Toward Resilient Networks with Fog Computing

Amir Modarresi* and James P.G. Sterbenz*†‡
amodarresi@ittc.ku.edu, jpgs@{ittc.ku.edu|comp.{lancs.ac.uk|polyu.edu.hk}}

*Information and Telecommunication Technology Center
Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS 66045, USA
www.ittc.ku.edu/resilinets

†School of Comp. and Comm. (SCC) and InfoLab21
Lancaster University, LA1 4WA, UK
‡Computing, The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

***Abstract*—Cloud computing is a solution to reduce the cost of IT by providing elastic access to shared resources. It also provides solutions for on-demand computing power and storage for devices at the edge networks with limited resources. However, increasing the number of connected devices caused by IoT architecture leads to higher network traffic and delay for cloud computing. The centralised architecture of cloud computing also makes the edge networks more susceptible to challenges in the core network. Fog computing is a solution to decrease the network traffic, delay, and increase network resilience. In this paper, we study how fog computing may improve network resilience. We also conduct a simulation to study the effect of fog computing on network traffic and delay. We conclude that using fog computing prepares the network for better response time in case of interactive requests and makes the edge networks more resilient to challenges in the core network.**

***Index Terms*—Network resilience, survivability, disruption tolerance; Future Internet, IoT; Cloud and fog computing, OpenFog; ns-3 network simulation**

I. Introduction

The emerging Internet of Things (IoT) [1], [2], [3], [4] has increased the growth of nodes at the edge-networks. Introducing new types of network protocols suitable for different data rates, range, and energy consumption has boosted this growth substantially. Technology advancement leading to low-price end-point devices with high processing power is another factor for this growth. Finally, having the cloud as a powerful centralised processing entity with high capacity storage in the backbone structure satisfies all essential elements to push complex applications to the edge nodes that generate significant traffic back to the cloud. Increasing dependability to the cloud as a centralised structure makes the edge nodes more vulnerable to the occurrence of any challenges in the core networks and the cloud. Furthermore, long distance from the cloud does not satisfy some application requirements such as low latency and response time. On the other hand, resource-poor devices at the edge networks require computation power to be provided by the cloud. Unexpected traffic load threatens the performability and usability of applications. Introducing fog computing [5], [6] is a potential solution to provide some answers to these problems.

Fog is a horizontal, highly virtualised layer located between the edge networks and the cloud that provides computing, storage, and networking services to edge devices [7]. Proximity to the users offers location awareness, low latency, high interaction, and low response time. High heterogeneity, large geographical distribution, and the high number of nodes are some of the characteristics that may promote resilience network if located in the right places. In this paper, we show that adding a fog layer to the architecture of the IoT increases network resilience. The rest of this paper is organized as follows: In Section II, we focus on the currently proposed models for fog computing and similar architectures as related work. We describe the fog architecture and its effects on network resilience in Section III. In Section IV, we perform a simulation to study the effect of fog computing and we compare it with cloud-only architecture. We conclude our paper in Section V.

II. Related work

IoT has confronted cloud computing with many challenges such as increasing the number of connections, the amount of data transferring to the cloud, and response time. More devices are attaching to the Internet that do not need any human interaction for data communication. Cisco predicts that 50 billion devices will be connected to the Internet by 2020 [8], which is 6.5 devices per person for the world population. Using inexpensive devices with limited processing power has made this growth practical. Some applications need low latency and response time, as well as low jitter. As an attempt to mitigate Cisco introduced *fog computing* between edge devices and the cloud to provide some more local processing and storage. In this model, all event-based and real-time queries are executed in the fog and with processed and refined data are transferred to the cloud where needed for more processing and decision-making applications.

The OpenFog Consortium [9], [10], a group of companies and universities including Intel, Cisco, ARM, Dell, and Princeton University, expands the fog definition after claiming that mandatory cloud connectivity is not adequate for the IoT. OpenFog considers fog computing as a horizontal architecture to provide a continuüm of distributed computing, storage, and network services from the cloud to the edge network. Moving computation near the edge supplies enough resources for sensors, actuators, and cyber-physical systems. The cloud

978-1-5386-0671-1/17/$31.00

and fog are mutually beneficial, in which some services work better on either or both. The application requirements and the current status of the network dictate which applications go to the cloud and which remains in the fog. Figure 1 illustrates the OpenFog architecture. Scalability, autonomy, RAS (reliability, availability, and serviceability), and hierarchy are considered as some of the primary attributes of this architecture.

