Middle East Technical University

Department of Computer Engineering

CENG 435

Data Communications and Networking

Assignment 1 - Submission

Kemahlı
yusuf.kemahli@metu.edu.tr

Şanlıer
sanlier.omer@metu.edu.tr

November 3, 2025

1 Introduction

This report has been prepared by **Yusuf Kemahlı** and **Ömer Faruk Şanlıer** as part of the **CENG435: Data Communications and Networking** course offered at **Middle East Technical University**. The purpose of this assignment is to provide a hands-on understanding of fundamental concepts in data communication by performing network measurements using the **OMNeT++ 6.2.0** simulation environment and the **INET 4.5.4** framework.

Throughout the assignment, multiple aspects of network performance have been examined through carefully designed simulation scenarios. The focus is on understanding how core networking parameters influence measurable quantities such as **throughput**, **utilization**, **delay**, **jitter**, **transmission time**, **propagation time**, and **queuing time**. Each of these metrics provides valuable insights into how communication systems behave under different conditions.

The results obtained from the conducted measurements are systematically analyzed and interpreted in the subsequent sections of this report. Each section presents the corresponding network metric, the methodology used to measure it, and the discussion of its behavior under varying parameters. This structure ensures a clear and consistent presentation of findings throughout the report.

1.1 Experiment Setup

All simulations in this assignment were carried out using the **OMNeT++ 6.2.0** simulation environment and the **INET 4.5.4** framework. The experiment configurations were defined primarily through two types of files: the **.ned** files, which describe the network topology, and the **omnetpp.ini** files, which specify simulation parameters such as link bitrates, interarrival times, and other runtime variables. The analysis of the obtained simulation results was performed using the **.anf** analysis files provided by OMNeT++, allowing detailed examination and visualization of the collected statistics.

The simulation environments were built in the **showcases/measurement** directory of INET, where each measurement type (e.g., **throughput**, **propagation time**) had its own dedicated subdirectory, such as `inet-4.5.4/showcases/measurement/throughput`. Within these directories, existing demonstration scenarios were modified to fit the requirements of each question in the assignment. For several tasks, the **EthernetLink** channel was extended to create custom links that allowed adjusting parameters such as the **bit error rate (BER)**. Additionally, **EthernetSwitch** modules were incorporated between source and destination nodes to simulate **multi-switch topologies** where required.

During the setup process, one of the main challenges was extending the **EthernetLink** to modify the **bit error rate** parameter effectively. To resolve this, the underlying implementation of **EthernetLink**, **DatarateChannel**, and **cDatarateChannel** source files was carefully examined to understand how parameters were structured and inherited. After analyzing these definitions, a simpler and more direct method was identified to enable BER configuration through the extended channel definition.

Another difficulty encountered was related to the installation of OMNeT++ and INET on a **macOS** environment. Initially, manual installation attempts resulted in version and dependency errors. This issue was resolved by switching to the **opp_env** setup method, which streamlined the installation and ensured that both OMNeT++ and INET were properly configured. By consulting documentation and community discussions, all installation-related errors were successfully addressed, enabling smooth execution of the simulation tasks.

2 Answers

2.1 Channel Throughput

2.1.1 Q1: Channel Throughput vs Interarrival Time (Exponential IAT)

Objective: To investigate how the channel throughput changes with varying packet interarrival times (**IAT**). The goal is to observe how decreasing the IAT (increasing the packet generation rate) affects the overall throughput and to identify where the channel becomes saturated.

Results: The experiment was performed for IAT values of **50 µs, 100 µs, 200 µs, 400 µs, and 800 µs**. Each configuration was repeated 20 times, and the mean throughput with its corresponding **95% confidence interval** was calculated. The results are summarized in Table 1 and visualized in Figure 1.

Discussion: As shown in Table 1 and Figure 1, when the packet interarrival time decreases, the mean throughput initially increases until it reaches the maximum channel capacity. For IAT values below approximately **100 µs**, the throughput remains nearly constant at around **99 Mbps**, indicating that the link is fully saturated. Beyond this

Table 1: **Mean Channel Throughput and 95% Confidence Intervals for Different IAT Values**

IAT (µs)	Mean Throughput (Mbps)	95% CI Lower	95% CI Upper
50	99.2556	99.2088	99.3024
100	99.0981	99.0585	99.1377
200	50.7472	50.5562	50.9381
400	25.3475	25.2176	25.4775
800	12.6592	12.5633	12.7551

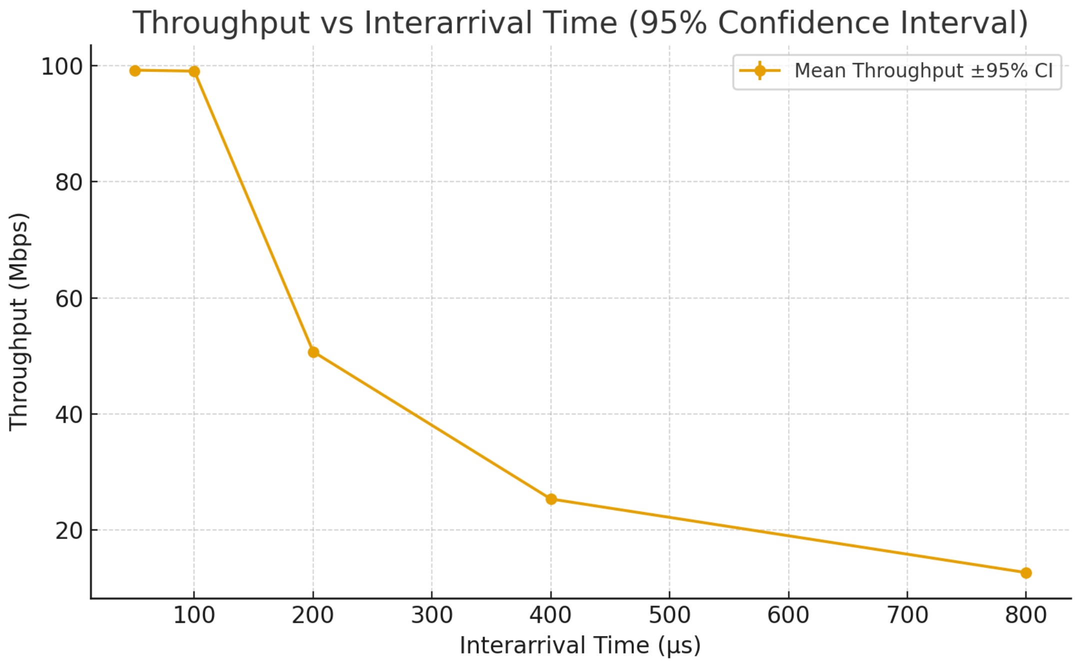

Figure 1: **Channel Throughput vs Interarrival Time (mean + 95% CI)**

point, as IAT increases (i.e., traffic load decreases), the throughput drops proportionally, demonstrating the expected inverse relationship between packet generation rate and throughput.

Conclusion: The results confirm that the channel throughput increases with decreasing IAT up to the link capacity limit (approximately 100 Mbps in this case). Further reduction in IAT does not increase throughput due to saturation, while higher IAT values result in underutilization of the link.

2.1.2 Q2: Channel Saturation Range for Interarrival Time

Objective: To determine the range of **Interarrival Time (IAT)** values for which the channel becomes saturated, meaning that further decreases in IAT do not lead to any increase in throughput. This analysis aims to identify the saturation threshold of the channel.

Results: To obtain a more precise saturation point, the IAT values were tested at smaller intervals between **25 µs** and **125 µs**. Each configuration was simulated with at least 20 repetitions, and the corresponding mean throughput values along with their **95% confidence intervals** are presented in Table 2 and Figure 2.

Table 2: **Channel Throughput for Fine-Grained IAT Values (Mean + 95% CI)**

IAT (µs)	Mean Throughput (Mbps)	95% CI Lower	95% CI Upper
25	99.0519	99.0519	99.0519
50	99.0514	99.0508	99.0520
75	99.0497	99.0482	99.0512
100	98.9801	98.9425	99.0177
125	81.1593	80.9440	81.3746

