Resilient IoT using Edge Computing

Hokeun Kim, UC Berkeley
Schahram Dustdar, TU Wien
Edward A. Lee, UC Berkeley

Abstract—Denial-of-service (DoS) attacks on the safety-critical Internet of Things (IoT) can lead to life-threatening consequences. The risk of distributed denial-of-service (DDoS) attacks has been increasing as more Things are connected to the Internet. However, current IoT solutions based on cloud computing cannot properly mitigate DoS threats because remote cloud servers lack means to detect and react to the attacks on local IoT systems. In this paper, we compare cloud computing and edge computing in terms of local context awareness and resilience for the IoT. We propose context-awareness levels to address availability threats and illustrate how context-aware edge computing enhances IoT's resilience against DoS attacks through our edge computing-based security solution for the IoT.

Availability can be safety-critical for many IoT systems. In October and November of 2016, a distributed denial-of-service (DDoS) attack on the building control systems shut down the heating systems of buildings in Finland for more than a week [1]. Winter is harsh in Finland, thus, failures in heating systems raised safety concerns, not just inconvenience. Meanwhile, the risk of DDoS attacks has almost doubled in 2017 compared to 2016 due to the increasing number of IoT devices, according to a recent report by Corero Networks [2]. An illustrative example of DDoS launched by the IoT was the Mirai botnet [3] which compromised hundreds of thousands of IoT devices and attacked Dyn DNS (Domain Name System) servers, disrupting Internet connections to major websites.

Breaching availability of computer systems is the main goal of denial-of-service (DoS) attacks. DDoS attacks fall into one category of DoS attacks, which send excessive network traffic to the target servers to exhaust their computational and communicational resources. For the IoT, including cyber-physical systems (CPS) interacting with humans and the physical world, availability attacks such as DDoS attacks can pose safety threats. Therefore, it is crucial to build IoT systems that are resilient against availability threats.

However, many current IoT solutions are built around cloud computing, which makes the system's availability dependent on remote cloud servers. Google's OnHub incident [4] demonstrated the risk of depending on remote servers for the IoT. On February 23, 2017, Google's smart routers, called OnHub, suddenly stopped working and the IoT devices connected to the routers also became unavailable only because Google's remote authentication servers failed.

Recently, devices at the network edge are becoming smarter and more capable. These devices are called *edge computers*. Edge computing [5], which uses edge computers to offer computational and communicational resources for Things, brings opportunities to build resilient IoT systems.

In this paper, we propose an authentication and authorization infrastructure for the IoT based on context-aware edge computing, along with context awareness levels of edge computers for building resilient IoT systems.

1 Perspectives of the IoT

Different perspectives of the IoT can lead to different IoT system designs. This section compares two different ways of viewing the IoT to highlight the importance of considering an underlying network architecture in designing resilient IoT systems.

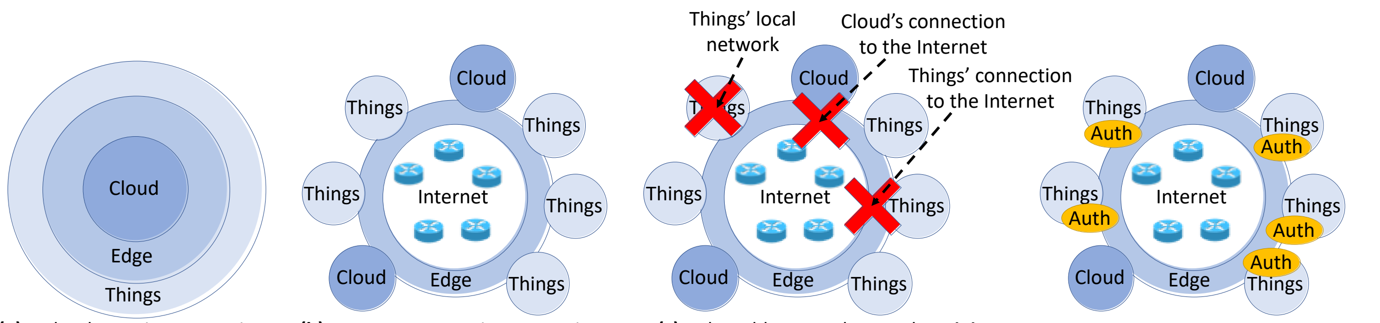
1.1 Cloud-centric perspective

A *cloud-centric perspective* of the IoT is a conceptual view which considers the cloud as a central platform for the IoT and edge computers as the edge of the cloud as shown in Fig. 1 (a). This perspective is widely adopted, for example, in *fog computing* [6], where the cloud comprises core services and the edge serves as local proxies for the cloud, mainly for offloading part of cloud's workload. From this perspective, edge computers play supportive roles for IoT services and applications. Cloud computing-based IoT solutions [7] use cloud servers for various purposes including massive computation, data storage, communication between IoT systems, and security. However, the cloud-centric perspective misses important facts in the real network architecture of the IoT:

- In the network architecture, the cloud is also located at the network edge, not surrounded by the edge.
- Computers at the edge do not always have to depend on the cloud; they can operate autonomously and collaborate with one another directly.

1.2 Internet-centric perspective

To better discuss how to protect IoT systems from DoS threats, we propose a new perspective for the IoT called *Internet-centric perspective* shown in Fig. 1 (b). This perspective views the Internet as a center of the IoT architecture and considers the edge as gateways to the Internet, not to the cloud. Each local network can be organized around the edge computers autonomously. Thus, the local systems are not always dependent on the cloud nor the Internet. The Internet-centric perspective captures essential aspects of the IoT:



(a) A cloud-centric perspective: Edge as "edge of the cloud" **(b)** An Internet-centric perspective: Edge as "edge of the Internet" **(c)** Vulnerable spots that can be exploited to cause DoS **(d)** SST architecture: Locally centralized, globally distributed Auth (authentication / authorization)

Fig. 1. From perspectives of the IoT to the SST (Secure Swarm Toolkit) architecture

- Things in the IoT belong to partitioned subsystems or local networks rather than belonging to a big centralized system directly.
- The cloud is also connected to the Internet via the edge of the network.
- Remote IoT systems can be connected directly via the Internet and their communication does not have to go through the cloud.
- The edge not only can connect Things to the Internet, but also can disconnect the traffic from outside to protect Things. For this, the local IoT system must be able to operate autonomously, although the system's performance might be affected once it loses the cloud's support.

2 Toward Resilient IoT Systems

With the Internet-centric perspective discussed above, how can we build resilient IoT systems? We can start with the tactics that the attackers would likely take to cause DoS.

Of course, the attackers can target the cloud to exhaust the resources of cloud servers. However, cloud servers are inside data centers and they are well protected against availability attacks from the outside, using many layers of defense including firewalls. It will be very challenging for attackers to take down a single data center, even with a lot of resources and efforts. Moreover, many commercial cloud services consist of globally distributed data centers, which makes it even more difficult to take down all data centers.

Alternatively, the attackers can try other approaches. Fig. 1 (c) shows weak points that attackers can exploit to cause DoS in IoT systems without directly attacking the cloud. As long as attackers can disrupt the IoT systems, the attackers succeed. The attackers can hamper the connection between the cloud and Things, for example, by making DNS services unavailable, as the Mirai botnet did. They can also attack the local network to disrupt the IoT services directly. Therefore, it is not enough to just protect the cloud servers. The individual local IoT networks also need to be protected.

From the discussion, we note a couple of fundamental requirements for resilient IoT systems.

First, we must be prepared for when the cloud is not available. The IoT systems should be able to provide at least vital services, for example, the heating systems in cold regions, even when the cloud is not available. In this sense, we can use edge computers as local controllers for Things as a backup for the cloud. In general, edge computers have more resources than Things and can be local central points.

Second, a local IoT system itself should be equipped with defense mechanisms against availability threats. The defense mechanisms include detecting and mitigating the impact of attacks, and reacting to failures to recover the system's availability. Edge computers can play key roles to implement such defense mechanisms. For example, since an edge computer sits between Things and the Internet, it can detect an incoming DDoS attack and protect the local network by blocking the external traffic. If the edge computers are aware the local system's characteristics and how the system should behave, for example, the expected volume of data traffic, desirable temperature ranges, they can detect the anomaly and recover the normal state. The edge computers can also use local resources and security measures to recover availability.

3 Resilient Security Solution for the IoT

We present our open-source edge computing-based IoT security solution that is resilient to availability threats. Called the *Secure Swarm Toolkit (SST)* [8], it is freely available on https://github.com/iotauth. SST provides authentication and authorization services for the IoT. *Authentication* is a process of identifying devices or users and *authorization* is a process of controlling access to important resources such as control of CPS. These two processes are critical for security, safety, and availability, as shown in Google's OnHub incident where authentication problems of Google servers led to the entire system's failure.