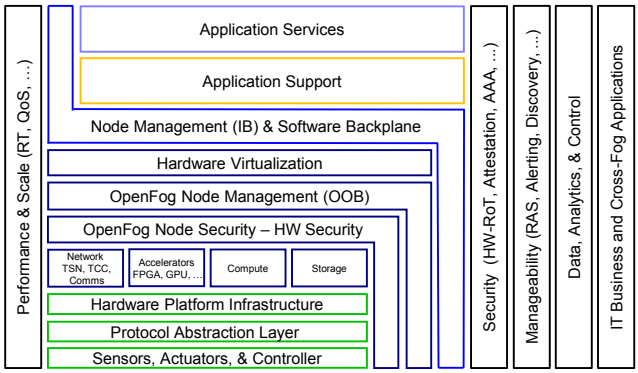

Fig. 1. OpenFog architecture [10]

Clouds at the edge is another solution by introducing private clouds and mini-clouds close to the edge [11]. It is suggested that this solution can be easily deployed in the LTE enhanced packet core (EPC). This solution is another way to confine the network traffic at the edge. It is also suggested that cloud-enabled user devices can contribute to expand the edge-cloud layer by leasing their resources.

There are other similar architectures including mobile cloud computing (MCC), cloudlet, and mobile edge computing (MEC); however, they have been utilised for other purposes, especially for the mobile environment. For instance, MCC [12], [13] has been specialised for the mobile environment by integration of cloud computing to increase performance, scalability, availability, and security for mobile devices. In this architecture, cloud resources such as computing and storage are used to support and run applications on mobile devices. In other words, mobile networks offer network access, while the cloud resources are responsible for running the mobile applications and storing user data. In another similar approach, resources from other mobile devices in the proximity are used to implement MCC [14].

Cloudlet [15] is a solution to overcome high delay and lack of resources in mobile phones by using trusted, resource-rich, well-connected computers to the Internet as a layer between the edge network and the cloud. In this solution, users run their requests on local machines installed in public areas instead of sending the requests to the cloud. A virtual machine is instantiated in the cloudlet according to the user request and destroyed when the service is completed. This solution redirects data traffic to a wireless LAN to get the benefit of higher bandwidth and overcome the delay in the mobile environment to access the cloud resources.

In traditional cellular networks, the base stations work as an access point to forward traffic to the core network without performing any processing. In order to reduce delay and traffic in the core network, MEC servers are attached to the base stations and supply computing power. If an MEC server can handle the process, the result returns to the user without entering the core network; otherwise, the request is sent to the cloud for further processing [16]. Nokia [17] has deployed MEC commercially to support smart vehicle and industrial IoT applications. Figure 2 depicts the proposed topology for MEC, in which the MEC servers are capable of processing both user and control traffic, instead of sending all data to the core network.

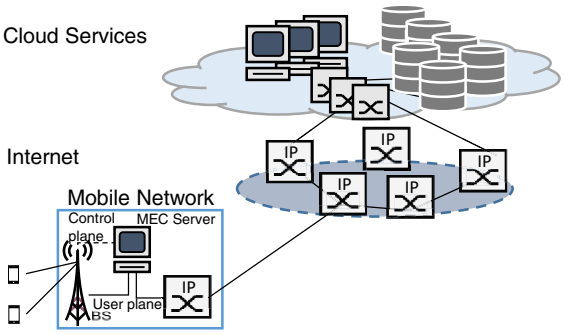

Fig. 2. Mobile edge computing topology

III. Analyzing Fog Architecture

To the best of our knowledge, there is currently no standard model to represent the fog architecture and characteristics of fog computing systematically. However, the OpenFog reference architecture [10] is one of the comprehensive models in this area. Security, scalability, openness, autonomy, RAS, agility, hierarchy, and programmablility are pillars of the OpenFog reference model. While some of these pillars such as openness and programmablility are important to represent a standard software model, we consider the other characteristics of this reference model for the rest of our study.

A. Challenges on fog computing

A *challenge* is any event that disrupts the normal operation of a network [18]. Furthermore, a *threat* is a challenge that exploits a *vulnerability* in the network to disrupt it. The result of this action is the failure to deliver specified network services if the error caused by the challenge manifests itself in the output. The common challenges for a communication network include malicious attacks, large-scale disasters, environmental challenges, unusual but legitimate traffic, human errors, sociopolitical and economic factors, and lower level failures [19]. We define resilience as *the ability of the system to provide and maintain an acceptable level of service in the face of various faults and challenges to the normal operation* [18], [20]. Figure 3 illustrates ResilieNets disciplines and categorises the common challenges. These disciplines are divided into two main groups, namely *challenge tolerance* and *trustworthiness*.

