Instructions

1. Write your name and N number on top.
2. Provide concise and clear explanations for all your answers.
3. The questions are roughly in the order of the lectures.
4. Use the other side of each sheet if you need more space.

1. (3 points) Explain the 2-choice hashing algorithm for exact-match hash tables.
   
   **Answer:** In the 2-choice hashing algorithm, we use two independent hash functions to compute two separate indices into the hash table. Then, we look at both indices and find whichever one has lower occupancy. We insert a new element at this index. To lookup an element, we again hash the element using both hash functions to find 2 indices, and compare the element against the elements stored at each of the 2 indices.

2. (2 points) On what metric is the 2-choice algorithm better than the standard hashing algorithm? Why?
   
   **Answer:** The 2-choice algorithm is better on the metric of likelihood of overflowing the hash tables given a certain number of inserts into the hash table (equivalently, given a certain occupancy ratio). This is because the 2-choice algorithm explicitly favors the hash table location that has lower occupancy, which means that it is less likely to overflow an already full location in the hash table. The standard hashing algorithm, on the other hand, pays no attention to how occupied a location is.

3. (3 points) Explain the head-of-line blocking problem in input-queued routers using an example.
   
   **Answer:** Imagine an input-queued router with 5 inputs and 5 outputs with the following sequence of packets stored at each first-in first-out queue (FIFO) located at each input. The numbers here represent the output ports for each of these packets. At each input, we represent the FIFO by making the leftmost packet the tail of the FIFO and the rightmost packet the head of the FIFO.

   Input 1: 2, 3, 4, 5
   Input 2: 1, 2, 3, 5
   Input 3: 1, 1, 4, 2
   Input 4: empty
   Input 5: empty

   Here, both input 1 and input 2 have a packet for output port 5. Only one can be scheduled in a particular time slot. Let’s say it’s input 1. But then the packet with output port 3 on input port 2’s FIFO can not be scheduled until the first packet on input port 2 (i.e., the packet with output port 5) is scheduled. This is despite the fact that output port 3 has no packets scheduled for it during this time slot and can very well accept the packet from input port 2. Hence, the capacity of output port 3 is wasted. This is called head-of-line blocking because a subsequent packet (3) is blocked by the head-of-line packet (5). This situation can persist indefinitely if there are many inputs with packets that have the same output port (i.e., an incast scenario).
4. (3 points) How do you resolve the head-of-line blocking problem?
   **Answer:** The solution is to use virtual output queues (VOQs): separate queues for each output at each input. Applied to the previous example, this means we’ll have $5 \times 4 = 20$ VOQs instead of 5 FIFOs, where each VOQ corresponds to an input-output port pair (excluding the pairs where input and output ports are the same). With a VOQ-based system, a matching algorithm such as PIM can look at the head packets of each of the VOQs in making its decision, instead of just the head packets of each FIFO. This allows it, for instance, to schedule a packet from input 2 to output 3 in the same slot that a packet is scheduled from input 1 to output 5, thereby getting around the HoL problem.

5. (6 points) Consider a variant of ALOHA where nodes have different transmission probabilities. For simplicity, we’ll assume there are 2 nodes, A and B. A transmits with a probability $1/2$ in every slot. B transmits with a probability $1/3$ in every slot. In any given slot, (1) what is the probability that A successfully transmits a packet without collision? and (2) what is the probability that B successfully transmits a packet without collision? Show your calculations.
   **Answer:** The probability that A transmits is $1/2$. The probability that B does not transmit is $2/3$. We’ll assume that both of these are independent events. Then the probability that A transmits without collision is the probability that A transmits and B does not, which is $1/2 \times 2/3$, which is $1/3$. Similarly, the probability that B transmits without collision is the probability that B transmits, but A does not, which is $1/3 \times 1/2$, which is $1/6$.

6. (3 points) Explain the hidden terminal problem in wireless networks such as WiFi.
   **Answer:** A hidden terminal problem occurs when two sender nodes are out of carrier-sense range of each other, but within data communication range of another receiver node. As a result, both sender nodes end up transmitting to the same receiver node, causing a collision at the receiver in the process. An example is this arrangement: A —– B —— C. Here A and C are out of range of each other, but they can both communicate with B. Neither A nor C can sense the other, and both end up transmitting and colliding.

