Analyzing the Impact of Malicious Network Traffic: Analysis & Validation

ALEXANDER M. VALENTIN, Kennesaw State University, USA

CCS Concepts: • **Networks → Network simulations**; **Network performance modeling**; *Network performance analysis*; *Network experimentation.*

Additional Key Words and Phrases: Distributed Denial of Service (DDoS), Botnet, Load-Balancing, Distributed Adaptive Routing, Bandwidth Exhaustion, Resource Depletion, Queuing Theory, Network Topology

ACM Reference Format:
Alexander M. Valentin. 2025. Analyzing the Impact of Malicious Network Traffic: Analysis & Validation. *ACM Trans. Access. Comput.* 1, 1, Article 1 (September 2025), 10 pages.

1 Introduction

The discrete-event simulation presented has been designed to analyze how Distributed Denial of Service attacks (DDoS) impact server response time, the ability to serve network traffic, and other network performance metrics. The simulation collects these results by effectively modeling legitimate and malicious network traffic, servers and server responses, client networks, and network routing. In doing so, the simulation comprehensively captures how safeguards such as load-balancing contribute to reduced latency, improved throughput, and decreased server deterioration when under load.

The aim of this report is to provide detailed analysis concerning the relationships between key simulation parameters – including key items such as the number of servers, individual server processing ability, and malicious and legitimate client network capabilities – and their impact on network service outcomes. This area of study is extremely relevant to the overall network system domain, especially considering the rise of DDoS attacks against large enterprise services year-over-year. According to Cloudflare's DDoS threat report for 2025 Q2, "Layer 3/Layer 4 (L3/4) DDoS attacks plunged 81% quarter-over-quarter to 3.2 million, while HTTP DDoS attacks rose 9% to 4.1 million. Year-over-year changes remain elevated. Overall attacks were 44% higher than 2024 Q2, with HTTP DDoS attacks seeing the largest increase of 129% YoY." [2] Given that the simulation implementation specifically models an HTTP flood attack, it is imperative that sensitivity analysis, scenario analysis, statistical analysis, and simulation validation be performed in order to ensure that the simulation accurately captures the real world behavior of this growing threat.

This report is directly built upon the data findings, results, and preliminary connections established in Milestone 3. The ten distinct simulation executions – outlined by scenarios such as Intense HTTP Flood, Mixed Workload Stress, Large Infected Client Assault, and Load Balancing Pressure – provide an excellent foundation for simulation analysis and validation, and remains well-positioned for fine-tuned simulation executions in the future. By leveraging previous findings and fine-tuned simulation executions, in combination with comprehensive data reporting and visualization tooling, we can effectively identify patterns and relationships among the configuration parameters and the network outcomes. In all, the findings outlined in this report shall ensure that the simulation accurately captures the real world behavior of this growing threat, and properly identifies key relationships that can potentially help quell the impact of network attacks in the future.

Author's Contact Information: Alexander M. Valentin, Kennesaw State University, Marietta, Georgia, USA.

2025. ACM 1936-7236/2025/9-ART1
https://doi.org/

2 Parameter Analysis

This section is designed to provide sensitivity analysis by which we explore how changes to the configuration parameters impact the behavior and outcomes of the malicious network traffic simulation. Furthermore, this section shall identify which parameters have the most influence on the system's behavior. While the simulation produces a comprehensive report of execution outcomes, the key outcome being observed is overall request success rate, as it provides a reliable means of gauging system health and service availability.

Ten finely-tuned simulation executions have been performed, with the addition of a baseline execution as a point of comparison. Across these executions, the key configurable parameters that have been varied include:

- NUM_SERVERS
- PROCESSING_POWER
- MAX_REQUEST_QUEUE_LENGTH
- MALICIOUS_TRAFFIC_RATE
- LEGITIMATE_CLIENT_COUNT

The baseline configuration for these executions was designed to model an enterprise system under moderate to increased load. In this execution, a majority of requests are served with minor drops and lapses in service. The baseline configuration has an overall success rate of 87.73%. This configuration exhibits this kind of behavior in order to better identify which changes in parameters alleviate network pressure or otherwise send the network into an outage. The baseline values for the configurable parameters is as follows:

- NUM_SERVERS = 8
- PROCESSING_POWER = 250
- MAX_REQUEST_QUEUE_LENGTH = 400
- MALICIOUS_TRAFFIC_RATE = 15
- LEGITIMATE_CLIENT_COUNT = 400

These numbers allow us to calculate the % change in input when assessing the sensitivity for each parameter. Furthermore, a high and low execution was performed for each parameter while the remaining baseline parameters remained the same. This was done in order to comprehensively assess the simulation's sensitivity across a wide range of potential system inputs, and to provide practical insights surrounding the data. The sensitivity values for each parameter are as follows:

Table 1. Sensitivity Analysis Results Demonstrating High Network Infrastructure Sensitivity.

Parameter	Case	Success %	% Change Input	% Change Output	Sensitivity
NUM_SERVERS	Low (6)	45.58%	-25.00%	-48.05%	1.9218
	High (10)	99.35%	25.00%	13.25%	0.5298
PROCESSING_POWER	Low (200)	56.44%	-20.00%	-35.67%	1.7833
	High (300)	99.70%	20.00%	13.64%	0.6822
MAX_REQUEST_QUEUE_LENGTH	Low (300)	69.41%	-25.00%	-20.88%	0.8353
	High (500)	90.64%	25.00%	3.32%	0.1327
MALICIOUS_TRAFFIC_RATE	Low (10)	99.51%	-33.33%	13.43%	-0.4028
	High (25)	38.45%	66.67%	-56.17%	-0.8426
LEGITIMATE_CLIENT_COUNT	Low (200)	93.37%	-50.00%	6.43%	-0.1286
	High (700)	49.11%	75.00%	-44.02%	-0.5870

Of the sensitivity calculations outlined above, the parameters that exhibit the highest sensitivity with respect to simulation outcomes are NUM_SERVERS and PROCESSING_POWER. It is clear from the calculations that the system is highly vulnerable to drops in overall request success rate when either of these two parameters are decreased. However, it is important to note that this sensitivity relationship is not symmetric and does not behave in the same fashion when increasing either of these parameters. As exhibited in the calculations above, the sensitivity for the high value for NUM_SERVERS is 0.5298 and the high value for PROCESSING_POWER is 0.6822. These values demonstrate that changes in the output are less than changes in the input, illustrating how increasing these parameters only marginally increase the overall request success rate. Ultimately, these observations build on the sentiment expressed in Milestone 3 that network processing capabilities play the largest role in simulation outcomes.

3 Scenario Testing

3.1 Different Conditions Tested

Although comprehensive and robust simulation scenarios were captured in Milestone 3, these simulations were more nuanced executions designed to explore specific aspects of the system's performance and capability. In contrast, the simulation executions outlined in this area will be fine-tuned to model real-world network scenarios, in an attempt to tailor results to authentic network situations and illustrate a clearer story surrounding simulation executions. Among these execution scenarios, user configurable parameters such as CPU utilization health weight (0.4), queue utilization health weight (0.6), increased utilization threshold (0.70), high utilization threshold (0.85), and critical utilization threshold (0.95) will remain the same in order to preserve the validity of execution outcomes and ensure that differences in simulation outcomes are directly attributable to simulation pipeline changes, as opposed to variations in calculation weights. Furthermore, simulation duration (250), interval data polling (5), and interval output polling (10) also remained the same throughout the course of simulation executions in order to maintain the same level of data granularity. The rest of the parameters for each simulation scenario are as follows:

Scenario 1: Daily Traffic with Mixed Malicious Traffic

- NUM_SERVERS: 15
- REQUEST_TIMEOUT: 3
- SERVER_TIMEOUT: 2
- PROCESSING_POWER: 500
- MAX_REQUESTS_CONCURRENT: 40
- MAX_REQUESTS_QUEUE_LENGTH: 800
- LEGITIMATE_TRAFFIC_RATE: 5
- LEGITIMATE_CLIENT_COUNT: 700
- LEGITIMATE_LOAD_SIZE_LOWER: 1
- LEGITIMATE_LOAD_SIZE_UPPER: 3
- MALICIOUS_TRAFFIC_RATE: 20
- MALICIOUS_CLIENT_COUNT: 200
- MALICIOUS_LOAD_SIZE_LOWER: 12
- MALICIOUS_LOAD_SIZE_UPPER: 20