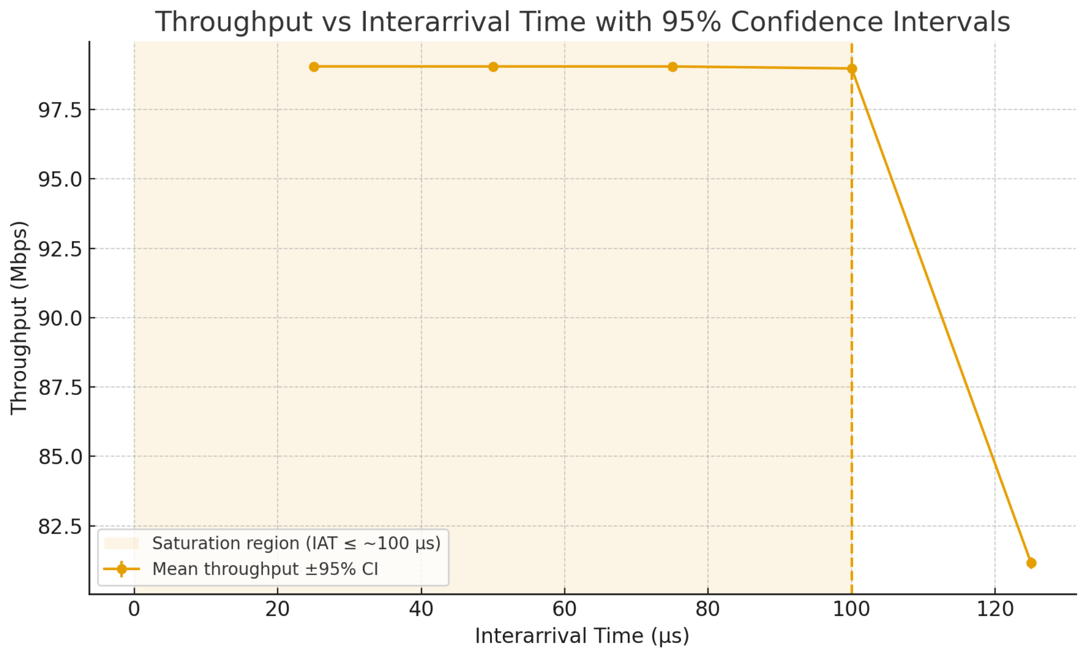

Figure 2: **Channel Throughput vs IAT (Fine-Grained Saturation Analysis)**

Discussion: The results in Table 2 indicate that for IAT values between **25 µs** and approximately **100 µs**, the throughput remains nearly constant at around **99.05 Mbps**, demonstrating that the link operates at its maximum capacity. When the IAT increases beyond **100 µs**, a noticeable drop in throughput is observed, and at **125 µs**, the throughput decreases to approximately **81 Mbps**. This confirms that the channel is fully saturated for IAT values smaller than roughly **100 µs**.

Conclusion: The saturation threshold of the Ethernet link is reached at an interarrival time of around **100 µs**. Below this threshold, the throughput remains stable and does not increase further, even with a higher packet generation rate. Above this point, the throughput decreases proportionally as the offered load declines.

2.1.3 Q3: Channel Throughput vs Ethernet Bitrate

Objective: To examine how varying the Ethernet **link bitrate** affects the overall channel throughput. This experiment aims to verify the proportional relationship between link capacity and achievable throughput under a fixed packet generation rate.

Results: The simulations were conducted with a constant packet interarrival time of **100 µs**, while the Ethernet bitrate was varied across **10 Mbps**, **100 Mbps**, **1 Gbps**, and **10 Gbps**. Each configuration was executed at least 20 times, and the mean throughput values with their corresponding **95% confidence intervals** were obtained. The results are summarized in Table 3 and illustrated in Figure 3.

Table 3: **Channel Throughput for Different Ethernet Bitrates (IAT = 100 µs)**

Bitrate (Mbps)	Mean Throughput (Mbps)	95% CI Lower	95% CI Upper
10	9.901	9.861	9.941
100	98.9937	98.9537	99.0337
1000	101.48	101.23	101.73
10000	101.4991	101.2491	101.7491

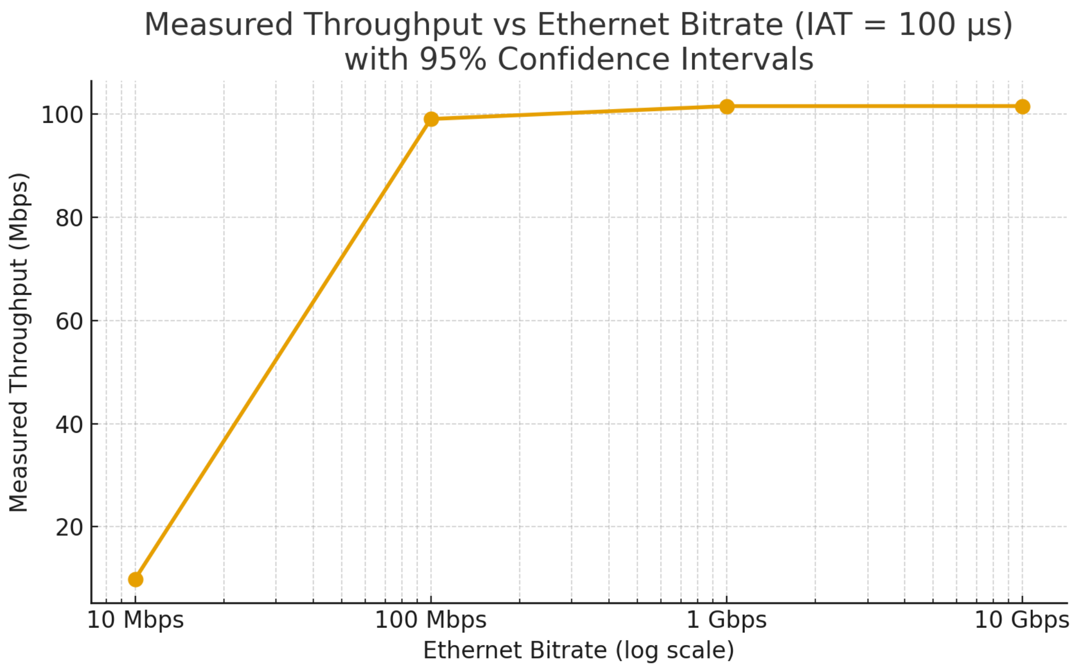

Figure 3: **Channel Throughput vs Ethernet Bitrate (IAT = 100 µs, mean + 95% CI)**

Discussion: The results clearly demonstrate that the throughput scales proportionally with the available link bitrate up to the saturation point of the network configuration. At lower bitrates (10 Mbps and 100 Mbps), the measured throughput values are very close to the nominal link capacities, confirming efficient channel utilization. However, when the bitrate is increased to 1 Gbps and 10 Gbps, the throughput plateaus around **101.5 Mbps**, indicating that the offered traffic rate, constrained by the fixed interarrival time of 100 µs, has become the limiting factor rather than the physical link capacity.

Conclusion: The throughput increases linearly with the link bitrate until the offered traffic rate saturates the transmission capacity defined by the packet generation rate. Beyond approximately **100 Mbps**, the system becomes source-limited, and further increases in link bitrate do not yield higher throughput values.

2.2 Channel Utilization

2.2.1 Q1: Channel Utilization vs Interarrival Time (Exponential IAT)

Objective: To analyze how the **channel utilization percentage** changes as the packet **interarrival time (IAT)** varies. The goal is to observe how utilization reflects the offered load and to determine the point where the channel reaches full usage.

Results: The experiment was conducted using exponential interarrival times of **50 µs**, **100 µs**, **200 µs**, **400 µs**, and **800 µs**. Each configuration was repeated 20 times to ensure statistical confidence. The measured mean utilization percentages and their corresponding **95% confidence intervals** are listed in Table 4 and illustrated in Figure 4.

Table 4: **Channel Utilization for Different Interarrival Times (Mean + 95% CI)**

IAT (µs)	Mean Utilization (%)	95% CI Lower	95% CI Upper
50	99.0518	99.0516	99.0520
100	98.991	98.965	99.017
200	50.81	50.65	50.97
400	25.49	25.39	25.59
800	12.73	12.65	12.81

Discussion: As shown in Table 4, the channel utilization remains nearly constant and close to **100%** for smaller interarrival times (50 µs and 100 µs), meaning the link is fully utilized. As the IAT increases beyond this range, the utilization decreases sharply — falling to about **50.8%** at 200 µs and dropping further to **12.7%** at 800 µs. This clearly reflects the expected inverse relationship between packet generation rate and utilization: a higher IAT results in a lower offered load, and consequently, reduced channel usage.

Conclusion: The channel remains fully utilized for IAT values up to approximately **100 µs**. Beyond this point, utilization declines proportionally with increasing interarrival time, consistent with theoretical expectations for exponentially distributed traffic generation.

2.2.2 Q2: Channel Saturation Range (Utilization → 1)

Objective: To determine the range of **Interarrival Time (IAT)** values where the link utilization converges to 1, meaning that further decreases in IAT do not result in higher channel utilization. This experiment aims to identify the saturation threshold under fine-grained IAT intervals.