3.1 Resilient edge computing-based architecture

SST's architecture is shown in Fig. 1 (d). SST has a *locally centralized, globally distributed* architecture [9], which has many potential advantages in building resilient IoT systems.

In SST, Things are authenticated and authorized by an edge-hosted *locally centralized* entity, called *Auth* [10]. By running security functions on the edge, the IoT systems can continue authentication and authorization processes even when cloud servers are unavailable. Moreover, Auths monitor the entire access activity between Things, which allows Auths to detect an anomaly in the system. Auths can also protect the local IoT networks from external attackers, using defense mechanisms including firewalls and physically disconnecting the DDoS traffic toward local systems. The locally centralized architecture enables Auths to react to compromised Things in a timely way.

The *globally distributed* architecture of SST not only makes the IoT systems scalable [8] but also enhances the resiliency of the IoT [11]. Auths hosted on edge computers will be more geographically distributed than cloud servers, making it even more challenging for attackers to take down the IoT system by attacking edge computers. The cloud servers in data centers may become unreachable by disrupting DNS services or the Internet connection to data centers. However, this type of attack will be less effective for SST because many of Auths will be reachable through local networks even when the Internet connection is unstable.

4 Context Awareness of Cloud and Edge

In computing, context has various meanings. Here, we consider *context* as information about the environments in which the IoT systems operate, including underlying platforms, available devices, network topology, location and time. *Context awareness* refers to the capability of computers in the IoT to sense and react to what is happening in their operating environments. Context awareness is crucial for the resilience of an IoT system because it enables the computers in the system to mitigate the impact of an attack and recover from a failure. Context awareness has been related to security of the IoT. Examples include using context awareness for trust initialization [12] and trust management [13]. In the IoT, the cloud and the edge will have different types of context awareness.

4.1 Global context in the cloud

The cloud will have a better holistic view of global context compared to the edge. In the smart city example shown in Fig. 2, we assume the subsystems are connected to the cloud which receives real-time data from the subsystems. Cloud servers will have a better understanding of what is happening in the city and which subsystem has problems, for example, whether the power failure in manufacturing systems is due to a failure in power plants. The cloud will also be able to control the subsystems and react to possible incidents using *global context awareness*. When the air quality measured by an environmental monitoring system is not healthy, the cloud can order smart buildings to close windows and activate air purifiers. For the traffic infrastructure, the cloud can monitor traffic situation and control traffic signals to ease a traffic jam.

4.2 Local context in the edge

The edge will have the better awareness of *local context*. Edge computers will have access to the raw communication packets and data within local IoT systems. For example, edge computers for the traffic infrastructure will be aware of the real-time video data at crossroads, and edge computers for environmental monitoring can analyze raw sensor data. Also, edge computers managed by the local administrators can access data not available to external systems for privacy reasons, such as the on-body monitoring data of medical centers or surveillance camera data of smart homes. The edge can view incoming and outgoing data from the local system, thus, it will be able to detect DDoS attacks toward the IoT system, which can be challenging for the cloud. Edge computers will be better aware of the network topology and locally available resources that can be used to mitigate threats and recover availability. Thanks to the proximity to local systems, edge computers will be able to detect availability threats and take better actions in a more timely fashion than remote cloud servers.

The cloud and the edge have different types of context awareness, and they are complementary to each other. The global context awareness of the cloud fosters collaboration between heterogeneous subsystems while the local context awareness of the edge enables subsystem-specific analysis and close interaction with Things. We will focus on the local context awareness of the edge with regard to building robust IoT systems.

5 Context Awareness and Resilience

Local context awareness is especially important for resilience. We propose five awareness levels for the edge computing-based IoT, *event*, *situation*, *adaptability*, *goal*, and *future* awareness, as shown in TABLE 1.

Event awareness is the simplest capability of sensing and monitoring environments. An event-aware system can react to sensed events according to predefined rules. Examples of defense mechanisms and tools used by event-aware systems include firewalls and rule engines. Data filtering and dissemination are provided by this level as infrastructural services to higher awareness levels.

Situation awareness is a more advanced capability for understanding the implication of a series of events and reacting to the situation based on understanding. For example, network intrusion detection systems (NIDS) not only monitor the network traffic but also detect network intrusion by analyzing characteristics or anomalies of the data traffic. Situation-aware systems often use statistical tools to detect anomalies.

