

# CompSci 311 Quiz 1

## Q1

You are the network communications engineer in a team that is developing an industrial solution that uses four drones. The drones need to work synchronously. To that end, each drone communicates data on its position, speed, and height to each of the other three drones in synchronization 64B messages that carry 64B. An acknowledgement frame contains 12B. To guarantee the best synchronization possible, drones send synchronization messages as soon as they receive the acknowledgement for their previous message. Assume that the drones work at a distance of 300 meters one from the other, that the speed of light is  $3 \times 10^8$  m/s and that there are no collisions.

What is the minimum needed bandwidth to ensure that drones can send the next synchronization message right after receiving the acknowledgement of the previous message? Explain your reasoning and calculations.

## Answer

### 1. Calculate the propagation delay between drones.

Propagation delay = Distance / Speed of light = 300 meters / ( $3 \times 10^8$  m/s) =  $10^{-6}$  seconds (1 microsecond)

### 2. Calculate the time it takes to transmit the messages

Let R be the data rate in bits per second (bps).

For the 64B synchronization message:

- Time to transmit = (64 bytes \* 8 bits/byte) / R = 512 bits / R

For the 12B acknowledgement message:

- Time to transmit = (12 bytes \* 8 bits/byte) / R = 96 bits / R

### 3. Calculate the minimum bandwidth needed

Total time for the message exchange = 2 \* Propagation delay + Time to transmit sync message + Time to transmit acknowledgement

Total time for the message exchange =  $2 * 10^{-6}$  s + 512 bits / R + 96 bits / R

The minimum R that satisfies the equation is given below:

$$R = (512 \text{ bits} + 96 \text{ bits}) / (2 \times 10^{-6} \text{ s})$$

$$R = 608 \text{ bits} / (2 \times 10^{-6} \text{ s}) = 304 \text{ Mbps}$$

i.e., the minimum bandwidth needed to ensure that drones can send the next synchronization message right after receiving the acknowledgement of the previous message is 304 Mbps.

---

## Q2

Consider a link with high latency and very high bandwidth that uses large sending and receiving windows. Discuss how you would handle transmission errors that require retransmission. Assume that the error detection code is flawless. Focus on aspects related to the coordination of the sending and receive windows: When and what does the sender retransmit? How does the sender decide what frames need retransmission? What feedback does the receiver send to the sender? What is the impact of a transmission error?

## Answer

The sender can use Automatic Repeat Request (ARQ), which includes Stop-and-Wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ, to handle transmission errors.

### 1. When the sender retransmits:

When the sender does not receive a positive acknowledgement (ACK) for a frame within a specified time window (timeout), it retransmits the frame.

### 2. How does the sender decide what frames need retransmission:

The choice of retransmission protocol (Go-Back-N or Selective Repeat) determines which frames the sender retransmits.

- In Go-Back-N ARQ, the sender maintains a single copy of unacknowledged frames and resends all frames starting from the earliest unacknowledged frame when a timeout occurs.
- In Selective Repeat ARQ, the sender maintains individual timers for each unacknowledged frame and only retransmits the specific frame that triggered a timeout.

### 3. What feedback does the receiver send to the sender:

The receiver sends feedback to the sender in the form of positive acknowledgements (ACKs). ACKs indicate that a frame was received correctly. For example, the receiver can send a negative acknowledgement (NACK) to the sender to indicate that a packet was not received and needs to be retransmitted.

[ref:<https://www.red5pro.com/blog/6-ways-webrtc-solves-ultra-low-latency-streaming/>]

### 4. Impact of a transmission error:

The impact of a transmission error can be that TCP runs out of buffer space and the transfer has to stop until the retransmitted lost packet has been received<sup>2</sup>.

[ref:<https://www.noction.com/blog/network-latency-packet-loss>]

---

## Q3

Our NetMaze assignment assumes reliable message delivery, which significantly simplifies protocol design. Consider a simplified scenario where messages from the server to a client can be lost. In this case, the server might need to retransmit messages to the client. Describe how you would adapt the behavior of the server and client to account for lost messages. Assume that messages from clients to the server are always delivered. Assume that both the server and the client follow the protocol correctly: in other words, assume that there are no bugs and focus only on handling the lost messages. Hint: Clearly describe when the server sends retransmissions for lost messages.

## Answer

Use a retransmission mechanism:

### Client behavior:

1. After the client receives a message from the server, it processes the message and sends an acknowledgement (ACK) back to the server, including the message's unique identifier.

### Server behavior:

1. The server stores a copy of each message it sends to a client in a buffer along with a timestamp indicating when the message was sent.
2. The server starts a timer for each message it sends. If an ACK for a particular message is not received within a predetermined timeout period, the server retransmits the message to the client.
3. When the server receives an ACK from the client for a particular message, it removes the corresponding message from its buffer and cancels the timer for that message.

---

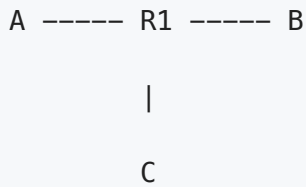
## Q4

Considering the exposed terminal problem in Wi-Fi networks:

- (a) Draw a diagram showing device placement and transmission (interference) range of each device.
- (b) Describe a sequence of events that leads to the exposed terminal problem in the context of the diagram in item (a) above.
- (c) Discuss the consequences of the exposed terminal problem.
- (d) Discuss whether RTS/CTS helps in dealing with the exposed terminal problem. Explain how RTS/CTS helps or why it cannot.

## Answer

(a)



There are four devices: A, B, R1, and C.

Device A and B are within the transmission range of R1.

R1 is within the transmission range of A, B, and C.

Device C is within the transmission range of R1 but not A or B.

(b)

- Device A wants to transmit data to R1. At the same time, R1 wants to transmit data to C.
- A senses the channel and finds it idle. It starts transmitting data to R1.
- R1 receives the transmission from A and starts transmitting to C.
- Device B also wants to transmit data to R1 but senses the channel busy due to A's ongoing transmission. Therefore, B defers its transmission, even though its transmission would not interfere with R1's transmission to C.
- B becomes an exposed terminal since it cannot transmit to R1, even though its transmission would not cause any interference at R1.

(c)

- The exposed terminal problem causes devices to defer their transmissions unnecessarily, leading to decreased network throughput.
- Waiting for the channel to become idle adds delays to the data transmission, increasing latency.

(d)

- When a device (e.g., A) wants to transmit data to another device (e.g., R1), it sends a short RTS packet.
- If the receiver (R1) is available to receive data, it sends a CTS packet back to the sender (A).
- All other devices in the network (e.g., B) that hear either the RTS or CTS packet will update their Network Allocation Vector (NAV) and defer their transmissions for the duration

specified in the RTS/CTS packets.

RTS/CTS can help alleviate the exposed terminal problem: When A sends an RTS to R1 and R1 responds with a CTS, B will overhear the CTS and update its NAV, deferring its transmission. However, C, which is not in the transmission range of A, will not hear the RTS from A. Therefore, when R1 sends its RTS to C, C will respond with a CTS, and R1 can transmit data to C without interference from A.

[ref:<https://www.tutorialspoint.com/the-exposed-terminal-problem>]