Enhanced Network Metric Prediction for Machine Learning-based Cyber Security of a Software-Defined UAV Relay Network

Dennis Agnew Alvaro del Aguila Janise McNair
Department of Electrical and Computer Engineering, *University of Florida, Gainesville, FL*
dennisagnew@ufl.edu, delaguila.alvaro@hotmail.com, mcnair@ece.ufl.edu

***Abstract*—Unmanned aerial vehicles (UAVs), e.g., drones, have become crucial assets in the military's fleet of vehicles. UAVs can provide limited bandwidth for tactical communications and can act as relays over battlefields. Modern drones provide much higher bandwidth than their legacy predecessors with dynamic antenna capabilities that would be useful in communicating around obstacles, such as urban corridors formed by rows of tall buildings that limit terrestrial lines of sight and attenuate high frequencies. While it is still more likely that one UAV is used for this purpose, a well-managed cluster of UAVs could increase the functionality of the entire terrestrial-drone network. Software-defined networking (SDN) is recognized as an effective way to manage distributed wireless networks. Our work proposed the use of a software-defined UAV (SD-UAV) network to provide well-coordinated, secure communication resources, and relaying capabilities to on-the-ground soldiers, military vehicles, and assets in an urban, signal-challenged environment. This paper contributes a queueing analysis of the framework in order for the network operator to derive the expected theoretical values of the network under normal conditions. With this, the network operator can create a normal baseline for comparison and detection of the presence of cyberattacks. We conduct a simulation, analysis, and discussion of our results and present our findings in this paper. Our analysis is validated by our simulation results for interarrival times, transmission delay, and packet count for our network allowing, the network operator to generate predicted values during the operation of our framework.**

***Keywords*—software-defined networking (SDN), cybersecurity, UAVs, machine learning**

I. Introduction

Unmanned aerial vehicles (UAVs), also known as drones, have proven to be extremely useful for both military and commercial applications, including communication relay, surveillance, network mobilization, and disaster relief. By 2035, it is anticipated that the number of UAVs used for defense will increase by 25% to 75% percent [1]. UAVs have been employed by the Army, Navy, and Air Force as relay nodes in tactical deployments to enhance communication quality in a variety of geographic regions to improve communication quality [2], [3], an increasing concern, as shown recently in Ukraine. Although Starlink's low-earth orbit (LEO) constellation has offered reliable communications to the area, military operations may potentially be impeded by a private, proprietary service [4].

Drones and other UAVs have been the subject of recent research that has examined their capabilities [5]. Despite the ongoing advancements in UAV capabilities, they nevertheless have constraints in terms of range and operational energy, and therefore can only remain airborne for a limited duration. Traditionally, a single UAV is typically used because of its efficient control and mobility. Mechanisms for managing and controlling a network of UAVs in a dynamic and adaptable manner are required in order to deliver the network as a communications service. Software-defined networking (SDN) can be utilized to control the dynamic behavior and capabilities of UAV clusters deployed tactically [6]. SDN is a networking concept that separates the data plane and control plane of networking forwarding devices, such as switches and routers and consolidates the controller into the SDN controller(s). Compared to conventional, or legacy, networking approaches, it enables more management/oversight, visibility, and security [7].

Researchers have conducted studies on the utilization of software-defined UAVs (SD-UAVs) and satellite communication (SATCOM) to enhance combat communications through the implementation of relay networks, multi-path Transmission Control Protocol (TCP) solutions, or autonomous configurations of UAV swarms in disaster relief zones, e.g., [8]–[10]. The military is currently engaged in continuous endeavors to employ UAVs for the establishment and operation of relay networks [11], [12]. Nevertheless, the aforementioned efforts fail to take into account SD-UAV networks operating within urban settings, and they also neglect to address the issue of safeguarding against potential security threats such as jamming, black hole, and gray hole assaults perpetrated by criminal entities.

To address this concern, our previous work [13] proposed the utilization of SD-UAVs to offer efficiently coordinated and secure communication resources and relaying capabilities to soldiers, military vehicles, and assets situated in an urban setting with limited signal availability as shown in Figure 1a. Our previous work completed an analysis to assess the mobility and packet delivery for the architecture within a simulated network. The architecture design featured SD-

'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.

UAVs that functioned as non-preemptive dual-priority (NPDP) queueing forwarding devices. The utilization of NPDP queues facilitates the prioritization of priority packets, enabling them to bypass the queue of regular packets and be serviced quickly after the completion of processing of the current regular packet in the service module. This process ensures that priority packets experience reduced waiting time within the forwarding SD-UAVs' queues and are expedited through the SD-UAV network faster than regular packets. This faster behavior will allow network operators and the SDN controller to respond to cyber threats to the network much faster than ordinary M/M/1 queueing models because their prioritized messages will propagate faster through the network. Additionally, a proposed multi-cyberattack detection model was introduced to safeguard communication by mitigating jamming, black hole, and gray hole attacks with the aid of the Light Gradient Boosting (LightBGM) machine learning (ML) algorithm. Through the utilization of a labeled dataset consisting of primarily queueing theory metrics such as interarrival time (IAT), transmission delay (TD), and packet count (PC), our model was trained and afterward yielded an average detection and classification accuracy of 98% for jamming, gray hole, and black hole attacks.

However, a constraint of our previous study is the necessity of pre-labeled data of IAT, TD, and PC, as the LightGBM algorithm is a supervised learning algorithm. The ability to acquire prelabeled data for utilization in critical operational domains may not always be feasible. Moreover, in the event of a zero-day cyberattack, which refers to an attack that exploits a vulnerability unknown to the public or network operators, the machine learning model trained on prelabeled data will be inadequate as it has not been previously exposed to such an attack. One potential approach to address these issues involves the implementation of a ground truth or baseline of the network's queueing metrics of IAT, TD, and PC under non-adversarial circumstances. Conducting a queueing analysis of the system would allow the establishment of this baseline for the expected behavior, hence facilitating the development of an appropriate solution for the network operator.

In this study, we utilize an adaptation of the Jackson network open (JNO) queueing model to generate more accurate network metrics [14]. The JNO model is characterized by interconnected queues, where the output of one queue is directly linked to the input of another queue. Furthermore, the JNO model is known to possess a product form solution, which allows for efficient, comprehensible analysis and evaluation of the network performance [15].

To the best of our knowledge, there are no prior studies that have that have used JNO data to perform network metric prediction. Therefore, this study makes the following contributions:

- A study is conducted to analyze the queueing dynamics of a non-preemptive dual priority Jackson open queueing SD-UAV network designed for tactical and urban deployment scenarios.
- In this study, we use JNO as a baseline for the expected behavior of a SD-UAV controller network.
- This study uses advanced queuing models as a novel way to reduce ML training delays for zero-day cyber-attack detection.

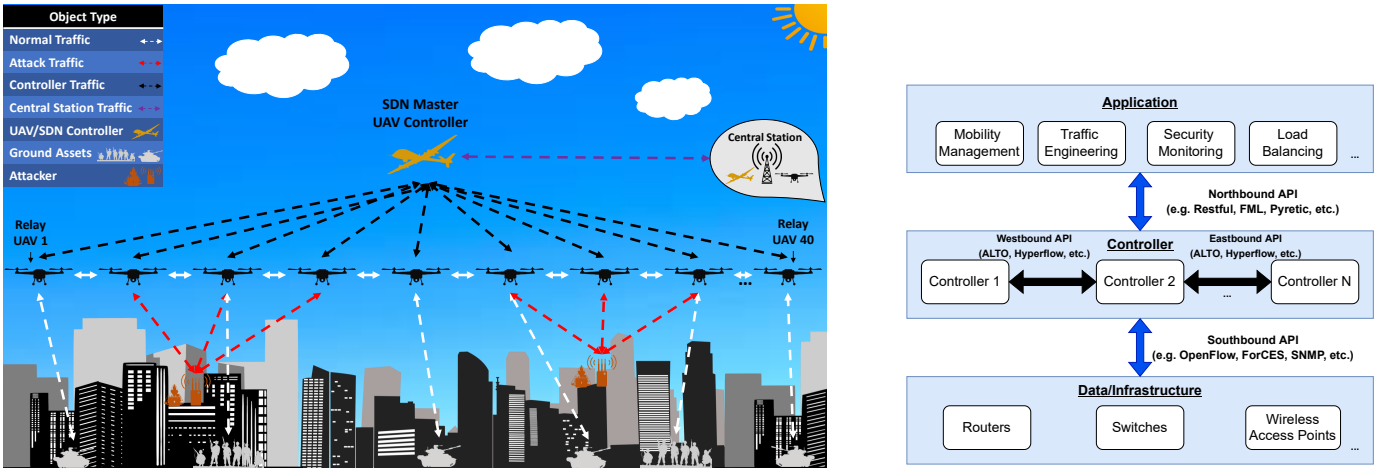
The rest of this paper is organized as follows. Section II provides a discussion of related work. Next, Section III provides a background of SDN, relay UAVs, and queueing modeling for UAVs. Section IV describes the SD-UAV network system architecture that the queueing analysis is based on and Section V describes the queueing model analysis. Section VI presents the findings of the simulation results. Lastly, Section VII concludes the paper.

