EECS 489 - FA 24
Discussion 8

Assignment 3 and Routing

Announcements

Assignment 3 is out.

Due date: **Friday, November 8, 2024**

Lateday policy:

You have 3 group latedays in total for assignment 2 - 4.

Agenda

- A3 Hints
- Routing Protocol Questions

A3 Hints

WTP-base

Your sender and receiver should work given:

- Timeouts/Large amounts of latency (think 100s of ms)
- Packet loss
- Packet corruption
- Multiple file transfers for 1 receiver lifespan
- Large binary or text file transfers (A video file, a very long text file, etc.)

A3 Hints

WTP-opt

Your sender and receiver should work given:

- Same conditions as last slide
- ACKs should have precisely the same seq as what was sent
- Packets that have been ACK’d should not be retransmitted

i.e make sure the expected “optimizations” are observable

<u>Do not use TCP sockets, the AG knows when you are doing this!</u>

Q1: IPv6

Which of the following were eliminated in IPv6?

1. Fragmentation
2. Checksum
3. Header Length
4. Version
5. TTL

Q1: IPv6

Which of the following were eliminated in IPv6?

1. **Fragmentation:** Hosts typically run an algorithm to discover the MTU at the beginning of communication, so no fragmentation is necessary.
2. **Checksum:** Higher-level protocols (e.g. TCP) generally take care of this, so IP can ignore it. Inefficient to do this for communications that don't need it.
3. **Header Length:** IPv6 has fixed header length, so this is not necessary.
4. Version: Still needed!
5. **TTL (kinda):** Replaced by *hop limit* instead, which is basically the same thing.

Q2: Forwarding Table

Consider a datagram network using 16-bit addressing. Suppose a router has 4 links, and packets are to be forwarded as follows:

Destination Address Range	Interface
from: 11100000 00000000 to: 11100000 00111111	0
from: 11100001 00000000 to: 11100001 11111111	1
from: 11100000 01000000 to: 11100000 10111111	2
otherwise	3

Provide a forwarding table using longest prefix matching.

Q2: Forwarding Table

Destination Address Range	Interface
11100000 00 (/10)	**0**
11100001 (/8)	**1**
11100000 01 (/10)	**2**
11100000 10 (/10)	**2**
otherwise	**3**

Q3: Packet Forwarding

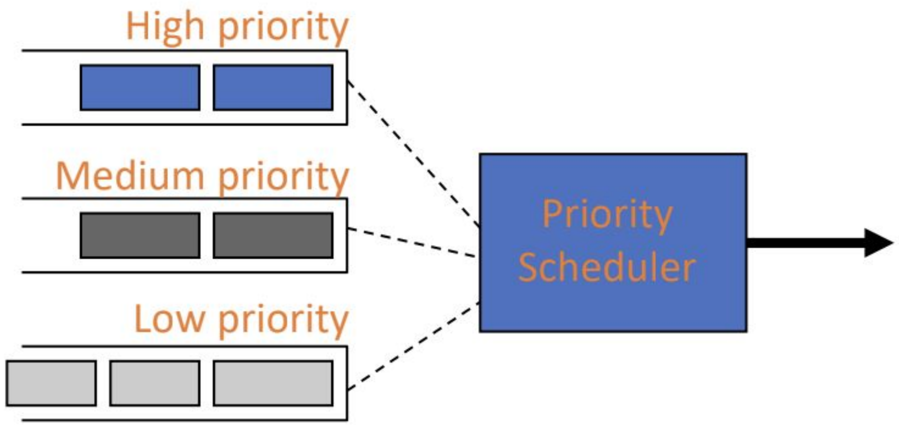
Describe how naive priority scheduling works. What problems could this possibly pose?

Q3: Packet Forwarding

Describe how naive priority scheduling works. What problems could this possibly pose?

Naive priority scheduling, when having multiple queues of packets, simply always services the queue with the highest priority (that requires servicing).

Starvation is the primary problem with this algorithm; if there are always packets available in higher-priority queues, a lower-priority queue will never be serviced, which is typically not ideal.



