IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 11, 2022, accepted April 8, 2022, date of publication April 21, 2022, date of current version May 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169137

The "Cyber Security via Determinism" Paradigm for a Quantum Safe Zero Trust Deterministic Internet of Things (IoT)

TED H. SZYMANSKI, (Member, IEEE)
Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada
e-mail: teds@mcmaster.ca

ABSTRACT The next-generation *Internet of Things* (IoT) will control the critical infrastructure of the 21st century, including the *Smart Power Grid* and *Smart Cities*. It will also support *Deterministic Communications*, where 'deterministic traffic flows' (D-flows) receive strict *Quality-of-Service* (QoS) guarantees. A *'Cybersecurity via Determinism'* paradigm for the next-generation *'Industrial and Tactile Deterministic IoT'* is presented. A forwarding sub-layer of simple and secure 'deterministic packet switches' (D-switches) is introduced into layer-3. This sub-layer supports many deterministic *Software Defined Wide Area Networks* (SD-WANs), along with 3 new tools for improving cyber security: *Access Control*, *Rate Control*, and *Isolation Control*. A *Software Defined Networking* (SDN) control-plane configures each D-switch (ie FPGA) with multiple deterministic schedules to support D-flows. The SDN control-plane can embed millions of isolated *Deterministic Virtual Private Networks* (DVPNs) into layer 3. This paradigm offers several benefits: 1) All congestion, interference, and *Distributed Denial-of-Service* (DDOS) attacks are removed; 2) Buffer sizes in D-switches are reduced by 1000+ times; 3) End-to-end IoT delays can be reduced to ultra-low latencies, i.e., the speed-of-light in fiber; 4) The D-switches do not require Gigabytes of memory to store large IP routing tables; 5) Hardware support is provided in layer 3 for the US NIST *Zero Trust Architecture*; 6) Packets within a DVPN can be entirely encrypted using *Quantum Safe* encryption, which is impervious to attacks by *Quantum Computers* using existing quantum algorithms; 7) The probability of an undetected cyberattack targeting a DVPN can be made arbitrarily small by using long *Quantum Safe* encryption keys; and 8) Savings can reach $10s of Billions per year, through reduced capital, energy and operational costs.

INDEX TERMS Cyber security, deterministic, the Internet of Things (IoT), quantum computing, zero trust, encryption, privacy, Software Defined Networking (SDN), industrial internet of things (IIoT), tactile Internet of Things, FPGA, Industry 4.0, deterministic Internet of Things.

I. INTRODUCTION

The next-generation *Internet of Things* (IoT) will control the critical infrastructure of the 21-st century, including *Smart Cities*, *Smart Healthcare*, the *Smart Power Grid*, *Smart Manufacturing*, and *Smart Transportation Systems*. The IoT will thus provide a critical infrastructure to enable much of the world's economic activity [1]–[3]. General Electric estimates that the next-generation IoT will control ≈ 50% of global GDP by year 2030, totalling ≈ $82 Trillion of GDP annually [1] (see section II).

Fig. 1a illustrates the USA electricity power grid, which includes over 9,000 power plants and about 3 million miles of transmission lines. Fig. 1b illustrates the USA natural gas pipeline network, which includes about half a million miles of pipelines. Cyberattacks against these critical infrastructures could have catastrophic consequences. In May 2021, the USA experienced a ransomware cyberattack against the *Colonial Pipeline*. This pipeline transports diesel, gasoline and jet-fuel from Texas to the east coast, and delivers about 45% of the east coast energy needs. The cybersecurity firm Emsisoft estimates that the USA was targetted by over 15,000 cyberattacks in 2020. The firm McAfee estimates that cybercrime cost the global economy $1 Trillion in 2020, and the firm Cybersecurity Ventures estimates that cybercrime will cost the global economy about $10.5 Trillion by 2025. Clearly, the world needs new ideas for the growing threat of cyberattacks.

The US *National Academy of Engineering* has identified 14 grand challenges for the 21-st century, which includes achieving fusion energy, carbon sequestration, clean water for the world, and *Securing Cyberspace* [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Chakchai So-In.

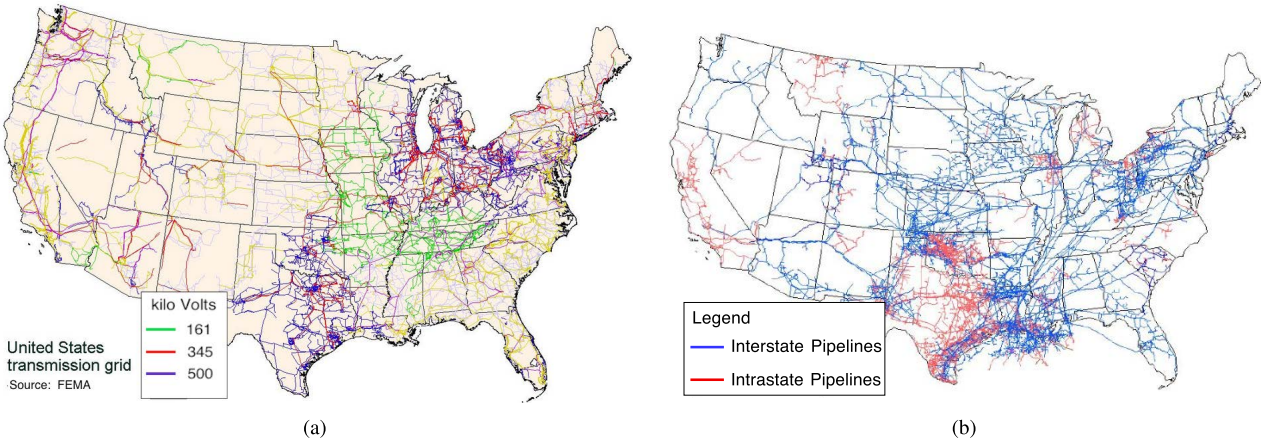

FIGURE 1. Critical infrastructures that are subject to cyberattacks. (a) USA electricity transmission grid. (b) USA natural gas pipeline network (source: US EIA).

In this paper, a new paradigm to significantly improve cybersecurity in the IoT is presented. A key feature of this paradigm is that a *Deterministic IoT* will allow a user to request the creation of a *'deterministic traffic flow'* (D-flow), with strict QoS (*Quality of Service*) guarantees on the data rate, delay, jitter and reliability. The *Deterministic IoT* thus requires a fair amount of pre-computation, in a centralized network control-plane with a global view of all existing D-flows and link utilizations. Specifically, the control-plane must pre-compute deterministic routes and schedules, to determine whether or not the user's request can be satisfied. In the *Deterministic IoT*, the timeline in which future events unfold is predetermined, and any deviation thereof represents an anomaly. This paper shows that the use of a centralized SDN control-plane with global knowledge of all D-flows and link utilizations can also significantly strengthen cybersecurity, to be impervious to attacks by foreseeable Quantum Computers, by explicitly controlling and tracking all communications.

According to the *Canadian Centre for Cyber Security*, the motivation and capabilities of cyberattackers varies widely. The most sophisticated cyberattacker is the *Nation-State Actor*. This actor has the financial backing of the nation-state, it is typically aligned with a national security agency, the level of technical expertise is very high, and the motivation is geo-political. The most popular cyberattacker is the *Cybercriminal*. This actor has limited financial backing, the level of technical expertise is typically moderate, and the motivation is profit from criminal activity (i.e., ransomware attacks).

A first goal of this paradigm is to allow each nation to significantly strengthen its national security, by reducing the number of cyberattacks against its critical infrastructure, from all types of cyberattackers including *Nation-State actors* and *Cybercriminals*. The number of cyberattacks per year can be reduced from potentially 1,000s, to a very small number, where the expected number of successful attacks per year from 'external' cyberattackers is zero. (An external cyberattacker does not have access to a secured machine.) A second goal of this paradigm is to dramatically lessen the impact of any successful cyberattack, by supporting a *Zero Trust Architecture (ZTA)*. A third goal of this paradigm is to save network operators $10s of Billions annually through reduced capital, energy and operational costs, by exploiting a more secure, efficient and easily-controllable deterministic networking infrastructure based upon *Software Defined Wide Area Networks (SD-WANs)*, using FPGAs. A final goal of this paradigm is to significantly reduce the costs of cybercrime to the global economy, costs which are currently estimated to be about $1 Trillion annually (in 2020).

This paper achieves its goals by exploring the intersection of 5 topics: (i) *Deterministic Communications*, (ii) *Post-Quantum Cryptography (PQC)*, (iii) the *Zero Trust Architecture (ZTA)*, (iv) the *Access Control System (ACS)*, and (v) the *Intrusion Detection System (IDS)*. These 5 topics are briefly reviewed in section II. While each topic has been studied by researchers in isolation, no prior research papers have explored the intersection of the first 2 or more topics.

In late 2021, the *'Log4j'* security threat was discovered. According to the US CISA (*Cybersecurity and Infrastructure Security Agency)*, this security threat is amongst the most significant security threats ever discovered, and hundreds of millions of computers world-wide are vulnerable to cyberattacks. This paper will also discuss how the proposed paradigm can mitigate the damage due to this type of cyberattack in section VII.

The existing IoT is based upon a *Best-Effort* (BE) communications paradigm, with inherently large latencies [5]–[7]. The BE-IoT consists of the union of about 80,000 'autonomous network domains' distributed around the world. In its simplest form, and in the absence of extra hardware/software such as *Firewalls* to limit connectivity, the BE-IoT consists of billions of computers (or devices), interconnected with a large number of *Best-Effort IP routers*. The computers and devices typically run *BSD (Berkeley*

Software Distribution) socket software (i.e., *Berkeley Sockets*), to provide connectivity. The primary appeal of the BE-IoT is the ubiquitous communications it offers: In the absence of *Firewalls*, any user can easily communicate with any of the billions of computers around the world, using the fairly simple *Berkeley Sockets* software interface and the *Internet Protocol (IP)* in layer 3. However, the primary appeal of the BE-IoT is also its primary weakness: In the absence of firewalls, cyberattackers are also free to communicate with any of these billions of computers around the world, over the *Berkely Sockets* interface. *Firewalls* can be added to limit connectivity, but a very large number of *Firewalls* would be needed to protect billions of computers, and each firewall must be properly configured, representing a significant challenge.

The BE-IoT network provides its best-effort to deliver traffic, but it suffers from several serious deficiencies: (i) It offers no inherent *Access Control* or *Rate Control*. Any user is free to send messages to any of the billions of computers in the world, at any data-rate. Thus, it is relatively easy to create *Distributed Denial of Service* (DDOS) attacks. Cisco estimates that there will be 15.4 million DDOS attacks in 2023 [8]; (ii) It offers no inherent *Isolation Control*. The BE-IoT cannot be partitioned into isolated subsets to contain cyberattacks; (iii) The BE-IoT can experience significant congestion, excessive delays of 100s of milliseconds, and high packet loss rates; (iv) The BE-IoT provides no deterministic QoS guarantees on the data rate, delay, jitter or delivery; (v) The paths taken by packets can change dynamically, due to the layer-3 dynamic routing protocol, causing large fluctuations in end-to-end delays; (vi) To mitigate congestion, the BE-IoT is typically *over-provisioned*, i.e., it operates well below its peak capacity, typically below 50 percent capacity, thereby incurring excess capital and energy costs of $10s of Billions per year (see section VII); (vii) To limit connectivity and improve security, numerous *Firewalls* are deployed in the BE-IoT. According to Cisco, up to 95% of firewalls are configured manually, such that operational costs can be 2 or 3 times the network capital costs. Thus, the operational costs of the BE-IoT are extremely high [8]. (viii) Layer 3 of the BE-IoT is inherently insecure (see section II.A). As a result, the global BE-IoT suffers from millions of cyberattacks per year, including DDOS, *Phishing*, *Spoofing*, *Replay*, *Man-in-the-Middle (MITM)*, *Ransomware* and *Remote Code Execution (RCE)* attacks. (Common IoT attacks are reviewed in Appendix C).

The BE-IoT offers a weak level of security by supporting *Best-Effort Virtual Private Networks* (BE-VPNs) in layers 3 and 4 (see section II). A BE-VPN is a set of one or more partially encrypted connections within the BE-IoT, that are reserved for the use of a single corporation or entity, as specified in the IETF *'Internet Protocol Security' (IPsec)* RFC [10].

Unfortunately, BE-VPNs have several significant security weaknesses (see section II for details): (1) Each BE IP router needs to read the *IPsec* packet-header to make layer-3 routing decisions. Thus, packet-headers must remain unencrypted and can be manipulated by cyber-attackers. A cyberattack in which an adversary modifies a packet header to masquerade as a trusted peer is called a *Spoofing* cyberattack, which can be used to create a more complex *Main-in-the-Middle* cyberattack; (2) In the absence of firewalls, a secured computer participating in a BE-VPN can be the target of a cyberattack from any of the billions of other computers around the world over the insecure *Berkeley Sockets* interface - *there is no isolation*; (3) In the absence of firewalls, a compromised computer participating in a BE-VPN is free to launch cyberattacks against any of the billions of other computers around the world over the insecure *Berkeley Sockets* interface - *there is no containment*; (4) The BE-VPNs still operate under the BE communications paradigm; they are still subject to congestion, interference and several types of cyberattacks, including the aforementioned *DDOS, Phishing, Spoofing, MITM, Ransomware* and RCE attacks. The security of BE-VPNs in layers 3 and 4 is thus severely inadequate for the *Smart Systems* of the future [9]–[11].

A traditional BE-VPN is also vulnerable to cyberattacks involving *Quantum Computers*. It is expected that *Quantum Computers* will be able to crack the *Public Key Cryptography (PKC)* used in BE-VPNs in the near future, potentially by the year 2030 (see section II). In the quantum-based *'Harvest Now, Decrypt Later'* cyberattack, an adversary may eavesdrop and record the encrypted communications of a BE-VPN today. The adversary may decrypt the communications a few years into the future, when a sufficiently powerful *Quantum Computer* exists. To combat this attack, the US NIST (*National Institute for Standards and Technology*) has initiated a project on *Post Quantum Cryptography (PQC)*. It is planning to standardize one or more *Quantum-Safe* public key encryption and digital signature schemes to be used in BE-VPNs, in the timeframe of 2022 to 2024. A BE-VPN which utilizes the forthcoming NIST PQC standards can be called a *'Quantum-Safe BE-VPN'*, and several companies and research institutes have recently demonstrated software for *Quantum-Safe BE-VPNs* (see section II). While *Quantum-Safe BE-VPNs* will be immune to the *Harvest Now, Decrypt Later* cyberattack, they are still vulnerable to the traditional cyberattacks associated with *BE-VPNs* described earlier, such as DDOS attacks.

Ultra-low latency networks have attracted significant attention recently [12]–[17]. The use of deterministic communications in small layer-2 networks, such as *Deterministic Ethernet* [15], is well-established (see section II). The next-generation IoT network is also expected to support *Deterministic Communications* [17], to offer ultra-low latencies. In a *Deterministic IoT*, D-flows must receive deterministic or guaranteed data-rates and strict QoS guarantees. They must be immune to congestion, interference and DDOS attacks. Deterministic communications thus offers a fundamentally new dimension for exploring cybersecurity in the IoT. The next-generation IoT is expected to encompass the *Industrial Internet of Things*, the *Tactile Internet of Things*, and the

Consumer Internet of Things projects, as they will all benefit from deterministic communications.

In this paper, we present a new security paradigm for the next-generation *Deterministic IoT*, called the *'Cybersecurity via Determinism'* paradigm. (This paper unifies several prior papers, and adds several new results on cybersecurity.) A new sub-layer of simple and secure 'deterministic packet switches' (D-switches) is introduced into layer 3, under the control of a *Software Defined Networking* (SDN) control-plane. This sub-layer implements one (or more) deterministic SD-WANs. We refer to this new sub-layer as 'layer-3a', the forwarding layer for deterministic traffic. The D-switches forward packets along predetermined paths, where the routing and scheduling have been performed in advance by the SDN control-plane. This new sub-layer introduces three new tools for enabling deterministic communications, and for strengthening cybersecurity, i.e., *Access Control*, *Rate Control* and *Isolation Control*. The SDN control-plane can configure millions of isolated, non-interfering D-flows into the new sub-layer. Each D-flow follows a fixed path through the IoT network, and supports a deterministic (or guaranteed) data-rate.

The proposed paradigm also supports *Isolation Control*, i.e., *Network Slicing*, where the network resources can be logically partitioned to yield millions of isolated, non-interfering *network partitions*. The traffic in each partition is isolated from the traffic in any other partition. Each *network partition* can be configured in 2 modes: (a) the *'un-encrypted'* mode; and (b) the *'encrypted'* mode, where every packet is encrypted at the source, and remains fully-encrypted until it reaches the destination. A *network partition* operating in the un-encrypted mode is called a *Deterministic Virtual Network* (a DVN). A *network partition* operating in the encrypted mode is called a *Deterministic Virtual Private Network* (a DVPN). A DVPN has much stronger security and privacy than the BE-VPNs used in today's BE-IoT, as will be shown.

The proposed paradigm also provides hardware support in layers 3 and 4 for the US NIST *Zero Trust Architecture* (ZTA) [18], [19] (see section II). Specifically, all communications within a DVPN are encrypted with long *Quantum Safe* keys, which are impervious to attacks by *Quantum Computers*, and every packet requires an *Authorization-Check* (see section III). The ZTA builds upon the concept of *Access Control Systems*, and it extends the access control concept to individual resources, where the access to every critical resource is determined by one (or more) rule-based *Access Control* policy-engines. Only D-flows which are approved by the *Access Control* policy-engines will be created by the SDN control-plane. The use of the SDN control-plane, combined with ZTA/ACS, and the *Intrusion Detection System* described ahead, can significantly improve cybersecurity.

To deploy this paradigm, some of the major cloud services providers, i.e. Google, Amazon, Apple, and Microsoft, could each implement their own deterministic SD-WAN in forwarding sub-layer 3a, to create millions of isolated *zero trust* DVPNs for their customers, as a new revenue service. Each critical public service, i.e., the *Smart Power Grid*, and each government department, i.e., the *Dept. of Defense*, can create many isolated DVPNs, each with significantly improved cybersecurity, from multiple independent cloud services providers, to achieve their mission. Several DVNs can also be created to support the efficient transmission of consumer-oriented BE-IoT traffic such as IP video (without default encryption). According to Cisco, about 82% of the traffic in the BE-IoT comprises IP video [26].

In the proposed paradigm, the tasks of routing and scheduling of D-flows are removed from the D-switches, and are performed in advance by the SDN control-plane. The SDN control-plane only admits authorized D-flows, each with a maximum deterministic data-rate which it cannot exceed. The proposed *Deterministic IoT* thus can quickly detect unauthorized packets targeting a DVPN, i.e., such packets will not have pre-authorization or pre-determined routes and schedules. The proposed paradigm thus provides a *'guaranteed' Intrusion Detection System*, where the probability of an undetected cyberattack can be made arbitrarily small. Section III shows that the probability of a successful undetected cyberattack by an external cyberattacker targeting a DVPN can be made arbitrarily small, and effectively zero, by using long *Quantum Safe* encryption keys, which are impervious to attacks from *Quantum Computers* using existing quantum algorithms. According to the US NIST, even *Quantum Computers* cannot crack an encrypted message using *Quantum Safe* encryption keys with thousands of bits [68].

The proposed *Cybersecurity via Determinism* paradigm exhibits several attractive properties. The first 6 properties have been established in prior papers, and they are repeated here for completeness. The last 9 properties are new, and are denoted with a *.

- The SDN control-plane can embed millions of isolated, non-interfering *network partitions* [17], [21], [22];
- All congestion, interference and DDOS attacks associated with the BE-IoT can be removed [17], [21], [22];
- IoT links in sub-layer 3a can operate at $\approx$ 100% of peak capacity, saving \$10s of Billions per year in excess capital and energy costs [17], [20];
- IoT buffer sizes can be reduced by a factor of 1,000+ times, and IoT end-to-end delays can be reduced to ultra-low latencies, i.e., the speed-of-light in fiber [17], [20], [21];
- Layer-3 D-switches do not perform any distributed routing or scheduling algorithms. They are much simpler than layer-3 IP routers, and they can be synthesized on a single integrated circuit, i.e., a Field Programmable Gate Array (FPGA), for a dramatic cost reduction [21];
- IoT energy-efficiency improves considerably [17], [21];
- The paradigm supports multiple SD-WANs, where each SD-WAN supports: *Deterministic communications*, PQC, the ZTA/ACS, and a guaranteed IDS. The use of SD-WANs can save network operators \$10s of Billions annually in reduced

operational costs, by exploiting a more secure, efficient and easily-controllable software-defined networking infrastructure [8];*
- The paradigm provides hardware support in layers 3 and 4 for the US NIST *Zero Trust Architecture*, wherein an *Access Control System* controls access to every critical resource using one (or more) rule-based policy-engines. The use of the SDN control-plane, the ZTA/ACS, and the guaranteed IDS can significantly reduce the number of cyberattacks from external cyberattackers, i.e., DDOS, *Spoofing*, *Phishing*, MITM, *Ransomware* and RCE attacks;*
- The packets within a DVPN can be fully encrypted from end-to-end using longer *Quantum-Safe* encryption schemes, which are impervious to attacks by *Quantum Computers* using existing quantum algorithms. This feature effectively eliminates the aforementioned types of cyberattacks, as well as the *Harvest Now, Decrypt Later* cyberattack;*
- In a DVPN, packet headers are not examined to make layer-3 routing decisions. Packets can be fully encrypted and cannot be manipulated by cyberattackers;*
- The probability an external cyberattacker can compromise any DVPN can be made arbitrarily small, and effectively zero, by using longer *Quantum Safe* encryption keys, which are impervious to attacks by *Quantum Computers* using existing quantum algorithms;*
- The probability an internal cyberattacker in a compromised DVPN can compromise any other remote DVPNs can be made arbitrarily small and effectively zero (*the damage is contained*);*
- The D-switches do not require Gigabytes of high-speed memory for insecure layer-3 IP routing tables;*
- The D-switches do not require a processor or operating system or the insecure *Berkeley Sockets* software, to support insecure layer-3 protocols such as: *Internet Control Message Protocol* (ICMP), *Border Gateway Protocol* (BGP), and *Interior Gateway Protocol* (IGP), which all introduce complexity and security threats;*
- The relatively-simple D-switches are much easier to secure compared to a layer-3 IP router.*

The routing and scheduling algorithms of the proposed SDN control-plane have been implemented in software. To test the system, a simplified hardware testbed network of 26 D-switches, arranged as a USA backbone network, has been embedded into an Altera FPGA. The SDN control-plane and hardware testbed work according to specifications [17], [21], [22].

Relationship to Prior Work: Some aspects of the proposed *Deterministic IoT* have been presented in prior IEEE publications. The hardware testbed and the performance of the *Deterministic IoT*, for other traffic scenarios, have been presented in [17], [21], and [22]. The stronger cybersecurity provided by the SDN control-plane for deterministic networks operating with DVNs has been presented in [21] and [22]. The multi-commodity *Max-Flow Min-Cost* routing algorithm used in the SDN control-plane has been presented in [62]. The low-jitter scheduling algorithms used in the SDN control-plane to achieve $\approx 100\%$ throughput with ultra-low latencies have been presented in [20] and [60].

However, these papers did not present the unifying concept of the *'Cybersecurity via Determinism'* paradigm. These previous papers did not explore SD-WANs which operate at the intersection of 5 distinct research topics, including: (i) *Deterministic Communications* (ii) PQC, (iii) the ZTA, (iv) the ACS, and (v) the IDS. They did not describe *Quantum-Safe* DVPNs, where entire packets are encrypted from end-to-end using *Quantum-Safe* encryption schemes, which are impervious to attacks by *Quantum Computers* using existing quantum algorithms. This paper also presents extensions of the *Quantum-Safe* AES and RIJNDAEL symmetric encryption algorithms to support encryption keys with 1,000s of bits.

This paper is organized as follows: Section II briefly reviews several topics, including the BE-IoT, the *Deterministic IoT*, PQC, the ZTA, the ACS, the IDS, VPNs, and the IETF Deterministic Networking project. Section III presents the key features of the *Deterministic IoT*. It also presents extensions of the AES and RIJNDAEL algorithms to support long keys. Section IV presents the SDN control-plane. Section V presents a security analysis. Section VI presents experimental results. Section VII discusses extensions and implications of the paradigm. Section VIII concludes the paper. Section IX contains the appendices. Appendix A contains a list of common acronyms. Appendix B describes a BE IP router. Appendix C describes common cyberattacks in the BE-IoT. Appendix D describes the most important cyberattacks in the IETF deterministic network.

II. REVIEW

A. SECURITY IN THE BE-IoT

1) LACK OF CENTRALIZED CONTROL

One of the original design goals of the Internet, attributed to P. Baran of the RAND Corporation, was the ability to withstand a nuclear attack which could disrupt a significant fraction of the network. A key feature was the lack of a centralized controller which could be compromised in an attack. This key feature of decentralized control is dominant today, several decades later.

