# Lab 02 - Simulating TCP Tahoe: Understanding Congestion Control Mechanisms

Sherif Saad

January 21, 2025

## Purpose

This lab aims to provide students with a hands-on understanding of the TCP Tahoe congestion control mechanism. Students will simulate TCP Tahoe using Python, observe its behavior under different network conditions, and analyze the effects of packet loss, slow start, congestion avoidance, and fast retransmit.

# 1 Lab Instructions

## 1.1 Task 1: Running the TCP Tahoe Simulation

1. Download the provided TCP Tahoe simulation code.
2. Run the script in a Python environment.
3. Observe the output, focusing on the changes in the congestion window (`cwnd`) and slow start threshold (`ssthresh`).

**Questions:**

- What happens to `cwnd` during slow start?
- How does TCP Tahoe handle packet loss?
- What triggers the transition from slow start to congestion avoidance?

## 1.2 Task 2: Analyzing the Simulation Code

1. Review the provided code to understand how TCP Tahoe is implemented.
2. Identify the sections of the code responsible for:
    - Slow start (exponential growth of `cwnd`).
    - Congestion avoidance (linear growth of `cwnd`).
    - Handling packet loss.
3. Add comments to the code explaining each section.

**Questions:**

- How is `cwnd` growth modeled during slow start and congestion avoidance?
- What role does `ssthresh` play in TCP Tahoe?

### 1.3 Task 3: Modifying the Simulation

1. Add a configurable maximum segment size (MSS) to the simulation.
2. Modify the code to log `cwnd` and `ssthresh` values to a CSV file.
3. Add a graphical plot of `cwnd` over time using Matplotlib.

**Questions:**

- How does the packet loss rate affect TCP Tahoe's performance?
- How does increasing the maximum congestion window size impact the simulation?

### 1.4 Task 4: Exploring Scenarios

1. Test the simulation with different packet loss rates (e.g., 0.05, 0.2, 0.5).
2. Observe how `cwnd` evolves under different scenarios.

**Questions:**

- How does a higher packet loss rate affect `cwnd` dynamics?
- What differences do you observe when modifying `ssthresh` values?

### 1.5 Task 5: Extending the Simulation

1. Research another TCP congestion control algorithm (e.g., TCP Reno).
2. Extend the code to include the chosen algorithm.
3. Compare the performance of the algorithms under identical conditions.

**Questions:**

- How does your chosen algorithm differ from TCP Tahoe?
- Which algorithm performs better under high packet loss?

## 2 Deliverables

- Annotated and commented Python code for the TCP Tahoe simulation.
- A CSV file logging `cwnd` and `ssthresh` values over time.
- A plot visualizing the dynamics of `cwnd` during the simulation.
- A short report answering all lab questions with observations and explanations.
- Extended simulation code with an additional congestion control algorithm, if applicable.

## 3 Assessment

- **10%**: Successful execution of the provided simulation code.
- **20%**: Annotated and commented code analysis.
- **30%**: Modifications (e.g., MSS, logging, plotting) implemented correctly.
- **20%**: Observations and analysis of different scenarios.
- **20%**: Extension with an additional TCP algorithm (e.g., TCP Reno).