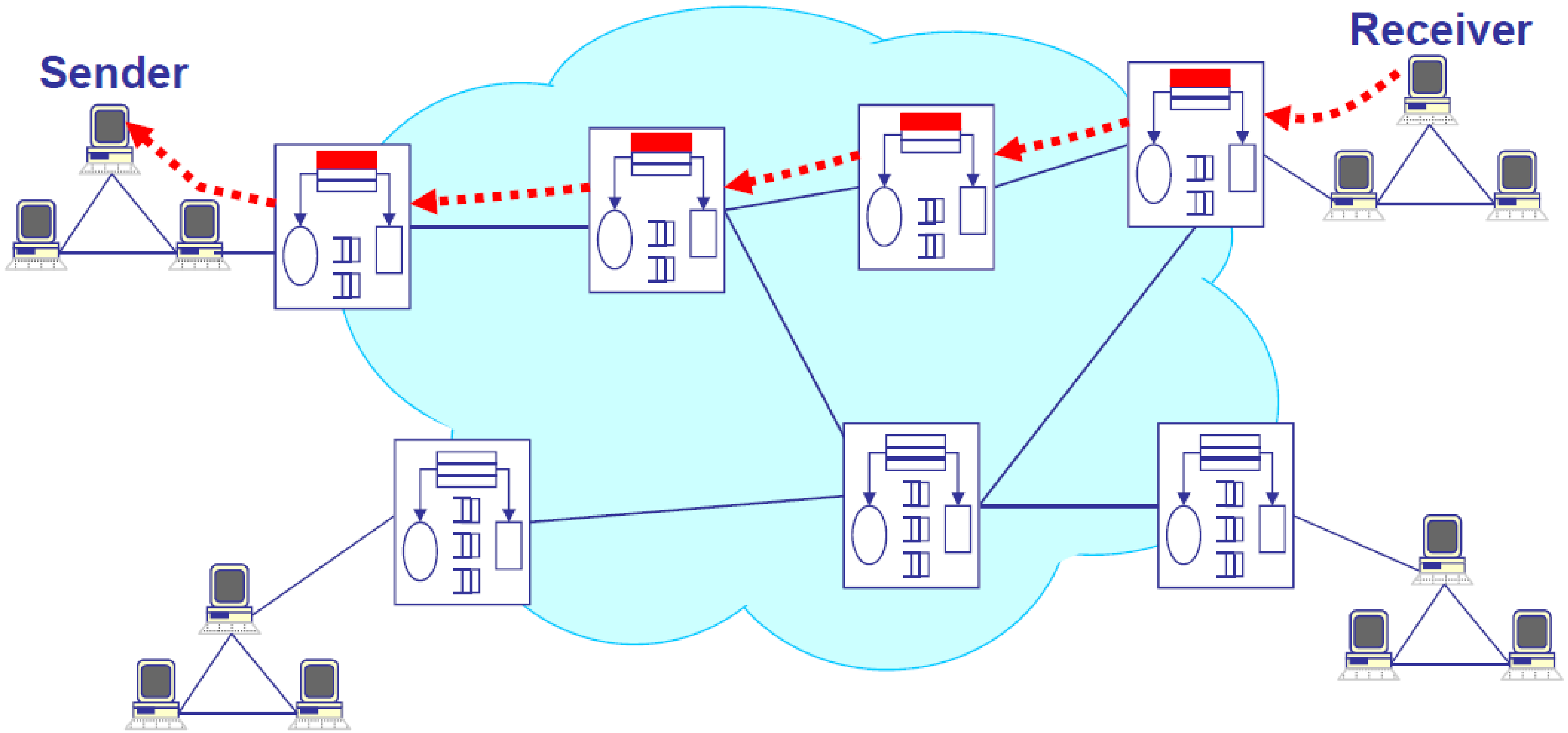
Integrated Services

- Install per-flow state



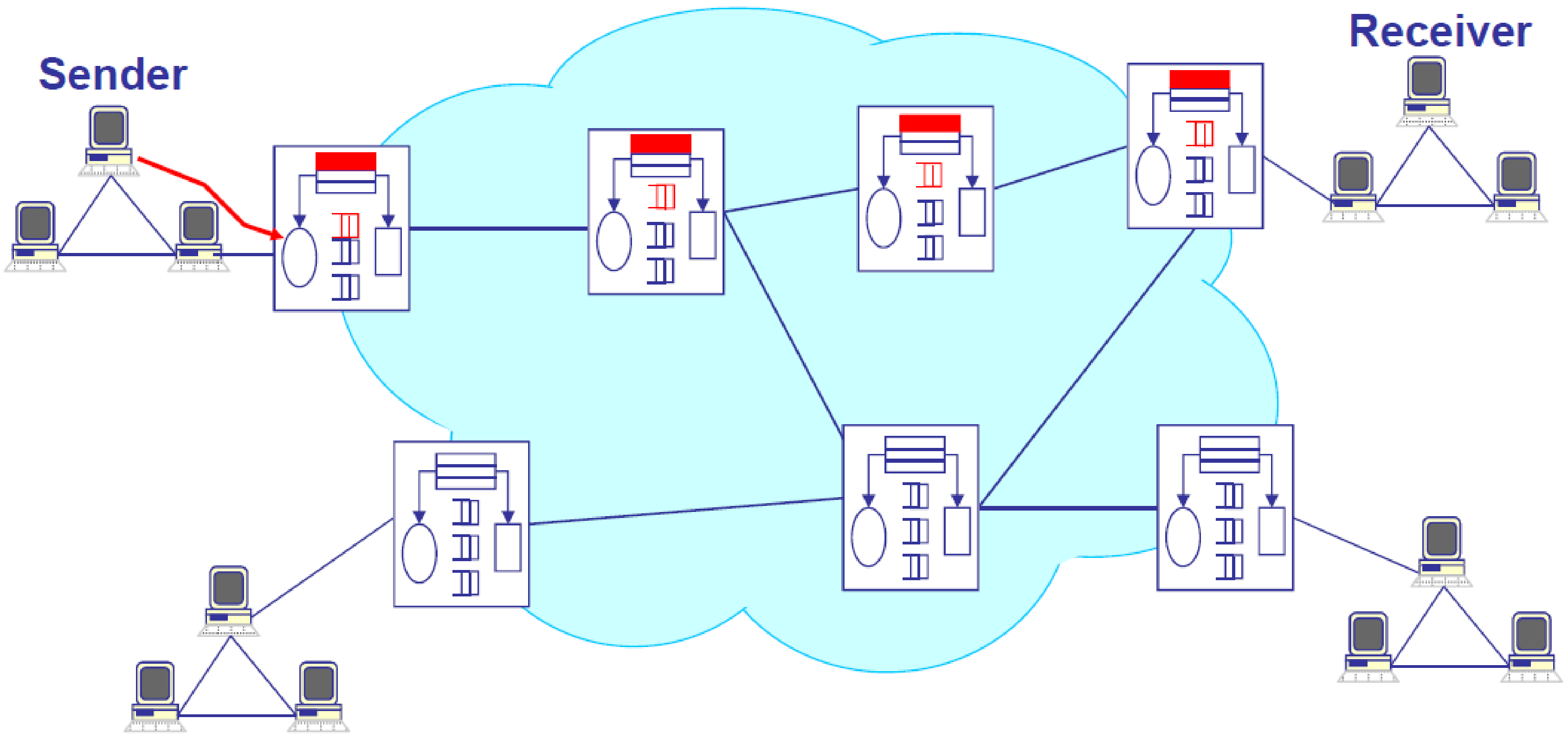
Integrated Services

- Install per-flow state



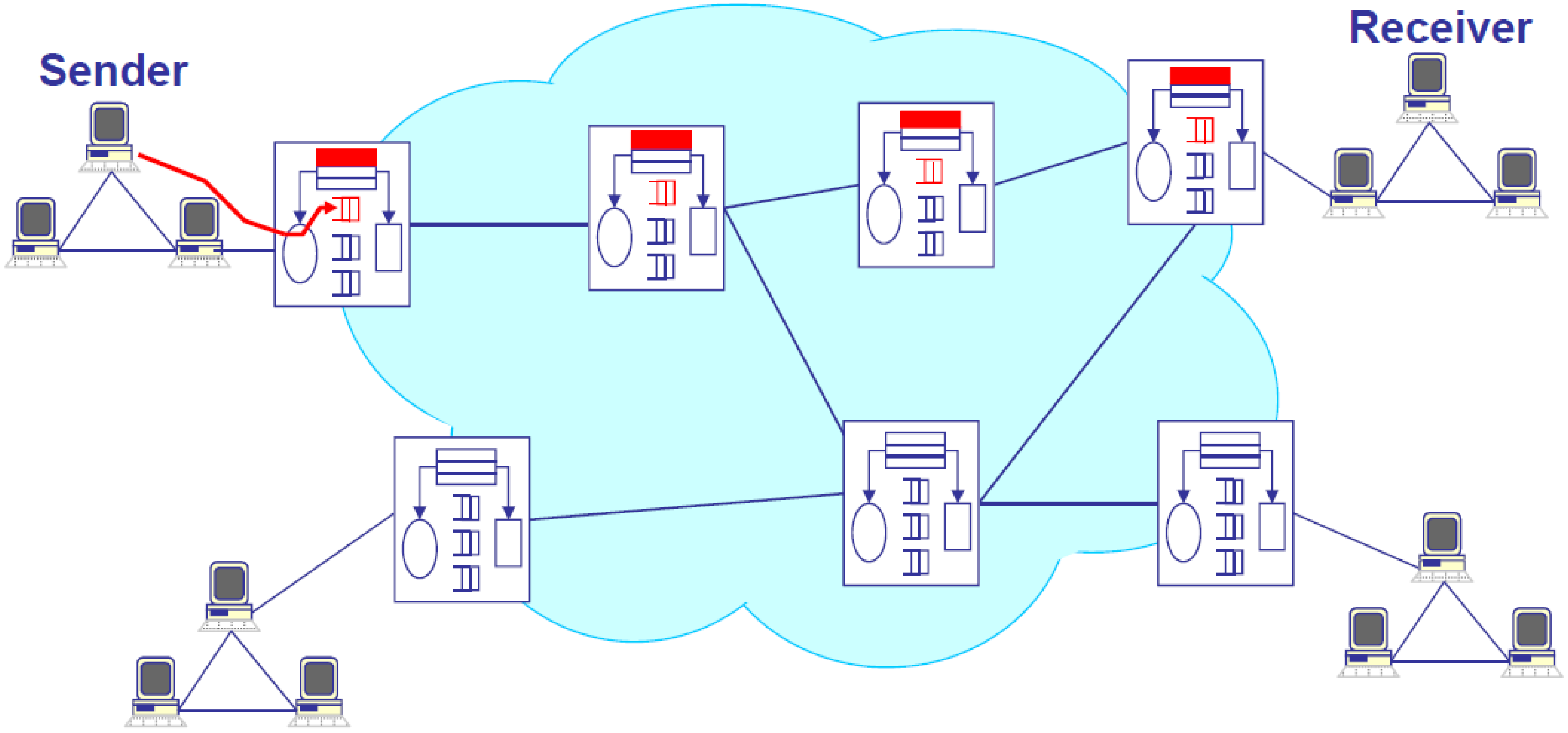