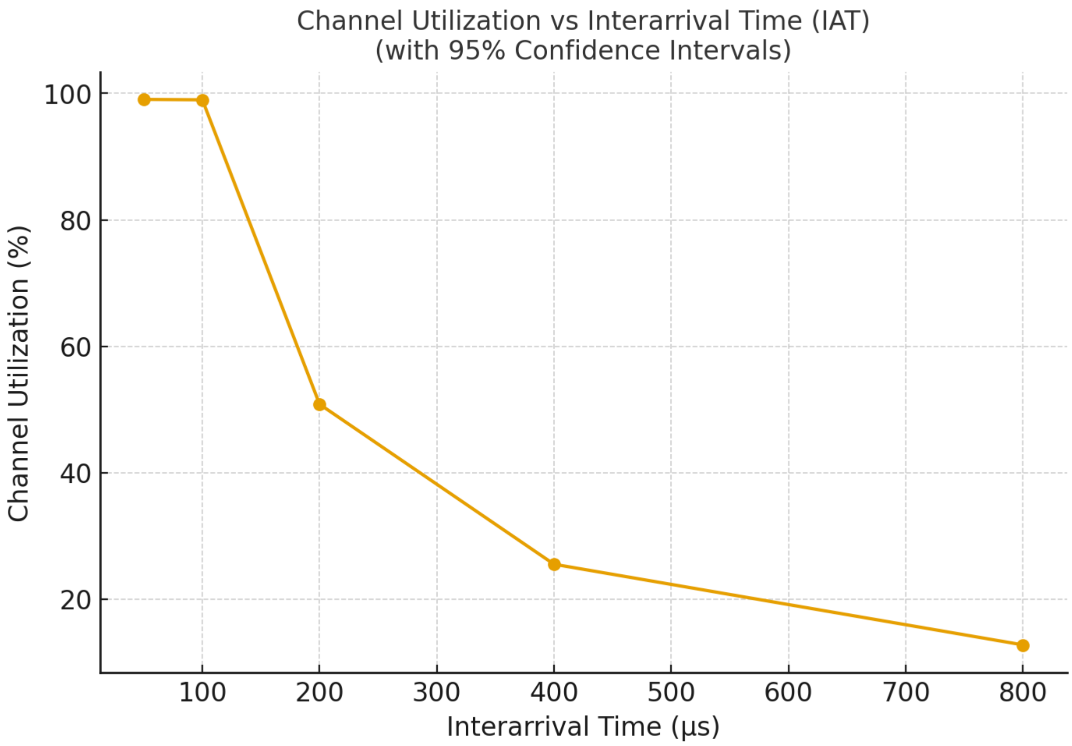

Figure 4: **Channel Utilization vs Interarrival Time (mean + 95% CI)**

Results: To obtain a more precise understanding of the saturation point, the IAT parameter was varied at smaller intervals between **25 µs** and **125 µs**. The corresponding mean utilization values and their **95% confidence intervals** are presented in Table 5 and visualized in Figure 5.

Table 5: **Fine-Grained Channel Utilization Measurements for Different IAT Values (Mean + 95% CI)**

IAT (µs)	Mean Utilization (%)	95% CI Lower	95% CI Upper
25	99.0522	99.0513	99.0531
50	99.0517	99.0499	99.0535
75	99.0500	99.0490	99.0510
100	95.8320	95.4000	96.2640
125	81.0390	80.9180	81.1600

Discussion: As seen in Table 5, the utilization remains nearly constant at around **99.05%** for IAT values up to approximately **75 µs**. This indicates that the link is fully saturated and operating at maximum efficiency within this range. However, at an IAT of **100 µs**, the utilization begins to decrease slightly to around **95.8%**, marking the onset of desaturation. Beyond this point, the utilization drops more significantly — reaching approximately **81%** at **125 µs**. These results clearly show that the link becomes saturated for all IAT values less than roughly **100 µs**.

Conclusion: The channel remains fully saturated and utilization converges to **1** for

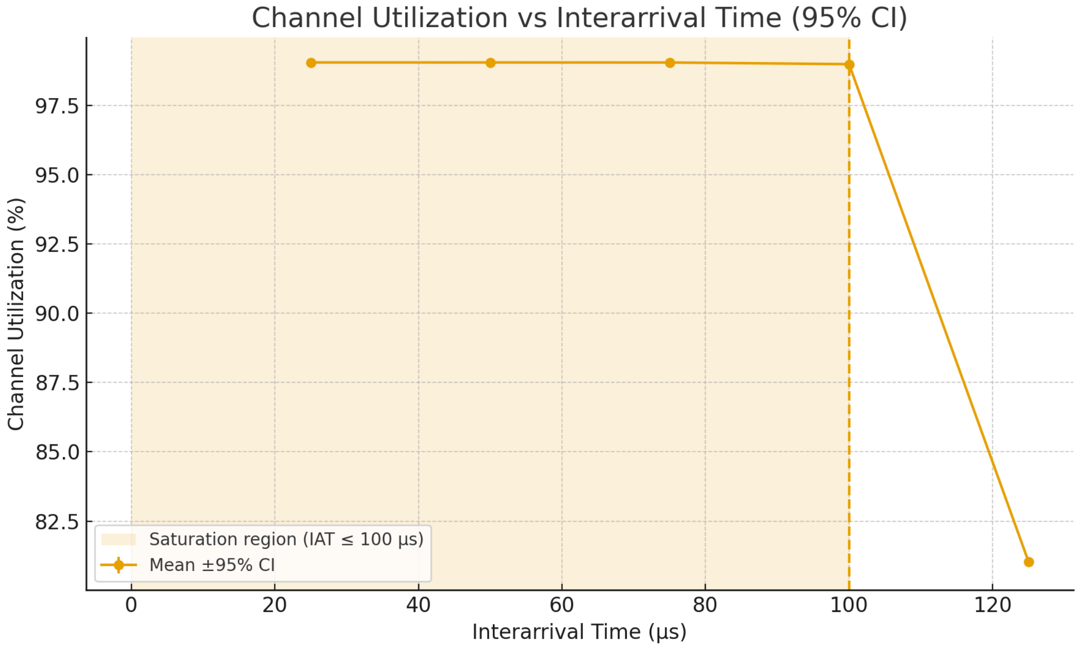

Figure 5: **Channel Utilization vs Interarrival Time (Fine-Grained Saturation Analysis)**

IAT values between **25 µs** and **100 µs**. When the IAT increases beyond **100 µs**, the utilization begins to decrease noticeably, indicating that the link is no longer saturated and the offered load is below capacity.

2.2.3 Q3: Channel Utilization vs Bit Error Rate (BER)

Objective: To investigate the effect of the **Bit Error Rate (BER)** parameter on channel utilization. For this purpose, the default **EthernetLink** channel was extended to create a **custom link** that allows setting the BER value directly. The goal is to observe whether the presence of bit-level transmission errors significantly impacts overall channel utilization.

Results: The simulations were conducted with a fixed packet generation rate while varying the BER parameter across values of $\mathbf{1 \times 10^{-3}}$, $\mathbf{1 \times 10^{-5}}$, $\mathbf{1 \times 10^{-7}}$, and $\mathbf{1 \times 10^{-9}}$. The case of $\mathbf{BER = 0}$ was excluded from the plotted results for clarity, as it produced nearly identical utilization values. Each configuration was run 20 times, and the mean utilization along with its **95% confidence interval** is summarized in Table 6 and visualized in Figure 6.

Discussion: As shown in Table 6 and Figure 6, varying the Bit Error Rate (BER) has an insignificant effect on the measured channel utilization. Across all tested BER values, the utilization remains nearly constant around **50.7–50.8%**. This behavior can be explained by the definition of the utilization metric: it reflects how busy the channel is, not how successfully data is delivered.

In this simulation setup, there is no retransmission mechanism or acknowledgment process implemented between the source and the destination. Therefore, even when bit

Table 6: **Channel Utilization for Different BER Values (Mean + 95% CI)**

BER	Mean Utilization (%)	95% CI Lower	95% CI Upper
0	50.86	50.67	51.05
1e–3	50.78	50.61	50.95
1e–5	50.75	50.57	50.93
1e–7	50.75	50.57	50.93
1e–9	50.79	50.63	50.95

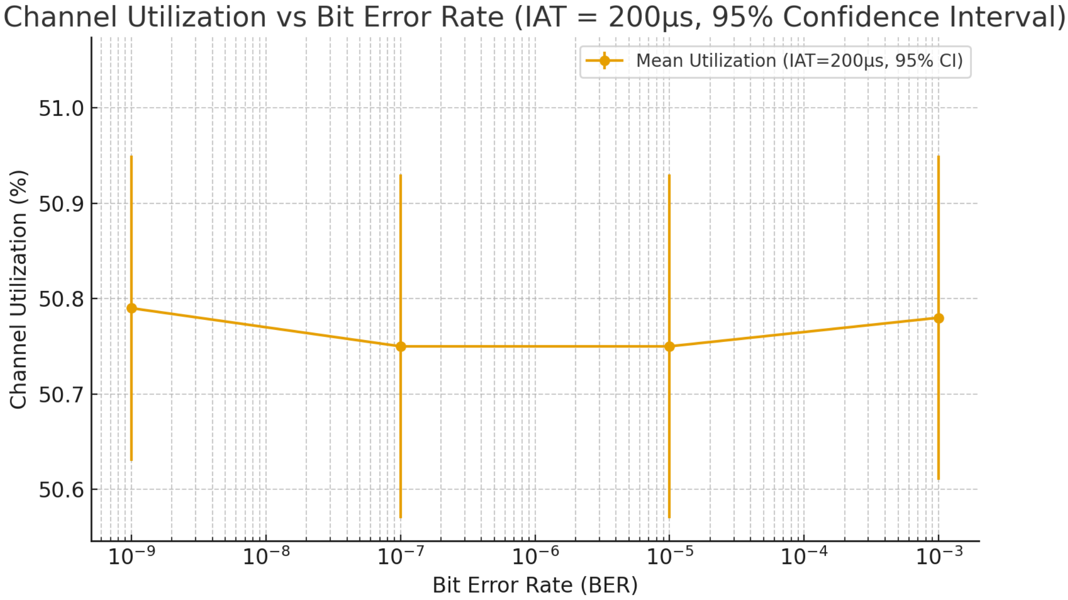

Figure 6: **Channel Utilization vs Bit Error Rate (mean + 95% CI, excluding BER=0)**

errors occur, the transmitter continues to send frames at the same rate, and the channel remains equally occupied. Although a higher BER would lead to more corrupted frames and thus a decrease in the **effective throughput**, the **utilization ratio** itself remains unaffected because the transmission activity on the channel does not change.

