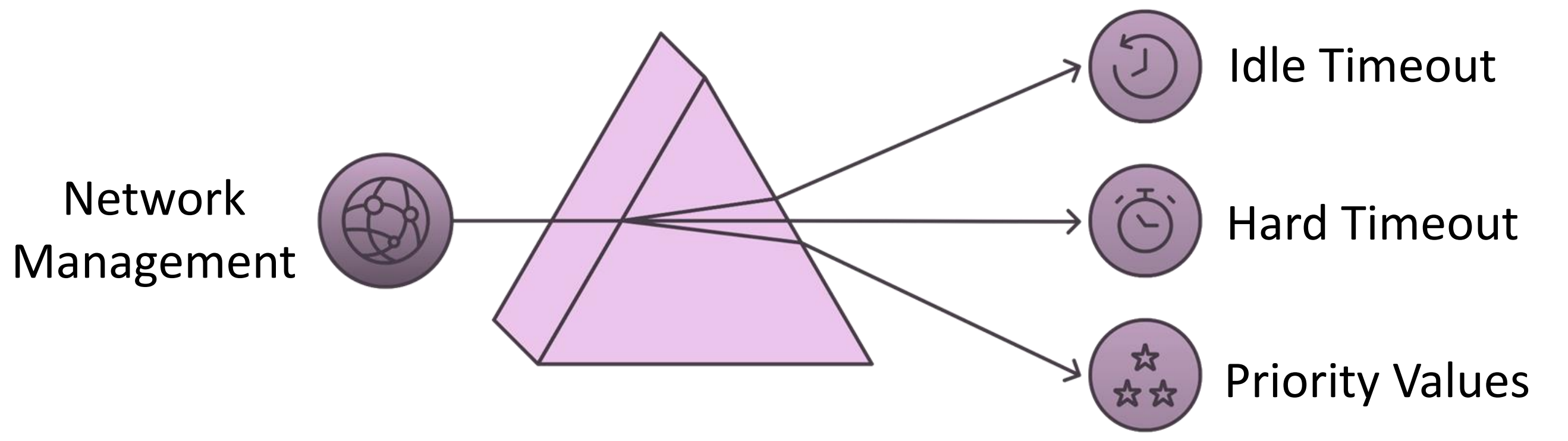
Improving Flow Rule Eviction Policy in Software Defined Networks

Saanusan K., Shriparen S.
Department of Computer Science, University of Jaffna
{2019csc041, shriparens}@univ.jfn.ac.lk

Abstract

Software Defined Networks (SDN) has emerged as an efficient alternative to traditional networking by separating the control plane from the data plane, allowing centralized control of the network [1]. This architecture enables more flexible and dynamic management of resources. Inefficiencies in flow rule management, such as improper idle or hard timeout configurations, are often cited as contributors to performance degradation, including delays, packet loss, and reduced throughput [2]. However, our study focuses not only on these timeout parameters but also on integrating flow rule priority values. By dynamically adjusting both timeout and priority values, particularly in response to traffic bursts or DDoS attacks, SDN can maintain network efficiency and deliver stable, reliable performance. Proper tuning of these factors enhances the ability to mitigate flow table inefficiencies, ensuring smoother network operations in high-stress scenarios.

Idle, Hard Timeouts & Priority Value of Flow Rules



- **Idle Timeout** clears inactive flows to prevent the flow table from becoming overloaded, improving resource usage and network efficiency [2].
- **Hard Timeout** ensures old flows expire on time, allowing space for new flows and maintaining overall network performance.
- **Priority Values** essential during congestion, enabling critical traffic to be prioritized, thus reducing delays and boosting performance.

