Packet Scheduling

- Decide when and what packet to send on output link
  - Usually implemented at output interface of a router

Goals of Packet Scheduling

- Provide per flow/aggregate QoS guarantees in terms of delay and bandwidth
- Provide per flow/aggregate protection
- Flow/Aggregate identified by a subset of following fields in the packet header
  - source/destination IP address (32 bits)
  - source/destination port number (16 bits)
  - protocol type (8 bits)
  - type of service (8 bits)
- Examples:
  - All packets from machine A to machine B
  - All packets from Berkeley
  - All packets between Berkeley and MIT
  - All TCP packets from EECS-Berkeley

Outline

- Token Bucket and Arrival Curve
  - Weighted Fair Queueing (WFQ)
  - Link sharing

Token Bucket

- A simple traffic model used to:
  - Enforce traffic sent by a sender (e.g., cable modem)
  - Characterize the traffic sent by a sender

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

\( \text{Maximum \# of bits sent} = \frac{b \times R}{R - r} \)
**Traffic Enforcement: Example**

- \( r = 100 \text{ Kbps} \); \( b = 3 \text{ Kb} \); \( R = 500 \text{ Kbps} \)

(a) \( T = 0 \): 1Kb packet arrives

(b) \( T = 2\text{ms} \): packet transmitted
\[ b = 3\text{Kb} - 1\text{Kb} + 2\text{ms} \times 100\text{Kbps} = 2.2\text{Kb} \]

(c) \( T = 4\text{ms} \): 3Kb packet arrives

(d) \( T = 10\text{ms} \): packet needs to wait until enough tokens are in the bucket!

(e) \( T = 16\text{ms} \): packet transmitted

**Source Traffic Characterization: Arrival Curve**

- Arrival curve – maximum amount of bits transmitted during an interval of time \( \Delta t \)
- Use token bucket to bound the arrival curve

**Arrival Curve: Example**

- Arrival curve – maximum amount of bits transmitted during an interval of time \( \Delta t \)
- Use token bucket to bound the arrival curve

**QoS Guarantees: Per-hop Reservation**

- End-host: specify
  - the arrival rate characterized by token bucket with parameters \( b,r,R \)
  - the maximum maximum admissible delay \( D \), no losses
- Router: allocate bandwidth \( r_a \) and buffer space \( B_a \) such that
  - no packet is dropped
  - no packet experiences a delay larger than \( D \)

**Packet Scheduling**

- Make sure that at any time the flow receives at least the allocated rate \( r_a \)
- The canonical example of such scheduler: Weighted Fair Queueing (WFQ)

**Outline**

- Token Bucket and Arrival Curve
  - Weighted Fair Queueing (WFQ)
- Link sharing
Recap: Fair Queueing

- Implements max-min fairness: each flow receives \( \min(r, f) \), where
  - \( r \) – flow arrival rate
  - \( f \) – link fair rate (see next slide)

- Weighted Fair Queueing (WFQ) – associate a weight with each flow

Fair Rate Computation: Example 1

- If link congested, compute \( f \) such that
  \[
  \sum_i \min(r_i, f) = C
  \]

Fair Rate Computation: Example 2

- Associate a weight \( w_i \) with each flow \( i \)
- If link congested, compute \( f \) such that
  \[
  \sum_i w_i \min(r_i, f) = C
  \]

Fluid Flow System

- 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

<table>
<thead>
<tr>
<th>Flow 1 (arrival traffic)</th>
<th>Flow 2 (arrival traffic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Size (bits)</td>
<td>Packet inter-arrival time (ms)</td>
</tr>
<tr>
<td>Flow 1</td>
<td>1000</td>
</tr>
<tr>
<td>Flow 2</td>
<td>500</td>
</tr>
</tbody>
</table>

Service in fluid flow system:

- **Flow 1** (arrival traffic)
  - Packet size: 100 bits
  - Packet inter-arrival time: 10 ms
  - Rate: 100 Kbps

- **Flow 2** (arrival traffic)
  - Packet size: 50 bits
  - Packet inter-arrival time: 50 ms
  - Rate: 50 Kbps

Fluid Flow System: Example 2

- Red flow has packets backlogged between time 0 and 10
  - Backlogged flow's queue not empty
- Other flows have packets continuously backlogged
- All packets have the same size
**Implementation In Packet System**

- Packet (Real) system: 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

**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 hundreds to thousands of flows!

**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 at which every backlogged flow receives service in the fluid flow system.
- \( C \) – 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**

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

**System Virtual Time: Example**

- 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

\[
F^{i}_{k+1} = \max( V(a^{i}_k) + L^{i}_k, F^{i}_k + \frac{L^{i}_k}{w_i})
\]

**Properties of WFQ**

- Guarantee that any packet is transmitted within packet length/link capacity of its transmission time in the fluid flow system.
  - Can be used to provide guaranteed services.
- Achieve max-min fair allocation.
  - Can be used to protect well-behaved flows against malicious flows.

**Outline**

- Token Bucket and Arrival Curve
- Weighted Fair Queueing (WFQ)
  - Link sharing
Hierarchical Link Sharing

- Resource contention/sharing at different levels
- Resource management policies should be set at different levels, by different entities
  - Resource owner
  - Service providers
  - Organizations
  - Applications

Packet Approximation of H-WFQ

- Idea 1
  - Select packet finishing first in H-WFQ assuming there are no future arrivals
  - Problem:
    - Finish order in system dependent on future arrivals
    - Virtual time implementation won’t work
- Idea 2
  - Use a hierarchy of WFQ to approximate H-WFQ

Problems with Idea 1

- The order of the 4th blue packet finish time and of the first green packet finish time changes as a result of a red packet arrival

Problem with Idea 2

- A packet on the second level can miss its deadline (finish time)

Solution

- Hierarchical-WFQ with a better implementation of WFQ, called Worst-Case Weighted Fair Queueing (WF2Q)
- Main idea of WF2Q
  - Consider for scheduling only eligible packets
  - Eligible packet at time t a packet that has started being serviced in the fluid flow system at time t
Example

Hierarchical-WF2Q Example

- In WF2Q, all packets meet their deadlines modulo time to transmit a packet (at the line speed) at each level

What You Need to Know

- Basic concepts
  - Arrival & service curve
  - Token-bucket specification
  - System virtual time / finish virtual time
  - WFQ properties
  - Link sharing requirements and challenges

- Mechanisms:
  - WFQ implementation in the fluid flow & packet system

- You don’t need to know
  - Details of WF2Q
  - How service curve works