Multimedia and QoS

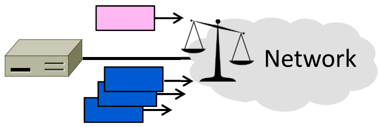
Lecture given by E. Lochin (ISAE-SUPAERO)

Slides extracted from David Wetherall lectures
Textbook A. Tanenbaum Computer Networks

Fair Queuing

Topic

- Sharing bandwidth between flows
  - WFQ (Weighted Fair Queuing)
  - Key building block for QOS

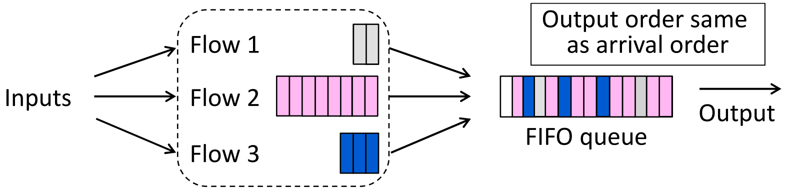


Sharing with FIFO Queuing

- FIFO "drop tail" queue:
  - Queue packets First In First Out (FIFO)
  - Discard new packets when full
  - Typical router queuing model
- Sharing with FIFO queue
  - Multiple users or flows send packets over the same (output) link
  - What will happen?

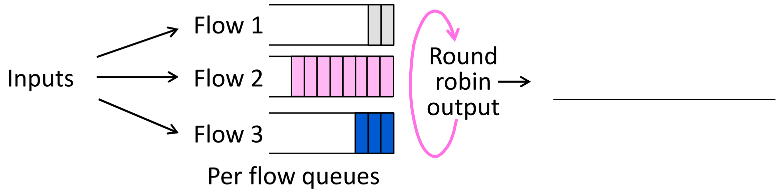
Sharing with FIFO Queuing (2)

- Bandwidth allocation depends on behavior of all flows
  - TCP gives long-term sharing – with delay/loss, and RTT bias
  - Aggressive user/flow can crowd out the others



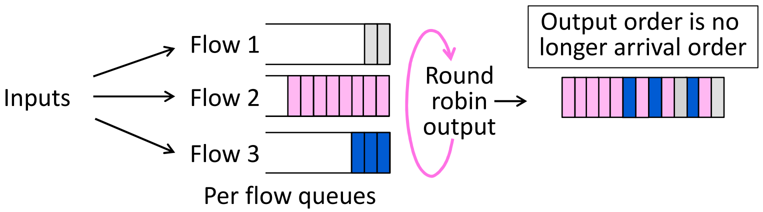
Round-Robin Queuing

- Idea to improve fairness:
  - Queue packets separately for each flow; take one packet in turn from each non-empty flow at the next output time



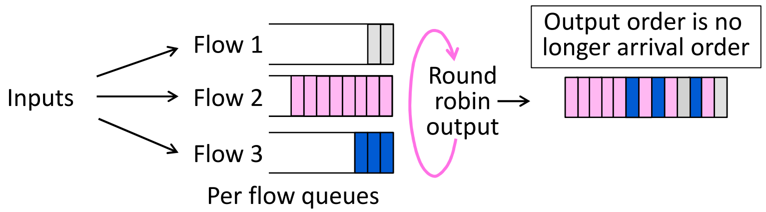
Round-Robin Queuing (2)

- Idea to improve fairness:
  - Queue packets separately for each flow; take one packet in turn from each non-empty flow at the next output time
  - How well does this work?



Round-Robin Queuing (3)

- Flows don’t see uncontrolled delay/loss from others!
- But different packet sizes lead to bandwidth imbalance
  - Might be significant, e.g., 40 bytes vs 1500 bytes



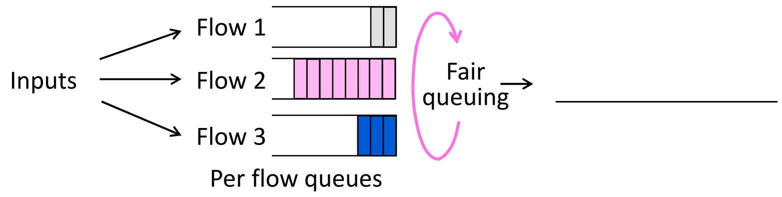
Fair Queuing

- Round-robin but approximate bit-level fairness:
  - Approximate by computing virtual finish time
  - Virtual clock ticks once for each bit sent from all flows
  - Send packets in order of their virtual finish times, $Finish(j)_F$
  - Not perfect – don't preempt packet being transmitted