An *adaptability-aware* system can change and modify itself if necessary when the system detects threats or failures. Adaptability awareness includes knowledge and control of the available resources and how the resources can be used to recover and maintain availability even under failures. Many reconfigurable systems will have this level of awareness, including software-defined networking (SDN). This level of awareness makes IoT systems more resilient even when some of the important components, for example the edge computers, become unavailable.

The *goal awareness level* introduces goals expressing the overall objectives and purposes of a self-adaptive system. When there are multiple goals, an IoT system must be able to resolve conflicting goals within resource constraints. Therefore, a goal-aware system can take priorities and trade-offs into account when adapting to new situations. Such systems include mixed-criticality systems which contain tasks with different criticality levels on a single platform.

Future awareness is an ultimate form of awareness that enables self-sustainable IoT systems. A future-aware system is capable of predicting longer-term effects of short-term

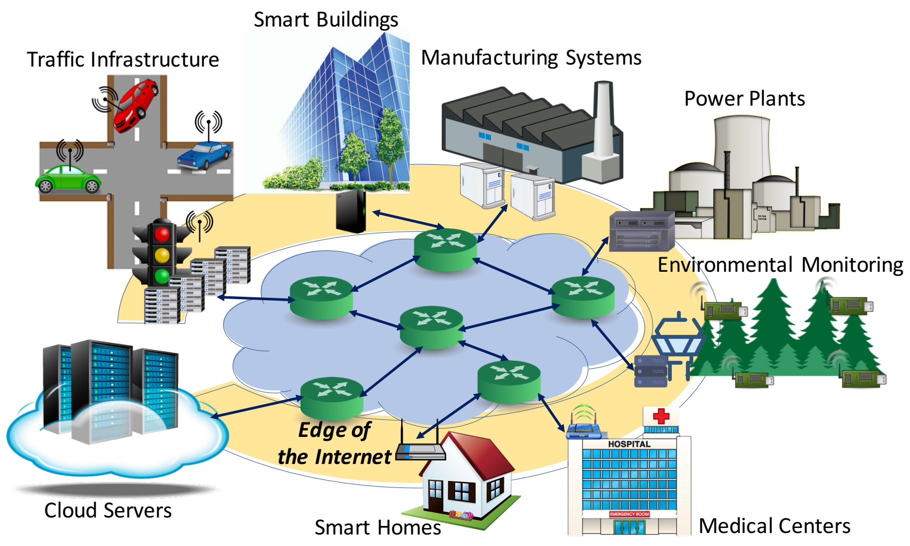


Fig. 2. Smart city with the edge of the Internet

adaptation actions and considering future resource provisions and constraints when utilizing short-term resources. For example, edge computers in a future-aware system should be able to anticipate wear-out and replacement cycles of Things such as battery-powered sensor nodes and take proper actions to maintain long-term availability.

In summary, the awareness levels help us understand what the IoT system should know to support a certain level of resilience. Event awareness is the most basic and fundamental level of awareness that needs to be part of higher levels of awareness. To reach the future awareness level, an IoT system will require all lower levels of awareness.

6 Context-Aware Edge Computing for Resilient IoT

Even with SST's architectural advantages, there still exist availability threats to edge computing-based IoT systems. To cause DoS in such systems, attackers will probably target edge computers rather than individual Things, to maximize the impact of an attack.

In distributed systems, it is common to replicate resources across distributed computers to increase availability. Such systems include content delivery networks (CDN), for example Akamai and Limelight Networks. However, distributing the authentication and authorization-related information is trickier than sharing content resources such as web pages, images, or videos because we need to consider trust among Things and edge computers.

6.1 Secure Migration and Awareness levels

Auths in SST maintain distributed trust relationships with one another. This allows other trusted Auths to take over authentication and authorization services for Things in case of their Auth's failure. We call this technique *secure migration*, in which the Things migrate from unavailable Auths to other *trusted* Auths to continue security services while keeping trust relationships intact.

Fig. 3 demonstrates how SST's secure migration technique implements context awareness levels to mitigate availability threats. SST's implementation provides practical insights for considerations in building a resilient IoT using edge computing at all five levels of awareness.

6.1.1 Event awareness

In SST, edge computers (Auths) and Things use simple mechanisms to sense network conditions and health status of the system. Auths check each other periodically using a heartbeat protocol. Things send authentication and authorization requests to Auths. They will notice communication failures with their Auths as events. Event awareness provides the base for higher-level awareness for taking further security measures.

