1 System Design

The system implements a complete network simulation with packet structures for data and ACKs, transmission buffers for reliable delivery and tasks to model the behaviour of the network.

- Tasks to implement such system are: senders, switch, and receivers:
  - Sender: Generates packets at random intervals and retransmits them if ACKs are not received within timeout.
  - Receiver: Accepts packets in correct order and sends ACK confirmations back to senders.
  - Switch: Forwards packets between nodes while randomly dropping some and adds network delays based on packet size and link capacity.
- The Send-and-Wait Protocol:
  - Sends one packet at a time and waits for its ACK before sending the next.
  - On timeout, only the lost packet is retransmitted, making it simple but slow.
- The Go-Back-N Protocol:
  - Uses sliding windows where senders can transmit N packets before waiting for ACKs, and receivers only accept in-sequence packets.
  - When timeouts occur, the entire window gets retransmitted, while ACKs cumulatively acknowledge all previous packets.

All components communicate through queues and the system tracks statistics until reaching the target of 2000 received packets per receiver.

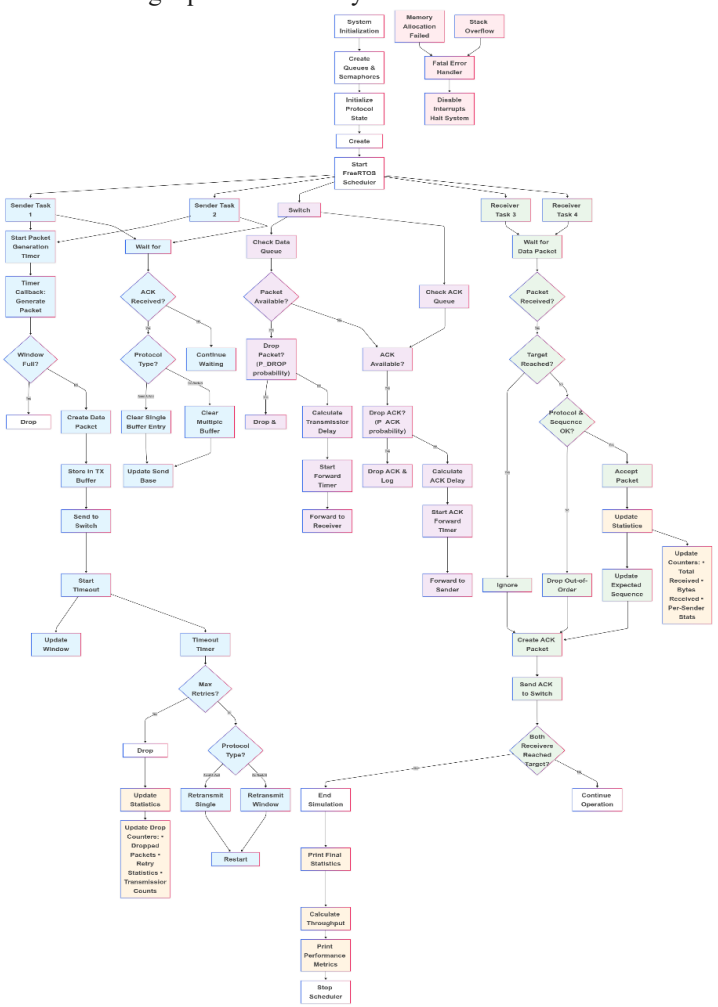

Figure 1: System Overview

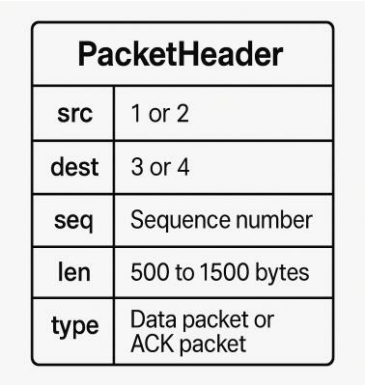

Figure 2: Packet Header Design

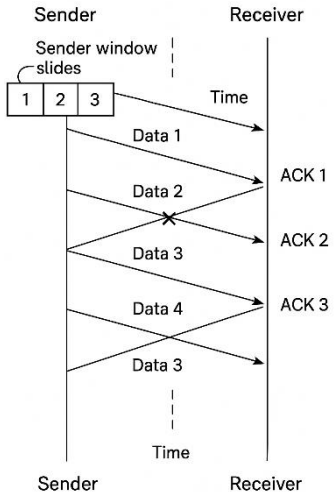

Figure 3: Go-Back-N Protocol

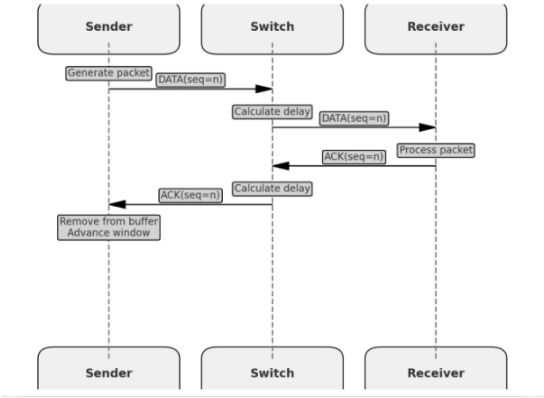

Figure 4: Normal Sequence

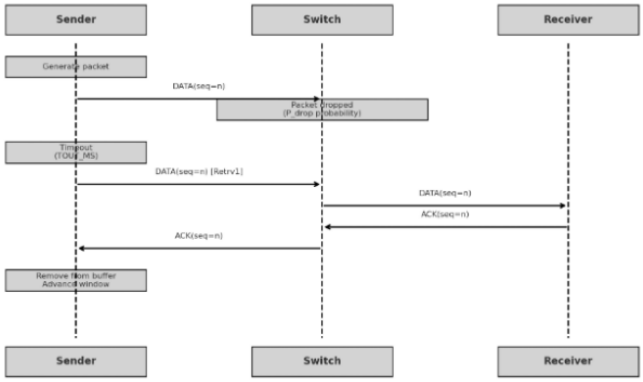

Figure 5: Packet Loss and Retransmission

The system includes:

- **Buffer & Queue Management:**
  - Transmission buffer clearing on timeouts.
  - Multiple queue types for data and ACK packets.