2) LAYER 3 INTERNET PROTOCOL (IP) IS INSECURE

Much of the BE-IoT was designed based upon a *Trust* model. The basic IPv4 and IPv6 packet formats do not have any means to *authenticate* the source or the destination of an IP packet, i.e., to confirm that they are who they claim to be. They also do not have any means to ensure the *integrity* of the packet contents, i.e., to ensure that the packet contents have not been altered. A cyberattacker can easily masquerade as a trusted peer. It can create malicious IP packets with fraudulent source or destination IP addresses, or fraudulent packet contents, and insert these into the IP network.

3) LAYER 4 TRANSPORT LEVEL SECURITY (TLS)

The *Berkeley Sockets* software uses the IP packet formats, and it also does not have any means for authenticating the packet headers, or verifying the integrity of the packet contents, and thus it is inherently insecure. Netscape developed a secure version of the socket software in the 1990s, called the *Secure Socket Layer (SSL)*. In 1999, the IETF proposed an update to SSL, and it re-named it as the *Transport Layer Security (TLS)* protocol, to signify the change in ownership [23]. The IETF TLS provides the means to authenticate the sender and receiver of a TLS message, and it provides the means to verify the integrity of the TLS message contents. TLS exploits the concept of the *Public Key Infrastructure (PKI)*. TLS will only open a secure connection from a user to a website with a valid *SSL or TLS Certificate*. The certificate indicates that the website has been vetted by a *Certificate Authority (CA)* within the PKI model, where the CA has the sole authority to issue SSL or TLS certificates. These certificates have a digital signature (i.e., the output of a hash function which uses a key), to ensure that they have not been tampered with. The SSL or TLS certificate contains the public key of the web-site, which can then be used to establish secure symmetric encryption keys between the user and the website, for secure communications with improved energy efficiency and performance.

4) LAYER 3 BGP ROUTING IS INSECURE

Unfortunately, TLS operates at layer 4 of the OSI (*Open Systems Interconnection*) model, the layer above *Internet Protocol (IP)* layer 3. Thus, all IP communications in layer 3 still operate on the trust model and are inherently insecure. Even the routing of IP packets in layer 3 is based on the insecure trust model [27]–[29]. The *Border Gateway Protocol (BGP)* implements the layer 3 distributed IP routing protocol. Each BGP router maintains a routing table, and the destination IP address in an incoming IP packet is used to select an outgoing link in each BGP router. Each autonomous system in the IoT is associated with a range of IP addresses called a 'routing prefix'. Each BGP router has a routing table, to associate a preferred outgoing IoT link with each routing prefix. The routing tables in each BGP router are updated dynamically, where each BGP router receives *Update* messages periodically from its peers, with updates to the BGP routing tables. These *Update* messages use the basic insecure IP packet format. Hence, a cyberattacker can easily send a fraudulent *Update* message to a BGP router, to direct the traffic to an alternate destination that it controls, where the packet can be copied or modified, for example as part of a *Harvest Now, Decrypt Later* cyberattack. The US NIST has proposed the use of *Access Control Lists (ACLs)* to improve the security of the BGP protocol in layer 3 [27], [28]. The IETF has proposed a secure version of the BGP routing protocol called *BGPsec* in 2017 [29]. Unfortunately, *BGPsec* only works well if a significant number of BE IP routers use it, and no statistics have been reported on how widely *BGPsec* has been deployed.

5) CISCO ANNUAL INTERNET REPORT (2018–2023)

According to the *'Cisco Annual Internet Report (2018-2023)'*, the BE-IoT is expected to interconnect about 29.3 Billion devices by 2023 [8]. The number of best-effort traffic flows supported by the IoT can be estimated at about 30 billion (assuming each device maintains one traffic flow), and about 50% of the traffic flows represent M2M (*Machine-to-Machine*) traffic. These devices are expected to generate vast amounts of data. According to Cisco, the BE-IoT is expected to carry about 9.1 Exabytes of traffic per day in 2021 (1 Exabyte = 1 billion Gigabytes), and 82% of this figure will be IP video traffic [26]. Achieving cybersecurity in the BE-IoT, with no centralized controller, with an insecure layer 3, with no *Access Control Systems*, and no global knowledge of which traffic flows exist at any one time, is an outstanding challenge, i.e., see [30]–[34].

6) MACHINE LEARNING, DEEP LEARNING AND ARTIFICIAL INTELLIGENCE

The use of machine learning and deep learning technologies, to process network traffic and statistics in an attempt to discover cyberattacks, has attracted significant attention: [35]–[38]. IBM has expanded upon this approach to include *Artificial Intelligence for Cybersecurity* [39]. While these approaches are promising, they must process vast amounts of network traffic, i.e., collectively potentially exabytes of traffic per day. They thus consume significant resources, i.e., large amounts of computational power, time, energy and capital costs.

7) TRANSFORMING THE INFRASTRUCTURE

According to Cisco, the growth of IoT costs is prohibitive [8]. Up to 95% of network configuration changes (configuring routers, firewalls, etc) are performed manually, such that operational costs are 2 to 3 times higher than network capital costs. Thus, operational costs of the BE-IoT can be measured in the $100s of Billions per year (see section VII). According to Cisco, a need to *'Transform the Infrastructure'* exists, using innovations such as SDN controllers and SD-WANs, which allow for unified domain controls and policies [8].

The proposed *Cybersecurity via Determinism* paradigm exploits the existence of a logically centralized SDN control-plane, which manages many deterministic SD-WANs. The control-plane supports the intersection of 5 distinct research topics, i.e., *Deterministic Communications*, PQC, ZTA, ACS, and IDS, to significantly improve cybersecurity.

B. THE DETERMINISTIC IoT

1) THE IETF IntServ AND DiffServ MODELS

The IETF has proposed the *Integrated Services (IntServ)* model to support deterministic communications in [40]. The *IntServ* model supports two classes of traffic, (i) best-effort traffic flows, and (ii) deterministic traffic flows which receive strict QoS guarantees. The use of *IntServ* in the Internet was proposed in the late 1990s, in 5 IETF RFCs:

- *Integrated Services in the Internet Architecture* [40],
- *The use of RSVP with IETF Integrated Services* [41],
- *Specification of Guaranteed Quality of Service* [42],
- *General Characterization Parameters for Integrated Service Network Elements* [43],
- *Network Element Service Specification Template* [44].

In the *IntServ* model, a deterministic traffic flow will reserve bandwidth in each link along and end-to-end path through the Internet, from the source to the destination. A resource-reservation protocol, such as the IETF RSVP protocol, is used to reserve bandwidth on each link along the end-to-end path. These RFCs did not address the issue of distributed versus centralized network control, or the routing and scheduling problems inherent with providing deterministic QoS guarantees. This early work formalized the concept of *Integrated Services* for the IoT network, which supports both best-effort and deterministic traffic flows, but it did not lead to any IETF standards.

The IETF has proposed the *Differentiated Services (DiffServ)* model to support prioritized best-effort traffic classes in the BE-IoT [45]. The *DiffServ* model supports 3 classes of prioritized traffic flows, called the *Assured Forwarding (AF)*, the *Expedited Forwarding (EF)*, and the *Default Forwarding (DF)* classes. The AF class provides the highest priority service, for the most demanding/important BE-traffic. The EF class provides medium-priority service, for low-latency, low-loss BE traffic. The DF class provides the lowest priority service, for regular BE traffic. Unfortunately, *DiffServ* cannot provide any deterministic QoS guarantees, as the BE-IoT is vulnerable to numerous cyberattacks, such as DDOS attacks, and it cannot provide guaranteed service. Nevertheless, the *DiffServ* model can be used in the proposed *Deterministic IoT* to support prioritized traffic classes, which can receive deterministic service and thereby provide deterministic guarantees. Furthermore, in the proposed *Deterministic IoT* one or more new traffic classes can be added for deterministic traffic, which provide increasingly strict QoS guarantees [20]. In the *DiffServ* model, the queueing and scheduling are simplified relative to the *IntServ* model, as each output link in a router can contain one class-queue for each traffic class. Each class-queue can buffer packets for 1000s of traffic flows that belong to that given traffic class, and these packets can receive simple *'First Come First Served' (FCFS)* service.

2) QoS GUARANTEES

The problem of achieving reasonable throughput and QoS guarantees in the Internet is a well-known problem [46]–[51].

Many BE IP routers use switches that employ *Input-Queueing* (IQ), or *Combined Input and Output Queueing* (CIOQ). The problem of designing packet buffers for IP routers is discussed in [52] and [53]. The problem of scheduling packets from multiple best-effort traffic flows through an IQ or CIOQ switch, to achieve reasonably fair service, is a well-known problem [54], [55]. The scheduling problem can be formulated as a matching problem on a bipartite graph. Reference [54] has shown that a *Maximum Weighted Matching (MWM)* algorithm can achieve 100% throughput in an IQ switch, but the time complexity renders the algorithm too complex in practice.

Traditional BE IP routers typically use the iSLIP scheduling algorithm, since it is fast and yields good performance. Under a random and uniform traffic model, the iSLIP algorithm can achieve a maximum throughput of 100% [55]. However, the delays can be large at high loads. In practice the BE IP routers (and the BE-IoT) are often *over-provisioned*, i.e., the links are designed to operate at peak loads well below 100%. Google has shown that links in the BE-IoT typically operate at loads between 20% and 30% [24], [25]. A BE-IoT link with a peak capacity of 200 Gigabit-per-second (Gbps), which operates at 25% load, will carry about 50 Gbps of traffic. In this case, about 75% of the link capacity is left unused, and about 75% of the capital cost of the BE-IoT link is wasted.

3) DETERMINISTIC TRAFFIC

The problem of scheduling packets from multiple deterministic traffic flows through an IQ or CIOQ switch, to achieve strict QoS guarantees, is another well-known problem which is summarized in [49] and [56]–[59]. It has been shown that IQ and CIOQ switches can achieve up to 100% throughput, when carrying deterministic traffic flows along with strict QoS guarantees, however the scheduling algorithms are often NP-complete.

The theory for a *Deterministic IoT* network which supports the IETF *IntServ* and *DiffServ* models for deterministic traffic was presented in [17], [20]–[22], and [60]–[64]. The proposed *Industrial and Tactile Deterministic IoT* network supports both best-effort traffic flows, and deterministic traffic flows with strict QoS guarantees for mission-critical traffic such as telerobotic surgery [21], [61]. These papers presented the concept of a logically centralized controller, along with routing and scheduling algorithms to achieve 100% throughput in a network of IQ or CIOQ switches, with near-minimal end-to-end delay and jitter, for all deterministic traffic flows simultaneously. The use of a centralized SDN control-plane to improve cybersecurity in the *Deterministic IoT* supporting *IntServ* and *DiffServ*, before the days of *Post-Quantum Cryptography*, was explored in [21] and [22].

Huawei has recently described a large-scale deterministic IP network in company documents, videos, and research papers [65], [66].

4) THE IETF DETERMINISTIC NETWORKING (DetNet) PROJECT

In late 2014, the IETF created a *Deterministic Networking (DetNet)* working group, and it embarked on a renewed effort to support deterministic traffic flows in layer-2 bridged networks and layer-3 *Wide Area Networks* (WANs), with bounds on latency, packet loss, jitter and reliability. This IETF DetNet effort is summarized ahead in section II.H.

C. QUANTUM COMPUTING AND QUANTUM SAFETY

1) QUANTUM SUPREMACY

In 2019, Google claimed that its 54-qubit *Sycamore Quantum Computer* reached *Quantum Supremacy*, i.e., it performed computations which take 10,000 years on a classical supercomputer, in about 200 seconds [67]. IBM, which had its own *Quantum Computer* with 53 qubits at that time, challenged this claim, stating that the computations on a classical supercomputer would only take 2.5 days. IBM has developed a *Quantum Computer* with 127 qubits in 2021. IBM aims to develop a *Quantum Computer* with over 1,000 qubits by 2023. Both IBM and Google aim to develop a *Quantum Computer* with 1 million qubits in 2030. Such a machine would exploit quantum error-correcting codes, to allow many qubits to remain coherent, and is expected to cost Billions of dollars.

2) PUBLIC KEY CRYPTOGRAPHY (PKC)

Currently, *Public-key Cryptography (PKC)* is used to secure much of the communications of the IoT [68]. According to the ETSI (*European Telecommunications Standardization Institute*) [69], PKC relies upon 2 classes of computationally-difficult problems which classical computers cannot solve:

- The *Integer Factorization* problem, which is used in the RSA (Rivest Shamir Adleman) cryptosystem [70];
- The *Discrete Logarithm* problem, which is used in ECC (Elliptic Curve Cryptography) [71].

According to the US NSA (*National Security Agency*), the terms *Quantum-Resistant*, *Quantum-Safe*, and *Post Quantum Cryptography* are all terms used to describe algorithms that are *'widely recognized by experts to be resistant to cryptanalytic attacks from both classical and Quantum Computers'* [72].

3) SHOR'S ALGORITHM

Two quantum algorithms relevant to cybersecurity are *Shor's algorithm* and *Grover's algorithm* [68]. *Shor's Algorithm* can factor very large numbers with exponential speedup [73]. It can be used to crack traditional *Public Key* encryption algorithms such as RSA. Researchers estimate that to crack RSA with a 2,048 bit key would require a *Quantum Computer* with 20 million qubits, and would take 8 hours [74]. Such a computer might be feasible in 1 or 2 decades. Recent research has shown that the same RSA key can be cracked in 177 days using 13,436 qubits and a multimode memory [75]. Public key encryption schemes such as RSA are very widely used, and hence there is a strong need for *Quantum Resistant* or *Quantum Safe* public key encryption schemes.

References [76], [77] have compared the time-complexity of cracking RSA encryption using Shor's algorithm on two competing *Quantum Computing* platforms, the (i) trapped-ion and (ii) superconducting platforms, when using finite resources. It was shown that given sufficiently long RSA keys (i.e., 8.51 KB and 85.1 KB for the trapped-ion and superconducting platforms respectively), Shor's algorithm can still take over 100 years to crack RSA. Thus, the security of RSA encryption with very long keys can still be guaranteed, when considering the time-complexity of *Quantum Computing* platforms using finite resources. The US NSA must ensure that classified information of the US government must remain secure for several decades, and it currently recommends the use of RSA keys with at least 3,072 bits [72].

4) GROVER'S ALGORITHM

Grover's Algorithm performs an unstructured database search, and can be used to crack symmetric encryption schemes such as AES [78], [79]. It can effectively evaluate $O(L)$ cypher keys, and recover the plaintext with high probability, with a quadratic speedup (in $O(\sqrt{L})$ quantum queries). It has been proven that unstructured searches cannot reach exponential speedup [68], and that *Grover's Algorithm* is asymptotically optimal [80], [81]. Grover's original algorithm finds the desired item in a database with high probability. It can be modified to yield an exact quantum search algorithm, to find the desired item with probability 1, while maintaining $O(\sqrt{L})$ quantum queries [82]. The optimality of this exact quantum search is established in [83].

Grover's algorithm can crack private key encryption algorithms such as AES (with small keys), but the speedup is much lower than exponential. Hence, the AES private key encryption scheme is considered to be *Quantum Safe*, when the keys are sufficiently large. AES with key lengths ≥ 256 bits are considered to be uncrackable with current technology. Recent research has shown that a *Quantum Computer* with 7,000 qubits could crack AES-256 for 7 rounds of computation [84], [85]. Such a computer may be feasible within a decade. However, AES-256 uses 14 rounds of computation, and hence there is a large *'Security Margin'*, equivalent to 7 rounds of computation, before AES-256 can be cracked. However, a new quantum search algorithm that exploits structure might be developed in the future, which could potentially improve upon *Grover's Algorithm*, and potentially render AES-256 crackable. Hence, there is a strong need for private key encryption schemes with very large security margins.

5) POST-QUANTUM CRYPTOGRAPHY (PQC)

The US NIST has initiated a project on *Post Quantum Cryptography (PQC)* in 2016 [86]. The US NIST initiated a *Post Quantum Cryptography Standardization Process* in 2017, with a goal to standardize one or more additional public-key encryption and key-establishment algorithms, and digital signature algorithms [87].

Proposals for the new PQC *Standardization Process* were due in Nov. 2017. In Dec. 2017, NIST approved 69 candidate algorithms for the first round competition. A status report on the first round of proposals was completed in Jan. 2019 [87], and 26 candidate algorithms proceeded to the second round competition, including 17 algorithms for public-key encryption and key-establishment, and 9 algorithms for digital

signatures. A status report on the 2nd round of proposals was completed in July 2020 [88]. The third and final round has 7 proposals under consideration for standardization, 4 for public-key encryption and key-establishment, and 3 for digital signatures. The third round also has 8 *Alternate Candidate* proposals under consideration, although they will not be considered for standardization in the 2022-2024 timeframe. The third round candidates are summarized by NIST in [88], and by the ETSI in [89]. The final NIST approved algorithm(s) for standardization will be released in the 2022-2024 timeframe.

6) SECRET PRE-SHARED KEYS

The IETF *Internet Key Exchange* version 2 (IKEv2) protocol currently uses *Public Key Cryptography* to negotiate secret keys used to secure the communications between 2 entities, and IKEv2 will be vulnerable to attack by *Quantum Computers*. The IETF anticipates that IKE will be extended to support *Quantum Safe* key exchange algorithms, once the NIST Standardization Process completes [90]. Until that time, quantum-safe communications between entities can be achieved using *Pre-Shared Keys* [69], [90]. In the proposed paradigm, secret *Pre-Shared Keys* can also be used to protect the communications between the secured components.

D. THE ZERO TRUST ARCHITECTURE (ZTA)

In April 2021, the *Ransomware Task Force*, a group of industry experts, submitted a report entitled *Combatting Ransomware - A Comprehensive Framework for Action*, to the US government [96]. In May 2021, the US president issued an *Executive Order* entitled *Improving the Nation's Cybersecurity* [97], which requires that the US advance towards a *'Zero Trust Architecture'*, as described by the US NIST [18].

In the traditional *Perimeter-Based* security model used in the BE-IoT over the last few decades, an enterprise typically has a *trusted-zone* (i.e., a corporate LAN and its resources), with a well-defined *security perimeter*. A user is first authenticated at the perimeter, using an *Access Control System* (described ahead), and is then admitted into the trusted-zone, where that user can typically access resources unimpeded. This model allows for considerable *lateral movement and internal cyberattacks* within the trusted-zone [18]. The growing use of a remote workforce, cloud computing and the IoT has lowered the relevance of this model.

Zero Trust (ZT) is a set of cybersecurity principles that *de-emphasize the defense of security perimeters. ZT emphasizes protecting resources, rather than network segments.* ZT assumes that all communications are monitored, and could be compromised by cyberattackers. ZT will use *micro-segmentation* to partition the enterprise resources into very small *'micro-perimeters'*, each as small as possible, and each associated with one service or resource or type of data. It will act as a *border control* for access to each enterprise resource, and *it will evaluate trust on a per-session basis*. The NIST ZTA will effectively maintain an *Access Control System* in the SDN control-plane, to control the access to every critical resource. Users, applications, computers and devices will be authenticated often, using the *Access Control System*. Even access to printers, or the BE-IoT network, or a Wifi network, are resources that require approval.

A typical ZT implementation for an enterprise using the BE-IoT was described in [19]. The ZTA may use an overlay network in OSI layer-7 (the *application layer*), with an SDN control-plane. The *Access Control System* has a rule-based *Policy Engine* (PE), to control access to enterprise resources. The ZTA includes *Policy Enforcement Points* (PEPs), i.e., secured components which enforce the policies. The ZTA will use commercially-available *'Next Generation Firewalls'* to protect each resource (or small group of resources). It will continually: (a) perform authentication of users, applications, computers and devices, and perform authorization checks for access to resources; (b) monitor and measure the security-posture of all the resources, and upgrade resources if necessary; (c) monitor network infrastructure and traffic, to improve the security-posture of resources; (d) process network data, and use insights to improve policy creation and enforcement. According to NIST, most enterprises will use a combination of *Perimeter-Based* and *Zero Trust* security, in the future.

The proposed *Cybersecurity via Determinism* paradigm also provides hardware support in layers 3 and 4 for both *Perimeter-Based* and *Zero-Trust* security principles, in addition to providing deterministic communications to the next generation IoT. The proposed paradigm introduces an SDN control-plane with *Access Control Systems* into OSI layer 3, to rigorously control access to the new deterministic forwarding sub-layer 3a. The proposed paradigm significantly reduces the *attack surface* available to cyberattackers in layer 3, when using the IETF DetNet security model [121] (see Appendix D). The proposed paradigm allows for the creation of millions of isolated DVPNs in layers 3 and 4. An enterprise may create 1,000s of DVPNs, where each DVPN protects an enterprise resource (or a small group of related resources). Each DVPN uses *Quantum Safe* encryption to secure the communications from end-to-end. The proposed paradigm supports *Isolation Control*, which makes it extremely difficult for: (a) an external cyberattacker to compromise any remote DVPN; and (b) an internal cyberattacker in a compromised DVPN to compromise any other remote DVPNs. The proposed paradigm also supports a *guaranteed IDS*, where it continually monitors traffic in sub-layer 3a, and detects virtually all unauthorized communications in real-time. Finally, the proposed paradigm can save $10s of Billions annually, in reduced capital and energy costs, by exploiting a much more efficient and cost-effective forwarding sub-layer 3a which can operate at effectively 100% utilization [17].

E. ACCESS CONTROL SYSTEMS (ACS)

The BE-IoT has a serious vulnerability to cyberattacks in layer 3: *the inability to easily control which of the billions of computers in the BE-IoT can access critical resources*. The

US NIST has recommended the creation of *Access Control Systems* to improve cybersecurity of the BE-IoT [92], [94].

There are several types of *Access Control Systems*, including *Role-Based Access Control (RBAC)* systems [92], [93], and *Attribute Based Access Control (ABAC)* systems [94], [95]. In an ABAC system, there are subjects with attributes, objects with attributes, and rules relating the two. The ABAC controls the access of subjects to objects, by evaluating rules that examine the attributes of both the subject and object. ABAC allows for the evaluation of an arbitrarily large number of discrete logical inputs, when an access decision is made.

The simplest ACS, used in many older commercial products which support VPNs, requires each user to enter two attributes; (i) a user-ID (typically an email address), and (ii) a secret password. In the perimeter-based security model, the user then typically gains access to an enterprise LAN (*Local Area Network*), and many resources on that network. In this case, the deployment of *Quantum Safe VPNs* will add little security, if a malicious user can access a secured system simply by entering a stolen/compromised password.

1) MULTI-FACTOR AUTHENTICATION

To strengthen the model, the concept of *Multi-Factor Authentication (MFA)* has been proposed. In this model, a user requires additional attributes to gain access: (i) potentially a one-time security code, sent by text-message to the user's cell phone, (ii) a positive match for biometric data, i.e., a fingerprint, a face or a voice recognition system. The use of MFA will significantly lower the risk of internal cyberattackers (i.e., attackers who have access to a compromised password), as access to a critical system will now require a user to pass additional attribute checks.

One of the challenges of implementing an *Access Control System* for the BE-IoT is the lack of a logically centralized infrastructure to support the system. For example, there are expected to be about 30 billion devices connected to the BE-IoT by 2023, and there is no centralized infrastructure which tracks these devices, or the billions of traffic flows that they create. There is no centralized infrastructure to store and organize the millions of rules that will be needed. There is no distributed infrastructure to enforce the *Access Control* decisions.

In the proposed *Cybersecurity via Determinism* paradigm, a logically centralized SDN control-plane manages multiple SD-WANs. The centralized SDN control-plane can support a hierarchy of *Access Control Systems*. It can act as a repository for the millions of rules and attributes used in a large enterprise, and it can organize these rules hierarchically.

F. INTRUSION DETECTION SYSTEMS (IDS)

According to the US NIST, an *Intrusion Detection System (IDS)* will monitor the events that occur on a computer network or system, looking for signs of malware, i.e., violations of the policies of the *Access Control System*. An *Intrusion Prevention System (IPS)* will perform *Intrusion Detection*, and will attempt to stop any detected intrusions, i.e., by disabling the transport of certain traffic flows. The study of *Intrusion Detection and Prevention Systems (IDPS)* has a long history [102]–[107]. The US NIST has published recommendations for an IDPS in [112] and [113].

An IDS is typically placed at a strategic point, to observe the traffic within a network domain. An IDS will typically record relevant information, it will notify network administrators to take corrective action (i.e., by reconfiguring a firewall), and it will produce reports. According to the US NIST, an IDS will typically use 4 technologies [112], [113]. (i) In the *Network-Based* technology, the system will monitor network traffic, devices and protocol activity, to detect suspicious activities. (ii) In the *Wireless* technology, the system will monitor wireless network traffic and wireless protocol activity, to detect suspicious activities. (iii) In the *Network Behaviour Analysis) (NBA)* technology, the system will monitor traffic to identify threats that generate unusual traffic patterns over much of the network, such as DDOS attacks. (iv) In the *Host-Based* technology, the system will monitor the events occurring in relation to a single host, i.e., an IDS may compare the state of important system files on a single host, over time. If any system files are changed, then an intrusion may have occurred.

In an attempt to improve detection rates, the latest IDS or IDPS will typically employ *Deep Packet Inspection (DPI)*, to detect *signatures* of intrusions or malware. A signature is a specific pattern, i.e., it may be a byte sequence in the network traffic, or known sequence of malicious instructions used by malware. An IDS has a database of known signatures of cyberattacks, and it filters packets against the known signatures, to detect intrusions.

