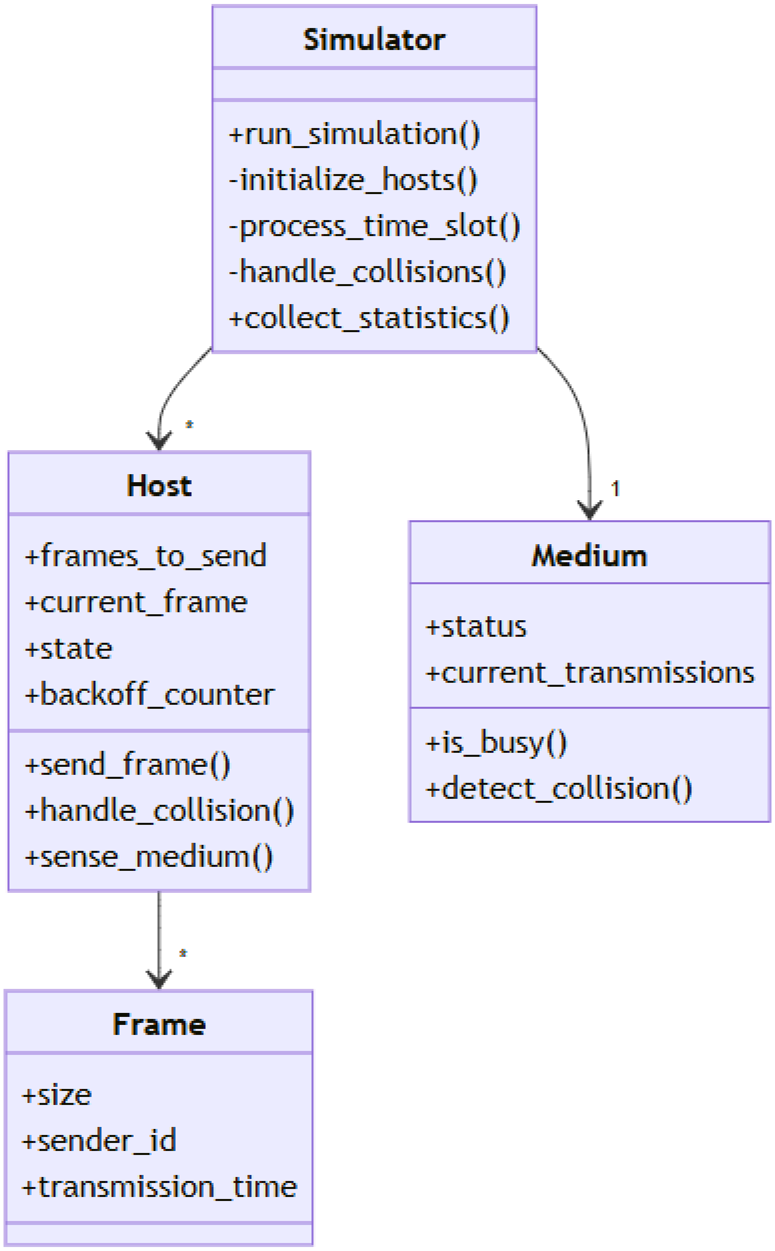
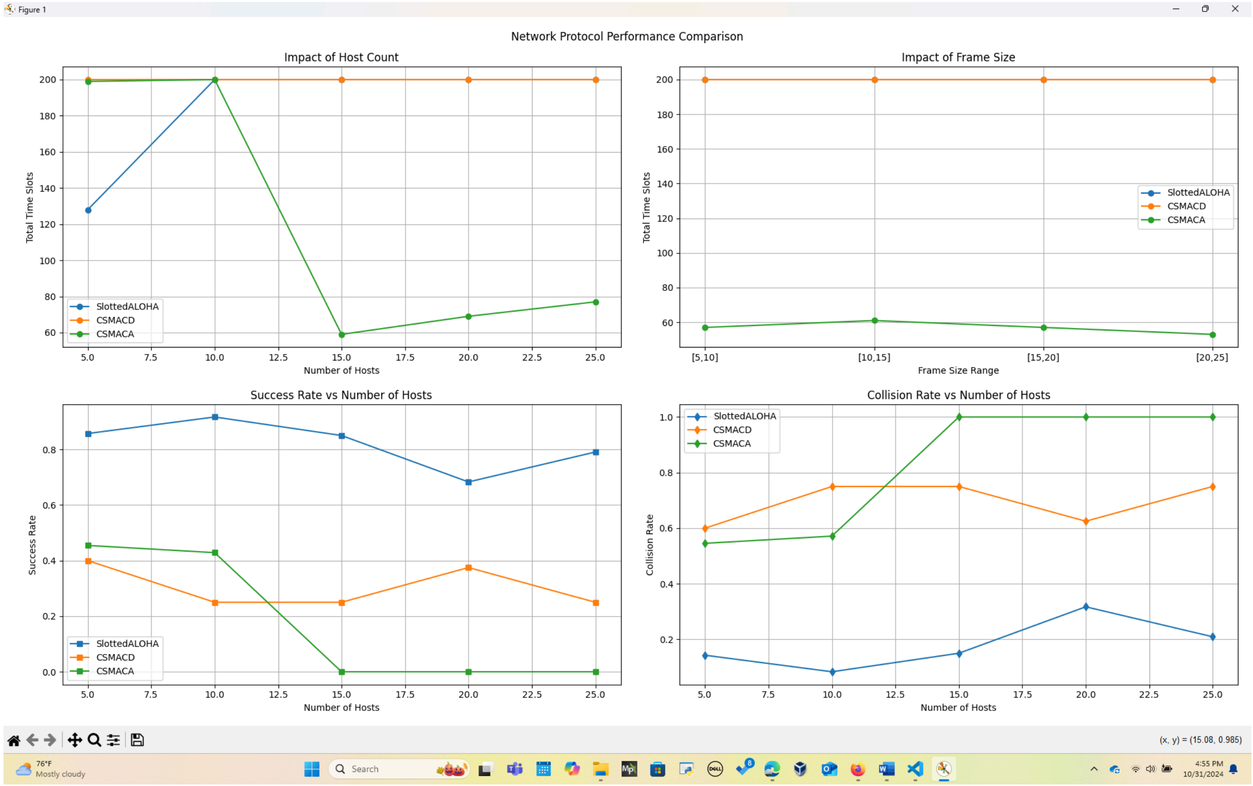
Md Khairul Islam
Project 01