6.1.2 Situation awareness

Situation awareness is used to trigger secure migration. With more resources and better local context awareness than Things, Auths can use resource-demanding but more accurate methods. Examples include sharing heartbeat response information for a possibly failed Auth and actively monitoring communication channels to make sure it is not because of issues in the communication media. Things keep track of failures in Auth's responses using simple counters and check whether the counter value exceeds a certain threshold. By cross-checking the information gathered by Auths and Things, the SST infrastructure can determine whether the events indicate a false alarm or an actual failure. When an actual failure is detected, the secure migration process begins.

6.1.3 Adaptability awareness

Auths are aware of the IoT systems' adaptability before failures. Adaptability awareness is critical for recovering availability in case of DoS attacks. In SST, Auths construct *migration policies* describing which Things should migrate to

TABLE 1
Awareness levels in IoT systems

Awareness Levels	Characteristics	Capabilities	Examples
Event-aware	A system collects simple events that trigger basic Event-Condition-Action rules. The system has no explicit knowledge of the resources needed nor whether the adaptation has a long-lasting (positive) effect.	To react to events based on predefined rules, regardless of any other factors in the situation.	Firewalls, Rule engines (e.g., Drools), IFTTT.com (If This Then That)
Situation-aware	Ability to perceive the status of a system by aggregating relevant events. The system understands the implication of individual events in a greater context.	To react to events properly in context, with the capability to collect and understand local contextual information.	Network Intrusion Detection Systems (NIDS), Anomaly detection systems
Adaptability-aware	Awareness of the possible adaptation capability of a system in its environment. At this level, cooperative adaptation can be conducted spontaneously based on the knowledge of adaptability.	To initiate spontaneous collaboration with other edge computers and controllable environmental conditions.	Reconfigurable Software-Defined Networking (SDN), Reconfigurable CPS
Goal-aware	Awareness of the goals of a system as a whole. In IoT systems, a goal not only includes the desired functionality of a service, but also non-functional properties and resource constraints imposed by the environment. In the presence of conflicting goals, this level of awareness also considers the potential trade-offs and priorities (criticalities) between goals.	To negotiate with other edge computers regarding resource allocation. To understand the significance of a potential failure and attempt to avoid it accordingly.	Mixed-criticality systems, (e.g., Avionic systems, Autonomous vehicles)
Future-aware	Awareness of a system's lifecycle describing long-term utilization and resource provisioning by the environment. This requires information on the probable future system state based on scheduled or expected future events. Ultimately, this level describes systems that can select appropriate short-lived adaptation actions that respect long-term resource constraints and goals.	To predict resource consumption, user behavior, and future resource requirements. To act according to predictions.	Self-sustainable smart city

