Denial-of-Service Attacks in a Software-Defined LEO Constellation Network

Dennis Agnew
Dept of Electrical & Computer Engineering
University of Florida
Gainesville, FL USA
dennisagnew@ufl.edu

Janise McNair
Dept of Electrical & Computer Engineering
University of Florida
Gainesville, FL USA
mcnair@ece.ufl.edu

***Abstract*—Satellite communication (SATCOM) is a critical infrastructure for tactical networks–especially for the intermittent communication of submarines. To ensure data reliability, recent SATCOM research has begun to embrace several advances, such as low earth orbit (LEO) satellite networks to reduce latency and increase throughput compared to long-distance geostationary (GEO) satellites, and software-defined networking (SDN) to increase network control and security. This paper proposes an SD-LEO constellation for submarines in communication networks. An SD-LEO architecture is proposed, to Denial-of-Service (DoS) attack detection and classification using the extreme gradient boosting (XGBoost) algorithm. Numerical results demonstrate greater than ninety-eight percent in accuracy, precision, recall, and F1-scores.**

***Keywords*—software-defined networking (SDN), cybersecurity, SATCOM, low earth orbit (LEO) satellites, machine learning, submarines**

I. Introduction

Satellite communication (SATCOM) is critical to tactical military networks, and satellites frequently serve as space-based relay stations, receiving, amplifying, and retransmitting signals back to ground entry points (GEPs). With the recent development of software-defined networking (SDN), researchers are exploring novel ways to integrate SDN with tactical SATCOM networks [1]. SDN is a networking concept that separates the control plane from the data plane of network forwarding devices to consolidate control within one or more controllers, allowing for greater management, visibility, and security [2], [3].

Recently, the private sector has developed initiatives for SDN SATCOM networks, such as SpaceX's Starlink and Amazon's Kuiper, that depend on software-defined low earth orbit (SD-LEO) satellite constellations. The Army [4] and the Department of Defense (DoD) [5] and have been working closely with Starlink and other vendors on the development of LEO constellations for use of the military. With support from both the US government and Ukraine, military forces in Ukriane have relied on Starlink as the backbone of its communication neworks [6]. LEO constellations provide higher throughput and lower latency than traditional geostationary (GEO) satellites [2]. As shown in Figure 1, this infrastructure can also be used for tactical SD-LEO constellations for military entities such as submarines, which depend on the discrete delivery of information to operate covertly in hostile locations all over the world. While on patrol and submerged for months at a time, submarines rarely communicate back to GEPs. To communicate with GEPs, submarines must surface from the oceans' depths. To avoid being discovered by opposing forces, submarines must be able to send and receive information as soon as possible so that they can resubmerge quickly. Currently, submarine crews typically communicate using GEO satellites at a distance @ 36,000km and a $\sim 250ms$ propagation delay. Future tactical networks may use LEO constellations at 1500km or less, with less than $\sim 30ms$ propagation delay [2].

In addition to lower delays, submarine communication links require heightened security. Messages are highly classified and often time-sensitive, which often make them prime targets for malicious forces. As shown in Figure 1, one common approach for malicious agents (e.g. hostile unmanned aerial vehicles (UAVs)) is to initiate denial-of-service (DoS) attacks, which are designed to overload a target with incoming network traffic in an effort to reduce available throughput or to completely crash the system. UAVs pose a threat to SATCOM networks, necessitating additional research, such as the one presented in this paper, to address this vital area of communication [7]. Network defense models are needed that can detect and mitigate these threats.

Previous researchers [1], [8] have investigated developing shipboard networks based on SDN or suggested SDN SATCOM networks for general tactical environments. Other research [9] has developed DoS defense techniques for ground stations. However, to our knowledge, no prior work has proposed an SD-LEO constellation network for submarines, nor a detection method for these networks against DoS attacks. This research contributes a DoS detection and classification framework for SD-LEO constellations for submarines, by employing extreme gradient boosting (XGBoost).

This paper is organized as follows. Section II presents the related work. Section III presents background information on SDN, network performance statistics used, types of DoS attacks, and the envisioned DoS attack scenario. Section IV

'DISTRIBUTION STATEMENT A. Approved for public release: distribution is unlimited.' This work was supported by the National Science Foundation under Grant Number 1738420 and by the University of Florida/Harris Corporation Excellence in Research Fellowship.

