Massachusetts Institute of Technology
6.829 Quiz (Fall 2019)

There are **12 questions** and **14 pages** in this quiz booklet. Answer each question according to the instructions given. You have **85 minutes** to answer the questions. The maximum score is 100.

If you find a question ambiguous, write down any assumptions you make. Explanations for answers may be given partial credit even if the answer is incorrect. **Be neat and legible.** If we can't understand your answer, we can't give you credit!

Use the empty sides of this booklet if you need scratch space. It will not be considered during grading.

Write your name in the space below and at the top of each page of this booklet. Please do this now.

You may use your two 2-sided "cheat sheet" and a calculator, but no other material.

Name:

MIT email/Kerberos ID:

Name: ______________________________

Congestion control

1. (4 pts) Two TCP flows share a bottleneck link of capacity $C = 1000$ packets per second. Flow 1 has a minimum RTT of 50ms, and flow 2 has a minimum RTT of 100 ms. Let W_1 and W_2 denote the congestion window size of flow 1 and flow 2 in packets, and let R_1 and R_2 denote rates of the flows in packets per second. Assume the bottleneck router has a large buffer and does not drop any packets.

Queuing delay

$$\frac{R_1}{R_2} = \frac{\frac{W_1}{RTT_1}}{\frac{W_2}{RTT_2}} = \frac{RTT_2}{RTT_1} = \frac{100ms + Q_d}{50ms + Q_d}$$

Circle the best choice in each case below.

- If $W_1 = 20$ and $W_2 = 20$, then:

$$\frac{W_1}{RTT_1} + \frac{W_2}{RTT_2} < C \Rightarrow Q_d = 0 \Rightarrow \frac{R_1}{R_2} = 2$$

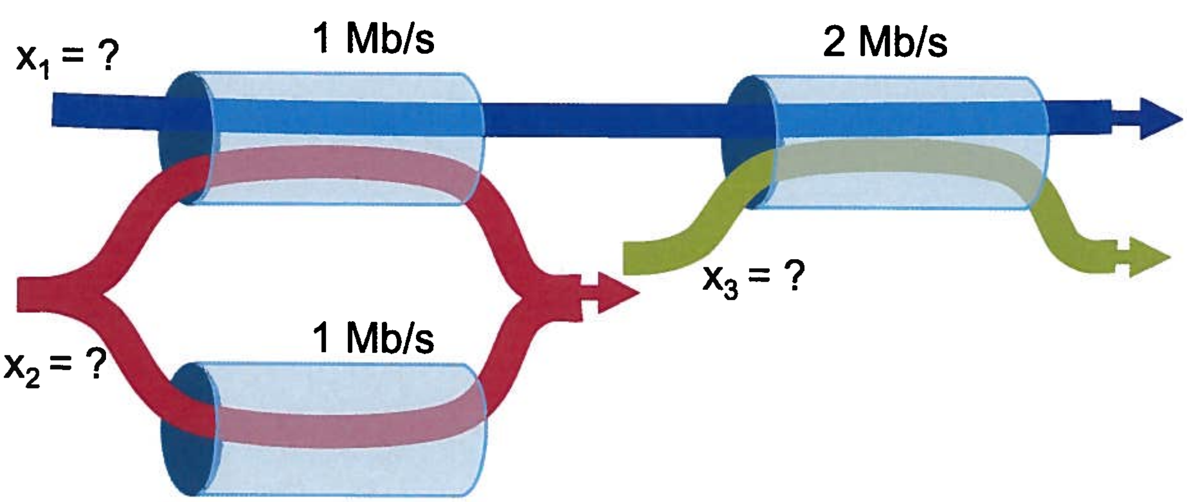
$R_1 / R_2 < 2$ $\qquad$ $R_1 / R_2 = 2$ $\qquad$ $R_1 / R_2 > 2$

- If $W_1 = 80$ and $W_2 = 80$, then:

$$\frac{W_1}{RTT_1} + \frac{W_2}{RTT_2} > C \Rightarrow Q_d > 0 \Rightarrow \frac{R_1}{R_2} < 2$$

$R_1 / R_2 < 2$ $\qquad$ $R_1 / R_2 = 2$ $\qquad$ $R_1 / R_2 > 2$

2. (2 + 2 + 4 = 8 pts) Consider the following topology with two links that are shared by three flows. Notice that the red flow's traffic is split across two links, and x_2 refers to the red flow's total rate on both links.