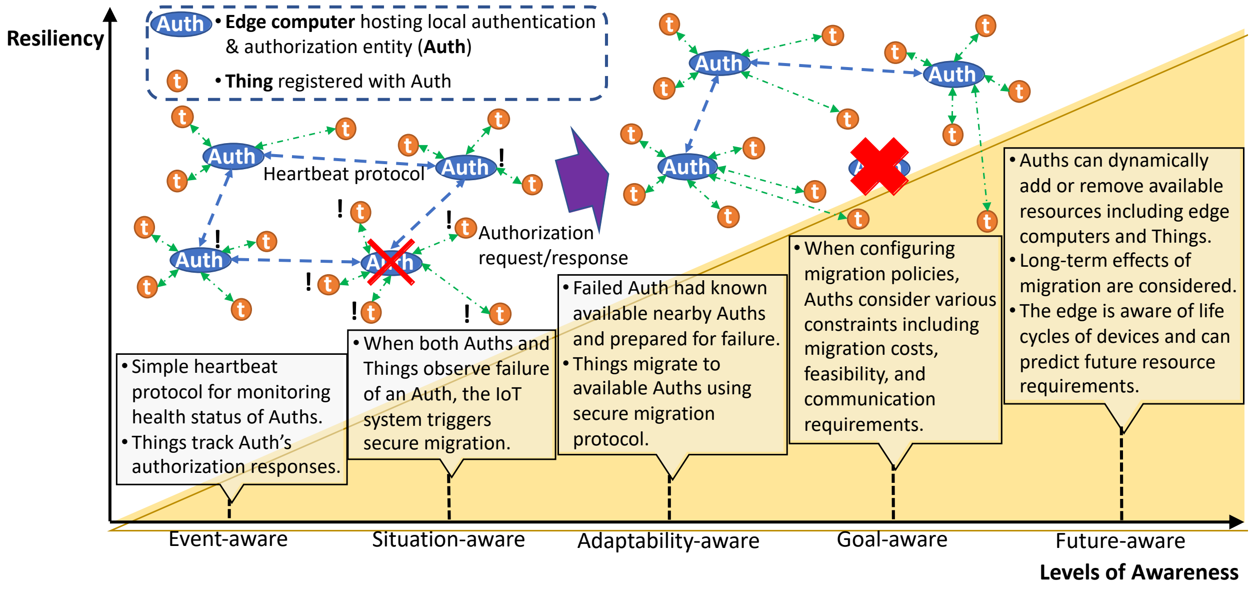

Resiliency
Auth • Edge computer hosting local authentication & authorization entity (Auth)
t • Thing registered with Auth
Auth
Heartbeat protocol
Authorization request/response
• Simple heartbeat protocol for monitoring health status of Auths.
• Things track Auth's authorization responses.
• When both Auths and Things observe failure of an Auth, the IoT system triggers secure migration.
• Failed Auth had known available nearby Auths and prepared for failure.
• Things migrate to available Auths using secure migration protocol.
• When configuring migration policies, Auths consider various constraints including migration costs, feasibility, and communication requirements.
• Auths can dynamically add or remove available resources including edge computers and Things.
• Long-term effects of migration are considered.
• The edge is aware of life cycles of devices and can predict future resource requirements.
Event-aware
Situation-aware
Adaptability-aware
Goal-aware
Future-aware
Levels of Awareness

Fig. 3. Levels of awareness and resilience in Secure Swarm Toolkit (SST)

which Auths when there is an Auth failure. An adaptability-aware migration policy considers factors that affect availability after migration, for example, access requirements between Things and trust between Auths.

During normal operations, an Auth sets up *migration credentials*, cryptographic tokens used to establish new trust relationships, for its Things and sends them out to other trusted Auths. The Auth also sends a list of trusted Auths and their network addresses to its Things. When an Auth failure occurs, the Things try sending migration requests to other available trusted Auths. Trusted Auths will accept the migration request when a Thing requests to the designated Auth, or will reject the request otherwise. This scheme allows dynamic changes in migration policies.

6.1.4 Goal awareness

For a given IoT system based on SST, there can be multiple possible migration policies due to a multitude of combinations of Auths and Things. Goal awareness is used to decide which migration policies lead to better availability by considering various constraints including communication costs, the capacity of Auths, load balancing, and signal reachability between Things and Auths. Specifically, SST uses Integer Linear Programming (ILP) to find the best migration policies under given constraints, including the computing power of the edge computer solving the ILP problem. SST currently supports up to this level of awareness.

6.1.5 Future awareness

Self-sustainable and future-aware IoT systems should be able to replace and renew worn-out resources. SST's secure migration can be easily extended to remove or add edge computers hosting Auths. To remove an old Auth, we can set a migration policy without the old Auth, turn it off, and trigger secure migration. When we have a new Auth, we may need to move Things from other Auths to the new Auth for better load balancing. For this, we first set up a migration policy that migrates Things to the new Auth, provide the Things with new Auth's network information, and enforce migration.

Thanks to SST's locally centralized and globally distributed architecture, adding and removing Things can be done completely locally without any global-level changes. A newly added Thing will be able to communicate with other Things that are authorized by the other Auth, as long as their Auths maintain trust and allow communication between those Things. Also, a removed Thing will not be able to communicate with others anymore as its access will be revoked by its Auth.

Auths can formulate and solve ILP problems considering longer-term effects of the short-term migration activities, for example, "what if an Auth to which Things have migrated fails later?". Although the current design of SST does not include awareness of future resources, we can extend ILP formulation to include the edge computers expected to be added in the future.

7 Experiments and Results

To show the resilience of the proposed approach with different context awareness levels, we carried out experiments by extending the experimental setup used in our previous work [11]. Fig. 4 illustrates the extended experimental setup, a simulated environment of a smart building with door controllers and user devices with door opening apps. This environment was modeled using floor plans of the 4th and 5th floors of Cory Hall at UC Berkeley and included 7 Auths hosted on edge computers, 35 door controllers and 45 user devices positioned as in Fig. 4 (a). In this environment, a user device must be authorized by its Auth to open a door. Each user device tried to open the nearby door every minute. Availability was measured by the portion of user devices successfully opened nearby doors. Each of Auths, door controllers, and user devices was executed on a Linux Container. The network was simulated using the ns-3 simulator (https://www.nsnam.org/) with a wired network for Auth-to-Auth communication and a wireless network for Auth-to-Thing and Thing-to-Thing communication. Each simulation was performed on Amazon AWS for 20 minutes in real time, 5 minutes before Auth failures and 15 minutes after failures.

