

## Providing Differentiated Services in the Internet

These slides are created by Dr. Yih Huang of George Mason University. Students registered in Dr. Huang's courses at GMU can make a single machine-readable copy and print a single copy of each slide for their own reference, so long as each slide contains the copyright statement, and GMU facilities are not used to produce paper copies. Permission for any other use, either in machine-readable or printed form, must be obtained from the author in writing.

## Motive

- Traditional IP delivery model:
  - Best effort
  - Flat pricing
  - TOS largely ignored
  - All packets are equal
- Problems:
  - No way to distinguish important/critical applications from others
  - Example: transaction slowdown in the Christmas season of 1999

## Integrated Services

- **Per-flow** resource reservation and service level commitment.
- Pro: Provide Quality of Service (QoS) guarantees to individual applications.
- Con: Potentially too much burden on the core part of the Internet.
  - Consider the number of flows going through a national backbone router.

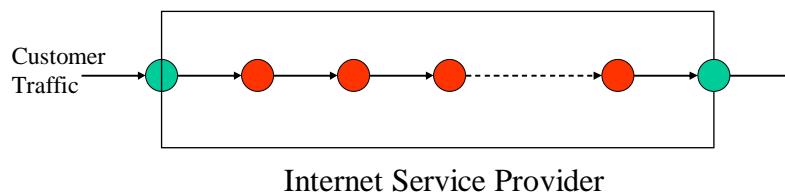
## Differentiated Services

- Resource reservation and service level commitment for **traffic aggregates**.
  - All traffic from a subnet
  - All TCP traffic, all real-time traffic, etc.
- Less burden on the network infrastructure.
- Can we still provide QoS guarantee to *individual* applications ?
- In this talk, we discuss the technologies that enable DiffServ and possibilities in service types.

## Building-Block Technologies

- Packet dropping policies
  - In the presence of network congestion, which packets are discarded ?
- Packet queueing and scheduling algorithms
  - In which order packets are delivered ?
- Traffic shaper/conditioner
- Traffic classifier
  - To classify incoming packets
- Service-level agreement (SLA)
  - The contract between an ISP and customer.

## Architecture



- Customer traffic aggregates are policed and shaped at the edge of the network (green routers).
- A small number of bits in the packet header are used to distinguish service classes.
- Core part of the network (red routers) uses these bits to decide how to treat packets

## Rational

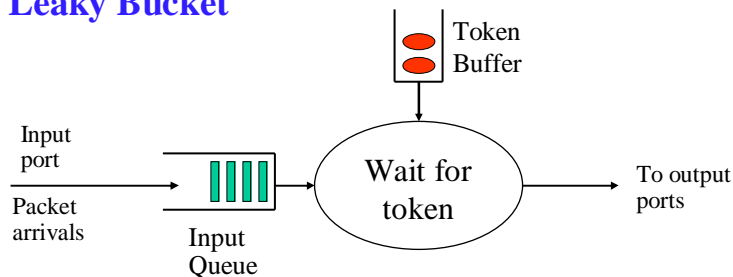
- ❑ Reduce the workload of the network core by pushing per-customer functions to the edge.
- ❑ For example, premium-service packets are policed and shaped at the edge on a per-customer basis.
- ❑ Premium-service packets from all customers are *not* distinguished once admitted to the network.
- ❑ Core routers sees only a small number of service classes, not individual customers.

## Per-Hop Behavior (PHB)

- ❑ A PHB defines how to treat a packet (priority in delivery, dropping policy, etc.)
- ❑ An IP datagram uses its type of service (TOS) field to specify a PHB.
  - Only 8 bits (hence not applicable to per-flow management)
  - Original semantics of TOS abandoned
  - Suitable to efficient (hardware) implementation

## Traffic Conditioner/Shaper

### □ Leaky Bucket



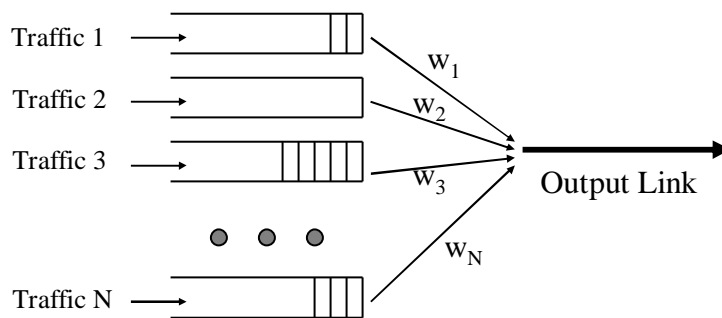
- Each traffic aggregate is associated with an **input queue** and a **token buffer**, both with predetermined capacities.
- Tokens arrive at a constant rate.
- Each packet consumes one token.