Challenge tolerance disciplines include all challenges to the network, split into the smaller subsets including survivability, traffic tolerance, and disruption tolerance. Survivability is divided into *many and targeted failures* such as natural disasters and fault tolerance that includes *few and random failures*. Unexpected legitimate traffic and abnormal traffic such as DDoS (distributed denial-of-service) attack traffic are categorized under *traffic tolerance*. *Disruption tolerance* includes all challenges related to the network environment such as delay, mobility, and connectivity, as well as device specific challenges such as energy. Furthermore, *trustworthiness* refers to measurable characteristics including dependability, security, and performability. In this study, we focus on challenge tolerance to fog computing in the context of the OpenFog reference architecture.

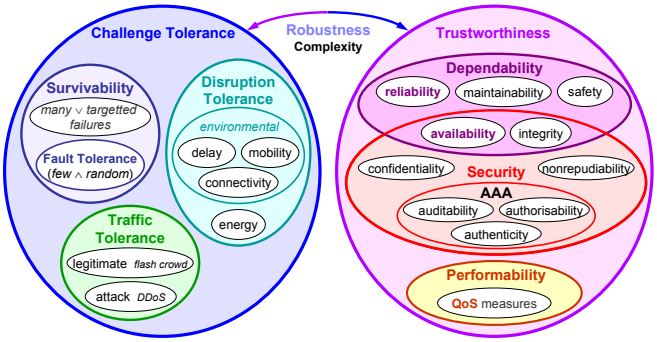

Fig. 3. ResiliNets disciplines [20], [18]

1) Survivability: Survivability considers all correlated (as well as random) failures. Correlated failures occur when a challenge affects a specific area of the network such as natural disasters or cascading failures through the network as well as when an attack is against multiple targets in the network. In IoT systems the Internet provides connectivity between the edge networks and the cloud. Studies have shown that the Internet as a *whole* is relatively resilient to local disasters and similar correlated challenges. There are many examples such as the 2011 Japan earthquake followed by a tsunami [21] and the 2007 Taiwan earthquake [22]. While in similar challenges a service disruption may occur in part of the Internet, the vast scale and distributed architecture of the Internet provides survivability. In IoT systems such as smart cities, the effect of service disruptions may manifest by disconnecting the edge networks from the cloud in the distressed area. Although the edge network may not be the target of the challenge, the service failure happens at the edge. If fog computing is added to the IoT system, another distributed layer between the edge network and the Internet containing computation, storage, and control that increases edge network autonomy. This autonomy leads to service continuity in the face of external service failure. While in this case the central decision making and long-term storage in the cloud are still an option, they will not be the only option.

The concept of autonomy presented in the OpenFog reference architecture [10] expands to discovery, management, security, and operation. Autonomy of discovery extends the ability of fog nodes to discover other resources including computing and storage in their proximity. This attribute promotes the survivability of the edge network encountering challenges to the local area.

Autonomy of management provides the ability to manage resources locally, including instantiation of services and provisioning of the environment around services such as rerouting of flows. This characteristic enhances the survivability of the edge networks.

Autonomy of security also contributes to the survivability of the system, while it promotes the trustworthiness. Currently, security is one of the biggest concerns in IoT systems. Recent DDoS attacks [23], [24] using *things* at the edge networks by malware such as Mirai [25], [26] have increased this concern. Autonomy of security in the fog nodes makes them one of the first lines of defense against security threats including DDoS. Fog nodes can authenticate minimal security services, even though the connection to the cloud has been disrupted. Autonomous reaction to threats and automatic security updates leads to higher survivability and trustworthiness.

Heterogeneity generally increases survivability by providing a diverse alternative [27]. Using various type of network connections at the edge networks and fog nodes provides higher survivability. Depending on the application, the network technology varies from low bit-rate, low-energy consumption, and long range such as IEEE 802.15.4 [28], ZigBee [29], and LoRaWAN [30], [31] to high bandwidth, short-range connectivity such as IEEE 802.11 and Bluetooth. Though neither of these groups are optimal for all applications, they can provide redundant paths for data and control flows until the challenge is resolved. Furthermore, the low price of communication devices using these technologies including Bluetooth LE, ZigBee, and LoRaWAN make it feasible to have all of them in one device. Current cell phones and set-top boxes are among the examples in which IEEE 802.11, Bluetooth, NFC, and LTE are currently implemented.