7. (3 points) Provide one solution to the hidden terminal problem.
   **Answer:** One solution is to use a centralized MAC protocol such as TDMA or FDMA, where a coordinator hands out time or frequency slots to the different transmitters to prevent them from colliding. If you were to use a decentralized MAC, such as WiFi, you could use the RTS/CTS exchange, where the two senders (A and C) send a request-to-send (RTS) to B, requesting B for permission to send data. B, in return, sends a clear-to-send (CTS) to one of A or C (depending on whose RTS reached it first). But this CTS is broadcast over the wireless network, so the other node can also hear it and back off for a time duration specified in the CTS.

8. (2 points) What causes bit errors at the physical layer of an Ethernet link or WiFi network? Give any two distinct causes.
   **Answer:** One cause is noise, which degrades the signal to noise ratio. Another is multipath, where signals reflected off different walls destructively interfere with each other, leading to a diminished signal strength relative to noise. A third is interference where multiple different senders transmit at once (like in the hidden terminal problem). Interference again leads to the signal strength being diminished relative to noise.

9. (2 points) How are such bit errors detected? Give an example of this detection algorithm at work.
   **Answer:** Checksums are the mechanism to detect bit errors, where a certain number of additional bits are added to a packet to check if the packet has suffered from bit flips in flight from the sender to the receiver. The simplest version of a checksum is a parity bit that is 0 if there is an odd number of 1s and 1 if there is an even number of 1s. The sender calculates the parity bit and sends it along with the data. The receiver checks if the parity bit that is received is consistent with the data that is received. For instance, if we had 7 data bits and 1 parity bit, the sender might transmit something like: 1100001, where the rightmost bit is 0 to reflect an odd number of 1s (3 1s) in the 7 data bits. If the receiver receives 11000011, the receiver knows something is wrong, because the data and parity bits are not consistent with each other. A parity bit can only detect one bit errors and can be fooled by a channel that can cause more than one bit errors in a packet.
10. (4 points) What quantitative performance metrics does a user care about when watching a streaming video? Why?

**Answer:**

1. Startup delay, the time before you can start playing *any* video.

2. Video quality, as measured by the average number of bits per second of video (the bit rate of the video stream). The higher the bit rate, the higher the quality.

3. The number of transitions in video quality over the course of a single video session.

4. The number of rebuffering events, where the video player stalls because it has no video to play.

11. (2 points) Consider a video streaming strategy where the user downloads the entire video at the highest quality before playing it. Give one advantage of this strategy. Give one disadvantage.

**Answer:** The advantage is high video quality. The disadvantage is high startup delay, especially for long and high quality videos.

12. (3 points) Why does a BitTorrent peer use the rarest piece first strategy when downloading different pieces of a file from other peers storing the file?

**Answer:** BitTorrent uses rarest piece first because the rarest piece has the greatest chance of vanishing because it stored by the smallest number of peers, who can leave at any point in time without warning. Rarest piece first ensures (1) selfishly, that a downloading peer gets access to the pieces that are most likely to disappear and (2) socially, that the rarest pieces get less rare over time because they are downloaded by more peers who themselves can share it with new peers.

13. (3 points) Compare and contrast BitTorrent and Bitcoin in terms of their networking and computational requirements.

**Answer:** BitTorrent is high on network capacity requirements because it transfers large files (e.g., movies, pirated software etc.) but low on computational requirements because the algorithms involved are quite simple (e.g., rarest piece first and tit-for-tat). Bitcoin is almost the exact opposite. It’s network capacity requirements are quite modest because each node just relays the blockchain, but the computational requirements are higher because many nodes (the heavyweight nodes and miners) need to either verify or certify transactions before relaying them.

14. (4 points) Say you have a switch with $k$ inputs and $k$ outputs. How can you build a network that supports the one-big-switch abstraction, i.e., the illusion of every server being connected to every other server through one big switch, using this $k$-port switch? How many servers can this network support as a function of $k$? You can restrict yourself to a 2-layer network with a layer of leaf switches and a layer of spine switches. Show your calculations.