IntServe: Data Path

- Per-flow classification



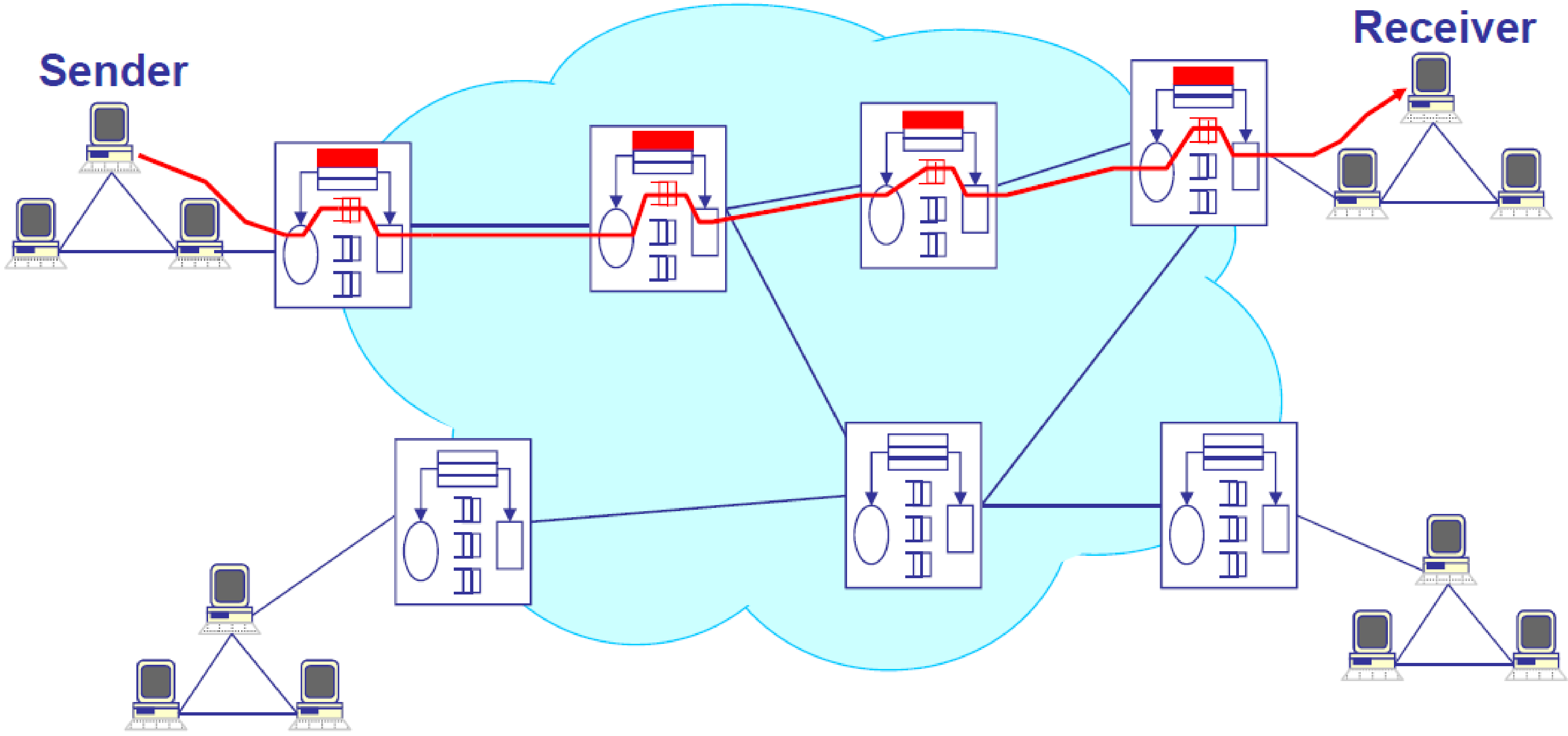
IntServe: Data Path

- Per-flow buffer management



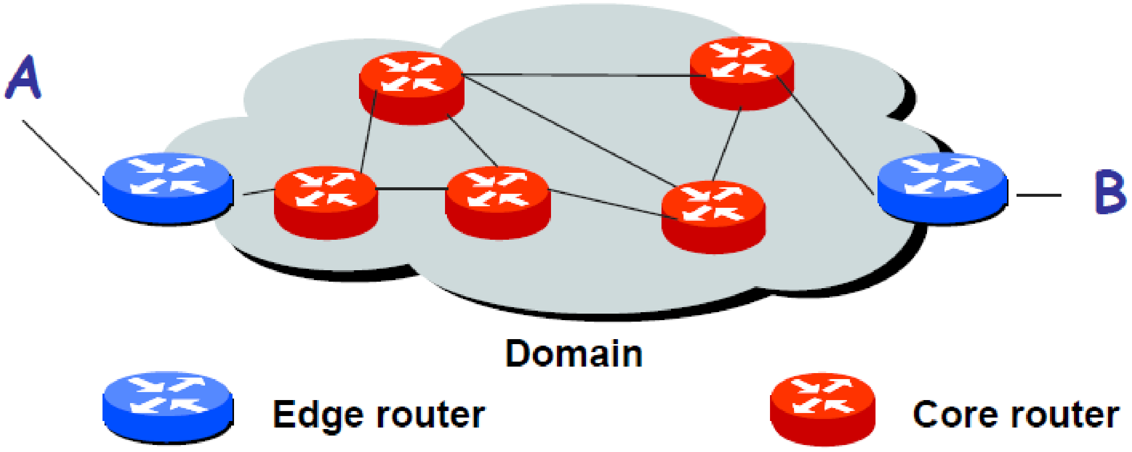
IntServe: Data Path

- Per-flow Scheduling



Differentiated Services

- Edge router
  - Shape & police traffic
  - Mark “class” of traffic in IP header field (e.g., gold service)
- Core router
  - Schedule aggregates according to marks in header
  - Drop lower-class traffic first during congestion



Summary

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- Quality-of-Service: What and Why?
- Per-hop QoS Enforcement
  - Buffer Managment: shaping, policing
  - Scheduling: Max-min, WFQ
- Network-wide Qos