Network Architecture

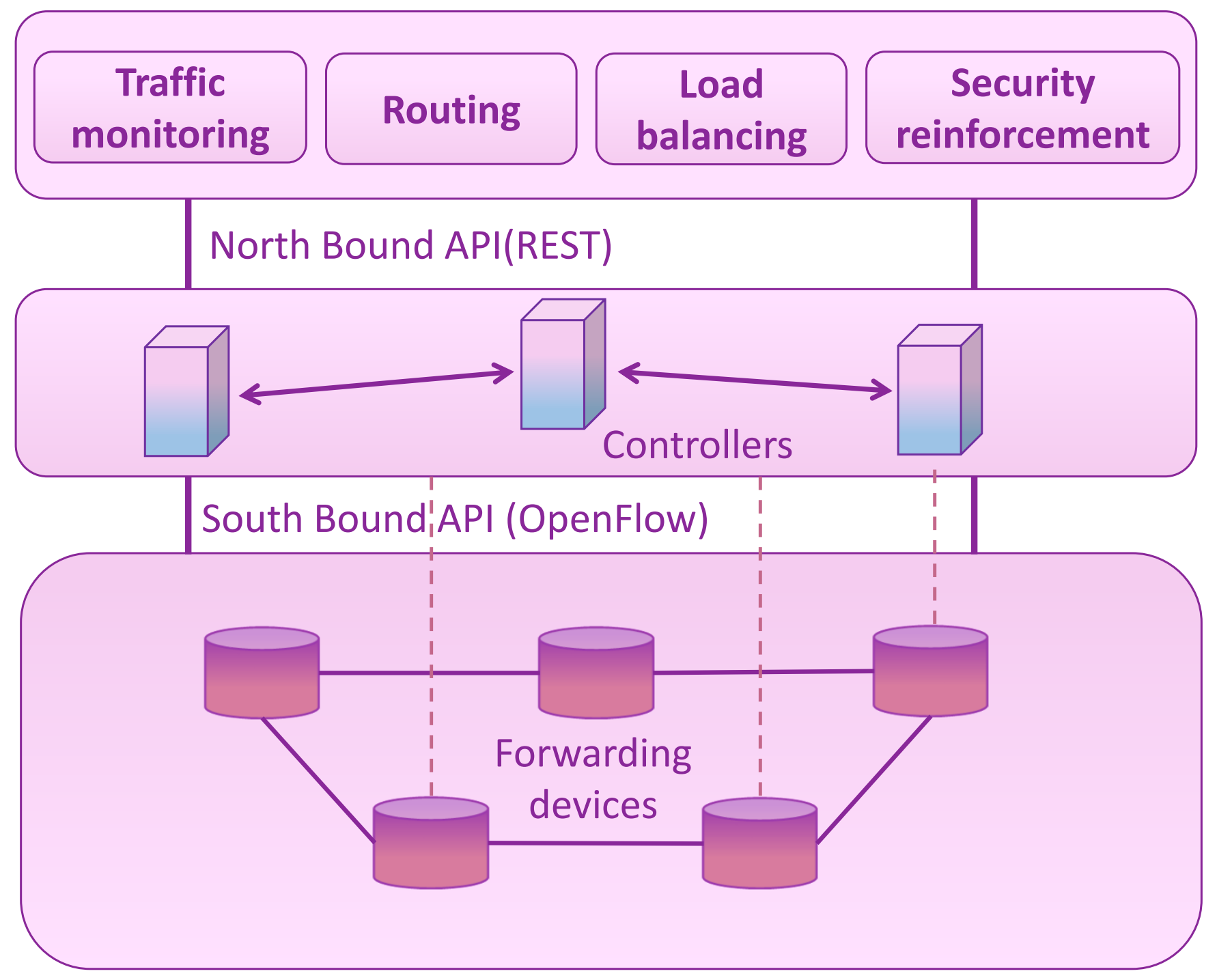