II. Related Work

To study the network behavior in SD-UAV networks, we utilize queueing performance analysis. Most SD-UAV studies that employ queueing models focus on the M/M/1 queueing approach, which does not capture the traffic demands of the network architecture. Some prior work considers multiple queueing models beyond M/M/1, as discussed below. However, these studies do not involve using these models as a way to reduce ML training delays for zero-day cyber-attack detection. This section presents a discussion of the relevant literature pertaining to queueing analysis studies in the context of SD-UAVs. We analyze prior solutions, with a particular emphasis on demonstrating the distinctions between this research study and the aforementioned earlier studies.

A. Literature Review of Queueing Analysis for SD-UAV Networks

[17] concentrates on addressing the upcoming demands of sixth-generation wireless communication, specifically pertaining to high network availability, enhanced communication convergence, and intelligent features for SD-UAV wireless communication networks. It presents three hierarchical SDN controllers (1 primary, 2 secondary) to address the single-point-of-failure problems commonly associated with classic singular–controller approaches for SD-UAV networks. The study presents a load-balancing algorithm and a robust hybrid routing algorithm as potential solutions to address issues related to controller overhead and cascading failure. The researchers additionally do a mathematical model study, employing M/M/1 and M/M/c queueing analysis for their primary and secondary controllers, respectively. They used MATLAB "SimEvents" module [18] to perform their simulations. The simulation results presented in their study illustrate a significant decrease of 60%, 40%, and 25% in the packet arrival rate, service rate, and utilization factor, respectively, as compared to alternative control-domain adjustment algorithms that are currently in use. However, their modeling analysis primarily focuses on their controller framework and makes little to no mention of the data plane layer (forwarding SD-UAVs). Furthermore, their work does not consider cascaded queues in forwarding drones and how it affects the performance of the controller drone, which is prevalent in SD-UAV systems.



(a) (b)

Fig. 1: (a) Envisioned SD-UAV Network Deployment in an Urban Environment (b) General SDN Architecture [16]

[19] concentrates on the advancement of digital twin technology for SD-UAV networks through the use of performance and queueing analysis for digital twin technology. The objective is to enable real-time monitoring, analysis, and virtualization of network performance through rapid prototyping and testing. This technique incorporates a routing flow adjustment algorithm to optimize network performance. The effectiveness of this methodology is demonstrated through the use of an M/M/1 queueing model also using the aforementioned MATLAB "SimEvents" module [18]. The findings of their study indicate that the framework they proposed exhibits superior performance compared to conventional queueing models that do not incorporate SDN and Dynamic Time integration. This benefit is observed in terms of the efficiency, reliability, and agility of UAV networks. Additionally, their framework offers a network analysis platform with real-time monitoring capabilities, along with improved packet processing time at the forwarding drones and controller, and enhanced scalability. Furthermore, their work does not consider cascaded queues in the forwarding drones and their effects on the controller drones, which is prevalent in SD-UAV systems.

[20] aims to address the challenges that arise while deploying an SD-UAV network in regions where a ground base station may not be accessible, but WiFi access points (APs) are still available. Hence, the researchers are primarily concerned with the creation of a SD-UAV-mounted base station (UBS) and WiFi APs. They examine the queueing delay patterns by analyzing the UBS positioning and the AP traffic offloading. The user population can be categorized into two groups: cellular subscribers (CSs) and WiFi subscribers (WSs). A CS is linked to the UBS and may be provided simultaneous access to the APs in order to facilitate WiFi traffic offloading. By utilizing the SDN controller and its global view, the primary objective of the SDN controller is to reduce the average M/M/1 queueing delay of the CSs. This is achieved by optimizing various factors such as spectrum allocation, user equipment, base station position, CSs association with the AP, and CS traffic offloading. Additionally, the SDN controller ensures that the delay performance for the WSs is maintained within the desired parameters. The researchers conducted a queueing analysis on the traffic arrival rate, λ, for each subscriber. In this analysis, the subscribers are treated as independent parallel M/M/1 queues. The findings of their study indicate that the SDN controller effectively minimizes the average queueing delay of CSs, while also ensuring that the maximum tolerable queueing delay of WSs is maintained. This optimization is achieved by considering factors such as the allocation of licensed spectrum portions, the position of UAVs, the association of CSs to WiFi, and the rate at which CS traffic is offloaded to the WiFi access point. Although their work considers delay optimization for M/M/1 queues, the network traffic models are limited to M/M/1 approach and do not consider predicting network behavior for security purposes. Furthermore, their M/M/1 approach omits M/M/C analysis which is more representative of multiple end-users in tactical network operations.

[21] aims to develop a mission-critical software-defined wireless sensor network (MC-SDWSN) that can effectively collect information in diverse and intricate situations. Additionally, the network aims to facilitate numerous mission-essential applications, including industrial automation and security surveillance. The proposed MC-SDWSN seeks to address the challenges associated with conventional Wireless Sensor Networks (WSNs), including resource usage, data processing, system compatibility, and stringent latency requirements. The design employed in their system incorporates SDN topologies, which integrate the hierarchical cloud and edge computing technologies. A unique centralized computing offload strategy is proposed in this study, based on the MC-SDWSN architecture, to demonstrate its feasibility in sensor network applications. The researchers performed a queueing study on their architecture, using the forwarding devices as M/M/1 queues. The simulation findings illustrate the enhanced offloading percentage of periodic computing tasks and latency

of various computing systems, as well as a superior delay guarantee compared to previous approaches. Furthermore, the forwarding agents and controller employed in their system are hardware switches as opposed to SD-UAVs, and their queueing analysis lacks the incorporation of JNO analysis that would be used for predictive network behavior for security purposes.

B. Summary

The aforementioned studies illustrate a subset of the advancements made in analyzing queueing systems for SD-UAV networks. The majority of research focuses on the analysis of standard M/M/1 queueing systems while neglecting the investigation of using queueing analysis for enhanced network metric prediction for ML-based cyber security of SD-UAVs. Furthermore, present research focuses primarily on the development of queueing analysis for the controller layer of SD-UAVs, with little regard for the forwarding layer queueing analysis and its effects on the controller's queueing performance.

III. Background

SDN has gained traction for UAV networking research and has begun to shift its networking management paradigm. SD-UAV networks exhibit distinct traits and characteristics in contrast to conventionally managed UAV networks. This section discusses the background of SDN for SD-UAVs and the application of queueing theory for SD-UAV security.

A. Software Defined Networking for SD-UAVs

SDN is a network management framework that facilitates user-driven control over the forwarding process in network nodes. The development of SDN spanned multiple decades and was ultimately achieved by a team of researchers at Stanford University [22]–[24]. SDN has the following characteristics, illustrated in figure 1b:

- The control plane and data plane are distinct entities that are decoupled from each other.
- The controller functions as the primary decision-making and coordinating entity, both internally and externally. The primary function of this entity is to facilitate the routing of traffic within the network and ensure the overall stability and operational state of the network.
- The determination of forwarding decisions depends upon flow policies rather than the ultimate destination. A flow can be defined as a standardized sequence of instructions that controls the transmission and reception of data packets between a specific source and destination. SDN controllers facilitate the establishment of flow tables through the implementation of policies. The implementation of flow tables is carried out through forwarding devices.
- The network possesses the capability to undergo programming via software programs that operate on the SDN controller.
- Application programming interfaces (APIs) are utilized for the purpose of transmitting data between different layers of the SDN infrastructure.

