# SCHOOL OF COMPUTING

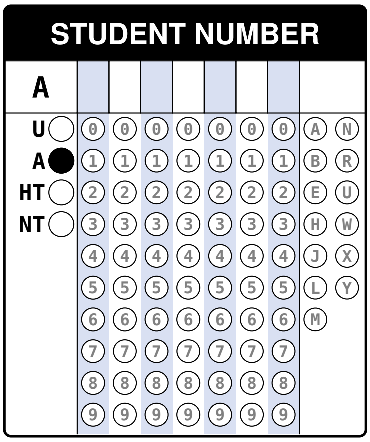
IN-CLASS ASSESSMENT FOR
Semester I, 2021/2022

**Solutions for CS5229 — Advanced Computer Networks**

Nov 2021 Time Allowed: 2 Hours

## Instructions (please read carefully):

1. This is a **<u>OPEN</u>** book assessment. You are allowed to have any materials with you.
2. You are not allowed to communicate with anyone during the exam.
3. Write down your **student number** on the right and using ink or pencil, shade the corresponding circle in the grid for each digit or letter. DO NOT WRITE YOUR NAME!
4. The assessment paper contains **SEVEN (7) questions** and comprises **NINETEEN (19) pages** including this cover page.
5. All questions must be answered in the space provided; no extra sheets will be accepted as answers. You may use the extra page behind this cover page if you need more space for your answers.
6. You are allowed to use pencils, ball-pens or fountain pens, as you like as long as it is legible (no red color, please).
7. **Marks may be deducted** for i) unrecognisable handwriting, and/or ii) poor shading.




| Question | Marks |
|---|---|
| Q1 | |
| Q2 | |
| Q3 | |
| Q4 | |
| Q5 | |
| Q6 | |
| Q7 | |
| **Total** | |

This page is intentionally left blank.

It may be used as scratch paper.

## Question 1: Short Questions [28 marks]

**A.** **[Warm-up]** Which of the following is the *biggest* advantage of the layered design for the Internet architecture in your view? [3 marks]

- ◯ It allows switches to process packets faster.
- ◯ It prevents designers from creating their own architectures.
- ◯ It allows changes to be made in one layer easily without affecting other layers.
- ◯ It makes it easy for changes to be made to network protocols.
- ◯ None of the above.

**B.** Explain your answer in Part (A) above. [3 marks]

It allows higher (and lower) layers to be modified without affecting the other layers as along as the interface between layers is not modified.

One might also be able to argue that layering allows us to separate concerns and hence makes network protocols easier to implement and debug.

This question is linked to Part (A) above. If the student is able to justify his/her decision well, full credit can be given even if it is not the same as what we expect. We are looking for understanding, not correct answers.

**C.** What in your view is the *biggest* disadvantage of the layered design for the Internet architecture? Explain. [3 marks]

A layered architecture will impose some overheads since headers needs to added or stripped when they move up or down the layers. It is quite clear that we can sometimes do better if we can coordinate across layers as demonstrated by the last lecture on Network-Distributed Systems Co-design.

**D.** **[ECN]** Explain why the ECN CE codepoint should not be set based on the instantaneous queue size. [3 marks]

The instantaneous queue size is likely to see considerable variations even when the router does not experience persistent congestion. We want to avoid spurious `cwnd` backoff. This question seeks to test that students understand what is congestion and why/how ECN works.

**E.** **[ECN Nonce]** Explain why there is an ECN nonce and how it works. [4 marks]

We want to allow the ECN sender to verify that the network elements are not erasing or modifying the CE codepoint probabilistically. Since we have two equivalent ECT codepoints, the sender can randomly choose one and verify that the bits echoed back by the receiver correspond to the version sent. This question seeks to test that students understand how the ECN nonce works.

