Quiz 2

You can ask the course staff any *specific* and *factual* questions you like *during* the quiz, and we will answer them! You are responsible only for reasoning, not knowing facts. Examples of factual questions include details of how an algorithm works, any formulas, and definitions of terms. Specific means that you need to know what question you may ask.

For essay questions, try to keep your answers as brief as possible while covering all the important points.

1 Essay 1 point

Frustrated by all the complexity in Media Access Control (MAC) protocols, Centra has decided to create a centralized node that computes schedules for which node should transmit at which frequencies and at what times for the next T seconds and broadcasts this schedule for everyone to obey. As they design the scheduling algorithm for such a centralized node, they find that it gets better throughput and latency than standard MAC protocols for some workloads and not for others, and are unable to see the pattern. Can you guess what that pattern might be?

For this question, you need not worry about fault tolerance or whether the broadcast reaches everyone. Think carefully about what the scheduler needs to know in order to work, and make all your assumptions explicit.

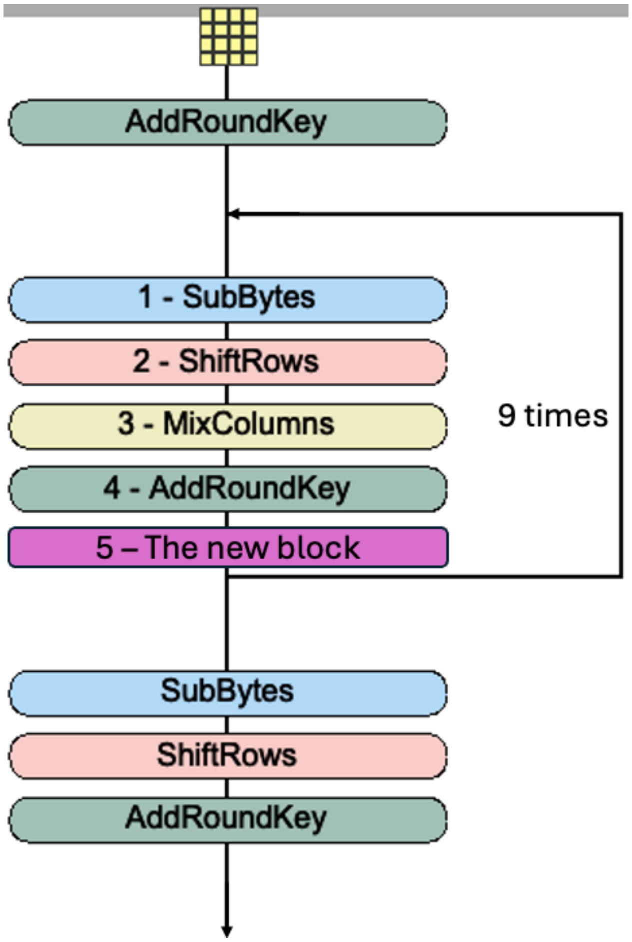
100%

5 min / 500 max (word limit) Must use at least 5 words

AES modification

Recall that AES encrypts each 128-bit

block of a message using the following steps (note, the precise details of what happens inside the existing steps is unimportant for this question):



Your friend has decided that they want to modify AES encryption (and correspondingly, decryption) by adding a fifth step to the part that is repeated 9 times (pictured as "the new block"). They have come up with a few options, some of which are deeply problematic. Can you figure out which ones and why?

Note: We are not asking you to analyze security here. That is too hard. Further, while there may be several non-security related problems, there is clearly one that is the most important. Your answer should reflect that.

2 Multiple Answer 1 point

Pick the ones that are deeply problematic

- [ ] Interpret it as a 128-bit unsigned integer x. Output x / 2, where "/" represents integer division
- [ ] XOR the 128-bit value with the randomly picked 128-bit constant 0xb4fe8ddaf8262d1d47e1ba277f75b8bd
- [ ] Interpret it as a 128-bit unsigned integer x. Output 2^128 - 1 - x
- [ ] OR the 128-bit value with the randomly picked 128-bit constant 0xb4fe8ddaf8262d1d47e1ba277f75b8bd
- [ ] AND the 128-bit value with the randomly picked 128-bit constant 0xb4fe8ddaf8262d1d47e1ba277f75b8bd

3 Essay 0.5 points

Briefly describe the problem you identified with some of the options above

1 min / 500 max (word limit)

Must use at least 1 word

4 Essay 1.5 points

Expanso believes it is a mistake to do Additive Increase, Multiplicative Decrease in congestion control. Instead they believe it is important to grab bandwidth fast and let it go slow. So they propose Multiplicative Increase, Additive Decrease (MIAD). Answer the following three questions about how it will perform:

1. What happens to the throughput when ours is the only flow in the network path?
2. What happens to the number of packets lost when ours is the only flow in the network path?
3. What happens to fairness when two or more flows running the MIAD algorithm share a network path?

You can make the following assumptions

- Applications always have enough data to send. We only care about steady state behavior. That is, the behavior long after the network has had time to settle
- Lost packets are retransmitted perfectly. That is, the sender knows exactly which packets are lost.
- When a network queue is overwhelmed, it drops at least one packet from all the flows. Otherwise it does not drop any packets.