A. What allocation achieves max-min fairness?

x_1 = __1__, x_2 = __1__, x_3 = __1__.

Name: ______________________________

B. What allocation maximizes the aggregate throughput ($x_1 + x_2 + x_3$)?

x_1 = 0, x_2 = 2, x_3 = 2.

C. Which of these allocations is proportionally fair (i.e., maximizes $\log x_1 + \log x_2 + \log x_3$)? (Circle all that apply. Use the space below to show your work.)

i. $x_1 = 1/3, x_2 = 5/3, x_3 = 5/3$

ii. $x_1 = 1/2, x_2 = 3/2, x_3 = 3/2$

iii. $x_1 = 2/3, x_2 = 4/3, x_3 = 4/3$

iv. $x_1 = 1 - a, x_2 = 1 + a, x_3 = 1 + a, \forall a \in [1/3, 2/3]$

maximize $\log x_1 + \log(2-x_1) + \log(2-x_1)$

differentiate $\Rightarrow$ $\frac{1}{x_1} - \frac{2}{2-x_1} = 0$

$\Rightarrow x_1 = \frac{2}{3} \Rightarrow x_2 = x_3 = 2 - x_1 = \frac{4}{3}$

3. (4 + 6 = 10 pts) Ben Bitdiddle has come up with a new congestion control algorithm, B-TCP, which updates the sender's congestion window, CWND (in packets), as follows:

- On each ACK: CWND ← CWND + 0.01
- On each congestion event (loss or ECN, at most once per RTT): CWND ← 0.5 CWND

A. Circle **True** or **False** for each statement below.

i. **True** / **False** B-TCP is an additive-increase multiplicative-decrease (AIMD) congestion control algorithm. it is MIMD

ii. **True** / **False** For any network scenario (RTT, link rate, packet loss rate, etc.), B-TCP always achieves equal or higher throughput than Reno.

B-TCP increases CWND by 0.01 CWND per RTT. This is less than 1 pkt per RTT if CWND < 10[illegible]

B. Which of the following expressions is proportional to the average throughput of a B-TCP flow as a function of the packet loss rate, p? Show your work below. (Circle the best choice.)

i. $1/\sqrt{p}$

ii. $1/p$

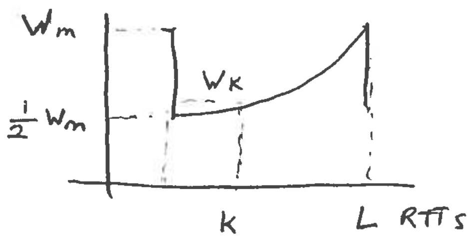
iii. $1/(p \log(1/p))$

iv. $\log(1/p)/p$

See next page.

Let $a = 0.01$

Name: ______________________



Window after k RTTs after drop $W_K = (1+a)^k \cdot \frac{1}{2} \cdot W_m$

$$W_L = W_m \Rightarrow \boxed{\frac{1}{2}(1+a)^L = 1} \quad (*)$$

totall # of sent pkts: $\sum_{k=0}^{L} W_K = \frac{1}{2} W_m \sum_{k=0}^{L} (1+a)^k = \frac{1}{2} W_m \frac{(1+a)^{L+1} - 1}{a}$

$$\sim \frac{W_m}{2a}(1+a)^{L+1} = \frac{W_m(1+a)}{a} \quad (\text{using } (*))$$

$$\Rightarrow \frac{W_m(1+a)}{a} = \frac{1}{p} \Rightarrow W_m \sim \frac{1}{p} \Rightarrow \underline{\text{throughput} \sim \frac{1}{p}}$$

Router-assisted Congestion Control

4. (8 pts) Let $p_{RED}(N)$ and $p_{PIE}(N)$ be the average drop probability when N long-running TCP Reno flows share a single RED or PIE bottleneck link of some speed C packets/sec. Assume that the link is fully utilized, the flows share the link equally, and the RTTs are the same with both schemes. Also assume that both schemes are tuned properly, so that they converge to a roughly constant drop probability.

