The Effects of IDS/IPS Placement on Big Data Systems in Geo-Distributed Wide Area Networks

Michael Hart[1], Eric Richardson[2], Rushit Dave[3]
College of Science, Engineering, & Technology, Minnesota State University, Mankato, United States[1, 3]
College of Health and Human Services, University of North Carolina Wilmington, United States[2]

Abstract*—Geographically-distributed wide-area networks (WANs) offer expansive distributed and parallel computing capabilities. This includes the ability to advance Wide-Area Big Data (WABD). As data streaming traverses foreign networks, intrusion detection systems (IDSs) and intrusion prevention systems (IDSs) play an important role in securing information. The authors anticipate that securing WAN network topology with IDSs/IPSs can significantly impact wide-area data streaming performance. In this paper, the researchers develop and implement a geographically distributed big data streaming application using the Python programming language to benchmark IDS/IPS placement in hub-and-spoke, custom-mesh, and full-mesh network topologies. The results of the experiments illustrate that custom-mesh WANs allow IDS/IPS placements that maximize data stream packet transfers while reducing overall WAN latency. Hub-and-spoke network topology produces the lowest combined WAN latency over competing network designs but at the cost of single points of failure within the network. IDS/IPS placement in full-mesh designs is less efficient than custom-mesh yet offers the greatest opportunity for highly available data streams. Testing is limited by specific big data systems, WAN topologies, and IDS/IPS technology.

***Keywords*—Information security; network topology; wide-area big data; wide-area networks; wide-area streaming**

I. INTRODUCTION

Increasingly, organizations must collect large amounts of data that is located in physically distanced data centers. Geographically-distributed big data server clusters provide massive scale data analytic capabilities across wide-area networks (WANs). Several big data frameworks in use at the time of this writing such as Apache Spark are deployed within single data centers [1]. However, big data clusters that run in local area networks (LANs) do not necessarily have the same challenges as WANs. For instance, LANs have certain advantages like bandwidth, shorter distance routing, and highly available communication at cheaper costs. LANs also have limitations spanning from local resources to global connectivity [2].

WANs enlarge the capabilities of LANs, offering expansive resources and connectivity for geo-distributed data streaming. For instance, WANalytics research is investigating how to optimize distributed structured query language (SQL) queries across WANs [3]. Subsequently, unsupervised machine learning provides several possibilities to enhance geo-distributed data streaming. For example, a sliding version of the hidden Markov model (SlidHMM) improves bottleneck detection in WAN data analytics [4]. Despite the latter progress, a survey on geo-distributed frameworks found that research is lacking in several areas. This includes decentralized architecture, data streaming, multi-clusters, information security, and privacy [1]. The objective of this work is to investigate the role of information security in geo-distributed big data analytic framework literature and provide subsequent steps toward securing this infrastructure in future research.

Organization of the paper is as follows. The authors perform a review of literature on the influence of information security on geographically-distributed big data systems in Section II. A methodology develops from the review that identifies procedures to test the performance of secured WAN topologies in Section III. The results of the testing and a discussion are given in Section IV and Section V respectively. Finally, Section VI concludes the study.

II. RELATED WORK

To better understand big data frameworks and their geographically-distributed contributions, Bergui [1] performed a survey of existing literature. A theme in progression centers around optimizing big data systems for the ever-increasing changes in network topology. Tuning these systems for WANs is complex, yet not always clear in existing literature. For example, in [1], bandwidth-aware systems do not always use resource managers like yet another resource negotiator (YARN) with specific WAN tuning capabilities. The researchers also emphasize further work is necessary to study information security and system architectures in geo-distributed big data systems. Trust models become more complex when distributing data between different governments. Researchers encourage designing authentication strategies and decentralized architecture frameworks capable of supporting more complex geo-distributed clusters [1].

To better understand research that helps optimize big data systems, the writers review data querying, transfer, placement, and their environments, which includes network topology.

A. Querying Data

Research on optimal geo-distributed computing architectures is ongoing. In study [3], the researchers introduce the term WANalytics, which they contrast with Wide-Area Big Data (WABD). WABD typically copies data from multiple data centers to a single data center where data analytics transpire. WANalytics is designed to support massive scale geo-distributed analytics across multiple data centers. Its goal focuses on reducing expensive WAN bandwidth while maintaining compatibility with data sovereignty restrictions [3].

Initial experimentations demonstrate that WANalytics can reduce data transfer costs by as much as 360 times compared to centralized data center methods. This occurs by allowing users to test SQL queries between data centers in Europe, North America, and South-East Asia. While WANalytics shows tremendous progress toward optimizing geo-distributed computing architectures, information security appears to be distant in this literature [3].

Demand for wide-area data analytics enforces the need to advance the capabilities of geo-distributed big data systems. For instance, Wang and Li [4] propose the Lube system framework to monitor, detect, and resolve bottlenecks in in geo-distributed data analytic queries. Benchmarks show optimizing scheduling policies across distributed data centers can lower query response times up to 33 percent when compared to other big data systems like Apache Spark. Similar to Lube [4], Turbo [5] has the ability to improve geo-distributed data analytics queries at runtime. Using machine learning, Turbo optimizes data analytic query execution plans across multiple physically distanced data centers. In a geo-distributed Google Cloud environment that spanned eight regions, Turbo lowered query completion times by 41% [5].

In study [6], the authors focus on common executions in wide-area network streaming analytics queries. Examples of common execution elements include shared data processing and input data. While improvements are achievable using common query executions in streaming analytics, researchers emphasize that without WAN awareness, weaker performance can exist in geo-distributed data center communications. WAN-aware multi-query optimizations that leverage common executions can reduce WAN bandwidth as much as 33% in contrast to systems that fail to use shared execution components. Therefore, multi-query efficiency may have some dependency on WAN-awareness [6]. Despite the advancements of wide-area data analytics in geo-distributed analytics, many questions remain. Researchers in [1] note that further work is beneficial to address variations in the structure of data, determine the optimal features to reduce query completion times, and construct a larger range of performance metrics to measure bottlenecks. Another complementary vein of research focuses on bulk data transfer.

B. Bulk Data Transfer

Transferring bulk data within inter-datacenter networks requires efficient strategies to reduce associated costs. Multimedia big data such as video streams and gaming content, compete for leftover bandwidth in backbone transport networks that connect geographically-distributed data centers. However, the exponential increase of data transfer these services need can degrade backbone networks [7]. Though certain algorithms can efficiently manage guaranteed traffic and reassignment [8], it is well understood that middleware and control plane protocols are amongst several layers of the architecture that require greater attention in research [9]. As one example of progress toward the latter goal, software defined networking (SDN) helps dissociate the control plane from data paths. This leads to more dynamic adjustment of data routing as network environmental attributes change [10].