Fault tolerance causing resistance to a few random challenges improves in fog computing. These challenges are random failures in hardware, protocols, and software. The openness and programmability pillars of the OpenFog architecture promote heterogeneity in various aspects including software and hardware, leading to more tolerance against such challenges. Rich connection redundancy provided by various protocols and designed for diverse conditions at both edge and fog layers is another attribute that increases survivability.

2) Traffic Tolerance: Unexpected legitimate traffic over the provisioned network capacity can be a challenge for a network. Some of these examples are the high rate of phone calls in the public switched telephone network (PSTN) and flash crowd of Internet access on 9/11. Such traffic in underprovisioned network capacity leads to high delay and service failure. One of the main ideas behind the introducing fog computing is reducing the volume of high-bandwidth raw data in the core network. For example, a surveillance camera capturing 30

frames/s can generate approximately 1 TB/day [10], but all of this data may not be needed. The problem is amplified when the processing time and storing data in the cloud are added, in addition to the delay to return the response to the edge network. Hierarchical n-tier fog nodes are a potential solution for this problem. Edge fog nodes perform processing close to the edge. In a multi-tier hierarchy, each higher layer performs more processing and adds meaningful metadata related to the environment. The final result is then transferred to the cloud. The ultimate outcome is less traffic in the core network and the cloud that leads to the higher traffic tolerance benefiting legitimate traffic.

Malicious traffic is also recognized more easily when fog computing is applied. Some low-power, low data-rate sensors at the edge networks send data periodically at a regular rate. Though monitoring this regularly is not easy in the core network, it is at the edge and provides one of the solutions for source-based DDoS attacks. These solutions detect the DDoS attacks by monitoring both inbound and outbound traffic of the source network and consider the ratio of flows [32], [33]. Fog nodes can interrupt data flows with anomalies patterns by knowing the regular pattern in the proximity. Since the fog nodes act autonomously, they can configure themselves according to the normal traffic pattern in the area.

The hierarchy in the fog architecture increases overall security. One solution to prevent DDoS is filtering traffic at the edge routers [34]. Knowing the valid IP addresses in the internal network allows the routers to validate the source IP addresses and drop spoofed IP packets before leaving the internal network [35]. Furthermore, the hierarchical structure enables fog nodes with even less computing power to contribute to the security of the network.

The scalability attribute of fog computing has an important role in traffic tolerance. Adding extra nodes to the environment during high traffic increases traffic tolerance as well.

3) Disruption Tolerance: Environmental challenges including weak or intermittent connectivity, mobility, and unpredictably long delay are the primary reasons for network disruption. Wireless networks are susceptible to connectivity challenges due to obstacles in the line-of-sight and noisy environments. The design for mobility has increased the usage of wireless devices at the edge networks. Hence, the edge networks are more prone to connectivity challenges than the core network. The fog layer improves disruption tolerance by providing close connectivity to the edge networks and consequently the IoT system.

Delay is another cause of degraded performability and usability of a service, particularly for mission critical and interactive services. One of the main reasons for introducing the fog architecture is reducing the uncontrolled delay between the edge networks and the cloud. Installing fog nodes along with their resources at the edge networks reduces the round-trip time significantly for short request/response queries. More complex queries and those queries that require more global information can be handled in a multilevel fog topology with more capable devices at higher levels. For instance, an image enhancement algorithm can be executed on a surveillance camera while face recognition is performed in the first fog layer and higher decision making about the user access to an specific area is executed in the next layer. The long-term pattern access of a specific user can be calculated in the cloud. It is obvious that the first few operations that need low latency are performed near the edge and will not suffer WAN unpredictable long latency.

Mobility is one of the specific challenges to the wireless networks when the end nodes move causing dynamic topologies and weak or intermittent connectivity that leads to long delay and even service disruption. Heterogeneity in communication protocols and related hardware installed in the fog nodes makes various types of connections possible, suitable for various environments, transmission ranges, and bit rates, resulting in higher tolerance to connectivity and mobility challenges.

