CS144
Intro to Computer Networks
Midterm Exam – Wednesday, Oct 21st, 2020

OPEN NOTES AND SLIDES

Your Name: Answers

SUNet ID: root @stanford.edu

In accordance with both the letter and the spirit of the Stanford Honor Code, I neither received nor provided any assistance on this exam.

Signature: ______________________________

- The exam has 10 questions totaling 63 points.
- You have 60 minutes to complete them.
- Some questions may be much harder than others.
- Keep your answers concise.
- For multiple-choice questions, circle all true answers. You will be credited points for correctly circled answers and well as for answers correctly left blank.
- You may express numerical answers as expressions if you don't do the arithmetic, but please explain each term when you do.

1	/3
2	/3
3	/3
4	/3
5	/3
6	/3
7	/10
8	/15
9	/15
10	/5
Total	/63

I Multiple Choice

1. [3 points]:

In Nandita Dukkipati's guest lecture *Congestion Control in the Real World*, which of the following were used as methods for estimating the congestion in a network?

Circle all that apply. (There may be zero, one, or multiple answers.)

A One of the packet's header fields can be modified by switches/routers in the network.

B The time between transmission of a packet and arrival of its acknowledgment.

C Size of the congestion window for a flow.

D Probability that a packet is dropped.

Answer:

A (e.g. ECN Marking), B (RTT), D (Packet loss)

2. [3 points]:

Why is the retransmission timeout (RTO) approximately equal to the round-trip-time (RTT)?

Circle all that apply. (There may be zero, one, or multiple answers.)

A A timeout much smaller than the estimated RTT will lead to unnecessary retransmissions.

B A timeout that is roughly equal to the RTT gives enough time to the receiver application to process data.

C A timeout that is roughly equal to the RTT gives the sender enough time to receive ACKs from the receiver.

D A timeout much larger than the estimated RTT may result in poor network utilization.

E If a packet is going to be dropped in the network, it is dropped in RTT amount of time after its transmission from the sender.

Answer:

A because timeout shorter than RTT means all the packets will be retransmitted before the acknowledgement arrives even if they are successfully delivered to the destination, C because confirmation regarding a successful delivery of a packet arrives in RTT seconds to the destination, D because waiting too long before transmitting a lost packet may cause throughput of the flow degrade as it would prevent the congestion window to proceed.

3. [**3 points**]:

Which of the following is true of the layering model?

Circle all that apply. (There may be zero, one, or multiple answers.)

A The network layer relies on the connection state maintained by the link layer in order to keep track of who it is communicating with.

B A consequence of layering is that a layer at the source host communicates with its peer layer at the destination, without concerning itself with the implementation details of the layers above and below.

C The layering model ensures that all packets with the same source and destination go through the same set of links and switches.

D The layering model allows for a TCP segment to encapsulate a payload that contains another TCP segment.

E None of the above.

4. [**3 points**]:

Suppose you send a web request (HTTP) over TCP over IP over WiFi. Which of the following is true?

Circle all that apply. (There may be zero, one, or multiple answers.)

A The first byte of the IP payload is a WiFi frame header.

B If you send the web request again, the exact same packet will be sent.

C The IP header contains the address of the next router.

D The checksum in the IP header guarantees that the IP addresses definitely haven't been changed.

E None of the above.

Answer:

None of the above is correct. The first byte of the IP payload is a TCP header. The link layer may have changed when you resend the web request. The Wifi frame contains information about which hop to send the packet to next. The checksum over the IP header doesn't guarantee that the IP header hasn't been changed, only decreases the probability.

5. **[3 points]:**

Which of the following routing methods require a router to know the network topology in order to route packets?

Circle all that apply. (There may be zero, one, or multiple answers.)

A Flooding

B Source routing

C Distributed routing with the Bellman-Ford algorithm

D Distributed routing with Dijkstra's algorithm

E Path-vector routing (e.g. BGP)

Answer:

Source routing and distributed Dijkstra-based routing such as OSPF both use aggregated global knowledge of the network's link state to compute low-cost paths.

Flooding requires no knowledge topology at all. Distance vector algorithms like Bellman Ford do route based on network topology, but they determine routes implicitly based on local, slowly-propagating information. BGP advertises explicit routes, but does not give routers global knowledge of topology; a given router may advertise or not advertise any paths it likes.

6. **[3 points]:**

Which of the following statements are true about routing?

Circle all that apply. (There may be zero, one, or multiple answers.)