Circle **True** or **False** for each statement.

- **True** / **False** The average queueing delay for RED increases as N increases.
- **True** / **False** For the same number of flows N, $p_{PIE}(N) < p_{RED}(N)$. $p_{PIE}(N) \simeq p_{RED}(N)$ (Consider TCP throughput equation.)
- **True** / **False** In PIE, the integral term in the drop probability calculation improves stability in the presence of feedback delay. Recall that the PIE algorithm updates the drop probability periodically as follows

$$p \leftarrow p + \underbrace{\alpha(q - q_{target})}_{\text{Integral term}} + \underbrace{\beta(q - q_{old})}_{\text{Difference term}}$$

Difference term improves stability.

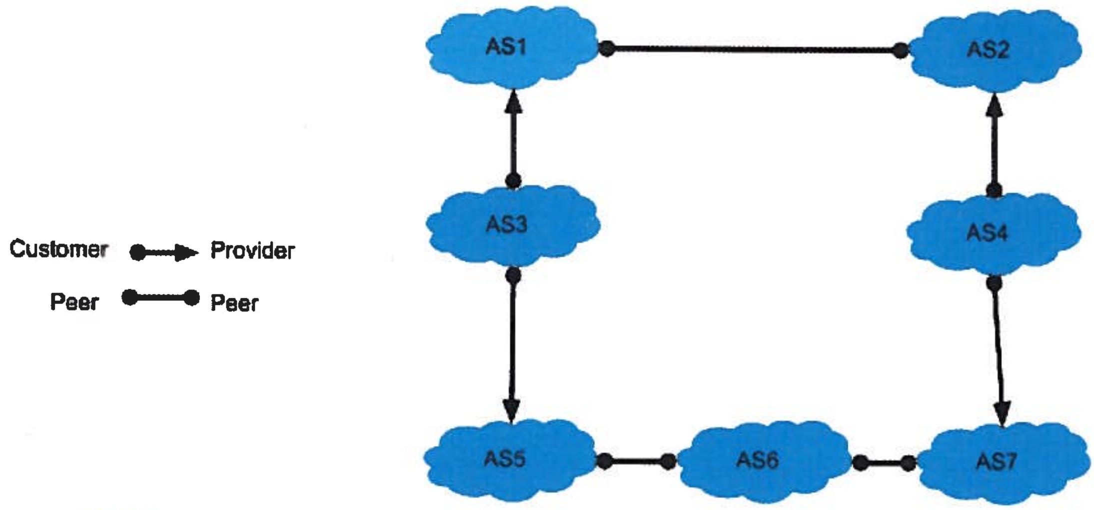
- **True** / **False** RED and PIE improve throughput compared to a DropTail queue.

RED/PIE improve queuing delay, not throughput

Name: ______________________________

Inter-domain routing

5. (8 pts) The network below is running the BGP routing protocol discussed in lecture. Circle **True** or **False** for each statement.



- **True** / **False** Autonomous System 3 (AS3) pays AS1 for access to its routes.
- **True** / **False** AS6 pays AS5 and AS7 for access to its routes.
- **True** / **False** AS4 has a route to AS3 through AS1.
- **True** / **False** AS5 receives a route announcement from AS6 that includes AS4.

Datacenter Networks

6. (8 pts) Circle **True** or **False** for each statement.

All packets of a flow are sent on the same path to avoid reordering. Different flows go on different paths

- **True** / **False** Equal-Cost-Multipath (ECMP) spreads packets of a flow evenly between equal-cost paths.

Probability of collision is lower with bigger buckets

- **True** / **False** ECMP balances load more uniformly across four 100 Gbit/sec links compared to forty 10 Gbit/sec links.