Particularly when sending bulk data transfers between geo-distributed data centers, researchers in study [10] highlight three primary services. This includes 1) task admission control, 2) data routing, and 3) store-and-forward. Task admission control rejects or accepts network transfer requests based upon whether they can be completed by a specified deadline. Data routing must choose the best path data should take to reach its destination, which can include rerouting through intermediate data centers. The concept of store-and-forward decides whether it is more efficient to store data temporarily within intermediate data centers and forward it at a more optimal time than the immediate time of execution. If so, decisions must be made to determine where the data is temporarily stored until it reaches its destination [10].

C. Data Placement

Subsequent focus on efficiently distributing data between data centers are algorithms that calculate cloud service provider (CSP) costs [11]. Certain data sent between CSPs can tolerate delays, which can be transferred using store-and-forward intermedia storage nodes with off-peak internet service provider (ISP) bandwidth that is already financed [12]. Multi-rate bandwidth on-demand (BoD) brokers employ scheduling algorithms to optimize the use of this residual bandwidth. As an example, the BoD broker in study [6] uses standby wavelengths within the wavelength division multiplexing (DWDM) layer to decrease peak network bandwidth. Adjustments are possible based on delay-intolerant and delay-tolerant transfer requests. Compared to relational algorithms like First-Come-First-Served (FCFS), more precise use of time slots in all wavelengths is optimal when peak bandwidth results in delayed or blocked requests [7].

When inter-datacenter networks are congested, certain storage decisions can help reduce additional network load. This includes the use of intermedia storage (IS) and edge storage (ES). ES allows certain types of jobs like bulk data transfers to leverage storage at the edge of network domains and forward it during periods of off-peak CSP bandwidth. In study [13], as network load increases there is a linear decrease in the success of bulk transfers. Bulk data transfers are optimal when the allowed wait time is twice the aggregated network load. In summary, the authors found that ES and IS perform similar when peak bandwidth times are small. Medium or less network load results in little difference between ES and IS. However, in this research IS performed significantly better than ES in times of high network load [13].

Research on bulk data transfer across low latency or congested links is helping advance several needs including scheduling optimization [9], bandwidth costs [11], and delay tolerance [12]. In reviewing related literature, information security is not a central component of big data transfers between geo-distributed cloud data centers [7, 10], inter-datacenter bulk transfers [11, 13], or research networks [8]. Additionally, while certain testing considers differences in specific network topology [7, 10] others do not [8, 12]. Therefore, opportunities may exist to study the influence of information security and network topology on bulk data transfers in low latency network environments. To explore this further, the authors turn to the role of network topology and geo-distributed big data systems.

D. Network Topology

Network topology influences several dimensions of geo-distributed big data systems, including the elasticity of nodes in a cluster [12]. A challenge of big data streaming is resource provisioning across shared cloud infrastructure. Particularly when the cloud tenant does not own infrastructure, it can be challenging to decipher the cause of poor performance on collective physical hardware that runs virtual machines (VMs). In study [14], the authors highlight the need for the dynamic rescheduling of big data streaming tasks using multitenant-aware resource provisioning that is independent of the VM hypervisor. Software defined networking (SDN) plays an impactful role in this provisioning by supporting load balancing between cloud-based VM clusters. In contrast to other network topologies, SDN can define its topology in real time. This in turn allows for additional cloud node elasticity [14]. In study [12], researchers focus on optimizing bulk data transfer in a geo-distributed data center system using SDN architecture. SDN elasticity promotes dynamic routing decisions using bulk data transfers in pieces in contrast to handling transfers as endless flows [12].

Like [13], researchers in study [15] highlight a need to optimize big data streaming strategies between geo-distributed data centers. The authors note that traditional methods for distributed data streaming such as task assignment are insufficient when high throughput data exists along with low latency WAN links [15]. Researchers also emphasize the need to perform data mining on data sent between WANs from streaming applications that perform user-clicks, social networks, and Internet of Things (IoT) hardware [16]. A proposed advancement is an SDN-based resource provisioning framework capable of monitoring WANs, identifying an optimal selection of big data worker nodes, and more efficiently assigning tasks to the chosen nodes. In initial tests, SDN resource provisioning results in minimal processing time that is 1.64 times faster on the tested environment, which included Apache Flink, Apache Spark, and Apache Storm [15].

One of the challenges of geo-distributed and wide-area network data analytics streaming is identifying performance problems when infrastructure is not under the control of the customer. Multitenant-aware resource provisioning using SDN network topology is a proposed solution when cloud computing hardware is shared amongst multiple customers [14]. Monitoring and increasing performance of multitenant streaming analytics also requires more advanced worker node and IoT placement strategies in low latency network topology. Streaming platforms like S4, Apache Storm, and Apache Spark were not initially designed for low latency analytics shared between users and applications in distributed IoT systems. However, improvements are being made in the streaming platforms. For instance, Apache Spark supports structured streaming via PySpark, a Python API. Spark streaming has the capability to stream data in micro-batches [1]. In study [16], the GeeLytics platform is introduced as an alternative streaming platform to address low latency networks. This includes more dynamic mechanisms to balance real-time streaming in the cloud and network edges. The proposed design is expected to reduce edge-to-cloud bandwidth use for IoT data analytics. It is also engineered to increase customer insight into multi-tenancy system efficiency [16].

Proposed in study [17], a worker node placement framework focuses on wide-area streaming analytics. It builds upon the Simple Additive Weighting (SAW) method. In this model, a central global manager determines how tasks are assigned across multiple edge data centers using a proposed SAW-based Node Ranking (SNR) algorithm. Task slots are determined based upon the amount of input data and processing power of each slot. Additionally, task slots communicate over the WANs links. This allows the global manager to maintain the status of key link metrics including cost, delay, and bandwidth as well as identify network topology changes. Researchers tested the SNR algorithm on Apache Flink, Apache Spark, and Apache Storm using small, medium, and large graphs to simulate different network sizes. Each big data system shows performance improvements compared to other worker node placement strategies [17].

WAN traffic costs are central to several recent advancements in geo-distributed streaming analytics research. Costs are influenced by network design. For instance, the hub-and-spoke design includes several network edges that interconnect via WANs to a central data warehouse. Popular streaming analytics service providers use this model at the time of this writing [18]. Important to this network model is determining the optimal amount of computation that should exist at the center of the topology or the edge. Based on the hub-and-spoke network topology, researchers have identified staleness or the delay in retrieving data results and WAN traffic as pivotal metrics. Experiments using common analytics from large CDNs highlight the need to minimize both latter metrics [18].