  - Buffer fullness checking before transmission.
- **Timing & Synchronization:**
  - Multiple timer types: generation, transmission, and forward timers.
  - FreeRTOS-based task scheduling with semaphores.
  - Timeout handling for both individual packets and entire windows.
- **Protocol Flexibility:**
  - Runtime protocol type switching capability.
  - Sequence number validation and out-of-order detection.
  - Adaptive window management with dynamic updates.

2 Results and Discussion

Table 1: System Statistics

P_{drob}	T_{out}	N	Throughput	Avg no.of trans.	Dropped packets due to more than 4 retries.	p_{drob}	T_{out}	N	Throughput	Avg no.of trans.	Dropped packets due to more than 4 retries.
0.02	150	1	12259.375	1.019	0	0.02	175	4	13102.066	1.058	13
0.04	150	1	12217.588	1.100	0	0.04	175	4	12978.032	1.181	18
0.08	150	1	12132.812	1.132	0	0.08	175	4	12838.375	1.324	47
0.02	150	2	12652.312	1.071	30	0.02	175	8	13520.000	1.064	15
0.04	150	2	12651.187	1.132	28	0.04	175	8	13474.400	1.111	26
0.08	150	2	12634.000	1.165	69	0.08	175	8	13445.333	1.113	33
0.02	150	4	13342.466	1.042	0	0.02	200	1	11448.705	1.018	0
0.04	150	4	13329.000	1.100	38	0.04	200	1	11340.176	1.097	0
0.08	150	4	13201.400	1.186	73	0.08	200	1	9586.836	1.113	1
0.02	150	8	14165.666	1.044	0	0.02	200	2	12435.625	1.071	125
0.04	150	8	13575.066	1.200	7	0.04	200	2	12401.500	1.219	119
0.08	150	8	13524.733	1.214	36	0.08	200	2	12288.250	1.850	251
0.02	175	1	12070.687	1.018	0	0.02	200	4	12690.125	1.108	71
0.04	175	1	11876.000	1.099	0	0.04	200	4	12678.562	1.093	57
0.08	175	1	11689.647	1.136	0	0.08	200	4	12658.187	1.087	68
0.02	175	2	12553.187	1.107	52	0.02	200	8	13431.066	1.086	24
0.04	175	2	12526.687	1.136	123	0.04	200	8	13419.800	1.183	29
0.08	175	2	12481.500	1.455	138	0.08	200	8	13392.666	1.214	35