Source of energy is another essential issue for mobile devices used in the edge network. The form factor and level of device mobility are two essential attributes that define the size and type of the source energy. Energy consumption is more critical for small form-factor devices, usually installed remotely such as sensors. Moreover, losing the source of energy causes service disruption. Unfortunately, all of these challenges are common at the edge networks. Autonomous characteristics and computational resources of the fog nodes and their proximity to the edge makes them candidates to monitor the energy level of the remote devices and define a policy for conserving energy such as redirecting unnecessary traffic toward other nodes.

IV. Simulation

In order to study the role of fog nodes in the IoT system, we simulate a network with various communication protocols at the edge network and few levels of the communication network between the edge and the cloud to install the fog nodes. We use ns-3 [36] to perform this simulation. We consider two scenarios: The first performs direct communication between the edge nodes and the cloud to simulate an IoT system without a fog layer. In the second scenario, the connection between the edge nodes and the cloud is established through the fog nodes. Figure 4 illustrates this topology for the simulation. The houses are connected through ISPs and the core network to the cloud. Each house has five 802.15.4 nodes and five 802.11b nodes connected by Ethernet. The 802.11 access point connects the whole network to the ISP. The 802.15.4 nodes generate low bit-rate traffic while wireless workstations produce high bit-rate traffic. We use variously bandwidth between houses and ISPs to represent different services. We consider four nodes in the core network and two different destinations in the cloud layer. We simulate five different experiments: two with the cloud-only scenario and three additionally with the fog. The two cloud experiments use low bit-rate and high bit-rate traffic. For the three experiments with the fog scenario, we first consider a fog node in the house as part of the access point. In the second experiment, the fog

Simulation parameters	value(s)
Scenarios	cloud, fog
Transmission scenarios	to cloud, to fog
Simulation time	80 s
Number of runs	10
Packet size	400 B
Application type	unlimited bulk send
Transport protocol	TCP
Addressing	IPv6
Layer 2 protocol	802.11, 802.3, 802.15.4, PPP
Rate	4 Mb/s, 3 Mb/s, 5 Mb/s, 4 Mb/s

TABLE I
SIMULATION PARAMETERS

nodes are moved to the ISPs, while in the third experiment we consider two fog layers, one inside the house and the other in the ISPs. In order to illustrate a simple figure, we just show the fog layer at the ISP nodes in Figure 4. We use unlimited continuous bulk traffic from the sources to the destinations. While the senders always have data to transmit, 802.15.4 has lower traffic rate because of its bit rate and packet size. We also use 6LoWPAN as the adaptation layer over the 802.15.4 MAC to assemble/disassemble IPv6 packets to match the 802.15.4 data link frames. Table I shows the simulation parameters.

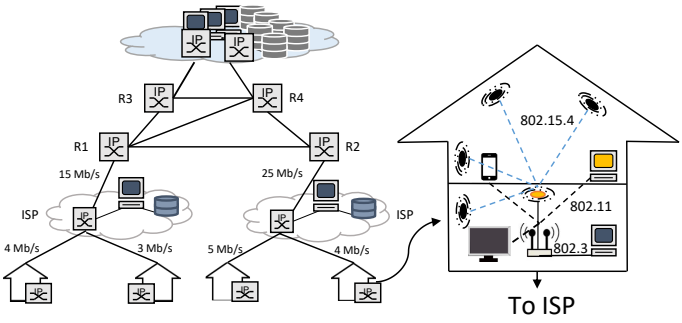

Fig. 4. Simulation topology

For the first experiment with the cloud-only scenario, IoT PAN (Personal Area Network) nodes send their low bit-rate data to the cloud nodes through their coordinator. Figure 5 illustrates the results for PDR (packet delivery ratio), delay, jitter, packet loss, and flow rate. Each house has four 802.15.4 nodes plus its PAN coordinator. Four nodes in two houses send data during the simulation. We observe in Figure 5 that the mean PDR is higher than 80 percent for most flows. In addition, the mean delay and jitter are also quite high. This is due to the fact that some packets are dropped over the links between the houses and ISP because of the low bandwidth. We observe that PDR is worse when the bandwidth between each house and the ISP reduces (the result not shown here). Though the sensors operating with 802.15.4 may not send data with a rate as high we simulate here, we assume for this study that the first bottleneck is the bandwidth between the edge and the core network. This results in long delay and high jitter between the cloud and the edge network for this case.