**F.** **[RTT Unfairness]** Assume that we have 2 TCP Reno flows sharing the same bottleneck. Flow A has an RTT that is twice that of Flow B. In the steady state, which flow will have a higher throughput? Explain. [4 marks]

Flow B (lower RTT). In Slow Start, the lower RTT flow will ramp up faster. During congestion avoidance, the lower RTT flow will also increase its `cwnd` faster. This question seeks to test that students understand how TCP Reno works.

**G.** **[RTT Unfairness II]** Assume that we have 2 BBR flows sharing the same bottleneck. Flow A has an RTT that is twice that of Flow B. In the steady state, which flow will have a higher throughput? Explain. [4 marks]

Flow A (higher RTT). In the steady state, BBR will have a `cwnd` of $2 \times$ BDP packets. The higher RTT flow will then have more packets in the buffer on average and hence have a higher throughput. This question seeks to test that students understand that BBR keeps $2 \times$ BDP packets in flight and are able to reason about its implications.

**H.** You are watching a Netflix movie on your mobile phone on 4G. You had earlier recorded an interesting video of your pet cat and you decided to forward that video to your friend. Assume that your mobile phone effectively has a dedicated line (both uplink and downlink) to the 4G base station and does not have to share any bandwidth with other mobile users, what do you expect would be the impact of the video upload on your Netflix watching? (**Hint:** Netflix uses TCP Reno) [4 marks]

You are likely to experiencing buffering and delays while watching the Netflix movie. The main reason for this is that the ACKs generated by the Netflix client on the uplink will be competing with your video uploaded and be delayed. This is to test that the students understood TCP ACK-clocking and can deduce that interfering with the ACKs on the return path will negatively impact the forward path.

## Question 2: Congestion Control [28 marks]

**A.** Consider a `cwnd`-based congestion control scheme where the window size changes as follows during congestion avoidance:

$$cwnd \leftarrow cwnd + MSS/cwnd$$

and as follows when there is a packet loss:

$$cwnd \leftarrow \alpha \times cwnd, \text{where } 0 < \alpha < 1$$

Suppose we have just one connection with RTT $t$ running over the bottleneck link at rate $C$. Assuming all packet losses are caused by buffer overflows only, what is the smallest amount of buffer required at a switch for this scheme to achieve 100% utilization for a long-lived TCP flow? Explain. [6 marks]

We know that $w_{max} = B + Ct$, where $w_{max}$ is maximum `cwnd` and $B$ is the buffer size.
also, the minimum send rate is $\frac{\alpha w_{max}}{t} > C \Rightarrow w_{max} > \frac{Ct}{\alpha}$. So,

$$\begin{aligned} B &= w_{max} - Ct \\ &> \frac{Ct}{\alpha} - Ct \\ &> (\frac{1}{\alpha} - 1)Ct \end{aligned}$$

Basically, the bigger the drop in the `cwnd` when a packet loss happens, the bigger the buffer needs to be! This question tests if the student has fully understood buffer sizing. We actually discussed this in lecture albeit for $\alpha = \frac{1}{2}$. This question tests that students understood what was discussed in lecture and can generalize it.

**B.** You found yourself working for a major tech company that makes a web browser. Given that you have taken CS5229, your boss asks you for ideas on how you can make web browsing faster. One colleague suggests that instead of using just one TCP connection when accessing content on a webpage, you can modify the browser to use multiple TCP connections instead? Assuming that your browser traffic also competes with other flows at the bottleneck, will this idea work? Explain. [4 marks]

Assuming that the TCP variant used by the browser is the same as that in the bottleneck, then the sharing of the bandwidth would roughly be proportional to the number of flows. More parallel TCP flows means your browser will capture a larger share of the bandwidth.

**C.** Suppose that your web browser is using TCP Reno and is currently sharing a bottleneck link with capacity $C$ with one other TCP Reno flow. Next, you modified your browser to use $n$ TCP Reno flows instead? Assuming all the competing flows have the same RTT, how much bandwidth do you expect your browser to get on average? Explain. [4 marks]

We assume that the $n$+1 TCP flows will share the capacity $C$ fairly. This suggests that the browser will likely take up $\frac{n}{n+1}C$ of the bandwidth.