Conclusion: The measured results confirm that **channel utilization is independent of BER** in the absence of retransmission mechanisms. Bit errors cause frame corruption but do not alter the channel's active transmission time. Consequently, within this setup, utilization remains stable across different BER values, while the effective throughput would be the metric affected by error rates.

2.3 End-to-end Delay

2.3.1 Q1: End-to-End Delay vs Interarrival Time (Exponential IAT)

Objective: To analyze how the **mean end-to-end delay** varies with different packet **interarrival times (IAT)** and to identify the congestion threshold where network delay begins to increase sharply.

Results: Figure 7 shows the mean end-to-end delay (mean bit lifetime per packet) as a function of the packet interarrival time. The shaded region represents the **95% confidence interval**. The simulation parameters used in this experiment are summarized below.

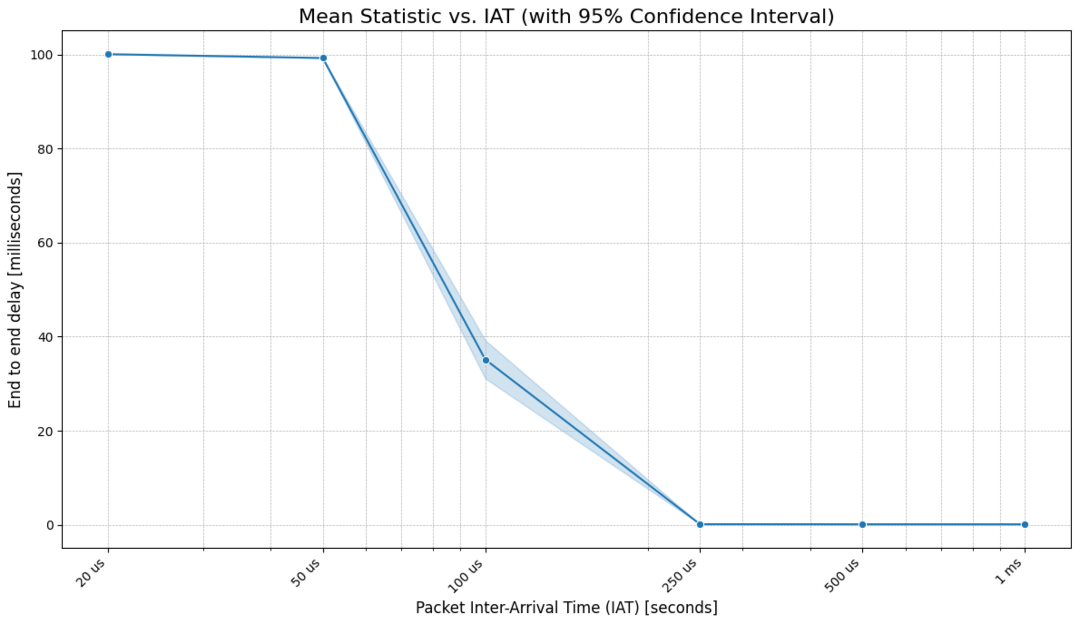

Figure 7: **Mean End-to-End Delay vs Interarrival Time (mean + 95% CI)**

Simulation Parameters:

- Ethernet Bitrate (R): 100 Mbps
- Packet Length (L): 1200 Bytes

Discussion: Figure 7 exhibits a classic congestion behavior characterized by a distinct transition point:

1. **High Congestion Region (IAT $\leq$ 50 μs):** When packets are generated faster than the channel can transmit them, queues accumulate rapidly, and the mean delay stabilizes around **100 ms**. The network operates under heavy congestion in this region.

2. **Threshold Region (IAT $\approx$ 100 μs):** A steep drop in delay is observed near **IAT = 100 µs**, which corresponds closely to the theoretical transmission time per packet:
$$T_{tx} = \frac{L_{\text{bits}}}{R} = \frac{1200 \times 8}{100 \times 10^6} = 96 \ \mu s$$
This confirms that the congestion threshold occurs when the packet arrival interval slightly exceeds the transmission time, allowing queues to dissipate.

3. **Low Congestion Region (IAT $\geq$ 250 μs):** As IAT increases, the offered load decreases, queues empty quickly, and the delay approaches the sum of the minimal propagation and transmission delays.

Table 7: **Representative End-to-End Delay Measurements for Different IAT Values**

IAT (µs)	Measurement Type	Mean Delay
20	End-to-End Delay	100.078 ms
50	End-to-End Delay	99.223 ms
100	End-to-End Delay	29.609 ms
250	End-to-End Delay	135.174 µs
500	End-to-End Delay	113.745 µs
1000	End-to-End Delay	106.655 µs

Conclusion: The results confirm that the mean end-to-end delay increases sharply under high load and converges to a minimal value once the packet generation rate drops below the channel capacity. The transition point around **IAT = 100 µs** accurately matches the theoretical congestion threshold.

2.3.2 Q2: End-to-End Delay vs Number of Switches

Objective: To investigate how the **mean end-to-end delay** varies with the number of intermediate **Ethernet switches** between the source and the destination.

Results: Figure 8 presents the relationship between the number of switches and the corresponding end-to-end delay. The error bars represent the **95% confidence intervals**. The simulation parameters are listed below.

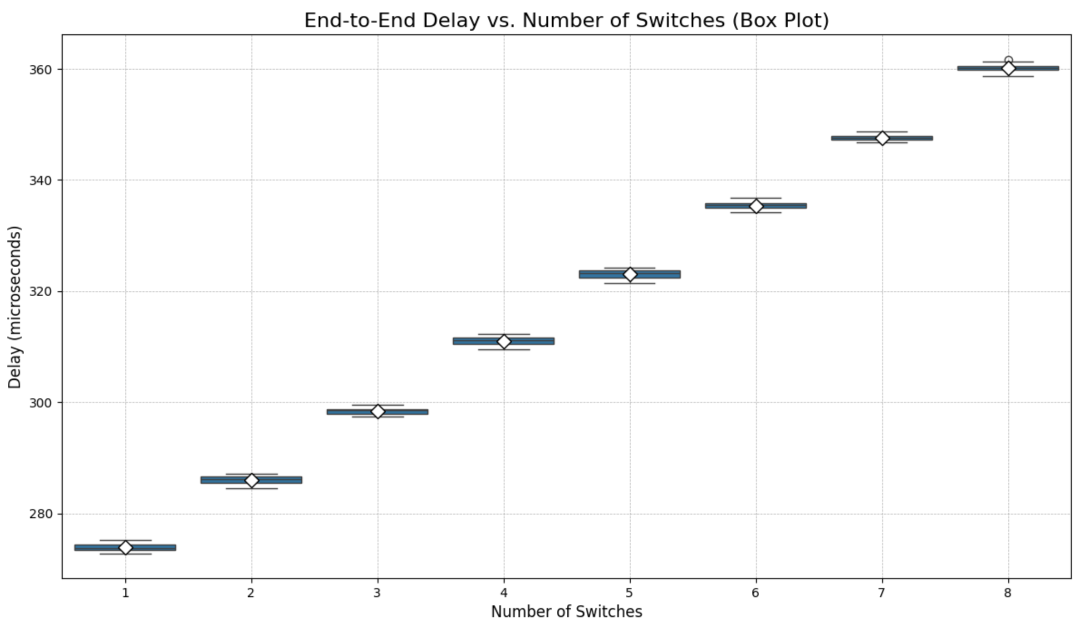

Figure 8: **Mean End-to-End Delay vs Number of Switches (mean + 95% CI)**

Simulation Parameters:

- Production Interval: 500 µs

- Packet Length: 1200 Bytes
- Inter-switch Bitrate: 1 Gbps
- Edge Bitrate (Source and Destination Links): 100 Mbps

Discussion: The delay increases approximately linearly with the number of switches, as each switch introduces additional processing, queueing, and propagation delays. The narrow confidence intervals indicate consistent results across repetitions. The nearly uniform growth pattern shows that all switches contribute a similar delay increment of about **12 µs per hop**, confirming a balanced and stable network setup. This trend demonstrates the scalability limitation of multi-switch Ethernet networks, emphasizing that minimizing intermediate switches is essential for maintaining low-latency communication in real-time applications.

Conclusion: End-to-end delay grows linearly with the number of switches due to the cumulative effect of individual link and processing delays. Each added hop contributes a small but consistent delay increment.

2.3.3 Q3: End-to-End Delay vs Number of Switches (with BER = 10^{-4})

Objective: To observe the impact of introducing a **bit error rate (BER)** on end-to-end delay.

Results: A BER of 1×10^{-4} was applied to the **100 Mbps** source and destination links, while the intermediate links remained error-free. Figure 9 shows the mean end-to-end delay as a function of the number of switches under this configuration.

Discussion: The results demonstrate that introducing a BER of 1×10^{-4} does not cause any measurable change in the end-to-end delay. The delay values closely match those in the no-error scenario, indicating that bit errors only affect packet integrity, not transmission timing. In this setup, since no retransmission or acknowledgment mechanisms are implemented, bit errors lead to corrupted frames but do not increase latency.

Conclusion: The presence of a bit error rate of 10^{-4} has no significant effect on the measured end-to-end delay. Latency in this system is primarily determined by the number of switches and link speeds rather than by occasional bit errors.

2.4 Packet Delay Variation (Jitter)

2.4.1 Q1: Packet Delay Variation (Jitter) vs Interarrival Time (Exponential IAT)

Objective: To examine how the **packet delay variation (jitter)** changes as the packet **interarrival time (IAT)** varies. The aim is to observe how fluctuations in traffic load influence the stability of packet transmission times.