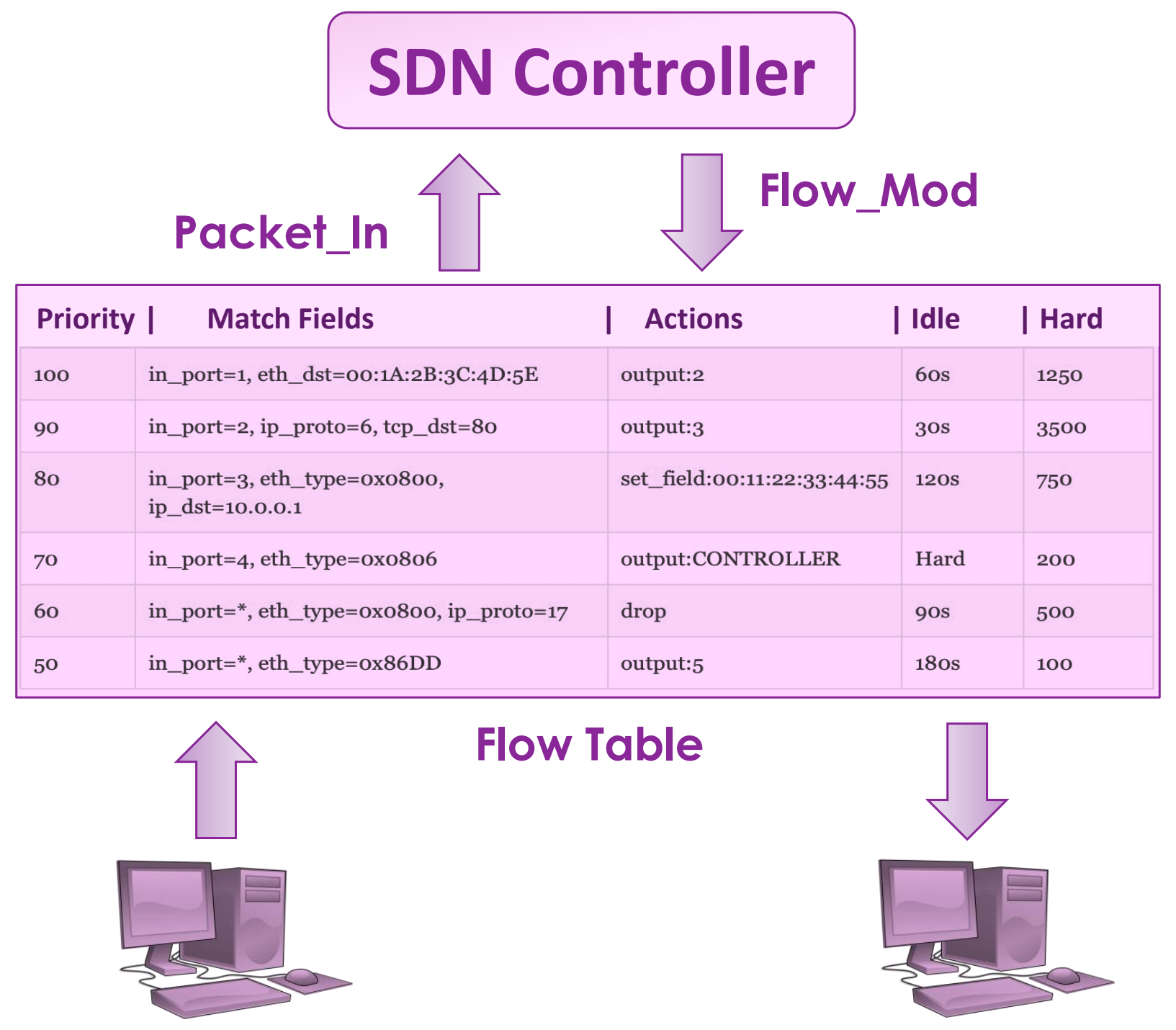


