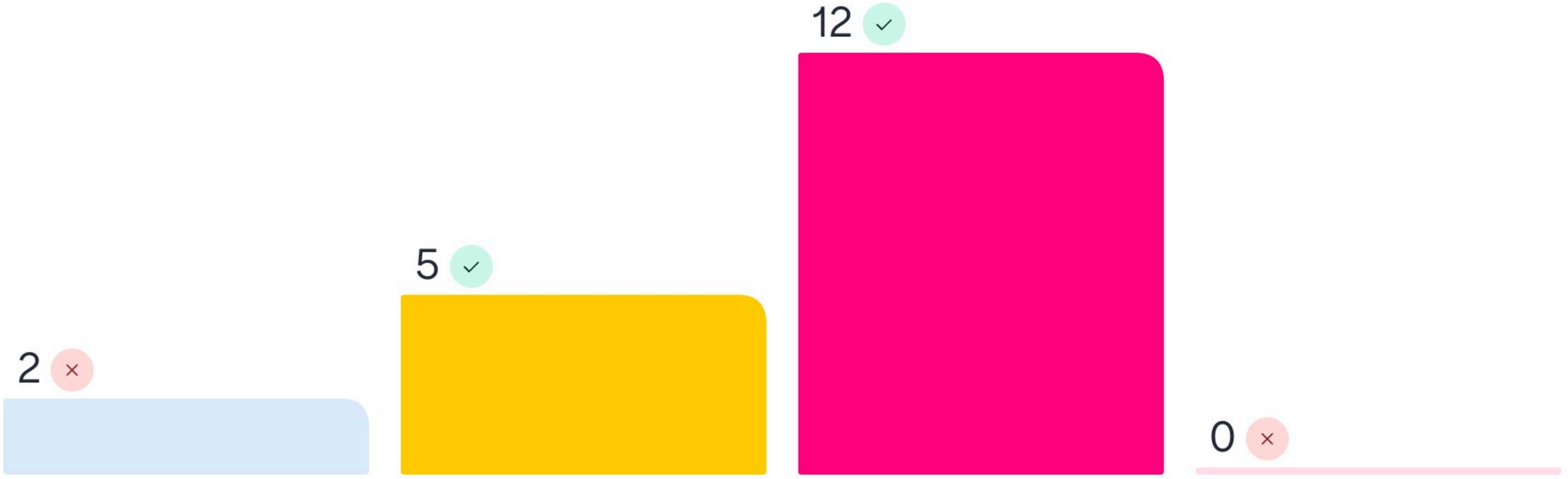
Introduction

- The goal is to network the classic arcade game Pong, making it a two-player online game
- Video 15 explores the challenges and solutions in designing a network protocol for a real-time game
- The focus is on understanding fundamental networking concepts through the lens of Pong
- The initial question is where to model the game: on one player's computer, a separate server, or distributed
- Using a central server also presents fairness issues if players have different network connections or distances to the server

What are the main reasons for choosing a distributed model for networked Pong?



