2018 IEEE 4th International Conference on Collaboration and Internet Computing

Envisioning an Information Assurance and Performance Infrastructure for the Internet of Things

Jamie Payton, Xiaojiang Du, Xubin He, and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA, USA
{payton, dux, xubin.he, jiewu}@temple.edu

***Abstract*—The Internet of Things (IoT), in which sensing and actuation is embedded in everyday objects connected via the Internet, has the potential to support an increased level of intelligent, dynamic decision making cross a wide array of domains, such as smart cities, intelligent agriculture, and emergency response management. However, IoT systems are vulnerable to security threats, which limits their widespread adoption. In addition, in the future, systems will be challenged by the large amounts of IoT data communicated over the network and stored in cloud-based data centers. In this vision paper, outline a roadmap for innovative research on IoT security and performance, including the creation of secure communication protocols, IoT network threat detection, elastic computing algorithms, and computational offloading in IoT systems.**

***Index Terms*—Internet of Things (IoTs), performance, security**

I. Introduction

The deployment of Internet of Things (IoT) devices is experiencing rapid growth and is expected to reach between 20 to 30 billion connected devices by the year 2020 [1]. Networked systems that integrate IoT devices have the potential to support intelligent cyber-human systems across a variety of application domains. For example, IoT-enabled smart city deployments can improve traffic management, pedestrian safety, parking, energy consumption, and emergency response. Other envisioned applications include intelligent inventory control, supply chain management, agriculture, and transportation. In order to realize the potential of these and other IoT-enabled systems, it is necessary to address two primary concerns:

IoT security. A key challenge is the ability to protect sensitive data in IoT networks and to secure IoT-enabled networks, particularly when used to support safety critical applications like emergency response. Because IoT-enabled devices are often resource-limited in terms of computational power and storage, it is not straightforward to apply existing security architectures and protocols. Currently, there is no end-to-end solution for secure communication across a large network of IoT devices. New approaches to securely authenticate users to, pair with, and bind to IoT devices are needed, along with end-to-end communication schemes. Furthermore, IoT devices are susceptible to new kinds of threats that target their specific capabilities [2], such as GPS spoofing attacks [3] that compromise location-aware operation in IoT systems. New network threat detection approaches must be developed to address IoT-specific vulnerabilities.

IoT performance. IoT deployments generate streaming data at large scales over extended time frames. As a result, they are typically supported by a cloud-based infrastructure. Within cloud-based data centers, open research challenges remain in storing large amounts of data reliably and efficiently. Additionally, task-resource allocation remains an open issue. Often, allocation of resources is static, requiring *a priori* knowledge of program behavior. However, the computational load, communication load, and the volume of data produced in IoT systems is highly dynamic. For example, on a smart campus, the population may sharply increase when there is a basketball game held in the campus arena. As fans enter the arena for the game, they trigger sensors (e.g., security cameras, acoustic sensors that monitor crowd excitement, floor pressure sensors that detect stationary and mobile groups) and actuators; as a result, the volume of IoT data will sharply increase and demand for network communication changes. Similarly, the need for other kinds of resources can rapidly and dramatically vary over time in response to changes in the IoT network. For example, in a campus emergency response scenario, increased computational resources may be needed to rapidly track persons of interest using wearable body cameras and campus security cameras.

The exploration of IoT systems research to address security and performance challenges requires implementation, deployment, and evaluation on networks of cloud servers, edge nodes, and IoT devices. While simulation is a useful tool, existing network simulators do not include methods for generating realistic sensor data and environmental context for IoT devices, particularly when nodes are mobile. Publicly accessible network testbeds for mobile and IoT devices, such as Fit IoT lab [4] and SmartSantander [5], include a limited range of IoT sensors, and do not consider the mobility of IoT devices carried or worn by humans. In addition, existing IoT testbeds and publicly accessible cloud infrastructures (e.g., Amazon Web Services) do not support experimentation with security protocols, vulnerabilities, or network attacks.