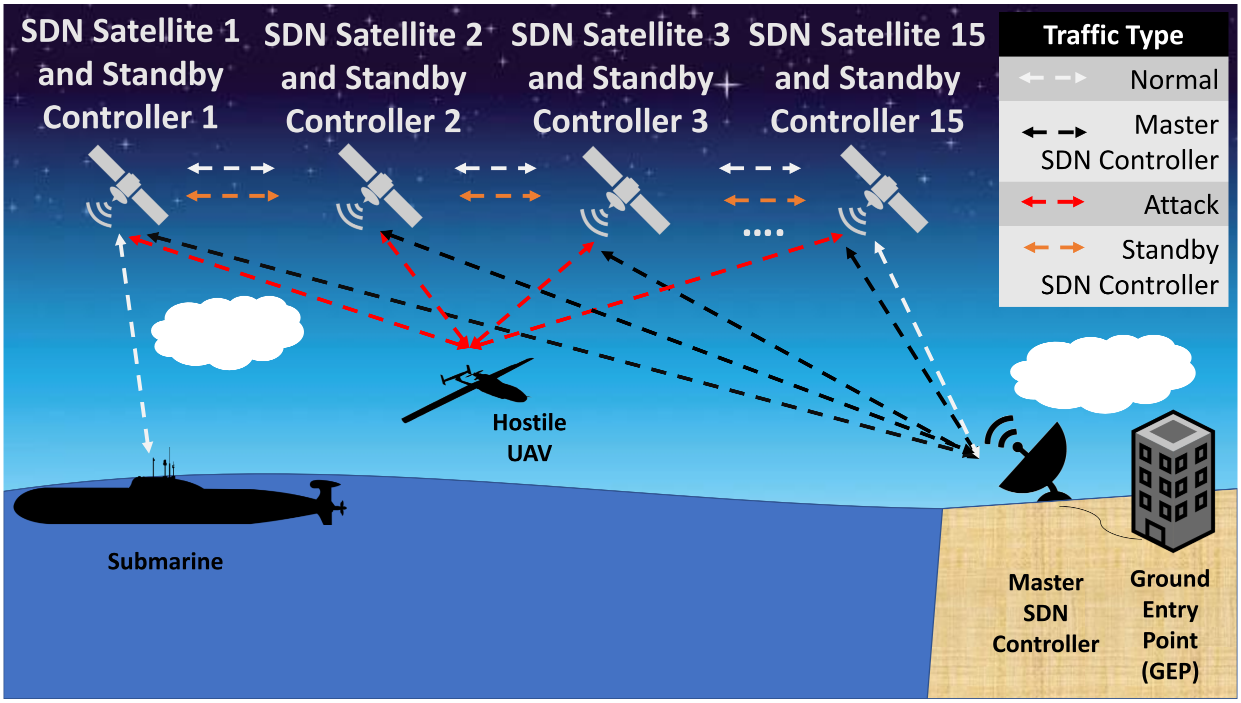
SDN Satellite 1 and Standby Controller 1
SDN Satellite 2 and Standby Controller 2
SDN Satellite 3 and Standby Controller 3
SDN Satellite 15 and Standby Controller 15
Traffic Type
Normal
Master SDN Controller
Attack
Standby SDN Controller
Hostile UAV
Submarine
Master SDN Controller
Ground Entry Point (GEP)

Fig. 1: Example SD-LEO Constellation for Submarines

discusses the SD-LEO constellation envisioned architecture. Section V discusses the simulation of SATCOM traffic of our proposed SD-LEO constellation and the creation of the datasets necessary to train our ML model for DoS attack detection. Section VI discusses the ML model classification DoS results, which detect and classify traffic based on attack severity with accuracy, precision, recall, and F1- scores above 98%. Lastly, Section VII concludes the paper.

II. Related Work

The framework utilizes software-defined networking to manage satellite and submarine communication. Within tactical networks, a variety of SDN applications have been investigated by researchers. Previous research [1], [8] has investigated the use of SDN in tactical networks due to its enhanced visibility, security, and network management potential.

In [8], researchers proposed to replace a naval ship's legacy onboard network with one that supports multiple SATCOM connections for improved communication. They proposed that each onboard switch be replaced with SDN switches. In addition, the researchers employed Multi-Path Transmission Control Protocol (MPTCP) to enhance end-to-end data delivery by creating multiple subflows within a single TCP session. Their framework allowed for improved load balancing of multiple onboard SATCOM connections. But their framework does not have an LEO constellation for naval entities or network security, like detecting DoS attacks, to protect such a framework, as this paper explains.