An IDS or IDPS can be characterized by the *Detection Rate*, *False Positive Rate*, *False Negative Rate*, and *Prevention Rate*. Unfortunately, these systems are prone to report *False Positives*, where legitimate traffic is reported as malware. These systems are only as accurate as the models they use to detect malware, and cyberattackers use many techniques to avoid detection. An IDS or IPS may also employ machine learning and deep learning technologies, to process large amounts of network traffic and history [108]–[111]. While these approaches are promising, they must process vast amounts of traffic, potentially exabytes of traffic per day collectively. They thus consume significant resources, i.e., large amounts of computational power, time for computations, energy and capital costs.

1) NEXT-GENERATION FIREWALLS

A traditional (first-generation) *Firewall* is a device, which uses a static set of rules to process traffic flows in real-time. A static rule may comprise an IP address and a socket port number, used in an IP packet header, that identify a potential cyberattacker. The *Next-Generation Firewalls* support much more analysis, such as *Deep Packet Inspection (DPI)*.

There are distinctions between a modern IDS, IPS and *Next-Generation Firewall*, as summarized by Huawei (please see web-article *Comparison and Differences between PDS*

vs IPS vs Firewall vs WAF, Aug. 11, 2021). The firewall is typically placed at the 'front side' of a network domain. It processes packets that arrive from BE-IoT in real-time, and it attempts to pro-actively block malicious packets from entering the network domain. An IDS is a passive device, that is usually placed after the firewall and within the network domain, to observe packets that have been admitted by the firewall. It alerts a network administrator to take action, when it detects a potential intrusion. (In some cases, an IDS may interact with a firewall to cause it to take action). An IDPS is a pro-active system; it will alert a network administrator when a potential intrusion is detected, and it will take action to prevent the intrusion (i.e., by blocking the malicious packets).

G. VIRTUAL PRIVATE NETWORKS (VPN)

1) BEST-EFFORT VPNs

The goal of a VPN is to provide secure communications between a limited number of entities, over an inherently insecure BE-IoT network. According to the ETSI, there are 2 main types of VPNs, *Site-to-Site* and *Remote Access* VPNs [69]. *Site-to-Site* VPNs will establish partially-encrypted connections called tunnels between 2 local private networks. These tunnels are usually are semi-permanent, i.e., they are established infrequently. (They are called partially-encrypted, because the packets headers are not encrypted.) *Remote Access* VPNs will establish a tunnel between a host computer and a remote private network. These tunnels are established frequently, and frequent authentication of a client (or device) is critical.

Existing VPNs that operate over the BE-IoT are hereafter called *BE-VPNs*, to distinguish them from the *Deterministic VPNs (DVPNs)* proposed in this paper. According to the ETSI, BE-VPNs attempt to achieve security using several protocols, operating at several layers in the OSI (*Open Systems Interconnection*) protocol stack [69]. The *Data Link* layer (layer 2) uses *MACsec (Media Access Control Security)*. The *Internet Protocol (IP)* layer (layer 3) uses *IPsec and IKE (Internet Key Exchange)*. The *Transport layer* (layer 4) uses *TLS (Transport Layer Security)*. The *Application* layer (layer 7) uses *SecureShell (SSH)*. (There are other protocols as well).

According to the ETSI, most of these BE-VPN protocols operate in a 2-step manner [69]. In step 1, initially most protocols accomplish entity authentication and key establishment, using PKC, while some protocols use pre-shared keys. Thereafter in step 2, all of these protocols accomplish data authentication and data confidentiality using encryption with symmetric (secret) keys, such as AES.

2) QUANTUM-SAFE BE-VPNs

The ETSI has outlined the requirements for *Quantum Safe* BE-VPNs in [69]. The first step in the establishment of BE-VPNs, which uses PKC for key exchange, must be revised to use the forthcoming NIST PQC standard. For example, software modules which previously performed PKC, must be replaced with new software modules to perform PQC.

3) THE SOFTWARE DIRECT DROP-IN REQUIREMENT

According to the ETSI, ideally the new PQC key exchange software can be organized as a software module with a *Direct Drop-in* capability for each BE-VPN protocol [69]. The new PQC software will likely use longer keys, and hence the computations will take more time. Hence, the direct drop-in software may affect the timing and memory requirements of the updated BE-VPN protocols.

Open source versions of software to support BE-VPNs are available, including: *OpenVPN, OpenTLS, OpenSSH, and OpenBSD*. It is expected that all of these open-source software systems will be upgraded to use *Quantum Safe* features once the US NIST PQC standard is confirmed.

4) WireGuard QUANTUM-SAFE BE-VPNs

One of the difficulties of *IPsec* and *OpenVPN* is the complexity of the software. It is estimated that these software packages contain over 500,000 lines of code. *WireGuard* is an alternative software package, which uses much less software to create BE-VPNs in the BE-IoT. *Wireguard* has been extended to use *Quantum Safe* key exchange. A recent study has examined two metrics, the time required and the number of BE-IP packets sent, for the protocol handshakes used in the key negotiation and exchange phase in *Wireguard* [114]. Unfortunately, these *Wireguard* BE-VPNs still use the insecure layer 3 IP, BGP and *Berkeley Sockets* protocols, and they are thus subject to many types of cyberattacks, i.e., DDOS attacks.

5) SOME EARLY TEST RESULTS

Several *Quantum Safe BE-VPN* software packages have been tested by industry. Verizon has tested a *Quantum-Safe BE-VPN* in August 2021. It uses a key exchange algorithm selected from the round 2 finalists in the US NIST PQC competition. Microsoft has also developed a *Quantum Safe BE-VPN*, which also uses a quantum safe key exchange algorithm selected from the finalists of the NIST PQC competition. Its source code is posted on Github. The company *Agilsec* has developed a *Quantum Safe BE-VPN* hardware module, which also uses a *Quantum Safe* key exchange algorithm selected from the finalists in the NIST PQC competition.

Studies on Post-Quantum authentication in TLS 1.3 have been recently completed [115], [116]. In reference [116], a few PQC signature algorithms were integrated into TLS 1.3, and TLS handshake latency and the effect on throughput were evaluated. Hardware accelerators were used to accelerate the PQC algorithms. BE-IoT networks with round-trip times varying between about 10 and 225 millisec were evaluated. On average, the TLS handshakes took about 110 millisec. The PQC signing times typically required < 15 millisec.

H. IETF ACTIVITIES IN DETERMINISTIC NETWORKS

1) DETERMINISM IN SMALL LAYER 2 NETWORKS

The IEEE 802.1 *Time Sensitive Networking* (TSN) standards [16] support the use of deterministic communications

in small layer-2 networks such as *Deterministic Ethernet*, for avionics and automotive applications. Multiple Ethernet transceivers share access to an Ethernet broadcast cable, through a repeating TDMA schedule. The repeating schedule reserves course-grain *'traffic-windows'* for several classes of traffic, including deterministic and best-effort traffic. To avoid collisions, all transceivers are tightly synchronized, typically to within 1 μsec of accuracy, using the TSN time synchronization technology. According to [16], this tight clock synchronization requires significant network bandwidth and it represents a significant load. Ethernet packets (frames) cannot be fragmented, and they are scheduled for transmission in real-time as they arrive at a transceiver, using a TSN 802.1Qbv *Time-Aware Shaper*. A large *Guard Interval* precedes each deterministic traffic-window, during which a new transmission of a lower priority frame cannot be initiated, as it may not have sufficient time to complete. These guard intervals represent idle periods, which reduce the throughput of the Ethernet broadcast network to noticeably less than 100%. (The 'fine-grain' scheduling of packets or fragments of packets can restore throughput in layer-3 to $\approx$ 100%, i.e., see [54].)

2) DETERMINISM IN LARGE LAYER-3 WANs

The IETF approved the *Deterministic Networking* (DetNet) group to explore the addition of deterministic communications to a private layer-3 *Wide-Area Network* (WAN) in 2015. *The goal is to create a Converged-WAN spanning large geographic distances, under the control of a single administrative entity, which supports both Deterministic and Best-Effort communications*.

The IETF has published several documents, including 4 RFCs:

- *Deterministic Networking Problem Statement* [118],
- *Deterministic Networking Architecture* [119],
- *Deterministic Networking Use Cases* [120],
- *Deterministic Networking Security Considerations* [121].

The IETF presents an *'Abstract Model'* of high-level requirements. The IETF does not specify detailed design implementations. The IETF expects that equipment vendors will implement their own proprietary solutions, to its high-level requirements. *For example, the IETF does not mandate; any routing algorithms, any fine-grain scheduling algorithms, any deterministic switch architecture, any router architecture, or the use of any pre-computed deterministic schedules of any kind.*

However, the IETF mandates that all layer-3 IP routers in the *Converged-WAN* are *tightly synchronized*, typically to within $\leq$ 1 μsec of accuracy, using the IEEE 802.1 TSN technology. The tight synchronization adds a significant load to the control traffic, and it is a potential security problem for a WAN which spans distances of 1000s of kilometers, as shown in Fig. 2b. In contrast, in the proposed *Deterministic IoT*, the D-switches do not require tight synchronization (see section III).

The IETF has proposed several 'Use-Cases' for deterministic networks [120], including (1) Professional Audio over the Internet, (2) Deterministic Radio Access Networks (D-RANs), (3) Deterministic Mobile Networks, and (4) Deterministic control for utilities such as the *Smart Power Grid* [122].

As shown in Fig. 1a, the existing power grid distributes vast amounts of power over a network of high-voltage transmission lines. According to the IETF, the ability to increase transmission line utilizations in the power grid by 10% can lead to potential capital cost savings of several Billions of dollars [122]. The future *Smart Power Grid* will require a very fast control system with end-to-end delays $\leq$ 5-10 milliseconds and with jitters $\leq$ 250 μsec [122]. The proposed *Deterministic IoT* spanning the continental USA shown in Fig. 2b can meet a 10 millisecond delay constraint over distances of about 2,000 kilometers, with a jitter is less than 16 μsec (see section VI).

3) DETERMINISTIC NETWORKING SECURITY CONSIDERATIONS

The IETF DetNet describes the most important security threats for its *Converged-WAN* in [121]. In this subsection, we outline some key similarities and differences between the IETF *Converged-WAN* and the proposed *Deterministic IoT*. Please see Appendix A for a summary of important acronyms used in this paper. Please see Appendix B for the design of a typical BE IP router. An IETF *Converged IP router* requires this functionality, plus the ability to handle deterministic traffic. Please see Appendix C for a summary of common cyberattacks in the BE-IoT. Please see Appendix D for a summary of the important security threats in the IETF *Converged-WAN*.

a: IETF CONVERGED-WAN NETWORK

The IETF Detnet proposes a private layer-3 *Converged-WAN*, under the control of a single administrative entity, which supports both Deterministic and Best-Effort traffic. IETF does not address adding deterministic communications in the general IoT, which consists of a large number of autonomous domains [119]. Existing BE IP routers in its layer-3 WAN will have to be replaced with new *Converged IP routers*, an expensive proposition. In contrast, the proposed *Deterministic IoT* adds deterministic communications and strong cybersecurity to the general IoT. It introduces a new sublayer 3a, comprising many deterministic SD-WANs, each consisting of many simple and secure low-cost D-switches, which provide deterministic services to layer-3.

b: IETF TRUSTED COMPONENTS

The IETF DetNet identifies *'trusted components'* (i.e., IP routers and control-plane), and it assumes that these function correctly and are trustworthy, when analysing cyberattacks (unless otherwise stated). It leaves the detailed design of the trusted components to the equipment vendors. The IETF acknowledges that 'perfect security' and

trustworthiness may be technologically or financially unfeasible, and that vendors may have to make design tradeoffs that affect security. In view of the US NIST *Zero Trust Architecture* described ahead, we use the term *'secured components'* to refer to the IETF *'trusted components'*. We also assume that the secured components in the proposed *Deterministic IoT* (i.e., D-switches and the SDN control-plane) function correctly and are trustworthy, when analysing cyberattacks (unless otherwise stated). Please see section III for design discussions.

c: INTERNAL VS. EXTERNAL CYBERATTACKS

The IETF DetNet distinguishes between *'Internal'* and *'External'* cyberattacks. An internal cyberattacker already has access to a secured computer in the system, perhaps by human error, or it may have access to a secret password or encryption key. An external cyberattacker does not have access to a secured computer, password or encryption key. In this paper, we address both types of cyberattacks.

d: PRECOMPUTED DETERMINISTIC PERIODIC SCHEDULES

As stated earlier, the IETF DetNet does not mandate the use of any pre-computed deterministic schedules. The IETF DetNet exploits the IEEE TSN standards, and it acknowledges that tightly-synchronized *'repeating schedules'* may be useful [119]. A vendor's implementation of the DetNet *Converged-WAN* could mimic the implementation of *Deterministic Ethernet*. It could use: (a) tightly-synchronized repeating *Time Division Multiplexing* (TDM) schedules with fairly large course-grain *'traffic-windows'* for Deterministic and BE traffic on each link; (b) a IEEE TSN 802.1Qbv *Time-Aware Shaper* (TAS) to schedule packets into the traffic-windows in real-time; (c) large *Guard Intervals* preceding each deterministic traffic-window, to prevent low-priority traffic from interfering with deterministic traffic, which lowers the link utilization.

In contrast, in the proposed *Deterministic IoT*, an SDN control-plane will pre-compute *'deterministic periodic schedules' (D-schedules)* for the D-switches in sub-layer 3a, for each domain. The D-schedules reserve time-slots for packet transmissions on a 'fine-grain' basis, where each packet has a guaranteed reservation. (In general, the fine-grain scheduling of packets through a switch to achieve maximum throughput and low-jitter is a well-known NP-hard problem. Hence, fine-grain scheduling of packets is often not addressed in standards.) Our D-schedules can reduce buffer sizes by a factor of 1,000+ times, they achieve bounded delay and jitter guarantees, and they can achieve very high link utilizations in sub-layer 3a ($\leq$ 100%), with ultra-low end-to-end latencies [20], [60].

e: FLOW ISOLATION

The IETF DetNet states that ideally all DetNet-flows will be isolated from one another, and from BE-flows. However, it leaves the detailed design to equipment vendors. It acknowledges that it may be technologically difficult or financially infeasible to isolate DetNet-flows from one another, or from BE-flows, in all cases. *Hence, a DetNet-flow might not receive true deterministic service.* In contrast, the D-flows in the proposed *Deterministic IoT* are strictly isolated, by the use of pre-computed D-schedules, and they receive true deterministic service.

f: IPsec SECURITY WEAKNESS

The IETF DetNet uses *IPsec* for security in layer 3 and 4. However, *IPsec* has several significant problems (as stated in section 1). Specifically;

- (i) *IPsec* packet headers are not encrypted. The *IPsec* packet payload can be encrypted, but the packet headers are read at intermediate IP routers, to identify DetNet flows and to make layer-3 routing decisions. Cyberattackers can thus manipulate *IPsec* packet headers;
- (ii) Secured computers within a BE-VPN are subject to cyberattacks from other computers around the world; *They are not isolated*;
- (iii) Compromised computers within a BE-VPN are free to initiate cyberattacks against other computers around the world. *They are not isolated, and the threat is not contained*;
- (iv) *IPsec* runs on the BE-IoT, and will suffer from congestion, interference and DDOS attacks.

In contrast, in DVPNs; (i) packets are entirely encrypted from end-to-end; (ii) D-switches do not read packet headers; (iii) all cyberattacks which manipulate *IPsec* packet headers are eliminated, and (iv) *network partitions* are strictly isolated from one another (to contain cyberattacks).

g: IETF UNTRUSTED COMPONENTS

The IETF DetNet acknowledges that in its *Converged-WAN* model, a DetNet-flow may pass through 'untrusted' components. An untrusted component may be a 'middle box' which performs *Network Address Translation* (NAT) or load-balancing [21]. Such untrusted components may be compromised by a cyberattacker, and they may initiate a cyberattack by modifying *IPsec* packet headers. In contrast, the proposed *Deterministic IoT* introduces a sub-layer 3a of secured D-switches, and the D-flows are only transported by secured components. Sub-layer 3a does not require any middle-boxes, i.e., NAT middle-boxes.

I. THE INDUSTRIAL AND TACTILE IoT AND INDUSTRY 4

General Electric (GE) developed the *Industrial Internet (of Things)* (IIoT) architecture to recognize the growing importance of interconnecting industrial machines and factories for *Industrial Automation* [1], [2]. GE argues that the shift to *Industrial Automation* will impact world economic activity on the same scale as the *Industrial Revolution* of the 19th century. According to the *World Economic Forum*, the world is undergoing a *4th Industrial Revolution*, often referred to as *Industry 4.0*.

According to GE, this revolution will increase world GDP by $15 Trillion by 2030, by reducing manufacturing costs. GE also estimates that the next-generation IIoT may control about $82 Trillion of global GDP by 2030, representing about 50% of the world's GDP. In March 2014, several companies (GE, Cisco, AT&T, IBM and Intel) formed the *Industrial Internet Consortium* (IIC), and by 2016 the consortium included over 250 companies, indicating strong industrial adoption.

The IIC has presented several draft *Industrial Internet Reference Architectures*, the latest in June 2019. Like the IETF, the IIC presents an *'Abstract Model'* of high-level requirements, and it does not specify detailed design implementations. The reader is referred to IIC web-site for additional information (www.iiconsortium.org).

The ITU initiated a project on the *Tactile Internet*, to describe a next-generation Internet network with low end-to-end latency (typically ≤ 1 millisecond), and high availability, reliability and security in 2014, for applications including smart transportation systems and industrial automation.

This paper argues that the proposed *Deterministic IoT* meets the key requirements of the IIC *Industrial IoT* and the ITU *Tactile IoT* projects; ultra-low latency, with very high availability, reliability and security [21]. The proposed *Industrial and Tactile Deterministic IoT* thus meets the demanding requirements of the *4th Industrial Revolution* i.e., Industry 4.0. It can also achieve exceptionally strong cybersecurity, and save $10s of Billions in reduced capital, energy and operational costs each year.

J. FPGAs WITH SILICON-PHOTONICS IO

A key feature of the proposed paradigm is the cost reductions that are possible, by using relatively simple D-switches in sub-layer 3a. A simple D-switch can be created using an *Application Specific Integrated Circuit (ASIC)*, or an *Field Programmable Gate Array (FPGA)*. D-switches are well-suited for fabrication using ASICs or FPGAs, since they offer a dramatic reduction in complexity, compared to a layer-3 IP router: (a) The tasks of routing and scheduling of D-flows have been removed from the D-switches, and have been migrated to the SDN control-plane; (b) The deterministic communications can reduce buffer sizes by a factor of 1,000+ times [17], [20]–[22]; (c) D-switches do not need Gigabytes of high-speed RAM to store insecure layer-3 routing tables; (d) D-switches do not need a processor or a Linux operating system running the insecure *Berkeley Sockets* software to implement insecure layer-3 protocols, i.e., ICMP, BGP and IGP. D-switches are also much easier to secure, compared to a layer-3 IP router. *One-Time-Programmable* FPGAs can also be used, as their functionality cannot be modified.

Furthermore, the microelectronics industry is currently developing ASICs and FPGAs which are integrated with Silicon Photonics optical transceivers [21], [126]. ASICS and FPGAs with 100s of Gbps of optical IO bandwidth may be commercially available within a decade. According to [21], a simple D-switch realized on a single FPGA with optical IO could achieve a capacity of a Terabit per second, while dissipating about 100 Watts of power, within a decade. These integrated photonic D-switches can be used to provide an energy-efficient forwarding sub-layer-3a, which can embed millions of isolated DVPNs, as shown in Fig. 2. The proposed paradigm can thus dramatically reduce capital costs and energy costs, by adding relatively simple D-switches in sub-layer 3a, as opposed to adding relatively complex Converged IP routers in layer 3.

III. THE DETERMINISTIC IoT–KEY FEATURES

A. SECURED COMPONENTS

The proposed *Deterministic IoT* supports a new forwarding sub-layer 3a, which implements multiple deterministic SD-WANs, and comprises 4 types of secured components:

- Deterministic Sources (D-sources)
- Deterministic Sinks (D-sinks)
- Deterministic Packet Switches (D-switches)
- The SDN Control-Plane

The SDN Control-Plane itself can be organized hierarchically into two types of *Access Control Systems*:

- Global Access Controller
- Enterprise Access Controllers

An *Enterprise Access Controller* can be further sub-divided into many smaller *Access Control Systems*:

- Employee Access Controllers
- SD-WAN Access Controllers
- Database Access Controllers
- DVPN Access Controllers

These controllers will provide the repository for the very large number of rules and attributes an enterprise uses in its ZTA/ACS.

The *Global Access Controller* manages the D-flows between the multiple SD-DWANs that comprise the *Deterministic IoT*. It stores and implements all the rules and attributes needed to maintain secure D-flows between the deterministic SD-WANs, managed by different enterprises. Each *Enterprise Access Controller* manages the access to all the resources of one enterprise. It can store and implement the large number of rules and attributes needed for a ZTA/ACS system which manages: (i) enterprise employees, (ii) the enterprise SD-DWAN, (iii) enterprise resources such as data-bases, and (vi) enterprise DVPNs. (Other controllers can be added, i.e., *Wireless Access Controllers*). Hereafter, this paper will focus on a single SD-WAN and its *DVPN Access Controllers*.

B. THE DETERMINISTIC TRANSCEIVERS

The D-sources and D-sinks enforce *Access Control*, *Rate Control* and *Isolation Control*. A computer system can only access the *Deterministic IoT* using these secured components. These components act as the *Policy Enforcement Points (PEPs)* in the US NIST *Zero Trust Architecture* [19].

A D-source retains a list of authorized D-flows for which it transmits data. For each D-flow, it transmits data at a

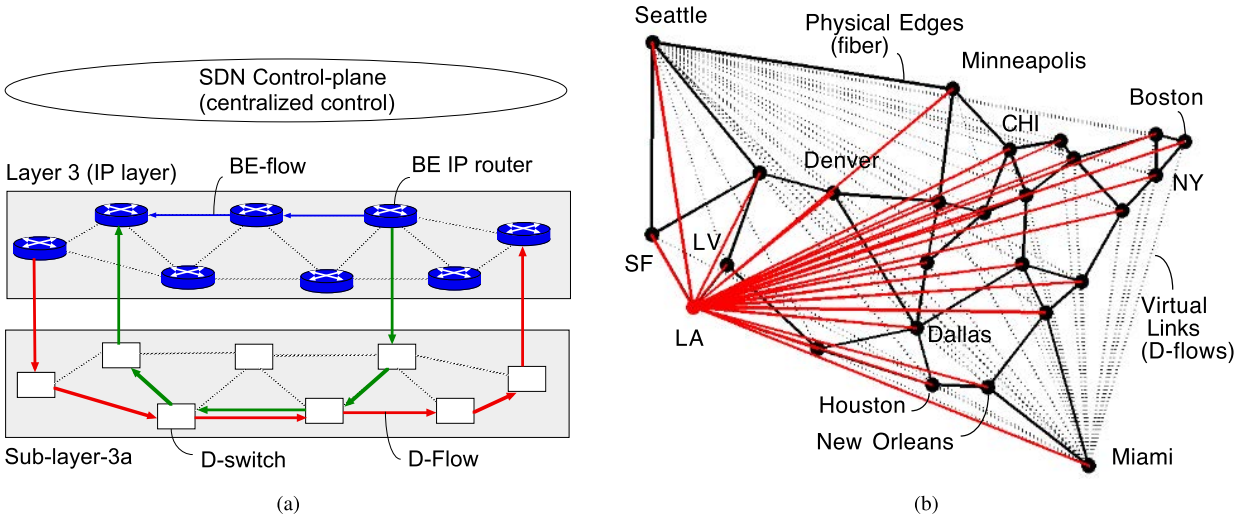

FIGURE 2. (a) Layer-3 IoT with BE IP routers, and sub-layer-3a with D-switches. (b) A Deterministic IoT spanning the USA with 3 DVPNs, originating at Los Angeles (LA), Seattle and Miami. Solid black lines denote fiber-optic edges. Dotted lines denote virtual 'point-to-point' links (i.e., D-flows). A DVPN with fully-encrypted D-flows from LA to every other major city are shown with bold red lines.

Guaranteed-Rate, and it has a secret encryption key and secret passwords (described ahead). The D-source communicates with the SDN control-plane over a D-flow, and it receives a D-schedule from the control-plane, which specifies the time intervals in which it may transmit data, for each D-flow. A D-sink retains a list of authorized D-flows for which it receives data. For each D-flow, it receives data at a Guaranteed-Rate, and it has a secret decryption key (usually the keys are symmetric) and secret passwords. The D-sink communicates with the SDN control-plane over a D-flow, and it receives a D-schedule from the control-plane, which specifies the time intervals in which it may receive data, for each D-flow.

IPv4 and IPv6 packets can contain up to 64 Kbytes. The D-sources can also fragment large IP packets into smaller fragments (i.e., 1 Kbytes), to minimize delays and to maximize throughput in sub-layer 3a. The fragments can be transmitted to fill the reserved transmission intervals, and the D-sinks can re-assemble the large IP packets. (Each fragment must pass the *Authorization-Check* explained ahead. It will also require a sequence number and a CRC checksum for error detection.)