## Discussion

- Average packet output rate is determined by token arrival rate.
- To accommodate input bursts and to allow output bursts too, use a large input queue and a large token buffer.
- To accommodate input bursts but to produce perfectly regulated constant-rate output, use a large input queue and a size-one token buffer
- To punish input bursts, uses a small input queue so that packets in bursts are dropped.
- Leaky buckets are used at input ports of edge routers.

## Packet Scheduling

- We have  $N$  traffic aggregates, each one with a weight  $w_i$  and corresponds to a packet queue.



CS 756

Page 11

- The problem is to schedule the delivery of packets in the system.
  - Packets pertaining to the same aggregate must be delivered in their arrival order.
  - Packets belonging to different aggregates can be delivered out of order.
- A straightforward solution: round robin
  - Problem: all aggregates equally important; weights ignored.
  - We need something better.

CS 756

Page 12

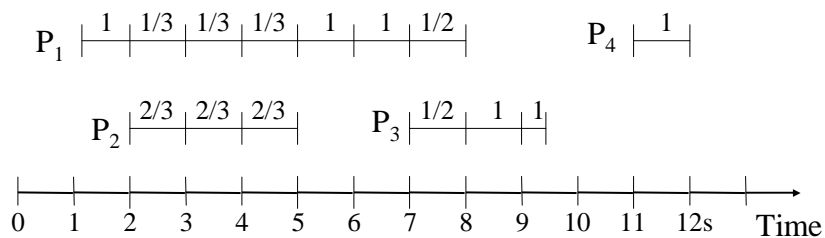
## Generalized Processor Sharing (GPS)

- An ideal reference model
- All non-empty queues will have fair access to the output link according to their weights, **simultaneously**.
  - GPS is only a model; it is unrealistic.

## Example

- Three traffic aggregates, A1, A2, and A3, with weights 1, 1, and 2, respectively.
- Link capacity is 10Mbps.
- If A3 is quiet, then A1 and A2 will each have 5Mbps of bandwidth.
- If all aggregates are busy, then A1, A2, and A3 will each have 2.5, 2.5, and 5 Mbps of bandwidth.

- Assume that
  - Packet P1 from traffic 1 of length 4 arrives at time 1
  - Packet P2 from traffic 3 of length 2 arrives at time 2
  - Packet P3 from traffic 2 of length 2 arrives at time 7
  - Packet P4 from traffic 1 of length 1 arrives at time 11
- With GPS, we have

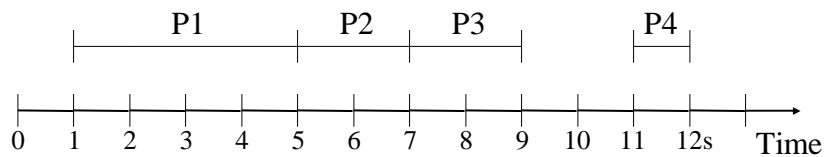


## Simulating GPS

- Of course, a link cannot transmit more than one packet at a time.
  - How do we simulate GPS as faithfully as possible ?
- Basic idea: packets are delivered in the increasing order of their completion times in GPS.
  - In our previous example, the four packets will be delivered in the order P2, P1, P3, and P4.
- Still impractical:
  - At time 1, how does the system know there will be P2 in the future ?

## Packet GPS (PGPS)

- Also known as **Weighted Fair Queueing (WFQ)**
- When we select the next packet for transmission, find the one that has the earliest completion time in the GPS system among “*known*” packets.
  - At time 1, P1 is the only known packet and its completion time in the GPS system with only P1 is 5; P1 is selected.



CS 756

Page 17

## PGPS Properties

- PGPS may deliver a packet later than GPS does.
  - Consider packet P2.
- It can be proved that this lateness is no worse than one packet time.
  - This shows that PGPS is a “high-fidelity” approximation of GPS, with bounded “distortions.”
- Moreover, the worst-case delay of packets belonging to traffic  $i$  can be derived from the bandwidth of the link and  $w_i$ .

CS 756

Page 18

## The Two-Bit DiffServ Architecture

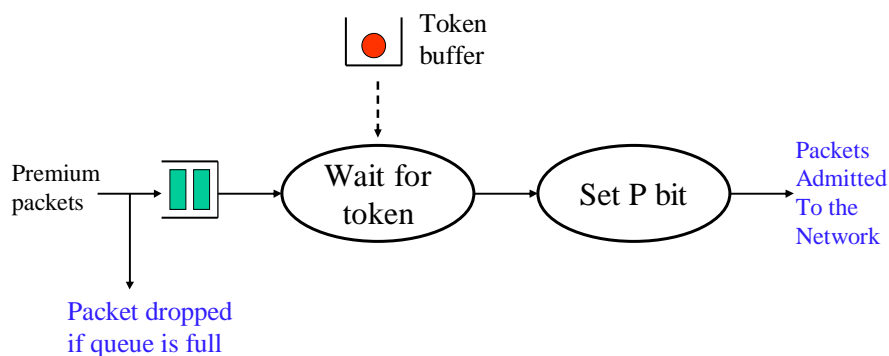
- ❑ RFC 2638
- ❑ Providing three classes of packet delivery services in one DiffServ Framework.
  - **Premium service** provides expedited delivery of packets.
  - **Assured service** provides reliable (low drop rate) delivery of packets.
  - **Basic service** corresponds to the traditional best-effort service model.
- ❑ Two bits (P-bit and A-bit) in TOS are used to specify the service class of a packet.