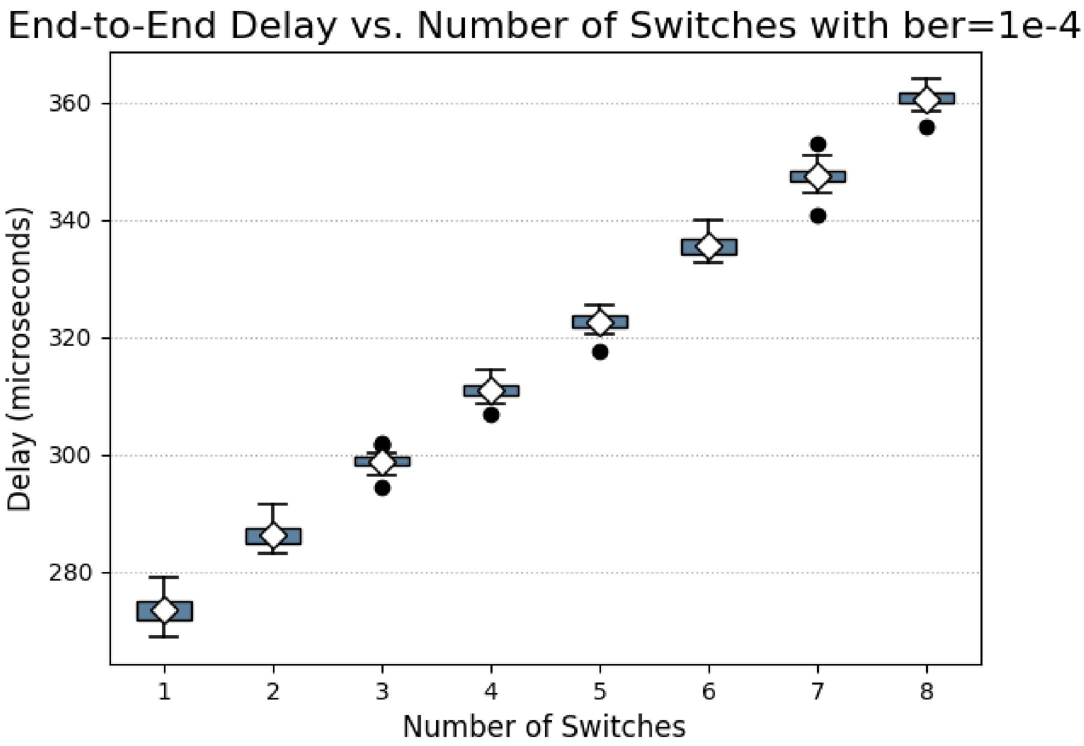

Figure 9: **Mean End-to-End Delay vs Number of Switches with BER** $= 10^{-4}$ **(mean + 95% CI)**

Results: The simulation was performed using exponential interarrival times of **50 µs**, **100 µs**, **200 µs**, **400 µs**, and **800 µs**. For each configuration, the mean jitter and its corresponding **95% confidence interval** were calculated based on 20 repetitions. The results are summarized in Table 8 and illustrated in Figure 10.

Table 8: **Mean Packet Delay Variation (Jitter) for Different Interarrival Times (Mean + 95% CI)**

IAT (µs)	Mean Jitter (µs)	95% CI Lower	95% CI Upper
50	51.37	51.25	51.49
100	1.53	1.28	1.78
200	0.0057	0.0042	0.0072
400	0.00926	0.0079	0.0106
800	0.0162	0.0158	0.0166

Discussion: As seen in Table 8, jitter values are significantly higher at smaller interarrival times. When **IAT = 50 µs**, the mean jitter reaches approximately **51 µs**, indicating high timing variability between consecutive packets due to increased traffic load and potential queueing delays. As the IAT increases to **100 µs**, jitter drops sharply to around **1.5 µs**, and for larger IAT values ($\geq 200\,\mu s$), it remains extremely small and

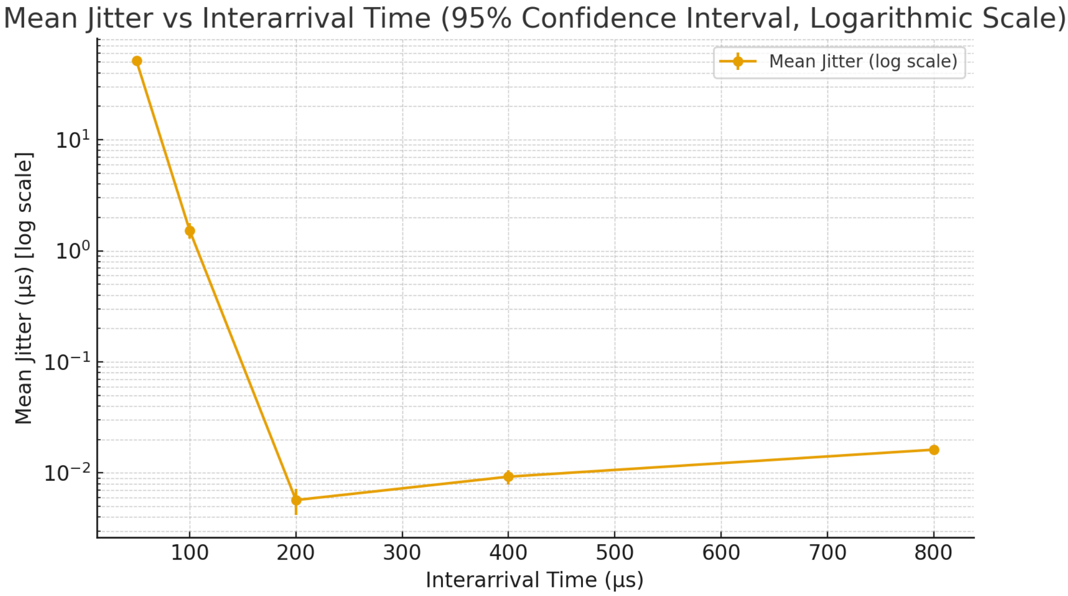

Figure 10: **Packet Delay Variation (Jitter) vs Interarrival Time (mean + 95% CI)**

stable. This inverse relationship clearly reflects that jitter is mainly driven by congestion and buffer delays , both of which are negligible at lower traffic loads.

Conclusion: The results demonstrate that **packet delay variation decreases rapidly as IAT increases**. At low interarrival times, high traffic intensity leads to increased timing fluctuations between packet arrivals, while at larger IAT values, packet transmissions occur at steady intervals with minimal delay variation.

2.4.2 Q2: Packet Delay Variation (Jitter) vs Link Bitrate

Objective: To analyze how the **packet delay variation (jitter)** changes with different **Ethernet link bitrates**. The purpose is to observe how increased link capacity affects the temporal stability of packet transmissions.

Results: The simulations were performed for Ethernet link bitrates of **10 Mbps**, **100 Mbps**, **1 Gbps**, and **10 Gbps**. The mean jitter values and their corresponding **95% confidence intervals** (expressed in microseconds) are listed in Table 9 and visualized in Figure 11.

Table 9: **Mean Packet Delay Variation (Jitter) for Different Link Bitrates (Mean + 95% CI)**

Bitrate (Mbps)	Mean Jitter (µs)	95% CI Lower	95% CI Upper
10	910.512	907.291	913.733
100	1.566	1.344	1.788
1000	0.00020143	0.00020090	0.00020196
10000	0.000021027	0.000020982	0.000021072

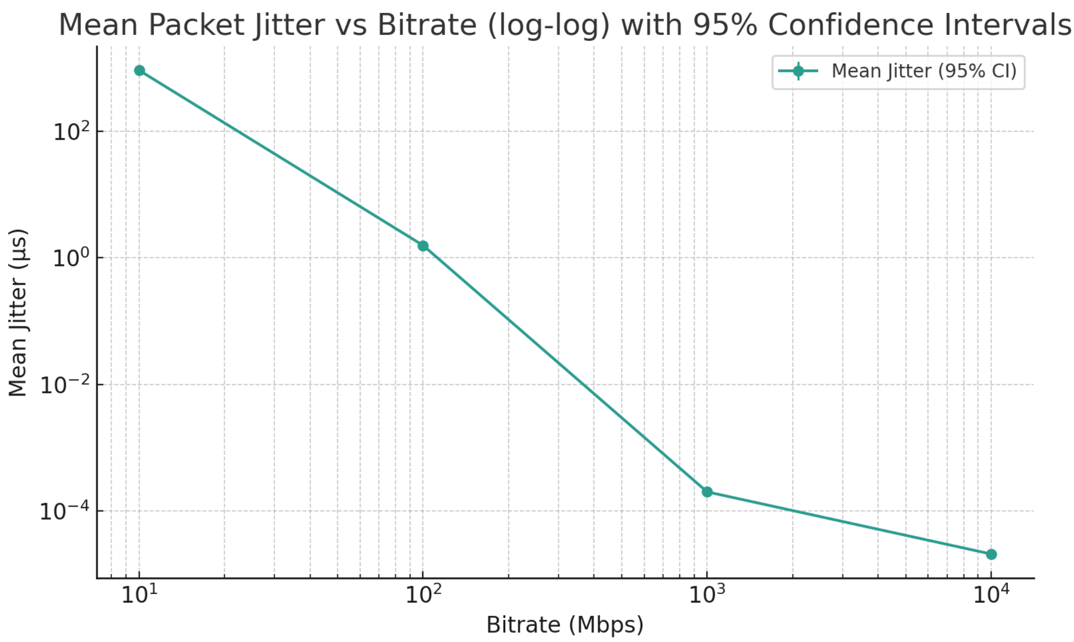

Figure 11: **Packet Delay Variation (Jitter) vs Link Bitrate (mean + 95% CI)**

Discussion: As presented in Table 9, the mean jitter value decreases drastically as the link bitrate increases. At **10 Mbps**, the jitter is extremely high ($\approx \mathbf{910}\,\boldsymbol{\mu s}$), indicating significant variation in packet transmission timing caused by longer serialization and queuing delays. When the bitrate increases to **100 Mbps**, jitter drops to around **1.5 µs**, and it becomes practically negligible at **1 Gbps** and **10 Gbps**, where values fall below **0.001 µs**. These results align with theoretical expectations, as higher bitrates shorten frame transmission times and reduce timing variability between consecutive packets.