AggNet is a subsequent advancement in research focused on reducing WAN traffic costs. Developed on the Apache Flink framework, AggNet [19] reduces WAN bandwidth by aggregating a percentage of real-time data analytics closer to the location of end users. Aggregation from AggNet implementation has shown 47% to 83% decreases in traffic costs when compared to traditional costs from relevant industry organizations that included Akamai and Twitter [19].

Although the hub-and-spoke network model is a cornerstone in recent geo-distributed streaming analytics work [18-19], researchers understand current network topology must change to meet the future needs of big data analytics. In study [20], researchers argue that high communication cost, data sovereignty, and data privacy challenge the feasibility of central data center designs. Proposing the concept of geo-distributed machine learning (Geo-DML), parameter server (PS) placement remains a challenge for distributing raw machine learning data between WANs. A proposed solution is using approximation algorithms capable of selecting the optimal data center for training using network cost. Results of using this strategy reduce communication cost up to 21.78% over other Internet network topology. However, the potential effect of IDS/IPS hardware is unknown [20].

E. Summary

Several advancements are occurring that improve geo-distributed big data systems. AggNet helps reduce WAN traffic by placing data closer to end users [19]. In study [18], researchers develop a hybrid online algorithm to determine optimal computation at the network edges versus the center in a hub-and-spoke WAN model. In small to large network topology, the SNR algorithm shows capability to optimize tasks across geo-distributed data centers using the simple-additive weighting method [17]. Subsequently, an approximation algorithm finds the best data center as the parameter server for machine learning training on two network topologies, which included a Google private WAN and a United States Internet with nine interconnected data centers [20]. Like research on bulk data transfer between geo-distributed data centers, little emphasis exists on information security in these papers [17-20]. Additionally, network topologies are limited to only a few different types of WANs [20] as well as traditional hub-and-spoke designs [18]. SDN-based networks also show promise in helping optimize resource provisioning but may need additional consideration as they gain more traction in geo-distributed WAN analytics [14].

The research that follows presents an elementary investigation into whether IDS/IPS placement impact the performance of big data systems operating between low-latency network topologies.

III. Methodology

The research design follows the information systems research framework outlined in study [21]. Three pillars of the framework include the environment, information systems research, and the coinciding knowledge base. Within the environment stage of the latter research methodology, this paper focuses on building modern IT infrastructure to support massive-scale data analytics. Subsequently, the research stage focuses on WAN simulations to evaluate supporting network topology for capable big data systems. The researchers add to the existing knowledge base by reporting on the effects of IDS/IPS placement on real-time data streaming systems in network topologies able to migrate into modern SDN-enabled WANs.

Following the design science methodology, business needs are the driver for building new information system artefacts [21]. Wide-area data analytics is gaining traction due to the increased need for businesses to analyze real-time data streams in multiple physical locations [6]. Notably, big data systems in geo-distributed data centers provide immense opportunity to support streaming massive amounts of data on low-latency WAN connections. Provisioning resources across modern SDN WAN architectures, provides big data systems like Apache Spark with more expansive horizontal scalable than centralized data centers [15]. To support the growing business need for geo-distributed streaming, the researchers design and implement current WAN topologies capable of efficiently and securely connecting physically distanced big data systems.

Investigators design and implement three well recognized Software-defined-wide area network (SD-WAN) arbitrary topologies outlined by study [22], including hub-and-spoke, full-mesh, and custom-mesh. Each are implemented across ISP leased lines. The applied network topology uses the specifications engineered by Cisco Systems in their Cisco Extended Enterprise SD-WAN Design Guide. These are located in Fig. 7 Hub-and-Spoke Topology with Cisco IR1101 and Fig. 8 Mesh Topology with Cisco IR1101 and SD-WAN in study [23].

A. Experimental Network

The experimental Cisco Systems network resides in an enterprise-class data center. Within the research network, the authors design and implement wide area network (WAN) data centers in four major United States cities. The central data center is located in New York, New York. From the New York data center, WAN links connect to data centers via routers in the cities of Orlando, Florida, Los Angeles, California, and Seattle, Washington. Router placement and configuration for each WAN parallel Cisco IR1101 and SD-WAN in [23]. WAN network latency between the latter data centers equals averages, at the time of this writing, in milliseconds (MS) published by AT&T in study [24]. The full-mesh network topology, which includes network latency for all WAN links, is outlined in Fig. 1.

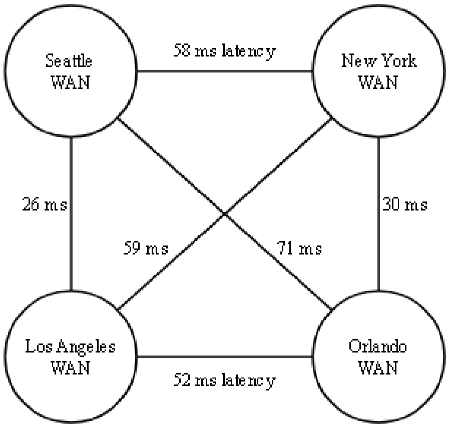

Fig. 1. WAN network latency.

B. Big Data System Architecture

The experimental environment includes four big data server clusters in each data center. Clusters are connected by the WANs and secured by intrusion detection systems (IDSs) and intrusion prevention systems (IPSs). All server and router hardware are the same make and model. Hardware has precisely the same specifications including physical CPUs, memory, and solid state disks. Each data center houses a Dell PowerEdge server running an updated Microsoft Hyper-V Server 2019. Virtual machines hosted in Hyper-V consist of Intel Xeon processors with five physical CPU cores and 24 gigabytes of memory.

Fig. 2 shows the big data system architecture for cluster one (C1) connected to the New York WAN. Each of the four system clusters parallel this architecture. The clusters consist of six big data system VMs running the Ubuntu 22.04 Long Term Support (LTS) server operating system. Two VMs are dedicated Apache Hadoop name nodes. The primary and secondary name nodes connect to four data nodes with a replication factor of three. Data nodes are configured as both Apache Hadoop and Apache Spark worker nodes. Name nodes connect to the WAN through a router and an IDS/IPS. The WAN routers at each site also have one external facing Dell PowerEdge server with 5 physical CPU cores and 24 gigabytes of memory. The latter WAN-connected Ubuntu 22.04 LTS servers measure and collect performance data between the geo-distributed data centers. The edge servers are also the source of all external data streams sent to the big data clusters. Table I shows the corresponding software and versions of the big data systems.