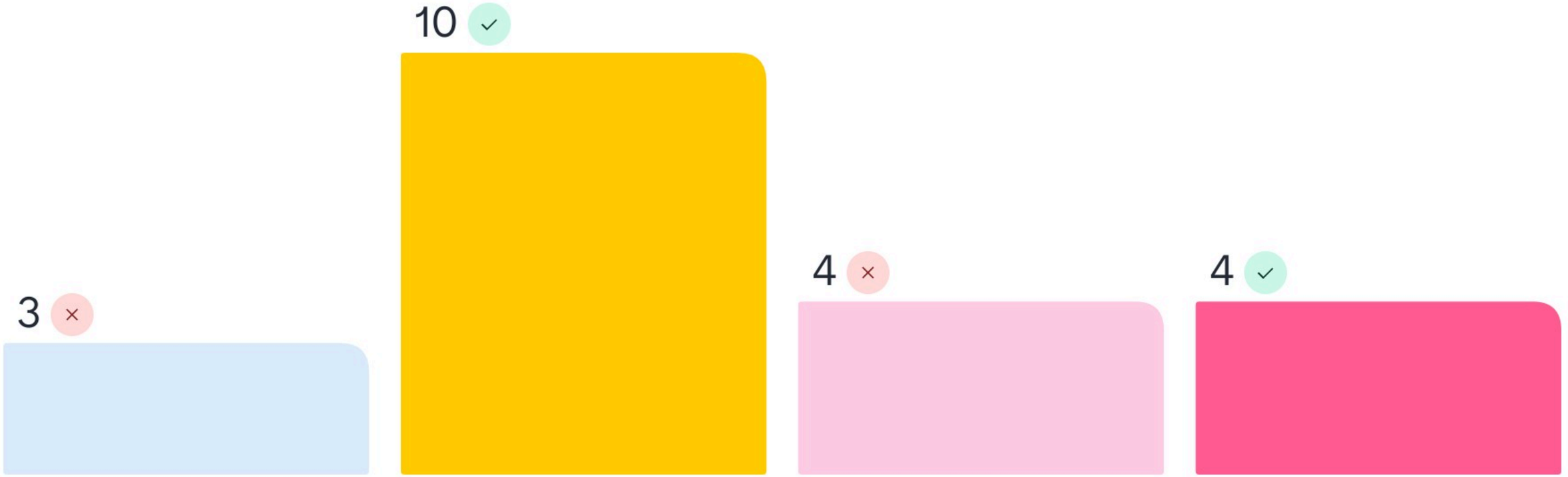
Distributed Model and Challenges

- The solution is to distribute the game model, splitting the workload between both players' computers
- Each player controls their own bat locally for immediate responsiveness
- Bat positions are exchanged over the network so each player sees both bats
- The ball's modeling is more complex and requires careful consideration of latency
- Initially, the ball is modeled on the side of the court where it is located
- This creates a deadlock when the ball crosses the center line, as neither side takes responsibility

Refining the Ball Model

- To fix the deadlock, the player sending the ball continues to send updates until it's well into the other side's court
- This ensures the receiving player recognizes the ball has crossed and takes over modeling
- Video 15 then examines the suitability of TCP (Transmission Control Protocol) for Pong:
- TCP ensures reliable data delivery but introduces unpredictable delays due to retransmissions and flow control
- For a real-time game like Pong, these delays can be detrimental to the gameplay experience

Which of the following challenges arise when using UDP for the Pong protocol?



Ensuring reliable delivery of every bat position update

Handling potential packet loss and out-of-order message arrival

Maintaining a constant frame rate for smooth animation

Implementing a mechanism to prevent deadlocks when the ball crosses the center line

UDP and its Implications

- The alternative is UDP (User Datagram Protocol), which offers minimal overhead and faster transmission but doesn't guarantee delivery
- With UDP, lost packets are not retransmitted, requiring the application to handle potential data loss
- Video 15 explains how frequent ball updates in Pong make UDP a viable option
- Even if some updates are lost, subsequent updates quickly replace them, minimizing the impact on gameplay
- However, UDP introduces a new challenge: ensuring reliable transfer of critical game events

Ensuring Reliability with UDP

- Despite using UDP, certain events in Pong still require reliability, such as the ball crossing the center line
- A simple handshake mechanism is implemented where the sending player continues sending ball updates until receiving a response
- This confirmation indicates the receiving player has taken over ball modeling, preventing deadlocks
- The next challenge is addressing latency, which becomes apparent when testing the game over a simulated network with delay
- A noticeable glitch occurs as the ball's position jumps when the modeling responsibility switches between players

Mitigating Latency Effects

- The solution is to optimize the ball modeling handover point
- Instead of splitting the court in half, each player models the ball when it's heading towards them
- This minimizes the visual impact of latency glitches, as they occur when the ball is furthest from the player
- Video 15 emphasizes the importance of acknowledging and accounting for both latency and packet loss in network game design
- Finally, the video outlines steps for creating a protocol specification for the networked Pong game

What key considerations when designing a protocol specification for a networked game like Pong?

TCP or UDP 7 Popular	Encoding 5	latency, reliability 5	What, when and how to send 4
Type of messages 1	Minimisation of information sent 1	When to send what message 1	Security 1

1 6

What key considerations when designing a protocol specification for a networked game like Pong?

How often do you update positions etc.

1

1 6

Protocol Specification

- The specification defines the communication rules, ensuring both players' implementations are compatible
- It includes details about:
- The chosen protocol (UDP in this case)
- Message types (e.g., bat updates, ball updates, initialization)
- Message content and frequency
- Rules for ball modeling handover to avoid deadlocks

Protocol Details

- Additional considerations in the specification include:
- Including velocity information in messages for smoother animation in case of packet loss
- Defining the coordinate system and range for consistent interpretation of positions
- Establishing a mechanism to determine player 1 and player 2, using random numbers to break ties
- Synchronizing the game start using initialization messages
- Optimizing message encoding for efficiency, potentially combining multiple updates into a single packet

Please ask your questions

14 questions
6 upvotes