As depicted in Figure 1b, the infrastructure layer comprises routers, switches, and access points. This layer represents the network's actual network equipment (e.g. relay/forwarding SD-UAVs), and it creates the data plane. The controller establishes communication with the data plane by transmitting instructions to the switches and routers (relay/forwarding SD-UAVs). This is achieved through the utilization of southbound programming interfaces, commonly referred to as Southbound APIs: OpenFlow [25], ForCES [26], PCEP [27], NetConf [28], or I2RS [29]. The SDN controller UAV would monitor the forwarding SD-UAVs and use southbound APIs to forward the instructions. In the scenario where multiple controller UAVs are present, intercommunication between these controllers is facilitated through the utilization of Eastbound and Westbound APIs, commonly referred to as East/West APIs: ALTO [30] or Hyperflow [31].

This feature enables the controllers to effectively manage and monitor the network at a global scale. The highest layer is referred to as the application plane. Within this layer, the network operator at the central station possesses the capability to establish network policies, contingent upon functional applications, to address diverse tasks including energy efficiency, access control, mobility management, and/or security management. The application layer facilitates the transmission of policies to the network by means of the control layer, employing Northbound APIs as a means of communication such as FML [32], Procera [33], Frenetic [34], and RESTful [30]. The network operator has the ability to transmit the required modifications to the controller SD-UAV through the use of these APIs, enabling the controller to implement the necessary adjustments within the infrastructure layer, depending on the desired outcomes.

In contrast with SDN, the forwarding agents of conventional networks control their own forwarding logic. To alter the network, the forwarding devices individually need to be reconfigured. Due to the presence of these obstacles, network management rules in conventional networks exhibit limited dynamism and pose challenges in terms of scalability for SD-UAV networks. SDN provides the network operator with the capability to swiftly modify data flows of forwarding SD-UAVs, hence facilitating the adaptation to fluctuating traffic demands and security threats.

B. Cyberattacks

The primary objective of attackers is to intentionally disrupt UAV networks in order to achieve personal benefits. Research on UAVs has explored a range of potential hazards posed by cyberattacks targeting UAV networks. Network operators must remain vigilant in order to mitigate the potential impact of various cyberattacks on their SD-UAV network. Zhenhua et a. [35] define the threats to UAVs network security as follows:

- **Injection Attacks:** Injection attacks can be classified as integrity attacks, wherein the perpetrators manipulate the sensitive information of UAVs by introducing false data. This malicious act aims to deceive by causing incorrect

data to be present in data streams, which subsequently are transmitted to the central station.

- **Fabrication Attacks:** Fabrication attacks are categorized as a subset of integrity attacks, frequently employed in conjunction with other attack methods to target UAVs. The attack procedure involves the utilization of a misleading identity by an attacker in order to acquire sensitive information pertaining to the UAVs and, subsequently, the attacker proceeds to transmit inaccurate data back to the central station.
- **Denial-of-Service (DoS) Attacks:** The objective of a DoS attack is to impede the connectivity of a UAV by flooding it with a substantial volume of fabricated packets, resulting in the depletion of the UAV's internal resources.
- **Jamming Attacks:** Jamming attacks are deliberate attempts to disrupt wireless communication connections and communication signals using technical methods. By transmitting an identical signal frequency, one may create interference in the UAV and interrupt its communication with the central station.
- **Network Eavesdropping Attacks:** Network eavesdropping attacks can be classified as passive attacks. Typically, perpetrators employ malicious devices with the intention of intercepting the communication between UAVs and air traffic control, thereby gaining insight into the aerial environment and detecting the presence of UAVs.
- **Man-in-the-middle Attacks:** In the context of a man-in-the-middle attack, the attacker strategically positions themselves between two UAVs engaged in communication. By impersonating either of the UAVs to each other, the attacker manipulates the flow of information, forcing the two UAVs to inadvertently transmit data to the attacker before it reaches its intended destination.
- **Replay Attacks:** Replay attacks compromise the integrity of UAVS. The perpetrator employs network surveillance or similar techniques to covertly acquire authentication credentials, which are subsequently transmitted to a recipient with the intention of misleading UAVs.
- **Worm hole and Black Hole Attacks:** Two adversarial UAVs, situated in distinct geographical positions, alter their flight paths in order to intercept and acquire communications from an alternative channel. Subsequently, they transmit these signals to an additional malevolent node over a specialized communication channel. When UAVs are subjected to attacks, a malevolent node intercepts a packet containing the UAV's position and afterward transmits it to another remote malevolent node via a tunnel. As a result, the manipulated data packet is then forwarded to neighboring nodes.
- **Sybil Attacks:** Sybil attacks encompass the act of transmitting messages through many peer identities in order to control UAV networks. Sybil attacks encompass three distinct dimensions, namely communication, participation, and identification. The communication component encompasses attacks that establish a connection between Sybil nodes and legitimate nodes within networks. In order to disrupt network operations, an attacker engages in communication with authentic nodes via established connections. In the context of participatory dimension attacks, attackers employ two distinct methods to introduce Sybil nodes into the network, either concurrently or sequentially, with the objective of gradually acquiring control. Identity dimension attacks refer to the malicious act of spoofing Sybil node identities. There are two distinct methodologies that attackers can employ in order to assume an individual's identity. The act of identity theft can be perpetrated by an attacker targeting a legitimate node that is either offline or has completely depleted its battery. An alternative approach involves the creation of fictitious identities that are not present within the network.
- **Hijacking Attacks:** UAVs depend on the visual data obtained by a camera in order to perform target tracking and avoid collisions. The procedure entails flight controllers making a request for the acquired videos from the kernel of the computer operating system through the issuance of a system call. In the event that an individual with malicious intent possesses knowledge of the system parameters and gains unauthorized access to the flight controller, it is possible for this individual to initiate a hijacking attack on the system call. This attack involves substituting a genuine camera with a virtual camera and relocating the UAV beyond its intended destination, therefore intentionally hijacking it.

C. Queueing Modeling for SD-UAVs Cybersecurity

Network monitoring can be used to detect cyberattacks; however, due to the random nature of network traffic, data collection, and training can be time-consuming [36], [37]. This paper proposes that the NPDP JNO queueing model is sufficient for training and reducing the associated delays. The application of queueing theory (QT) enables the mathematical evaluation and formulation of models for SD-UAV networks [17]. QT is a branch of mathematics that focuses on the analysis and modeling of waiting lines or queues. From a networking perspective, forwarding devices can be classified as servers, while packets can be referred to as customers. Network operators can derive several performance measures from a network, such as interarrival time (IAT), transmission delay (TD), and packet count of packets received at a server, by assessing the queueing of packets at the forwarding device. For SD-UAV networking, the UAVs are interpreted as servers, while the packets they handle can be regarded as customers. The network status may be inferred by analyzing the observed average arrival rate (λ) and service rate (μ) of packets, as well as the aforementioned metrics. By utilizing these metrics, the detection of cyberattacks within a SD-UAV network can be achieved by training a ML model on non-attack (normal) and attack network samples. As seen in Figure 2, the consequences of a DoS attack on a server are depicted. The victim node experiences a high influx of packets, resulting in a cascading impact on the system. This leads to an elevated arrival rate

(λ_1) at the victim node, as well as a reduction in both the service rate (μ_1) at the victim node and the arrival rate (λ_2) at the destination node.