[1] established an SDN-based battlefield with multiple SATCOMs and unmanned Aerial Vehicles (UAVs). Using SDN and MPTCP, researchers created an integrated testbed that facilitated enhanced control, visualization, and link management. Their architectures monitored and relayed information throughout the framework using GEO satellites. In addition, their custom testbed enables real-time network emulation, which could be utilized in future projects. However, their framework does not include LEO constellations or consider network security such as defenses against DoS cyberattacks as proposed in this paper.

[9] proposed and developed a DoS and Distributed DoS (DDoS) attack detection framework for satellite networks. Their work focuses on mitigating Internet Control Message Protocol (ICMP) DoS and DDoS attacks. An attacker could gain access to a satellite ground station (GS) and use it to launch ICMP attacks against other interconnected ground stations. The researchers created a testbed that simulated the connection of four GS to a GEO satellite. The researchers simulated in Matlab the effects of a compromised GS launching ICMP DoS/DDoS attacks against other compromised GS and developed a mitigation technique that analyzes the average number of ICMP packets sent and restricts connections to adversary compromised GS. Their framework neither provides nor mentions protection for LEO satellites or tactical SATCOM networks.

Ongoing efforts by military [4] [5] aim to create LEO constellations with the help of commercial vendors. Furthermore, military efforts are developing efforts against jamming. However, these efforts fail to consider SD-LEO constellations for submarines nor mention DoS attack detection against illegitimate access users such as UAVs. Therefore, the con-

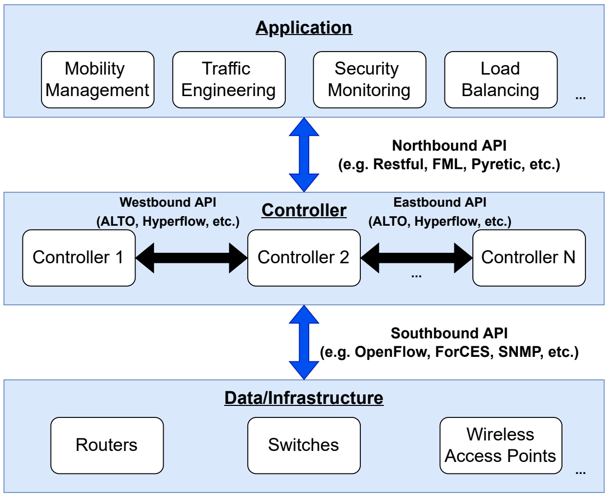

Fig. 2: General SDN Architecture [10]

tributions of this paper are as follows:

- A novel Software- Defined LEO constellation architecture proposed for use of submarines
- A novel DoS detection and classification severity framework for SD-LEO constellations

III. Background

A. Software-Defined Networking (SDN)

The concept and practice of SDN is appealing to those in the networking industry due to the programmability of network devices. SDN was coined at Stanford University to describe the concepts and techniques for a software protocol that allows a server to tell network switches where to send packets [11]. OpenFlow is one of the earliest software-defined networking (SDN) standards. In order to allow the SDN controller to more easily interact with the forwarding plane of network devices like switches and routers, both physical and virtual (hypervisor-based), it first defined the communication protocol in SDN architectures. As represented in figure 2, SDN can be described along three planes:

1) **Application Plane:** It covers SDN applications for network management, policy implementation, and security services.
2) **Control Plane:** This is a logically centralized control framework that runs the network operating system and provides hardware abstractions to SDN applications. A flow in SDN is a set of instructions followed by a series of packets between the source and the destination. Controllers populate the flow tables of forwarding devices with the flows.
3) **Data Plane:** A collection of forwarding components used to move traffic flows in response to instructions from the control plane.