**Answer:** The leaf layer is made up of $k$ port switches. Half of the ports on each leaf-layer switch are connected to the spine layer; the remaining half are connected to servers. To accommodate the $k/2$ ports on each of the $k$ leaf-layer switches, we have a spine layer with $k/2$ switches with $k$ ports each. This way each of the remaining ports on the leaf layer can be connected to a port on the spine layer. This arrangement supports $k^2/2$ servers, which is significantly more than the $k$ servers that can be supported by a $k$-port switch at large $k$.

15. (2 points) Consider the problem of transferring 100 PB of data from New York to California. How long will the data transfer take on a 1 Gbit/s transcontinental link? A PB (petabyte) is 1000 TB (terabyte), and a TB is 1000 GB (gigabyte). Also note that storage units are in bytes (PB, TB, and GB) while throughput is in bits/sec.

**Answer:**

$$\frac{8 \times 100 \times 10^{15}}{10^9} = 8 \times 10^8 \text{seconds} = 25 \text{years}.$$
16. (2 points) How long will it take to transfer 100 PB of data if you write the data to hard drives and then transfer these hard drives using a truck across the country? Let’s assume each hard drive has a capacity of 10 TB and can support a write throughput of 1 Gbit/s. You can assume you can write to the requisite number of hard drives in parallel. You can also assume that it takes 4 days to drive across the country and 2 days to load and unload the drives onto the trucks.

**Answer:** You need 10000 disks. If you write each disk in parallel, it takes about \( \frac{10^{12} \times 8}{10} \) = 80000 seconds or roughly a day to write the data. Add 4 days for the travel and 2 days for the loading/unloading, it gives you about a week.

17. (2 points) What is the data size at which it is faster to use a truck rather than the Internet for transferring the data?

**Answer:** The time for sending data using a truck is six days + however long it takes to transfer data onto the disks. So, it would have to take you six days to transfer data on the Internet before it might be faster to use a truck. The amount of data you can transfer in 6 days on a 1 Gbit/s link is \( 86400 \times 6 \times 10^9 \) bits, which is about 65 TB. So after this point, it may be worthwhile to use a truck.

18. (4 points) What security properties does TLS provide its users? Be as precise as possible with the answers.

**Answer:**

1. Confidentiality: Preventing an adversary from looking at the contents of the message.
2. Integrity: Preventing an adversary from tampering with the message.
3. Authentication: Guaranteeing that a sender is indeed talking to a specific receiver, such as www.google.com using certificates certified by certificate authorities.

19. (2 points) What security property does TLS not provide?

**Answer:** Privacy or anonymity: TLS does not hide who is talking to whom.

20. (4 points) In the context of network security, particularly in relation to surveillance and censorship, in what ways is a nation state adversary far more powerful than a garden-variety adversary?

**Answer:**

1. Legal provisions: Covert or overt laws that allow surveillance or allow the nation to compel companies into providing consumer data (e.g., call records). For instance, the PRISM program.
2. Money: The ability to build large-scale compute infrastructure to brute-force cryptography (e.g., the Logjam attack). The ability to enlist large teams to explore and exploit security vulnerabilities (e.g., Tailored Access Operations).
3. Physical access: The ability to tap into cables and other infrastructure (e.g., Tempora and MUSCULAR).

21. (5 points) Explain how censorship can be implemented using (1) DNS, (2) an unencrypted set of HTTP request headers, (3) the TCP layer packet headers, (4) BGP-based interdomain routing, and (5) the IP layer packet headers? We don’t need you to give actual instances of censorship. This is only a question on how censorship can be technically implemented at each layer of the Internet stack.

**Answer:**

1. DNS: Sending a DNS response for particular domain names that point to dummy servers or state-hosted servers.
2. HTTP headers: Censoring certain search queries on a search engine by looking at the HTTP headers.
3. TCP headers: Censoring SSH traffic on TCP port 22 to prevent SSH tunnels as a way to circumvent censorship.
4. BGP: Advertising a route to a black hole (e.g., the case of Pakistan Telecom and YouTube in 2008).
5. IP: Blocking specific IP address, such as known IP addresses of Tor relay nodes.