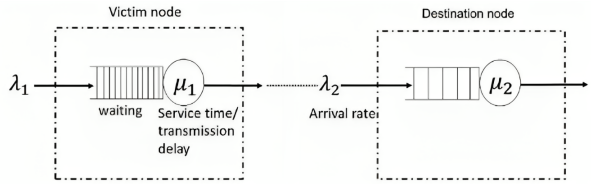


Fig. 2: Victim Representation of a Cyberattack [38]

The aforementioned metrics can be captured by the forwarding UAVs and subsequently transmitted to the controller for onward transmission to the network operator, who will then proceed to conduct further analysis. Based on the measurements obtained, the network operator can infer the type of attack that has occurred and assess the extent of its impact on the network. Unique to SDN, the operator has the ability to utilize the SDN controller in order to redirect network traffic away from a compromised UAV. This allows for the maintenance of network connectivity and throughput within the network. In order to make informed decisions based on QT analysis, it is crucial that the accuracy of the QT measurements is maximized to ensure appropriate actions are taken. Hence, it is imperative to take into account the design of the QT modeling. QT utilizes Markovian queues. Markovian queues are distinguished by their adherence to the Poisson process for arrival rates and exponential distribution for service rates, hence facilitating memoryless arrival and service rates. There exists a variety of Markovian models, including but not limited to M/M/1, M/M/c, M/M/c/K, M/G/1, and M/M/∞ [39]. Hence, it is essential for the network operator to consider the queueing dynamics of the SD-UAVs within their network, as it significantly impacts the metrics they will obtain and afterward utilize for the purpose of cyberattack detection.

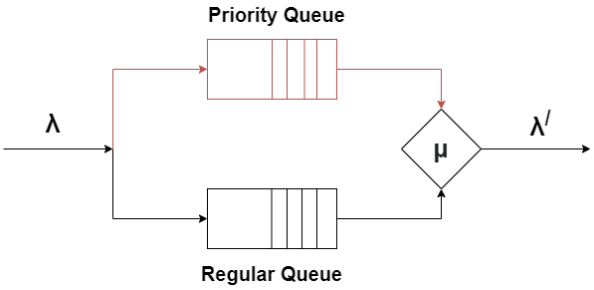


Fig. 3: Non-Preemptive Dual Priority M/M/1 Queue

The suggested model, as depicted in Figure 1a, presents the SD-UAVs as a modified Jackson open network of non-preemptive dual priority (NPDP) M/M/1 queues. According to Chen et.al, [14], Jackson networks are often based on the concept that numerous queues are interconnected, with the input of one queue being forwarded to another with all traffic having equal priority, and the network has a product form solution. However, our model adheres to these identical assumptions, with the exception that not all packets have the same priority and that the model incorporates two distinct packet types: priority and regular. As depicted in Figure 3, priority and regular packets enter the UAV as λ and then queue into different queues according to their labeling. Then they are serviced at the service module, μ, then exit the UAV as $\lambda^{/}$. Priority packets are processed in a non-preemptive, faster manner over regular packets at the service module. This implies that if a packet arrives in the priority queue while the server is currently processing a regular packet, the server will complete the processing of the regular packet before attending to all the priority packets. Only after servicing all the priority packets will the server resume processing the regular packets which is discussed further in section V.

IV. SD-UAV Network System Architecture

A. Overview

The architecture depicted in Figure 1(a) comprises 40 relay drones, 1 master controller/drone, and 1 central station (CS) situated in an urban battlefield. The network operator has the capability to monitor the network and deploy supplementary drones at the central station. Each drone functions as NPDP JNO queue. The packets that are received at a node that originates from the master controller are equipped with a priority tag in its packet header and are placed in a distinct queue that possesses a higher priority compared to the regular packet queue. In addition, emergency signals or messages from the leaders on the ground can also utilize the priority tags to send their messages faster in the network. The prioritization of command traffic from the controller or CS will facilitate expedited network reconfiguration and enhance the speed of reaction to threats as those commands are propagated through the network faster than regular traffic.

As depicted in Figure 1(a), the aerial drones are positioned above the structures in order to serve as relay nodes for transmitting data from ground assets. The master controller is responsible for supervising the forwarding drones and implementing flow installations in the forwarding UAVs to effectively direct traffic toward the desired destination inside the cluster. If the master controller experiences a failure, the relay drones will transition to legacy ad-hoc routing until a new master controller can be deployed from the CS.

The role of the CS involves the supervision of network traffic and the application of ML techniques to identify potential cyber threats within the network. Upon detection of an attack, the CS establishes communication with the master controller, directing it to issue mitigation commands to the relay drones in the form of isolating compromise SD-UAVs allowing for subsequent investigation of these devices before readmittance into the network. In order to achieve this objective, the initial step involves the retrieval of network statistics from the relay drones by the master controller. Subsequently, this data is transmitted to the CS for the purpose of processing and labeling for future training of models.

Supervised machine learning models are rendered ineffective in scenarios involving zero-day attacks or the absence of prelabeled data, as aforementioned in section I. This is due to the unavailability of labeled instances of the attack and inadequate data for model training purposes. The mathematical analysis presented in this research can be employed by network operators to compare the existing network operation metrics with the anticipated metrics derived from the analysis provided in this study. Subsequently, the network operator will possess the ability to implement proactive defensive measures in order to safeguard the integrity of the network against potential cyber threats.

B. Controller Functionality

The SDN controller is constructed using various applications that operate autonomously and harmoniously. Preexisting applications are essential for the functionality of the controller. The OpenFlow protocol is an essential application for the controller since it enables access to the forwarding plane of network switches or routers. This access is necessary for the controller to compute network pathways.

The subsequent requisite application entails the implementation of a forwarding app, which serves the purpose of determining the optimal path selection and calculating the corresponding paths. In a static network, it is advantageous to proactively compute and select paths, hence establishing the optimal path options for each host prior to the transmission of packages. In the current operational context involving mobile UAVs, there is a pressing requirement for a responsive approach to dynamically compute pathways as per immediate demands. The "Reactive Forwarding app" is proposed as an optimal forwarding application for the controller. It operates by computing and selecting the shortest available path, in terms of hop count, for transmitting a package from a host. The applications are developed using the Java programming language, and they possess the capability to be readily customized in order to adapt their functionality to suit the requirements of the controller and network. There are different ways to set up a controller:

- Cloud based SDN controllers, which can be accessed, monitored, modified and controlled from anywhere with a connection in real-time, with the dependency on the connectivity, losing functionality when there is no connection.
- Hardware installed controllers, installing its applications and modifications manually, monitoring the network in person, and not dependent on its connectivity to an external source.

Our architecture employs a hardware master controller as shown in Figure 1a. The master controller is responsible for maintaining a comprehensive network perspective for the forwarding drones and implements flow installations in the forwarding UAVs to effectively direct traffic towards the desired destination inside the cluster. If the master controller experiences a failure, the relay drones will transition to legacy ad-hoc routing until a new master controller can be deployed from the control station. The master controller in this architecture uses the Open Network Operating System (ONOS) controller. As described in our prior work [5], our customized ONOS controller employs reactive forwarding, the shortest path algorithm, and code modification techniques to dynamically establish the most efficient paths between hosts as needed. Furthermore, the routing algorithm prioritizes paths with reduced energy consumption in order to conserve the batteries of the drones. Additionally, it selects routes that circumvent network bottlenecks.

V. Queueing Model Analysis

A. Categorization of Architecture Components

In the proposed SD-UAV network, the SD-UAVs consist of the controller and the forwarding UAVs. The controller drone is tasked with managing the SD-UAV network by controlling the movement of packets in the network using installed flow rules in the forwarding drones. The controller drone has a global view of the network and can make forwarding decisions based on the availability of links. The forwarding drones are responses for receiving and moving packets based on the instructions of the controller drone. In a tactical operation, the drones are tasked with maintaining a dynamic nature of communication and configuration needs since the network may need to adapt in response to ground asset activity.