**D.** Consider the same scenario as Part (C) above, except, you modified your browser to use $n$ BBR flows instead? The bottleneck still has the other TCP Reno flow. How much bandwidth do you expect your browser to take up? Explain. [4 marks]

It depends. How BBR will interact with Reno will depend on the bottleneck buffer. There is not enough information to decide. This is to teach students that not every question necessarily has an answer. It is equally important to realize that some questions have no fixed answer and context matters.

**E.** **[Explicit congestion notification]** Your company realized that the performance of your browser is not very good for wireless links, so it developed a scheme that allows for the browser to detect the wireless network capacity at the end host and to provide a one-bit congestion notification signal (either FASTER or SLOWER) in the ECN bit in every ACK. The server now uses a new TCP variant where the window size changes as follows with each ACK received during congestion avoidance: (i) when the signal is FASTER:

$$cwnd \leftarrow cwnd + MSS$$

and as follows when the signal is SLOWER:

$$cwnd \leftarrow cwnd - MSS$$

Since the congestion signalling is very good, you can assume that there are no packet losses.

How do you expect this new variant to perform compared to TCP Reno? [4 marks]

This new version of TCP will indeed become more responsive to changing available bandwidths because it is effectively MIMD.

**F.** After running the new TCP variant described in Part (E) for some time, you discover a problem with it when it competes with other TCP flows. What is the problem? Suggest a possible fix. [6 marks]

The problem is that MIMD does not allow the flow to converge to fairness. The easiest solution if to add an additive component as follows. when the signal is FASTER:

$$cwnd \leftarrow cwnd + MSS + MSS/cwnd$$

and as follows when the signal is SLOWER:

$$cwnd \leftarrow cwnd - MSS + MSS/cwnd$$

Just modifying the solution to make it AIMD would also work: That is, when the signal is FASTER:

$$cwnd \leftarrow cwnd + MSS/cwnd$$

However then the scheme will lose the benefits mentioned in the previous question. This question is meant to test that students understood how AIMD affects efficiency and fairness.

3 points for highlighting that MIMD does not give fairness and 3 points for the fix.

## Question 3: Max-Min Fairness [8 marks]

In this question, we investigate how rate allocations and network utilization can change depending on the notion of fairness adopted by the network by considering the network with 4 flows shown in Figure 1. The numbers in the figure indicate the bandwidth of the various links in Mb/s. Note that the network links are bi-directional (full duplex).

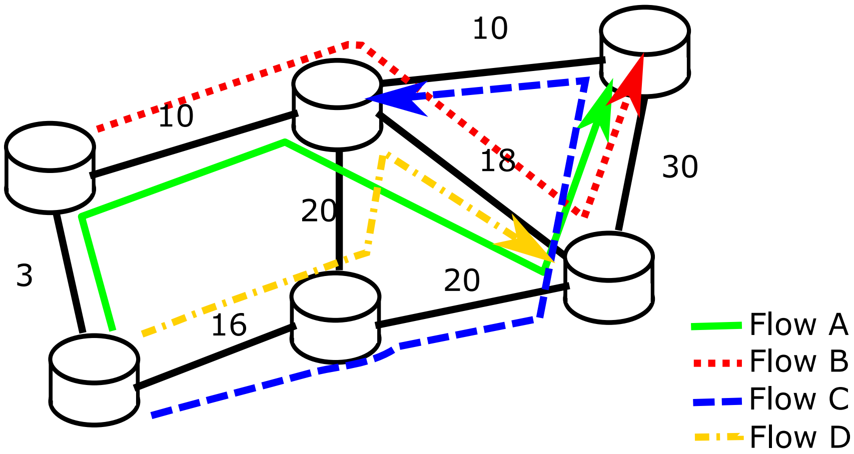



Figure 1: Network.