Figure 1: Basic SDN Design.

Structure: Three layer of SDN will be monitored by Top layer. Controllers in the middle layer responsible for decision making. Devices responsible for packet forwarding in the Bottom layer [3].



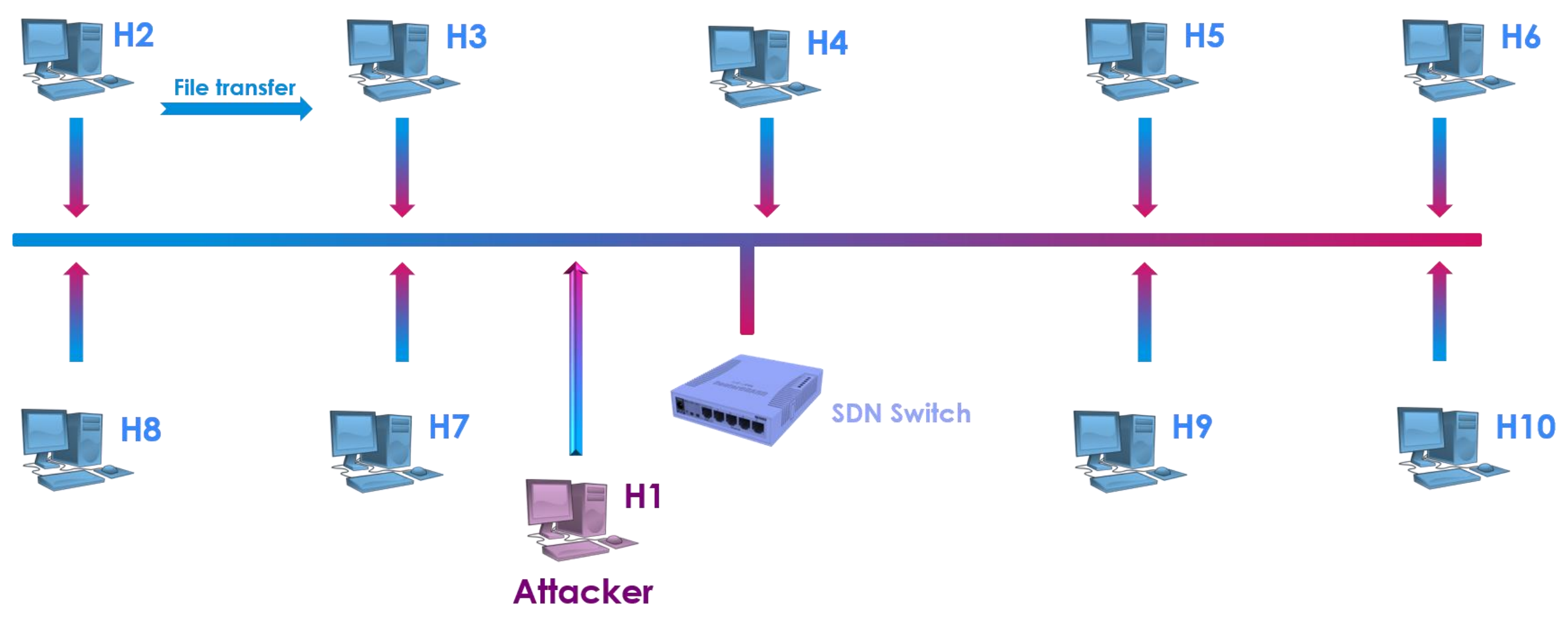
Priority	Match Fields	Actions	Idle	Hard
100	in_port=1, eth_dst=00:1A:2B:3C:4D:5E	output:2	60s	1250
90	in_port=2, ip_proto=6, tcp_dst=80	output:3	30s	3500
80	in_port=3, eth_type=0x0800, ip_dst=10.0.0.1	set_field:00:11:22:33:44:55	120s	750
70	in_port=4, eth_type=0x0806	output:CONTROLLER	Hard	200
60	in_port=*, eth_type=0x0800, ip_proto=17	drop	90s	500
50	in_port=*, eth_type=0x86DD	output:5	180s	100

Figure 2: Simple SDN Network Packet Flow.