Figure 4 presents the queueing model of the different types of SD-UAV in our model. Each UAV and the central station can be generalized to the following labels:

- **SD-UAV$_{\mathbf{A}}$:** It is considered the first forwarding drone the packet encounters in route to the final destination. If there are no matching entries for the packet, this drone sends it the controller as a packet-in message for it to analyze and give the SD-UAV$_{\mathrm{A}}$ drone additional to flow rules where to send the packet. If there is an entry, the SD-UAV$_{\mathrm{A}}$ drone sends it to the next hop(s) (SD-UAV$_{\mathrm{A+j}}$).
- **SD-UAV$_{\mathbf{A+j}}$:** This is the next logical hop(s) after initial reception of the message by SD-UAV$_A$. Similarly, if there is no matching rule for the packet, each drone forwards it to the controller. If there is a matching rule, it forwards it to the next hop, j, or the final destination shown as $q_{EX}\lambda_{A+j}$.
- **SD-UAV$_{\mathbf{C}}$:** This is the SDN controller of the network. When receiving packets in the form of packet-in messages from the forwarding drones, the SD-UAV$_C$ drone is tasked with leveraging its global view of the network to provide instructions flow rules to the forwarding drones on where to send the packet next. The controller is also responsible for providing movement and reconfiguration commands to the forwarding drones based on changes in the environment or commands from the central station.
- **Central Station (CS):** The central station (CS) is where the network operator will be located. They will be responsible for pulling network performance statistics from the controller and leveraging ML techniques to detect the presence of a cyberattack. The CS will then

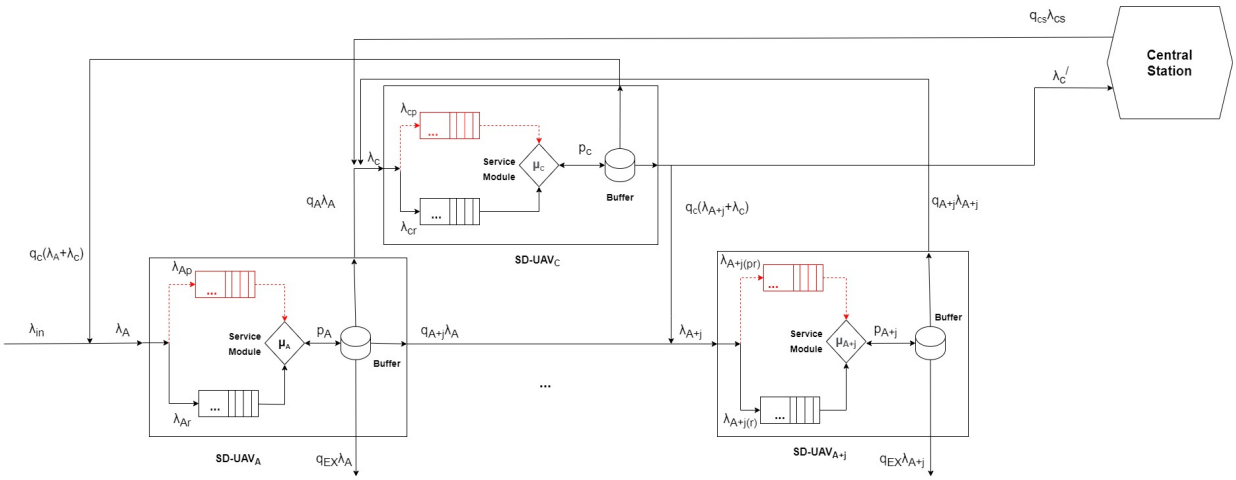


Fig. 4: NPDP Open Network Jackson Model of SD-UAV

communicate the alert of an attack to the controller to take the necessary actions for the network such as re-configuring the topology.

B. Statistical Analysis

In this section, we present our statistical analysis of our architecture. For the analysis, we have the following assumptions:

- We assume the SD-UAVs are in a steady state, and not transient. In a steady state, the arrival rate of packets λ is less than the service rate μ. Therefore, $\frac{\lambda}{\mu} < 1$, and $0 < \lambda < \mu$.
- There are only two priorities: priority and regular. Regardless of priority, the service module on the SD-UAV will service each packet with the same service rate, μ. However, priority packets are serviced first non-preemptively over regular packets.
- Since the network is assumed to be in a steady state, λ_{in} to a UAV is equal to λ_{out} of SD-UAV according to Burke's theorem [40].

As mentioned in section I, we perform an NPDP JNO analysis which means that our equations have a product-form solution. Let the subscripts pr and r denote priority and regular, respectively, and let service rate by the service module we denote as μ. Subscript A, denote the first UAV in the connection (SD-UAV$_\text{A}$) and $A+j$ denotes the UAV(s) that are/is j hops away from SD-UAV$_\text{A}$ (SD-UAV$_\text{A+j}$). Let subscripts CS and C denote the Central Station, and the SDN controller, respectively.

The arrival rate of the packets that initially enter the network from the ground asset is denoted as λ_{in}, where $\lambda_{in} = \lambda_{pr} + \lambda_r$. The possibility that a packet is forwarded to the next hop is denoted as q_i, where i is the next hop. Therefore, λ_i for each drone class can be denoted as follows:

$$\lambda_A = \lambda_{in} + q_c(\lambda_A) \tag{1}$$

$$\lambda_{A+j} = q_{A+j}\lambda_A + q_c(\lambda_{A+j}) \tag{2}$$

$$\lambda_c = q_A\lambda_A + q_{A+j}\lambda_{A+j} + q_{cs}\lambda_{cs} \tag{3}$$

where in equations (1) and (2), SD-UAV$_A$ and SD-UAV$_{A+j}$ will receive reply traffic from the controller that originally originated from each respective UAV. The probability of such traffic is denoted by q_i. The probability density function for λ_i is defined for $t \geq 0$:

$$f(t) = \lambda_i e^{\lambda_i t} \tag{4}$$

The average interarrival time (IAT) of packets experienced at SD-UAV is defined:

$$\overline{IAT_i} = \frac{1}{\lambda_i} \tag{5}$$

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

$$g(s) = \mu_i e^{-\mu_i s}, \forall \geq 0 \tag{6}$$

where the average service time, $\overline{T}_{st}$, of a SD-UAV can be denoted:

$$\overline{T_{st}} = \frac{1}{\mu_i} \tag{7}$$

Let p^i_{int} represent the traffic intensity of packets arriving at SD-UAV$_i$:

$$p^i_{int} = \frac{\lambda_i}{\mu_i} \tag{8}$$

Utilizing Little's law [41], the total waiting time is defined as transmission delay (TD) for both priority and regular classes by k:

$$W_k = TD_k = \frac{L_k}{\lambda_k} \tag{9}$$

where L_k is the average number of packets in the SD-UAV$_i$. It is defined as follows [39]:

$$L_k = L_k^{que} + p_k \tag{10}$$

where $p_k = \frac{\lambda_k}{\mu}$ and L_k^{que} denote the average number of packets for each class in the queues. They are represented in equations (11) and (12) and the average total number for both classes is defined in equation 13 [39]:

$$L_{pr}^{que} = \frac{\lambda_{pr} p}{\mu - \lambda_{pr}} \tag{11}$$

$$L_r^{que} = \frac{\lambda_r p}{(\mu - \lambda_{pr})(1-p)} \tag{12}$$

$$L_{pr+r}^{que} = \frac{p^2}{1-p} \tag{13}$$

where $p = \frac{\lambda_{pr}}{\mu} + \frac{\lambda_r}{\mu}$. Therefore, regular packets wait in a queue of UAVs longer than priority customers when $p < 1$ as shown in equation 14 [39]:

$$W_r^{que} = \frac{p}{(\mu - \lambda_{pr})(1-p)} = \frac{p/(\mu - \lambda_{pr})}{1-p} = \frac{W_{pr}^{que}}{1-p} > W_{pr}^{que} \tag{14}$$

From Little's law, the average total waiting time, or transmission delay ($\overline{TD}_k$) of SD-UAV$_i$ can be derived from equations (8),(9), and (10):

$$\overline{TD}_{pr} = \frac{\lambda_r + \mu}{\mu(\mu - \lambda_{pr})} \tag{15}$$

$$\overline{TD}_r = \frac{\mu^2 + \lambda_r \lambda_{pr} + \lambda_{pr}^2 - \lambda_{pr}\mu}{\mu(\mu - \lambda_{pr})(\mu - \lambda_r - \lambda_{pr})} \tag{16}$$

Let T denote the time period of observation. The packet-count (PC) for received or sent packets during steady state can both be modeled as follows for SD-UAV class A and $A + j$:

$$PC_i = \lambda_i T_i \tag{17}$$

For class c, the SDN controller, the PC can be modeled as follows:

$$PC_{c_i} = T(q_A \lambda_i + j(q_i \lambda_i)) \tag{18}$$

where j represents the number of hops necessary to reach the final destination.