CS 756

Page 19

## Premium Traffic Conditioner

- ❑ The following block diagram is applied on a per-customer basis, at the edge of the network.



CS 756

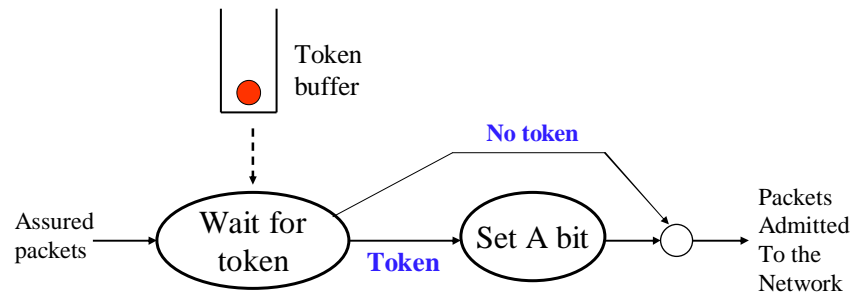
Page 20

## Discussion

- ❑ Token arrival rate conforms with the packet arrival rate specified in the contract.
- ❑ A small token-buffer is used to enforce the rate premium packets are admitted to the network.
  - The authors recommend size-1 token buckets.

- ❑ The size of the input queue reflects to what extend we want to tolerate short-term bursts.
  - A factor that must be specified in contract.
  - Premium packets in a burst will be stored in the queue to wait for the arrival of tokens.
  - When the queue is full, the customer violates the contract and subsequent packets are discarded.

## Assured Traffic Conditioner

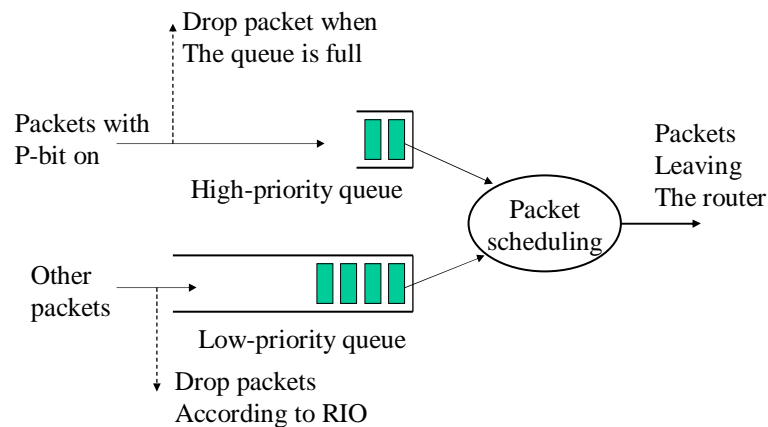


## Discussion

- ❑ The size of the token buffer reflects the length of bursts that we would like to tolerate, according to the contract.
- ❑ All assured-service packets are admitted to the network.
- ❑ However, out-of-profile packets lose the “assured privilege.”
  - Without the A-bit on, they will be treated just like basic-service packets.

## Forwarding Engine

- ❑ This is a per output interface mechanism, at both edge and core routers.



## Discussion

- ❑ Premium packets go to the high-priority queue.
- ❑ Both assured and basic packets go to the low-priority queue.
- ❑ Scheduling: Packets in the low-priority queue are transmitted only when the high-priority is empty.
- ❑ The size of the high-priority queue is set to the number of incoming premium streams.
- ❑ The rest of the buffer space are used by the low-priority queue.
- ❑ The low priority queue uses a packet-dropping algorithm called **RIO**.

## RIO

- ❑ RED with two sets of parameters.
- ❑ The first set of parameters ( $\min 1_{th}$ ,  $\max 1_{th}$ ,  $P_{max1}$ ) is applied to packets with A-bit on.
- ❑ The second set of ( $\min 2_{th}$ ,  $\max 2_{th}$ ,  $P_{max2}$ ) is applied to packets with A-bit off.
- ❑ Values of the two parameter sets are selected in a way such that “A” packets will be much less likely to be selected to drop.

## RIO Con't

- ❑ To ensure RIO drops non-A packets much earlier than it drops A packets:
  - $\min 1_{th} \gg \min 2_{th}$
- ❑ To ensure RIO drops non-A packets with higher probability:
  - $P_{max1} \ll P_{max2}$
- ❑ To ensure RIO drops *all* non-A packets much earlier than it drops *all* A packets:
  - $\max 1_{th} \gg \max 2_{th}$

# Putting Them Together

