Section 1: Week 2: SDN Problem Statement

Nate Bachmeier

TIM-7010: Computer Networking and Mobile Computing

July 7th, 2019

North Central University

# Table of Contents

[QoS in Software Defined Networks 2](#_Toc13428192)

[What is TCAM 2](#_Toc13428193)

[Influence of Legitimate Traffic 3](#_Toc13428194)

[Influence of Malicious Traffic 4](#_Toc13428195)

[Understanding DDoS Scenarios 4](#_Toc13428196)

[Challenges from Limited Data Sets 5](#_Toc13428197)

[Challenges in Managing Application Level Attacks 5](#_Toc13428198)

[Alternative Solutions 6](#_Toc13428199)

[Systems based on Traditional Networking 6](#_Toc13428200)

[Systems based on OpenFlow 7](#_Toc13428201)

[Conclusions 8](#_Toc13428202)

[References 9](#_Toc13428203)

# QoS in Software Defined Networks

Software Defined Networking (SDN) represents the next evolutionary step in network design by enforcing a clear separation of application, control, and data planes. The separation of duties results in (1) hardware switches reduced to simple packet forwarding devices; (2) standardized viewing and modifying the network configuration across vendors; and (3) general purpose programming languages can register for networking events across the pipeline. Having these capabilities enables networks to be highly dynamic and reactive to issues impacting the Service Level Agreements (SLA).

An open research area within software-defined networks are mechanisms for increasing the supportable size of Policy-Based Routing (PBR) in the Open Flow Tables. Ternary Content-Addressable Memory (TCAM) introduce these limits, as there is a finite amount on each physical networking device (Shood, Yu, & Xiang, 2006). If these tables are unable to continue growing at a sustainable rate, this will lead to challenges managing large scale dynamic networks. These limitations are due to an expected explosion in both legitimate (e.g., IIoT and 5G) and malicious traffic (e.g., DDoS).

## What is TCAM

A commodity workstation uses Random Access Memory (RAM) and requires the application to provide an *address* to retrieve the *content*. Network packets entering into the OpenFlow switch have the opposite requirement; the destination’s virtual IP (content) needs mapping to a virtual switch port (address).

Ternary Content-Addressable Memory (TCAM) addresses this requirement by allowing each bit in the content to represent states (a) on, (b) off, or (c) doesn’t care about ‘x.’ In a single clock cycle, these wildcards are applied and queried across the entire routing table (Ullah, Ullah, Afzaal, & Lee, 2019).

A limited amount of TCAM memory is available on each device due to (1) the chips are expensive to produce; (2) requires significant power for complex circuits; (3) the power consumption emits large amounts of heat; and (4) the complex circuitry reduces amount of memory that can be placed per square centimeter.

According to Ullah et al., a typical chip contains on the order of 1000 x 144-bit words. The number of words does not directly map to the number of supported devices. Vendors can implement filter policies using multiple ‘allow’ and ‘drop’ actions requiring additional Flow Table entries. On traditional static networks, these are sufficient resources but will limit innovation on future IIoT and 5G environments.

To partially mitigate the scenario, vendors have introduced the notion of ‘Flow Groups’ as a mechanism to group multiple flows into the same policy entry. However, many business-critical use cases, such as DDoS mitigation and ensuring QoS require more fine-grained policies.

# Influence of Legitimate Traffic

The rise of Industrial Internet of Things (IIoT) and 5G wireless are expected to cause a 1000-fold increase in the number of connected devices (Petel, Ali, & Sheth, 2018) (Frodigh, 2018). The OpenFlow switch records each of these devices in their local flow tables as a requirement their traffic. Roaming devices add further load across multiple physical switches as they need to cache the policy for the configured duration.

Many scenarios of the Industry 4.0 movement require Quality of Service guarantees for safety reasons. Frodigh used a contrived example with a balancing robot that relied on external network services for calibration information. As he talked, the network signal was increasingly delayed causing the robot to ‘wobble drunkenly.’ Eventually, the robot tipped over, representing catastrophic failure.

Mitigating these QoS scenarios requires either under-utilizing networking gear or using more granular priority policies. As the size of the network increases, it becomes prohibitively expensive to under-utilize networking equipment (Jain et al., 2013). Network policy will need to be detailed enough to handle legitimate degradation of performance as well as network attacks, as a requirement of both safety and maintain continuity of core business services.