Conclusion: The experiment clearly shows that **packet delay variation is inversely proportional to link bitrate**. Low-speed links experience high jitter due to longer transmission and queuing times, while high-speed links provide near-deterministic packet timing, resulting in almost zero jitter at gigabit and higher rates.

2.5 Transmission Time

2.5.1 Q1: Transmission Time vs Ethernet Bitrate

Objective: To investigate how the **mean transmission time** changes with different **Ethernet link bitrates**. The aim is to verify the inverse relationship between link speed and transmission delay.

Results: Figure 12 shows the mean transmission time as a function of Ethernet bitrate. The error bars represent the **95% confidence interval**. The simulation parameters used in this experiment are listed below.
Simulation Parameters:

- Production Interval: 100 µs
- Packet Length: 800 Bytes (truncated by 100 Bytes)

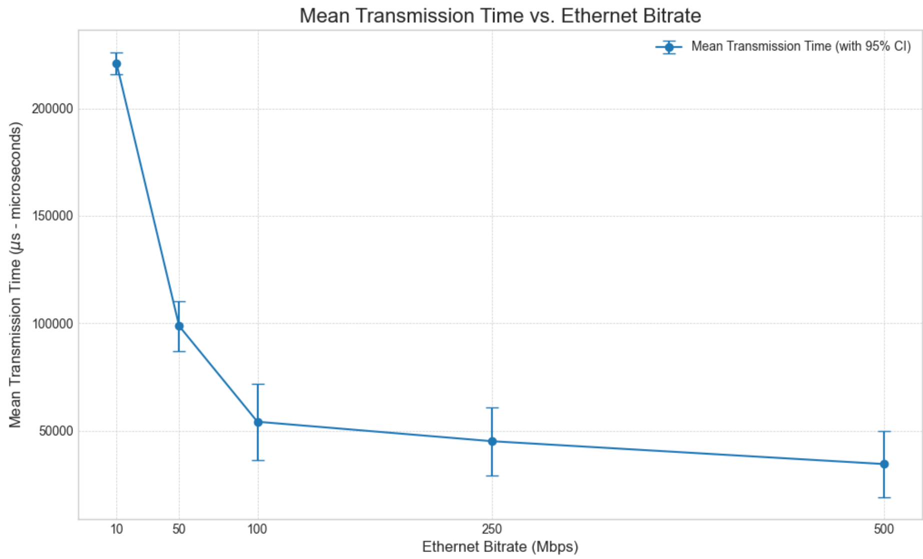

Figure 12: **Mean Transmission Time vs Ethernet Bitrate (mean + 95% CI)**

Discussion: As shown in Figure 12, the mean transmission time decreases sharply as the Ethernet bitrate increases. This inverse relationship reflects that higher bitrates enable packets to be serialized and transmitted faster, directly reducing per-packet transmission delay.

At lower link rates such as **10 Mbps**, the channel capacity is limited, resulting in relatively high transmission times. As the bitrate increases to **50 Mbps** and **100 Mbps**, the transmission time drops significantly, demonstrating substantial improvement in throughput. Beyond **100 Mbps**, the reduction becomes marginal, indicating that other minor delays (e.g., processing or queueing) dominate total latency. The narrow confidence intervals confirm high consistency across repetitions.

Conclusion: Transmission time is inversely proportional to Ethernet bitrate. Increasing the link rate reduces serialization delay and thus lowers the total time required to send each packet.

2.5.2 Q2: Transmission Time vs Packet Length

Objective: To examine how **packet length** affects the mean transmission time when the link bitrate is fixed.

Results: Figure 13 illustrates the relationship between mean transmission time and packet length at a link speed of **100 Mbps**. The error bars represent the **95% confidence intervals**. The simulation parameters are listed below.

Simulation Parameters:

- Production Interval: 100 µs
- Bitrate: 100 Mbps

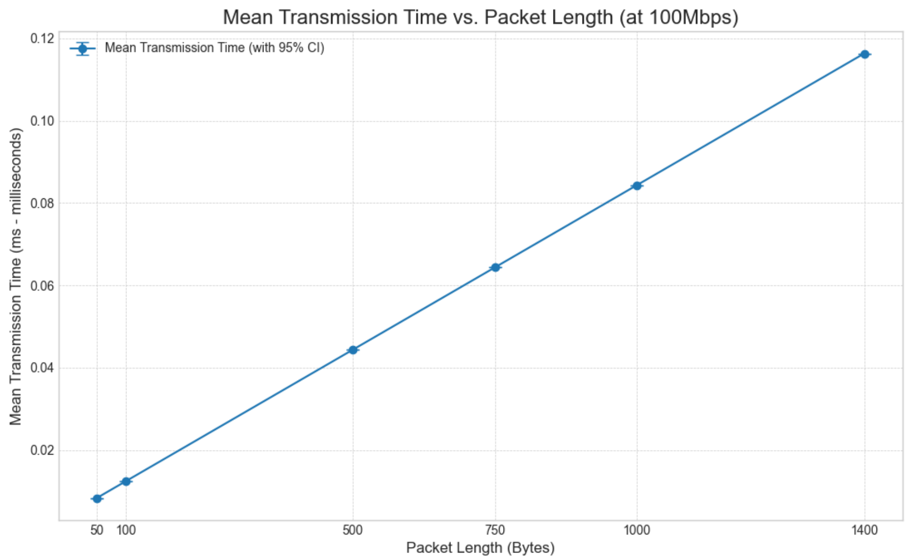

Figure 13: **Mean Transmission Time vs Packet Length (mean + 95% CI)**

Discussion: The figure clearly shows that transmission time increases nearly linearly with packet length. Larger packets contain more bits to be transmitted, and since the bitrate is constant, the serialization time grows proportionally. The 95% confidence intervals remain narrow, confirming that the observed behavior is deterministic and consistent across all simulations.

This observation aligns with the theoretical relationship:

$$T = \frac{L}{R}$$

where T denotes transmission time, L is the packet length (in bits), and R is the link rate (in bits per second).

Conclusion: Transmission time is directly proportional to packet length under constant bitrate conditions. Increasing packet size increases the number of bits that must be transmitted, linearly extending the total transmission duration.

2.5.3 Q3: Transmission Time vs Bit Error Rate (BER)

Objective: To assess the effect of varying the **bit error rate (BER)** on mean transmission time.

Results: The mean transmission time as a function of BER is shown in Figure 14. The error bars indicate the **95% confidence interval**. The simulation parameters are summarized below.

Simulation Parameters:

- Production Interval: 100 µs

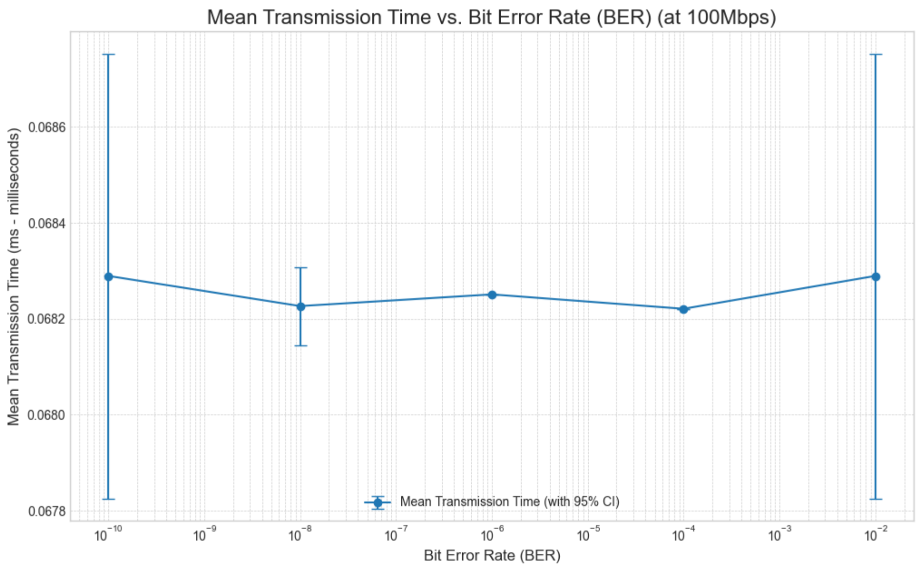

Figure 14: **Mean Transmission Time vs Bit Error Rate (mean + 95% CI)**

- Bitrate: 100 Mbps
- Packet Length: 800 Bytes (truncated by 100 Bytes)

Discussion: The results indicate that the mean transmission time remains almost constant (approximately **0.0682 ms**) across all tested BER levels ranging from 10^{-10} to 10^{-2}. The small variations fall within the bounds of statistical uncertainty and are not indicative of any systematic trend. This stability shows that BER has no measurable effect on transmission duration in the current setup, as bit errors only affect frame integrity, not serialization time.