As shown in the diagram, routers, switches, and access points comprise the infrastructure layer. This layer, which represents the physical network equipment in the network, forms the data plane. Through application programming interfaces (APIs), information is transmitted across SDN architecture planes. The controller uses southbound APIs such as OpenFlow [11], ForCES [12], PCEP [13], NetConf [14], or IRS [15] to communicate with the data plane. Multiple controllers interact via Westbound and Eastbound APIs, such as ALTO [16] or Hyperflow [17], when present. The topmost layer is the application plane. At this layer, the network operator may utilize functional applications for energy efficiency, access control, mobility management, and/or security management. The application layer uses northbound APIs such as FML [18], Procera [19], Frenetic [20], and RESTful [21] to communicate with the control layer. The network operator can use these APIs to communicate the required modifications to the control layer, thereby enabling the controller to make the necessary adjustments to the infrastructure layer.

B. Network Performance Statistics

The Poisson traffic model has been utilized to model LEO constellation traffic [22]. Each satellite represents the M/M/c queue [23], i.e. cc $\geq$ 1, in which packet arrival is governed by a Poisson process and queue service time is governed by an exponential distribution. The following equation describes the volume or intensity of traffic:

$$P_{util} = \frac{\lambda}{c\mu} \tag{1}$$

The arrival rate of the packets is represented by λ, the number of servers is represented by c (i.e. 15 satellites) and the service rate of the packets is represented by μ. The inter-arrival time (IAT) is the time difference (Δt) between packet arrivals. It has an exponential distribution with parameter λ. The probability density function is defined as the following for t $\geq$0:

$$f(t) = \lambda e^{\lambda t}. \tag{2}$$

The average IAT is defined as

$$IAT = \frac{1}{\lambda} \tag{3}$$

The service time follows an exponential distribution with parameter μ. The probability density function is as follows:

$$g(s) = \mu e^{-\mu s}, \forall \geq 0 \tag{4}$$

where $\frac{1}{\mu}$ is the average service time of the system. Utilizing Little's theorem, the total waiting time is defined as transmission delays (TD), and represented as the following:

$$W = TD = \frac{1}{\mu - \lambda} \tag{5}$$

The normal distribution of network packet arrivals (i.e. non-attacked packets) into each system was decided by the probability of witnessing a number of packet arrivals in a period from [0,T]. This equation is used to model the traffic volume of the bus:

$$P(n\ arrivals\ in\ interval\ T) = \frac{(\lambda T)^n e^{-\lambda T}}{n!} \tag{6}$$

where T is the IAT, and n represents the number of packets. The packet-count (PC) is modeled as the following:

$$PC = \lambda T \tag{7}$$

C. Types of Denial-of-Service Attacks

DoS and DDoS attacks have been known to disrupt terrestrial networks. However, DoS and DDoS attacks are becoming more common in satellite networks and a variety of DoS attacks can disrupt satellite networks [9]:

- **Type of Service (ToS) Floods:** By foraging an IP packet header, an adversary can control explicit congestion notification (ECN) and differentiated services (DiffServ) flags in a ToS attack. An attacker spoofs the ECN field to reduce the throughput of satellite-linked client-server connections, making the server unavailable for legitimate requests. DoS attacks can be exacerbated by using the DiffServ flag to prioritize attack traffic over legitimate traffic.
- **Synchronization (SYN) Floods:** SYN floods overflow network resources by using half-open Transport Control Protocol (TCP) connections. After a client sends a SYN message, a server waits for an Acknowledgement (ACK) packet to open a half-open TCP connection. The server sends a SYN-ACK message for every client SYN and waits for the final acknowledgment. The server keeps a database of connections awaiting acknowledgements. As a server's resources are finite, creating many open connections can intentionally exhaust them, making them unavailable for legitimate traffic.
- **Ping Floods:** In ping floods, an adversary takes control of a set of network nodes and sends a large number of pings from multiple servers worldwide, flooding the target with false traffic and making it unavailable. If the network node is a GEP or satellite, its dependent nodes will lose connectivity or have a lower quality of service (QoS).

These attacks are capable of wreaking havoc on both terrestrial and SATCOM networks. These attacks impact the Quality of Experience (QoE) of network nodes and result in decreased throughput and increased latency, or transmission delay. Our proposed framework uses network performance statistics described in section III-B to identify satellite nodes impacted by the presence of these attacks.