We contend that a publicly accessible research infrastructure for Information Assurance and Performance in the Internet of Things (IAP-IoT) is needed. Such an infrastructure would enable a number of innovative research projects on IoT security and performance, as well as on the related supporting technologies, such as robust data storage schemes and elastic provisioning algorithms for cloud-based IoT systems. In this vision paper, we outline a set of research challenges that can

978-1-5386-9502-9/18/$31.00 ©2018 IEEE
DOI 10.1109/CIC.2018.00043

be supported by the development and deployment of such an infrastructure.

II. Security Research Challenge: Secure IoT Communication

IoT holds promise for enabling more intelligent and responsive applications that improve our daily lives, from enabling improved traffic management, emergency response, and water and energy infrastructure in smart cities to enabling smart refrigerators that order groceries and smart showers that detect fall incidents in smart homes. A defining characteristic of these and other IoT-enabled systems is the ability to sense and act upon conditions within the environment. Often, that means collecting private and sensitive information about an individual's activities, habits, and location. As such, it is important to consider the privacy and security of IoT data.

IoT devices are small and resource-constrained, which limits the set of solutions that can be deployed on them; for example, it is often not feasible to deploy computationally-intensive cryptographic operations that are the standard in conventional security protocols. New, lightweight approaches for end-to-end communication are needed to ensure: (1) secure pairing, such that only an authorized user can pair a controlling application with the IoT device; (2) secure binding, which allows only the authorized user to use the paired controlling application to configure the device; and (3) end-to-end encryption of communication with the IoT device.

Secure pairing. It is a common practice in industry that at bootstrapping, an IoT device is open for any user to pair with the device. Such openness allows for flexibility, but is also a potential security hazard. How can we allow a user to claim ownership of and securely connect to the IoT device? One option is to require co-location with the physical IoT device at the time of pairing, using an approach that requires visual inspection of a "secret" displayed on the device. For example, instead of working with an open Access Point, the IoT device can use Wi-Fi Protected Access II (WPA2) and display a onetime passcode on a LED. The user can use this passcode to connect to the IoT device. Once a user is paired with the IoT device and configuration is done, the IoT device cannot be reset and will not get into the AP mode displaying a onetime password unless the user explicitly un-pairs herself and the device. Once the user relinquishes her ownership, the device is reset to the factory setting and another user can take the ownership, pair with the IoT device, configure and use it.

Secure binding. During binding, the paired controlling application can configure the IoT device and connect it to the Internet and to its authorized users. A series of authentication steps can be used for developing an approach to secure binding between the user and an IoT device: a. An IoT device authenticates a user for the purpose of device operation; b. The IoT device authenticates a publish/subscribe server for its genuineness; c. The user authenticates a publish/subscribe server for its genuineness; d. The publish/subscribe server authenticates the IoT device for the use of the server; e. The publish/subscribe server authenticates the user for the use of the server; f. The user authenticates the device for the use of the device resources (if there is such a need).

End-to-end encryption. We want end-to-end encrypted communications between a user and a device so that the server does not know the content for user privacy. One challenge is how the user and device share keys. A potentially viable approach is pre-shared-key scheme [6]. That is, during secure binding, the user and the device establish a pre-shared-key that will be used later to encrypt their communication. If the user gets a certificate from the device, then the RSA-based or authenticated Diffie-Hellman key exchange can be used. That is, during the key exchange process, the user and device sign the messages they send to the other party. Research is needed to study how to generate and update keys for encryption and integrity from the pre-shared-key.