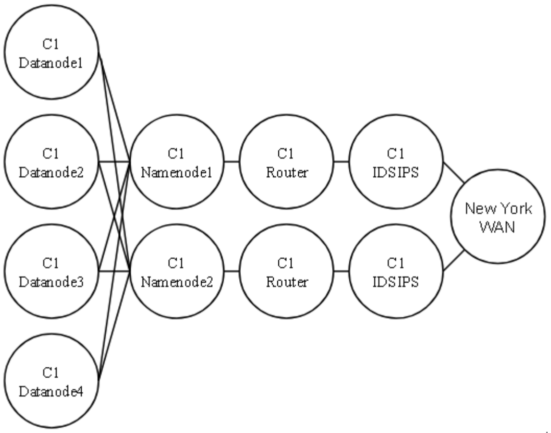

Fig. 2. Cluster architecture.

Experiments use Suricata for the intrusion detection system (IDS) and intrusion prevention system (IPS). Suricata is well supported by the open-source community as a modern world-class IDS/IPS [25]. It allows researchers to customize packet bundling techniques to analyze stream data sets efficiently and effectively [26]. Suricata is compiled with the emerging threats open ruleset [25]. Specific Suricata rules allow the unique public IP addresses of the streaming clients to connect to a primary and secondary name node in each data center cluster. Streams and associated IPS rules use the customized TCP port range of 9990 – 9999 on each big data cluster. With the exception of the experimental data streams and SSH for system administrator IP addresses, no other external traffic is allowed into the data center networks by the IDSs/IPSs.

TABLE I. EXPERIMENTAL SOFTWARE VERSIONS

Software	Version
Hadoop	3.3.6
Iptables	1.8.7
Nmon	16
OpenJDK	8u412
Pdsh	2.31-3
Pyspark	3.5.1
Python	3.10.12
Spark	3.5.1
Suricata	6.0.4
Tcpdump	4.99.1
Ubuntu	22.04.4

C. Streaming Architecture

Within the big data system architecture, the primary and secondary name nodes are configured as Apache Spark streaming servers. Fig. 3 outlines the big data streaming architecture. From an Ubuntu server on each WAN, 1 GB streams are sent to the primary and secondary name nodes. To process the data streams the authors developed a big data streaming application using Apache PySpark. The application facilitates the unstructured data streams to Apache Spark on the primary and secondary name nodes. It uses the Spark context object and PySpark streaming class instudy [27] to develop the streaming functions. Each application instance processes word counts on the data streams. Word counts are aggregated using key value pairs using Spark in-memory computation and subsequently written across the Hadoop Distributed File System (HDFS) for long-term data analytics. HDFS block sizes are configured for 128 MBs.

D. Benchmarking Technologies

Simulation is one of several methods in the design science research framework [21] that helps assess and refine novel artifacts. Central to this work is determining how modern IDS/IPS placements impact the performance of geo-distributed big data system clusters. The researchers use raw network performance statistics between connecting WANs to evaluate real-time data streams. In study [28] researchers evaluate raw network performance using httping and iperf3 on anonymous circuit-based communications. The networking utilities were able to effectively measure the average latency and throughput between hubs in a metropolitan area. Iperf3 is also used in WAN environments to test network capacity. Researchers investigated the transfer of science big data across WANs in study [29] using NVMe over Fabrics (NVMe-oF). NVMe-oF is able to provide enhanced non-volatile memory functionality for storage networking fabrics. Methods in the study successfully use iperf3 to test for bottlenecks in the networks [29].

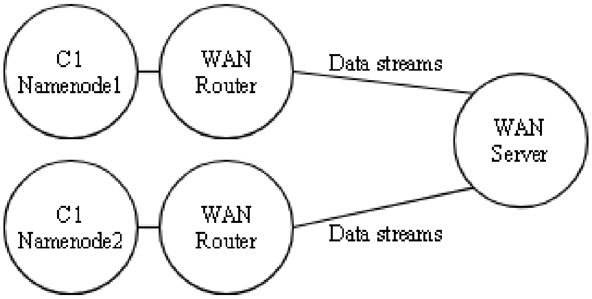

Fig. 3. Streaming architecture.

Like [29] iperf3 measures latency between the geo-distributed data centers in this study. In Fig. 2, iperf3 resides on the name node servers, IDS/IPS servers, and the WAN servers. Network latency is measured between the edge of each WAN and the name node clusters. Similar to study [26], the authors combine TCP packets into streams to analyze the network data. Libpcap, tcpdump [30], and Nigel's performance Monitor for Linux (nmon) [31] facilitate the raw network packet captures. Nmon uses the “-s” option to collect network packets every second throughout the duration of the Apache Spark streaming tests.

In addition to using network bitrate to test data streaming performance, it also determines the optimal location of the IDSs/IPSs in this study. To establish the optimal IDS/IPS placement in each network topology, researchers iteratively run the experiments with each recommendation in the Cisco Extended Enterprise SD-WAN Design Guide [23]. The authors base the final IDS/IPS location selection on the best raw network bitrate for each topology in the testing that follows.

The proposed research methodology uses a design science approach to investigate the impact of IDS/IPS placement on geo-distributed big data systems. It outlines the system architectures and benchmarking processes in the coinciding experiments. Next, the authors implement the proposed tests and report the results of the evaluations.

IV. Results

Hub-and-spoke in Fig. 4 is the first experimental network topology (T1) that tests the IDS/IPS performance of geo-distributed big data systems. WAN connections source from a central data center in New York, NY to the remote cities of Seattle, Los Angeles, and Orlando. The authors automated the tests using the Python programming language and Bourne-Again SHell (bash) scripting. This includes a start and stop script.

A. Experimental Environment

A start script prepares a consistent experimental environment for each iteration of the performance testing. A stop script resets the environment to the original state, ensuring each test begins with the same configuration. The start script begins by starting each Suricata IDS/IPS service and checking the compiled security rules. After the IDS/IPS is functioning properly, the script starts Apache Hadoop and Spark. At this stage, a health check ensures HDFS is operating correctly across the clusters. If the distributed file system is unhealthy, it exits after logging error codes. If HDFS is healthy, TCP ports 9990-9999 open for Apache Spark streaming.