Scenario 2: Holiday Peak with Moderate Malicious Traffic

- NUM_SERVERS: 15
- REQUEST_TIMEOUT: 3
- SERVER_TIMEOUT: 2
- PROCESSING_POWER: 500
- MAX_REQUESTS_CONCURRENT: 40
- MAX_REQUESTS_QUEUE_LENGTH: 800
- LEGITIMATE_TRAFFIC_RATE: 5
- LEGITIMATE_CLIENT_COUNT: 900
- LEGITIMATE_LOAD_SIZE_LOWER: 1
- LEGITIMATE_LOAD_SIZE_UPPER: 3
- MALICIOUS_TRAFFIC_RATE: 28
- MALICIOUS_CLIENT_COUNT: 250
- MALICIOUS_LOAD_SIZE_LOWER: 12
- MALICIOUS_LOAD_SIZE_UPPER: 20

Scenario 3: Coordinated HTTP Flood

- NUM_SERVERS: 15
- REQUEST_TIMEOUT: 3
- SERVER_TIMEOUT: 2
- PROCESSING_POWER: 500
- MAX_REQUESTS_CONCURRENT: 40
- MAX_REQUESTS_QUEUE_LENGTH: 800
- LEGITIMATE_TRAFFIC_RATE: 5
- LEGITIMATE_CLIENT_COUNT: 700
- LEGITIMATE_LOAD_SIZE_LOWER: 1
- LEGITIMATE_LOAD_SIZE_UPPER: 3
- MALICIOUS_TRAFFIC_RATE: 30
- MALICIOUS_CLIENT_COUNT: 350
- MALICIOUS_LOAD_SIZE_LOWER: 12
- MALICIOUS_LOAD_SIZE_UPPER: 20

3.2 Results Under Each Scenario

Table 2. Scenario 1 Results: Daily Traffic with Mixed Malicious Traffic

Metric	Value
Traffic Summary	
Total Generated Requests	1,874,800
Total Served Requests	1,866,063
Total Dropped Requests	8,737
Drop Breakdown	
Dropped - Queue Full	17
Dropped - Timeout	140
Dropped - High Load	740
Dropped - No Server	7,840
Performance Metrics	
Overall Success Rate	99.53%
Overall Drop Rate	0.47%
Server Health Metrics	
Average CPU Utilization	63.8%
Average Queue Utilization	1.4%
Average Health Score	0.71
Average Offline Percentage	1.3%

Table 3. Scenario 2 Results: Holiday Peak with Moderate Malicious Traffic

Metric	Value
Traffic Summary	
Total Generated Requests	2,875,000
Total Served Requests	2,180,057
Total Dropped Requests	694,943
Drop Breakdown	
Dropped - Queue Full	402
Dropped - Timeout	10,893
Dropped - High Load	150,831
Dropped - No Server	532,817
Performance Metrics	
Overall Success Rate	75.83%
Overall Drop Rate	24.17%
Server Health Metrics	
Average CPU Utilization	82.0%
Average Queue Utilization	65.3%
Average Health Score	0.260
Average Offline Percentage	1.0%

Table 4. Scenario 3 Results: Coordinated HTTP Flood

Metric	Value
Traffic Summary	
Total Generated Requests	3,500,000
Total Served Requests	1,148,861
Total Dropped Requests	2,351,139
Drop Breakdown	
Dropped - Queue Full	1,332
Dropped - Timeout	1,317
Dropped - High Load	29,771
Dropped - No Server	2,318,719
Performance Metrics	
Overall Success Rate	32.82%
Overall Drop Rate	67.18%
Server Health Metrics	
Average CPU Utilization	56.1%
Average Queue Utilization	28.6%
Average Health Score	0.584
Average Offline Percentage	1.1%