III. Security Research Challenge: Detecting IoT Network Threats

Even with secure end-to-end protocols, we can expect that IoT-enabled networks will be subject to cyberattacks. Although there are a wide range of detection methods for traditional network attacks, IoT devices are small and resource-constrained, which limits the set of solutions that can be deployed on them. To combat this, a number of IoT-specific approaches to detecting network threats have been developed [7], [8], but most result in significant energy consumption or require supplementary hardware that cannot be retrofitted for use in existing IoT deployments. Furthermore, the use of IoT devices introduces new system vulnerabilities and opportunities for cyberattacks. For example, IoT applications often rely on location-aware services, typically supported by GPS; since small, resource-constrained IoT devices rely on unencrypted GPS signals, they are highly susceptible to GPS spoofing.

It is particularly important to detect and mitigate such threats in IoT networks, as they are increasingly introduced to support safety critical applications like emergency response, water filtration plants, and energy grids. Behavior-based network security threat detection approaches, which derive models of normal, expected behavior from a history of network activity, have been shown to be effective for detecting a broad range of known and unknown attacks in traditional networks (see [9] for survey). However, these approaches are limited in their ability to identify attacks that are specific to mobile and IoT-enabled applications, such as GPS spoofing.

As a starting point for addressing these limitations, we envision a context-aware approach that leverages sensing and actuation on IoT devices to detect emerging threats, like GPS spoofing, in IoT systems. A key insight is that users often act in particular roles for a given IoT application; those roles often have a predictable behaviors in a given application context. For example, an emergency responder has a predictable set of physical actions (e.g., administer oxygen, transport supplies) and associated data accesses and application service interactions (e.g., record vital signs, request additional resources) in a given setting (e.g., co-located with victim, at Mobile

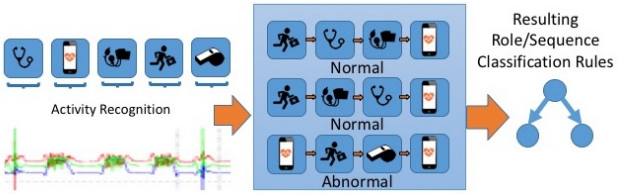

Fig. 1: Role-based behavioral threat detection for IoT

Operations Command Center) given their particular role (e.g., Lead Emergency Medical Technician, Incident Commander).

We believe that a promising approach is to extend behavior-based threat detection approaches to incorporate the use of environmental context, application usage, and network communication for a user acting in a given role in an IoT-enabled network. In particular, role-based behavioral models that include *physical actions* can help to detect anomalous behaviors that represent potential network threats. By examining sensor data from IoT devices, we can learn models that detect anomalous sequences of physical activities, application usage, and network communications. For example, we can derive an expected behavior pattern in which an emergency responder may first collect first aid equipment, apply a blood pressure cuff to a co-located victim, access medical records and report vital signs to a nearby medical facility, and apply oxygen. From such derived patterns, we can find anomalies. For example, a responder administering first aid to a co-located victim should not access criminal records; the anomaly may indicate that the users account has been compromised.

As illustrated in Figure 1, we apply a multi-phase machine learning approach to develop a framework for role-based behavioral threat detection. The first phase of our approach applies supervised machine learning techniques to wearable IoT sensor data to develop a model of component physical activities (e.g., sweep flashlight, administer oxygen mask to victim) performed by users in IoT networks. In the second phase, we apply a sequence detection technique to identify sequences of component actions that comprise a role-based application-specific activity. For example, administering first aid may consist of: using a light to check a victims pupil dilation, checking a victims blood pressure, checking a victims heart rate, using an emergency response application to check the victims medical history, administering oxygen, and using the emergency application to schedule transport for the victim to a nearby medical treatment facility. Once these sequences are identified, a rule inferencing algorithm [10] is applied to identify normal, expected role-based behavioral sequences. These behavioral sequences include physical actions, physical location, application usage, and network communication. The learned model of role-specific sequences is then used to detect anomalous behaviors for a given role in IoT networks.