Each name node on four geo-distributed big data clusters runs a parallel Python application that facilitates the system and network performance benchmarking. The Python application invokes the PySpark streaming application, establishing 1 GB data streams to the primary and secondary names nodes. Throughout the experiments, a health check monitors the Apache Hadoop and Spark logs. If the Python application fails in-memory processing or HDFS writes at any time during the real-time stream, the application exits after logging error codes. The start script sleeps for 30 seconds following invocation of the Python application to ensure streaming is functional.

Following successful execution of streaming services, a series of bash shell commands collect and aggregate raw network performance statistics using libpcap, tcpdump, nmon, and iperf3. Data aggregation is per cluster. For example, data combines from the two name nodes and two IDSs/IPSs for each site into a single file. Measurement and results are from transmission control protocol (TCP) network traffic. Tcpdump and nmon results are collected from real-time TCP traffic. Intervals for each tool are set to write performance data every second. Nmon executes with the default settings with the exception of the “-s” syntax for seconds. Iperf3 uses the IP address of each server, the connecting port, and the interval in seconds, and the bidirectional traffic syntax.

Tests invoke in parallel across each cluster using the start script. To ensure saturation, the authors ran the tests ten times for twelve minutes each. Each test produces 720 unique rows of data, of which the middle 600 rows are selected for analysis to avoid potential anomalies at the beginning or ending of the testing. Data analysis begins and ends on the same timestamp for each cluster.

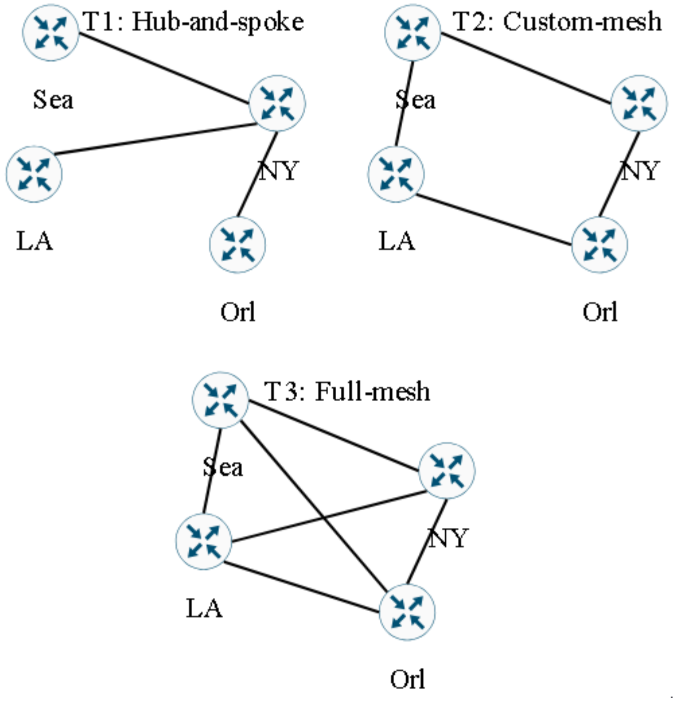

Fig. 4. WAN network topology (T).

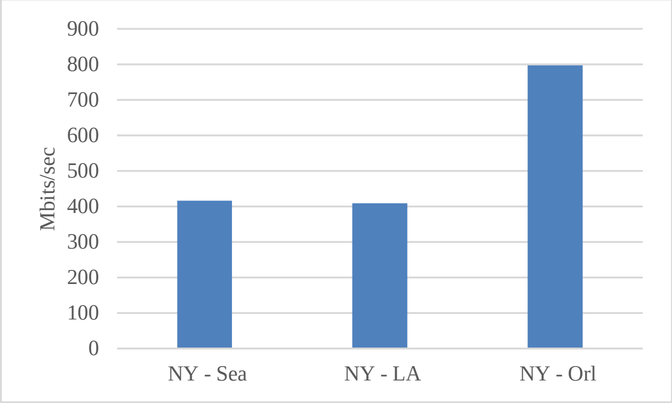

Fig. 5. Hub-and-spoke topology (T1) bitrate.

B. Hub-and-Spoke Topology

Topology 1 (T1) represents the hub-and-spoke WAN experiments. The New York data center connects to Orlando, Seattle, and Los Angeles. WAN latency is 30 milliseconds to Orlando, 58 milliseconds to Seattle, and 59 milliseconds to Los Angeles. Consistent with the cluster architecture in Fig. 2, Spark streams run from the WAN VM through dual Suricata IDSs/IPSs before reaching the primary and secondary Apache Hadoop name nodes. Data streams over three WANs are sent to the primary and secondary name nodes of each big data cluster. The name nodes load balance 128 MB HDFS block writes with a replication factor of three across the data nodes.

Fig. 5 outlines the network bitrate from the WANs to the name nodes measured in megabits per second (mbits/sec). From the New York data center, the rates are 416.496 mbits/sec to Seattle, 409.346 mbits/sec to Los Angeles, and 796.833 mbits/sec to Orlando. The mean bitrate for the hub-and-spoke topology is 540.892 mbits/sec.

C. *Custom-Mesh Topology*

Topology 2 (T2) represents the custom-mesh WAN experiments. In the custom-mesh network topology, the IDSs/IPSs protect the big data systems at the edge of each LAN. Dual routes exist through each IDS/IPS to the primary and secondary Hadoop name nodes. WANs have redundant paths to each LAN, allowing data streams alternative routes in case of a network failure. Testing establishes a total of eight data streams to Apache Spark. For example, in Fig. 4, New York has a data stream from Seattle and Orlando.

In the custom-mesh topology, the New York data center connects to Orlando and Seattle. WAN latency is 30 milliseconds to Orlando and 58 milliseconds to Seattle. Data streams from New York to Los Angeles route through either Seattle or Orlando. The Los Angeles data center connects to Seattle and Orlando. WAN latency from Los Angeles is 26 milliseconds to Seattle and 52 milliseconds to Orlando.

Fig. 6 outlines the network bitrate from the WANs to the name nodes measured in megabits per second (mbits/sec). From the New York data center, the rates are 795.578 mbits/sec to Orlando and 415.931 mbits/sec to Seattle. Rates from Los Angeles to Seattle are 915.41 mbits/sec and Los Angeles to Orlando 464.22 mbits/sec. The mean bitrate for the custom-mesh topology is 647.729 mbits/sec, which is 106.837 mbits/sec greater than the hub-and-spoke network topology.