D. Denial-of-Service Attack Scenario

Denial-of-Service attacks can cause irreparable damage to SATCOM networks and cripple vital network infrastructure [9]. Consequently, it is essential to develop DoS attack detection and mitigation strategies that can identify these attack types. Figure 1 details our envisioned DoS attack scenario. It depicts an enemy UAV with unauthorized access to the SATCOM network. It follows a predetermined route while transmitting illegal, false packets to the constellation's satellites, thereby consuming satellite resources. Therefore, the satellites experience a decrease in service rate, μ, which increases the latency experienced by the submarine's legitimate traffic to the GEP. First, we record network performance statistics such as the interarrival times, transmission delays, and packet count described in section III-B. The data is used to train a machine learning model to detect and identify the types of attacks.

IV. SD-LEO Constellation Network Architecture

As depicted in Figure 1, we envision a 15-satellite SD-LEO constellation to provide global intermittent connectivity [24]. At one end of the constellation, SDN Satellite 1 connects the submarine. At the other end of the constellation, SDN Satellite 15 connects to the ground-based master controller. While on patrol, the submarine will send packets to any nearby satellite it can reach. The master controller will install flows through the satellites' routing path to successfully route the information through the constellation to the destination GEP. Additional standby controllers will be aboard each LEO satellite in the event that the master controller at the GEP in a particular location cannot be reached. If unreachable, the standby controllers of the satellites will temporarily take over and instruct the satellite on the optimal path to route packets to the GEP. As depicted in Figure 2, this inter-controller communication makes use of the East/West API employed by distributed SDN frameworks. This will enable a robust and adaptable solution depending on network routing needs.

The GEP and master controller are tasked with monitoring network traffic and implementing a machine learning (ML) process to detect potential DoS attack events. The master controller can collect satellite statistics, including IAT, TD, packets sent and received, etc. This data can then be labeled by a network operator and used to train our machine learning model. After successful training, the GEP and master controller will be able to use a machine learning algorithm to detect future DoS attacks. Upon detecting a malicious satellite, the master controller is then able to use SDN flows to isolate compromised satellites and reroute connections through safer links until the network operator or team can investigate the compromised device.

V. Simulation

SimComponents [25], a network traffic simulation program built on the SimPY process-based discrete-event simulation framework, was used to model SD-LEO constellation traffic using user datagram protocol (UDP) packets found in LEO constellations. The packet inter-arrival time and packet latency on the links were modeled using representative LEO SATCOM testbed values [1], [24]. Furthermore, we added Gaussian noise to the obtained data to increase data variability.

We completed a 1-hour network traffic simulation for the 15 satellite described in Section IV on a Linux server running Redhat Enterprise 8.6 with an Intel® 5th Gen Core$^{\text{TM}}$ i5-6500 CPU @ 3.20GhZ. The traffic was composed of both attack/DoS traffic and normal traffic. Attack/DoS traffic made up $\sim 7\%$ of the dataset. The GEP monitors network traffic to detect major changes in data delivery that may indicate the beginning of a DoS attack. To simulate the normal and attack classes, we varied the throughput by reducing the port-based rate, creating a loss of 0% or 90% of traffic, and defined these cases as class 0 and 1, where class 0 is normal traffic and classes 1 is DoS-affected traffic. From the data, we produced four CSV files with 3600 rows x 15 column matrices, one for each of the following four features: inter-arrival time, transmission delay/latency, number of packets sent, and number of packets received. Each row is a time stamp for each second in an hour, and each column represents a satellite's full communication. Our data set was then processed using the extreme gradient boosting (XGBoost) [26], a gradient-boosting algorithm for decision trees. We chose it for its quick training times, interpretability, simplicity, and enhanced performance over other comparable models such as neural networks [27]–[29].

VI. Numerical Results

K-fold cross-validation ($k = 5$) was applied to the XGBoost output. We computed the average and standard deviation of the accuracy, precision, recall, and F1-score of the five folds. As shown in Table I, our model was able to classify DoS attacks and normal traffic with scores greater than $> 98\%$. Using these methods, a master controller located in the GEP would be able to detect DoS attacks.

To further demonstrate the efficacy of our detection algorithm, we included the results of each individual class in Table II for the output of K-fold cross-validation. Each individual class achieved performance exceeding $> 97\%$. Class 0 performed the best due to the natural skew of training examples used to train the XGBoost algorithm, as only 7% of attack data was incorporated. The class with the lowest overall performance was class 2. This may be because the algorithm has a harder time distinguishing this attack class from class 0 due to similar network statistics observed during attack simulation. XGBoost is able to identify differences in the traffic pattern in order to make the correct selection, despite the fewer attack samples available for training in comparison to normal traffic. However, class 1 still performed above $> 97\%$.