VI. Simulation Results

The aim of this paper is to decrease the data collection and training time necessary for network security of a SD-UAV network deployed in an urban environment. The network traffic of the SD-UAV was evaluated by utilizing SimComponents [42], a network traffic simulation software developed based on the SimPY process-based discrete event simulation framework. The open source code of SimComponents was altered to include a non-preemptive priority M/M/C queueing model, as detailed in Section V. The simulation was designed based on the UHF/VHF radio communication commonly employed in military operations. The adjustment of port rate and queue size parameters of the servers, as presented in Table I, allowed for the attainment of packet inter-arrival times (IAT), transmission delay (TD), and packet count (PC) of received packets. These simulation parameters were derived from the military UAV wireless communication standards STANAG 4586, MIL-STD-6016, and relevant research [17], [19], [21].

In order to maintain the integrity of our analysis, we chose the minimum subset of our network that included all four SD-UAV classes, as described in Section V, such as the initial forwarding drone, next logical hop drone, controller drone and central station. This subset consisted of two drones designated for forwarding (SD-UAV$_A$ and SD-UAV$_{A+j}$), one controller (SD-UAV$_c$), and one central station. As demonstrated in Algorithm 1 and Table I, the packet arrival rates λ_{in} and λ_{cs} are first set to 100 packets/sec each. The simulation is then executed for a duration of 60 seconds. The metrics $\overline{IAT}$, $\overline{TD}$, and $\overline{PC}$ values are recorded. Next, the values of λ_{in} and λ_{cs} are increased by 100 packets/sec until λ_{in} and λ_{cs} reach rates of 1000 packets/sec. As aforementioned, we assume and simulate that the network has reached a steady state, therefore $0 < \lambda < \mu$ for all SD-UAVs within the network.

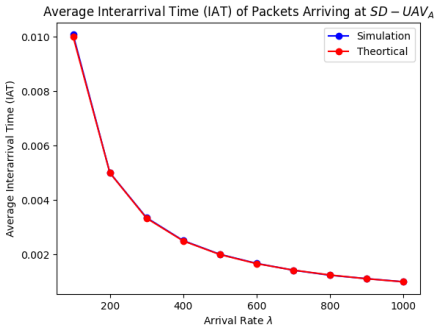
Our analysis makes the assumption that the next hop (SD-UAV$_i$) is ready to receive a transmission once the current SD-UAV has received all necessary fragments of the message and does not consider the time necessary for the next j hop to be ready for message reception, t_j. We complete the simulation on a desktop running Microsoft Windows 11 with an Intel® 12th Gen Core™ i7-12700K CPU @ 3.6GHz with 16GB of RAM. The subsequent section will provide the outcomes of the simulation for each metric and offer an analysis of the findings.

Algorithm 1 SD-UAVs Data Collection

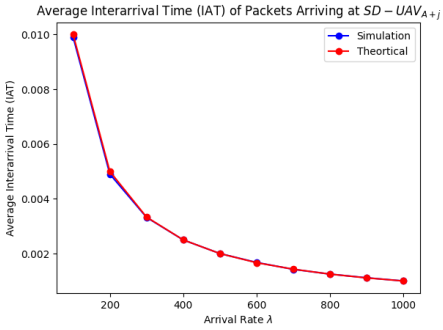
1: Create arrays for the IAT, TD, and PC average values
2: **for** λ_{in} and $\lambda_{cs} \leq 1000$ packets/sec **do**
3: Simulate SD-UAVs communication for 60 seconds for current λ_{in} and λ_{cs} and parameters
4: Record IAT, TD, and PC over duration of simulation
5: Calculate and Append $\overline{IAT}$, $\overline{TD}$, and $\overline{PC}$
6: Increment λ_{in} and λ_{cs} values by +100 packets/sec each
7: **end for**

TABLE I: Simulation Parameters

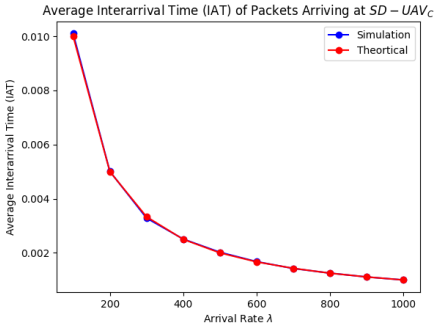
Parameter	Value
Probability packet will be sent to the controller from forwarding SD-UAVs, q_i	0.05
Probability packet will be sent to next j hop from SD-UAV$_A$, q_{A+j}	0.95
Probability packet will be sent from the central station to the controller, q_{cs}	0.05
Probability controller will send packets to the central station	0.90
Probability of a priority packet for λ_i	0.35
Probability of a regular packets for λ_i	0.65
Average packet size	1400 Bytes
Average SD-UAV$_i$ service rate, μ_i	1100 packets/sec
Average arrival rate at SD-UAV$_i$ and controller	$a_1 = 100$ packets/sec $a_{n+1} = a_n + 100$ packets/sec $a_{n+1} \leq 1000$ packets/sec
Number of forwarding UAVs in the data plane	2
Number of controllers	1
Number of central stations	1
Simulation Time for each λ_{in} and λ_{cs}	60 seconds



(a)



(b)



(c)

Fig. 5: Average interarrival times over increasing λ_{in} and λ_{cs}

A. Interarrival Time Performance Analysis

A network's performance is greatly affected by the arrival rate of received packets, λ. If $\lambda \geq \mu > 0$, then the network is not considered in a steady state, and the service module will not be able to service packets quickly enough to empty the queues of incoming packets, resulting in packet drop. Malicious entities may attempt to deliberately produce this behavior by injecting malicious packets at a high rate, λ_{attack}. Hence, it is imperative for the network operator to be aware of the incoming packet arrival rate encountered by an SD-UAV. The measurement of $\overline{IAT}$ packets by the network operator is a suitable approach, as these values exhibit an inverse relationship with the parameter λ. The magnitude of the IAT values indicates the rate at which packets are being received by an SD-UAV. Fluctuations in $\overline{IAT}$ readings may indicate the emergence of a potential cyberattack as packets are arriving faster than anticipated to the SD-UAV. The $\overline{IAT}$ experienced by an SD-UAV can be calculated based on equation (5). To calculate the expected $\overline{IAT}$ experienced at an SD-UAV in the proposed architecture, the network operator can use the expected arrival rate, λ, experienced at each SD-UAV which we have derived in equations (1), (2) and (3).

To validate our analysis, we perform the aforementioned simulation and compare our theoretical results from our analysis in Section V with the simulation results in as shown in Figure 5. We provide results for $\overline{IAT}$ for SD-UAV$_A$, SD-UAV$_{A+j}$, and SD-UAV$_c$ for Figures 5(a)(b)(c), respectively. As the λ increases, the average $\overline{IAT}$ decreases as packets naturally flow faster into the SD-UAV network at each UAV. The simulation matches the theoretical values very precisely validating our equations.

B. Transmission Delay Performance Analysis

Transmission delay can be defined as the amount of time it takes for a packet to be serviced at the service module of a packet and sent to the next j hop. The magnitude or fluctuation of the $\overline{TD}$ values of an SD-UAV could indicate the presence of a cyberattack as shown in Figure 2. Utilizing little's law, the $\overline{TD}$ can be derived from the number of items in the system, L and the arrival rate, λ. To validate our analysis, we complete the aforementioned SD-UAVs simulation and record the $\overline{TD}$ of packets as they leave the SD-UAV for each increasing λ. Acting as a NPDP JNO network, each packet in the simulation receives a label of priority or regular during creation and transmission and is queued and transmitted as such through the network in accordance to its label.