C. ISOLATED NETWORK PARTITIONS AND DVPNs

Fig. 2a illustrates IP routers in layer 3 (the 'Internet Protocol' or IP layer), and D-switches in sub-layer 3a. Fig. 2b illustrates an SD-WAN spanning the continental USA, with 26 nodes (cities) and 82 edges. In Fig. 2b, the bold black lines represent *physical edges* (i.e., fiber-optic links) between cities, and the dotted lines represent *virtual 'point-to-point' edges* (i.e., D-flows) between cities. The existence of D-switches in sub-layer 3a will change the network topology seen by an IP router in layer 3, thus making the IP routers much more efficient. An IP router in layer-3 will view a D-flow as a dedicated point-to-point fiber-optic link with a deterministic (or guaranteed) data-rate between two remote cities. Packets transmitted on a D-flow will bypass several intermediate IP routers in layer-3, as they will pass through the D-switches instead. Therefore, the D-flows are drawn as dotted point-to-point lines directly connecting 2 cities, in Fig. 2b.

A *Network Partition* is a collection of D-flows, which are typically under the control of a single administrative entity, i.e., an enterprise. *Network Partitions* are isolated, i.e., they are completely independent of one another. The traffic within one *network partition* cannot interfere with the traffic from another network partition. The traffic within a *network partition* can: (a) remain unencrypted, or (b) it can be completely encrypted. In the latter case, the encrypted *network partition* is called a *Deterministic Virtual Private Network* (DVPN).

In Fig. 2b, three DVPNs are embedded into the network. Three cities, Seattle, Miami and Los Angeles (LA), each have a DVPN which interconnects the specified city to every other city. Specifically, each of these cities has a DVPN with 25 D-flows, with one D-flow to every other city in the network. A DVPN with 25 D-flows originating at LA and directed to every other city is shown by the bold red lines.

D. THE DVPN ACCESS CONTROLLERS

Under the proposed paradigm, each enterprise will have a logically centralized controller (the *DVPN Access Controller*) for management and control of all of its DVPNs. This controller is also a secured component, and it can be viewed as a sub-component of the SDN control-plane. (In practice, the DVPN-controller may be a distributed software system residing in one or more datacenters.) This DVPN-controller retains a list of D-sources and D-sinks which are authorized for use by the enterprise. It will also contain the attributes of

these components, and the rules used to enforce the ZTA and ACS.

This controller can request a new D-flow, between authorized D-sources and D-sinks, with a deterministic rate, with a requested reliability and security level, and with a maximum delay and jitter bound, from the SDN control-plane. If the request can be granted, the SDN control-plane will acknowledge the creation of the new D-flow. It will update the relevant secured components (D-source, D-switches, and D-sink), with the necessary D-schedules and secret symmetric encryption keys (and secret passwords explained ahead). The secured computers within the DVPN can then communicate over this D-flow. If the SDN control-plane is unable to satisfy the request, it will be denied. (The DVPN-controller can also implement *User Authentication* at each secured computer within a DVPN.)

Note that a secured computer within a DVPN cannot create or modify a D-flow unilaterally, as all requests to create or modify D-flows must originate from the DVPN controller, and must be approved by the SDN control-plane (i.e., the *Enterprise Access Controller* and *DVPN Controller*). A secured computer within a DVPN cannot read any secret encryption keys from a secured component,

E. AUTHORIZATION-CHECK

Every packet in a DVPN must pass an *Authorization-Check*, after it is decrypted at a D-sink. Each packet (or fragment of a large IP packet) has an *Authorization-Token* with A bits (A $\approx$ 256 bits), that identifies the packet as valid. The *Authorization-Check* provides a second level of protection for packets in a DVPN, beyond encryption. This *Authorization Check* implements a *Policy-Enforcement Point* in the ZTA/ACS, and it implements the guaranteed IDS.

We describe a simple *Authorization-Check*, given that the packet is already encrypted with a *Quantum Safe* encryption scheme. Assume the SDN control-plane initializes the the D-source and the D-sink of each authorized D-flow with a sequence of R secret passwords (i.e., random numbers), each with $\approx$ 176 bits. For a lower level of security, R $\approx$ 8, while for a very high level of security, R $\approx$ 1,024. Each D-source also has a counter with $\approx$ 64 bits which records its *'relative-time'*, i.e., the time elapsed since the device was last 'reset'. Each tick of *relative-time* could represent 20 nanoseconds. (A 64-bit counter could thus count for over 1,000 years.) The *Authorization-Token* consists of a relative-time stamp, plus a 16-bit sequence number, plus the next secret password in the sequence. The R secret passwords may be re-used in a circular manner. (The sequence number is necessary when large IP packets are fragmented into smaller units before transmission.) The token may optionally be encrypted with another long *Quantum Safe* encryption key, for additional security.

For a malicious packet to pass the *Authorization-Check*, an external cyberattacker must perform 5 steps: (a) find greater than R (denoted R+) encrypted packets that belong to one D-flow (which is extremely challenging given that 1,000s of encrypted D-flows may share one fiber, so it will be very difficult to associate encrypted packets with an individual D-flow); (b) successfully decrypt these R+ packets; (c) recover the *Authorization-Tokens* in these packets, and recover the R+ secret passwords; (d) successfully predict the correct *Authorization-Token* for a future packet belonging to the D-flow; and (e) overwrite a future legitimate packet for the D-flow with the malicious packet, at the right time, on the right fiber. The probability of decrypting even one encrypted packet can be made arbitrarily small by using longer *Quantum Safe* encryption keys, which are impervious to attacks by *Quantum Computers* using existing quantum algorithms. The following property summarizes the situation.

Property 1 (Malicious Packets and Authorization-Checks): The probability that any malicious packet generated by an external cyberattacker can pass an *Authorization-Check* at a secured component can be made arbitrarily small by using longer *Quantum Safe* encryption keys, which are impervious to attacks by *Quantum Computers* using existing quantum algorithms.

This *Authorization-Check* also detects *Replay Attacks*, where a cyberattacker observes and records a prior legitimate encrypted packet transmission, and re-inserts the old legitimate transmission as a malicious packet, while overwriting a newer legitimate transmission. In this case, the cyberattacker does not have to actually decrypt any packets. Such a packet will fail the *Authorization-Check*, for 3 reasons: (a) the relative-time stamp will be invalid, (b) the sequence number will be invalid, and (c) the secret password will be incorrect.

F. CYBERSECURITY OF THE SECURED COMPONENTS

The cybersecurity of the D-sources, D-sinks and D-switches can be ensured by synthesizing them directly into hardware, i.e., ASICs or FPGAs, rather than software. The hardware functionality must be *immutable* (i.e., it cannot be changed). These components cannot have a processor which uses a linux-based operating system, or use the *Berkeley Sockets* software for socket communications, which introduce security threats. The hardware design should undergo *Formal Verification* of correctness, using theorem-proving technologies. These components only accept heavily-encrypted commands from the SDN control-plane or DVPN controller, which are received on a special DVPN reserved for management and control. Over the next decade, these components will be well-suited for fabrication using the next-generation of FPGAs or ASICs, which are expected to include optical transceivers.

In order to compromise any secured component, an external cyberattacker would have to generate malicious encrypted packets that appear to be from the SDN control-plane, and have these packets pass the *Authorization-Check* at the secured component. The cyberattacker could then update the D-schedules, or add or remove D-flows, or update encryption/decryption keys. However, by *Property 1* stated earlier, the probability of passing the *Authorization-Check* can be

made arbitrarily small, by using longer encryption keys. The following property summarizes the situation.

Property 2 (Cybersecurity of the Secured Components-1): The probability that any secured component can be compromised by an external cyberattacker can be made arbitrarily small, by using longer *Quantum Safe* encryption keys.

Furthermore, there is a second level of protection for property 2. Even if an external cyberattacker could compromise one secured component, i.e., a D-source, the cyberattack would be detected by either the D-switches or a D-sink. For example, suppose a cyberattacker compromises a D-source, to change the deterministic rate of a D-flow, or to create a new D-flow to a new destination. This attack will be detected by the D-switches traversed by the D-flow, and by the D-sink which receives the D-flow. The SDN control-plane maintains the D-schedules in these secured components in consistent states, which an external cyberattacker cannot do, unless the cyberattacker manages to compromise multiple secured components simultaneously, which becomes even more difficult to achieve. The following property summarizes the situation.

Property 3 (Cybersecurity of the Secured Components-2): If a D-source is ever compromised by an external cyberattacker, to adjust any deterministic rate of any D-flows, the cyberattack will be detectable by some other secured component, as the D-schedules will be inconsistent.

Consider the case where an internal cyberattacker compromises a secured computer within a DVPN. The internal cyberattacker will be unable to create or modify any D-flows unilaterally, as only the SDN control-plane can perform these functions. The internal cyberattacker will not gain access to any secret encryption keys or passwords, as the hardware of the secured components will not allow this. The internal cyberattacker will be unable to communicate with any other computers in remote DVPNs. The internal cyberattacker will only be able to communicate with other computers within its DVPN over the existing D-flows. The use of isolated DVPNs will therefore *contain the damage* caused by this internal cyberattacker to the DVPN to which it belongs. The following property applies.

Property 4 (Cybersecurity of a Computer Within a DVPN): If an internal cyberattacker manages to compromise a secured computer within a DVPN, the internal cyberattacker will have access to existing D-flows of the DVPN. However, the internal cyberattacker will not be able to create or modify D-flows, it will not gain access to the secret encryption keys or authorization passwords, and it will not be able to communicate with remote DVPNs. Any damage will be contained to the compromised DVPN. (Please see section V for an expanded discussion.)

G. THE DETERMINISTIC PACKET SWITCH - CIOQ ARCHITECTURE

The *Combined Input and Output Queues* (CIOQ) best-effort switch is used in many commercial BE IP routers. A D-switch with *Combined Input and Output Queues* is shown in Fig. 3. The D-switch has N input ports (IPs) and N output ports

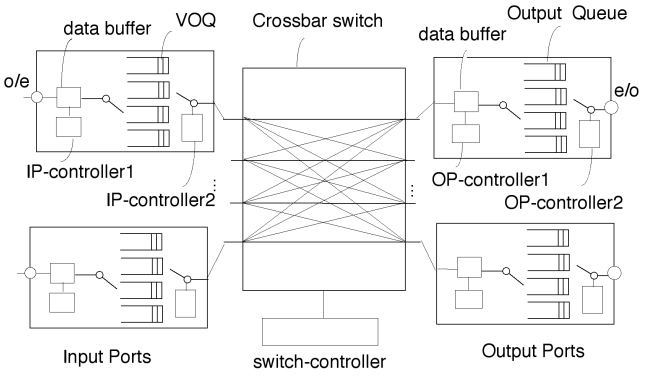

FIGURE 3. A CIOQ D-switch with 5 deterministic controllers.

(OPs), each incident to an optical fiber. It has an unbuffered NxN crossbar switch to provide connections between the input and output ports.

1) VIRTUAL OUTPUT QUEUES

Each input port has N *Virtual Output Queues* (*VOQs*), where $VOQ(j, k)$ buffers packets which arrive at input port j and depart on output port k (for $j \in [1 \ldots N]$ and $k \in [1 \ldots N]$). Each output port has N output queues (*OQs*), where $OQ(j, k)$ buffers data which arrives at input port j and departs on output port k.

2) DETERMINISTIC SWITCH CONTROLLERS

In the proposed D-switches, the routing and scheduling of packets is performed in advance in the SDN control-plane, for a dramatic reduction of hardware and power consumption. We assume a discrete-time D-switch, which transfers data in time-slots, with F time-slots in a scheduling-frame.

There are several types of D-switches, and several types of queueing strategies, which differ in hardware complexity and levels of control of the D-flows [20]. The D-switch in Fig. 3 allows the control-plane to exert a high level of control of D-flows. The D-switch in Fig. 3 has 5 controllers: *IP-controller1* directs an incoming packet into a *VOQ*. *IP-controller2* selects data from a *VOQ*, to be forwarded to an output port. *OP-controller1* directs data arriving at an output port to an *OQ*. *OP-controller2* selects data to transmit onto a fiber, from an *OQ*. The *switch-controller* connects input ports to output ports.

In Fig. 3, all 5 controllers use D-schedules, which have been pre-computed by the SDN control-plane. No packet headers are processed to make layer-3 routing or scheduling decisions. (The D-schedules are effectively high-speed lookup-tables.) As a result, the D-switches do not need: (a) many Gigabytes of expensive, high-speed RAM used for layer-3 routing tables; (b) a processor or a Linux operating system running the insecure *Berkeley Sockets* software to implement insecure layer-3 protocols, such as ICMP, BGP and IGP, which add significant costs and security threats.

The D-switch can be controlled by a master-controller, which receives commands from the SDN control-plane. The

TABLE 1. Security considerations.

	Key Length	Brute Force Search	Grover Search
AES-128	128	$O(2^{128})$ $\approx O(10^{38})$	$O(2^{64})$ $\approx O(10^{19})$
AES-256	2x128 = 256	$O(2^{256})$ $\approx O(10^{77})$	$O(2^{128})$ $\approx O(10^{38})$
ext-AES-512	4x128 = 512	$O(2^{512})$ $\approx O(10^{154})$	$O(2^{256})$ $\approx O(10^{77})$
ext-AES-1024	8x128 = 1,024	$O(2^{1024})$ $\approx O(10^{308})$	$O(2^{512})$ $\approx O(10^{154})$
RIJNDAEL (4x8)	256	$O(2^{256})$ $\approx O(10^{77})$	$O(2^{128})$ $\approx O(10^{38})$
ext-RIJNDAEL (4x16)	512	$O(2^{512})$ $\approx O(10^{154})$	$O(2^{256})$ $\approx O(10^{77})$
ext-RIJNDAEL (4x16)	8x512 = 4,096	$O(2^{4,096})$ $\approx O(10^{1,233})$	$O(2^{2,048})$ $\approx O(10^{616})$
ext-RIJNDAEL (4x64)	2,048	$O(2^{2,048})$ $\approx O(10^{616})$	$O(2^{1,024})$ $\approx O(10^{308})$
ext-RIJNDAEL (4x64)	8x2,048 =16,384	$O(2^{16,384})$ $\approx O(10^{4,932})$	$O(2^{8,192})$ $\approx O(10^{2,466})$

master-controller in each D-switch can exchange control packets with the SDN control-plane, using *Quantum Safe* encrypted communications over a dedicated DVPN reserved for network management and control.

3) SYNCHRONIZATION

The D-switches could be tightly synchronized using the IEEE 802.1 TSN technology, but a more secure method is proposed next. Each D-switch is 'loosely synchronized' with its neighbors, and receives a *Start-of-Frame* (SOF) signal (or encrypted packet) from each neighbor, which indicates when the next scheduling frame from that neighbor will start. As described in section VI, a D-switch may receive a SOF signal (or packet) from each neighbor roughly once every millisecond. Note that the IETF states that its *Converged-WAN* could use repeating TDM schedules with neighbors [119]. The IETF *Converged IP routers* are also tightly synchronized in time using the IEEE 802.1 TSN technology, typically to within 1 μsec. Thus, if the IETF *Converged-WAN* uses repeating schedules, then it will also require that: (a) all repeating TDM schedules start at the same time (i.e., to within 1 μsec of accuracy), or (b) a SOF signal is sent from each neighbor to denote when the repeating TDM schedule starts. It will also require that control-signals are sent from each neighbor to identify the start-times and end-times of all course-grained traffic-windows, typically to within 1 μsec of accuracy. The IETF *Converged-WAN* will thus have a considerably higher overhead of control-packets to maintain tight synchronization over distances of 1,000s of kilometers, which increases the threat of cyberattacks (see Appendix C).

H. QUANTUM SAFE END-TO-END ENCRYPTION

This section illustrates how long *Quantum Resistant* or *Quantum Safe* encryption keys can be used to strengthen the cybersecurity of a DVPN. We focus on the *Quantum Safe Advanced Encryption Standard* (AES). The IETF has referred to AES as the *'Gold Standard'* in private key encryption [101]. It is well-known, it is used around the world, and its security has been extensively studied, but any other *Quantum Safe* scheme can be used. For example, IETF RFC 7289 proposes the *Chacha20* cipher as an alternative to AES encryption, and the *Poly1305* authenticator as an alternative to traditional digital signature algorithms [101]. *Chacha20* can be faster than AES, on devices that do not have hardware to accelerate AES computations.

The US NIST adopted the AES in 2001, to provide strong encryption for potentially millions of IoT devices, some with limited processors and limited memory, such as smart cards [98]. AES will encrypt a relatively small array of 4 $\times$ 4 bytes (with 128 bits).

The AES algorithm is a subset of a more general encryption algorithm called *RIJNDAEL* [99], [100]. The creators of *RIJNDAEL* acknowledged that it can be extended to handle larger data and encryption key sizes, but that there was no need for such extensions at the time (in the year 2001).

The AES/RIJNDAEL algorithm consists of multiple iterations called *rounds*, each comprising 4 basic steps:

1) *AddRoundKey* - XOR the data with a round key
2) *SubBytes* - substitute bytes using a lookup table
3) *ShiftRows*, i.e., shift row i left by i bytes ($1 \leq i \leq 3$)
4) *MixColumns* - i.e., each column is computed using linear combination of some previous columns

The final NIST standard specifies 3 versions of AES, which all encrypt a small block size (16 bytes of data). AES-128, AES-196 and AES-256 use encryption keys with 128, 196 and 256 bits respectively, for stronger security. They also use 10, 12 and 14 rounds of computation respectively, for stronger security. AES is an example of a *Key-Alternating-Cipher* with multiple rounds of computation, in which the data is initially ex-ored with an encryption key before the first round, and in which the data is ex-ored with another key called the *round key* during each subsequent round. Thus, AES-128 uses one independent 128-bit encryption key initially, plus a dependent *round key* for each round of computation. The *round keys* are generated using logical operations on prior keys [98].

Consider the basic AES-128 algorithm with 10 rounds of computation. There is one independent 128-bit encryption key, and 10 dependent round keys. The lack of independence of the round keys used is considered a weakness in AES.

The AES algorithm can be extended to handle longer encryption keys, simply by using more independent 128-bit encryption keys. Consider an extension of the AES-128 algorithm, to use 4 independent 128-bit encryption keys, for a combined encryption key length of 512 bits. Denote this extended algorithm the ext-AES-512 algorithm in Table 1. According to [99], [100], the number of rounds should increase by 4 for each increment of 128 bits added to the encryption key. Hence, the number of rounds should increase by ≈ 12. A key observation is that the length of the independent encryption keys can be made very large.

One weakness of AES is the small amount of data (16 bytes) that is encrypted. Consider encrypting a larger data array of 4×8 bytes (with 256 bits), using the RIJNDAEL algorithm [99], [100]. Denote this algorithm as RIJNDAEL(4×8)-256 in Table 1.

Consider encrypting an even larger data array of 4×16 bytes (with 512 bits), using an extended RIJNDAEL algorithm. Denote this algorithm as ext-RIJNDAEL(4×16)-512 in Table 1. In this case, the initial encryption key and all round keys will have 512 bits. (The larger data and encryption key sizes will increase the number of rounds accordingly.)

For stronger security, consider extending the algorithm to use 8 truly independent 512-bit keys, for a combined encryption key length of 4,096 bits. Denote this algorithm as ext-RIJNDAEL(4×16)-4,096 in Table 1.

In Table 1, column 3 illustrates the number of possibilities a traditional computer must search in a successful attack using *Brute Force Search*, reflecting the complexity in cracking the extended AES or RIJNDAEL algorithms. In Table 1, column 4 illustrates the number of possibilities a *Quantum Computer* must search in a successful attack using *Grover's* algorithm.

It is important to note the astronomical difficulties of cracking these algorithms. The life of the universe is about 13.5 billion years, or $\approx 10^{21}$ seconds. Researchers estimate that AES-256 can withstand cyberattacks for the foreseeable future, given that the best quantum search algorithm results in only quadratic (rather than exponential) speedup. In other words, even *Quantum Computers* cannot perform a successful attack on AES-256, which requires 1,000s of qubits and about 10^{38} quantum queries of the AES algorithm using *Grover's* search.

According to [76], [77], the minimum time for the simplest quantum operation in a superconducting quantum computing platform is $\approx$ 10 nanoseconds. The minimum time for the evaluation of a quantum query of the AES-256 algorithm will be several orders of magnitude larger. Thus, a lower bound of time needed to crack AES-256 can be estimated at $10^{38} \times 10^{-8} \approx 10^{30}$ seconds, which represents billions of times of the life of the universe.

The extended AES and RIJNDAEL private key encryption algorithms presented in Table 1, with very long keys, can thus offer exceptionally strong encryption, which is resistant to attacks by *Quantum Computers* using known quantum algorithms.

IV. THE SDN CONTROL-PLANE

Fig. 4 and 5 illustrates a flow-chart for the SDN control-plane. The SDN control-plane will compute 5 D-schedules for each D-switch, which allow an encrypted packet of a D-flow to traverse the network with a unique feature: the D-switches do not examine packet headers to make layer-3 routing decisions. As a result, packets in D-flows can remain encrypted from end-to-end, resulting in improved privacy and cybersecurity. Specifically, these D-schedules define the time intervals when authorized packet transmissions may occur, on every fiber in the network.

The flow-chart is rather intricate and contains many steps. A reader unfamiliar with switch design may wish to skip this section on the first read through the paper, and revisit this section afterwards.

The following notation will be used in Fig. 4. The variable s will denote a D-switch, for $s \in [1 \ldots S]$. Let every D-switch have N input ports and N output ports. The variable j will denote an input port, for $j \in [1 \ldots N]$.. The variable k will denote an output port, for $k \in [1 \ldots N]$. The variable f will denote a D-flow, for $f \in [1 \ldots G]$. (For the purpose of scheduling, a traffic class (i.e., a *DiffServ* traffic class) with a guaranteed data-rate is treated as D-flow with a guaranteed data-rate.) The variable F will denote the length of a scheduling-frame, in time-slots.

1) MAX-FLOW MIN-COST ROUTING OF D-FLOWS

The SDN control-plane has a global view of the network. In Fig. 4 box 2, the SDN control-plane will route every D-flow along a fixed path, from a D-source to a D-sink. The routing algorithm ensures that no bandwidth capacity constraints are violated at any D-switch, and on any fiber optic link. This step yields 2 matrices $A(f, s)$ and $D(f, s)$. In each D-switch s, a D-flow f arrives at a input port $j = A(f, s)$, and departs on an output port $k = D(f, s)$. Every D-flow f has a deterministic or guaranteed data-rate to be satisfied, denoted $RATE(f)$. The control-plane uses a *Maximum Flow Minimum Cost* (MFMC) routing algorithm [62]. No other routing algorithm can achieve a higher aggregate throughput with a lower cost. The MFMC routing algorithm can route D-flows to achieve up to $\approx$ 100 % utilization of the links in sub-layer 3a. (A very small fraction of each link's capacity is used for *Start-of-Frame* signals or packets).

2) COMPUTE AGGREGATE TRAFFIC RATE MATRICES

In box 3, the SDN control-plane can determine an aggregate traffic rate matrix $T(j, k)$ for each D-switch s. This matrix represents the aggregate traffic demand between the input and output ports of each D-switch, resulting from potentially millions of D-flows. This step yields a 3D array $T(j, k, s)$, where the 3rd index identifies the D-switch s. For every D-flow f that traverses D-switch s, the data-rate $RATE(f)$ is added to element $T(j, k, s)$, where $j = A(f, s)$ and $k = D(f, s)$.

3) LOW-JITTER SCHEDULES FOR THE AGGREGATE TRAFFIC RATE MATRICES

In box 4, for every D-switch s the aggregate traffic rate matrix is scheduled. The scheduling of traffic through a CIOQ switch to meet QoS guarantees has a long history spanning a few decades [54], [56]–[59]. The problem of scheduling traffic in a CIOQ switch to achieve maximum throughput and minimum jitter is known to be NP-hard.

