

COMPUTER NETWORKS ASSIGNMENT 3

SIMULATION OF GO- BACK-N PROTOCOL

Presented By:

Aarya Gupta (2022006)

Adarsh Jha (2022024)

IIITD | 2024

OVERVIEW

Key Objectives:

- Implement Go-Back-N protocol to simulate a data link layer protocol.
- Use UDP as the underlying transport to facilitate full-duplex data transfer.
- Aim: Understand and simulate packet transmission, acknowledgment, and retransmission with the use of 2 different machines.

ABOUT GO-BACK-N

- **Go-Back-N is an ARQ (Automatic Repeat reQuest) protocol with error control.**
- **Uses a sliding window for sequence numbers (modulo-8 numbering) with a sender window size of 7.**
- **Retransmits unacknowledged frames after a timeout, maintaining flow and error control.**

HOW DATA LINK LAYER IS USED AND CONCEPT OF USING MULTIPLE MACHINES?

- **Data Link Layer (DLL):** Simulates packet handling, encapsulation, and acknowledgment at DLL level using sequence and acknowledgment numbers.
- **Multiple Machines:** Uses two machines with DL_Entity_1 as client and DL_Entity_2 as server, communicating over datagram sockets for frame transmission.

USE OF MULTITHREADING

- **Multithreading is used to handle concurrent tasks such as:**
- **Packet generation at random intervals.**
- **Packet transmission and reception.**
- **Timeout handling for frame retransmission in the Go-Back-N protocol.**

OUTPUT: MACHINE 1

```
53096 Simulation Completed.  
53097 Total Frames Sent: 10707  
53098 Total Retransmissions: 707  
53099 Average Delay per Packet: 0.1543 seconds  
53100 Average Number of Times a Frame was Sent: 1.07  
53101
```

OUTPUT: MACHINE 2

```
89235 Simulation Completed.  
89236 Total Frames Sent: 19463  
89237 Total Retransmissions: 9463  
89238 Average Delay per Packet: 0.3450 seconds  
89239 Average Number of Times a Frame was Sent: 1.95  
89240
```

TERMINAL OUTPUT

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS JUPYTER COMMENTS

```
C:\Users\aaarya\Github_Projects\CN_Assignments\CN_ASS_3\Using Two Machines>python3 machine1.py
Starting Simulation on Machine 1
Simulation Completed on Machine 1. Output saved to machine1_output.txt

C:\Users\aaarya\Github_Projects\CN_Assignments\CN_ASS_3\Using Two Machines>
```

```
C:\Users\aaarya\Github_Projects\CN_Assignments\CN_ASS_3\Using Two Machines>python3 machine2.py
Starting Simulation on Machine 2
Simulation Completed on Machine 2. Output saved to machine2_output.txt

C:\Users\aaarya\Github_Projects\CN_Assignments\CN_ASS_3\Using Two Machines>
```


ASSUMPTIONS

Case 1: Lower drop probability, lower delays

```
{  
  'P': 0.01, # Lower drop probability  
  'T3': 0.05,  
  'T4': 0.1,  
  'output_file': 'simulation_case1.txt'  
}
```

Case 2: Higher drop probability, higher delays

```
{  
  'P': 0.1, # Higher drop probability  
  'T3': 0.1,  
  'T4': 0.2,  
  'output_file': 'simulation_case2.txt'  
}
```

CONCLUSION

- **Successfully simulates Go-Back-N protocol with packet drop and delay to mirror real network conditions.**
- **Results provide insights into packet delay, retransmissions, and error control in data link protocols.**
- **Enhances understanding of network layer functionality using Go-Back-N in a controlled environment.**

IIITD | 2024

THANK YOU