3.3 Comparison and Insights

The execution scenarios outlined in this section are inspired by the different network conditions enterprise service clusters face year-round. The first scenario is designed to simulate daily traffic to an ecommerce service such as Amazon.com, with a low volume of malicious traffic mixed in with high legitimate traffic in order to simulate daily operations for a service of this scale. This scenario allows us to establish a comparative baseline for the other scenarios, as it captures the expected operating conditions of the service for any given day. The next simulation scenario is designed to capture this same service's behavior when faced with holiday peaks, characterized by an increased presence of both legitimate and malicious traffic. Finally, the last simulation scenario is designed to capture a large, coordinated attack against the service, characterized by an increase in malicious traffic during what would otherwise be normal operating conditions.

Under normal operating conditions, there were a total of 1,874,800 requests generated across 250 simulated seconds. Furthermore, there were 700 legitimate and 200 malicious clients in this instance, with a legitimate traffic rate of 5 requests per second and a malicious traffic rate of 20 requests per second. Under these conditions, the scenario performed well within what was expected for simulated daily traffic. The simulation results provide valuable insight concerning the scenario, as illustrated by a 99.53% request success rate and a 0.47% request drop rate combined with minimal server downtime and zero timeout drops. These scenario results establish a realistic base of comparison for an average day of service by which we can begin to establish insights surrounding the data as a whole.

Under the holiday peak scenario, there were a total of 2,875,000 requests generated across the same simulated time frame, an increase of 53.3%. Furthermore, the number of legitimate clients increased 29% to 900 while the number of malicious clients rose to 250. There was an associated rise in malicious traffic to 28 requests per second while legitimate traffic continued at the same rate as the baseline scenario. Under these conditions, the holiday peak scenario performed within what was expected for a system under duress. The overall request success rate dropped from 99.53% to 75.83% within the time period, while the drop rate rose from 0.47% to 24.17%. Despite these metrics indicating a drop in service availability, server downtime hovered around the 1% mark in both instances. This illustrates that request drops during the holiday scenario were not induced by the network being overwhelmed and ultimately shutting down due to the influx of traffic but were instead the result of request process timeouts due to the load size exceeding the processing capabilities of the servers in the cluster. This effectively captures that there are enough servers to extinguish the threat of an outage, but the servers are too slow to be effective in servicing network traffic.

The coordinated HTTP flood scenario posed the greatest challenge for the target service. Under this scenario, there were a total of 3,500,000 requests generated, an increase of 86.7% across the simulated time frame. Furthermore, the number of legitimate clients remained the same as the baseline at 700, while the number of malicious clients rose to 350, an increase of 75%. This scenario had the steepest growth in malicious traffic rate, rising 50% from the baseline to 30 requests per second while the legitimate traffic rate remained the same as the baseline at 5 requests per second. These scenario parameters were designed to capture the expected traffic for the target service, in combination with a coordinated HTTP flood DDoS attack. Since the scenario was designed to capture expected traffic in combination with underlying malicious traffic, the parameters concerning legitimate traffic are unchanged from the baseline. The HTTP flood scenario performed within what was expected for this style of attack, as service availability significantly diminished. The overall request success rate dropped from 99.53% to 32.82%, while the overall drop rate increased from 0.47% to 67.18%. It is imperative to note that of the 3,500,000 requests generated during this scenario, 2,318,719 were dropped as a direct result of no server being available. This indicates that unlike the holiday peak scenario where servers simply didn't have the ability to process all of the incoming requests efficiently, in this instance, the 75% increase in malicious clients and 50% increase in malicious request rate warranted 66.2% of requests being dropped due to an outage.

Looking at these three scenario outcomes, it is clear that the simulation is disproportionately affected by malicious traffic as opposed to legitimate traffic. This is illustrated when looking at the scenario outcomes across the holiday peak and HTTP flood scenarios. Despite the holiday peak scenario having 100 more total clients than the HTTP flood scenario, the additional 100 malicious clients as well as the 7.1% difference in attack rate in the flood scenario triggered a 43.01% reduction in overall success rate, even when considering the 200 client reduction in legitimate clients between the holiday peak scenario and coordinated attack scenario. Furthermore, the data suggests that the modes of network failure have differing levels of severity as evidenced by the overall success rate between the holiday scenario and HTTP flood scenario. While the system is under duress in the holiday scenario, the data demonstrates that the network can still service a majority of the incoming requests and the overall network impact is not as severe. However, as evidenced by the flood scenario, there is a tipping point by which malicious request generation becomes severe to overall network service availability causing a critical decline in overall request processing.