The SDN control-plane uses a fast recursive scheduling algorithm based upon *Recursive and Fair Stochastic Matrix Decomposition* (RFSMD) [20], [60]. Given an $N \times N$ doubly stochastic traffic matrix $D(j, k)$, where the sum of each row or column $\leq F$, this algorithm will compute a D-schedule

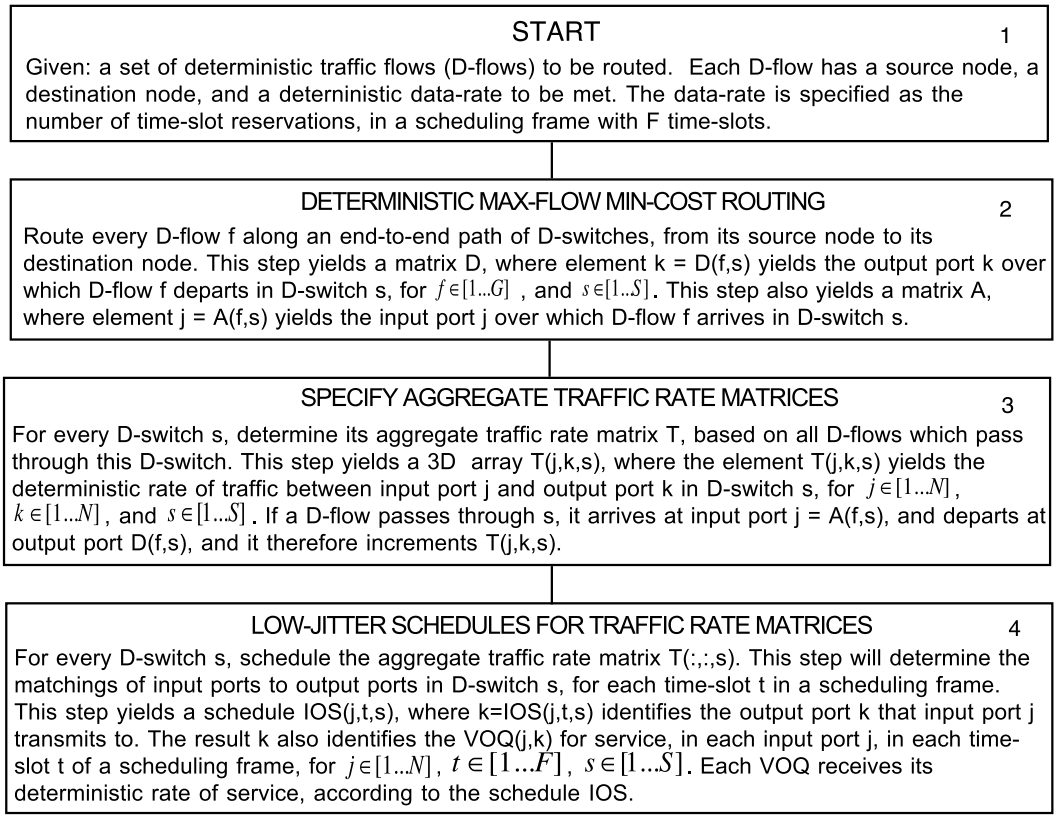

FIGURE 4. SDN flow-chart, part 1.

consisting of F permutations, where F is the length of the scheduling-frame. The algorithm is exceptionally fast, with an execution time of $O((NF)log(NF))$. The algorithm also achieves up to 100% throughput, and a bounded and very low-jitter, to minimize queueing delays. Our low-jitter D-schedules can reduce end-to-end delays to the speed of light in fiber (see section VI).

This scheduling yields a 3D array $IOS(j, t, s)$, where $k = IOS(j, t, s)$ yields the output port k. The index k identifies the $VOQ(j, k)$ at input port j that will transmit, in time-slot t of D-switch s. Let the value s be fixed, to identify a 2D matrix for D-switch s. Thus, $k = IOS(j, t)$ determines which $VOQ(j, k)$ at input port j of D-switch s is scheduled to transmit, for each time-slot of a scheduling frame. This first D-schedule provides each *VOQ* with its deterministic or guaranteed rate of transmission.

4) LOW-JITTER SCHEDULES OF D-FLOWS ONTO OUTPUT LINKS

In Fig. 5 box 5, the D-flows in each *VOQ* are scheduled for transmission on each output link k, in each D-switch s. The guaranteed-rate service that each *VOQ* was allocated in box 4 is in turn allocated to the D-flows buffered within each *VOQ* in box 5. Algorithms to schedule the D-flows within each *VOQ* are given in [20]. These scheduling algorithms also minimize the jitter, which will reduce queue sizes and end-to-end queueing delays.

For scheduling purposes, a traffic class with a deterministic data-rate on an IoT link is treated as a D-flow, in boxes 5, 6, and 7. A traffic class can be a *DiffServ* traffic class, or a new traffic class. Each traffic class receives a deterministic rate of service at each D-switch. In this manner, a single transmission schedule can also support all D-flows and all traffic classes [20], [63]. Each traffic class can have its own class-queue within the VOQ, which can contain 1,000s of D-flows. The D-flows within a traffic class can receive simple *FIFO (First In First Out)* service. Hence, by using traffic classes the queueing and scheduling is simplified, relative to the *IntServ* model [20]. In a D-switch s, box 5 yields an matrix $TFQ(j, t, s)$, where $f = TFQ(j, t)$ yields the D-flow (or the traffic class) f which receives service, at input port j of the D-switch s at time-slot t. In the SDN control-plane, this matrix for D-switch s can be stored in a 3D array $TFQ(j, t, s)$ by fixing s.

In box 5, the matrix *TFQ* yields a D-schedule for every D-switch, which identifies the D-flow (or the traffic class) to receive service, for each VOQ, in each time-slot of a scheduling-frame. Using these *TFQ* schedules, the control-plane will also compute for each D-switch, the schedule of D-flows (and traffic classes) which depart on each output port k, in each time-slot of a scheduling frame. Call

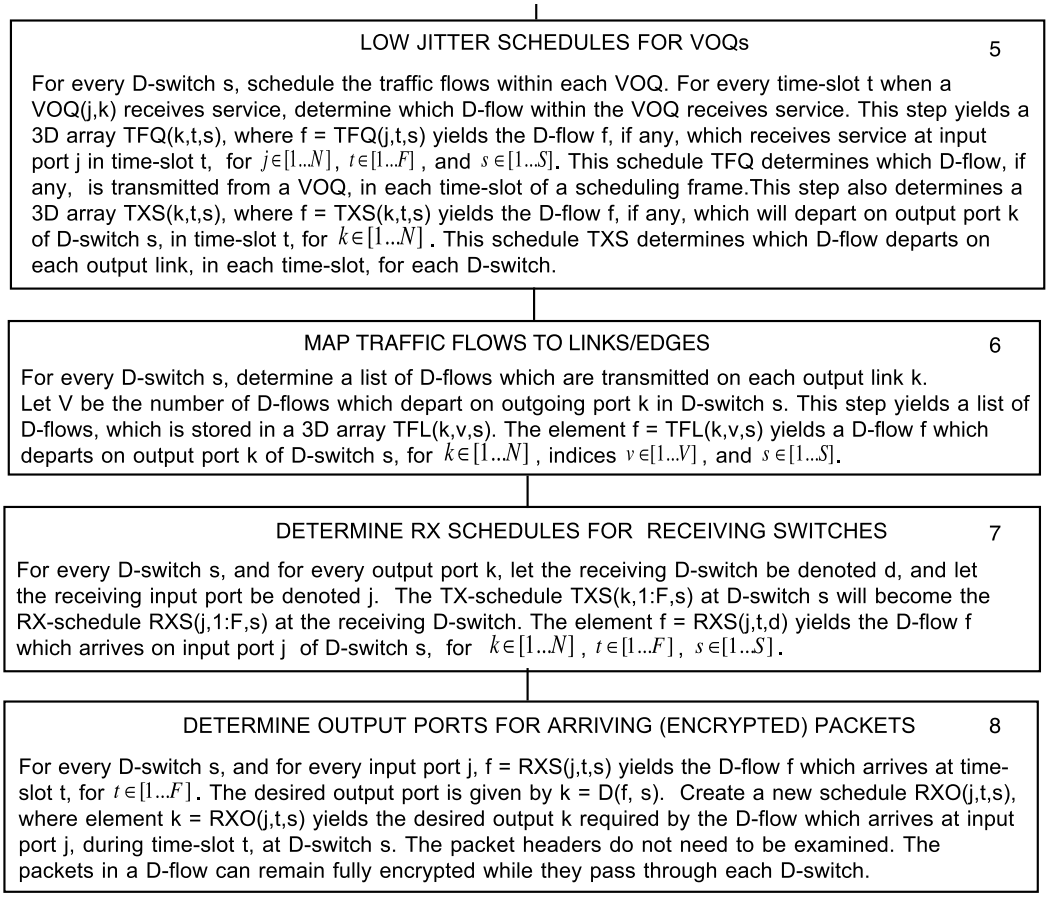

FIGURE 5. SDN flow-chart, part 2.

this schedule the *Deterministic Transmission* (*DTX*) schedule. Step 5 will thus yield an array $DTX(k, t, s)$, where $f = DTX(k, t, s)$ yields the D-flow (or traffic class) which departs on output link k, at time-slot t, in D-switch s.

5) MAP THE D-FLOWS ONTO LINKS/EDGES

In box 6, for each D-switch, a list of D-flows (and traffic classes) which depart on each output port k is determined, from the routing information in box 2. A list of D-flows (and traffic classes) which arrive on input port j of each D-switch s is also determined, from the routing information in box 2. These lists are used to strengthen cybersecurity.

6) COMPUTE DRX SCHEDULES FOR D-SWITCHES

In box 7, a *Deterministic Reception* (*DRX*) schedule is computed for every input port j at every D-switch d. In every D-switch s, the output port k will lead to an input port j at a receiving D-switch d. Hence, the *Deterministic Transmission* (*DTX*) schedule of output port k of D-switch s will become the *Deterministic Reception* (*DRX*) schedule of input port j of receiving D-switch d. The schedule $f = DRX(j, t, d)$ identifies the D-flow f (or traffic class) which will arrive at each time-slot t in a scheduling frame, at input port j of D-switch d. Each D-switch d will now know the precise arrival time-slots of D-flows (or traffic classes) on each of its incoming ports.

7) COMPUTE D-SCHEDULES TO AVOID PROCESSING PACKET HEADERS

In Box 8, the SDN control-plane will determine another D-schedule, which will allows encrypted packets to traverse a D-switch, without examining packet headers. D-switches do not need to examine packet headers, in order to make layer-3 routing decisions. In box 8, in each D-switch s, and each input port j, a new schedule *RXO* is computed, which identifies the output port k needed by the D-flow which arrives on input port j of D-switch s, for every time-slot t in a scheduling frame. The SDN control-plane has already determined that D-flow $f = DRX(j, t, s)$ will arrive at switch s on input port j, at time-slot t. The output port k used by this D-flow f in D-switch s at time slot t is given by $k = RXO(j, t, s)$. The desired output port k in D-switch s can be determined from the list of D-flows associated with each output port k, which was computed in box 6. This schedule $RXO(j, t, s)$ represents another D-schedule, which will remove the need for a D-switch to process the packet-headers of arriving

D-flows. For example, at input port j of D-switch s, every packet which arrives at time-slot t will be routed to output port $k = RXO(j, t, x)$, and this information identifies the $VOQ(j, k)$ to receive the packet. (To determine the output port needed by a packet in a traffic class, some extra processing is required. For example, each traffic class can be associated with a distinct destination node).

By performing this flow-chart, every D-switch can receive multiple D-schedules, which provide every D-flow with its deterministic data-rate, and which will also remove the need for (a) D-switches to perform the tasks of routing and scheduling of D-flows or traffic classes, or (b) to process any packet-headers. Therefore, packets can be completely encrypted at the source, and they can remain encrypted while they traverse the network from end-to-end. This approach eliminates many Gigabytes of high-speed RAM (memory) to store insecure layer-3 BGP routing tables. It also eliminates the need for a processor running a Linux operating system and the insecure *Berkeley Sockets* software, to update layer-3 BGP routing tables, using insecure layer-3 protocols such as ICMP, BGP, and IGP.

(Note that the schedules for IP-controller2 and OP-controller1 are dependent upon the D-schedule in the switch-controller, and they can also be determined by performing some logic operations on the latter schedule, thus removing 2 D-schedules.)

8) THE MANAGEMENT AND CONTROL DVPN

Some DVPNs will be created to last for long periods of time, i.e., days, weeks, months or years. The SDN control-plane can configure one special long-term DVPN for management and control of the *Deterministic IoT*. This DVPN has a D-flow interconnecting the SDN control-plane to every secured component under its control. (The SDN control-plane is distributed, so multiple D-flows may be used to interconnect these components.) This management and control DVPN is immune to congestion, interference, and DDOS attacks, and the end-to-end delays are reduced to the speed of light in fiber.

V. SECURITY PROPERTIES OF THE DETERMINISTIC IoT

A. ATTACKS BY AN EXTERNAL CYBERATTACKER

This sub-section will consider the challenges that an external cyberattacker will face to inject a malicious packet into the fiber in sub-layer-3a, to perform any type of cyberattack, i.e., a *Man-in-the-Middle* attack. There are thousands of kilometers of fiber in the network. Note that it will be extremely difficult for a cyberattacker to access a fiber and install an untrusted component without detection, as the SDN control-plane can monitor each fiber-optic link in sub-layer-3a periodically (i.e., every few milliseconds), to detect any anomalies. Nevertheless, suppose that a cyberattacker has gained access to a fiber, and has installed an *untrusted component* that it controls, so it can access the encrypted communications. Suppose the cyberattacker has a *Quantum Computer*. The cyberattacker can inject a malicious encrypted packet in 2 manners; (a) insert a malicious packet in a time-interval in which no transmission is observed, or (b) insert a malicious packet to overwrite a legitimate packet transmission. The above scenarios are improbable, but they represent the simplest ways to inject a malicious packet.

If a cyberattacker injects a malicious packet during a time-interval in which no transmission was scheduled, the malicious packet will be quickly detected at the receiving D-switch (or D-sink), as it will violate a D-schedule. If a cyberattacker injects a malicious packet by overwriting a legitimate transmission, or if it transmits during a time interval when a transmission is allowed (but not observed), then that malicious packet will be forwarded along the path, until it reaches its destination. At the destination, the malicious packet will be received by a D-sink, and it will be decrypted.

In order to be accepted as a legitimate packet, the malicious packet must pass the *Authorization-Check*. In section III, it was shown that the probability an external cyberattacker can transmit even one undetected malicious packet into a DVPN that passes an *Authorization-Check* can be made arbitrarily small, by using longer *Quantum Safe* encryption keys.

B. ATTACKS BY AN INTERNAL CYBERATTACKER–CONTAINMENT

Consider an *'internal'* cyberattacker, who already has compromised a secured computer within a DVPN. An *internal* cyberattacker may gain access to a secured computer through a human error. The administrative entity in control of a DVPN is ultimately responsible for the software running on its secured internal computer systems within the DVPN. There are 2 cases to consider:

Internal Cyberattacker - Case 1: A computer compromised by an internal cyberattacker within a DVPN may attempt to compromise other computers around the world, in remote DVPNs. Such attacks are impossible, since the communications of a DVPN are isolated, and cannot leave the DVPN. There is a double level of protection here. Even if a compromised computer could send a message to a remote DVPN, the message would fail the remote *Authorization-Check*, and the remote DVPN would view the message as one from an *external cyberattacker*. The previous discussion would apply: the probability an *external cyberattacker* could compromise any one of the millions of remote DVPNs can be made arbitrarily small.

Internal Cyberattacker - Case 2: A compromised computer within a DVPN may attempt to compromise other computers/resources within the same DVPN. To protect against this scenario, each administrative entity must: (a) run anti-virus software on all of its internal computer systems; and (b) implement a *'Next-Generation Firewall'* and IDS on its internal computer systems, to detect and prevent any cyberattacks originating from within the administrative entity's own secured computers. However, as stated earlier, if an administrative entity's secured internal computer system becomes compromised by an internal cyberattacker, then the damage will be contained to the DVPN to which it belongs.

The damage will not spread to the millions of other DVPNs in the system. Furthermore, if an administrative entity adheres closely to the US NIST *Zero Trust Architecture*, then each DVPN will protect a *micro-perimeter* with a single resource. Hence, an internal cyberattacker will only gain access to a very small amount of resources.

C. SECURITY OF THE SDN CONTROL-PLANE AND DVPN CONTROLLERS

Exceptionally-strong cybersecurity requires that the SDN control-plane and DVPN-controllers are secure. Specifically:

- The SDN control-plane and DVPN controllers cannot be compromised by a cyberattacker.
- The communications between the SDN control-plane, DVPN controllers and the secured components cannot be compromised.

The SDN control-plane (and DVPN-controllers) are distributed software systems, which are expected to execute in several data-centers simultaneously. The data-centers are interconnected with the special management and control DVPN, using ultra-low latency D-flows. To achieve exceptionally strong security, multiple copies of each system can execute in parallel, in different data-centers, and majority voting logic is used to make decisions. For example, 5 copies of each system can operate in parallel, and every decision would require the consent of 3 copies, otherwise a problem has been detected. This scheme would allow 2 copies of the system to be rendered inactive, i.e., due to a fire, earth-quake or other catastrophe, and the system would still function. It would be virtually-impossible for an external cyberattacker to compromise a system executing in one data-center, let alone compromise multiple parallel systems simultaneously in different data-centers, especially when all communications use long *Quantum Safe* encryption keys with 1,000s or 10,000s of bits. As summarized in section III, the extended AES and RINJDAEL encryption algorithms are *Quantum Safe* when using 256 bit keys, let alone when using keys with 1,000s of bits.

1) SCALABILITY OF THE SDN CONTROL-PLANE

To achieve scalability, the SDN control-plane interacts with many independent DVPN-controllers. Each corporate entity has its own DVPN-controller, to interact with the centralized SDN control-plane, and to control the secured components within its own DVPNs.

D. A CONCISE SUMMARY OF SECURITY PROPERTIES

Properties 1-4 were presented in section III. Properties 5-14 are summarized next.

Property 5 (Guaranteed Data-Rate): Every D-flow will receive a deterministic (or guaranteed) data-rate through a path of D-switches, from a D-source to a D-sink. The data-rate can be expressed as a guaranteed number of time-slot reservations within a scheduling-frame consisting of F time-slots. Assuming all fiber-optic links support a data-rate of 200 Gbps, and letting F=32K, then each time-slot reservation represents a data-rate of about 6 Megabits per second.

Property 6 (Routing Authorized D-Flows): The SDN control-plane will route every D-flow along a fixed path of D-switches, using a *Max-Flow Min-Cost* routing algorithm [62], which can achieve $\approx 100\%$ utilization of the link capacity in sub-layer 3a. (Typically a very small fraction of the link capacity is used for a *Start-of-Frame* signal/packet). The SDN control-plane can also determine 2 lists for each D-switch, to strengthen cybersecurity: (i) the list of authorized D-flows that arrive (or depart) at every incoming (or outgoing) fiber, respectively. This property solves a significant weakness in today's BE-IoT, in that layer 3 IP packet routing using BGP is insecure (see section II.A).

Property 7 (The DTX Schedule): The SDN control-plane can compute a *Deterministic Transmission* (DTX) schedule for every fiber leaving a D-switch. This schedule identifies the D-flow (or traffic class) with a transmission reservation, in each time-slot of the scheduling frame. Using the scheduling algorithms in [20], the schedules can be circularly rotated and still minimize buffer sizes and delays, so that the D-switches do not need to be tightly synchronized. However, each D-switch must recognize a *'Start-of-Frame'* signal/packet from each of its neighbors, roughly once every millisecond. The size of the data queues can be reduced by factors of 1,000+, compared to a BE IP router, and the end-to-end queueing delays can be reduced to the speed of light in fiber. The DTX schedule is used in the proposed *guaranteed IDS*, to detect any data transmission which occurs in a time-interval for which no transmission has been scheduled.

Property 8 (The DRX Schedule): The SDN control-plane can compute a *Deterministic Reception* (DRX) schedule for every fiber arriving at a D-switch. This schedule identifies the D-flow (or traffic class) with an arrival reservation for that fiber, for every time-slot of a periodic scheduling frame. The DRX schedule is also used in the proposed *guaranteed IDS*, to detect any data reception which occurs in a time-interval for which no reception has been scheduled.

Property 9 (End-to-End Packet Encryption): In a DVPN, a packet can be encrypted at the D-source using a long *Quantum Safe* encryption key, and it can remain fully encrypted as it traverses the network from end-to-end. Even the packet headers can be encrypted, as they are not examined at intermediate D-switches to make layer-3 routing decisions. This property solves 2 significant weaknesses in the BE-VPNs used in today's BE-IoT; (a) The BGP routing used in IP layer 3 of the BE-IoT is insecure; and (b) Layer 3 IP packet headers cannot be encrypted, and remain visible to manipulation of cyberattackers.

Property 10 (Edge-Disjoint Paths for Reliability): For mission-critical applications, multiple edge-disjoint paths can be allocated between a D-source and a D-sink in sub-layer 3a, to provide very strong reliability. The D-source can transmit multiple copies of each packet, one over each path. The D-sink will eliminate duplicate copies and keep one copy of each packet. This scheme is similar to the IEEE 802.1 TSN

FRER (Frame Replication and Elimination for Reliability) proposal used in layer-2 networks. It is also similar to the IETF Detnet *PREOF (Packet Replication, Elimination and Ordering Function)* proposal. (Mission-critical applications may also use *Forward-Error Correcting (FEC)* codes.) The cost of providing multiple paths in sub-layer 3a (using inexpensive D-switches) is much lower than the cost of providing multiple paths in layer 3 (using expensive IP routers).

Property 11 (Edge-Disjoint Paths for Cybersecurity): The use of multiple edge-disjoint paths for a D-flow within one SD-WAN, as described in property 10, can also significantly improve cybersecurity, in addition to improving reliability. Also, the use of multiple DVPNs from multiple independent SD-WANs managed by independent cloud services providers, i.e., potentially Google or Amazon, can also significantly improve cybersecurity. In both cases, it becomes increasingly difficult for a cyberattacker to compromise multiple paths in one SD-WAN, or multiple DVPNs in independent SD-WANs, simultaneously to avoid detection.

Property 12 (External CyberAttacker, Malicious Packet, Case 1): Any malicious packet transmission from an external cyberattacker, which occurs at a time-interval for which no transmission has been scheduled by the SDN control-plane, can be quickly detected.

Proof: Such a transmission will be detected by the guaranteed IDS. Consider a packet which arrives at a D-switch (or a D-sink) in a time-slot for which no arrival was scheduled. This packet will violate the DRX schedule and must be un-authorized. Such a packet will be quickly detected by the guaranteed IDS.

Similarly, consider a packet which departs from a D-switch (or a D-source) in a time-slot for which no departure was scheduled. This packet will violate the DTX schedule and must be un-authorized. Such a packet will be quickly detected by the guaranteed IDS and the SDN control-plane can be informed for corrective action.

Property 13 (External Cyber-Attacker, Malicious Packet, Case 2): Any malicious packet transmission from an external cyberattacker targeting a DVPN, which occurs by over-writing a legitimate encrypted packet transmission scheduled by the SDN control-plane, can be detected by the guaranteed IDS, with probability arbitrarily close to 1.0.

Proof: The malicious packet will be delivered to a D-sink, where it will be decrypted. The packet must pass the *Authorization-Check* to be accepted. According to section III, the probability a malicious packet from an external cyberattacker can pass an *Authorization-Check* can be made arbitrarily small, by increasing the length of the *Quantum Safe* encryption keys.

Furthermore, an external cyberattacker will be unable to target any specific destination in the DVPN. The malicious packet will be received by the D-sink associated with the D-flow (or traffic class). This property holds since the SDN control-plane determines the packet routing in DVPNs, and D-switches do not examine packet-headers within a DVPN to make routing decisions. Hence, an external cyberattacker cannot direct a malicious packet to an arbitrary destination.

Property 14 (Containment of Internal CyberAttacker): An internal cyberattacker may gain access to a secured computer system by human error. It is the responsibility of the administrative entity in charge of a DVPN to ensure that its own computers within the DVPN are secure. The entity should run anti-virus software, and implement *Next-Generation Firewalls* on its secured computers. However, the proposed paradigm provides significantly improved security, against internal cyberattackers:

- (1) The internal cyberattacker can communicate over the established D-flows within the DVPN. However, it cannot read the secret encryption/decryption keys, or create or modify D-flows, as the hardware implementation of the trusted components will not allow this.
- (2) An internal cyberattacker cannot communicate with the external world, as the communications of a DVPN are limited to within the DVPN.
- (3) An internal cyberattacker cannot initiate a cyberattack against the millions of other computers external to the DVPN, as the DVPN is completed isolated.
- (4) An internal cyberattacker can only communicate over authorized D-flows within its own DVPN, limiting its ability to compromise other computers/resources to those within the same DVPN.
- (5) If the entity adheres to the US NIST ZTA, then each DVPN will protect a single network resource, thereby limiting the damage caused by an internal cyberattacker.

In summary, if an administrative entity's secured internal computer system ever becomes compromised by an internal cyberattacker, the damage will be contained to the DVPN in question.

Property 15 (Security per DVPN): The SDN control-plane can issue a private *Quantum Safe* encryption key of suitable length, to the D-source and D-sink of every D-flow that it establishes in a DVPN. A moderate security level may entail encryption keys with 100s of bits. A high security level may entail encryption keys with 1,000s of bits. A very high security level may entail encryption keys with 10,000s of bits.

VI. EXPERIMENTAL RESULTS

Fig. 2b illustrates a USA *Deteministic IoT* network, with 26 cities and 82 edges. Each city has a D-source, a D-sink, and a D-switch. This testbed was implemented on an Altera FPGA. The testbed used a scheduling frame with 1,024 time-slots. It transmitted small packets of size $\approx$ 20 bytes, at a rate of over 400 million packets per second. The hardware testbed results were identical to the results of a software simulator, which was written in Matlab, and the results were consistent with theoretical expectations [20].