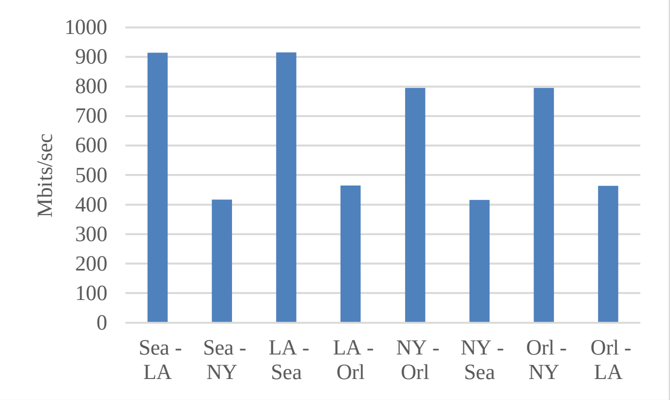

Fig. 6. Custom-mesh topology (T2) bitrate.

D. *Full-Mesh Topology*

Topology 3 (T3) is a full-mesh WAN design. As highlighted in Fig. 4, data centers have WAN paths to each city, providing the most redundancy of the designs. Twelve data streams are sent to the primary and secondary name nodes of each big data cluster through dual IDSs/IPSs. This is shown in the cluster architecture in Fig. 2.

Within the full-mesh topology, New York has WAN connections to data centers in Orlando, Seattle, and Los Angeles. In sequence, WAN latency from the New York data center to Orlando is 30 milliseconds, to Seattle 58 milliseconds, and to Los Angeles 59 milliseconds. Likewise, the Los Angeles data center connects to Seattle, Orlando, and New York. Los Angeles WAN latency to Seattle is 26 milliseconds and 52 milliseconds to Orlando. Orlando to Seattle WAN latency is the largest at 71 milliseconds.

Fig. 7 shows the network bitrate from the WANs to the name nodes measured in megabits per second. New York data center bitrates are 761.068 mbits/sec to Orlando, 414.98 mbits/sec to Seattle, and 409.33 mbits/sec to Los Angeles. Los Angeles data center bitrates are 462.771 mbits/sec to Orlando and 870.065 mbits/sec to Seattle. Seattle bitrates are 882.696 mbits/sec to Los Angeles, 414.995 mbits/sec to New York, and 341.19 mbits/sec to Orlando. The mean bitrate for the full-mesh topology is 544.637 mbits/sec. The mean rate is 3.745 mbits/sec more than the hub-and-spoke topology and 103.092 mbits/sec less than the custom-mesh topology.

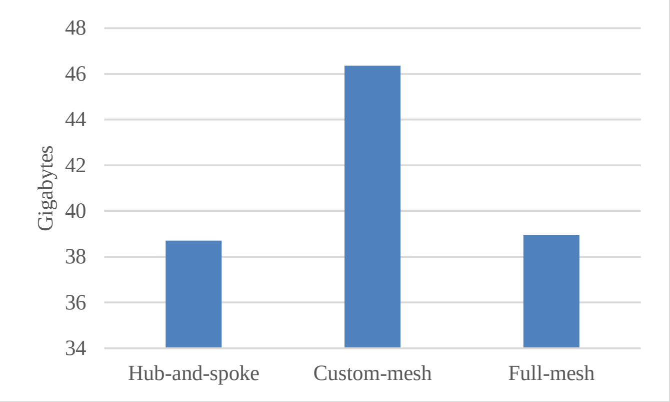

Fig. 7. Mean WAN data stream transfers in gigabytes.

E. *Streaming Data Transfers*

Fig. 8 highlights the mean data transfer rates of the Apache Spark streams through the WAN links. Fig. 9 illustrates the total data transfer rates of the Apache Spark streams through the WANs.

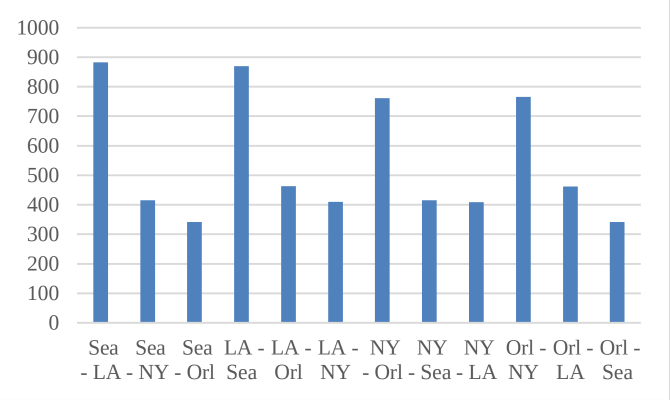

Fig. 8. Full-mesh topology (T3) bitrate in mbits/sec.

Gigabytes were converted from megabytes for the total data stream transfers. Total gigabytes transferred across the WAN network links for the hub-and-spoke network topology is 116.116. Mean gigabytes transferred between the data center sites is 38.705. Custom-mesh produces a mean of 46.364

gigabytes and a total of 370.915 gigabytes. Full-mesh network topology delivers a mean of 38.956 gigabytes and a total data transfer of 467.474 gigabytes.

Full-mesh has a mean data transfer rate slightly greater than hub-and-spoke. On the contrary, mean custom-mesh data transfer produces 7.659 gigabytes more than the hub-and-spoke network topology and 7.408 gigabytes more than the full-mesh topology.

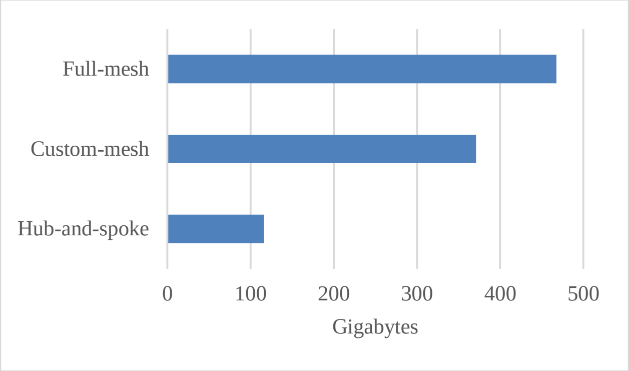

Fig. 9. Sum WAN data stream transfer in gigabtyes.

F. WAN Performance

Table II highlights the total WAN latency of each network topology along with the total amount of data transfer from the data streams. Table II also notes the number of internet service provider (ISP) leased lines used for each WAN topology in the experiments. Hub-and-spoke network topology results in an average of 38.705 gigabytes of data transfer per ISP leased line. Custom-mesh has an average data transfer of 92.728 gigabytes per leased line while full-mesh has an average data transfer of 77.912 gigabytes per leased line.