Since no retransmission or acknowledgment mechanisms are implemented in this simulation, bit errors cause packet corruption but do not introduce additional delay. The minimal fluctuations observed can be attributed to quantization and rounding in measurement precision.

Conclusion: The mean transmission time is unaffected by variations in BER within the tested range. Transmission delay is determined solely by link rate and packet length, while bit errors influence data accuracy rather than timing.

2.6 Propagation Time

2.6.1 Q1: Propagation Time vs Link Length (Custom EthernetLink)

Objective: To measure how the **propagation time** changes with varying **link lengths** by extending the default **EthernetLink** channel. The goal is to verify the expected linear relationship between cable length and propagation delay.

Results: The experiment was conducted by creating a **custom EthernetLink** that allowed the **length** parameter to be modified directly. The link lengths were varied from

10 m to **1,000,000 m**, and the corresponding mean propagation times were recorded. Since propagation delay is a deterministic property of the channel, no variation or confidence interval range was observed. The results are presented in Table 10 and plotted in Figure 15.

Table 10: **Measured Propagation Times for Different Link Lengths**

Length (m)	Mean Propagation Time (s)	Mean (µs)	95% CI (µs)
10	5×10^{-8}	0.05	[0.05, 0.05]
100	5×10^{-7}	0.5	[0.5, 0.5]
1,000	5×10^{-6}	5	[5, 5]
10,000	5×10^{-5}	50	[50, 50]
100,000	5×10^{-4}	500	[500, 500]
1,000,000	5×10^{-3}	5,000	[5000, 5000]

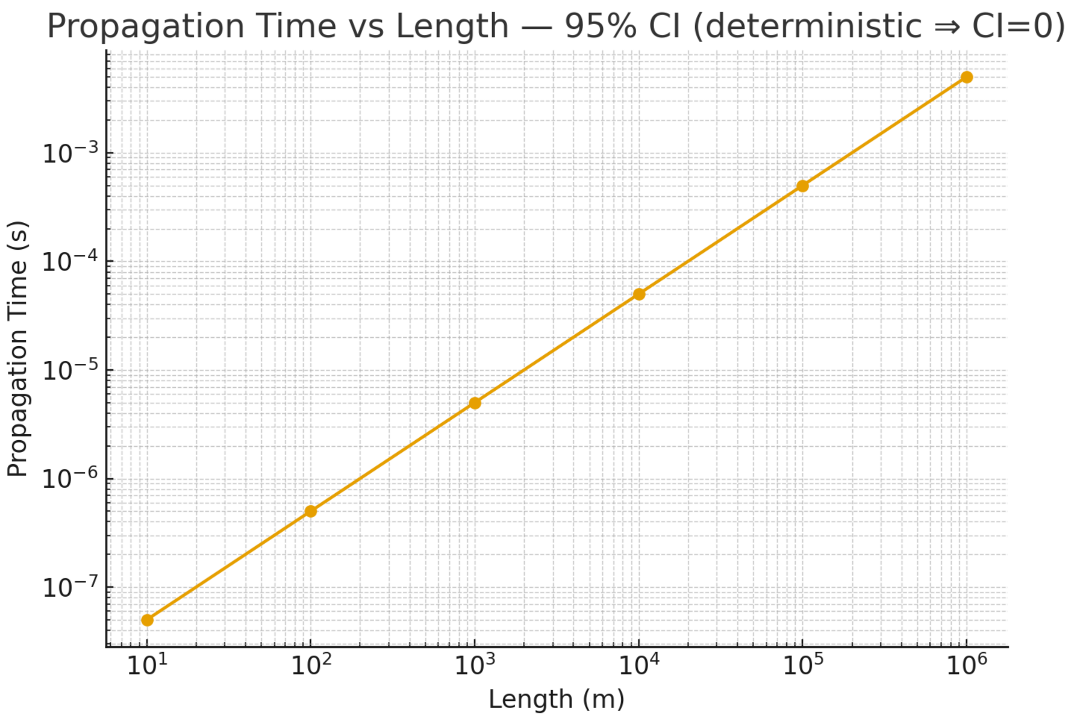

Figure 15: **Propagation Time vs Link Length (Deterministic Linear Relationship)**

Discussion: The measured results show a perfectly linear increase in propagation time with respect to cable length. Specifically, the propagation time increases by a factor of ten for every tenfold increase in link length, which confirms the expected relationship $t_p = \frac{L}{v}$, where L is the cable length and v is the propagation speed (approximately 2×10^8 m/s for copper cables). Because this relationship is purely physical, all repetitions produced identical results, and the 95% confidence intervals have zero width.

Conclusion: The propagation time is directly proportional to the cable length, as predicted by theoretical models. The deterministic results confirm that the extended

EthernetLink channel correctly computes propagation delay based on the defined cable length parameter.

2.6.2 Q2: Propagation Time vs Number of Switches

Objective: To measure how the **total propagation time** changes as the number of **Ethernet switches** between the source and destination increases. The goal is to verify the expected linear increase in propagation delay as additional links are introduced.

Results: The experiment was conducted using a **linear switch topology**, where each added switch introduces an additional link between network devices. The setup used **Eth100M** connections for the source and destination edges, and **Eth1G** connections between intermediate switches. The number of switches was increased from **1** to **8**, resulting in 2 to 9 links in total. For each configuration, the propagation time was measured 20 times. Since propagation delay is deterministic, the standard deviation and 95% confidence interval width were both zero. The results are summarized in Table 11 and shown in Figure 16.

Table 11: **Measured Propagation Time for Different Numbers of Switches (Deterministic Results)**

Switches	Links	Mean Propagation (ns)	95% CI (ns)
1	2	100.0	[100, 100]
2	3	150.0	[150, 150]
3	4	200.0	[200, 200]
4	5	250.0	[250, 250]
5	6	300.0	[300, 300]
6	7	350.0	[350, 350]
7	8	400.0	[400, 400]
8	9	450.0	[450, 450]

Discussion: The results in Table 11 show a perfectly linear relationship between the number of switches and the total propagation time. Each additional switch introduces one more link, contributing a constant propagation delay of approximately **50 ns** per link. As a result, the total propagation time increases linearly from **100 ns** (for one switch) to **450 ns** (for eight switches). This confirms the additive nature of propagation delay across multiple links in a serial topology.

This linear trend directly follows the physical relationship $t_p = \frac{L}{v}$, where t_p represents the propagation time, L is the total cable length, and v is the propagation speed (approximately 2×10^8 m/s for copper cables). Since each link in the topology adds an identical cable segment of equal length, the total propagation delay increases in proportion to the total link length. Therefore, every newly added switch extends the total path length by a fixed amount, causing a corresponding linear increase in overall propagation time.

Conclusion: The experiment verifies that **total propagation delay increases linearly with the number of switches**, since each link contributes a fixed propagation time. The deterministic results match theoretical expectations and validate the accuracy of the propagation delay modeling in the simulation environment.

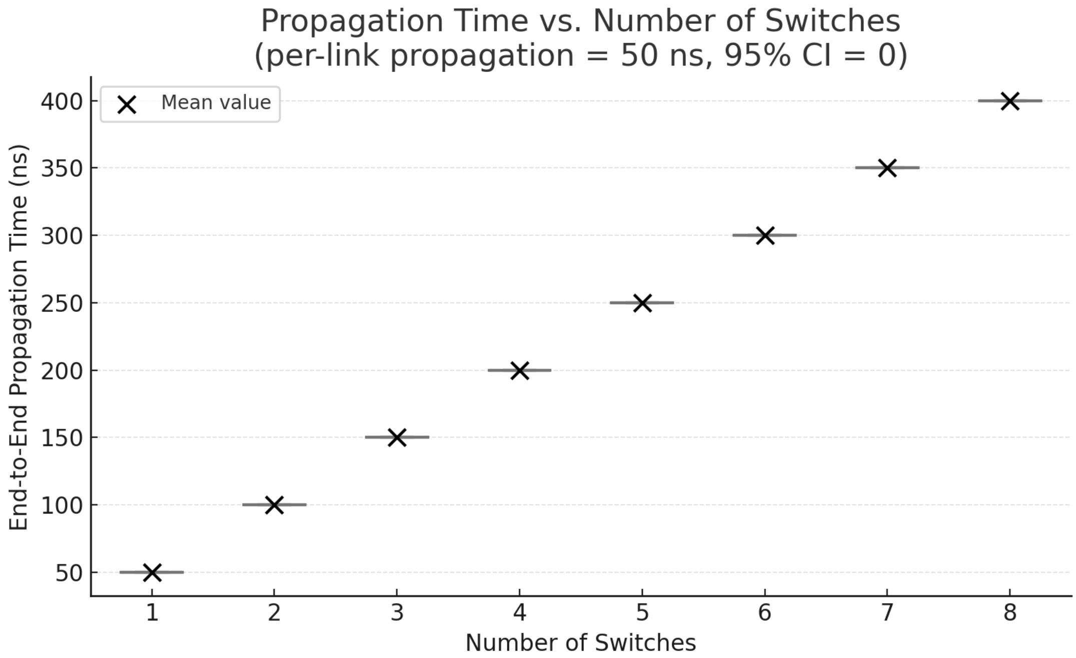

Figure 16: **Propagation Time vs Number of Switches (Linear Growth)**

2.7 Queuing Time

2.7.1 Q1: Queueing Time vs Interarrival Time (Exponential IAT)

Objective: To analyze how the **mean queueing time** changes as the packet **interarrival time (IAT)** varies, and to identify the congestion threshold where queueing delay begins to rise sharply.