For the sake of precision, the MIAD algorithm is shown below. The function is triggered every Round Trip Time (RTT).

```
if (a packet was lost in the last RTT):
    cwnd = cwnd - 1
else:
    cwnd = cwnd * 2
```

5 min / 500 max (word limit) Must use at least 5 words

5 Essay 0 points

This question is not graded and most students will not need to write anything here. But, if you feel you need to give context for any of your answers to any of your questions, you may use this space

1000 max (word limit)

Picking the transport abstraction

You are using QUIC as the transport protocol. It offers three interfaces to the application:

1. A traditional TCP-like mode that guarantees bytes will be delivered in exactly the order they were sent
2. A datagram mode: You can send at most 1200 bytes. It may or may not reach the other end. QUIC makes no attempt to retransmit the data if it is lost
3. Streams mode: the application can send multiple objects. The receiver gets one stream per object. Bytes within an object arrive in the same order in which they were sent

For each use-case on the right, which kind of interface will you use? The priority is to maximize performance. But among choices that maximize performance, pick the option that minimizes application developer effort (e.g. the developer does not want to write

6 Multiple Choice 0.5 points

You are designing a game server for playing chess and open a QUIC connection to transmit moves.

- 3
- 1
- 2

Clear my selection

7 Multiple Choice 0.5 points

You are designing a game server for a first-person shooter game. Here it is more important to get the *latest* user input than to get all user inputs.

- 2
- 3
- 1

Clear my selection

code for retransmission, concurrency etc unless required)

8 Multiple Choice 0.5 points

A software update downloader program that simply writes all the downloaded files to disk. You may assume the disk is much faster than the network.

- 2
- 3
- 1

Clear my selection

9 Multiple Choice 0.5 points

A software update downloader program that also needs to uncompress and verify files after writing them to disk. These operations may take time.

- 2
- 3
- 1

Clear my selection

10 Multiple Choice 1 point

Suppose you could only have one of the three modes for *all* possible applications and you wanted to maximize performance, even at the cost of developer effort. Which of these modes would you pick?

- 3
- 2
- 1

Clear my selection

11 Multiple Answer 1.5 points

Eve wants to eavesdrop on Alice's video call by eavesdropping on the TLS connection between Alice and zoom.us. The conversation is not recorded. To do so, she must compromise *something*. Which of these can lead to a successful attack?

The TLS connection uses RSA + Diffie Hellman key exchange + AES as discussed in class. In all cases, she can intercept the packets to try and mount a Man-in-the-Middle (MitM) attack.

- [ ] Knowing the secret key of LetsEncrypt, a well-known certificate authority, but not one that zoom.us uses
- [ ] Getting unrestricted access to Alice's computer several days *after* the conversation ends (and after she logged out)
- [ ] Knowing zoom.us's public key before the conversation starts
- [ ] Knowing zoom.us's secret key before the conversation starts
- [ ] Recording all the packets sent between the two and figuring out zoom.us's secret key after the conversation ends
- [ ] Getting unrestricted access to Alice's computer several days *before* the conversation starts (and before she had logged in)

Swiching

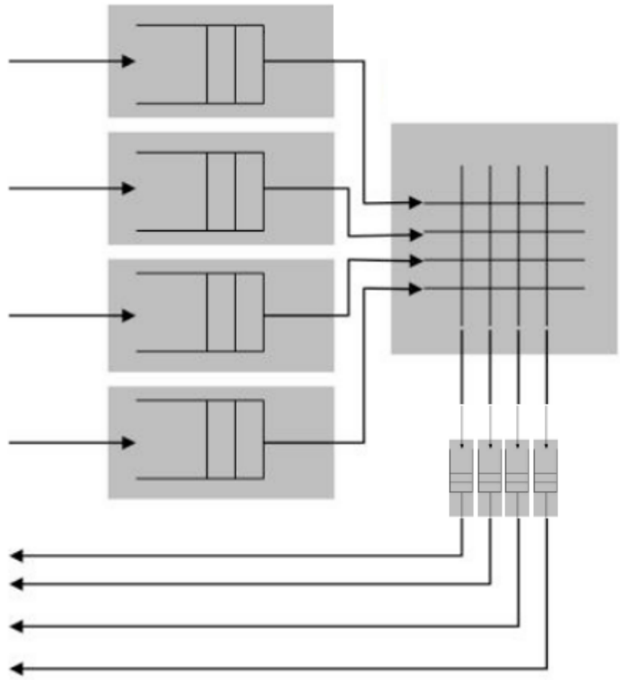
Recall that many switches are designed with **input ports** that feed into **input queues**, which in turn feed into a **crossbar** connecting to **output queues** and finally to **output ports**. Each physical port on a switch is divided into an input and an output port. Thus, an *N*-port switch has *N* input ports and *N* output ports. This structure is illustrated below. (Note: the output queues are shown smaller only due to the limitations of your instructor's diagramming skills.)

12 Numeric 0.5 points

Define **total bandwidth** as the total number of bytes per second leaving all output ports combined. It depends on the traffic pattern (i.e., which input ports send to which output ports).

What is the maximum total bandwidth achievable, over all traffic patterns, for a 16-port switch (in bytes per second)?

Answer



Time is divided into **slots.** In each slot, a **matching** is selected based on the demand between the input and output queues, and at most **100 bytes** can be sent along each edge in that matching. There are **1 million slots per second** (i.e., each slot is **1 μs** long).

Each input (and output) port can receive (and transmit) up to **100 bytes per slot.**

13 Numeric 0.5 points

Design a traffic pattern that results in the **minimum possible total bandwidth**, while ensuring that **every input port always has at least one packet waiting to be sent** (i.e., each input port is continuously backlogged, but the choice of destination port is up to you). What is the total bandwidth for such a pattern?

Answer

14 True or False 0.5 points

Consider traffic patterns that cause the most **queuing at the input ports.** Is it possible to make the number of packets waiting in the input queues grow **arbitrarily large** (assuming the queues have enough memory to hold them)?

- True
- False

15 True or False 0.5 points

What about output ports? Can you make their queues arbitrarily large?

- True
- False