In a pilot study, 9 participants played a role in a simulated exercise to search for a lost hiker. The protocol included using a shoulder radio, sending their location, searching with a flashlight, and blowing a whistle at certain points in the search. When the hiker was found, participants were instructed to use an application to request backup, blow a whistle, perform a high wave to signal other personal, check eyes, take blood pressure, and use the app to send vitals. We applied a random forest classification algorithm to sensor data collected from accelerometers and gyroscopes embedded on smartwatches to identify physical actions that are performed by the participants. We then evaluated our rule inferencing approach to detect anomalous activity sequences that indicate a potential network threat. We injected two types of attacks into the network traffic: a GPS spoofing attack and a data exfiltration attack that is enabled by the compromise of a user account. Because attacks are typically rare, we include a small number of attack-related samples: only 15 instances of attack-related behaviors out of thousands. Our results show promise for detecting anomalous behaviors in IoT-enabled networks, with an average accuracy of 96% for classifying anomalous and normal behaviors.

While the pilot study shows that role-based behavioral approaches shows promise in IoT network threat detection, a more extensive exploration is needed. To support approaches like this one, research challenges include:

Developing a scalable approach to labeling ground truth activity sequences. To generalize the approach and support exploration across a broad range of domains, activities, and additional potential threats, data collection exercises are needed with large numbers of participants, actions, and activity sequences. Even when testbeds incorporate wearable IoT sensors that can be used for activity sequence recognition, it is typically necessary to manually capture ground truth labels of actions and activity sequences. This is a considerable challenge when performed at a large scale. To reduce the burden of manual labeling, active learning for activity recognition [**?**] can be applied, intelligently prompting users to provide labels for recently performed activity sequences that are not adequately captured by our learned model. Using active learning in a way that minimizes user interruptions and maximizes correct self-labeling is an open challenge. Vetted, large scale data sets are needed by the research community for benchmarking and to reduce the overhead associated with data collection. However, the activities of interest may vary across projects; as such, another open challenge is to promote multi-model active learning, where several relevant activities could be selected to correctly label for a single action performed by the user.

Developing a probabilistic approaches for identifying anomalous activity sequences. The approach outlined here generates a deterministic set of rules for identifying normal and abnormal role-based behaviors. However, a more flexible approach would incorporate probabilistic models for detecting potential anomalous activities.

Exploring deep learning approaches for identifying anomalous activity sequences. Deep learning methods offer a powerful approach to identifying patterns and learning models for anomaly detection. However, deep learning models require large amounts of training data, but existing data sets are quite small. The proposed infrastructure will enable the collection of large data sets from wearable and fixed IoT sensors and offer the opportunity deep learning for role-based behavioral

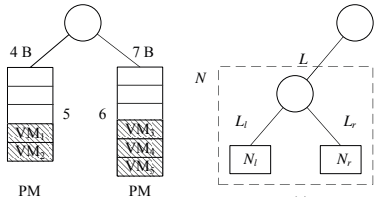

Fig. 2: (a) virtual node and (b) aggregation tree

threat detection in IoT networks.

IV. Performance Research Challenge: IoT Data Center Task Allocation

Often, IoT devices are supported by cloud-based systems, with much of the vast amounts of streaming IoT data, and even computation, offloaded to the cloud. Task-resource allocation has been an important component in cloud-based data center networks (DCNs), and will be increasingly important as more IoT systems are deployed. An open problem in task-resource allocation is the ability to provision the maximum admissible load (MAL) of VMs in physical machines (PMs). The limitation of static load distribution is that it assigns tasks to nodes in a once-and-for-all manner, and thus, requires *a priori* knowledge of program behavior. To avoid load redistribution during a run time where the load grows, we introduce *maximum elasticity scheduling*, which has the maximum growth potential subject to the node and link capacities.