Figure 6(a) and Figure 6(d) represent priority and regular packets TD at SD-UAV$_A$, respectively; Figure 6(b) and Figure 6(e) represent priority and regular packets TD at SD-UAV$_{A+j}$, respectively; and Figure 6(c) and Figure 6(f) represent priority and regular packets TD at SD-UAV$_c$, respectively. For each figure, the simulation and theoretical values closely follow each other. Notably, there exists a fluctuation in the simulation value, wherein it alternates between slowing and mimicking the theoretical value. This phenomenon arises as a result of natural functions of the device processing speed of the simulation is tested on. However, each simulation value closely follows its theoretical value. Overall, all the figures demonstrate a positive correlation between the $\overline{TD}$ values and the increasing λ of incoming packets. This relationship occurs due to the queues of the SD-UAVs being more congested, leading to longer service times for the incoming traffic. Also notably, for Figures 6 (a)(b)(c) of the priority packets have significantly less exponential curvature compared to Figures (d)(e)(f) of the regular packets at SD-UAV$_i$. Although these figures are still exponential, they have less curvature compared to the regular figures because these packets spend significantly less time in the queue, W_k^{que}, as the λ_{in} and λ_{cs} increases than regular packets due to them being served non-preemptively as

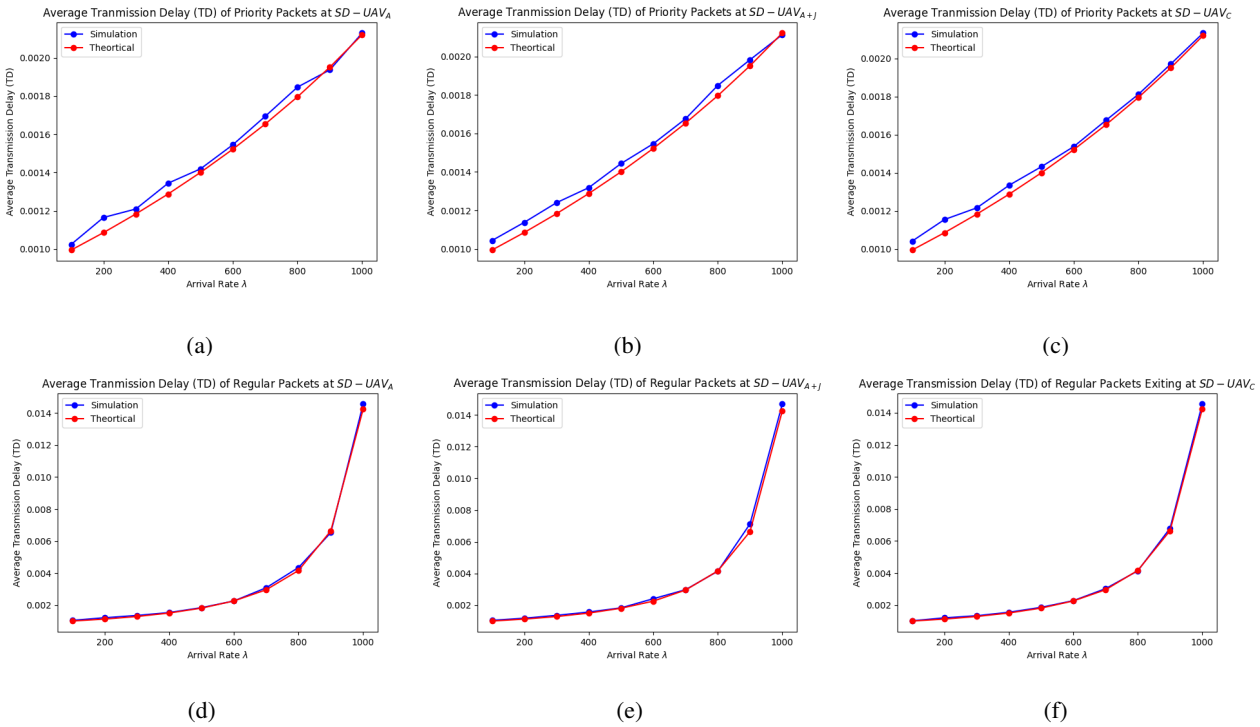


Fig. 6: Average transmission delay over increasing λ_{in} and λ_{cs}

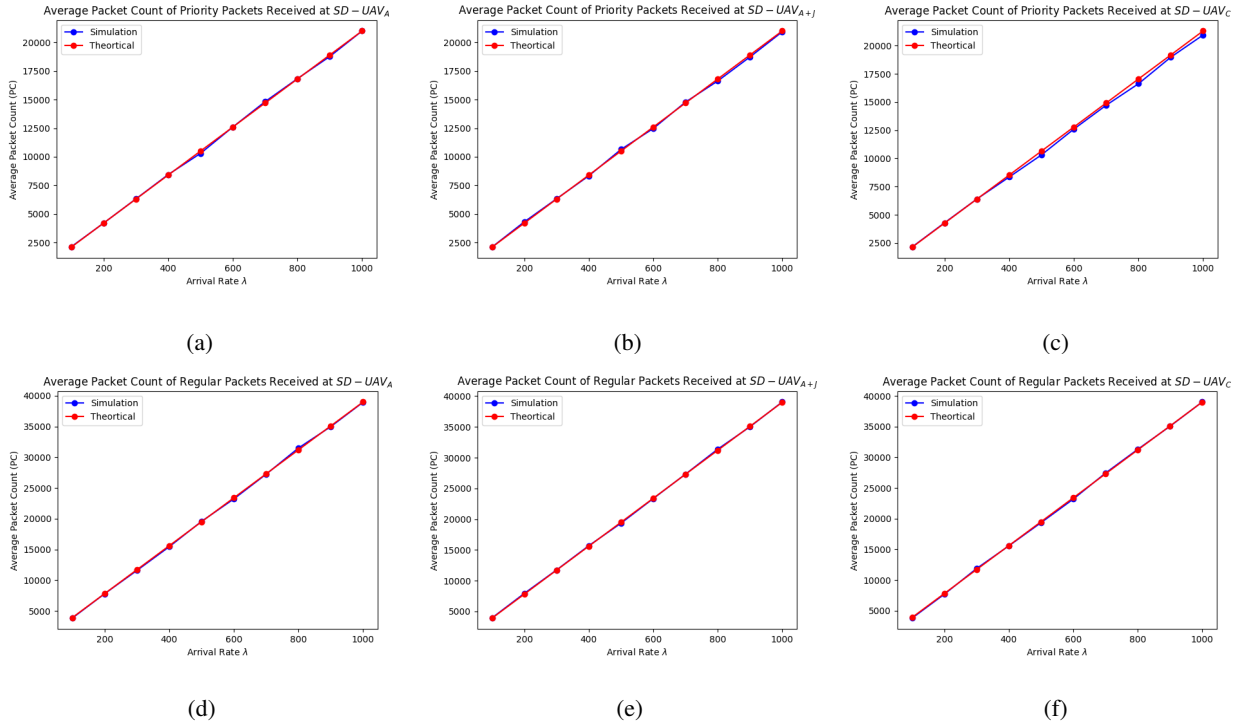


Fig. 7: Average packet count over increasing λ_{in} and λ_{cs}

well as there being naturally fewer priority packets to fill the queues since the probability of a priority packet occurrence is 35%. However, if $\lambda_{pr} = 1.0$ and $\lambda_r = 0.0$, the NPDP queue would act as standard M/M/1 queue and the $\overline{TD}$ values would exhibit a more exponential curvature as the priority packets would spend more time in the queue waiting to be serviced

as each packet has wait for the priority packet in front of it to complete. Overall, Figure 6 validates our analysis.

C. Packet Count Performance Analysis

The average amount of packets received at SD-UAV can denoted as $\overline{PC}$. Abnormal $\overline{PC}$ can indicate normal or malicious activity is occurring within the network. Malicious actors can flood an SD-UAV or cause its neighbors to drop packets to affect the behavior of an SD-UAV. Therefore, it is imperative that the network operator be able to determine the expected $\overline{PC}$ values that should be received at each SD-UAV. From our analysis, the expected packet count of packet received is demonstrated in equations (17)(18).