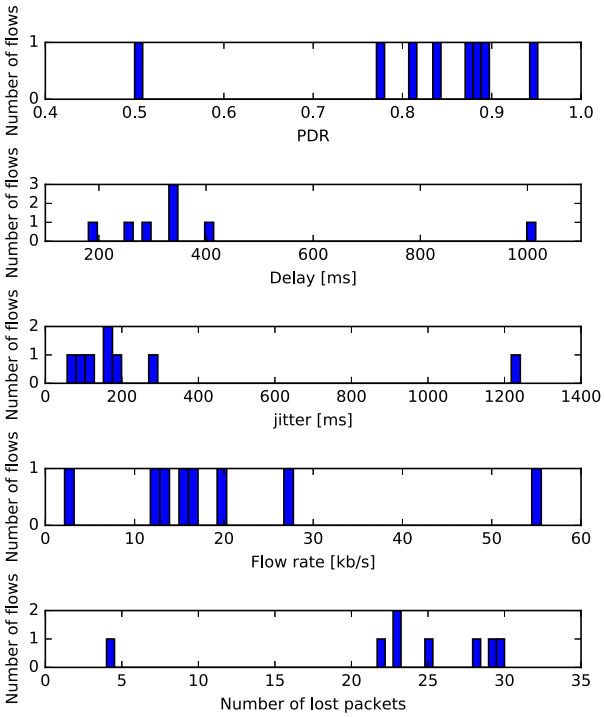

Fig. 5. Low bit-rate traffic from house to the cloud

The above result is amplified if we consider the same conditions in the first experiment but with high bit-rate traffic. Figure 6 shows the results when the wireless workstations send traffic instead of 802.15.4 nodes. The 802.11 nodes are less power-constrained with high data-rate. PDR decreases and delay increases significantly, due to the fact that more packets are dropped over the communication path between houses and ISP with retransmission is required. This effect confirms that having local processing and storage entities can reduce both delay and traffic over the access links. We should emphasise that we have considered high bandwidth links with low delay in the core network, therefore it does not affect delay as long as there is enough bandwidth capacity in the network. However, unusual traffic on the path between ISP and the cloud causes even longer delay and lower PDR. Though jitter still has a reasonable value, it does not make any difference with such a high delay. These results confirm that having a high-capacity edge network connected with a low bit-rate access link is unsuitable for interactive requests. While houses take advantage of high-bandwidth internal networks, the connections between houses and ISPs have not improved at the same pace.

In the next experiment, we consider a fog node as part of the access point in each house and we transmit traffic to this node. The results are illustrated in Figure 7. We observe that PDR is over 97 percent for all flows. As we mentioned before, we use unlimited TCP bulk traffic to saturate the network.

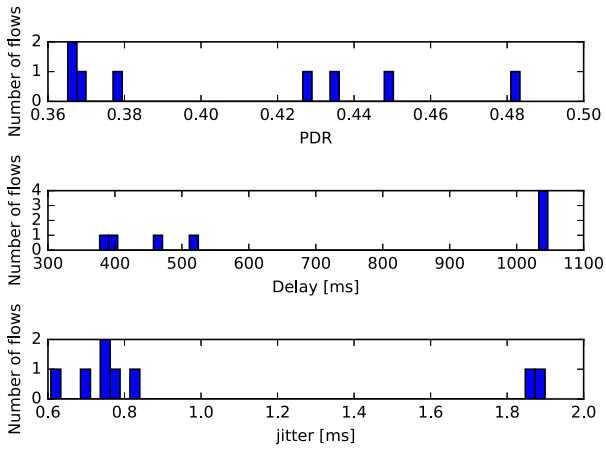

Fig. 6. High bit-rate traffic from house to the cloud

Therefore, high collision, packet loss, and retransmission are inevitable, with high delay resulting. We use 802.11b for all wireless experiments. Considering the short range and small number of connected devices in a common residential wireless network, the observed delay is high; however, it is still suitable for interactive requests. The low jitter is another good sign to support interactive requests. Another good reason to use this topology is that unexpected traffic flows inside the core network and heavy traffic load in ISPs do not affect the performance of the home network. In addition, transmitting processed instead of raw data decreases the overall network traffic.