$Arrive(j)_F$ = arrival time of j-th packet of flow F
$Length(j)_F$ = length of j-th packet of flow F
$Finish(j)_F = \max(Arrive(j)_F, Finish(j-1)_F) + Length(j)_F$

Fair Queuing (2)

- Suppose:
  - Flow 1 and 3 use 1000B byte packets, flow 2 uses 300B packets
  - What will fair queuing do?



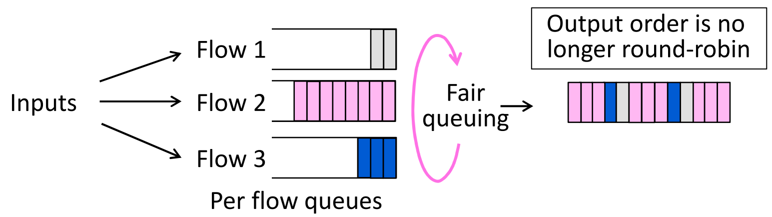
Fair Queuing (3)

- Suppose:
  - Flow 1 and 3 use 1000B packets, flow 2 uses 300B packets
  - What will fair queuing do?

Let $Finish(0)_F=0$, queues backlogged [$Arrive(j)_F < Finish(j-1)_F$]
$Finish(1)_{F1}=1000$, $Finish(2)_{F1}=2000$, …
$Finish(1)_{F2}=300$, $Finish(2)_{F2}=600$, $Finish(3)_{F2}=900$, 1200, 1500, …
$Finish(1)_{F3}=1000$, $Finish(2)_{F3}=2000$, …

Fair Queuing (4)

- Suppose:
  - Flow 1 and 3 use 1000B byte packets, flow 2 uses 300B packets
  - What will fair queuing do?



WFQ (Weighted Fair Queuing)

- WFQ is a useful generalization of Fair Queuing:
  - Assign a weight, $Weight_F$, to each flow
  - Higher weight gives more bandwidth, e.g., 2 is 2X bandwidth
  - Change computation of $Finish(j)_F$ to factor in $Weight_F$

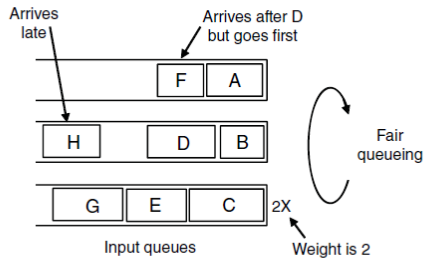
$Arrive(j)_F$ = arrival time of j-th packet of flow F

$Length(j)_F$ = length of j-th packet of flow F

$Finish(j)_F$ = max $(Arrive(j)_F, Finish(j\text{-}1)_F) + Length(j)_F / Weight_F$

WFQ Example

- An example you can work through …



Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

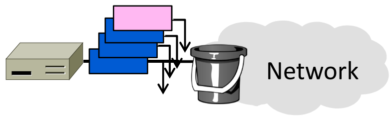
Using WFQ

- Lots of potential!
  - Can prioritize and protect flows
  - A powerful building block
- Not yet a complete solution
  - Need to determine flows (user? application? TCP connection?)
  - Difficult to implement at high speed for many concurrent flows
  - Need to assign weights to flows

Traffic Shaping

Topic

- Shaping traffic to constrain bursts
  - Token buckets
  - Key building block for QOS

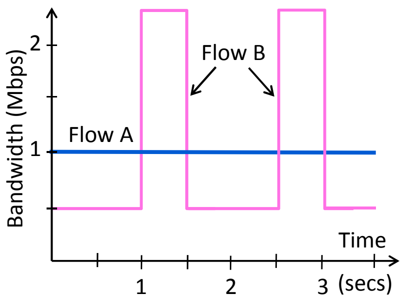


Motivation

- Shaping traffic flows constrains the load they may place on the network
  1. Limiting the total traffic enables bandwidth guarantees
  2. Limiting bursts avoids unnecessary delay and loss
- How should we shape traffic?
  - Real apps generate varying traffic – unrealistic to smooth it out