- **A** Tier 1 networks must use BGP-4 for internal routing, but other Autonomous Systems are free to chose their own internal routing schemes.
- **B** If a router or link fails, distance-vector protocols will potentially converge again to the correct solution more slowly than link-state protocols.
- **C** BGP-4 allows Autonomous Systems to choose to always use the shortest path to a given destination if they want to, though they may choose not to.
- **D** OSPF is a link-state protocol: each router has a partial or complete view of the link topology and calculates routes based on this graph.
- **E** RIP is a link-state protocol: each router has a partial or complete view of the link topology and calculates routes based on this graph.
- **F** If an application uses source routing, then it must know the topology in advance AND must use the shortest path to reach the destination.

Answer:

Tier 1 networks can still choose their own internal routing schemes; BGP-4 is only required externally. Distance vector protocols are subject to the "bad news travels slowly" phenomenon. A neighboring AS may simply not advertise the shortest path to a given destination, in which case the origin AS has no way of knowing about it; and anyway, it only advertizes the AS path, not the actual hops between router taken insie a ISP transit AS's. OSPF is a link-state protocol, but RIP is a distance-vector protocol. Source routing does not require that the source use the shortest path.

II Stop-and-Wait vs. Sliding-Window

7. [10 points]:

(Note added at 12.30pm PDT for clarification: The stop-and-wait protocol just means that the cwnd is always set to have 1 packet outstanding.)

Suppose we have a network connecting two hosts with an RTT of 100 ms and a bottleneck link rate of 30 Mb/s. Suppose that one host is trying to send a message to the other (i.e., the communication is in one direction). In this question, you will consider the effectiveness of a stop-and-wait protocol versus a sliding-window protocol. Assume that packets are 1500 bytes and that there is no queueing delay in the routers.

a. What is the maximum throughput between these two hosts (in Mb/s)?

Answer:

The maximum throughput is governed by the bottleneck link: $\boxed{30 \text{ Mb/s}}$.

b. In a standard 1-packet stop-and-wait protocol, what percent of the maximum throughput will be used?

Answer:

One packet is transferred every RTT. Thus, the total throughput is 1500 *bytes*$/100$ *ms* $= 0.12$ *Mb/s. This is* $\boxed{0.4\%}$ *of the maximum throughput.*

c. Now suppose we use a sliding-window approach where we always have n oustanding packets at a time. What percent of the maximum throughput will be used (in terms of n)?

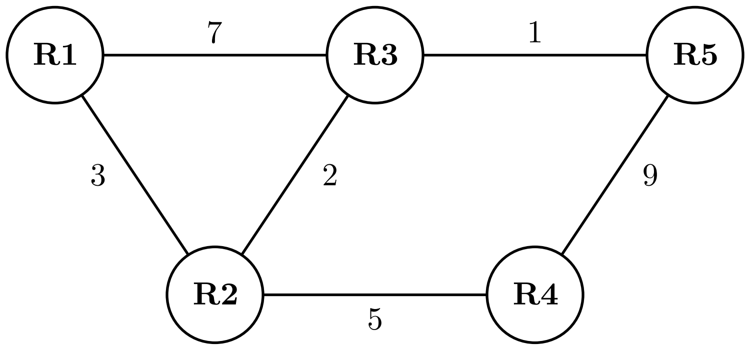
Answer:

In one RTT, we receive $1500n$ *bytes. The computation is the same as the previous question, yielding* $\boxed{0.4n\%}$ *of the maximum throughput.*

III Bellman–Ford

8. [15 points]:

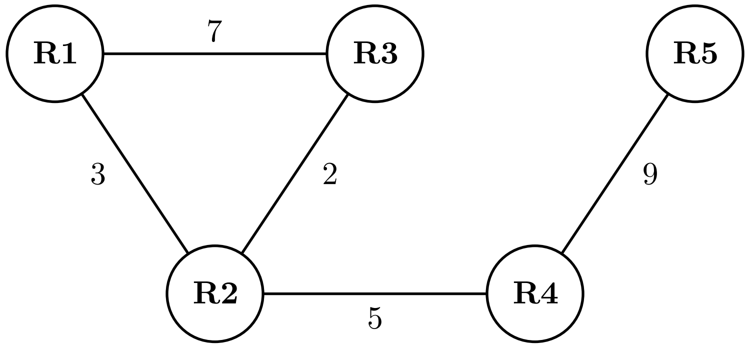
The figure below shows a network of five routers. Each link has a cost. For example, the link from R1 to R2 has a cost of 3.



a. Use the Bellman-Ford algorithm to complete the table below. A row in the table corresponds to the distance vector to R5 at an iteration of the algorithm. Each cell in the table (element of a distance vector) consists of a router's lowest known cost to R5 as well as the next hop in the lowest-cost path, if applicable. The first two rows are completed for you. Stop filling in the table when further iterations do not change the distance vector. You may not need all rows. Ties are broken randomly.