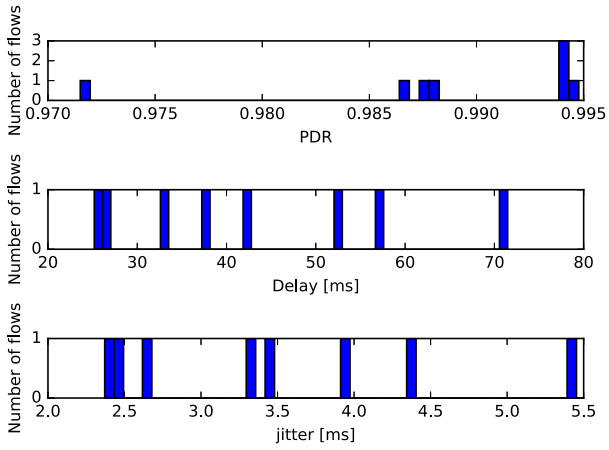

Fig. 7. Fog at the houses

The next experiment confirms that moving the fog nodes away from the edge network with the same bottleneck between the houses and ISPs produces a better result than sending the traffic directly to the cloud. In this experiment, we consider a fog node in each ISP. As is illustrated in Figure 8, nearly all flows have a lower delay than the similar experiment when we send high bit rate traffic to the cloud (Figure 6). We also observe higher PDR than the similar experiment with the cloud. This experiment confirms that even adding a fog layer farther from the edge network can reduce the network delay, while it makes the edge network less susceptible to challenges in the core network.

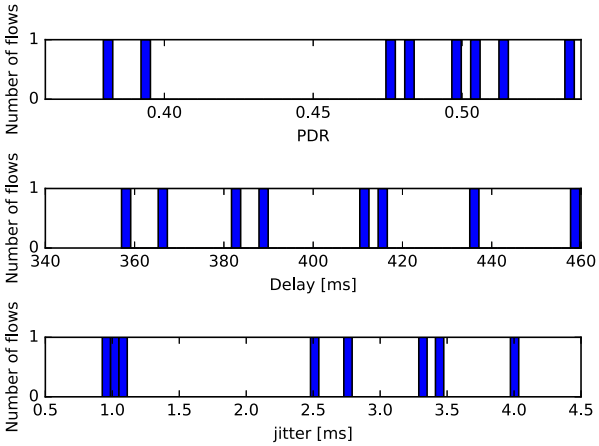

Fig. 8. Fog at the ISPs

In the last experiment, we consider two layers of fog nodes, one inside each house and the other layer in ISPs. We monitor the traffic between the access points and ISPs and transmit 30 percent of traffic generated in each house to ISPs during the simulation time; the results are shown in Figure 9. We observe higher PDR and lower delay. While the delay is still high because of high traffic at each house, the overall delay has reduced due to less traffic on the link between houses and ISPs. This experiment indicates that a multi-layer fog architecture can reduce traffic in the core network while it preserves the autonomy of the edge networks. The overall result is higher resilience against challenges in the core network. A multi-layer architecture can also improve the network resilience due to its distributed architecture and proximity to the edge. Nodes in each fog layer can cooperate with each other to reduce the effect of challenges in the edge networks.

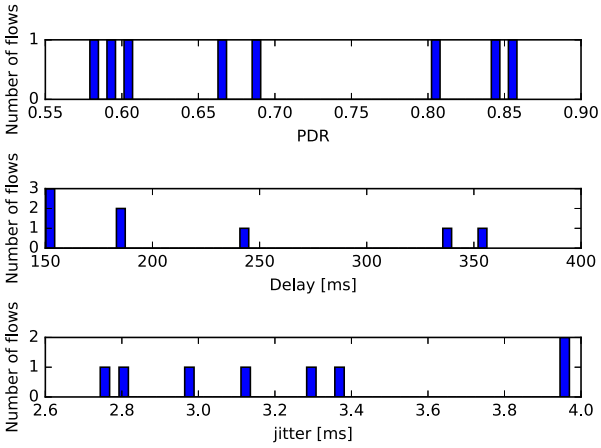

Fig. 9. Two-layer fog at houses and ISPs

V. Conclusion

In this paper, we explained how the fog architecture improves network resilience by providing autonomy through the local processing to the edge network. We conducted various network simulations to study network traffic with and without a fog layer. We applied various changes to the network architecture by adding fog layers and show how the fog architecture can affect network parameters. We used the latest fog architecture introduced by the OpenFog consortium. We also assert that having fog nodes in the edge networks make the architecture more appropriate for interactive requests than the cloud-only architecture. For our future work, we continue the simulation study by imposing challenges to the core network and also using an algorithm for load balancing among fog nodes at the edge. We expect that balancing the load among edge nodes leads to higher resiliency at the edge network due to decreasing delay and increasing the performability of the services.

References

[1] IEEE, "IEEE Standard Association, P2413." https://standards.ieee.org/develop/project/2413.html, May 2015.