For this paper, the SDN control-plane programmed 983 D-flows into the USA topology with 26 cities and 94 edges, to achieve a high link utilization of $\approx$ 99% in sub-layer-3a, and the performance was determined using the software simulator. (Similar testbeds for the USA *Deterministic IoT* were

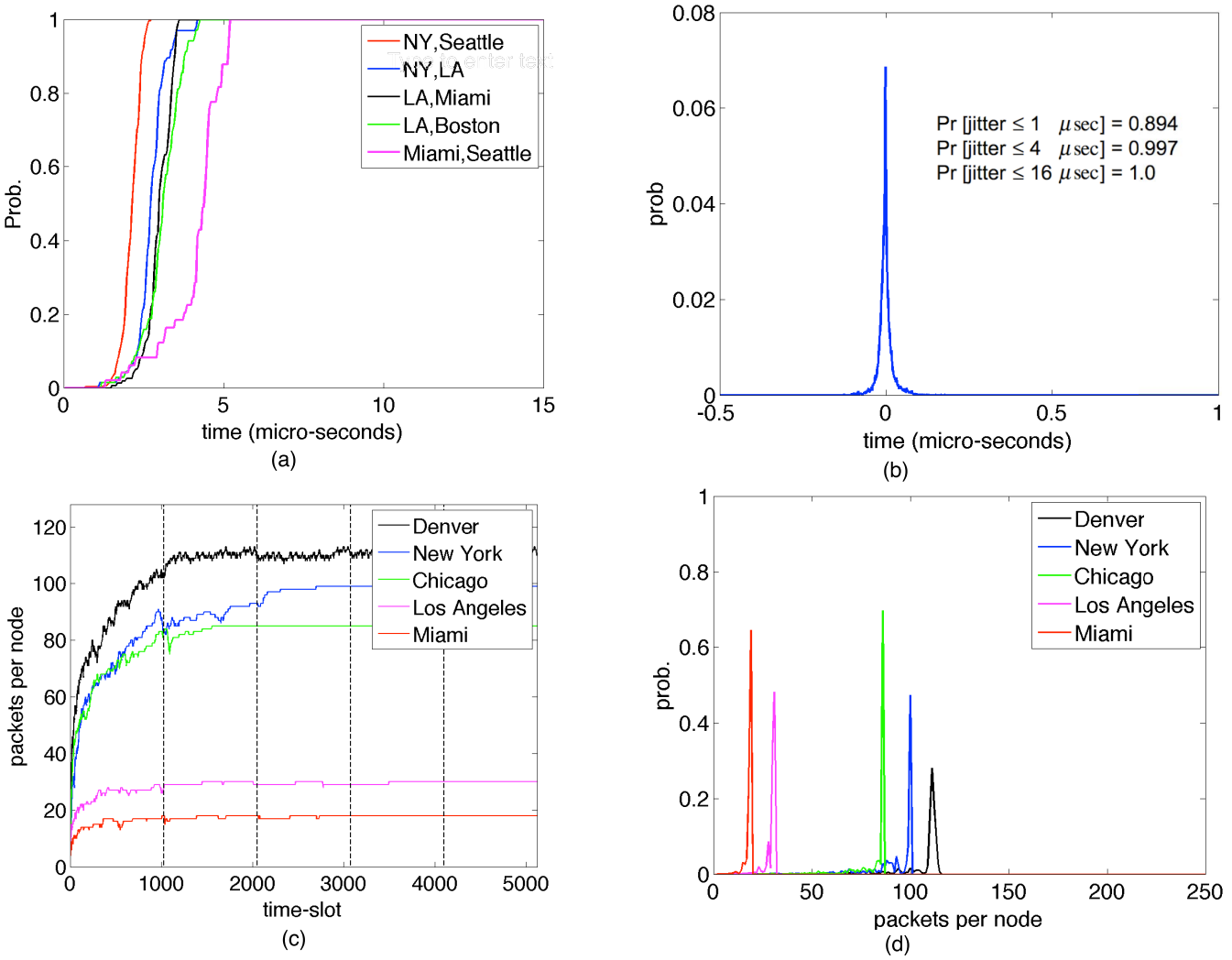

FIGURE 6. (a) End-to-end queuing delay CDF on selected D-Flows. (b) End-to-end queueing delay jitter distribution for all D-flows. (c) Packets queued per D-switch, versus time. (d) Distribution of packets queued per D-switch.

presented in [17] and [22]. In those testbeds, ≈ 310 D-flows were programmed into the USA topology, to achieve a link utilization of ≈ 92% in sub-layer 3a.)

In Fig. 6, let the fibers in sub-layer 3a operate at 200 Gbps, consistent with today's Silicon Photonics transceivers. Assume 1,024-byte IP packets are used, and that an IP packet transmission takes 1 time-slot. (Larger IP packets can use more time-slots, or they can be fragmented into 1K byte fragments, which are sent over sub-layer 3a.) Therefore, each time-slot has a duration of ≈ 41 nanoseconds.

Fig. 6a illustrates the end-to-end queueing delays for several D-flows, in microseconds. The end-to-end queueing delays are all ≤ 5 microseconds. Consider the D-flow between LA and Miami. The distance between these two cities is ≈ 3800 km, and the fiber latency is ≈ 19 milliseconds. The queueing delay is ≈ 1,000 times smaller than the end-to-end fiber delay, consistent with the theory presented in [20].

Fig. 6b illustrates the end-to-end delay jitter of the packets, when they arrive at a D-sink. Over 99% of all packets experience a delay jitter ≤ 4 μsec. These jitters are much smaller than the end-to-end fiber delays in the USA network, which are measured in milliseconds.

Fig. 6c illustrates the evolution of the number of packets queued in each city versus time, assuming an empty network at time-slot 0. The vertical lines represent the start of a new scheduling frame (with F=1,024 time-slots). The evolution reaches a deterministic pattern which repeats for each scheduling frame. The steady-state is reached after ≈ 4 scheduling frames (i.e., ≈ 4,096 time-slots).

1) ROUTER BUFFER SIZING RULES OF THUMB

Fig. 6d shows that at most about 120 packets are buffered any node, even when the links in sub-layer 3a operate at ≈ 99% utilization. The well-known *Bandwidth-Delay-Product* (BDP) buffer-sizing rule of thumb is used to find approximate buffer sizes in BE IP routers (see Appendix B). Assume an average *Round-Trip-Time* (RTT) of 250 milliseconds for the flows traversing a link (as in [52]), and a link of capacity C = 200 Gbps. The BDP rule of thumb states that the worst-case

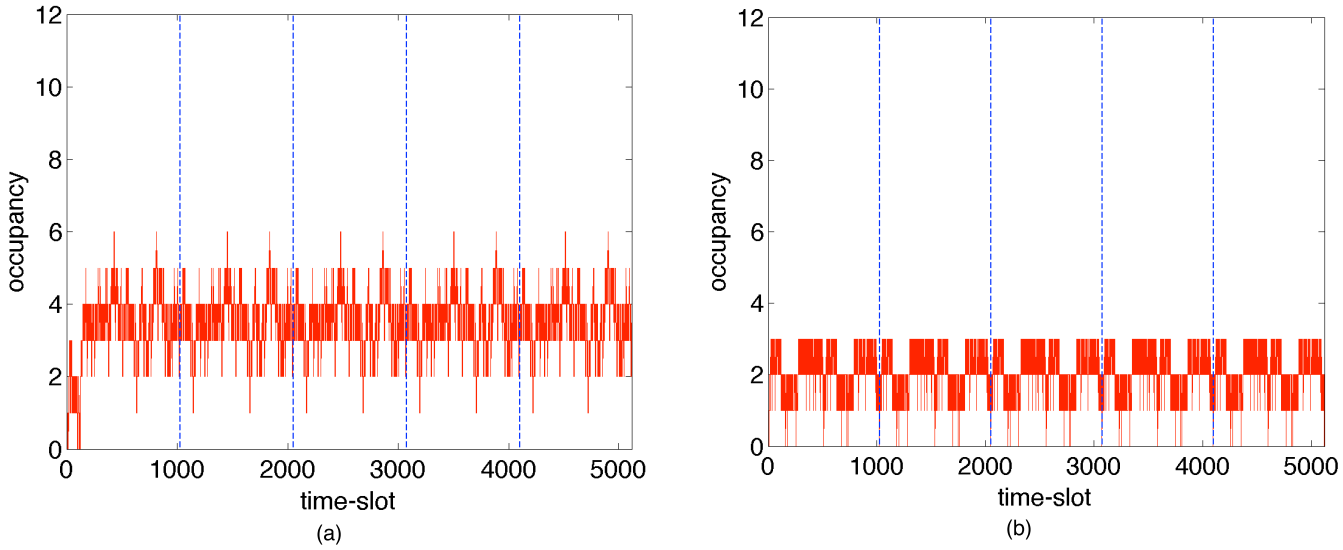

FIGURE 7. Occupancy of two VOQs in the Las Vegas switch (in packets).

buffer size for the link, to avoid exhausting a buffer is $RTT \cdot C$ or about 6.25 Gigabytes, or about 6.1 million packets (at 1K-bytes each). A BE IP router with degree 3 would require worst-case buffer sizes of $\approx$ 18 million packets. This extensive amount of buffering represents a phenomena called *Buffer Bloat*, where a BE IP router may buffer many packets from an individual BE-flow [6]. As shown in Fig. 6d, the use of D-switches has reduced the worst-case buffer size from $\approx$ 18 million packets down to $\approx$ 120 packets, a reduction of $\approx$ 150,000 times.

Reference [52] has presented a *Small Buffer* rule of thumb, where the worst-case buffer size for each IoT link is $RTT \cdot C/\sqrt{N}$, where N is the number of long-lived TCP (*Transmission Control Protocol*) flows traversing a link. Letting $N \approx 1,024$, the *Small Buffer* rule indicates that the worst-case buffer size for each link is about 195 Megabytes, or about 195K packets, a significant reduction. When compared to the *Small Buffer* rule, the use of D-switches has reduced the buffer size for a router of degree 3 from $\approx$ 585K packets down to $\approx$ 120 packets, a reduction of $\approx$ 4,875 times.

Our analysis and experimental results indicates that for deterministic traffic, a new rule for buffer sizes can be stated. The *Deterministic Buffer Size* rule states that the amount of buffering required in a D-switch, to avoid exhausting a buffer and to maintain 100% throughput, is K packet buffers per D-flow, where K is a small number that depends upon the *Smoothness* of the deterministic service that a D-flow receives [20], [63]. According to [20], [63], K $\approx$ 1/2 packet per D-flow when very smooth schedules are used. The *Smoothness* can be defined as the worst-case lead or lag in the service a D-flow receives, relative to a perfectly-scheduled D-flow. The smoothness is called the *Normalized Service Lead/Lag* in [20] and [63].

Fig. 7 illustrates the occupancy (i.e., the number of packets queued) per VOQ, for 2 VOQs in the Las Vegas switch, versus time. The packet arrivals and departures for each VOQ form deterministic processes, with no randomness. The processes converge to steady-state patterns, after $\approx$ 2..4 scheduling-frames. As stated in properties 7 and 8, malicious packets from an external cyberattacker (that are transmitted when no transmissions have been scheduled) will violate the deterministic DTX or DRX schedules in a D-switch, and will be quickly detected by the guaranteed IDS.

TABLE 2. Revenue from 2020 annual reports.

Company	2020 Annual Revenue (USD)
Arista	$2.32 Billion USD
Cisco	$49.8 Billion USD
Huawei	(891.368 billion CNY) $136.7 Billion USD
Juniper	$4.445 Billion USD
Nokia	(29.1 Billion Euros) $25.0 Billion USD
Total	$218.3 Billion USD

VII. EXTENSIONS AND IMPLICATIONS

This section discusses some extensions and implications of the proposed paradigm.

A. CAPITAL COSTS OF THE BE-IoT

Table 2 shows the 2020 yearly revenue for some major BE-IoT equipment manufacturers (in US dollars), from their 2020 annual reports. (Nokia purchased Alcatel/Lucent Technologies in 2015). Cisco reports revenues of $49.3 Billion in 2020. Approximately 72% of this figure represents products, while 28% represents services. The total revenue for these companies in 2020 is $\approx$ $218 Billion USD. Assuming 72% of this figure represents global product sales, then the global capital costs of BE-IoT equipment can be estimated at $\approx$ $157 Billion USD, in 2020.

Google has reported that BE-IoT links operate at about 25% utilization [25]. Assuming the global BE-IoT operates at 25% utilization, then ≈ 75% of the capital costs are effectively wasted, i.e., the wasted capital costs are about $120 Billion US per year in 2020. If the use of SDN control-planes and SD-WANs can improve the utilization of the global IoT network by ≈ 25% to 50%, then the potential savings in capital costs are ≈ $39 Billion to $78 Billion USD per year, respectively.

B. OPERATIONAL COSTS OF THE BE-IoT

Cisco has estimated that up to 95% of the network configuration changes in BE-IoT equipment (i.e., routers and firewalls) are performed manually [8]. Cisco estimates that the operational costs of manually maintaining the current BE-IoT infrastructure is about 2 or 3 times the capital costs. Given that global capital costs can be estimated at ≈ $157 Billion USD per year in 2020, then according to Cisco's estimates, the annual operational costs of the BE-IoT can be estimated to be ≈ $300 Billion up to $450 Billion USD per year. (These figures represents less than 1% of global GDP in 2020, so they seem realistic given the pervasiveness of the BE-IoT.) If the use of SDN control-planes and SD-WANs can improve the controllability and reduce operational costs by ≈ 50% per year, then the potential savings in operational costs can be estimated at ≈ $150 Billion to $225 Billion USD per year, respectively.

Cisco has argued for the need to *'Transform the Infrastructure'*, to include SDN controllers and SD-WANs, to allow for unified domain controls and policies [8]. The proposed paradigm exploits an SDN control-plane and SD-WANs, which combine 5 distinct topics (deterministic communications, PQC, the ZTA, the ACS and the IDS) to significantly improve cybersecurity.

C. The Log4j CYBERATTACK

The log4j cyberattack was first reported in Dec. 2021 [128], [129]. Log4j is widely-used open-source software from the *Apache Software Foundation*, that is used in servers to log events such as diagnostic messages, and to communicate the results to system administrators. For example, it can record the user names of all users attempting to log into a web-server. It is used in video games such as Minecraft, and in Apple iCloud and Amazon Web Services.

To enhance its functionality, log4j can perform remote *lookups*, i.e., JNDI (*Java Naming and Directory Interface*) lookups and Java RMI (*Remote Method Invocation*) registry lookups [130]. The JNDI program can utilize naming and directory services from many widely available remote servers, i.e., (i) LDAP (*Lightweight Directory Access Protocol*) servers, to look up data such as usernames.

Unfortunately, JNDI does not protect against lookups from servers controlled by an attacker, and thus log4j is vulnerable to cyberattacks. To launch an attack, an attacker queries a web-server, in an attempt to trigger a log record. The query includes malicious text, which will cause the web-server to access a remote LDAP server with malware, that the attacker controls. The attacker can then take control of the web-server. It is expected that log4j vulnerabilities will be around for many years to come, given the wide-spread use of log4j in other software products.

The US CISA in document CVE-2021-44228 has several recommendations to protect vulnerable assets [128], [129]:

- Remove the asset from the network (unplug the cable)
- Implement a Firewall to stop lookups from unknown servers
- Restrict the asset's communications to the Internet or the enterprise network

More detailed steps can be taken, but these require more specialized equipment, i.e., a *Next-Generation Firewall* or an IDPS that performs *Deep Packet Inspection*:

- Block outbound LDAP traffic, or implement an allowlist to only allow access to known good destinations
- Block outbound RMI traffic, or implement an allowlist to only allow access to known good destinations
- Block outbound DNS traffic, or implement an allowlist to only allow access to known good destinations

The CISA term *allowlist* represents a simplified *Access Control System*.

How the Proposed Paradigm can Mitigate log4j Risk: Consider a web-server for a critical infrastructure, such as the *Smart Power Grid*. Suppose that it receives messages from 1,000s of remote computers and devices which monitor the grid, and it uses the log4j software to log information. The proposed paradigm offers several levels of protection: (i) All communications between the web-server and remote devices will belong to one large DVPN, or several smaller DVPNs, and these DVPNs are completely isolated from the rest of the IoT. No unauthorized entity (i.e., external cyberattacker) will be able to send a message to the server or any computer/device. (ii) The web-server will only be able to access other remote servers, that have been pre-approved by the ZTA/ACS. The web-server will be blocked from accessing any arbitrary remote server that is not approved by the ZTA/ACS. (iii) By implementing a ZTA, the probability that an internal cyberattacker exists and can access the web-server is minimized, as all users, devices and applications are continually authenticated using a rule-based ACS and *Multi-Factor Authentication*. The proposed paradigm thus implements the key recommendations of the CISA.

D. DETERMINISTIC VIDEO DISTRIBUTION NETWORK

According to Cisco, global IP networks will carry 9.1 Exabytes of traffic per day in 2021 (an Exabyte equals one billion Gigabytes) [26]. The amount of IP traffic crossing the global IP network every minute will equal the gigabyte equivalent of all movies ever made. Globally, IP video will form 82% of all IP traffic in 2021, and this figure is slowly increasing with time.

A 4K UHD (*Ultra High Definition*) video stream with a standard frame rate has a bit-rate of typically < 60 Mbps,

while a high frame rate has a bit-rate of typically < 100 Mbps. Consider a 200 Gbps BE-IoT link that operates at 50% utilization, thus carrying 100 Gbps of traffic. According to Cisco, about 82 Gbps will this figure will be IP video traffic, on average. Assuming each video stream requires 100 Mbps, the minimum number of IP video streams carried on this link is ≈ 820.

Video traffic is well suited for transport over deterministic SD-WANs in sub-layer 3a. A single video stream is very bursty. However, the aggregation of 100s of independent video streams significantly lowers the burstiness, as the aggregated stream converges toward a constant bit-rate video stream [17]. Given that about 82% of all IP traffic is video traffic in 2021, there is a significant opportunity to migrate much this traffic to the SD-WANs in deterministic sub-layer 3a, where it can be carried with ultra-low latency, and with effectively close to 100% link utilization. This migration would lower the amount of traffic carried in the BE-IoT in layer 3, thus decreasing the annual capital costs of deploying relatively inefficient and expensive BE IP routers in layer 3. It will increase the amount of traffic carried in the *Deterministic IoT* in sub-layer 3a, thus increasing the much lower capital costs for deploying highly-efficient and lower cost D-switches in sub-layer 3a.

E. ROUTING AND QUEUEING COMPARISON, D-FLOWS VS BE-FLOWS

The layer 3 BGP protocol introduces considerable complexity and security weaknesses into a BE IP router. A BE IP router might process anywhere between 100 million and 1 billion IP packets per second, representing a tremendous amount of computational work and energy. Furthermore, the resulting BGP routing is not secure, as cyberattackers can maliciously update BGP routing tables in BE IP routers. It is estimated that layer 3 packet header processing consumes about 60% of the power of a BE IP router [127].

The proposed paradigm removes all routing and scheduling algorithms, from the D-switches in sub-layer 3a, and moves them into the SDN control-plane. This migration results in a dramatic simplification of the D-switches, relative to a BE IP router, such that a D-switch can be implement on one (or a few) integrated circuits (FPGAs or ASICs). Consider the routing of D-flows in sub-layer 3a using the *Max-Flow Min-Cost* routing algorithm in [20]. According to [20], the routing for 416 D-flows will take ≈ 8.17 seconds, on a dual core microprocessor. The SDN control-plane resides in one (or more) data-centers, and it is expected that between 1,000 and 50,000 multi-core processors within a data-center will be applied to perform the routing and scheduling (each with 4-8 cores). Assume 1,000 quad-core processors are assigned to the routing, and they achieve $\approx 50\%$ of the peak performance. The routing time for 416 D-flows in the SDN control-plane is ≈ 4.1 millisecond. The routing time per D-flow can thus be estimated at ≈ 10 μseconds.

Once the routing is complete, the D-schedules are computed by the SDN control-plane. Assume each D-flow represents a *High Definition* (HD) IP video stream, with a data-rate of about 6.1 Mbps, over a period of 2 hours. Assume a scheduling-frame with 32K time-slots, where each time-slot has a duration of 41 nanosec. This scheduling frame is reused repeatedly for 2 hours, and each D-flow sends about 5.4 million packets in 2 hours. The scheduling of these 416 D-flows in a dual core processor, into one scheduling-frame, is estimated to take ≈ 40 seconds. The scheduling of these 416 D-flows in the SDN control-plane, using 1,000 quad-core processors, is expected to take $\approx$ 20 milliseconds. The scheduling time per D-flow per hop is thus ≈ 48 μsecond. Assume the D-flow will traverse a path with 10 BE IP routers (i.e., 10 hops). The total routing+scheduling time, for 1 D-flow in 10 hops, is dominated by the scheduling time, and ≈ 490 μseconds. Once the routing and scheduling have been completed, the D-schedules can be re-used for extended periods of time, i.e., days, weeks, or months. The total queueing time for 1 packet in a D-flow, per hop, is ≈ 1 μsec, as observed from Fig. 6(a). The total queueing time for 5.4M packets in a D-flow, per hop, is $\approx$ 5.4 sec. The total queueing time for 5.4M packets in a D-flow, over 10 hops, is thus ≈ 54 seconds.

Consider the performance of BE-traffic flows, which attempt to carry the same amount of video traffic in the BE-IoT network. The IP packet headers will be repeatedly processed in all the BE IP routers traversed by the BE-flows, over a long period of time, to perform the routing and scheduling. The BGP routing will consume a significant amount of time and energy, and the BGP routing will be insecure. Assume each BE packet incurs a time of ≈ 5 nanosec per hop, for routing and scheduling overhead (please see Appendix B). The total routing+scheduling time for 5.4M BE packets in one hop is thus about 26 milliseconds, and the total routing+scheduling time for 5.4M BE packets in 10 hops is thus about 260 milliseconds. Assume the total queueing time per packet per hop will be ≈ 5 milliseconds. The total queueing time of 5.4M BE packets per hop will be $\approx 27{,}000$ seconds.

Table 3 compares the routing+scheduling time and total queueing delays, for D-flows and BE-flows. Comparing the total time for routing+scheduling, the D-flows are about 540 times faster than the BE-flows. Much of this time advantage comes from the use of 1,000 quad-core processors to perform the routing and scheduling of D-flows in a data-center, plus the fact that the D-schedules are re-used 1,000s of times. Comparing the total queueing time, the D-flows are about 5,000 times faster. Much of this time advantage comes from the *Deterministic Buffer Size* rule that applies, where each D-flow buffers K packets per hop, for a small number K (where K is ≈ 0.5 for the D-switch shown in Fig. 3). In contrast, each BE-flow buffers typically between 5 and 10 packets in each BE IP router in this example, as shown by the large total queueing time, contributing to a phenomena known as *Buffer Bloat* [6].

The paradigm can thus yield significant improvements in performance, energy, and security, for applications in which

TABLE 3. Comparison of routing + scheduling time, for D-flows vs. BE-flows.

Number of packets (and type of flow)	Routing+Scheduling time (10 hops)	Total Queueing Time one hop	Total Queueing Time (end-to-end, 10 hops)
1 packet, D-flow	not applicable	$\approx$ 1 μsec	$\approx$ 10 μsec
5.4M packets, D-flow	$\approx$ 490 μsec	$\approx$ 5.4 sec	$\approx$ 54 sec
1 packet, BE-flow	$\approx$ 50 nanosec	$\approx$ 5 millisec	$\approx$ 50 millisec
5.4M packets, BE-flow	$\approx$ 260 millisec	$\approx$ 27,000 sec	$\approx$ 270,000 sec
Improvement (D-flow vs. BE-flow)	$\approx$ 540 times	$\approx$ 5,000 times	$\approx$ 5,000 times

long term traffic flows can be established, i.e., in networks that manage critical infrastructures, or networks that deliver IP video traffic.

F. USING MPLS-LIKE FLOW-LABELS IN SUB-LAYER 3a

Fig. 2 illustrates the proposed deterministic forwarding sub-layer 3a. A key feature of sub-layer 3a is that it *'Transforms the Infrastructure'* as Cisco advocates [8], i.e., sub-layer 3a does not need to perform any complex layer 3 routing or scheduling algorithms at all, and it does not need to process IP packet headers, resulting in a dramatic simplification of the D-switches.

The current BE-IoT includes many network domains which employ *Multi Protocol Label Switching* (MPLS), where each MPLS packet includes a *flow-label* in its header. The proposed paradigm can be modified so that D-switches do perform some simple processing in sub-layer 3a. The D-switches still retain a dramatic simplification compared to a BE-IP router, as they do not perform complex layer 3 routing and scheduling algorithms.

For example, each packet in sub-layer 3a can use a *flow-label* to identify each flow. Flow-labels typically have about 20-24 bits (MPLS flow-labels have 24 bits, and IPv6 flow-labels have 20 bits). Each input port in a D-switch can have a high-speed *flow-table*, with an entry for each possible flow-label. When a packet arrives at an input port, its flow-label is extracted, and used to read a row of the flow-table. The row yields the desired output port for the packet, and a new flow-label to be used for the outgoing packet. The SDN control-plane will maintain the flow-tables in each D-switch. The purposes of this approach are three-fold: (a) to keep the complex layer 3 routing and scheduling algorithms in the SDN control-plane, so that D-switches remain simple and secure; (b) to eliminate the need to synchronize D-switches in sub-layer 3a, as each packet will now carry a flow-label in its header to be used in a lookup-table in each D-switch; (c) To retain the security features of the proposed paradigm (as every packet in a DVPN must pass the *Authorization Check*).