We model the network as a tree G in a typical DCN. Each leaf node is a physical machine (PM) and each internal node is a switch. A load at a leaf node is called a computation load, and it determines the communication load (bandwidth). We use the *hose model* [11] for communication where each node has an aggregated performance guarantee to the set of all other nodes. Figure 2 shows a two-level, three-node binary tree where each PM (leaf node) is represented as a slotted rectangle (e.g., VM slots or computation loads) and each internal node (switch) is represented as a circle. Numbers associated with nodes and links are the available VM slots and the communication bandwidth, respectively. We assume that each VM has B Gbps total communication with other VMs. Using the hose model, the communication load of the left link and the right link is the same: the lesser of the two leafs assigned VMs is multiplied by B. This is analogous to the maximum flow of a cut in a tree. We study the following two provisioning problems: (1) Given a graph G with available node and link capacities, what is the MAL of G under the hose model? (2) Given a load that is admissible, what is the optimal schedule so that the uniform growth rate at all leaf nodes is maximized under the capacity constraint?

The optimal schedule of the second problem is called the schedule with the maximum elasticity. The MAL in Figure 2a is 10, with 4 VMs (loads) assigned to the left leaf node and 6 assigned to the right leaf node. Both the left link and the right leaf node reach maximum capacities. Suppose that we now have a load of 5 to be assigned, which is below the MAL. The schedule with maximum elasticity assigns 2 loads to the left leaf node and 3 loads to the right.

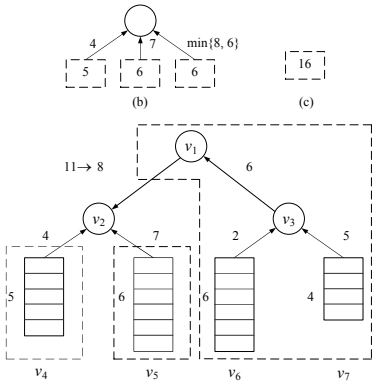

Fig. 3: An optimal solution with 3-level abstraction

Our preliminary work [12] starts with an iterative calculation process (called simple solution) for MAL when the root is given. The directed binary tree that is used to represent the orientation is called an aggregation tree. The simple solution iteratively abstracts the given tree in a bottom-up manner to the root of the given binary tree. As shown in Figure 2, the basic unit of the abstraction is a two-level, three-node branch that becomes one virtual node at the higher level. In this abstraction, one internal node and two virtual nodes serve as the child nodes of the internal node. At the bottom level of the tree, a virtual node is a leaf node. At all other levels, a virtual node is abstracted from the branch rooted at the same node. Suppose N_i (L_i) and N_r (L_r) have available node space (link bandwidth) for the left and right virtual nodes, respectively, $N = \min\{N_i, L_i\} + \min\{N_r, L_r\}$. The minimization operation ensures that the value of each branch satisfies both node space and link bandwidth requirements. This abstraction process continues level-by-level until the tree is reduced to a single virtual node. The available capacity of this virtual node is the MAL. Once the MAL is determined, we can iteratively determine the schedule that achieves the maximum elasticity. The process is top-down as the load is partitioned based on the proportion of left and right branch loads (i.e., $\min\{N, L\}$).

To obtain the optimal solution, we apply the simple solution to different orientations (i.e., different roots in addition to the given tree root used in the simple solution) of the aggregation tree and select the best one (i.e., the orientation with the maximum MAL). Fig. 3 shows an example of an orientation that generates maximum MAL at node v_2 and three levels of abstraction. A virtual node with v_3, v_6, and v_7 has a load of $\min\{6, 2\} + \min\{4, 5\} = 6$. The MAL is 16, which is $\min\{5, 4\} + \min\{6, 7\} + \min\{6, 6\}$ as shown in Fig. 3b. If a load of 8 is given, top-down load distribution can be applied to obtain the maximum elasticity. For example, the load assignment at the level-2 abstraction (Fig. 3b) is calculated based on the proportions of three branches: $\min\{5, 4\}$, $\min\{6, 7\}$, and $\min\{6, 6\}$. That is, the optimal load assignments are 2, 3, and 3 to the right, middle, and right branches, respectively. The final load assignment is (v_4, v_5, v_6, v_7) = $(2, 3, 1, 2)$, which maximizes the uniform growth rate. [12] shows that the

optimal solution uses 2 log n+1 steps (n is the number of leaf nodes). The computation and communication complexities are both linear to n. Some future challenges of elastic scheduling include the following:

Extending the elastic model to general trees and other topologies. The optimal solution can be easily extended to any k-nary tree structure. The key difference is that each internal node needs to keep track of the virtual load value of each branch. Another challenge is the extension to other structures The SDN controller can program a particular topology selecting appropriate ports of each switch and server. One interesting subarea to explore elasticity is multiple paths routing like Multiprotocol Label Switching (MPLS) [13].

Developing other communication load models. To generalize the model, we set $L = f(N)$, where f is a constant multiplier, say c. To avoid remapping f, we can scale-down the available link bandwidth by a factor of c; therefore, the same optimal solution can be applied. Our approach cannot be directly applied when the mapping function is nonlinear because the total communication load generated depends on the way the computation load is partitioned. For multiple requests, we can examine the approach where one Virtual Private Network (VPN) [14] is used for each request.

Exploring special configurations for efficiency gain. When we consider elasticity, the bottleneck must be either in the link or the node in terms of growth capability. We can consider special situations under which the simple solution is optimal. A given tree infrastructure is a computational-bottleneck if for any two-level, three-node subtree, $N_l = \min\{N_l, L_l\}$ and $N_r = \min\{N_r, L_r\}$. The intuition behind the computational bottleneck structure is that elasticity bottlenecks appear at the leaf nodes. For example, for any two-level, three-node subtree in a fat-tree topology, where $L > L_l + L_r$. This fat-tree structure is frequently used in DCNs because upper links usually carry more traffic, so a higher bandwidth must be used. Given a binary tree that has a computational-bottleneck or is a fat-tree, the simple solution is optimal.

V. Performance Research Challenge: Reliable, High Performance IoT Data Storage

With the explosively growing online digital data, particularly via a large number of various IoT devices, how to store the data efficiently and reliably are two major concerns to researchers. To address these concerns, deduplication techniques that identify and eliminate duplicate data chunks have been proposed [15], [16], [17]. However, deduplication fundamentally changes the reliability of stored objects. As a result, the data deduplication may not provide sufficient level of fault tolerance to the system, in which case, erasure coding [18], [19], [20] can be considered. Erasure coding encodes original data blocks into an expanded set of encoded blocks, such that once there is certain number of original data blocks fail, we can always reconstruct the failed data block with the encoded blocks [21], [22]. However, simply employing erasure coding into the deduplication system may impact the encoding performance and system reliability.

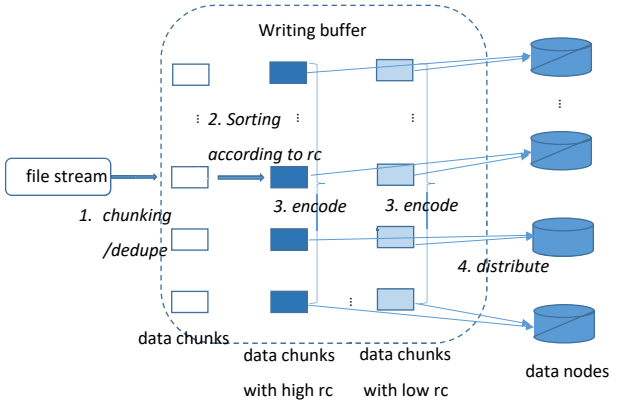

Fig. 4: Data flow of reference counter dedup.

Several works have been proposed recently which integrate erasure coding into deduplication to ensure the data reliability and availability [23], [24], [25]. However, little work has been done to explore erasure coded data deduplication systems for IoT applications. To address this problem, we envision a reference-counter-aware data deduplication scheme [26] to provide robust storage systems for IoT applications.