We consider two types of rate allocations as follows :

1. **Fair Rate:** With this rate allocation, each switch independently determines the number of flows sharing the bottleneck link and limits each flow to a rate no more than $C/N$, where $C$ is the capacity of the bottleneck link and $N$ is the number of flows sharing the link.
2. **Max-Min Fair:** With this rate allocation, the switches coordinate with each other and perform rate allocation so as to maximize the minimum rate achievable by each flow in the network.

**A.** Using **fair rate** allocation, work out the final rate allocations for the 4 flows shown in Figure 1. [4 marks]

| Flow | Rate (Mb/s) |
|---|---|
| Flow A | 3 |
| Flow B | 5 |
| Flow C | 8 |
| Flow D | 6 |

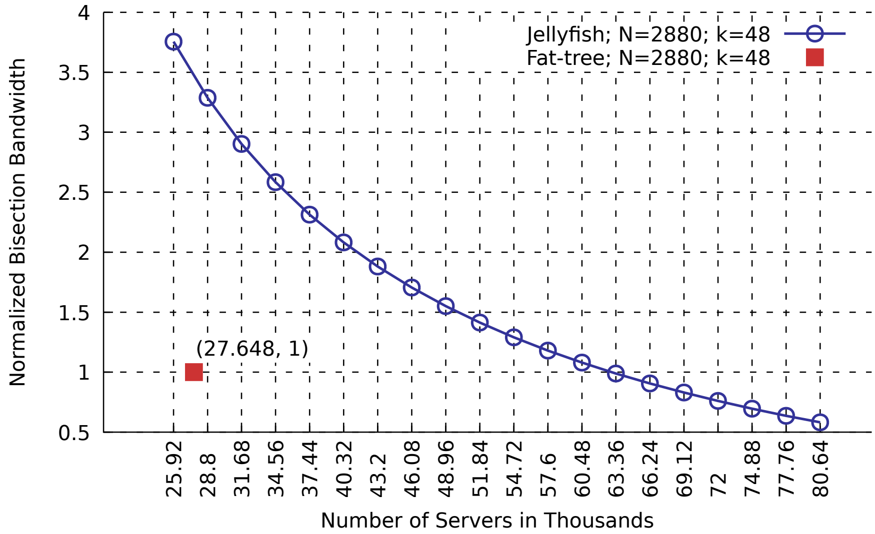
**B.** Work out the final rate allocations for the 4 flows for the network in Figure 1 with **max-min** allocation. [4 marks]

| **Flow** | **Rate (Mb/s)** |
|---|---|
| Flow A | 3 |
| Flow B | 7 |
| Flow C | 8 |
| Flow D | 8 |

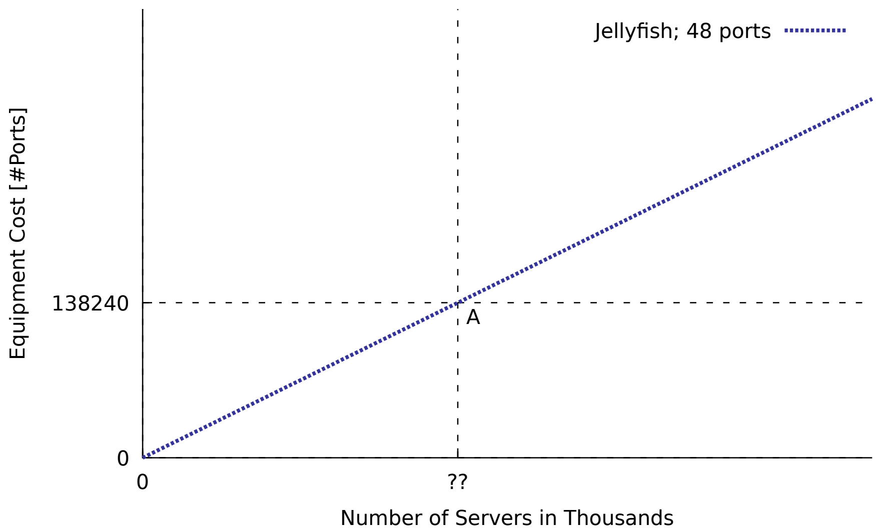
This question tests that student have fully understood the concepts of fair rate and max-min fairness discussed in class.