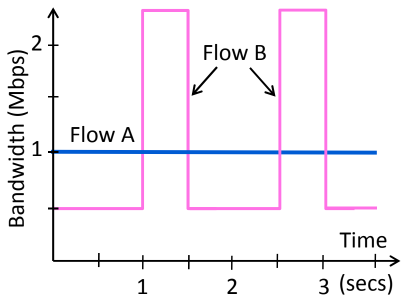
Motivation (2)

- Flow A and flow B have the same average rate
  - 1 Mbps over 3.5 secs
  - But they have very different behaviors!
- Average rate alone is not a good descriptor of behavior …



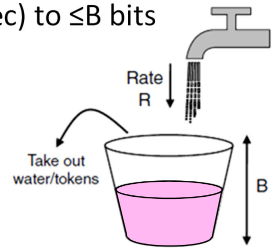
Motivation (3)

- How should we describe traffic flows to the network?
  - Average rate matters; relates to long-term bandwidth
  - Burstiness also matters; relates to short-term bandwidth
- Two characteristics useful
  - More expressive than average
  - Still relatively simple



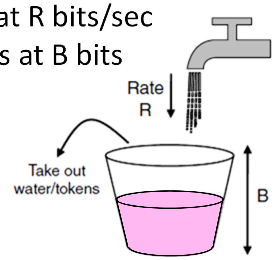
Token Bucket

- (R, B) token bucket constrains:
  - Average rate to ≤R bits/sec
  - Bursts (over R sec) to ≤B bits



Token Bucket (2)

- Sending removes tokens (or credits) from the bucket; no credit, no send
  - Continually fills at R bits/sec
  - Bucket overflows at B bits

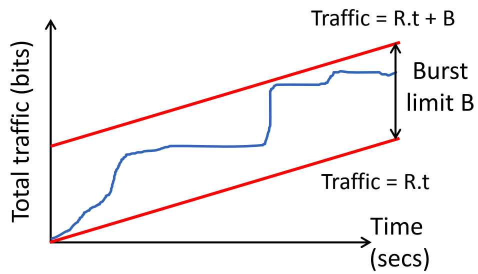


Token Bucket (3)

- Constrains greatest traffic over time

Token Bucket (4)

- Constrains greatest traffic over time



Shaping vs. Policing

- Shaping modifies traffic near the source to fit within an (R, B) profile
  - Run (R, B) token bucket at the source
  - Pass sent packets to the network when there are tokens
  - Delay (queue) packets while more tokens arrive
- Lets user condition their traffic to meet the network contract

Shaping vs. Policing (2)

- Policing verifies that traffic within the network fits an (R, B) profile
  - Run (R, B) token bucket at network edge
  - Let packets into the network when there are tokens
  - Demote or discard packets when there are insufficient tokens
- Lets network check traffic to verify it meets the user's contract

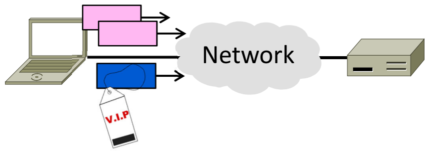
Usage for QOS

- Token buckets help the user and network regulate traffic for QOS
  - Network can limit the traffic for preferential treatment
  - User can flexibly select that traffic
- Special treatment is implemented with other means such as WFQ

Differentiated Services

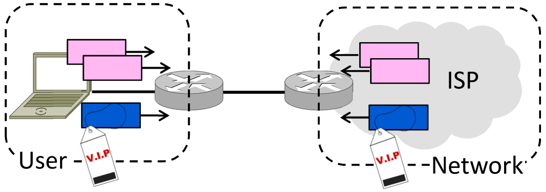
Topic

- Treating different traffic flows differently in the network
  - Coarse QOS (grades of service)
  - Gradually being deployed



Motivation

- User runs Skype and BitTorrent
  - Or remote desktop, gaming, web, etc.
  - How can we give preference to flows?