Exploring a reference-counter aware data deduplication. The traditional Round-Robin placement of the chunks in deduplication system causes high extra encoding overhead. By randomly placing and encoding the chunks, the reliability of the system is not ensured. Further research is needed to improve reliability. One approach forward is to encode data chunks according to their reference-counters. Figure 4 shows the data flow of such a scheme. First, files go through the chunking and dedupe process, which are the same with the traditional deduplication system. Second, the unique chunks will then go to the writing buer and get sorted by their reference-counters, which will determine how they get erasure coded. Chunks with high/low reference-counter will be encoded together respectively. The third step erasure-codes chunks into stripes. In the fourth step, after encoding, chunks will go into the container buer and be flushed into the storage nodes when the container is full. By doing this we can exploit the trade-off between storage eciency and reliability among dierent erasure codes after they are employed in the deduplication system.

Developing a risk-aware failure identification scheme to expedite failure recovery. In a traditional failure identification scheme, all data chunks share the same identification time threshold. However, failure identification schemes [27] can be developed which are failure-aware. In such schemes, chunk failures in data stripes experiencing different numbers of failed chunks can be identified using different time thresholds. For those chunks in a high risk stripe (a stripe with many failed chunks), a shorter identification time is adopted, thus improving the overall data reliability and availability. For those chunks in a low risk stripe (one with only a few failed chunks), a longer identification time is adopted, thus reducing

the repair network traffic. Therefore, the reliability, availability, and serviceability of systems can be improved simultaneously.

VI. Conclusions

In this vision paper, we identified several challenges in systems research on security and performance in the IoT. Exploring these research challenges requires a realistic testbed for implementation, deployment, and evaluation. Such a testbed should includes a set of IoT devices that sample a wide range of environmental data, facilities for annotating and accessing IoT data with ground truth labels, and the ability to experiment with IoT vulnerabilities and the configuration of security protocols. Furthermore, the testbed should allow for investigating the reliability, availability, and serviceability of cloud-based infrastructure for underlying task allocation and cloud storage of IoT data and services at the large scales that we expect in envisioned deployments.

Beyond the challenges identified here, a testbed that provides such services is essential for high-fidelity evaluations of IoT, edge, and data center research projects, providing researchers with the ability to support their experiments with realistic network and data conditions that are not supported by simulation software. Making such a testbed publicly accessible is essential to broadly support systems research that focuses on IoT security and performance research. With recent funding from the National Science Foundation, we plan to develop such a testbed with a web-based interface that allows researchers to reserve resources of the infrastructure to execute their own experiments. Evaluation metrics supported within the proposed infrastructure include those related to IoT device and system security, load distribution on virtual and physical machines, power consumption, communication overhead, computation overhead, and latency. In addition, researchers will be able to request access to testbed data (may need to be anonymized) network traffic, IoT sensor data, and mobility traces collected within the infrastructure. As such, this project not only advances the research infrastructure available at Temple University (TU), it increases the capacity for experimental evaluation for algorithms and protocols designed for IoT device and system, edge, data center and storage system.

Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. CNS-1828363.

References

[1] D. Minoli, K. Sohraby, and B. Occhiogrosso, "Iot considerations, requirements, and architectures for smart buildingsenergy optimization and next-generation building management systems," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 269–283, 2017.

[2] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," in *3rd International Conference on Electronic Design*, 2016, pp. 321–326.

[3] Z. Zhang, M. Trinkle, L. Qian, and H. Li, "Quickest detection of gps spoofing attack," in *Military Communications Conference,*, 2012.

[4] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele *et al.*, "," in *2nd IEEE World Forum on Internet of Things*, 2015, pp. 459–464.

[5] A. Vakali, L. Anthopoulos, and S. Krco, "Smart cities data streams integration: experimenting with internet of things and social data flows," in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, 2014, p. 60.