Observations:

- **Throughput:** We will discuss it more later**.**
- **Average Number of Transmissions:**
  - Increases with higher P_drop values.
  - Go-Back-N (N>1) shows higher transmission counts because retransmitting one packet triggers retransmission of subsequent packets in the window.
- **Dropped Packets Due to Max Retries:**
  - Increases with P_drop as higher drop probability leads to more packets exceeding 4 retry attempts.
  - Go-Back-N shows significantly more drops.
  - This occurs because Go-Back-N retransmits entire windows, making it more likely to hit the 4-retry limit.

Go-Back-N behavior:

- Go-Back-N is like a frustrated teacher who makes the entire class repeat a lesson when one student doesn't understand.
- In a noisy classroom (high packet loss), this becomes counterproductive (more repetition just creates more chaos).
- The 4-retry limit acts like a "give up" rule, but Go-Back-N hits this limit faster because it's retransmitting whole chunks instead of individual packets. It's like trying to shout louder in a storm instead of waiting for it to pass.
- The data shows Go-Back-N's "repeat the whole lesson" approach backfires in lossy conditions, for example: how dropped packets due to 4 retries jump from 1 (N=1) to 251 (N=2) at high loss rates

After Collecting the data in all windows while varying P_drop and T_out, we used MATLAB in order to represent this data graphically and find the required functions.

Plots:

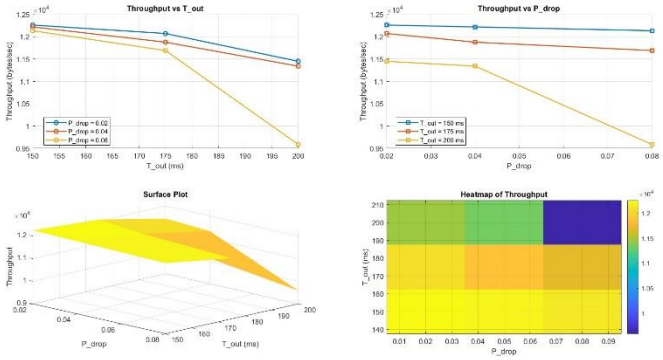

Figure 6: Throughput vs. (P_drop/Tout)@N=1

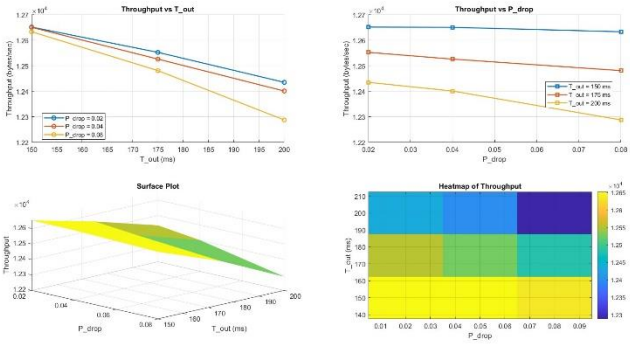

Figure 7: Throughput vs. (P_drop/Tout)@N=2

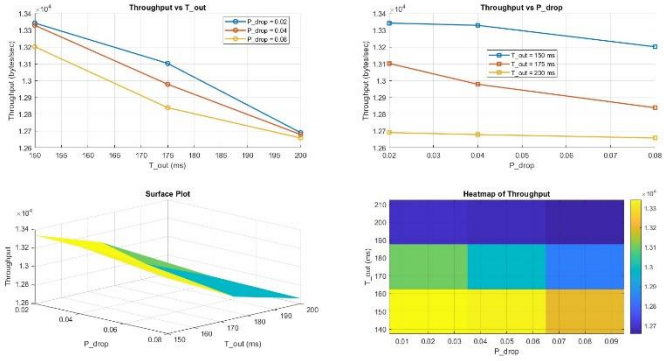

Figure 8: Throughput vs. (P_drop/Tout)@N=4

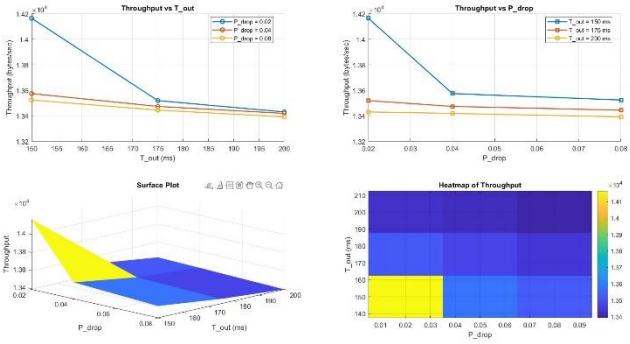

Figure 9: Throughput vs. (P_drop/Tout)@N=8

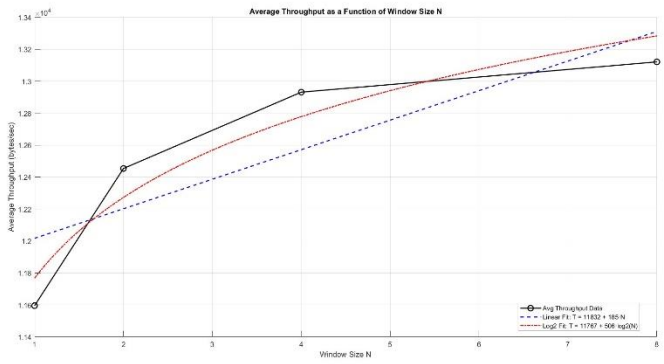

Figure 10: Throughput as a function of N

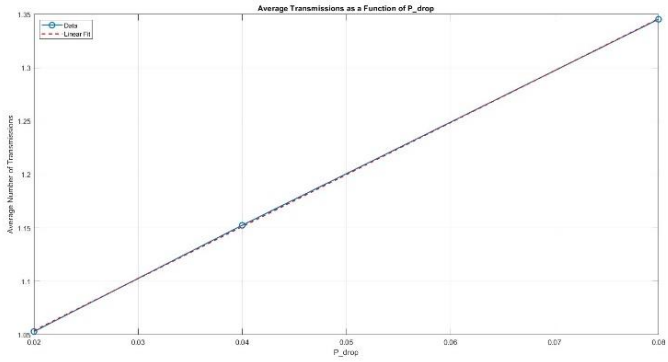

Figure 11: Avg no. of transmissions as a function of P_drop

Notes on plots:

1. Throughput mostly increases with N: Because you can send more packets before waiting for ACKs, which reduces simulation time therefore increasing throughput.
2. Throughput decreases with P_drop: More packets get lost and must be resent, wasting time.
3. Throughput slightly decreases when T_out increases: After a loss, you wait longer before retransmitting, this stalling increases the simulation time.
4. Average number of transmissions increases when P_drop increases: As packet loss rises, more packets must be resent, so the average transmissions per packet goes up.
5. **Heatmaps show:**
   - Higher throughput (yellow regions) occurs at lower timeout values and lower drop probabilities.
   - Performance degrades (blue regions) as both parameters increase.
6. **3D surface plots show:**
   - Small changes in timeout or drop probability cause big changes in throughput (Once a system is already performing poorly (high timeouts, high drops), making it slightly worse doesn't hurt performance much more. But when it's performing well, small increases in problems cause big performance drops.)
7. **Functions:**
   - Linear Fit: Throughput = 11759 + 242·N
   - Logarithmic Fit: Throughput = 11731 + 624·log(2N)
   - Linear Fit**:** Average Transmissions ≈ 0.9565 + 4.8679 *P_drop
   - These functions agree with the behavior we expected and discussed.

3 Conclusion

The simulation successfully compared Send-and-Wait and Go-Back-N protocols under varying conditions. The results show that increasing window size improves throughput, while higher packet loss and timeouts degrade performance. Go-Back-N performs better in low-loss environments but struggles with high packet loss due to unnecessary retransmissions. The derived linear relationships confirm the expected protocol behaviors.