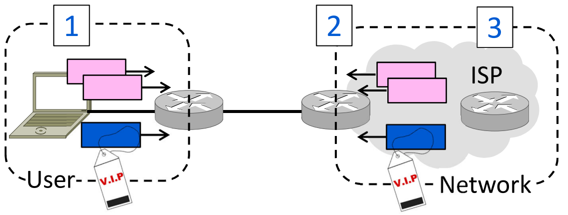


Differentiated Services

- Idea is to treat different kinds of traffic differently in the network
  - Have a few kinds of network service (GOLD, SILVER, BRONZE)
  - Different kinds get better or worse treatment in the network
  - Map apps to the right kind of service

Differentiated Services (2)

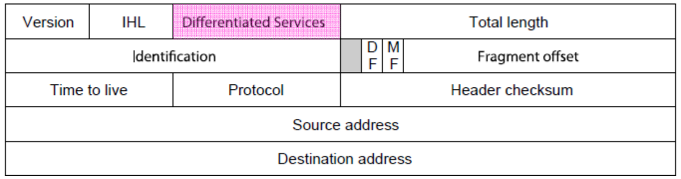
- Architecture:
  1. User marks packet with desired service (e.g., Skype=GOLD)
  2. Network polices traffic levels at boundary (token bucket)
  3. Network provides different forwarding (WFQ at routers)



1. Marking Packets

- Use bits in IPv4/IPv6 header to mark the kind of service
  - 6-bit DSCP (Differentiated Services Code Point)

IPv4 Header



Marking Packets (2)

- Many possible DSCP markings for different service/apps
  - Supported services depend on configuration of network

Service Name / Meaning	DSCP Value	Traffic Need (App example)
Default forwarding / Best effort	0	Elastic (BitTorrent)
Assured forwarding / Enhanced effort	10-38	Average rate (streaming video)
Expedited forwarding / Real-time	46	Low loss/delay (VoIP, gaming)
Precedence / e.g., Network control	48	High priority (Routing protocol)

Marking Packets (3)

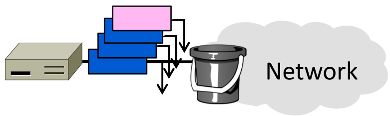
- Traffic is marked by user
  - Depends on local policies, e.g., gaming = expedited?
- May be done as part of host
  - Let OS or app classify their traffic
- May be done inside the network
  - Using heuristics, such as ports

2. Policing Packets

- Network (ISP) checks incoming traffic meets service contract
  - Not more expedited traffic than agreed (and paid for!)
  - Only allowed markings, e.g., no network control from users

Policing Packets (2)

- Policing is done with token bucket
  - Can demote "out of profile" traffic by re-marking (e.g., to default / best effort) or prioritizing for loss

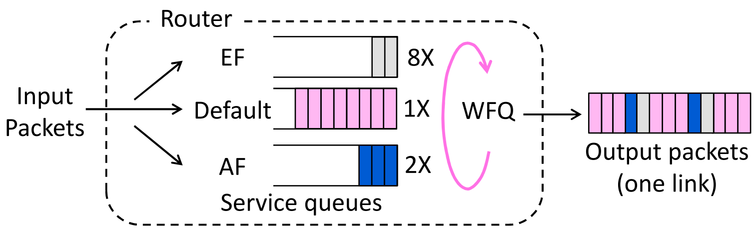


3. Forwarding Packets

- Network (ISP) routers use WFQ (and more) instead of FIFO
- The different kinds of service are the different flows/queues
- DSCP values are used to map packet to the right flow/queue

Forwarding Packets (2)

- Services are defined as "per hop behaviors"
  - No guarantee for end-to-end service through a network
  - Need small amounts of high priority traffic for good service



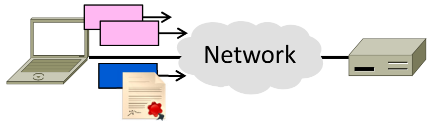
Deployment

- QOS provides value when it is deployed across the network
  - Not much use if only your ISP!
- QOS is tightly tied to pricing
  - "All my packets are high priority"
- Makes deployment slow/difficult …

Rate and Delay Guarantees

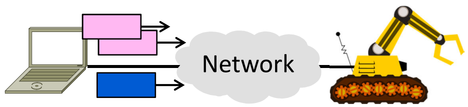
Topic

- Guaranteeing performance for traffic flows across in the network
  - This is "hard QOS" with a firm guarantee for a traffic flow



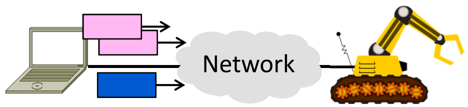
Motivation

- Sometimes we want guaranteed service – like the telephone network
  - Minimum rate and maximum delay regardless of how other flows behave
  - e.g., robotic control?