Iteration	R1	R2	R3	R4
0	∞	∞	∞	∞
1	∞	∞	*1, R5*	*9, R5*
2	*8, R3*	*3, R3*	*1, R5*	*9, R5*
3	*6, R2*	*3, R3*	*1, R5*	*8, R2*
4				
5				

b. Now assume that the link between R3 and R5 fails, as in the figure below.

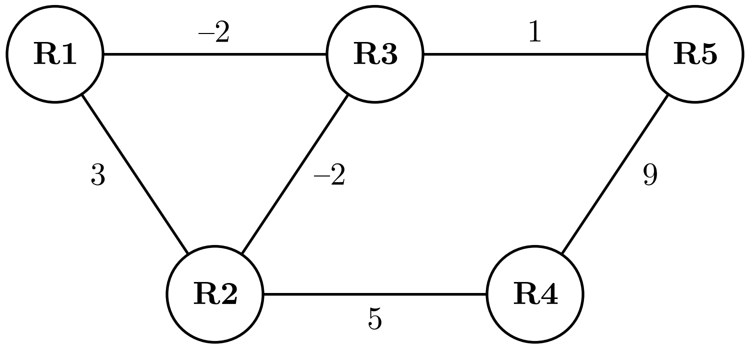


Using your final distance vector from part (a) as the 0^{th} iteration, how many iterations does it take for the Bellman-Ford algorithm to converge on a new distance vector (not including the 0^{th} iteration)? Assume that **no optimizations are used to improve the algorithm**. We will only grade your final answer, but showing work in the table below will make you eligible for partial credit.

Iteration	**R1**	**R2**	**R3**	**R4**
0	*6, R2*	*3, R3*	*1, R5*	*8, R2*
1	*6, R2*	*3, R3*	*5, R2*	*8, R2*
2	*6, R2*	*7, R3*	*5, R2*	*8, R2*
3	*10, R2*	*7, R3*	*9, R2*	*9, R5*
4	*10, R2*	*11, R3*	*9, R2*	*9, R5*
5	*14, R2*	*11, R3*	*13, R2*	*9, R5*
6	*14, R2*	*14, R4*	*13, R2*	*9, R5*
7	*17, R2*	*14, R4*	*16, R2*	*9, R5*
8				

Number of Iterations: ___*7*___

c. In lecture, someone asked if networks can use link cost metrics that include negative costs for some links. Consider a modification to the network from part (a) to use negative link costs, shown below.



In no more than 30 words, what is the problem with finding minimum-cost paths in this network?

Answer:

There are no minimum-cost paths, since any path can be made to have arbitrarily small cost by repeatedly cycling around R1, R2, and R3.

IV AIAD Congestion Control

9. [15 points]:

A software engineer is asked to implement the TCP congestion control mechanism for a company that has **many** customer flows. However, instead of an AIMD mechanism, the engineer mistakenly implements an AIAD (Additive Increase Additive Decrease) logic which increases the congestion window (*cwnd*) by 1 each time a full window of packets has been correctly acknowledgement, but decreases the *cwnd* by subtracting a given constant D every time a packet is lost.

The next day, a customer calls to complain about the low throughput their flow achieves when connecting to the company. The customer reports that their flow has a loss rate p. When the engineer tries to debug the case, they confirm that all the customer flows have a constant RTT of r seconds, and the network drops packets only when the bottleneck queue is full. Please help the engineer solve the problem.

Note: Throughout this question you should assume the window size has the same value, W, each time a packet is dropped. (You actually do not need this assumption to reach the correct result, but it makes the math much simpler.)

a. What would be the throughput of the customer flows if the engineer had successfully implemented AIMD (increase by 1, decrease by half) mechanism in terms of the loss rate p and the RTT r? Express your answer in packets per second.

Answer:

$$T = \sqrt{\frac{3}{2}} \times \frac{1}{\sqrt{p} \times r}$$

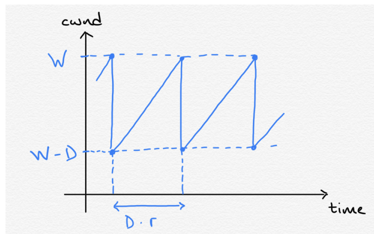
b. For an AIMD flow that has $p = 1.5\%$, RTT $= 10$msec, and a packet size of 10000 bits, calculate the throughput in Megabits per second.