Q4: Packet Forwarding

Suppose we have queues with the following packet sizes (ordered from front to back). Assume that packet forwarding algorithms break ties between queues by choosing the queue with the lowest number (i.e. Queue 1 > Queue 2 > Queue 3). **What is the order of queues serviced under a round-robin scheduling algorithm?**

- **Queue 1 (Priority Weight 5):** 1000, 500, 500
- **Queue 2 (Priority Weight 2):** 200, 200, 200, 200, 200, 200, 200, 200
- **Queue 3 (Priority Weight 1):** 400, 400

Q4: Packet Forwarding

Suppose we have queues with the following packet sizes (ordered from front to back). Assume that packet forwarding algorithms break ties between queues by choosing the queue with the lowest number (i.e. Queue 1 > Queue 2 > Queue 3). **What is the order of queues serviced under a round-robin scheduling algorithm?**

- **Queue 1 (Priority Weight 5):** 1000, 500, 500
- **Queue 2 (Priority Weight 2):** 200, 200, 200, 200, 200, 200, 200, 200
- **Queue 3 (Priority Weight 1):** 400, 400

1, 2, 3, 1, 2, 3, 1, 2, 2, 2, 2, 2, 2

Q5: Packet Forwarding

Suppose we have queues with the following packet sizes (ordered from front to back). Assume that packet forwarding algorithms break ties between queues by choosing the queue with the lowest number (i.e. Queue 1 > Queue 2 > Queue 3). **What is the order of queues serviced under a fair queuing scheduling algorithm?**

- **Queue 1 (Priority Weight 5):** 1000, 500, 500
- **Queue 2 (Priority Weight 2):** 200, 200, 200, 200, 200, 200, 200, 200
- **Queue 3 (Priority Weight 1):** 400, 400

Q5: Packet Forwarding

Suppose we have queues with the following packet sizes (ordered from front to back). Assume that packet forwarding algorithms break ties between queues by choosing the queue with the lowest number (i.e. Queue 1 > Queue 2 > Queue 3). **What is the order of queues serviced under a fair queuing scheduling algorithm?**

- **Queue 1 (Priority Weight 5):** 1000, 500, 500
- **Queue 2 (Priority Weight 2):** 200, 200, 200, 200, 200, 200, 200, 200
- **Queue 3 (Priority Weight 1):** 400, 400

Fair queueing will always choose the queue with the lowest number of bytes served so far.

Serviced:	Init.	**Q1**	**Q2**	**Q3**	**Q2**	**Q2**	**Q3**	**Q2**	**Q2**	**Q1**	**Q2**	**Q2**	**Q2**	**Q1**
Q1 Bytes	0	1000	1000	1000	1000	1000	1000	1000	1000	1500	1500	1500	1500	2000
Q2 Bytes	0	0	200	200	400	600	600	800	1000	1000	1200	1400	1600	1600
Q3 Bytes	0	0	0	400	400	400	800	800	800	800	800	800	800	800

Q6: Packet Forwarding

Suppose we have queues with the following packet sizes (ordered from front to back). Assume that packet forwarding algorithms break ties between queues by choosing the queue with the lowest number (i.e. Queue 1 > Queue 2 > Queue 3). **What is the order of queues serviced under a weighted fair queuing scheduling algorithm?**

- **Queue 1 (Priority Weight 5):** 1000, 500, 500
- **Queue 2 (Priority Weight 2):** 200, 200, 200, 200, 200, 200, 200, 200
- **Queue 3 (Priority Weight 1):** 400, 400

Q6: Packet Forwarding

Suppose we have queues with the following packet sizes (ordered from front to back). Assume that packet forwarding algorithms break ties between queues by choosing the queue with the lowest number (i.e. Queue 1 > Queue 2 > Queue 3). **What is the order of queues serviced under a fair queuing scheduling algorithm?**

- **Queue 1 (Priority Weight 5):** 1000, 500, 500
- **Queue 2 (Priority Weight 2):** 200, 200, 200, 200, 200, 200, 200, 200
- **Queue 3 (Priority Weight 1):** 400, 400

Weighted fair queuing computes the metric of **(number of bytes served / priority weight)** and services the queue with the lowest such metric.

Serviced:	Init.	**Q1**	**Q2**	**Q3**	**Q2**	**Q1**	**Q2**	**Q1**	**Q2**	**Q2**	**Q3**	**Q2**	**Q2**	**Q2**
Q1 Metric	0	200	200	200	200	300	300	400	400	400	400	400	400	400
Q2 Metric	0	0	100	100	200	200	300	300	400	500	500	600	700	800
Q3 Metric	0	0	0	400	400	400	400	400	400	400	800	800	800	800