To validate our analysis, we complete the aforementioned simulation for each λ_{in} and λ_{cs} value and record the packet count of packets received at each SD-UAV$_i$. The results of our simulation can be seen in Figure 7. Figure 7(a) and Figure 7(d) represent priority and regular packets $\overline{PC}$ at SD-UAV$_A$, respectively; Figure 7(b) and Figure 7(e) represent priority and regular packets $\overline{PC}$ at SD-UAV$_{A+j}$, respectively; and Figure 7(c) and Figure 7(f) represent priority and regular packets $\overline{PC}$ at SD-UAV$_c$, respectively. For each figure, the simulation and theoretical values closely follow each other. In Figure 7(c), there are some slightly notable separation of the simulation and theoretical values as λ_{in} and λ_{cs} continues to increase which is due to system functions during simulation that is slightly present in the other sub figures as well. Furthermore, the figures do not exhibit exponential growth like the other figures due to the linear relationship shown in equations (17)(18). Overall, the simulation values closely follow the theoretical values and validate our analysis.

VII. Conclusion

This study presents an enhanced network metric prediction for machine learning-based cybersecurity of a SD-UAV relay network. In our recent study [13], we presented findings that highlighted the efficacy of collecting queueing data, including interarrival times (IAT), transmission delay (TD), and packet count (PC) queueing performance metrics, for the purpose of training and detecting cyberattacks such as jamming, black hole, and gray hole attacks. This study aimed to build upon previous research by showing a comprehensive queueing analysis. This study will provide network operators with the capability to reliably forecast the different aforementioned metrics within our framework, thus enabling network operators to proactively implement defensive measures in order to limit the impact of cyberattacks. To validate our results, we conducted a simulation utilizing a modified version SimComponent, a Python toolkit based on the open-source SimPy framework. Subsequently, we compared the projected values obtained from our analysis with the simulation. The analysis was validated by the simulation findings. This will serve as a way to speed up ML-based training to do identification for cyberattack risks.

Future work will seek to integrate our analysis with an intrusion detection system (IDS) for rapid real-time data generation, training, and threat mitigation within a SD-UAV network.

References

[1] A. Hanscom and M. Bedford, "Unmanned aircraft system (uas) service demand 2015-2035," *Literature review & projections of future usage*, 2013.

[2] G. Djukanovic, D. N. Kanellopoulos, and G. Popovic, "Evaluation of a uav-aided wsn for military operations: Considering two use cases of uav," *International Journal of Interdisciplinary Telecommunications and Networking (IJITN)*, vol. 14, no. 1, pp. 1–16, 2022.

[3] M. Kryk, K. Malon, and J. M. Kelner, "Propagation attenuation maps based on parabolic equation method," *Sensors*, vol. 22, no. 11, p. 4063, 2022.

[4] M. Gongadze, "Spacex's starlink becomes crucial tool in ukrainian war effort," Jan 2023. [Online]. Available: https://www.voanews.com/a/spacex-s-starlink-becomes-crucial-tool-in-ukrainian-war-effort/6922510.html

[5] A. del Aguila, J. V. Mendoza, S. B. Mandavilli, and J. McNair, "Remote and rural connectivity via multi-tier systems through sdn-managed drone networks," in *IEEE Military Communications Conference (MILCOM)*. IEEE, 2022, pp. 956–961.

[6] J. McCoy and D. B. Rawat, "Software-defined networking for unmanned aerial vehicular networking and security: A survey," *Electronics*, vol. 8, no. 12, p. 1468, 2019.

[7] S. H. Haji, S. R. Zeebaree, R. H. Saeed, S. Y. Ameen, H. M. Shukur, N. Omar, M. A. Sadeeq, Z. S. Ageed, I. M. Ibrahim, and H. M. Yasin, "Comparison of software defined networking with traditional networking," *Asian Journal of Research in Computer Science*, vol. 9, no. 2, pp. 1–18, 2021.

[8] Q. Zhao, P. Du, M. Gerla, A. J. Brown, and J. H. Kim, "Software defined multi-path tcp solution for mobile wireless tactical networks," in *IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 1–9.

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

[10] B.-W. Chen and S. Rho, "Autonomous tactical deployment of the uav array using self-organizing swarm intelligence," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 52–56, 2020.

[11] K. D. Atherton, "Why the us army wants an "aerial tier network" for better communications," Jun 2022. [Online]. Available: https://www.popsci.com/technology/army-develops-aerial-tier-network/

[12] O. Media, "Military embedded systems," Dec 2022, https://militaryembedded.com/comms/satellites/laser-comms-for-network-relay-demonstrated-at-us-navy-facility-by-general-atomics.

[13] D. Agnew, A. del Aguila, and J. McNair, "Detection of cyberattacks in an software-defined uav relay network," in *IEEE Military Communications Conference (MILCOM)*. IEEE, 2023, pp. 504–509.

[14] H. Chen, D. D. Yao *et al.*, "Fundamentals of queueing networks: Performance, asymptotics, and optimization," vol. 4, pp. 15–33, 2001.

[15] S. P. Meyn and D. Down, "Stability of generalized jackson networks," *The Annals of Applied Probability*, pp. 124–148, 1994.

[16] 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.

[17] M. A. B. S. Abir, M. Z. Chowdhury, and Y. M. Jang, "A software-defined uav network using queueing model," *IEEE Access*, pp. 91 423–91 440, 2023.

[18] M. A. Gray, "Discrete event simulation: A review of simevents," *Computing in Science & Engineering*, vol. 9, no. 6, pp. 62–66, 2007.

[19] M. A. B. S. Abir and M. Z. Chowdhury, "Digital twin-based software-defined uav networks using queuing model," in *2023 10th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2023, pp. 479–483.

[20] M. A. Ali, Y. Zeng, and A. Jamalipour, "Software-defined coexisting uav and wifi: Delay-oriented traffic offloading and uav placement," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 988–998, 2020.

[21] F. Xu, H. Ye, F. Yang, and C. Zhao, "Software defined mission-critical wireless sensor network: Architecture and edge offloading strategy," *IEEE Access*, vol. 7, pp. 10 383–10 391, 2019.

[22] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[23] K. Nisar, E. R. Jimson, M. Hijazi, and S. K. Memon, "A survey: Architecture, security threats and application of sdn," *Journal of Industrial Electronics Technology and Application*, vol. 2, no. 1, pp. 64–69, 2019.

[24] Y. Zhang and M. Chen, "Performance evaluation of software-defined network (sdn) controllers using dijkstra's algorithm," *Wireless Networks*, vol. 28, no. 8, pp. 3787–3800, 2022.

[25] 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.

[26] 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.

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

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

[29] S. Hares and R. White, "Software-defined networks and the interface to the routing system (i2rs)," *IEEE Internet Computing*, vol. 17, no. 4, pp. 84–88, 2013.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] Z. Yu, Z. Wang, J. Yu, D. Liu, H. Song, and Z. Li, "Cybersecurity of unmanned aerial vehicles: A survey," *IEEE Aerospace and Electronic Systems Magazine*, pp. 1–25, 2023.

[36] Z. Hu, R. Odarchenko, S. Gnatyuk, M. Zaliskyi, A. Chaplits, S. Bondar, and V. Borovik, "Statistical techniques for detecting cyberattacks on computer networks based on an analysis of abnormal traffic behavior." *International Journal of Computer Network & Information Security*, vol. 12, no. 6, 2020.

[37] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, and J. Li, "Performance comparison and current challenges of using machine learning techniques in cybersecurity," *Energies*, vol. 13, no. 10, p. 2509, 2020.

[38] A. Starke, K. Nagaraj, C. Ruben, N. Aljohani, S. Zou, A. Bretas, J. McNair, and A. Zare, "Cross-layered distributed data-driven framework for enhanced smart grid cyber-physical security," *IET Smart Grid*, vol. n/a, no. n/a. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/stg2.12070

[39] D. Gross, *Fundamentals of queueing theory*. John Wiley & Sons, 2008.

[40] P. J. Burke, "The output process of a stationary m/m/s queueing system," *The Annals of Mathematical Statistics*, vol. 39, no. 4, pp. 1144–1152, 1968.

[41] J. D. Little and S. C. Graves, "Little's law," *Building intuition: insights from basic operations management models and principles*, pp. 81–100, 2008.

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