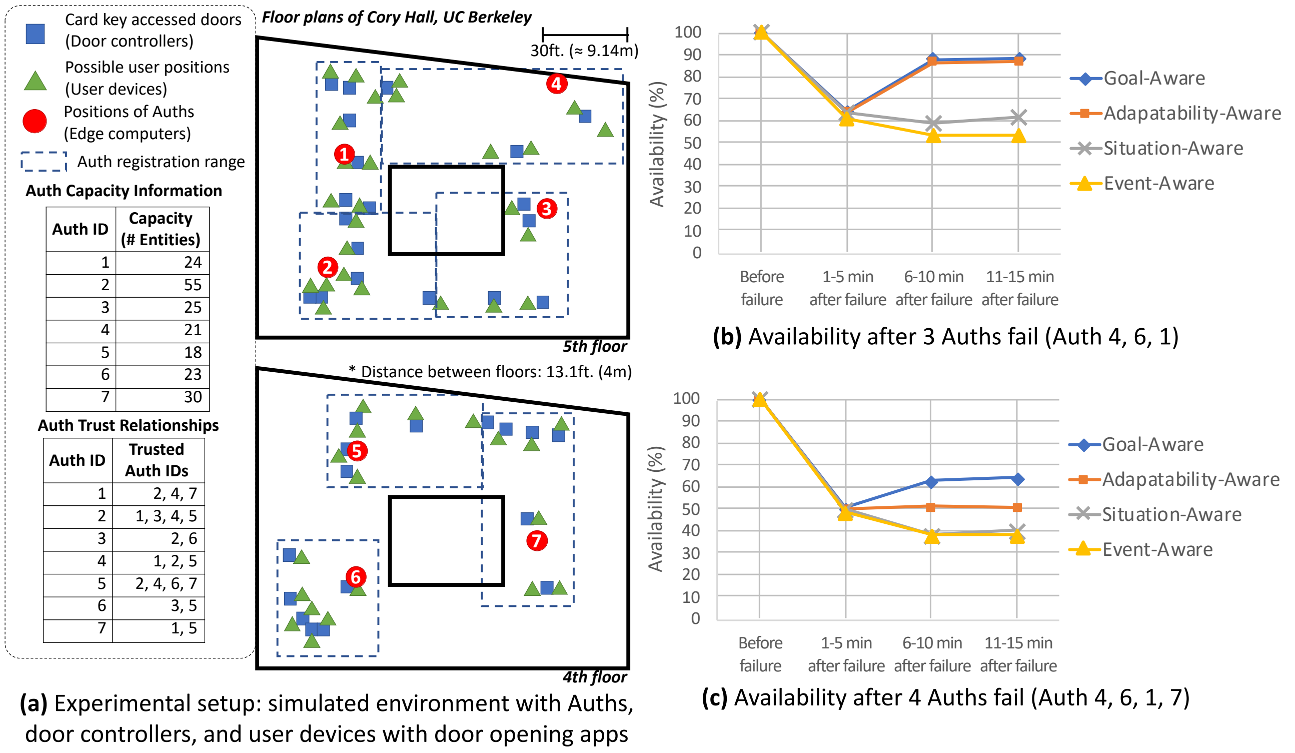
We compared four different awareness levels, event, situation, adaptability, and goal awareness levels. The event-aware SST was set to just retry and wait for the recovery of Auths. The situation-aware level was able to detect Auth failures and trigger an ad-hoc migration which migrates Things to nearest Auths first. The migration policy of the adaptability-aware SST considered trust between Auths and communication requirements between Things, in this case, which user device should be able to communicate with which door controller. The goal-aware SST also considered the overall system's goal, including Auths' capacity and load balancing.