## Question 4: Jellyfish [13 marks]

Use the information from Figures 2a and 2b to answer the following questions. You would also need to know the basic formulations of the JellyFish network topology. If you do NOT have a calculator and find it too difficult to do the calculations manually, then you may leave the final answer as an expression (to be worked out by a calculator).




(a) Normalized bisection bandwidth versus the number of servers supported; equal-cost curves [partial plot from JellyFish Figure 2(a)]




(b) Equipment cost vs. the number of servers for commodity-switch 48 ports at full bisection bandwidth. Under optimal routing, with random-permutation traffic [incomplete JellyFish Figure 2a]

Figure 2: Partially reproduced graphs from the JellyFish paper.

**A.** For Figure 2b, find the expressions (formulae) for the X and Y coordinates of a JellyFish network with 48-port switches (dotted line plot). [3 marks]

The X axis represents the number of servers in thousands. For a JellyFish network topology, the total number of servers are given by $N \times (k-r)$, where $N$ is the number of switches, $k$ is the degree of the switch and $r$ is the number of ports of a switch used to connect other switches. Therefore, in this case, the expression for the X coordinate is, $X = \frac{N\times(k-r)}{1000} = \frac{N\times(48-r)}{1000}$.

The Y axis represents the total number of ports. The total number of ports is simply the number of switches times the degree of each switch. Therefore, the expression for the Y coordinate is, $Y = N \times k = N \times 48$.

**B.** For point A shown in Figure 2b, find the value of $N$, the total number of switches. [3 marks]

We know that $Y = N \times 48$. The Y coordinate for point A is 138,240. Therefore, the value of $N$ is given by, $N = \frac{Y}{48} = \frac{138240}{48} = \mathbf{2880}$.

**C.** For point A shown in Figure 2b, find the value of $r$ i.e. the number of ports of a switch used to connect to other switches. [4 marks]

Figure 2b represents the equipment cost vs the number of servers at full bisection bandwidth. For point A, we know that the value of $k$ is 48 and that of $N$ is 2880. For these exact same values, we do have a line plot in Figure 2a. The Y value of 1.0 in Figure 2a represents full bisection bandwidth. Therefore, a JellyFish network with $(N = 2880; k = 48)$ at full bisection bandwidth has 63.36 thousand servers. Now, the number of servers is given by $N \times (k-r)$. Therefore, $63360 = 2880 \times (48-r)$. Solving for $r$, we get $r = \mathbf{26}$.

**D.** For point A shown in Figure 2b, find the value of the X coordinate. [3 marks]

From subquestion (A), we know that the expression for Figure 2b's X coordinate is, $X = \frac{N \times (48 - r)}{1000}$. Substituting $N$ and $r$ derived in subquestions (B) and (C), we get $X = \frac{2880 \times (48 - 26)}{1000} = \mathbf{63.36}$.

OR

We know that the X coordinate of point A is the number of servers (in thousands) corresponding to a JellyFish network with $(N = 2880; k = 48)$ at full bisection bandwidth. Referring to Figure 2a, we can directly find the number of servers (in thousands) supported by a $(N = 2880; k = 48)$ JellyFish network at normalized bisection bandwidth of 1 (full bisection). These turn out to be **63.36**.

## Question 5: Geographic Routing with Bloom Trees [10 marks]

A Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. False positive matches are possible, but false negatives are not.

An empty Bloom filter is a bit array of $m$ bits, all set to 0. $k$ different hash functions defined, each of which maps or hashes some set element to one of the $m$ array positions, generating a uniform random distribution.