TABLE II. WAN LATENCY VERSUS DATA TRANSFER

Topology	ISP Leased Lines	Total WAN Latency	Total Data Transfer
Hub-and-spoke	3	147 ms	116.1166 Gbs
Custom-mesh	4	166 ms	370.9158 Gbs
Full-mesh	6	296 ms	467.4745 Gbs

G. Summary

To measure whether IDS/IPS placement impacts geo-distributed big data systems, the researchers study WAN connections between the remote cities of Los Angeles, Orlando, New York, and Seattle. Data centers in each city host big data clusters running Apache Hadoop and Spark. Data streams are sent through the IDSs/IPSs from the WANs to each of the four big data clusters. The researchers develop a novel Python application that uses PySpark streaming classes to facilitate real-time geo-distributed massive data streaming. Performance measures use raw network traffic data to demonstrate the results of three prominent network designs; hub-and-spoke, custom-mesh, and full-mesh. Results illustrate the ability to load balance data streams through IDS/IPS locations with the lowest WAN latency in custom-mesh topology while continuing to offer alternative network paths to geo-distributed data centers. Next, the authors discuss these results.

V. Discussion

Live data streams across four unique geo-distributed data centers show variability in real-life scenarios. Though researchers were able to optimize bandwidth through three different WAN topologies, there are clear performance differences that decision makers should consider when architecting secure clusters for WABD.

A. Geo-Distributed IDS/IPS Placement Performance

Researchers were able to achieve the fastest data streams across geo-distributed data centers using a custom-mesh network design. IDS/IPS placement in the custom-mesh network topology achieves a mean of 106.836 mbits/sec more network bitrate than the optimized hub-and-spoke topology. Similarly, on average the custom-mesh topology is 103.091 mbits/sec faster than the full-mesh design.

In this study, IDS/IPS placement within the full-mesh network design results in slightly faster mean bandwidth available for WABD data streams than the hub-and-spoke network topology. Full-mesh benefits from a mean of 3.745 additional mbits/sec across the WAN architecture. While full-mesh network topology has additional benefits over both hub-and-spoke and custom-mesh such as more fault tolerance, this comes at the cost of expensive WAN bandwidth [22].

In the experiments, hub-and-spoke has three ISP leased lines. Custom-mesh has four leased lines while full-mesh has six leased lines. When reviewing Table II, custom-mesh is able to transfer 54.023 more gigabytes of streaming data through the IDSs/IPSs per leased line than the hub-and-spoke network topology. This comes at a cost of only one additional ISP leased line in these experiments. However, it also adds an extra path of redundancy between each site, eliminating potential single points of failure in the hub-and-spoke network topology.

Custom-mesh also transfers 14.916 gigabytes more data per leased line than the full-mesh topology. Despite this result, full-mesh benefits from an additional redundant path to subsequent data centers. Similar to custom-mesh, full-mesh provides more bandwidth than the hub-and-spoke topology. In comparison, full-mesh produces 39.206 gigabytes more data per leased line than hub-and-spoke. While data centers in the full-mesh design could experience several network failures before losing complete connectivity to another site, it also comes at the cost of three additional ISP leased lines over the custom-mesh topology.

B. Limitations

This paper does not address pricing, which limits the analysis of geo-distributed IDS/IPS placements specific to big data streaming. Although the results of this study give some indication of potential efficiency of various IDS/IPS locations for geo-distributed big data systems, it is financially inconclusive as many variables determine the costs of implementing and maintaining each of the network designs in real-life environments. For instance, in study [15], custom topology resulted in considerable pricing differences for data transfer alone, ranging from $0.02 to $0.25 per GB of data transfer.

Research efforts are advancing big data worker node placement using several available data points. For example, in study [17] the simple-additive weighting method strategically places data streaming tasks using data transmission cost, latency, and bandwidth. However, algorithms lean upon available network data without considering human factors. Future research is important to consider more closely defined pricing models for IDS/IPS placement specific to geo-distributed WAN data streaming.

This paper is also limited to initial benchmarking of three traditional WAN topologies that use manual IDS/IPS placement methods. To advance this research, existing algorithms could consider IDS/IPS latency within avant-garde WAN topologies. For example, the approximate parameter server placement (APSP) algorithm proposed by study [20] could be tested in IDS/IPS environments to identify if the randomized rounding method is still applicable. Similarly, future research could test IDS/IPS locations using WAN topology-aware frameworks introduced in study [15] and study [17].

Finally, IDS/IPS benchmarking is limited to a Python streaming application engineered for Apache Spark. Similar to study [17], researchers may consider other big data streaming systems like Apache Flink and Apache Storm along with varied streaming applications developed in Scala and/or Java.

VI. Conclusion

This paper develops a PySpark streaming application in Python capable of benchmarking geo-distributed data centers secured by IDSs/IPSs. The application sends data streams across the WAN topologies of hub-and-spoke, custom-mesh, and full mesh. In each topology, the researchers optimize IDS/IPS placement using industry best practices and experimentation. The proposed placements show several tradeoffs. Hub-and-spoke has the least aggregate WAN latency and the fewest number of ISP leased lines but at the cost of single points of failure within the WAN topology. Custom-mesh network topology benefits from the fastest raw network performance. It also has dual paths to geo-distributed data centers at a cost of only one additional ISP leased line. Full-mesh offers the most fault tolerance and raw data streaming bandwidth. However, it requires a minimum of two additional ISP leased lines over custom-mesh. In summary, IDS/IPS placement in custom-mesh network topology allows engineers to customize the amount of high availability across WANs while reducing associated costs of leased lines. Advancing this work could include evolving network topology for WANalytics, automating IDS/IPS placement, testing alternative big data streaming systems, and incorporating financial costs into IDS/IPS placement determination. Subsequently, researchers may consider testing existing or new worker node placement algorithms in WABD IDS/IPS environments.

References

[1] M. Bergui, S. Najah, and N. S. Nikolov, “A survey on bandwidth-aware geo-distributed frameworks for big-data analytics,” *Journal of Big Data*, vol. 8, no. 40, pp. 1-26, Feb. 2021, doi: 10.1186/s40537-021-00427-9.

[2] “Cluster Mode Overview,” The Apache Software Foundation, June, 2023. [Online]. Available: https://spark.apache.org/docs/latest/cluster-overview.html.

[3] A. Vulimiri, C. Curino, P. Godfrey, T. Jungblut, K. Karanasos, J. Padhye, and G. Varghese, “WANalytics: Geo-distributed analytics for a data intensive world,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, in SIGMOD ’15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 1087–1092. doi: 10.1145/2723372.2735365.

