CS6111 – Computer Networks Lab – Lab 9 – 20.09.2024

Batch – N&Q

Exercise 1: Implementing Basic Flow Control Algorithm (Stop-and-Wait)

- Implement the sender and receiver. The sender sends one packet, and the receiver sends an acknowledgment (ACK).
- Introduce network conditions with:
  - **No packet loss and no delay**.
  - **Packet loss (e.g., 10% loss)**.
  - **Delay (e.g., 100 ms delay)**.

1. Measure throughput: How many packets can be sent and received successfully in a given time (e.g., 10 seconds).
2. Measure efficiency: How efficiently the sender can transmit packets without being idle for long periods (e.g., waiting for ACK).
3. Experiment with different window sizes and compare the results.

Questions:

- What is the impact of packet loss and delay on throughput?
- How does efficiency change as network conditions worsen?

Exercise 2: Implementing Karn's Algorithm (Handling Retransmission Timeouts)

- Modify the Stop-and-Wait protocol to include Karn's algorithm for timeout calculation.
- Introduce a dynamic timeout based on the **Round Trip Time (RTT)**.
- Simulate network conditions with varying levels of packet loss (e.g., 5%, 20%) and delay (e.g., 50 ms, 200 ms).

1. Measure throughput and efficiency under varying network conditions.
2. Compare the performance with and without Karn's algorithm.

Questions:

- How does Karn's algorithm improve efficiency and throughput compared to the basic Stop-and-Wait?
- What role does accurate timeout calculation play in Karn's algorithm when network delay and packet loss increase?

Exercise 3: Implementing Jacobson's Algorithm (Improved RTT Estimation with Variance)

- Use the same Stop-and-Wait protocol, but modify the timeout calculation to use both **Smoothed RTT (SRTT)** and **RTT Variance (RTTVAR)**.
- The formula to calculate the timeout is: Timeout=SRTT+4×RTTVARTimeout = SRTT + 4 \times RTTVARTimeout=SRTT+4×RTTVAR

Simulate network conditions with:

- **Varying delays** (e.g., 20 ms, 100 ms, 200 ms).
- **Jitter** (randomized delay variability between packets).

1. Measure throughput and efficiency under varying levels of jitter and delay.
2. Compare Jacobson's algorithm with Karn's algorithm in terms of adaptability to changing network conditions.

Questions:

- How does Jacobson's algorithm outperform Karn's algorithm in networks with jitter?
- What impact does considering RTT variance have on throughput and efficiency?

Exercise 4: Comparative Analysis of Throughput and Efficiency

Set up simulations where you introduce various network conditions:

- **No packet loss, minimal delay**.
- **High packet loss (20% packet loss)**.
- **High delay (200 ms)**.
- **High jitter (random delay variation)**.

1. Measure and compare the throughput of each algorithm.
2. Measure and compare the efficiency (number of retransmissions, packet loss recovery, timeout handling).
3. Use tools like graphs to visualize performance metrics over time.

Questions:

- Which algorithm maintains the highest throughput under low-delay conditions?
- Which algorithm is more resilient to packet loss and high delay?
- How does each algorithm handle jitter and variations in network delay?

Exercise 5: Efficiency under Extreme Conditions

Simulate extreme network environments such as:

- **Very high packet loss (e.g., 40%)**.
- **Very high delay (e.g., 500 ms)**.
- **Network congestion with varying bandwidth**.

1. Measure the ability of each algorithm to recover from packet loss, retransmissions, and high delays.
2. Analyze throughput and efficiency under these harsh conditions.

Questions:

- How do each of the algorithms handle the retransmission of lost packets?
- Which algorithm shows the best recovery from severe network degradation?