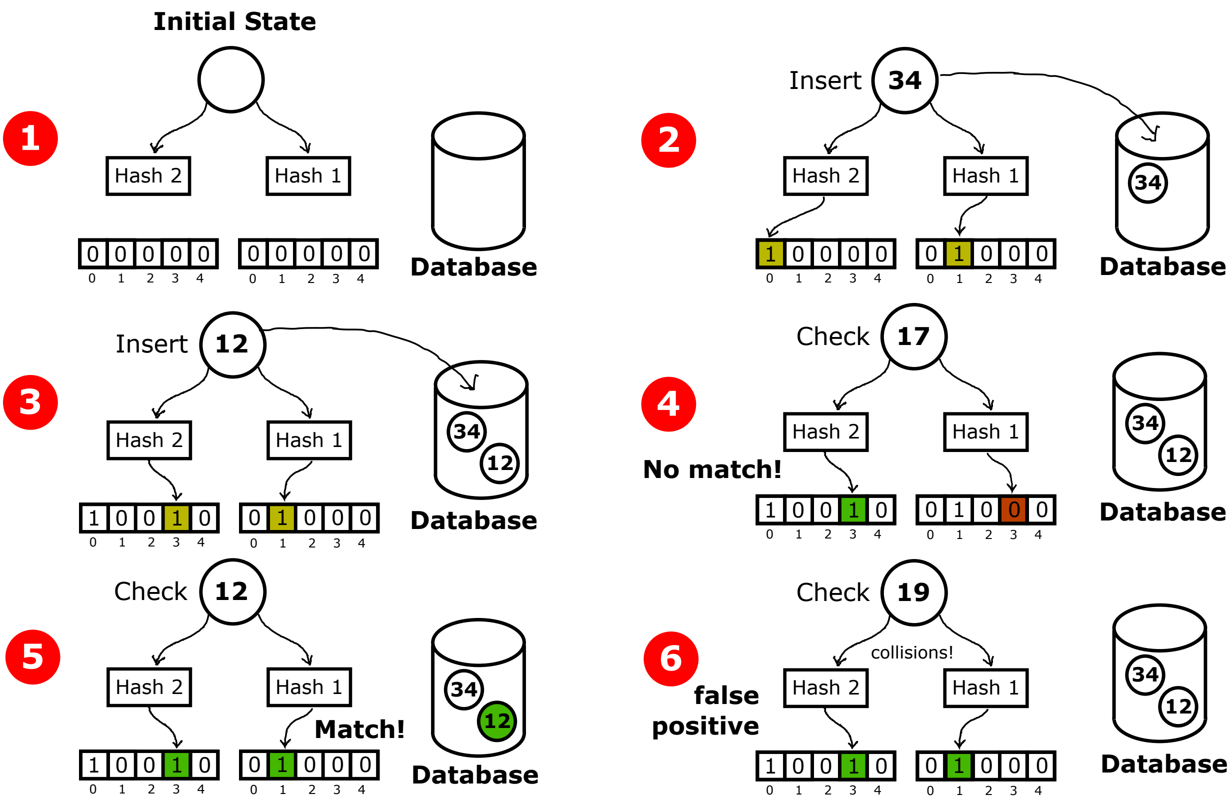



Figure 3: Bloom filter.

To add an element, feed it to each of the $k$ hash functions to get $k$ array positions. Set the bits at all these positions to 1.

To query for an element (test whether it is in the set), feed it to each of the $k$ hash functions to get $k$ array positions. If any of the bits at these positions is 0, the element is definitely not in the set; if all the bits are set to 1, then we assume that the element is in the set. It is possible to have false positives. This is illustrated in Figure 3.

After learning about GDSTR, you decided that it might be a good idea to use Bloom filters for geographic routing, instead of convex hulls.

**A.** Describe how you would use Bloom filters to maintain state in the nodes that can be used for geographic routing. [5 marks]

Instead of convex hulls, each node now maintains a Bloom filter. We build trees like GDSTR. All leaf nodes will have only one element their coordinate hashed into the Bloom filter. Each parent node will maintain a Bloom filter consisting of the OR of all the Bloom filters of its children and its own coordinate.

This question tests that students have fully understood how the convex hulls work in GDSTR.