Answer:

$$T = \sqrt{\frac{3}{2}} \times \frac{1}{\sqrt{1.5 \times 10^{-2}} \times 10 \times 10^{-3}} = 10^3 \mathit{packets/sec} = 10^3 \times 10^4 \mathit{bps} = 10 \mathit{Mb/s}$$

c. For the AIAD algorithm implemented by the engineer, draw the *cwnd*-vs-time graph for the customer flow in the steady state. Denote the maximum *cwnd* achieved as W. Show the time between two successive drops.

Answer:



d. What is the number of packets sent during the time between two successive drops in terms of the maximum *cwnd* (W) and D? (Assume that a drop is detected immediately by the sender.)

Answer:

The number of packets sent between two successive drops is the area under a period of $cwnd$*-vs-time curve.*

$$A = \frac{W + (W - D)}{2} \times D = \frac{2WD - D^2}{2}$$

e. What is the maximum *cwnd* (W) of the flow in terms of the loss rate (p per packet) and D?

Answer:

A packet is lost every A *packets.*

$$p = \frac{1}{A} = \frac{2}{2WD - D^2}$$

Simplify the equation above to find W*.*

$$W = \frac{1}{pD} + \frac{D}{2}$$

f. What is the throughput the customer flow achieves with the current implementation in terms of the loss rate p and the RTT r?

Answer:

$$T = \frac{A}{D \times r} = \frac{2W - D}{2r}$$

We can use the answer to the previous question to eliminate W in the expression.

$$T = \frac{2(\frac{1}{pD} + \frac{D}{2}) - D}{2r} = \frac{1}{p \times D \times r}$$

g. For the same network conditions in part (b), calculate the throughput of the customer flow, in Megabits per second, if $D = 9$ packets. Repeat the same calculation for $D = 1$ packet and comment on the importance of picking D.

Answer:

For $D = 9$:

$$T = \frac{1}{p \times D \times r} = \frac{1}{1.5 \times 10^{-2} \times 9 \times 10 \times 10^{-3}} = 740.7 \textit{packets/sec} = 7.4 \textit{Mb/s}$$

For $D = 1$:

$$T = \frac{1}{p \times D \times r} = \frac{1}{1.5 \times 10^{-2} \times 1 \times 10 \times 10^{-3}} = 6666.6 \textit{packets/sec} = 66 \textit{Mb/s}$$

Setting D a very large value causes the flow to back off too much, and the throughput decreases. On the other setting a very low D value makes the flow very aggressive against losses. Setting a "just right" decrease amount helps flows to utilize the network resources more efficiently and fairly. AIMD mechanism, by dynamically adapting the decrease amount to the flows current window size, allows a stable allocation of network resources.

V Traceroutes Again

10. [5 points]:

A few weeks after learning about the Internet and IP in their networking class, Kristen and Ewen decide to re-measure how long packets take to travel from San Francisco to New York. Kristen performs a traceroute from a computer in San Francisco to a computer located in New York and gets the following (fictional) output:

```
1  sf.example.com (208.64.252.229) 0.363 ms 0.427 ms 0.458 ms
2  b.example.com (171.60.53.13) 79.011 ms 78.988 ms 79.004 ms
3  c.example.com (171.64.1.12) 76.129 ms 76.198 ms 76.087 ms
4  d.example.com (171.64.255.144) 80.654 ms
   e.example.com (171.64.255.148) 80.605 ms
   f.example.com (171.64.255.149) 80.643 ms
5  newyork.example.com (171.67.76.65) 84.204 ms 84.200 ms 84.215 ms
```

a. What protocol does each router use to send a packet back to the source?

Answer:

ICMP

b. After looking at her traceroute, Kristen tells Ewen that there might be a link between d.example.com and newyork.example.com, but we can't be sure from the output. Is Kristen correct? Why or why not?

Answer:

Kristen is correct. The link may be between d.example.com, e.example.com, or f.example.com and newyork.example.com. We can't tell from the output.

c. Ewen notices that the RTTs reported in line 3 are shorter than the RTTs reported in line 2 and argues that it's impossible for a further hop to take a shorter time: Kristen must have performed the traceroute incorrectly. Is Ewen correct? Why or why not?

Answer:

Ewen is incorrect. Each line represents different packets with different TTLs, not the same packet across multiple hops. It's therefore possible that the packet with TTL 3 took a shorter time to reach its expiration than the packet with TTL 2.