Otherwise flows can blast packets without restraint

- **True** / **False** VL2's performance isolation property relies on traffic implementing congestion control.

Location and application specific IP address

- **True** / **False** VL2 uses more than one IP address for each host in the network.

Name: ______________________________

7. (4 pts) You are tasked with designing a datacenter network for a fast-growing startup. You anticipate a maximum deployment of 40,000 servers for the datacenter, and you have 16-port switches, with each port operating at 10 Gbit/sec. You decide to design a Clos topology, which consists of some number of levels of switches. The switches in level *i* connect only to switches in levels *i-1* and *i* + 1, except for the switches in layer 1 (the top-of-rack layer), which connect to the servers and the level 2 switches. What is the fewest number of levels of switching required to provide 10 Gbit/sec of bandwidth to each server for any traffic matrix? (Show your work below.)

# of levels required = 5

To ensure there is enough capacity for all traffic patterns, we need the bandwidth in both directions to be equal, except for the top layer, which can use all ports.

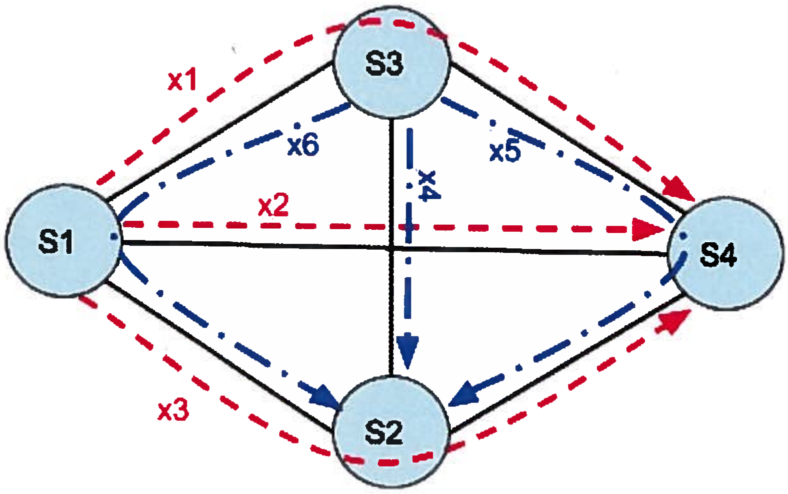
So we have $\underbrace{16}_{\text{top layer}} \cdot \underbrace{8^{n-1}}_{\text{other layers}} \geq 40000$

$\Rightarrow n \geq 5$

Name: ______________________________

Traffic Engineering

8. (8 pts) KIWI networks is a trending global cloud provider with a wide-area network shown below. Assume each link in this network has a capacity of 1Gbps in each direction. For the given traffic matrix, write the traffic engineering allocation on the paths shown that maximizes total throughput and provides fairness between the S1→ S4 and S3 → S2 traffic. The paths are marked by x1, x2, x3 for S1 to S4 traffic and x4, x5, x6 for S3 -> S2 traffic.