# Influence of Malicious Traffic

DDoS attacks are continuing to grow in frequency against enterprises. Akamai Technologies is responsible for the management of Global Content Delivery Networks (CDN); they have reported an annualized 60% increase in attacks (Singh, Singh, & Kumar, 2017). A literature review suggests that many businesses expect to leverage software-defined networking solutions as their mitigation strategy. However, the proposed solutions are (1) based on small simulated data sets; (2) addressed only half the scenario; and; (3) ignored the scalability concerns of granular policy requirements.

## Understanding DDoS Scenarios

A Denial of Service (DoS) attack occurs when a malicious actor performs some action on a resource to prevent another user from accessing it. A Distributed Denial of Service (DDoS) occurs when a malicious actor uses multiple intermediaries to execute the attack.

There are two broad categories of DDoS attacks (a) Network Level and (b) Application Level. Attacking the networking level is often easy to detect because of the sheer volume and statistically anomalous packet headers. Application level attacks are harder to distinguish as they are mixed with legitimate traffic and flow into every corner of the network.

## Challenges from Limited Data Sets

Service providers are unwilling to share the network traces of DDoS attacks with researchers, as they are concerned about the privacy of their users. Without realistic large datasets, most researchers operate on limited or simulated traffic (Prasetiawan, Abdurohman, & Yulianto, 2017). They have instead focused on statistical models for detecting networking level DDoS scenarios. Many of these models are tweaks to Feinstein's 2003 solution that relies on simple Chi-Squared Tests. Li et al. demonstrated that using a Long-Term Short-Term (LTSM) neural network could boost the detection confidence to nearly 99%.

After the network level attack is detected, resiliency strategies, such as deploying virtual network functions (VNF) (e.g., firewalls), can mitigate the issue at the edge and making portion a mostly solved problem (Shood, Yu, & Xiang, 2006).

## Challenges in Managing Application Level Attacks

Application level attacks have been ignored or addressed with impractical solutions. Singh et al.’s survey identified multiple publications that relied on headers inspection of HTTP GET Requests to confirm that the user-agent was a human. Others have proposed very naïve solutions that assume the attacker has no technical expertise, such as requiring the client to support JavaScript as a mitigation strategy. Ultimately any solution the relies on data provided by the client (or through obscurity) will never be secure.

Application Level attacks occur after the edge while interacting with internal systems. Modern microservice designs can further hide these traffic patterns as services cascade calls to other services. Detecting these situations requires a holistic view of the entire network, like the vantage point available to the SDN controller. Classification algorithms can use the macro view to determine if the user’s behavior is human, bot, or malware.

When an attack is detected, the controller needs to remediate all flows that are under the control of that specific user. Presently, the OpenFlow policies are too coarse and would target a group, such as throttling all users of the web application. Effectively such a remediation policy is performing the denial of service *for* the attacker!

Due to the distributed nature of the attack, authoring policy at a more granular level can be challenging to scale. Eventually, the OpenFlow tables will become saturated as it is a finite resource, QoS will not be guaranteed, and then the robot tips over.

# Alternative Solutions

## Systems based on Traditional Networking

Businesses today address these challenges through (1) provisioning excess network capacity; (2) deploying Web Application Firewalls; and (3) assuming it will not happen to them (Singh, Singh, & Kumar, 2017).

Provisioning additional capacity can be in the form of segmented networks that are air-gapped from attackers. However, anyone that has worked in distributed systems can attest that bad patches can result in similar levels of chaos. If the network operating system is unable to detect and quarantine the misconfigured devices, they will degrade the overall health of the topology. A design goal of the network operating system should include reducing the blast radius of both erroneous and malicious scenarios.

Web Application Firewalls (WAF) can detect a subset of malicious attacks, such as SQL Injections and specific HTTP flooding attacks. These technologies do not protect against an army of bots requesting public web pages as a mechanism to exploit an asymmetry of computation requirements. That is not to suggest that deploying WAF is a terrible idea; it is better than nothing and should be a standard tool in every network administrators toolbox.

Another conventional solution is to do nothing and assume that it will not impact the business. Ignoring the risks will only procrastinate the scenario to a time not chosen by the company. Like the lousy patch scenario, the root cause might not be a malicious actor, but some event such as a successful viral marketing video.

## Systems based on OpenFlow

The OpenFlow 1.3 specification includes support for network packet queues. These are used by current ‘state-of-the-art’ solutions to enforce Quality of Service (Mirchev, 2015). An example implementation might create N priority queues, then use a fair sharing algorithm to process them accordingly. If a flow is detected to be violating a resource quota, then is demoted to a lower level priority queue.