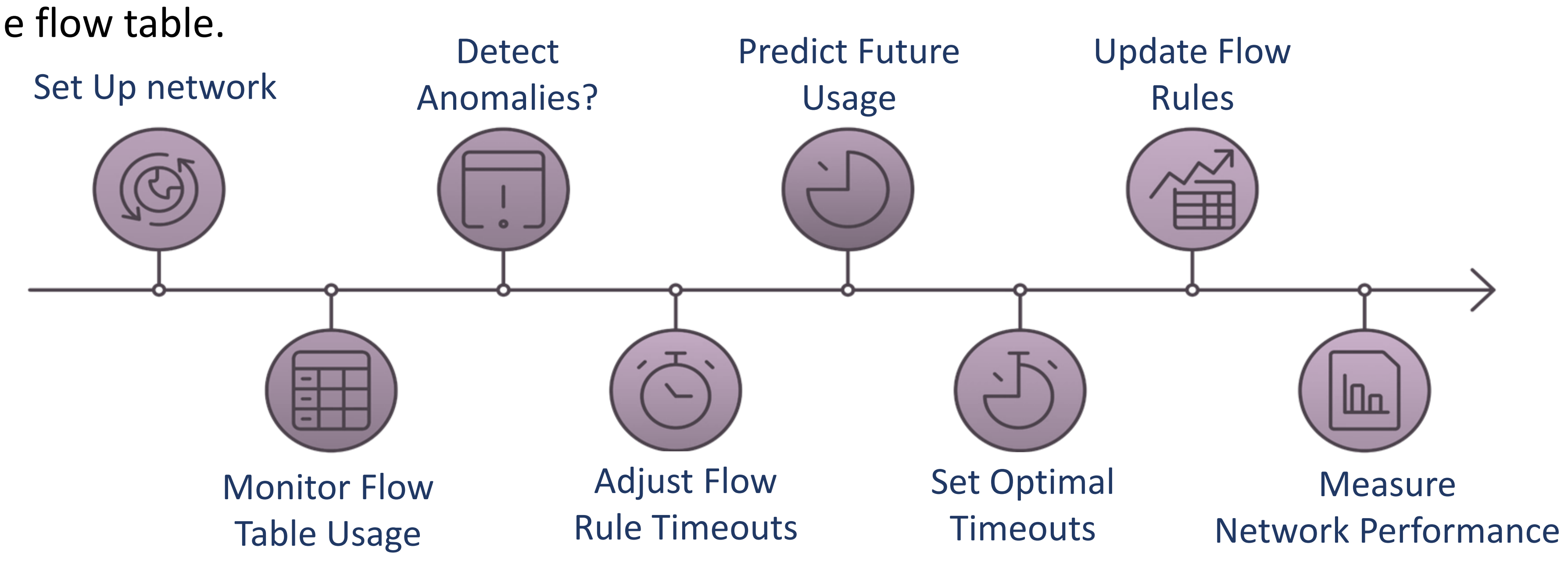
Flow Table: With the arrival of new packet, new flow rule will be generated by the controller and being installed in the Flow Table with timeout values and the Priority level [3].

Experimental Setup

1. In the topology implemented on Mininet, we add 10 hosts and an attacking host, simulating both the attack and normal traffic.



2. During the simulation, we monitor the network, and when an anomaly in the flow table occupancy pattern occurs, we begin adjusting the flow table values based on the severity of the anomaly and update the flow table.