Traffic matrix:
S1 -> S4: 3 Gbps
S3 -> S2: 3 Gbps

x_1 = 1/2 Gbps

x_2 = 1 Gbps

x_3 = 1/2 Gbps

x_4 = 1 Gbps

x_5 = 1/2 Gbps

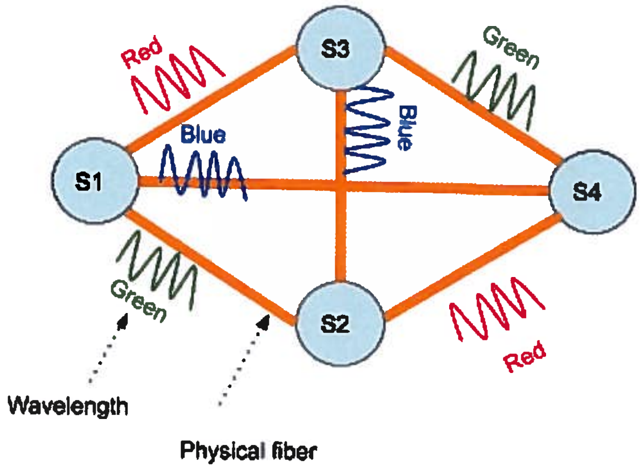
x_6 = 1/2 Gbps

Total throughput = ____________ Gbps

Name: ______________________________

Optical Networks

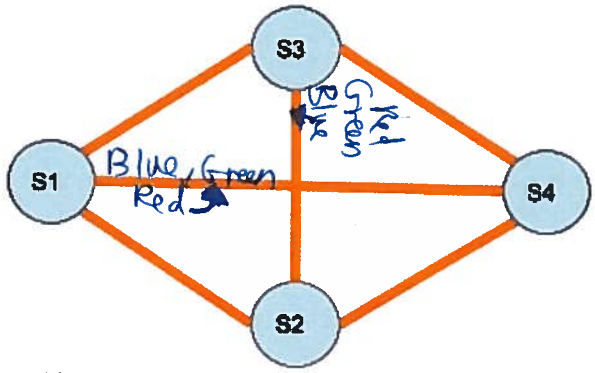
9. (7 pts) The previous question dealt with the IP layer traffic engineering of KIWI networks. The figure below shows the wavelength allocation for this network. Each node has three transceivers for transmitting/receiving Red, Blue, and Green wavelengths. Each wavelength can carry 1 Gbps of traffic, and we have assigned one wavelength to each fiber, as shown below:



Can you come up with a better assignment of wavelengths to fibers to achieve the optimal throughput for the given traffic matrix? Draw the wavelength assignment in the figure below, and provide its throughput. Assume that the traffic matrix never changes. Constraints of a valid wavelength assignment are:

- (a) each wavelength requires a transceiver on both sender and receiver nodes
- (b) each node has a total of 3 transceivers important to satisfy ~~this~~ all the constrain
- (c) each fiber can carry between 0 to 3 wavelengths
- (d) two transceivers tuned to the same wavelength cannot use the same fiber

We can put Three wavelengths on direct paths between $S_1 \to S_4$ and $S_3 \to S_2$