Results: Figure 17 shows the mean queueing time as a function of packet interarrival time. The error bars represent the **95% confidence intervals**. The simulation parameters used in this experiment are listed below.

Simulation Parameters:

- Bitrate: 100 Mbps
- Packet Length: 1200 Bytes

Discussion: Figure 17 reveals three distinct operating regions:

1. **High Queueing Time at Small IAT ($\leq$ 50 μs):** At very small interarrival times, the mean queueing delay rises dramatically, peaking near **2000 ms (2 s)**. This represents a state of severe congestion where packets arrive faster than they can be serviced, causing rapid buffer buildup.
2. **Transition Region (Knee Point, around 50–100 µs):** As the IAT increases slightly beyond **50 µs**, the mean queueing time experiences a sharp decline. This point corresponds to the "knee" of the curve, marking the transition from a congested to a stable regime where the system's service rate matches the packet arrival rate.

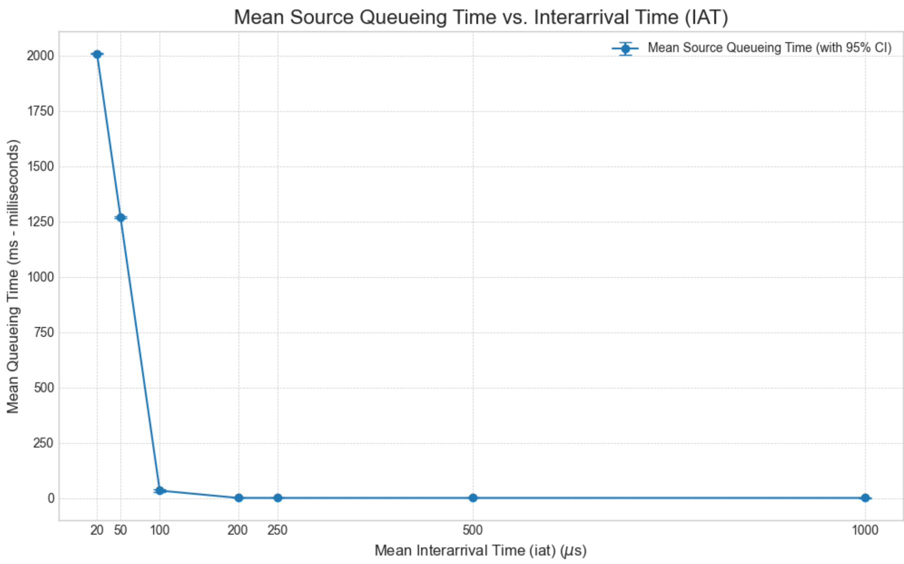

Figure 17: **Mean Queueing Time vs Interarrival Time (mean + 95% CI)**

3. **Stable Region at Large IAT ($\geq$ 100 μs):** For larger interarrival times, the queueing delay approaches zero and remains stable. In this region, packets are processed almost immediately upon arrival, and no significant buffering occurs.

This behavior is consistent with classical queueing theory, where the system utilization ratio ($\rho = \lambda/\mu$) determines queue length and waiting time. When $\rho > 1$, the arrival rate exceeds the service rate, leading to unbounded queue growth. At $\rho = 1$, the system transitions from stable to unstable operation — exactly the threshold observed around **IAT = 50–100 µs**.

Conclusion: The experiment confirms that queueing delay increases sharply under heavy load and becomes negligible at lower traffic intensities. The transition between unstable and stable operation occurs near the point where the packet generation rate equals the service capacity of the 100 Mbps link.

2.7.2 Q2: Queueing Time vs Number of Switches

Objective: To observe how the **mean queueing time** changes as the number of **Ethernet switches** between source and destination increases, and to identify any bottleneck-induced effects.

Results: Figure 18 presents the measured mean queueing time as a function of the number of switches in the network path. The error bars indicate the **95% confidence intervals**. The simulation parameters are provided below.
Simulation Parameters:

- Production Interval: 200 µs

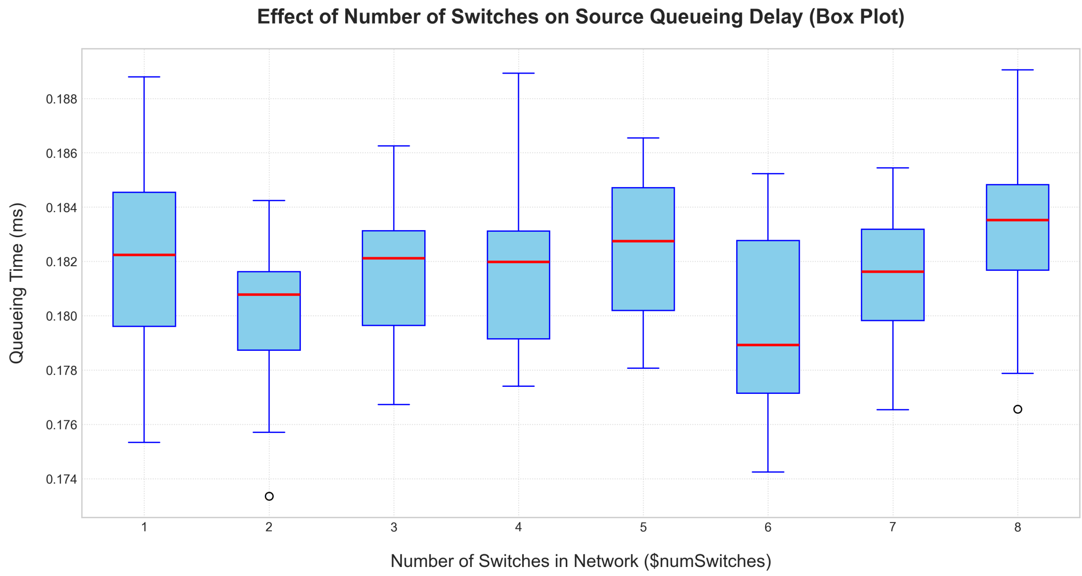

Figure 18: **Mean Queueing Time vs Number of Switches (mean + 95% CI)**

- Packet Length: 1500 Bytes
- Inter-switch Bitrate: 1 Gbps
- Source-to-Switch Bitrate: 100 Mbps

Discussion: The results show that the number of switches has **no significant effect** on queueing delay. The dominant factor influencing queueing is the **bottleneck link**, which in this setup is the 100 Mbps source-to-first-switch connection. Because the intermediate links operate at 1 Gbps (10 times faster), nearly all queue buildup occurs at the network entry point. Subsequent switches contribute negligible queueing time since they handle traffic well below their capacity.

The small variations observed are attributed to statistical noise in the simulations rather than any systematic dependency on the number of switches. This behavior highlights that queueing characteristics are governed primarily by the slowest link in the data path.

Conclusion: Queueing time remains unaffected by the number of switches in the path, as the primary queueing occurs at the source–first-switch bottleneck. The results underline the importance of identifying bottleneck links in network design, as they determine queueing performance regardless of the number of high-capacity hops that follow.

3 Author Contribution

Yusuf Kemahlı installed and configured the **OMNeT++ 6.2.0** environment on his computer, which was later replicated on **Ömer Faruk Şanlıer**'s system to ensure consistent development setups. Both authors collaboratively explored the fundamentals of OMNeT++ and INET, learning the basics of simulation setup and the logic behind each measurement scenario before proceeding with individual tasks.

The first question was completed jointly by both authors, covering the implementation, simulation, and data visualization phases. This initial collaboration provided a shared understanding of the OMNeT++ measurement framework and laid the foundation for consistent methodologies across all subsequent experiments.

After the initial joint work, the tasks were divided as follows: **Yusuf Kemahlı** took responsibility for Questions **2, 4, and 6**, while **Ömer Faruk Şanlıer** handled Questions **3, 5, and 7**. Despite this division, both students continuously discussed their progress, clarified conceptual questions, and supported each other by reviewing simulation setups, code, and results. Whenever a technical or theoretical challenge arose, both authors conducted joint research and troubleshooting sessions to reach a common understanding.

During the reporting phase, each author brought forward their respective findings and written analyses. These were integrated under a unified report structure that reflected a consistent style and depth of explanation. Both authors carefully ensured that all expected learning outcomes of the assignment were addressed, and that each experiment's theoretical and experimental aspects were fully understood by both contributors.

Throughout the process, both authors maintained full visibility of each other's code, results, and analyses. Errors, anomalies, or noteworthy observations were openly discussed, verified through additional runs when necessary, and documented in the final report. As a result, both **Yusuf Kemahlı** and **Ömer Faruk Şanlıer** gained comprehensive insight into every experiment, ensuring that each student understood and contributed to the complete assignment from both a theoretical and practical perspective.