4 Validation

Validation ensures that the simulation implementation produces reasonable results that align with real-world expectations. Fortunately, there is an extensive amount of public data concerning enterprise network systems and their performance.

It is imperative to note that as outlined in Section 2.4 of Milestone 3, the main issue when initially collecting simulation results was memory exhaustion and unexpected simulation outcomes. When first implementing research-driven configuration parameters, it became apparent that utilizing enterprise-scale infrastructure parameters would not be feasible, as these configuration values generated results that were far too large to store and accurately manage. In many cases, final values would appear as zero or negative numbers, indicating overflow and memory allocation issues. As a result, extensive research was conducted in order to establish infrastructure ratios that accurately captured the real-world relationships of core simulation components. In the final implementation, many of these values have been scaled down roughly 100:1 while maintaining their relative sizes to one another. According to an industry report published by OneChassis, enterprise data centers typically house between 500 and 5,000 servers, while in the simulation implementation the number of servers can be between five and fifty.

When comparing outputs for refined simulation scenarios against network traffic data provided by Wikimedia's Grafana installation, in combination with OneChassis' infrastructure report, it becomes clear that the simulation is accurate in modeling real-world network behavior and that simulation parameter values are within reasonable ranges. Wikimedia is the nonprofit organization that hosts Wikipedia, and their service provides a comprehensive and comparable baseline in order to help validate the network simulation's outcomes. According to Wikimedia's Grafana installation [7], as of November 12th the service receives anywhere between 100,000 and 200,000 requests per second across a twenty-four hour period. Furthermore, according to Wikipedia's own technical FAQ, the service is maintained by approximately 350 servers. With this information, we can determine that each server receives anywhere from 286 to 572 requests per second. Scaling these ranges to the simulated fifteen server network, the total request generation rate within the daily traffic scenario should fall between 4,290 and 8,580 requests per second. Although not explicitly listed as a simulation outcome, the total requests per second – 1,874,800 generated requests over a 250 unit time frame – for the scenario is 7,500. This illustrates that the request generation is accurate for a simulation of this size, and values are properly scaled for simulating daily traffic for an enterprise service such as Wikipedia. Furthermore, Wikimedia's Grafana dashboard reports an average of 4.87 errors per second across the same twenty-four hour period, an error rate of approximately 0.003%. While the baseline simulation's overall drop rate of 0.47% is higher than what is reported by Wikimedia, this difference can be attributed to advanced network infrastructure and architectural decisions in combination with the network implications listed earlier in the reading. Despite this difference, the exceptionally low values of drop rates are still a testament to the validity of the implementation approach and serve to reflect lapses in service when under load. In all, these values illustrate that the model is behaving reasonably.

While output comparison has been a crucial and effective tool for validating the simulation, face validation also supports that the final implementation is correct. As outlined in Milestone 1, the simulation design and core architectural components have been carefully crafted according to academic literature such as:

- *DDoS Attack and Detection Methods in Internet-Enabled Networks* by Kazeem A. Adnan A. and Anish M. [1]
- *Botnets Multiply and LevelUp* by Chris C., published by NETSCOUT SYSTEMS, INC [3]
- *BOTNET: Lifecycle, Architecture and Detection Model* by Kazeem A. Adnan A. and Anish M. [4]
- *Queuing Analysis* by William S. [6]
- *Routing Algorithms: A Review* by Ujjwal S., Vikas K. and Shubham K., under the guidance of Dr. Jayasheela [5]

The literature above has served as an extensive means of validation and has been instrumental in ensuring that the final simulation behaves as accurately as possible within the context of network behavior. To illustrate, the core architectural design for the Botnet component within the simulation is directly modeled after the centralized C&C topology outlined in Divya N., Pooja W., Sanjay S., and Deepak K's academic paper, furthermore, the server processing design is heavily inspired by William Stalling's findings in his paper Queuing Analysis. These literature driven decisions encompass the entirety of the simulation's architectural design, ensuring that the final implementation is as robust and comprehensive as possible given simulation's limitations.