Traffic matrix:
S1 -> S4: 3 Gbps
S3 -> S2: 3 Gbps

Total throughput: ________ Gbps

Name: ______________________________

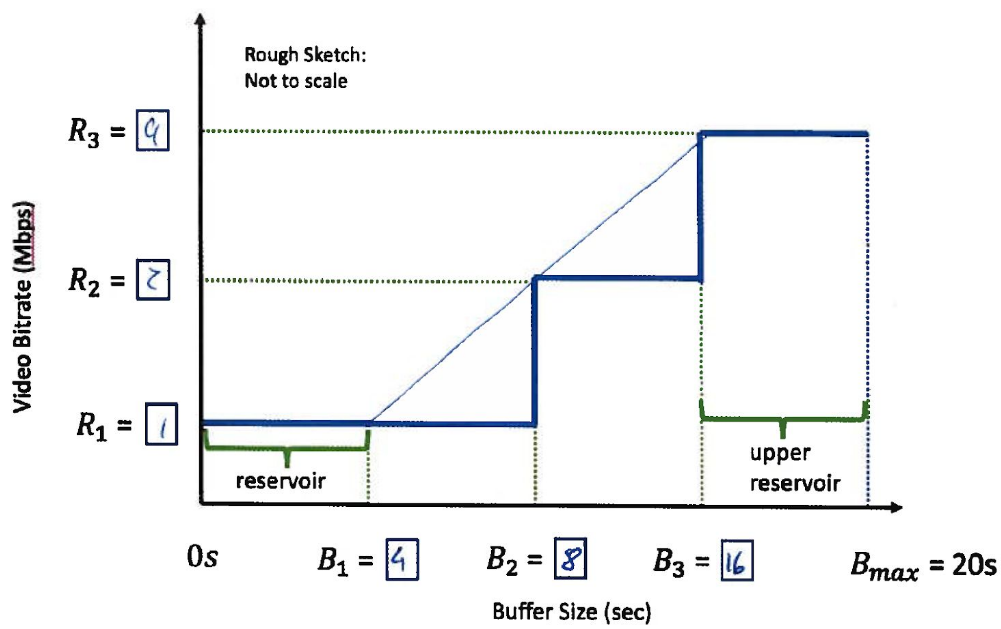
Video

10. (3 + 5 + 5 = 13 pts) Alyssa P. Hacker is using the buffer-based (BBA-0) algorithm she implemented in Lab 3 to watch BigBuckBunny (her favorite video) with DASH. At some time, Alyssa's throughput suddenly reduces to **1 Mbit/s**, and remains that way until the end of the video. At this time, Alyssa notices that a chunk had just been downloaded, giving her **12 seconds** of video in her playback buffer. Also, Alyssa notices that she only has 5 more video chunks left to view before the video ends.

    The available bitrates for this video are r_1 = 1 Mbit/s, r_2 = 2 Mbit/s, and r_3 = 4 Mbit/s. The buffer based algorithm has a reservoir of 4 seconds, an upper reservoir of 4 seconds, and a maximum buffer size of 20 seconds. The chunks are 4 seconds long, and all chunks for a given bitrate have the same size.

    To refresh your memory, a BBA-0 client always picks the lowest bitrate when its buffer occupancy is below the reservoir, and the highest bitrate when its buffer occupancy is less than the maximum buffer size by at most an amount equal to the upper reservoir. (See figure below).

    A. The following figure shows a rough sketch of the bitrate decisions made by BBA-0 as a function f of the buffer size. Label the three available rates R_1, R_2, R_3 on the y-axis and the corresponding buffer sizes B_1, B_2, B_3 (in seconds) on the x-axis, such that $f(b_1) = r_1$, $f(b_2) = r_2$, $f(b_3) = r_3$.



Name: ____________________________

B. What are the bitrates of the next 5 chunks fetched by BBA-0, in order?

Notation:

Bi = Buffer used for bitrate calculation by BBA-0 for chunk

ti = Download time of chunk, r(Bi) = Bitrate of chunk

Solution:

B0 = 12s ⇒ r(B0) = 2Mbps, t0 = 8s

B1 = B0 - t0 + 4 = 8s ⇒ r(B1) = 2Mbps, t1 = 8s

B2 = B1 - t1 + 4 = 4s ⇒ r(B2) = 1Mbps, t2 = 4s

B3 = B2 - t2 + 4 = 4s ⇒ r(B3) = 1Mbps, t3 = 4s

B4 = B3 - t3 + 4 = 4s ⇒ r(B4) = 1Mbps, t4 = 4s

C. After watching the video, Alyssa is disappointed with the QoE achieved by BBA-0. Meanwhile she has figured out a way to hack the network to get exact network throughput forecast into the future. Now Alyssa always knows what her network throughput is going to be in the future. Assume that Alyssa's QoE for the next *K* chunks is given by:

$$QoE = \begin{cases} \sum_{k=1}^{K} R_k, & \text{if no rebuffering on any chunk} \\ -\infty, & \text{otherwise} \end{cases}$$

where R_k stands for the bitrate of the k^{th} chunk. Note that Alyssa's QoE is linear in the bitrates of the chunks fetched, and she has zero tolerance for any rebuffering. Also there is **no** smoothness penalty.

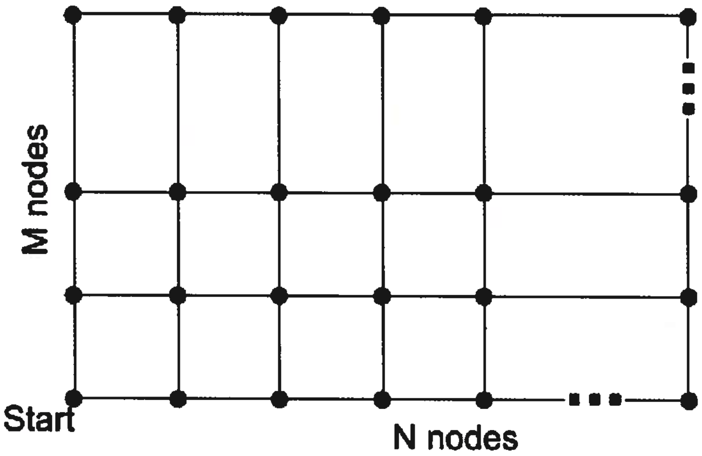
What is the **optimal choice** of bitrate decisions for the next 5 chunks in order to maximize her QoE? In case there are multiple optimal choices possible, you only need to provide one of them. Similar to the previous problem assume that **12 seconds** of video is available in the playback buffer at the beginning, and 5 chunks of video are remaining to be fetched at 1Mbps network throughput.