G. MIGRATING SUB-LAYER 3a INTO THE BE IP ROUTERS

In Fig. 2, the deterministic forwarding sub-layer 3a exists below the BE IP routers in layer 3, with the motivation to *'Transform the Infrastructure'*, as Cisco advocates [8]. Nevertheless, it is possible to migrate the sub-layer 3a into the BE-IP routers, by using *Time Division Multiplexing* (TDM), as discussed in section II.H for *Deterministic Ethernet* networks. (The IETF *Converged WAN* also uses a repeating TDM schedule.) Each BE-IP router can absorb one (or more) D-switches, and the D-switches can still be completely controlled by the SDN control-plane. These resulting routers can be called *Integrated Services/Differentiated Services* routers, i.e., *ISDS* routers.

Each IoT link can use a repeating TDM schedule, with alternating windows of time defined for best-effort traffic and deterministic traffic. Suppose an IoT link is configured to support BE traffic 50% of the time, and the windows have a duration of 0.1 millisecond. Assuming a 200 Gbps link, and assuming sub-layer 3a uses packets with 1,024 bytes, then each time-slot in a deterministic window represents $\approx$ 41 nanoseconds, and a 0.1 millisec window will hold $\approx$ 2,439 time-slots. The SDN control-plane can schedule the transmissions for deterministic traffic into these windows, with no significant changes to any algorithms. The SDN control-plane can still retain complete control over the routing and scheduling, and packets arriving at a DVPN at a D-switch must still pass the *Authorization Check*, such that the security properties remain intact. This approach undoes some of the *Transformational Effect* of introducing a separate sub-layer 3a, and it will increase the capital cost of the IoT network by forcing all BE IP routers to be upgraded to ISDS routers, but it retains the very strong security features of the proposed paradigm, and it is a less transformational approach that can be taken.

VIII. CONCLUSION

Cyber-security remains an outstanding challenge [4], and the world needs new ideas for the cybersecurity crisis. This paper has presented the *'Cybersecurity via Determinism'* paradigm for the next-generation *Deterministic IoT*, *Industrial IoT*, and *Tactile IoT*. The paradigm exploits the intersection of 5 distinct topics, including: (i) *Deterministic Communications*, (ii) *Post-Quantum Cryptography*, (iii) *Zero Trust Architectures*, (iv) *Access Control Systems* and (v) *Intrusion Detection Systems*. It is shown that the use of a logically centralized SDN control-plane, with global knowledge of all existing D-flows and link utilizations, can significantly strengthen cybersecurity in the IoT, by explicitly controlling and tracking all communications.

This paradigm introduces a new forwarding sub-layer (3a) of simple and secure D-switches. This sub-layer supports many deterministic SD-WANs, and it gives *Network administrators* 3 tools for enabling deterministic communications, and for significantly improving cybersecurity: *Access Control*, *Rate Control* and *Isolation Control*. This sub-layer also provides hardware support in layers 3 and 4 for the US NIST *Zero Trust Architecture* [18]. An SDN control-plane can embed millions of isolated DVPNs into the SD-WANs of sub-layer 3a, each consisting of many isolated D-flows. The DVPNs are immune to congestion, interference and DDoS attacks. Packets in a DVPN are encrypted from end-to-end using *Quantum Safe* encryption, which is impervious to attacks by *Quantum Computers* using existing quantum algorithms. Even the packet headers are encrypted, and they cannot be viewed or compromised by a cyberattacker. The D-switches do not examine packet headers to make layer-3 routing or scheduling decisions, resulting in a dramatic simplification in hardware complexity. The D-switches do not implement insecure layer 3 protocols, such as the *Berkeley Sockets* and the BGP routing protocols, resulting in a dramatic improvement in security.

The paradigm allows each nation to signifiant strengthen its national security, by reducing the number of cyberattacks against its critical infrastructure. The probability that a DVPN can be compromised by an external cyberattacker can be made arbitrarily small, and effectively zero, by using longer *Quantum Safe* encryption keys. The probability that a DVPN, which has already been compromised by an internal cyberattacker through human error, can compromise a remote DVPN can be made arbitrarily small, and effectively zero, by using longer *Quantum Safe* encryption keys. Generalizations of the *Quantum Safe* AES and RIJNDAEL algorithms to use longer keys have also been presented. The paradigm can save network operators $10s of Billions per year (and potentially $100s of Billions per year) in reduced capital, energy and operational costs, by exploiting a more efficient and easily-controllable deterministic software defined networking infrastructure, which exploits FPGAs. The paradigm can save the global economy a significant fraction of the global costs of cybercrime, which are currently estimated at about $1 Trillion per year in 2020 (and rising to potentially $10 Trillion per year in 2025). In summary, the paradigm can significantly strengthen cybersecurity, well beyond what is possible with today's BE-IoT using existing security protocols.

APPENDIX

A. APPENDIX A: COMMON ACRONYMS

Table 4 briefly summarizes the most common acronyms used in this paper.

B. APPENDIX B: A BE IP ROUTER

A Best-Effort (BE) IP router which interconnects 32 optical fibers (each at 200 Gbps), has a peak throughput of 6.4 Tbps. In a layer-3 BE IP router, the packet headers are processed to make best-effort routing and scheduling decisions, in real-time. Assume that IP packets have an average size of 1K bytes. The BE IP router will process about 781 million packet-headers per second. It may have a queue for data-plane packets waiting for their packet headers to be processed, to perform the routing. It may also have a queue for control-plane packets, which have work to perform (i.e., to implement the BGP routing protocol and update the BGP routing tables). Packet-header processing can consume about 60% of the power of a BE IP router [127].

Each autonomous system in the IoT is associated with a range of IP addresses that it can reach, called a *routing prefix* (or simply a prefix). An IPv4 prefix is typically identified using 4 hexadecimal numbers to represent 32 bits, where the most significant bit values identify the prefix, i.e., the string 192.51.100.0/24 uses the most significant 24 bits of the given IPv4 address as the prefix. This prefix can be obtained from an IPv4 address by using a bit-mask to identify the relevant bits. In this case the bit-mask is 255.255.255.0. Each router maintains a BGP routing table, which associates a preferred outgoing IoT link with each routing prefix that it is aware of. As of 2021, IPv4 had about 861,000 prefixes, and IPv6 had about 109,000 prefixes. Many IP routers have a hardware limit of 1M routing table entries, which is expected to be reached by late 2023. These tables are maintained in a distributed manner, where each router receives an *Update* message from its peers, i.e., where a new preferred link is identified for a routing prefix.

BGP routing in inherently insecure, as it uses the basic insecure IP protocol in layer 3 to maintain BGP routing tables [27]–[29], [131]. In addition, there is a considerable amount of work to do, to perform the routing for a single packet. The destination IP address must be extracted from the packet header, the prefix must be extracted, and it must be compared to all the prefixes stored in the BGP routing table, to see if a match is found. Typically, a BGP routing table may contain up to 1M routing prefixes, and all these prefixes are typically searched in parallel, requiring an extensive amount of hardware and energy. It is estimated that the time involved to perform one BGP table lookup requires 5 nanoseconds [132].

The router will require about 4-8 Gigabytes of high-speed RAM (memory), to store IP routing tables. BE IP routers often use a buffer-sizing rule-of-thumb called the *Bandwidth-Delay-Product* (BDP) rule to store IP packets in a queue awaiting transmission on an outbound link [52]. Assume the mean *Round-Trip-Time* (RTT) is $\approx$ 250 milliseconds, for BE flows traversing a link. Assume the link capacity is $C = 200$ Gbps. According to the BDP rule of thumb, each link requires data buffers for $RTT \cdot C \approx 50$ Gigabits, or $\approx$ 6.25 Gigabytes. Thus, a BE IP router with 32 fiber-optic links would require $\approx$ 200 Gigabytes of high-speed memory to buffer data, and at least 4-8 Gigabytes of memory to store IP routing tables. It would be impossible to fit such a system onto a single ASIC or FPGA.

Reference [52] has presented a *Small Buffer* rule of thumb, which states that each link requires data buffers for

TABLE 4. Commonly used acronyms.

Acronym	Description
ACS	Access Control System
AES	US NIST Advanced Encryption Standard
BE-IoT	the existing Best-Effort (BE) IoT network (see numerous IETF documents)
BE-flow	a Best-Effort traffic flow, in todays BE-IoT, with potentially 10s...100s of millisec of latency
BE IP Router	a Best-Effort IP router used in the BE-IoT, which supports BE-flows
BE-VPN	a layer-3 IPsec 'Virtual Private Network' consisting of multiple partially-encrypted BE-flows in the BE-IoT
BSD, BGP	Berkeley Sockets Distribution, Border Gateway Protocol,
D-IoT	the Deterministic IoT (D-IoT) network in sub-layer 3a proposed in this paper which comprises multiple deterministic SD-WANs
D-flow	a strictly Deterministic traffic flow in the proposed Deterministic IoT, with ultra-low latency
D-switch	a simple deterministic packet switch, which operates in forwarding sub-layer 3a of the proposed Deterministic IoT, and which handles D-flows
D-schedule	a 'fine-grain' deterministic periodic schedule, used to forward packets of D-flows through a D-switch
DetNet Converged-WAN	the IETF DetNet Converged-WAN network (see IETF DetNet documents)
DetNet-Flow	ideally, a Deterministic traffic flow in the IETF DetNet Converged-WAN, with bounded latency (however, strict determinism is subject to vendor's implementation details)
DetNet Converged IP Router	a Converged IP router for the IETF Detnet Converged-WAN, which supports DetNet-flows and BE-flows
DVN	a layer-3 'Deterministic Virtual Network' consisting of multiple un-encrypted D-flows in the D-IoT
DVPN	a layer-3 'Deterministic Virtual Private Network' consisting of multiple fully-encrypted D-flows in the D-IoT
ETSI	European Telecommunications Standards Institute
IETF	Internet Engineering Task Force
ICMP, IGP	Internet Control Message Protocol, Interior Gateway Protocol
IDS, IDPS	Intrusion Detection System, Intrusion Detection and Prevention System
IKEv1, IKEv2	IETF Internet Key Exchange, version 1 and version 2
NIST, NSA	US National Institute for Standards in Technology, US National Security Agency
PKC, PKI	Public Key Cryptography, Public Key Infrastructure
PQC	Post Quantum Cryptography
Quantum Resistant, Quantum Safe	these terms describe 'cryptographic algorithms that run on standard encryption/decryption devices and are widely recognized by experts to be resistant to cryptanalytic attacks from both classical and quantum computers' [72]
SDN, SD-WAN	Software Defined Networking, Software Defined Wide Area Network
TLS	Transport Layer Security (see IETF RFC 8446, Aug. 2018)
ZTA	Zero Trust Architecture

$RTT \cdot C/\sqrt{N}$ bits, where N is the number of long-lived TCP (*Transmission Control Protocol*) flows that traverse the link. Assuming $N = 1,024$, this rule would reduce the amount of buffering required for one IoT link to ≈ 1.56 Gigabits. A BE IP router with 32 links would thus require about 50 Gigabits of buffering, It would be impossible to fit this amount of high-speed memory for buffering on a single integrated circuit.

In contrast, a D-switch can reduce the memory needed for buffering data by 1,000+ times. It can also eliminate the need for a processor and memory, running a Linux operating system and the insecure *Berkeley Sockets* software, to implement insecure layer-3 protocols such as ICMP, BGP and IGP. The simplification allows a D-switch to be implemented in hardware in a single ASIC or FPGA integrated circuit, for a dramatic cost reduction, and it is much easier to secure compared to a complex BE IP router.

C. APPENDIX C–COMMON CYBERATTACKS IN THE BE-IoT

Table 5 illustrates some common cyberattacks in the BE-IoT.

1) DOS AND DDOS ATTACKS

According to Cisco, a DOS (*Denial of Service*) attack will flood a device (or web-server or network) with superfluous traffic, to exhaust resources or network bandwidth, resulting in either degraded performance, or outright service outage [8]. In a DDOS (*Distributed DOS*) attack, multiple compromised devices launch the attack. A *Botnet* is a network of devices which have been compromised with malware, and *Botnets* are typically used in DDOS attacks. The average DDOS attack has an intensity of 1 Gbps, enough to take down most organizations. About 33% of DDOS attacks last about 1 hour, 60% last for one day or less, and 15% last for 1 month or longer. Some motivations for DDOS attacks are: (i) financial gain through extortion, or (ii) gaining a competitive advantage. The top industries targeted by DDOS attacks include: (i) online gaming, (ii) service providers, (iii) cloud services such as AWS and Azure, (iv) governments, (v) financial services, and (vi) online retailers. According to Cisco, the number of DDOS attacks is expected to reach 15.4 million per year by 2023. In Dec. 2021, the Microsoft Azure networking team observed and thwarted a 3.47 Tbps DDOS attack, which is believed to be one of the largest DDOS attacks in history. The attack originated from about 10,000 compromised devices in 10 different countries (please see ZDnet article *Microsoft - Heres How We Stopped the Biggest Ever DDOS Attack*).

2) SPOOFING ATTACK

In a *Spoofing* attack, an entity (a person or computer program or IP packet) masquerades to appear to be from another entity, by falsifying data. The TCP/IP protocol does not use any form

TABLE 5. BE-IoT - common security threats.

Threat/Attack Name	Type of Threat/Attack in Best-Effort IoT	Type of Threat/Attack in Proposed D-IoT	key features (or comments)
DDOS Attack	overload or more servers with many malicious traffic flows from many compromised devices	—- Effectively Eliminated —- in proposed D-IoT	probability of adding one malicious traffic flow can be made arbitrarily small
Spoofing attack	Modify BE-flow packet headers to masquerade as a trusted peer	—- Effectively Eliminated —— the D-IoT does not use packet headers	the D-IoT does not use packet headers
Phishing attack	Provide malicious email or link to malicious website	—- Effectively Eliminated—- links to malicious websites will not be pre-approved	all communications with external entities require pre-approval
Spear Phishing attacks	personalized contact to an individual Provide malicious email or link to malicious website	—- Effectively Eliminated—- links to malicious websites will not be pre-approved	all communications with external entities require pre-approval
Man in the Middle (MITM) attack	cyberattacker is interposed between two communicating entities	—- Effectively Eliminated ——	probability cyberattacker can communicate with both entities can be made arbitrarily small
Replay attack	a valid encrypted packet is observed and recorded, and re-introduced at a later time, as a malicious packet	—- Effectively Eliminated ——	probability of inserting an undetected malicious packet can be made arbitrarily small
Reconnaissance attack (Harvest Now, Decrypt Later attack)	eavesdropping on BE VPN flows	—— Effectively Eliminated ——	all communications in a DVPN use Quantum Safe encryption
Malware attack (ie Remote Code Execution attack) (ie Application Vulnerability attack) (ie Cross Site Scripting attack) (ie Ransomware attack)	vulnerable web-application can render control to cyberattacker under special circumstances	—- Effectively Eliminated for External Cyberattackers —-	all communications with external entities require pre-approval (and valid TLS certificates)

of authentication of the source or destination IP addresses in the IP packets. In an IP spoofing attack, typically the source IP address in the packet header is modified to appear to be from an alternate sender. These attacks can be used to create a *Man-in-the-Middle* (MITM) attack, in which a cyberattacker interposes itself between two communicating entities to intercept their communications without their knowledge. MITM attacks often occur when two entities communicate over unprotected wireless networks. *Spoofing* attacks can be mitigated by using *Next Generation Firewalls*, which perform *Deep Packet Inspection*, or by taking other steps to authenticate the source and destination IP addresses of IP packets. In a *Domain Name System (DNS) Spoofing* attack, an internet domain name is misrepresented in the DNS server (i.e., it maybe spelt incorrectly), causing the victim to visit a malicious web-site, such as a fraudulent bank website (which resembles the legitimate bank website). The cyberattacker can then collect sensitive information from the victim, and also install malware to gain longer-term access to the victim's machine.

3) PHISHING ATTACKS

A *Phishing* attack uses deception, with a goal to steal sensitive user information, such as login credentials and credit card information. An attacker typically masqerades as a trusted entity by sending fraudulent information, and convinces the victim to open an email, or text message, or visit a malicious web-site. As of 2020, *Phishing* was one of the most common cyberattacks. Most phishing attacks are delivered by email to a large pool of targets (i.e., *Bulk Phishing* attacks). These emails may use a spoofed source address, and appear to be from legitimate banks and financial services, email and cloud services providers, or streaming services providers. Some phishing attacks include a link to a malicious website. The malicious website usually appears to be legitimate, and it may prompt the potential victim to log into their account, to address an urgent issue. Google has a service entitled *Google Safe Browsing* (see https://safebrowsing.google.com), which monitors websites and reports websites which may be compromised to their owners. Results are also reported in many web browsers, such as *Google Chrome*. During Oct. 2021, Google reported about 5 million potentially compromised web-pages per week. In Oct. 2021, the average response time for an owner to address the warning was about 47 days.

4) SPEAR PHISHING ATTACKS

In a *Spear Phishing* attack, a specific organization or a specific individual is targeted. These attackers may collect information about the intended target, so that the malicious emails and web-sites are more likely to succeed.

5) MALWARE ATTACK (i.e., REMOTE-CODE-EXECUTION ATTACK)

In a *Malware* or *Remote Code Execution* (RCE) attack, a remote cyberattacker can typically assume control of a victim's computer or device, typically by downloading malicious software to that computer or device. The attacker can then typically execute any OS commands or run arbitrary malicious code on the victim's machine. Typically, the victim's machine has a *Remote Code Execution Vulnerability* in its hardware or software, which a cyberattacker exploits.

TABLE 6. IETF DetNet—most important security threats.

Threat/Attack Name	Type of Threat/Attack in Detnet Converged WAN	Type of Threat/Attack in Proposed Deterministic IoT	key features (or comments)
Delay Attack	Delay Detnet-flow packets	compromise trusted components or compromise SDN control-plane	probability can be made arbitrarily small by using Quantum-Resistant encryption keys
Detnet Flow Header Modification attack (Spoofing attack)	Modify Detnet-flow packet headers	—-Not Applicable—— (D-IoT does not use packet headers)	this attack does not exist in the proposed D-IoT
Detnet Resource Segmentation attack	Detnet-flows are not strictly isolated	—-Not Applicable—- (D-flows are isolated)	this attack does not exist in the proposed D-IoT
Packet Replication (over multiple paths) attack	modify Detnet-flow packet-header or inject malicious packets	compromise trusted components or compromise SDN control-plane	probability can be made arbitrarily small by using Quantum-Resistant encryption keys
Path Choice Modification attack	modify Detnet control-plane packets or inject malicious control-plane packets	compromise trusted components or compromise SDN control-plane	probability can be made arbitrarily small by using Quantum-Resistant encryption keys
Compromised control-plane attack	compromise Detnet control-plane software running on a data-center	compromise SDN control-plane	probability can be made arbitrarily small by using Quantum-Resistant encryption keys
Reconnaissance attack	passive eavesdropping on Detnet-flows	insert undetected untrusted component to eavesdrop	all D-flows are encrypted 1,000s of D-flows share a link
Time-Synchronization attack	compromise IEEE TSN Time Synchronization protocol packets	compromise signals (or encrypted packets) between trusted components	probability can be made arbitrarily small

The US NIST maintains a list of software applications with vulnerabilities in a *National Vulnerability Database (NVD)* (available at: https://nvd.nist.gov/vuln). The MITRE Corporation maintains a system called the *Common Vulnerabilities and Exposures (CVE)* system, which ranks the severity of common vulnerabilities.

6) MALWARE VIA EMAIL ATTACHMENTS

One of the most popular methods of initiating a malware (RCE) attack is via email attachments. Typically, an email to sent to a potential victim as part of a phishing or spear-phishing attack. The email typically has a malicious attachment. According to the MITRE Corp, the attachment may be (a) an executable file, (b) a Microsoft Office document, (c) a PDF file, (d) an archived (.zip or .rar file) or (e) an html file. A Word attachment may use the *Rich Text Format (RTF)*. (The opening of an RTF file often requires the execution of remote code, and hence all emails should only be opened in plaintext mode.) Any of these attached files can contain links (URLs) to compromised websites that the victim is enticed into visiting. The text of the email message usually gives a plausible reason why the victim should open the file/attachment. Upon opening the attachment, the payload exploits a vulnerability on the victim's computer, or directly executes on the victim's computer. (See document MITRE-Phishing: Spearphishing Attachment, Available at: *https* : *attack.mitre.org/techniques/T*1566/001.)

7) RANSOMWARE ATTACK

In this attack, a cyberattacker assumes control of a victim's machine, and downloads software to encrypt the data files accessible from that machine. Some attacks may encrypt entire disks. The victim must typically pay a ransom, often using anonymous cryptocurrency such as *Bitcoin*, to receive a decryption key, which can be used to recover the original data files. According to Cisco, most ransomware infections occur from malicious emails, websites and attachments. To protect against ransomware attacks, Cisco recommends that users enable *MultiFactor Authentication*, use a *Next-Generation Firewall*, and use an *Intrusion Prevention System (IPS)*. They also recommend controlling access to critical resources.

8) CROSS-SITE SCRIPTING ATTACK (XXS)

Some web-applications are vulnerable to *Cross-Site Scripting* attacks, which are a form of RCE attacks. A cyberattacker can inject a *Client-Side Script* into the web pages viewed by other users, thereby bypassing the default ACS, and assume control of the victim's machine. A *Client-Side Script* is a small program that is processed by the web browser of the client. These programs are automatically downloaded, compiled and executed by the web browser, when a client visits a website. Javascript is a popular client-side scripting language.

9) SQL INJECTION ATTACK

According to Cisco, many web-servers use SQL (*Structured Query Language*) in their websites [8]. In an SQL attack, an attacker inserts malicious code into a web-server's user interface, causing the web-server to reveal sensitive information (i.e., user names and passwords). Such attacks are relatively easy to create, as an attacker can submit malicious code into a vulnerable website search box.

D. APPENDIX D: IETF DETNET SECURITY THREATS

The IETF has summarized the most important security threats for its DetNet *Converged-WAN* in [121]. We briefly summarize these threats in Table 6, and explain how they are addressed in the proposed *Deterministic IoT* (D-IoT).

1) DELAY ATTACK

In this attack, packets in a DetNet-flow are delayed by a cyberattacker, causing a deterministic maximum delay bound to be exceeded. The attack can target the data-plane or the control-plane. In the proposed D-IoT, D-flows traverse D-switches according to D-schedules. In order to delay any packets, a cyberattacker must first compromise a secured component (a D-source or D-switch), and then insert a new D-schedule, which delays one or more D-flows. (Alternatively, the cyberattacker must compromise the SDN control-plane, or the DVPN-controller, and then modify the D-schedule of a secured component). By security properties 1 and 2, the probability a cyberattacker can compromise any secured component, by injecting malicious control-plane packets, can be made arbitrarily small by using longer *Quantum Safe* encryption keys.

2) FLOW HEADER MODIFICATION ATTACK

In this attack, the packet header of a DetNet-flow is modified by the cyberattacker. The attack can target data-plane packets or control-plane packets. The proposed D-IoT does not examine the packet headers, and hence this attack does not exist in the proposed D-IoT.

3) RESOURCE SEGMENTATION ATTACK

This attack occurs when resources which should be reserved for DetNet-flows are consumed by BE-flows. For example, a DetNet *Converged IP Router* may have a queue for control-plane packets, which have work to perform (i.e., to implement the BGP protocol and update IP routing tables). If DetNet-flows share this queue with BE-flows, then the DetNet-flows could be delayed if a cyberattacker floods the *Converged IP router* with BE-flows [121]. This attack does not exist in the proposed D-IoT for 2 reasons: (a) The D-IoT does not process packet headers; (b) It strictly isolates D-flows from BE-flows, using sub-layer 3a.

4) PACKET REPLICATION ATTACK

To achieve very high reliability, DetNet can allocate multiple paths to a DetNet-flow, for improved reliability. Each packet is replicated and sent along a different path. This attack disrupts the packet replication process, thereby affecting reliability. In the proposed D-IoT, packets are replicated and transmitted along distinct paths, according to D-schedules. To implement this attack in the D-IoT, a cyberattacker must first compromise a secured component (D-source or D-switch), and then insert a new D-schedule, which disrupts the packet replication handling. (Alternatively, the cyberattacker can compromise the SDN control-plane or DVPN-controller, and then modify the D-schedule of a secured component). By security properties 1 and 2, the probability a cyberattacker can compromise any secured component, by injecting malicious control-plane packets, can be made arbitrarily small by using longer *Quantum Safe* encryption keys.

5) PATH CHOICE ATTACK

This attack disrupts the manner in which distinct paths are selected. In the DetNet *Converged-WAN*, distinct end-to-end paths are established in real-time as needed, using layer-3 signalling (control-plane) packets. The criteria for path selection typically include: path latency, jitter, reliability, and packet loss rate. A cyberattacker could attempt to manipulate these metrics, to affect the path choice. In the proposed D-IoT, the distinct paths are established in advance by the SDN control-plane. To implement this attack in the D-IoT, a cyberattacker must first compromise a secured component (a D-source or a D-switch), and then insert a new D-schedule, which affects the path choice. (Alternatively, the cyberattacker can compromise the SDN control-plane or DVPN-controller, and then modify the D-schedule of a secured component). By security properties 1 and 2, the probability a cyberattacker can compromise any secured component, by injecting malicious control-plane packets, can be made arbitrarily small by using longer *Quantum Safe* encryption keys.