While the current simulation implementation provides a comprehensive and valid means of assessing malicious network traffic, there are multiple key limitations within the model that must be addressed. As outlined in Milestone 1, request response time is not impacted by the network topology and the communication time between network components is zero. While this is sufficient for conducting executions in the simulated network environment, network topology in real-world systems can have a pronounced effect on network response times and processing capabilities. To further this point, request routing algorithms and traffic safeguards in real-world network systems often utilize metrics concerning network topology in order to drive routing and request handling decisions, in this simulation implementation, the only metric driving routing decisions is the health of the servers within the cluster. Beyond limitations induced by network topology decisions, the servers within this simulation are all identical. While this is sufficient for the simulated network environment, enterprise services may not maintain homogeneous network servers due to their individual business needs.

5 Statistical Summary

5.1 Key Statistics from Simulation Executions

Table 5. Statistical Summary of Simulation Results Demonstrating High Data Variability.

	Mean	**Std. Deviation**	**Min**	**Max**
Overall Success Rate (%)	75.39%	22.95%	38.45%	99.70%
Overall Drop Rate (%)	24.60%	22.95%	0.30%	61.55%
Drops - No Server	138,278.73	166,806.37	0	467,140
Drops - Timeout	11,827.27	14,030.68	343	40,570

5.2 Variability in Results

Given the key statistics across the eleven distinct parameter analysis executions, metrics concerning success rate, drop rate, server availability timeouts, and request processing timeouts demonstrate significant variability between runs. Success rate ranged anywhere from 38.45% to 99.70%, with an average success of 75.39% and a standard deviation of 22.95%. This indicates that there was severe variability in performance concerning the success rate across the eleven executions. Furthermore, this sentiment is echoed when observing the overall drop rate across these executions. The drop rate maintained the exact same standard deviation of 22.95%, with an average drop rate of 24.60% and values ranging between 0.30% and 61.55%. For discrete request service counts, server availability drops ranged anywhere from 0 to 467,140 dropped requests. As such, the standard deviation was 166,806.37 for server availability drops, and the average across runs was 138,278.73. Finally, the average drops induced by a process timeout was 11,827 across eleven executions, with a minimum of 343 and a maximum of 40,570. As such, the standard deviation for this metric was 14,030.68 across the eleven executions.

5.3 Confidence in Findings

A confidence interval gives a range where the true average likely falls, accounting for variability in the data. For a 95% confidence interval with n runs, mean $\bar{x}$, and standard deviation s, the calculation is:

$$CI = \bar{x} \pm 1.96 \times \frac{s}{\sqrt{n}}$$

Given this equation, the confidence intervals for each metric are as follows:

- Overall Success Rate (%): [61.82%, 88.96%]
- Overall Drop Rate (%): [11.03%, 38.17%]
- Drops (No Server): [39,724, 236,833]
- Drops (Timeout): [3,536, 20,119]

6 Conclusion and Final Remarks

In all, the findings outlined in this report have presented a comprehensive analysis of simulation outcomes, identified clear data relationships and patterns, and has successfully validated the implementation approach as a whole. Furthermore, it is clear that the malicious network traffic simulation is an effective tool for modeling the impact of malicious network traffic on network systems. Through the analysis and validation methods outlined in this report, several important findings have been identified.

Through extensive sensitivity analysis it has become clear that the simulation is most sensitive to changes in network scale and processing capabilities, specifically NUM_SERVERS and PROCESSING_POWER. Reductions in these parameters saw the greatest impacts on overall request success rates, as evidenced by their respective 1.9218 and 1.7833 sensitivities. On the other hand, increases in these parameters were not nearly as pronounced and did not have a significant impact on simulation outcomes. This relationship ultimately captures a nuance concerning diminishing returns when expanding network infrastructure.