Note that because throughput is constant (= 1 Mbps) if r_0, r_1, r_2, r_3. is an optimal choice of bitrates, then any permutation of it should also achieve the same QoE (since there is no penalty for smoothness & rebuffering is not allowed) and hence should also ~~achieve~~ be optima[l]
Without loss of generality, let us assume $r_0 \geq r_1 \geq r_2 \geq r_3 \geq r_4$
Note that $r_0 = 4$ Mbps would lead to rebuffering. $r_0 = 1$ Mbps is clearly not optimal (see B.). Hence $r_0 = 2$ Mbps. Now, before fetching the next chunk, buffer size is $12 - 8 + 4 = 8s$. Now, $r_1 = 2$ Mbps is optimal. Similarly, $r_2 = r_3 = r_4 = 1$ Mbps would lead to no rebufferin[g]
Hence one optimal choice is 2, 2, 1, 1, 1
Any permutation of this will fetch full score.

Name: ______________________________

Blockchain relay network

11. (5 + 5 = 10 pts) Consider the network shown below.



A. The node marked as 'Start' mines a new block at time t=0, which is propagated through the network as follows. Each node that receives a block sends a hash of it to each of its neighbors. Each neighbor uses the hash to check if it already has that block, and if it does not, it requests the block and the node sends it to that neighbor. The one-way delay between any pair of connected nodes is d seconds, the link rate is C bytes/second and the block size is B bytes. Assume that there is no queuing delay in the network. How long does it take for all MN nodes in the network to get the block.

a. $d(M+N) + B/C$
b. $d(M-1)(N-1) + B/C$
c. $(M-1)(N-1)(2d + B/C)$
d. $\underbrace{(M+N-2)}_{\#\text{ rounds}}\underbrace{(3d + B/C)}_{\text{delay per round}}$

It would take M+N-2 rounds to reach all nodes
Each round takes 2d delay for hash check.
And d + B/c for block transmission, so ~~a~~ tot
of 3d + B/c delay.

B. Alyssa P. Hacker has come up with an improvement to this design, where a node that has received the hash of a block (but not necessary the block itself yet) immediately asks its neighbors if they want the block. The node starts sending the block to neighbors that request it as soon as it starts to receive it (even before obtaining the entire block). How long does it take to propagate the block to all MN nodes in this case? (For simplicity, assume an idealized implementation where a node can send each bit as soon as it receives it.)