Experiments and Results

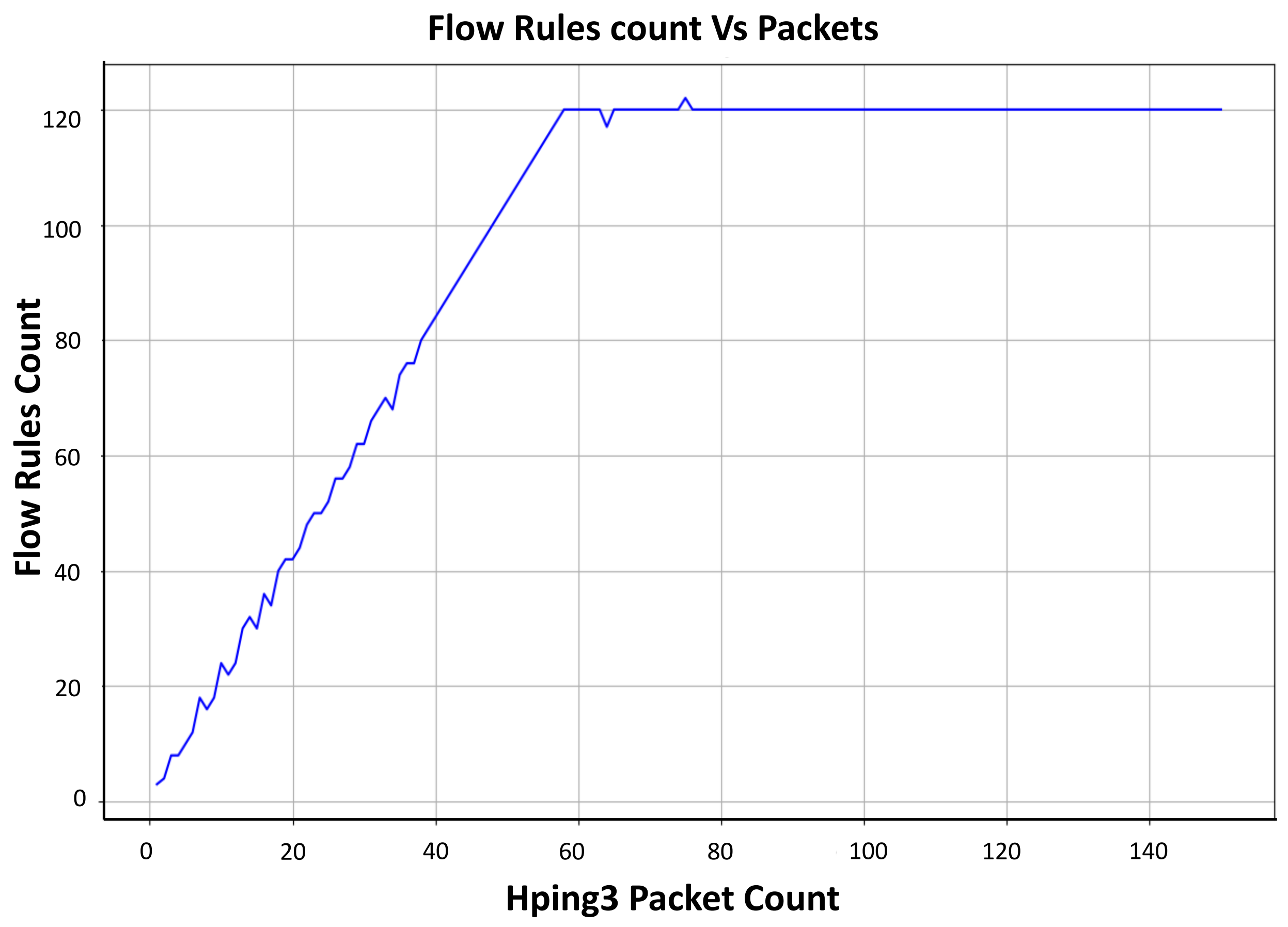


Figure 3: Generated flow rules count with transferred packet count.

Flow rule count: For now, Mininet simulation switch can have 120 flow rules in the table actively. Got these readings with Hping3 tool, by overwhelming the Mininet topology with the attack generated by the attacking tool.