TABLE I: Performance Results for k-fold Cross Validation (k=5) for Normal and DoS Traffic

Algorithm	Avg. Accuracy $\mu \pm \sigma$	Avg. Precision $\mu \pm \sigma$	Avg. Recall $\mu \pm \sigma$	Avg. F1-score $\mu \pm \sigma$
XGBoost	99.77 ± 0.03	99.33 ± 0.13	98.87 ± 0.18	**99.10 ± 0.13**

TABLE II: Individual Performance Results for k-fold Cross Validation (k=5) for Classes 0 and 1

Class	Avg. Accuracy $\mu \pm \sigma$	Avg. Precision $\mu \pm \sigma$	Avg. Recall $\mu \pm \sigma$	Avg. F1-score $\mu \pm \sigma$
Class 0	99.91 ± 0.07	99.82 ± 0.05	99.91 ± 0.073	**99.87 ± 0.03**
Class 1	97.51 ± 0.73	98.81 ± 0.93	97.51 ± 0.01	**98.15 ± 0.39**

The k-fold cross-validation's cumulative confusion matrix is shown in Figure 3. These results show the ability of our classifier approach to reliably determine the classes of traffic, with class 0 being the normal traffic and class 1 resulting in 90% throughput loss. We attribute our detection system's effectiveness to XGBoost's well-documented performance, as well as the predictive features we extracted from the simulation. Additional analysis and results are presented for additional DoS attack classes within [30].

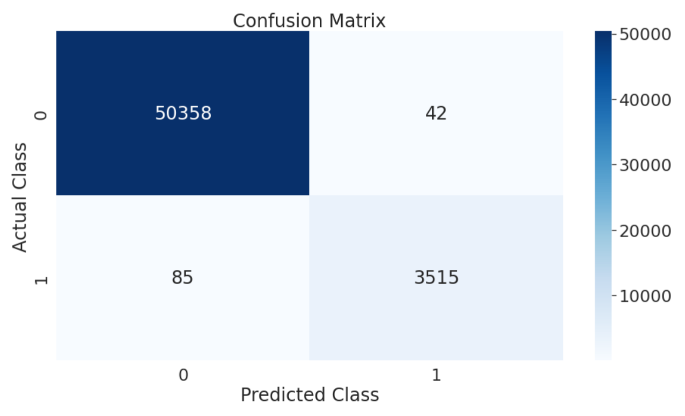

Fig. 3: Confusion Matrix of XGBoost ML Algorithm k-fold Cross Validation ($k = 5$) for Classes 0 and 1

VII. Conclusion and Future Work

A heterogeneous cyberattack detection framework that is able to detect DoS attacks by severity is proposed in this paper, as well as an SDN SATCOM network that is designed specifically for use by submarines. Our detection framework is, to the best of our knowledge, the first of its kind to apply these techniques to a SD-LEO SATCOM network. Through the utilization of SimComponent, we successfully generated the dataset required for the training of the XGBoost algorithm and our model demonstrated average accuracy, precision, recall, and F1-scores that were greater than $> 98\%$.

DoS attack classes beyond those covered by this paper are also discussed, along with their respective analyses and results in [30]. In subsequent work, we plan to develop this framework further so that it can identify a wider variety of cyberattacks, including man-in-the-middle (MITM), botnet, false data injection, and others. In addition, we will develop mitigation techniques after attack detection to keep throughput stable.

References

[1] Q. Zhao, A. J. Brown, J. H. Kim, and M. Gerla, "An integrated software-defined battlefield network testbed for tactical scenario emulation," in *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 373–378.

[2] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband leo satellite communications: Architectures and key technologies," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 55–61, 2019.

[3] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "Vnf-based service provision in software defined leo satellite networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 6139–6153, 2021.

[4] A. Walker, "Army's eyes on resilient multi-orbit satcom," Nov 2020. [Online]. Available: https://www.army.mil/article/240491/armys_eyes_on_resilient_multi_orbit_satcom

[5] V. Machi, "Space development agency," Jun 2021. [Online]. Available: https://www.sda.mil/us-military-places-a-bet-on-leo-for-space-security/

[6] M. Wall, "1,300 spacex starlink terminals with ukraine's military went offline due to funding shortfall: Report," Nov 2022. [Online]. Available: https://www.space.com/ukraine-spacex-starlink-terminals-offline-funding-shortfall

[7] P. Tedeschi, S. Sciancalepore, and R. Di Pietro, "Satellite-based communications security: A survey of threats, solutions, and research challenges," *Computer Networks*, p. 109246, 2022.