[2] ITU, "Overview of the Internet of Things." https://www.itu.int/rec/T-REC-Y.2060-201206-I, June 2012.

[3] IETF, "The Internet of Things concept and problem statement." http://tools.ietf.org/id/draft-lee-iot-problem-statement-00.txt, 2010.

[4] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the Internet of (important) things," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 3, pp. 1389 – 1406, 2013.

[5] D. Evans, "The Internet of Things - how the next evolution of the Internet is changing everything." https://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, 2011.

[6] Cisco, "Cisco fog computing: Unleash the power of the Internet of Things." http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf, 2015.

[7] 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*, MCC '12, (New York, NY, USA), pp. 13–16, ACM, 2012.

[8] Cisco, "The Internet of Things reference model." http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf, 2014.

[9] O. C. A. W. Group, "OpenFog." https://www.openfogconsortium.org/, 2017.

[10] O. C. A. W. Group, "OpenFog reference architecture for fog computing." https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf, 2017.

[11] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 27–32, Oct. 2014.

[12] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.

[13] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[14] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using MapReduce." http://www.dtic.mil/docs/citations/ADA512601, 2009.

[15] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, pp. 14–23, Oct 2009.

[16] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. of the Sixth International Conference on Advances in Future Internet*, pp. 48–54, IARIA, 2014.

[17] Nokia, "Nokia website." https://networks.nokia.com/solutions/multi-access-edge-computing, 2017.

[18] J. P. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245–1265, 2010.

[19] E. K. Çetinkaya, D. Broyles, A. Dandekar, S. Srinivasan, and J. P. Sterbenz, "A comprehensive framework to simulate network attacks and challenges," in *IEEE/IFIP 2nd International Workshop on Reliable Networks Design and Modeling (RNDM)*, (Moscow, Russia), pp. 538–544, October 2010.

[20] J. P. Sterbenz and D. Hutchison, "ResiliNets: Multilevel resilient and survivable networking initiative." http://wiki.ittc.ku.edu/resilinets, 2016.

[21] Dyn, "Japan quake." http://dyn.com/blog/japan-quake/.

[22] A. Popescu and T. U. E. Zmijewski, "Quaking tables: The Taiwan earthquakes and the Internet routing table," in *APRICOT*, (Bali), Renesys Corp, 2007.

[23] S. Moss, "University suffers DDoS attack from IoT vending machines." http://www.datacenterdynamics.com/content-tracks/security-risk/university-suffers-ddos-attack-from-iot-vending-machines/97808.article, 2017.

[24] U. Lindqvist and P. G. Neumann, "The future of the Internet of Things," *Commun. ACM*, vol. 60, pp. 26–30, Jan. 2017.

[25] Wikipedia, "Mirai (malware)." https://en.wikipedia.org/wiki/Mirai_(malware), 2017.

[26] J. Biggs, "Hackers release source code for a powerful DDoS app called Mirai." https://techcrunch.com/2016/10/10/hackers-release-source-code-for-a-powerful-ddos-app-called-mirai/, 2017.

[27] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Redundancy, Diversity, and Connectivity to Achieve Multilevel Network Resilience, Survivability, and Disruption Tolerance (invited paper)," *Telecommunication Systems*, vol. 56, no. 1, pp. 17–31, 2014.

[28] IEEE, "IEEE standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr- wpans)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, Sept 2011.

[29] Zigbee, "ZigBee Alliance." http://www.zigbee.org/, May 2015.

[30] A. LoRa, "LoRa Alliance." https://www.lora-alliance.org/What-Is-LoRa/Technology, 2016.

[31] A. LoRa, "LoRaWAN Specification." https://www.lora-alliance.org/For-Developers/LoRaWANDevelopers, 2016.

[32] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," in *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pp. 312–321, IEEE, 2002.

[33] T. M. Gil and M. Poletto, "MULTOPS: A data-structure for bandwidth attack detection.," in *USENIX Security Symposium*, pp. 23–38, 2001.

[34] P. Ferguson, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing." RFC 2827, 2000.

[35] A. Modarresi, S. Gangadhar, and J. P. G. Sterbenz, "A framework for improving network resilience using SDN and fog nodes," in *RNDM'17 - 9th International Workshop on Resilient Networks Design and Modeling (RNDM 2017)*, (Alghero, Sardinia, Italy), Sept. 2017.

[36] n. Consortium, "ns-3 website." https://www.nsnam.org/, 2015.