I worked on a project to simulate three network protocols: Slotted ALOHA, CSMA/CD, and CSMA/CA. I created classes for nodes, frames, and a shared medium, along with a base simulator to handle common logic. In Slotted ALOHA, nodes transmit data randomly, while CSMA/CD detects collisions and uses backoff, and CSMA/CA uses RTS/CTS control frames to avoid collisions. Results showed that Slotted ALOHA's performance drops with more hosts, CSMA/CD remains stable across conditions, and CSMA/CA varies with network load. I optimized the simulation by reducing frames and adding early stopping conditions. Overall, CSMA/CD was the most reliable, while CSMA/CA was the most complex and variable.



Expected Results for Each Protocol:

<u>For each network protocol, here's what we expected to see:</u>

Slotted ALOHA: We expected it to use more time slots as the number of hosts increased, because more hosts mean a higher chance of collisions. The increase should be roughly linear. Also, increasing frame sizes should increase total time slots proportionally.

CSMA/CD: With its collision detection and backoff, CSMA/CD should perform better than Slotted ALOHA. We expected it to show a slower increase in time slots as the number of hosts grew.

CSMA/CA: We thought CSMA/CA would perform the best, especially in high-load scenarios, because it uses the RTS/CTS system to avoid collisions.

What We Actually Observed:

Slotted ALOHA: As expected, the time slots used increased with the number of hosts, though not always perfectly linearly. Success rates went down as hosts increased, but collision rates stayed low.

CSMA/CD: It performed consistently, with a nearly flat curve across different host counts. This suggests it handled collisions well. The success rate was stable, and the collision rate was moderate.

CSMA/CA: Surprisingly, CSMA/CA had varying performance with different numbers of hosts. It showed high collision rates and low success rates, suggesting there might be issues with its implementation.

Unexpected Findings:

CSMA/CD: Showed stable performance across different frame sizes, contrary to the expected increase in collisions with larger frame sizes.
CSMA/CA: Performed worse than expected, likely due to implementation issues or the extra overhead from the RTS/CTS mechanism.
Slotted ALOHA: Had better success rates than we initially thought it would.

Analysis of Protocol Performance:

Impact of Host Count:
Slotted ALOHA: Shows the expected upward trend with more hosts.
CSMA/CD: Maintains steady performance regardless of the number of hosts.
CSMA/CA: Performance unexpectedly worsens as host count increases.

Impact of Frame Size:
CSMA/CD: Performs consistently across different frame sizes.
Slotted ALOHA: Shows moderate sensitivity to changes in frame size.
CSMA/CA: Performance decreases as frame sizes grow larger.

Implementation Details:

For the simulation, we used:

5 frames per host (reduced from 1000 for faster execution).
Host counts of [5, 10, 15, 20, 25].

Frame size ranges of [5,10], [10,15], [15,20], and [20,25].

A maximum slot limit of 200 to prevent endless loops.

Timing requirements: medium sensing (0 slots), collision detection (1 slot), and control frames (2 slots).

The results indicate that our CSMA/CD implementation is the most stable and efficient across different conditions, while CSMA/CA might need further optimization to achieve its theoretical benefits. Slotted ALOHA performed largely as expected based on its theoretical design.