**B.** Briefly describe how geographic routing can be done with these trees of Bloom filters. [5 marks]

We first try greedy forwarding. When a packet reaches a dead end, store its coordinates in the packet. Then we pick a tree and tranverse the tree. If the destination node is in any of the Bloom filters of any child, we will route to that child. If not, we forward the packet to the parent. This continues until the packet reaches the root. If the packet is deliverable, then the Bloom filter for one of the root's children should contain the destination. If not, then the packet is undeliverable. As we traverse the tree, we also check if we end up in a node that is nearer to the destination than the node where we started tree traversal. If so, we switch back to greedy forwarding.

This question tests that students have understood geographic routing and also how tree traversal works.

## Question 6: XOR-based DHT [10 marks]

In class, we learnt about how Chord is able to achieve $O(\log n)$ routes with its finger table. Today, we will consider a different (but similar) DHT, where like Chord, the nodes and objects have $k$-bit identifiers. The nodes in the system can be arranged into a binary tree from their prefixes (higher order bits of the identifier), as shown in Figure 4.

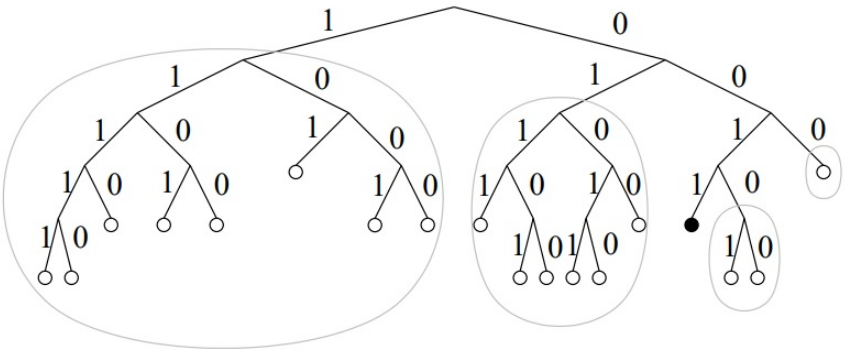


Figure 4: Binary tree for all the nodes in the system. For the node 0011*, there are 4 subtrees to which pointers need to be maintained.

Each node in the DHT, will maintain pointers (fingers) to up to $n$ other nodes with identifiers that are different from its own identifier in one bit position, starting from the highest-order bit. For example, in the example in Figure 4, the node 0011... will maintain pointers to one node (any node) in the 4 subtrees 1*, 01*, 000* and 0010*.

**A.** Given the above description, explain how we will do a lookup in this new DHT. [4 marks]

Given any destination, it must be in one of the subtrees corresponding to each maintained pointer. We simply either recursively or iteratively query the pointers.

This question tests that the students understand how DHT routing works.

**B.** Explain how we can be sure that the lookup algorithm you described in Part (A) above is guaranteed to find the destination. How many hops do we expect the lookup to take? [4 marks]

In this DHT, called Kademlia, we are effectively concerned about the XOR-distance (number of different bits) between the ids. In each step of the lookup, we will reduce the number of bit difference between the destination and the node queried. Since we are dealing with a binary tree, the number of steps would be bound by $O(\log n)$.

This question tests that the students can reason about what's essentially binary search in DHT routing.

**C.** In order to ensure that all the $n$ pointers are valid, a node will have to ping these nodes periodically to ensure that they are alive. Suggest how we might be able to do better and reduce the frequency of these pings. [2 marks]

We can potentially remember more than one node in each subtree and then use parallel queries to give us a better chance of contacting a live node. This question checks if the students remember what they learnt from EpiChord.

## Question 7: What did you learn? [3 marks]

What are the 3 most important lessons that you think you learnt in CS5229? Explain [3 marks]

There are no right answers here. Students will get credit for 3 well-explained learning points that are reasonable and justified. Clearly if the student makes a false statement, marks will be deducted.

**— E N D O F P A P E R —**

Scratch Paper

– H A P P Y H O L I D A Y S ! –