[8] S. Nazari, P. Du, M. Gerla, C. Hoffmann, J. H. Kim, and A. Capone, "Software defined naval network for satellite communications (sdn-sat)," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 360–366.

[9] M. Usman, M. Qaraqe, M. R. Asghar, and I. Shafique Ansari, "Mitigating distributed denial of service attacks in satellite networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 6, p. e3936, 2020.

[10] D. Agnew, N. Aljohani, R. Mathieu, S. Boamah, K. Nagaraj, J. McNair, and A. Bretas, "Implementation aspects of smart grids cyber-security cross-layered framework for critical infrastructure operation," *Applied Sciences*, vol. 12, no. 14, p. 6868, 2022.

[11] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[12] E. Haleplidis, J. H. Salim, J. M. Halpern, S. Hares, K. Pentikousis, K. Ogawa, W. Wang, S. Denazis, and O. Koufopavlou, "Network programmability with forces," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1423–1440, 2015.

[13] J. P. Vasseur and J. L. Le Roux, "Path computation element (pce) communication protocol (pcep)," Tech. Rep., 2009.

[14] R. Enns, "Netconf configuration protocol," Tech. Rep., 2006.

[15] G. Huston, "Analyzing the internet's bgp routing table," *The Internet Protocol Journal*, vol. 4, no. 1, pp. 2–15, 2001.

[16] R. Alimi, R. Penno, Y. Yang, S. Kiesel, S. Previdi, W. Roome, S. Shalunov, and R. Woundy, "Application-layer traffic optimization (alto) protocol," Tech. Rep., 2014.

[17] A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for openflow," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, vol. 3, 2010, pp. 10–5555.

[18] T. L. Hinrichs, N. S. Gude, M. Casado, J. C. Mitchell, and S. Shenker, "Practical declarative network management," in *Proceedings of the 1st ACM workshop on Research on enterprise networking*, 2009, pp. 1–10.

[19] A. Voellmy, H. Kim, and N. Feamster, "Procera: A language for high-level reactive network control," in *Proceedings of the first workshop on Hot topics in software defined networks*, 2012, pp. 43–48.

[20] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic: A network programming language," *ACM Sigplan Notices*, vol. 46, no. 9, pp. 279–291, 2011.

[21] W. Zhou, L. Li, M. Luo, and W. Chou, "Rest api design patterns for sdn northbound api," in *2014 28th international conference on advanced information networking and applications workshops*. IEEE, 2014, pp. 358–365.

[22] I. Gragopoulos, E. Papapetrou, and F.-N. Pavlidou, "Performance study of adaptive routing algorithms for leo satellite constellations under self-similar and poisson traffic," *Space communications*, vol. 16, no. 1, pp. 15–22, 2000.

[23] M. Haviv, "Queues–a course in queueing theory," *The Hebrew University, Jerusalem*, vol. 91905, 2009.

[24] V. Bonneau, B. Carle, L. Probst, and B. Pedersen, "Low-earth orbit satellites: Spectrum access," *European Commission, Directorate-General Internal Market, Industry, Entrepreneurship and SMEs*.

[25] G. Bernstein, "Basic network simulations and beyond in python introduction." [Online]. Available: https://www.grotto-networking.com/DiscreteEventPython.html

[26] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[27] F. Giannakas, C. Troussas, A. Krouska, C. Sgouropoulou, and I. Voyiatzis, "Xgboost and deep neural network comparison: The case of teams' performance," in *International Conference on Intelligent Tutoring Systems*. Springer, 2021, pp. 343–349.

[28] K. B. Abou Omar, "Xgboost and lgbm for porto seguro's kaggle challenge: A comparison," *Preprint Semester Project*, 2018.

[29] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting xgboost algorithm for prediction and classification of different datasets," *International Journal of Control Theory and Applications*, vol. 9, no. 40, 2016.

[30] D. Agnew and J. McNair, "Detection of denial-of-service attacks in a software-defined leo constellation network," *GOMACTech*, 2023.