The experimental results in Fig. 4 (b) and Fig. 4 (c) show the availability of the experimental IoT system when 3 Auths failed and when 4 Auths failed, respectively. The results show that higher awareness levels recovered higher availability, for example, the situation-aware SST detected the failures and triggered ad-hoc secure migration while the event-aware SST did not. The adaptability-aware SST could recover even higher availability by considering which Auths can be trusted by Things after migration and which Things need to communicate. The goal-aware SST performed better especially in the 4 Auths failure case because load balancing became more critical when less Auths were left after failures.

8 Conclusions and Future Work

Guaranteeing availability is critical to making the IoT secure and safe. In addition to the architectural merits of edge computing against availability threats, better context awareness leads to a more resilient design of IoT systems as shown with our authentication and authorization infrastructure for the IoT. Context awareness levels proposed in this paper can be concrete guidelines for IoT system designers. Implementing each level of awareness may not always be possible due to constraints, however, it is important to consider lower-level awareness as a foundation on which higher-level awareness is implemented.

As future work, we plan to study awareness levels for other aspects of protecting the IoT. Context awareness can be used to authenticate a user's identity, for instance, based on



Auth Capacity Information

Auth ID	Capacity (# Entities)
1	24
2	55
3	25
4	21
5	18
6	23
7	30

Auth Trust Relationships

Auth ID	Trusted Auth IDs
1	2, 4, 7
2	1, 3, 4, 5
3	2, 6
4	1, 2, 5
5	2, 4, 6, 7
6	3, 5
7	1, 5

(a) Experimental setup: simulated environment with Auths, door controllers, and user devices with door opening apps

(b) Availability after 3 Auths fail (Auth 4, 6, 1)

(c) Availability after 4 Auths fail (Auth 4, 6, 1, 7)

Fig. 4. Experimental setup and results

a user's location or temporal behavior. For event awareness, the edge computers can use sensors to detect other types of DoS attacks, including signal jamming attacks, or power-drain attacks. Auths can use situation awareness based on statistics to detect application-layer threats such as service abuse or cybercrimes. Adaptability-aware edge computers can protect the privacy of sensitive information depending on the on-going agenda in smart conference rooms. Future awareness is the most under-explored area where we will research further for an IoT system's life-cycle speculations such as demand prediction for IoT services.

References

[1] L. Mathews, "Hackers Use DDoS Attack To Cut Heat To Apartments," *Forbes*, Nov. 2016. [Online]. Available: https://www.forbes.com/sites/leemathews/2016/11/07/ddos-attack-leaves-finnish-apartments-without-heat/

[2] "Corero DDoS Trends Report | Q2Q3 2017," Corero Network Security, Tech. Rep., 2017.

[3] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[4] I. Morris, "Google's Latest Failure Shows How Immature Its Hardware Is," *Forbes*, Feb. 2017. [Online]. Available: http://www.forbes.com/sites/ianmorris/2017/02/24/googles-latest-failure-shows-how-immature-its-hardware-is/

[5] W. Shi and S. Dustdar, "The Promise of Edge Computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16.

[7] A. Botta, W. de Donato, V. Persico, and A. Pescap, "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, Mar. 2016.

[8] H. Kim, E. Kang, E. A. Lee, and D. Broman, "A Toolkit for Construction of Authorization Service Infrastructure for the Internet of Things," in *Proceedings of the 2nd ACM/IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Pittsburgh, PA, Apr. 2017, pp. 147–158.

[9] H. Kim and E. A. Lee, "Authentication and Authorization for the Internet of Things," *IT Professional*, vol. 19, no. 5, pp. 27–33, Oct. 2017.

[10] H. Kim, A. Wasicek, B. Mehne, and E. A. Lee, "A Secure Network Architecture for the Internet of Things Based on Local Authorization Entities," in *Proceedings of the 4th IEEE International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, Austria, Aug. 2016, pp. 114–122.

[11] H. Kim, E. Kang, D. Broman, and E. A. Lee, "An Architectural Mechanism for Resilient IoT Services," in *Proceedings of the 1st ACM Workshop on Internet of Safe Things (SafeThings)*, Delft, The Netherlands, Nov. 2017.

[12] M. Lohstroh, H. Kim, and E. A. Lee, "Contextual Callbacks for Resource Discovery and Trust Negotiation on the Internet of Things: Work-in-progress," in *Proceedings of the 13th ACM International Conference on Embedded Software (EMSOFT)*, Seoul, South Korea, Oct. 2017, pp. 14:1–14:2.

[13] Y. Ben Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Trust management system design for the Internet of Things: A context-aware and multi-service approach," *Computers & Security*, vol. 39, pp. 351–365, Nov. 2013.