One of the challenges with this approach is that OpenFlow supports a finite number of queues. Using inefficient sharing algorithms can result in the problem moving from the high priority queue to the low priority queue. While this is a preferred state, it is not sustainable long term. Eventually, the background priority applications will timeout and need to compensate for the operations.

Another new solution was published last month, which replaces the TCAM design with Static Random-Access Memory (SRAM) Field Programmable Gate Arrays (FPGA) devices. Their approach improved read operations by 2.5x using 2/3rds less power. The performance improvement came from reprogramming the logic gates into multiple filter masks, then taking a union of the network masks to reconstruct the same answer as TCAM.

However, there is a performance penalty each time the routing table needs to be updated. The delay occurs because reconfiguring the logic gates requires blocking I/O during the rearranged. In the worst case, this can result in all gates being touched and require over 500 clock cycles to complete. The authors propose that partitioning and sorting algorithms could be used to reduce the probability that large numbers of gates need to be modified. Additional research is required in this area as it might not be performant for specific highly dynamic environments (e.g., wireless IIoT and 5G services).

# Conclusions

Ensuring Quality of Service with software-defined networks is a more natural fit than traditional networks, as the system can be dynamic and responsive to a holistic view. However, many of the same challenges continue to exist in the new paradigm. These systems need to express granular policy so the expected increases in legitimate (e.g., IIoT) and malicious (e.g., DDoS) traffic can ‘co-exist’ without disrupting the continuity of core business services.

While research into preventing DDoS has existed for some time, it has been lacking in scope and focusing on half the problem space. Application Level attacks are growing in ‘popularity’ at an alarming rate of 60% annually.

Defenses against these attacks will require innovations from software-defined network controllers to isolate the malicious flows and apply policies to throttle or block them. That will require solving two problems (1) build more efficient classifiers to determine if a user is malicious or not, and (2) expand the size of the OpenFlow Tables for more granular policies.

# References

Feinstien, L., Schnackenberg, D., Balupari, R., & Kindred, D. (2003). Statistical Approaches to DDoS Attack Detection and Response. *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX’03)*.

Frodigh, M. (2018, May 30). *Live from ICSE: Conference Opening + Keynote from Magnus Frodigh*. Retrieved from YouTube: https://www.youtube.com/watch?v=cpeMmMh7Syk

Jain et al. (2013). B4: Experience with a Globally-Deployed Software Defined WAN. *SIGCOMM’13, August 12–16, 2013, Hong Kong, China.*

Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., & Long, L. (2018). Detection and defense of DDoS attack–based on deep learning in OpenFlow‐based SDN. *Int J Commun Syst. 2018;31:e3497*, 1-15.

Machado, C., Granville, L., Schaeffer-Filho, & A. (2016). ANSwer: Combining NFV and SDN Features for Network Resilience Strategies. *2016 IEEE Symposium on Computers and Communication (ISCC)*.

Mirchev, A. (2015). Survey of Concepts for QoS improvements via SDN. *Network Architectures and Services, September*, 33-40.

Petel, P., Ali, M., & Sheth, A. (2018). From Raw Data to Smart Manufacturing: AI and Semantic Web of Things for Industry 4.0. *IEEE INTELLIGENT SYSTEMS July/August*, 79-85.

Prasetiawan, D., Abdurohman, M., & Yulianto, F. (2017). IMPROVING DISTRIBUTED DENIAL OF SERVICE (DDOS) DETECTION USING ENTROPY METHOD IN SOFTWARE DEFINED NETWORK (SDN). *ComTech, Vol. 8 No. 4 December 2017*, 215-221.

Shood, K., Yu, S., & Xiang, Y. (2006). Software-Defined Wireless Networking Opportunities and Challenges for Internet-of-Things: A Review. *IEEE INTERNET OF THINGS JOURNAL, VOL. 3, NO. 4, August*, 453-463.

Singh, K., Singh, P., & Kumar, K. (2017). Application layer HTTP-GET Flood DDoS attacks: Research landscape and challenges. *Computers & Security 65*, 344-372.

Ullah, I., Ullah, Z., Afzaal, U., & Lee, J. (2019). DURE: An Energy- and Resource-Efficient TCAM Architecture for FPGAs With Dynamic Updates. *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 27, NO. 6 June 2019*, 1298-1307.