[4] H. Wang and B. Li, “Mitigating bottlenecks in wide area data analytics via machine learning,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 155–166, 2020, doi: 10.1109/TNSE.2018.2816951.

[5] H. Wang, D. Niu, and B. Li, “Turbo: Dynamic and decentralized global analytics via machine learning,” in *Proceedings of the ACM Symposium on Cloud Computing*, in SoCC ’18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 14–25. doi: 10.1145/3267809.3267812.

[6] A. Jonathan, A. Chandra, and J. Weissman, “Multi-query optimization in wide-area streaming analytics,” in *Proceedings of the ACM Symposium on Cloud Computing*, New York, NY, USA, 2018, pp. 412–425. doi: 10.1145/3267809.3267842.

[7] A. Yassine, A. A. N. Shirehjini, and S. Shirmohammadi, “Bandwidth on-demand for multimedia big data transfer across geo-distributed cloud data centers,” *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1189–1198, Dec. 2020, doi: 10.1109/TCC.2016.2617369.

[8] K. Rajah, S. Ranka, and Y. Xia, “Advance reservations and scheduling for bulk transfers in research networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1682–1697, Nov. 2009, doi: 10.1109/TPDS.2008.250.

[9] K. Rajah, S. Ranka, and Y. Xia, “Scheduling bulk file transfers with start and end times,” *Computer Networks*, vol. 52, no. 5, pp. 1105–1122, Apr. 2008, doi: 10.1016/j.comnet.2007.12.005.

[10] Y. Wu, Z. Zhang, C. Wu, C. Guo, Z. Li, and F. C. M. Lau, “Orchestrating bulk data transfers across geo-distributed datacenters,” *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 112–125, Mar. 2017, doi: 10.1109/TCC.2015.2389842.

[11] T. Nandagopal and K. P. N. Puttaswamy, “Lowering inter-datacenter bandwidth costs via bulk data scheduling,” in *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, May 2012, pp. 244–251. doi: 10.1109/CCGrid.2012.70.

[12] N. Laoutaris, G. Smaragdakis, R. Stanojevic, P. Rodriguez, and R. Sundaram, “Delay-tolerant bulk data transfers on the internet,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1852–1865, Dec. 2013, doi: 10.1109/TNET.2012.2237555.

[13] S. Yue, X. Lin, W. Sun, and W. Hu, “Modeling sparse store-and-forward bulk data transfers in inter-datacenter networks with multiple congested links,” *IEEE Transactions on Cloud Computing*, vol. 11, no. 3, pp. 1–14, 2022, doi: 10.1109/TCC.2022.3225977.

[14] C. Vicentini, A. Santin, E. Kugler Viegas, and V. Abreu, “SDN-based and multitenant-aware resource provisioning mechanism for cloud-based big data streaming,” *Journal of Network and Computer Applications*, vol. 126, Nov. 2018, doi: 10.1016/j.jnca.2018.11.005.

[15] H. Mostafaei and S. Afridi, “SDN-enabled resource provisioning framework for geo-distributed streaming analytics,” *ACM Trans. Internet Technol.*, vol. 23, no. 1, Feb. 2023, doi: 10.1145/3571158.

[16] B. Cheng, A. Papageorgiou, F. Cirillo, and E. Kovacs, “GeeLytics: Geo-distributed edge analytics for large scale IoT systems based on dynamic topology,” in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 565–570. doi: 10.1109/WF-IoT.2015.7389116.

[17] H. Mostafaei, S. Afridi, and J. Abawajy, “Network-aware worker placement for wide-area streaming analytics,” *Future Generation Computer Systems*, vol. 136, pp. 270–281, Nov. 2022, doi: 10.1016/j.future.2022.06.009.

[18] B. Heintz, A. Chandra, and R. K. Sitaraman, “Optimizing Timeliness and Cost in Geo-Distributed Streaming Analytics,” *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 232–245, 2020, doi: 10.1109/TCC.2017.2750678.

[19] D. Kumar, S. Ahmad, A. Chandra, and R. K. Sitaraman, “AggNet: Cost-Aware Aggregation Networks for Geo-distributed Streaming Analytics,” in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, 2021, pp. 297–311. doi: 10.1145/3453142.3491276.

[20] Y. Li, C. Fan, X. Zhang, and Y. Chen, "Placement of parameter server in wide area network topology for geo-distributed machine learning," *Journal of Communications and Networks*, vol. 25, no. 3, pp. 370–380, 2023, doi: 10.23919/JCN.2023.000021.

[21] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.

[22] S. A. Ibrahim Hussein, F. W. Zaki, and M. M. Ashour, "Performance evaluation of software-defined wide area network based on queueing theory," *IET Networks*, vol. 11, no. 3–4, pp. 128–145, May 2022, doi: 10.1049/ntw2.12039.

[23] "Cisco Extended Enterprise SD-WAN Design Guide," *Cisco Systems, Inc.*, July, 2024. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/EE/DG/ee-WAN-dg.pdf.

[24] "U.S. Network Latency," *AT&T*, June, 2024. [Online]. Available: https://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html

[25] "Suricata user guide," *Open Information Security Foundation*, June, 2024. [Online]. Available: https://docs.suricata.io/en/latest/.

[26] M. Kalinin and V. Krundyshev, "Security intrusion detection using quantum machine learning techniques," *Journal of Computer Virology and Hacking Techniques*, vol. 19, no. 1, pp. 125–136, Mar. 2023, doi: 10.1007/s11416-022-00435-0.

[27] "pyspark.streaming.StreamingContext," *The Apache Software Foundation*, June, 2024. [Online]. Available: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.streaming.StreamingContext.html.

[28] V. Nunes, J. Brás, A. Carvalho, D. Barradas, K. Gallagher, and N. Santos, "Enhancing the Unlinkability of Circuit-Based Anonymous Communications with k-Funnels," *Proceedings of the ACM on Networking.*, vol. 1, pp. 1-26. Nov. 2023, doi: 10.1145/3629140.

[29] S. Y. Yu *et al.*, "Analysis of NVMe over fabrics with SCinet DTN-as-a-Service," *Cluster Computing*, vol. 25, Aug. 2022, doi: 10.1007/s10586-021-03433-x.

[30] "TCPDUMP & LIBPCAP," *The Tcpdump Group*, June, 2024. [Online]. Available: https://www.tcpdump.org/.

[31] "nmon for Linux," *IBM*, June, 2024. [Online]. Available: http://nmon.sourceforge.net.