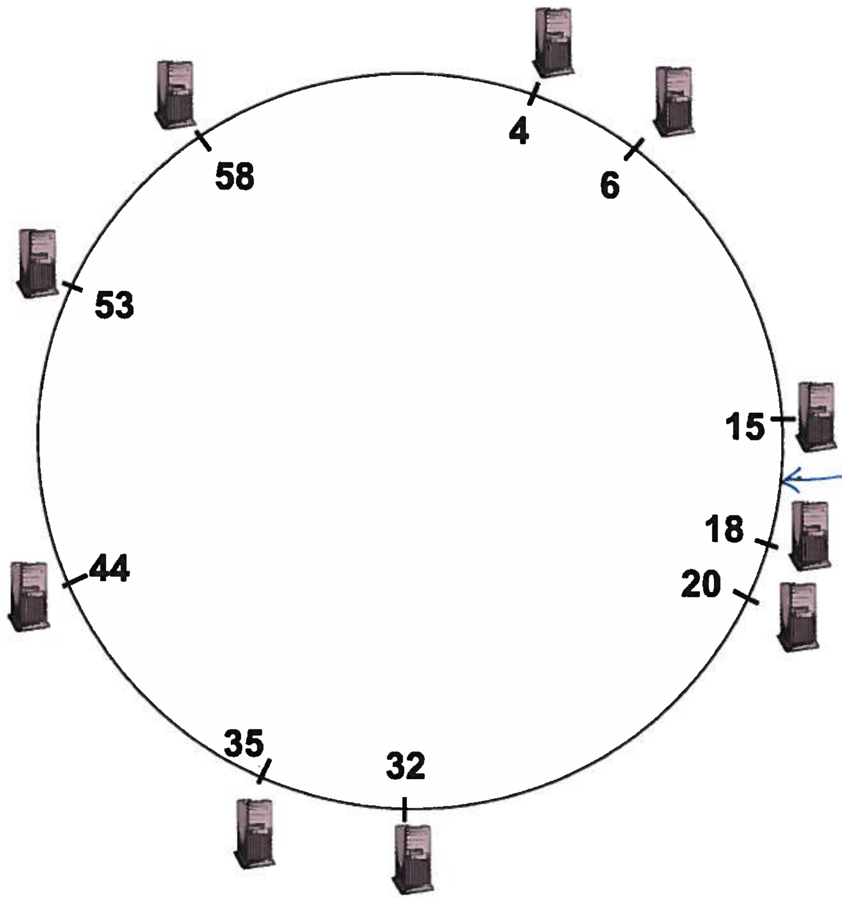
a. $d(M+N) + B/C$
b. $d(M-1)(N-1) + B/C$
c. $(M-1)(N-1)(2d + B/C)$
d. $(M+N-2)(3d + B/C)$

There is a delay of d(m+N-2) for hash to propagate to all nodes in the network, and a combined transmission delay of 2d + B/c, adding the total gives
d(m+N-2) + 2d + B/c

Name: ____________________________

Chord

12. (2 + 4 + 6 = 12 pts) Consider the following Chord network with 6-bit IDs for keys and nodes. Suppose each node has the correct successor and finger tables.



New entry

16

A. Suppose Node 16 wishes to join the network, and it knows about Node 20. What query (lookup) does Node 16 issue via Node 20 to join the network?

lookup (16)

B. What is the sequence of node IDs traversed to answer the query in part A, assuming that there is no caching of node or key information at any node?

Ans :- $\boxed{20 \rightarrow 53 \rightarrow 6 \rightarrow 15 \rightarrow 18}$

Lookup at 20 :- $(20 + 2^5) \% 64 = 52 \rightarrow$ Node 53

Lookup at 53 :- $(53 + 2^4) \% 64 = 5 \rightarrow$ Node 6

lookup at 6 :- $(6 + 8) \% 64 = 14 \rightarrow$ Noc 15

lookup at 15 :- $(15 + 2^2) \% 64 = 19 \rightarrow$ Node 18

has key 1

Name: ______________________

C. Once Node 16 has received the response to the query in part A, a sequence of nodes must invoke stabilization before Node 16 is correctly integrated into the network. Fill in the blanks below to show a sequence of two stabilization invocations that result in the correct successors and predecessors for all the nodes, and the successor and predecessors at each intermediate step.

After Node 16 joins and before any stabilization:

Node	Successor	Predecessor
15	18	6
16	18	NULL
18	20	15

After Node 16 calls stabilize():

Node	Successor	Predecessor
15	18	6
16	18	NULL
18	20	16

After Node 15 calls stabilize():

Node	Successor	Predecessor
15	16	6
16	18	15
18	20	16

For reference, the stabilization code is given on the next page.

```
// periodically verify n's immediate successor,
// and tell the successor about n.
n.stabilize()
    x = successor.predecessor;
    if (x ∈ (n, successor))
        successor = x;
    successor.notify(n);

// n' thinks it might be our predecessor.
n.notify(n')
    if (predecessor is nil or n' ∈ (predecessor, n))
        predecessor = n';
```