Motivation (2)

- Could provision a dedicated circuit (or build a network), but expensive
- Can we have statistical multiplexing together with hard guarantees?

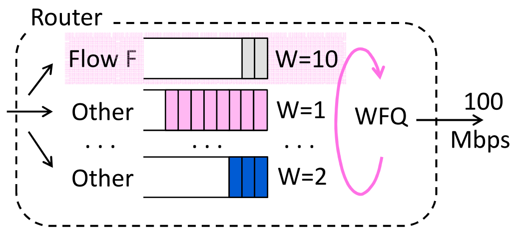


Admission Control

- Suppose flow F needs rate ≥R Mbps and delay ≤D secs
- We must decide whether to admit or reject it from the network
  - This is <u>admission control</u>
  - Rejecting should be infrequent
- Key point is we need the ability to control load to make guarantees

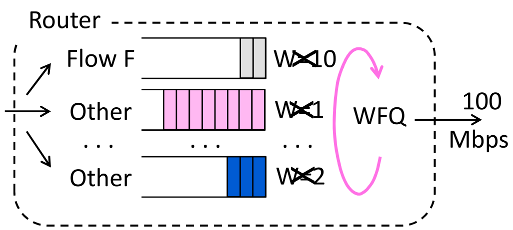
Router Rate Guarantee

- WFQ can guarantee rate at a router
  - What rate will Flow F get?



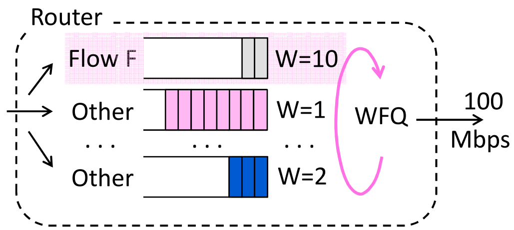
Router Rate Guarantee (2)

- Consider N flows with weight 1
  - Each flow gets 1/Nth share under load
  - Or at least 100/N Mbps



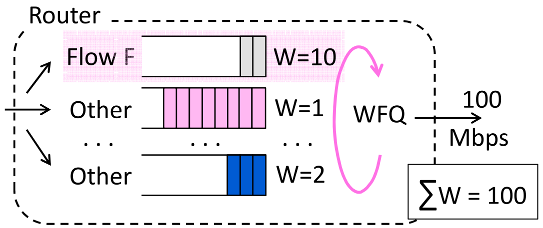
Router Rate Guarantee (3)

- Consider flow F with weight 10
  - Suppose weight of all flows is 100



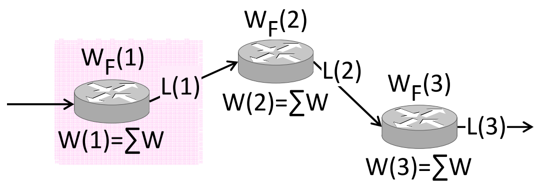
Router Rate Guarantee (4)

- Consider flow F with weight 10
  - Flow F gets ≥(10/100).100 = 10 Mbps



Network Rate Guarantee

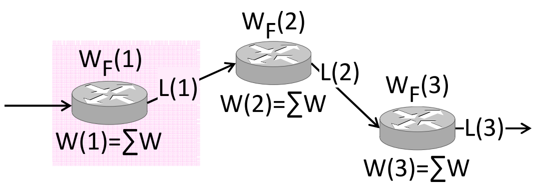
- We can guarantee a minimum rate for a network path by guaranteeing it at each router



Network Rate Guarantee (2)

- Condition for each router:

For all routers i:
$W_F(i) / W(i) * L(i) \geq R$ Mbps

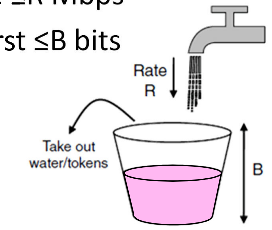


Delay Guarantee

- What about the queuing delay?
  - How much larger than latency might the delay be, given rate guarantee?
- It depends on the traffic flow
  - If exceeds R Mbps then queues may build and delay will grow …
- Need to shape traffic for guarantee
  - We'll use token buckets ☺

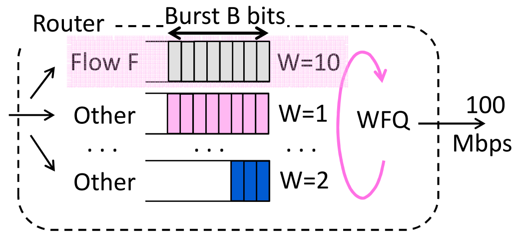
Router Delay Guarantee

- Assume traffic flow F is shaped by an (R, B) token bucket
  - Long-term rate ≤R Mbps
  - Short-term burst ≤B bits



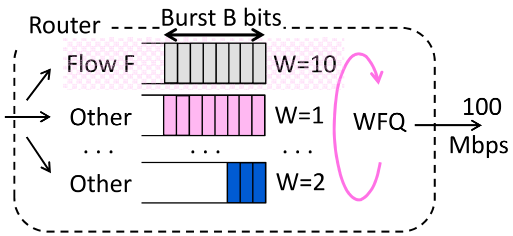
Router Delay Guarantee (2)

- What is delay of flow F at a router?
  - Traffic shaped by (R, B) token bucket
  - WFQ with weight set for rate ≥ R Mbps



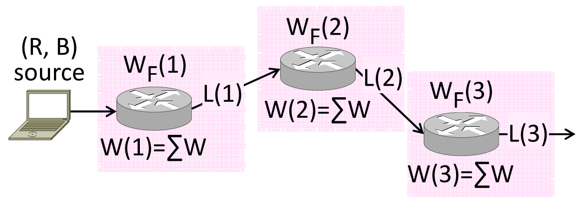
Router Delay Guarantee (3)

- In worst case B arrives all at once
  - So queuing delay is ≤B/R seconds



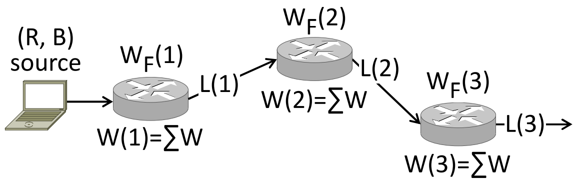
Network Delay Guarantee

- What is the delay across N routers?
  - This is tricky! Each router add delays
  - Bound of N*B/R is too loose
  - Intuitive argument follows …



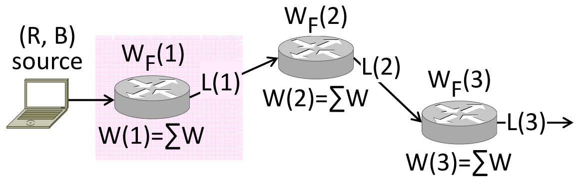
Network Delay Guarantee (2)

- If traffic is perfectly smooth at rate R (no bursts) then queuing delay is zero
  - Packet enters router just in time to leave
  - Delay is latency (propagation, transmission)



Network Delay Guarantee (3)

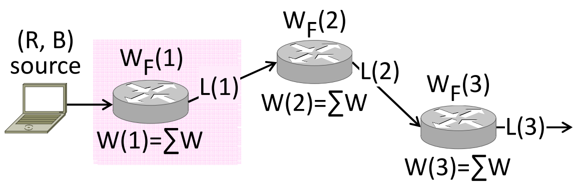
- Observe if traffic pays for burst B at one router, it is smoothed for the next
  - Burst delay is only paid once!



Network Delay Guarantee (4)

- Delay across N routers:

$$\boxed{\text{Delay} \leq \text{Latency terms} + B/R}$$



Rate/Delay Guarantee

- Given a network with:
  - (R, B) shaped traffic flow
  - WFQ routers with proper weights
  - Sharing via statistical multiplexing
- We can guarantee the flow a minimum rate and maximum delay
  - Rate is ≥R Mbps
  - Delay is ≤ latency + B/R secs
  - Regardless of how other flows behave