6) COMPROMISED CONTROL-PLANE ATTACK

In this attack, the DetNet control-plane is attacked and compromised. In the proposed D-IoT, by security properties 1 and 2, the probability an external cyberattacker can compromise the SDN control-plane or a DVPN-controller can be made arbitrarily small, by using longer *Quantum Safe* encryption keys.

7) RECONNAISSANCE ATTACK

In this attack, an attacker has gained access to a communications link, and can observe and potentially record the communications. According to [121], this attack is the most difficult attack to detect and to protect against. According to [121], DetNet-flows can be identified by a cyberattacker, since they use un-encrypted packet headers. In the proposed D-IoT, packets in a DVPN are entirely encrypted, and potentially 1,000s of encrypted D-flows share one fiber-optic link. It will be extremely difficult for a cyberattacker to identify which encrypted packets belong to one given D-flow, without decrypting many (or all) of the packets on a link. By security property 1, the probability an external cyberattacker can decrypt even one encrypted packet in a DVPN can be made arbitrarily small. Hence, the threat of this attack is greatly reduced in the proposed D-IoT.

8) TIME SYNCHRONIZATION ATTACK

In the DetNet *Converged-WAN*, all IP routers are tightly synchronized in time, typically to within 1 μsec of accuracy, using the IEEE 802.1 TSN standards. As stated in [16], the tight time synchronization can create a significant load of control-plane packets. All these packets are subject to cyberattacks, and they thereby increase the *'attack surface'*. In this attack, the control-plane packets are compromised to disrupt the tight time synchronization, which can cause DetNet packets to miss their deterministic time bounds.

In the proposed D-IoT, the D-switches are not tightly synchronized in time. A D-switch will receive a *Start-of-Frame (SOF)* signal (or encrypted packet) from each neighbour roughly once every millisecond. The number of control signals (or packets) is negligible. Hence, the *attack surface* is much smaller, i.e., there are far fewer control-plane packets to attack, and the threat of this attack is greatly reduced in the proposed D-IoT. Furthermore, if encrypted packets are used to signal the SOF, the probability a malicious control-plane packet can pass the *Authorization-Check* can be made arbitrarily small, by using longer *Quantum Safe* encryption keys.

ACKNOWLEDGMENT

The author would like to thank the reviewers and subject-area editor for the reviews and feedback, which have helped shape the article.

REFERENCES

[1] P. C. Evens and M. Annunziata. General Electric Corp. (Nov. 2012). *Industrial Internet: Pushing the Boundaries of Minds and Machines*. [Online]. Available: https://ResearchGate.net

[2] *Industrial Internet Insights Report 2015*, GE and Accenture, Boston, MA, USA, 2015. [Online]. Available: https://www.smartindustry.com/assets/Uploads/SI-WP-GE-industrial-internetinsights.pdf

[3] *Industrial Internet of Things: Unleashing the Potential of Connected Products and Services*, World Economic Forum, Cologny, Switzerland. Jan. 2015. [Online]. Available: https://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf

[4] US National Academy of Engineering. *NAE Grand Challenges for Engineering: Secure Cyberspace*. Washington, DC, USA. Accessed: 2008. [Online]. Available: http://www.engineeringchallenges.org/challenges.aspx

[5] M. Ford, "Workshop report: Reducing internet latency, 2013," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 80–86, Apr. 2014.

[6] J. Gettys, "Bufferbloat: Dark buffers in the internet," *ACM Queue-Virtualization*, vol. 9, pp. 40–54, Nov. 2011.

[7] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-host congestion control for TCP," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 304–342, 3rd Quart., 2010.

[8] *CISCO Annual Internet Report (2018-2023)*, CISCO, San Jose, CA, USA. [Online]. Available: https://cisco.com/c/en/us/solutions/collateral…/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

[9] I. Butun, P. Osterberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 44–616, Nov. 2019.

[10] S. Kent and K. Seo, "Security architecture for the internet protocol," IETF, Fremont, CA, USA, Tech. Rep. RFC 4301, Dec. 2005.

[11] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thome, L. Valenta, and B. VanderSloot, "Imperfect forward secrecy: How Diffie-Hellman fails in practice," in *Proc. ACM Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 5–17.

[12] A. Singla, B. Chandrasekaran, P. B. Godfrey, and B. Maggs, "The internet at the speed of light," in *Proc. ACM Hotnets*, Oct. 2014, pp. 1–7.

[13] G. Fettweis, H. Boche, T. Wiegand, E. Zielinski, H. Schotten, P. Merz, S. Hirche, A. Festag, W. Haffner, M. Meyer, E. Steinback, R. Kraemer, R. Steinmetz, F. Hofmann, P. Eisert, R. Scholl, F. Ellinger, E. Weib, and I. Reidel, "The Tactile Internet," Int. Telecommun. Union, Geneva, Switzerland, ITU-T Technol. Watch Rep., Tech. Rep., Aug. 2014.

[14] M. Maier, M. Chowdhury, B. P. Rimal, and D. P. Van The, "Tactile internet: Vision, recent progress, and open challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 45–138, May 2016.

[15] IEEE 802 Tutorial. (2012). *Deterministic Ethernet: 802.1 Standards for Real-Time Process Control, Industrial Automation, and Vehicular Networks*. [Online]. Available: https://www.ieee802.org

[16] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, Sep. 2019.

[17] T. H. Szymanski, "Supporting consumer services in a deterministic industrial internet core network," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 110–117, Jun. 2016.

[18] (Aug. 2020). US National Institute for Standards and Technology. *Zero Trust Architecture*. [Online]. Available: http://csrc.nist.gov/publications

[19] A. Kerman, O. Borchert, S. Rose, and A. Tan, "Implementing a zero trust architecture," MITRE Corp., McLean, Virginia, USA, Tech. Rep., Oct. 2020.

[20] T. H. Szymanski, "An ultra low latency guaranteed-rate internet for cloud services," *IEEE Trans. Netw.*, vol. 24, no. 1, pp. 36–123, Feb. 2016.

[21] T. H. Szymanski, "Securing the industrial-tactile Internet of Things with deterministic silicon photonics switches," *IEEE Access*, vol. 4, pp. 8236–8249, 2016.

[22] T. H. Szymanski, "Security and privacy for a green Internet of Things," *IEEE IT Prof.*, vol. 19, no. 5, pp. 34–41, Oct. 2017.

[23] E. Rescorla, "The transport layer security (TLS) protocol version 1.3," *IETF RFC*, vol. 8446, pp. 1–160, Aug. 2018.

[24] A. Odlyzko, "Data networks are lightly utilized, and will stay that way," *Rev. Netw. Econ.*, vol. 2, no. 3, Jan. 2003.

[25] A. Hassidim, D. Raz, M. Segalov, and A. Shaqed, "Network utilization: The flow view," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1429–1437.

[26] CISCO. *Global—2021 Forecast Highlights*. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/…vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf

[27] K. Sriram and D. Montgomery, "NIST Special Publication 800-189, Resilient interdomain traffic exchange: BGP security and DDOS mitigation,", pp. 1-70, Dec. 2019. Available at:. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-189

[28] O. Borchert, K. Lee, K. Sriram, D. Montgomery, P. Gleichmann, and M. Adalier, "BGP secure routing extension (BGP-SRx): Reference implementation and test tools for emerging BGP security standards," NIST, Gaithersburg, MD, USA, NIST Technical Note 2060, Sep. 2021, pp. 1–149. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/TechnicalNotes/NIST.TN.2060.pdf, doi: 10.6028/NIST.STN.2060.

[29] M. Lepinski and K. Sriram, "BGPsec protocol specification," IETF, Fremont, CA, USA, Tech. Rep. RFC 8205, Sep. 2017.

[30] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[31] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[32] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, Apr. 2019.

[33] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A survey on privacy and security of Internet of Things," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100312.

[34] N. M. Karie, N. M. Sahri, W. Yang, C. Valli, and V. R. Kebande, "A review of security standards and frameworks for IoT-based smart environments," *IEEE Access*, vol. 9, pp. 121975–121995, 2021.

[35] Y. Xin, L. Kong, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[36] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, Apr. 2020.

[37] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1686–1721, Apr. 2020.

[38] O. S. Althobaiti and M. Dohler, "Cybersecurity challenges associated with the Internet of Things in a post-quantum world," *IEEE Access*, vol. 8, pp. 157356–157381, 2020.

[39] (2020). IBM Security. *Artificial Intelligence (AI) for Cybersecurity: Beyond the Hype: AI in Your SOC*. [Online]. Available: https://www.ibm.com/downloads/cas/9EDONM6M

[40] R. Braken, D. Clark, and S. Shenker, "Integrated services in the internet architecture—An overview," IETF, Fremont, CA, USA, Tech. Rep. RFC 1633, Jul. 1994.
[41] J. Wroclawski, "The use of RSVP with IETF integrated services," IETF, Fremont, CA, USA, Tech. Rep. RFC 2210, Sep. 1997.
[42] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service," IETF, Fremont, CA, USA, Tech. Rep. RFC 2212, Sep. 1997.
[43] S. Shenker and J. Wroclawski, "General characterization parameters for integrated service network elements," IETF, Fremont, CA, USA, Tech. Rep. RFC 2215, Sep. 1997.
[44] S. Shenker and J. Wroclawski, "Network element service specification template," IETF, Fremont, CA, USA, Tech. Rep. RFC 2216, Sep. 1997.
[45] D. Black and P. Jones, "Differentiated services (DiffServ) and real-time communications," IETF, Fremont, CA, USA, Tech. Rep. RFC 7657, Nov. 2015.
[46] X. Xiao and L. M. Ni, "Internet QoS: A big picture," *IEEE Netw.*, vol. 13, no. 2, pp. 8–18, Mar. 1999.
[47] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 29–40, Jan. 2004.
[48] A. Meddeb, "Internet QoS: Pieces of the puzzle," *IEEE Commun. Mag.*, vol. 48, no. 1, pp. 86–94, Jan. 2010.
[49] G. Nong and M. Hamdi, "On the provision of quality-of-service guarantees for input queued switches," *IEEE Commun. Mag.*, vol. 38, no. 12, pp. 62–69, Dec. 2000.
[50] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Feb. 1993.
[51] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," *IEEE/ACM Trans. Netw.*, vol. 2, no. 2, pp. 137–150, Apr. 1994.
[52] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 281–292, Aug. 2004.
[53] S. Iyer, R. R. Kompella, and N. McKeown, "Designing packet buffers for router linecards," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 705–717, Jun. 2008.
[54] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.
[55] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.
[56] C. E. Koksal, R. G. Gallager, and C. E. Rolus, "Rate quantization and service quality over single crossbar switches," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1962–1973.
[57] C.-S. Chang, W.-J. Chen, and H.-Y. Huang, "Birkhoff-von Neumann input-buffered crossbar switches for guaranteed-rate services," *IEEE Trans. Commun.*, vol. 49, no. 7, pp. 1145–1147, Jul. 2001.
[58] I. Keslassy, M. Kodialam, T. V. Lakshman, and D. Stiliadis, "On guaranteed smooth scheduling for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 13, no. 6, pp. 1364–1375, Dec. 2005.
[59] C.-S. Chang, D.-S. Lee, and C.-Y. Yue, "Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 644–656, Jun. 2006.
[60] T. H. Szymanski, "A low-jitter guaranteed-rate scheduling algorithm for packet-switched ip routers," *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3446–3459, Nov. 2009.
[61] T. Szymanski and D. Gilbert, "Provisioning mission-critical telerobotic control systems over internet backbone networks with essentially-perfect QoS," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 5, pp. 630–643, Jun. 2010.
[62] T. H. Szymanski, "Max-flow min-cost routing in a future-internet with improved QoS guarantees," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1485–1497, Apr. 2013.
[63] T. H. Szymanski, "Method to achieve bounded buffer sizes and quality of service guarantees in the internet network," U.S. Patent 9 584 431, B2, Feb. 28, 2017.
[64] T. H. Szymanski, "Reduced-complexity integrated guaranteed-rate optical packet switch," U.S. Patent 10 687 128, Jun. 2, 2020.
[65] B. Liu, S. Ren, C. Wang, V. Angilella, P. Medagliani, S. Martin, and J. Leguay, "Towards large-scale deterministic IP networks," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, Jun. 2021, pp. 1–9.
[66] J. Krolikowski, S. Martin, P. Medagliani, J. Leguay, S. Chen, X. Chang, and X. Geng, "Joint routing and scheduling for large-scale deterministic IP networks," *Comput. Commun.*, vol. 165, pp. 33–42, Jan. 2021.
[67] F. Arute *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019. [Online]. Available: https://www.nature.com/articles/s41586-019-1666-5.pdf, doi: 10.1038/s41586-019-1666-5.
[68] R. A. Perlner and D. A. Cooper, "Quantum resistant public key cryptography: A survey," in *Proc. 8th Symp. Identity Trust Internet*, Apr. 2009, pp. 85–93.
[69] *Quantum safe Virtual Private Networks*, Standard ETSI TR 103 617 v1.1.1, ETSI (European Telecommunications Standards Institute), Aug. 2018. [Online]. Available: https: www.etsi.org/standards-search
[70] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
[71] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer, 2006.
[72] *Quantum Computing and Post Quantum Cryptography, FAQs (Frequently Asked Questions)*, document PP-21-1120, US NSA (National Security Agency), Aug. 4, 2021, [Online]. Available: https://media.defense.gov/2021/Aug/04/2002821837/-1/-1/1/Quantum_FAQs_20210804.PDF
[73] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.
[74] C. Gidney and M. Ekerå, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits," *Quantum*, vol. 5, p. 433, Apr. 2021.
[75] É. Gouzien and N. Sangouard, "Factoring 2048-bit RSA integers in 177 days with 13 436 qubits and a multimode memory," *Phys. Rev. Lett.*, vol. 127, no. 14, Sep. 2021, Art. no. 140503.
[76] K. Li, P.-G. Yan, and Q.-Y. Cai, "Quantum computing and the security of public key cryptography," *Fundam. Res.*, vol. 1, no. 1, pp. 85–87, Jan. 2021.
[77] K. Li and Q.-Y. Cai, "Practical security of RSA against NTC-architecture quantum computing attacks," *Int. J. Theor. Phys.*, vol. 60, no. 8, pp. 2733–2744, Aug. 2021.
[78] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, p. 325, Jul. 1997.
[79] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, May 1996, pp. 212–219.
[80] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM J. Comput.*, vol. 26, no. 5, pp. 23–1510, Oct. 1997.
[81] G. Castagnoli, "Highlighting the mechanism of the quantum speedup by time-symmetric and relational quantum mechanics," *Found. Phys.*, vol. 46, no. 3, pp. 360–381, Mar. 2016.
[82] G. L. Long, "Grover algorithm with zero theoretical failure rate," *Phys. Rev. A, Gen. Phys.*, vol. 64, no. 2, Jul. 2001, Art. no. 022307.
[83] F. M. Toyama, W. van Dijk, and Y. Nogami, "Quantum search with certainty based on modified Grover algorithms: Optimum choice of parameters," *Quantum Inf. Process.*, vol. 12, no. 5, pp. 1897–1914, May 2013.
[84] X. Bonnetain, M. Naya-Plasencia, and A. Schrottenloher, "Quantum security analysis of AES," *IACR Trans. Symmetric Cryptol.*, pp. 55–93, Jun. 2019.
[85] M. Grassi, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying grover's algorithm to AES: Quantum resource estimates," in *Post-Quantum Cryptography*. Cham, Switzerland: Springer, Feb. 2016, pp. 29–43.
[86] L. Chen, S. Jordan, Y. K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. (Apr. 2016). *Report on Post-Quantum Cryptography*. U.S. NIST Interagency/Internal Report (NISTIR). [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf
[87] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y. K. Liu, C. Miller, D. Moody, R. Peralta, and R. Perlner, "Status report on the first round of the NIST post-quantum cryptography standardization process," U.S. NIST Interagency/Internal Rep. (NISTIR), Gaithersburg, MD, USA, Tech. Rep. 8240, Jan. 2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf, doi: 10.6028/NIST.IR.8240.

[88] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y. K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone, "Status report on the second round of the NIST post-quantum cryptography standardization process," U.S. NIST Interagency/Internal Rep. (NISTIR), Gaithersburg, MD, USA, Tech. Rep. NISTIR 8309, Jul. 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf, doi: 10.6028/NIST.IR.8105.

[89] *Quantum Safe Public Key Encryption and Key Encapsulation*, Standard ETSI TR 103 823 v1.1.2, ETSI (European Telecommunications Standards Institute), Oct. 2021. [Online]. Available: https: www.etsi.org/standards-search

[90] S. Fluhrer, P. Kampanakis, D. McGrew, and V. Smyslov, "Mixing preshared keys in IKEv2 for post quantum security," IETF, Fremont, CA, USA, Tech. Rep. RFC 8774, Jan. 2020.

[91] *Quantum Safe Signatures*, Standard ETSI TR 103 616 v1.1.2, ETSI (European Telecommunications Standards Institute), Sep. 2021. [Online]. Available: https: www.etsi.org/standards-search

[92] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: Towards a unified standard," in *Proc. ACM Workshop Role-Based Access Control*, vol. 10, Jul. 2000, pp. 1–17.

[93] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," *IEEE Comput.*, vol. 43, no. 6, pp. 79–81, Jun. 2010.

[94] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (ABAC) definition and considerations (draft)," NIST, Gaithersburg, MD, USA, Tech. Rep., Apr. 2013, pp. 1–54. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-162.pdf

[95] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *IEEE Comput.*, vol. 16, no. 2, pp. 85–88, Feb. 2015.

[96] (Apr. 2021). Institute for Security and Technology (IST). *Ransomware Task Force: Combatting Ransomware—A Comprehensive Framework for Action: Key Recommendations From the Ransomware Task Force*. [Online]. Available: https://securityandtechnology.org/ransomwaretaskforce/report/

[97] (May 12, 2021). The White House. *Executive Order on Improving the Nation's Cybersecurity*. [Online]. Available: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

[98] (2001). US National Institute of Standards and Technology (NIST). *Announcing the Advanced Encryption Standard (AES)*. [Online]. Available: https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines

[99] J. Daemen and V. Rijmen. (1999). *AES Proposal: Rijndael*. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

[100] J. Daemen and V. Rijmen, *The Design of Rijndael*. New York, NY, USA: Springer-verlag, Mar. 2002.

[101] Y. Nir and A. Langley, "Chacha20 and poly1305 for IETF protocols," IETF, Fremont, CA, USA, Tech. Rep. RFC 7539, May 2015, p. 45.

[102] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May 1994.

[103] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Comput. Netw.*, vol. 31, no. 8, pp. 805–822, Apr. 1999.

[104] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," *Comput. Commun.*, vol. 49, pp. 1–17, Aug. 2014.

[105] M. Masdari and H. Khezri, "A survey and taxonomy of the fuzzy signature-based intrusion detection systems," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106301.

[106] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, Feb. 2009.

[107] V. Jyothsna, V. R. Prasad, and K. M. Prasad, "A review of anomaly based intrusion detection systems," *Int. J. Comput. Appl.*, vol. 28, no. 7, pp. 26–35, 2011.

[108] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *Eai Endorsed Trans. Secur. Saf.*, vol. 3, no. 9, e2, 2016.

[109] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[110] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[111] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[112] R. Bace and P. Mell. (Nov. 1, 2001). *NIST Special Publication on Intrusion Detection Systems*. [Online]. Available: https://www.nist.gov/publications

[113] K. Scarfone and P. Mell. (Feb. 2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST Special Publication 800-94. [Online]. Available: https://www.nist.gov/publications

[114] A. Hulsing, K. C. Ning, P. Schwabe, F. J. Weber, and P. R. Zimmermann, "Post-quantum wireguard," in *Proc. IEEE Symp. Security Privacy (SP)*, May 2021, pp. 304–321.

[115] E. Crockett, C. Paquin, and D. Stebila, "Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH," IACR Cryptol. ePrint Arch., Int. Assoc. Cryptol. Res., Lyon, France, Tech. Rep. 2019/858, Aug. 2019, p. 858. [Online]. Available: https://eprint.iacr.org/2019/858.pdf

[116] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, "Post-quantum authentication in TLS 1.3: A performance study," IACR Cryptol. ePrint Arch., Int. Assoc. Cryptol. Res., Lyon, France, Tech. Rep. 2020/071, Jan. 2020, p. 71. [Online]. Available: https://eprint.iacr.org/2020/071.pdf

[117] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: Deterministic IP-enabled industrial Internet (of Things)," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 36–41, Dec. 2014.

[118] N. Finn and P. Thubert, "Deterministic networking problem statement," (IETF Internet-Draft, Standards Track)," Fremont, CA, USA, Tech. Rep. RFC 8557, May 2019.

[119] N. Finn, P. Thubert, B. Varga, and J. Farkas, "Deterministic networking architecture," IETF Internet, Fremont, CA, USA, Tech. Rep. RFC 8655, Oct. 2019.

[120] E. Grossman, "Deterministic networking use cases," IETF Draft, Fremont, CA, USA, Tech. Rep. RFC 8578, May 2019.

[121] E. Grossman, T. Mizrahi, and A. Hacker, "Deterministic networking (DetNet) security considerations," IETF, Fremont, CA, USA, Tech. Rep. RFC 9055, Jun. 2021.

[122] P. Wetterwald and J. Raymond, "Deterministic networking utilities requirements," IETF Internet Draft, Fremont, CA, USA, Tech. Rep., Jan. 2016, pp. 1–26. [Online]. Available: https://www.ietf.org/archive/id/draft-wetterwald-detnet-utilities-reqs-02.txt

[123] S. Shah and P. Thubert, "Deterministic forwarding PHB (04)," IETF Internet Draft, Fremont, CA, USA, Tech. Rep., Aug. 2015. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-svshah-tsvwg-deterministicforwarding-04

[124] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Comput. Netw.*, vol. 12, pp. 93–279, Jan. 2017.

[125] F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN control: Survey, taxonomy, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 54–333, Jan. 2018.

[126] Y. A. Vlasov, "Silicon-CMOS integrated nano-photonics for computer and data-communications Beyond 100G," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 67–72, Feb. 2012.

[127] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 223–244, May 2011.

[128] (Dec. 23, 2021). Cybersecurity and Infrastructure Security Agency (CISA). Alert (AA21-356A). *Mitigating Log4shell and Other Log4j-Related Vulnerabilities*. [Online]. Available: https://www.cisa.gov/uscert/ncas/alerts/aa21-356a

[129] (Dec. 10, 2021). NIST NVD (National Vulnerability Database). CVE-2021-44228 Detail. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2021-44228

[130] M. Hadad. (Dec. 12, 2021). *Apache Log4j Vulnerability CVE-2021-44228 Raises Widespread Concern*. [Online]. Available: https://blogs. juniper.net/en-us/security/apache-log4j-vulnerability-cve-2021-44228-raises-widespread-concerns.

[131] G. Huston. (Aug. 2021). *A Survey on Secure Inter-Domain routing Part 2—Approaches to Securing BGP*. [Online]. Available: https://labs.apnic.net/?p=1467
[132] G. Huston. (Jan. 2020). *BGP in 2019—The BGP Table*. [Online]. Available: https://blog.apnic.net/2020/01/14/bgp-in-2019-the-bgp-table/

TED H. SZYMANSKI (Member, IEEE) received the Ph.D. degree in ECE from the University of Toronto. From 1987 to 1991, he was a Professor with the Department of Electrical Engineering, Columbia University, working within the NSF-sponsored Center for Telecommunications Research. From 1991 to 1998, he was a Professor with the Department of Electrical and Computer Engineering, McGill University, working within the Networks of Centers of Excellence (NCE) Program of Canada. Since 1999, he has been a Professor with the Department of Electrical and Computer Engineering, McMaster University. From 2001 to 2011, he held the Bell Canada Chair in Data Communications at McMaster University. From 1993 to 2003, he led the Optical Architectures project within a ten-year national research program funded by the NCE Program of Canada, which demonstrated a free-space "intelligent optical backplane" using CMOS/SEED photonic packet-switches with about 1K optical channels. He holds a patent on the basic photonic backplane architecture. Collaborators included Nortel Networks (now Ericsson), Newbridge Networks (now Alcatel), Lockheed-Martin/Sanders, and McGill, McMaster, Toronto, and Heriot-Watt Universities. His research group also demonstrated the first photonic FPGA (a CMOS logic substrate integrated with many microscopic optical IO), fabricated through the U.S. ARPA/Lucent/Coop CMOS/SEED foundry service. His current interests include cybersecurity, deterministic communications, the deterministic IoT, smart cyber-physical systems, energy efficiency, and the industrial and tactile Internet of Things. He holds a number of patents in the area of deterministic networks and the deterministic IoT. He is listed in the top 2% of researchers in the field of networking and telecommunications, according to a database recently compiled by Stanford University.