[6] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks, Elsevier*, vol. 5, no. 1, pp. 24–34, Jan. 2007.

[7] M. T. Arafin, D. Anand, and G. Qu, "A low-cost gps spoofing detector design for internet of things (iot) applications," in *Proceedings of the on Great Lakes Symposium on VLSI 2017*, 2017, pp. 161–166.

[8] K. Jansen, N. O. Tippenhauer, and C. Pöpper, "Multi-receiver gps spoofing detection: error models and realization," in *Proceedings of the 32nd Conference on Computer Security Applications*, 2016, pp. 237–250.

[9] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[10] W. W. Cohen, "Fast effective rule induction," in *Machine Learning*, 1995, pp. 115–123.

[11] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive, "A flexible model for resource management in virtual private networks," in *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, 1999.

[12] J. Wu, S. Lu, and H. Zheng, "On maximum elastic scheduling in virtual private networks with the hose model," in *IEEE ICC*, 2018.

[13] B. S. Davie and Y. Rekhter, *MPLS: technology and applications*. Morgan Kaufmann Publishers, 2000.

[14] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener, "Algorithms for provisioning virtual private networks in the hose model," *IEEE/ACM transactions on networking*, vol. 10, no. 4, pp. 565–578, 2002.

[15] C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk, W. Kilian, P. Strzelczak, J. Szczepkowski, C. Ungureanu, and M. Welnicki, "Hydrastor: A scalable secondary storage." in *FAST*, vol. 9, 2009, pp. 197–210.

[16] M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezis, and P. Camble, "Sparse indexing: Large scale, inline deduplication using sampling and locality." in *FAST*, vol. 9, 2009, pp. 111–123.

[17] B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system." in *Fast*, vol. 8, 2008, pp. 1–14.

[18] X. Li, M. Lillibridge, and M. Uysal, "Reliability analysis of deduplicated and erasure-coded storage," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 3, pp. 4–9, 2011.

[19] C. Wu, X. He, G. Wu, S. Wan, X. Liu, Q. Cao, and C. Xie, "Hdp code: A horizontal-diagonal parity code to optimize i/o load balancing in raid-6," in *IEEE/IFIP 41st International Conference on Dependable Systems & Networks*, 2011, pp. 209–220.

[20] S. Wan, Q. Cao, C. Xie, B. Eckart, and X. He, "Code-m: A non-mds erasure code scheme to support fast recovery from up to two-disk failures in storage systems," in *2010 IEEE/IFIP International Conference on Dependable Systems&Networks (DSN)*. IEEE, 2010, pp. 51–60.

[21] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," in *International Workshop on Peer-to-Peer Systems*, 2002, pp. 328–337.

[22] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "Xoring elephants: Novel erasure codes for big data," in *Proceedings of the VLDB Endowment*, vol. 6, no. 5, 2013, pp. 325–336.

[23] M. Xu, Y. Zhu, P. P. Lee, and Y. Xu, "Even data placement for load balance in reliable distributed deduplication storage systems," in *IEEE 23rd International Symposium on Quality of Service*, 2015, pp. 349–358.

[24] C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, "R-admad: High reliability provision for large-scale de-duplication archival storage systems," in *Proceedings of the 23rd International Conference on Supercomputing*, 2009, pp. 370–379.

[25] M. Xiao, M. A. Hassan, W. Xiao, Q. Wei, and S. Chen, "Codeplugin: Plugging deduplication into erasure coding for cloud storage." in *HotCloud*, 2015.

[26] T. Liu, S. A. X. He, and C. Wu, "Reference-counter aware deduplication in erasure-coded distributed storage system," in *Proceedings of the IEEE Int'l Conf. on Networking, storage, and Architecture*, 2018.

[27] J. Fang, S. Wang, and X. He, "Rafi: Risk-aware failure identification to improve the ras in erasure-coded data centers," in *Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC)*, 2018.