Furthermore, through analyzing execution outcomes when conducting scenario analysis, a key relationship between malicious and legitimate traffic arose. As stated in Section 3.3, when analyzing the three scenario outcomes, it is clear that the simulation is disproportionately affected by malicious traffic as opposed to legitimate traffic. This is illustrated when looking at the scenario outcomes across the holiday peak and HTTP flood scenarios. Despite the holiday peak scenario having 100 more total clients than the HTTP flood scenario, the additional 100 malicious clients as well as the 7.1% difference in attack rate triggered a 43.01% reduction in overall success rate, even when considering the 200 client reduction in legitimate clients between the holiday peak scenario and coordinated attack scenario. This illustrates that the nature of the traffic plays a significant role in the overall network impact, a direct result of the difference in mean request payload size.

These findings are further solidified via extensive validation and real-world data comparison. As outlined in Section 4, when comparing the daily traffic outcomes with real data gathered from Wikimedia's Grafana installation, it is clear that the simulation implementation closely aligns with what is expected for a real enterprise network system. Wikipedia's traffic, at this simulation's scale, would be anywhere between 4,290 and 8,580 requests per second across fifteen servers. As documented in this report, the daily traffic scenario generated 7,500 requests per second, falling directly within the expected range. Furthermore, the daily traffic's overall request success rate was 99.53% and the overall drop rate was 0.47%, with the marginal differences between the expected and actual values being attributed to differences in network infrastructure in combination with simulation limitations.

While the simulation exhibits multiple key strengths like accurately modeling network behavior and the attacking behavior of a hostile network, incorporating realistic network safeguards, and external validation suggesting that the model is operating within what is realistic and expected for a real-world service, there are still limiting factors surrounding the simulation's implementation. As outlined in Milestone 1, request response time is not impacted by the network topology and the communication time between network components is zero. While this is sufficient for conducting executions in the simulated network environment, network topology in real-world systems can have a pronounced effect on network response times and processing capabilities. Furthermore, request routing algorithms and traffic safeguards in real-world network systems often utilize metrics concerning network topology in order to drive routing and request handling decisions. However, in this implementation the only metric driving routing decisions is individual server health. Finally, the servers within this simulation are all identical. While this is sufficient for the simulated network environment, enterprise services may not maintain homogeneous network servers due to their individual business needs. In all, while the simulation is extremely robust and gives users a fantastic basis of understanding for the impact of malicious traffic, the implementation is slightly limited with regard to network architecture and topology.

There are definitely areas for improvement to enhance the model to an even higher real-world network standard. However, the malicious network traffic simulation in its current state serves as an excellent tool for gathering a foundational understanding of the impact of network attacks and analyzing overall network performance.

References

[1] Kazeem B. Adedeji, Adnan M. Abu-Mahfouz, and Anish M. Kurien. 2023. DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges. *Journal of Sensor and Actuator Networks* 12, 4, Article 51 (2023). doi:10.3390/jsan12040051

[2] Cloudflare. 2025. DDoS Threat Report for 2025 Q2. https://radar.cloudflare.com/reports/ddos-2025-q2. Accessed: 2025.

[3] Chris Conrad. 2022. *Botnets Multiply and Level Up.* DDoS Threat Intelligence Report. NETSCOUT SYSTEMS, INC. Report Number: SEC04.

[4] Divya Nagpal, Pooja Wadhwa, Sanjay Singh, and Deepak Kumar. 2023. BOTNET: Lifecycle, Architecture and Detection Model. In *Proceedings of the International Conference on Cyber Security and Cryptography.* 112–125.

[5] Ujjwal Sharma, Vikas Kumar, Shubham Kumar, and Dr. Jayasheela. 2024. Routing Algorithms: A Review. *arXiv preprint* (2024). https://arxiv.org/abs/2403.11228v1

[6] William Stallings. 2022. *Queuing Analysis in Computer Networks.* Pearson Education.

[7] Wikimedia Foundation. 2025. Home: W Wiki Status - Wikimedia Grafana Dashboard. https://grafana.wikimedia.org/d/O_OXJyTVk/home-w-wiki-status. Accessed: 2025-11-12.