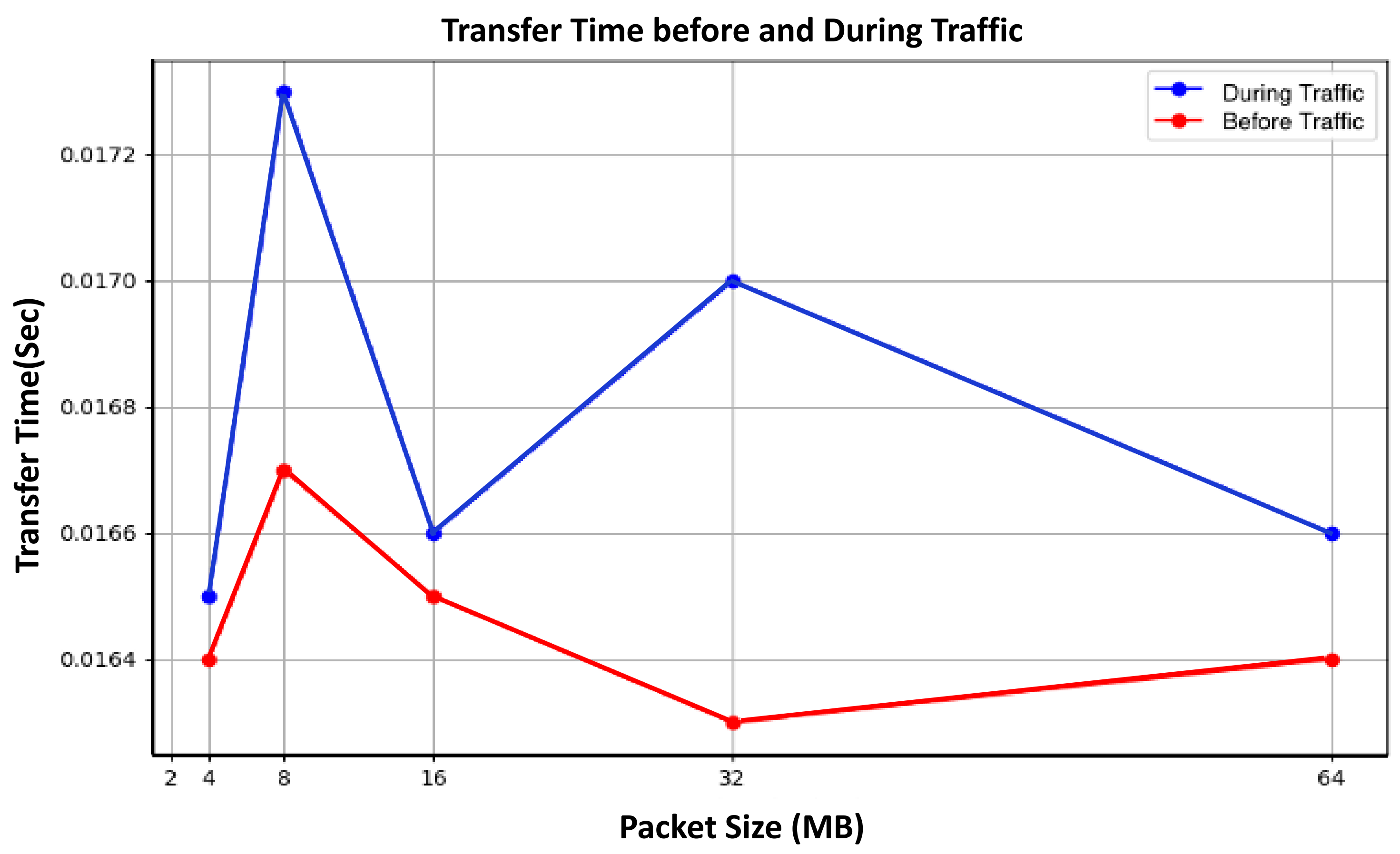


Figure 4: Transfer time increasing when attack happens on network.

Transfer time delay: When the attack happens over the network, significant delays in the transfer time for same size of packets will be there, because of the ineffective flow table utilization. In our work we are looking for the solution to effectively use the flow table space.

Conclusion

This work examines the impact of adjusting idle timeout, hard timeout, and priority values on flow table efficiency in SDN. For each configuration, we tested multiple scenarios. Idle timeout effectively removed inactive flows, freeing up flow table space, while hard timeout ensured flows expired in a timely manner. Adjusting priority allowed critical traffic to be handled more efficiently, especially under high loads. Our findings show that optimizing these parameters improves network performance and responsiveness, reducing latency and packet loss during high-demand periods or network anomalies.

References

[1] Eliyan, L. F., *et al.* (2021). DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges. Future Generation Computer Systems, vol.122, pp.149-171.
[2] Isyaku, B., *et al.* (2020). Adaptive and hybrid idle–hard timeout allocation and flow eviction mechanism considering traffic characteristics. Electronics, vol.9(11), pp.1983.
[3] Sriskandarajah, S., *et al.* (2020). Control channel denial-of-service attack in SDN-based networks. In Moratuwa Engineering Research Conference (MERCon), pp.325-330.