Edge Computing Enabled Smart Firefighting: Opportunities and Challenges

Xiaopei Wu
Wayne State University
Detroit, Michigan
xiaopei.wu@wayne.edu

Robert Dunne
Wayne State University, Detroit Fire Department
Detroit, Michigan
rkdunne@wayne.edu

Qingyang Zhang
Anhui University, Wayne State University
Hefei, China
qyzhang@wayne.edu

Weisong Shi
Wayne State University
Detroit, Michigan
weisong@wayne.edu

ABSTRACT

By collectively leveraging advanced communications systems, sensing, drones, wearable technologies and large-scale data analysis, smart firefighting is envisioned as the next generation firefighting with the capacities of gathering massive real-time scene data, transferring them into useful information and insights for fire responders, and even providing them with more safe and accurate decisions. For smart firefighting, timeliness and accuracy are two foremost system requirements, yet they are unsatisfied in many applications. One reason for such dilemma is due to the underlying used computing architecture (*i.e.* cloud computing) that can produce extra latency in large-scale data transmission. To address this problem, we explore the firefighting field utilizing edge computing and discuss the overall system architecture, opportunities, challenges, as well as some early technical suggestions on building edge-enabled smart firefighting. To validate the feasibility of edge computing, we simulate the firefighting context and respectively deploy a video-based flame detection algorithm on a local Intel's edge computing platform and a remote Amazon EC2. The preliminary results show that edge computing can significantly increase system's reactive speed, with on average 50% reduction in system latency.

CCS CONCEPTS

• **Computer systems organization → Distributed architectures**;

KEYWORDS

Timeliness, Accuracy, IoT, Smart Firefighting, Cloud Computing, Edge Computing

HotWeb'17, October 14, 2017, San Jose / Silicon Valley, CA, USA

ACM ISBN 978-1-4503-5527-8/17/10…$15.00
https://doi.org/10.1145/3132465.3132475

ACM Reference format:
Xiaopei Wu, Robert Dunne, Qingyang Zhang, and Weisong Shi. 2017. Edge Computing Enabled Smart Firefighting: Opportunities and Challenges. In *Proceedings of HotWeb'17, San Jose / Silicon Valley, CA, USA, October 14, 2017,* 6 pages.
https://doi.org/10.1145/3132465.3132475

1 INTRODUCTION

Firefighting is the act of using different kinds of techniques and equipment to extinguish fires, rescue trapped people and minimize casualties and property damage. On the fireground, accurate and timely data is vital for effective firefighting. The more data the fire responders have (such as the firefighter's localization and physiological condition, burning building's floor-plan, the hazardous area's location, and the number of trapped occupants and their corresponding location), the higher opportunities of saving more people, guaranteeing firefighter's safety and limiting fire damage.

However, fire responders still rely on the outdated, inefficient and even unreliable equipment or systems: field data is very limited and the rescue tactics are made by Incident Commanders (IC) based on their own experiences. The evolving new technologies including sensing, drones, wearable devices, communication systems, artificial intelligence and IoT are enabling us to collect vast amounts of real-time data, extract useful information and even guide safer rescue tactics for the fire responders. Modern firefighting is also referred to as smart firefighting by the NIST's research roadmap plan [9]. The Department of Homeland Security launched the NGFR program [14] to leverage various innovative technologies to make fire responders more protected, connected and fully aware.

For future firefighting, a large volume of field data will be produced by a variety of field devices. Yet, if they not well processed, the data might be useless or even misleading to the fire responders. Thus, different advanced data processing, analytics as well as algorithms will be jointly or separately exploited to extract, learn, and estimate useful and meaningful information (*e.g.* firefighter's location on the floor-room granularity) for the IC. Nevertheless, such process is usually computation-intensive and energy-draining, making it inappropriate to execute on the local on-site, resource-limited devices.

In contrast, the cloud computing paradigm with the low-cost, unlimited resources center, is the best candidate for the large-scale data processing and complex algorithm execution. In reality, many firefighting-related systems or ongoing projects have proposed to depend on the cloud such as the localization and tracking from the TRX System's NEON Personnel Tracker [8] and Precision Location and Mapping System in [13]. However, the cloud is Internet-based, and its issue is the extra latency in interacting with the remote cloud. This problem is becoming much more severe as the prevalence of IoT-based applications with unprecedented volume and a variety of data generated each day. Obviously, the traditional cloud computing model is not quite fit for the time-crucial firefighting system. Take the indoor firefighter localization and tracking system as an example. Such system usually needs to run the localization algorithm frequently (around one time per 30 seconds). In addition to the traditional radio raging based triangulation, extra sensor fusion algorithm is introduced to calibrate the position accuracy. When using the cloud computing model, the position precision will be definitely guaranteed due to the calibration, while the response time might fluctuate significantly as the dynamics of available network bandwidth and workload on the cloud.

On the fireground, each second matters and accurate data is highly desirable. To balance the trade-off between these two metrics, we envision to apply a new computing model-edge computing-into the time-sensitive firefighting field. This is an emerging architecture which will perform the complex data processing and analysis in the proximity of the data source, instead of on a remote cloud server, to improve the system response time, conserve network bandwidth and potentially address the security concerns and privacy exposure.

In this paper, we first introduce edge computing. Then we present several popular time-sensitive applications for effective smart firefighting. The potential opportunities, system architecture and challenges about edge computing enabled smart firefighting are discussed in Section 4. In Section 5, we simulate the firefighting context and demonstrate the performance advantages of edge computing in reducing the system latency and conclude this paper in Section 6.

2 WHAT IS EDGE COMPUTING?

Edge computing([15, 17, 18]), also referred as fog computing([10–12]), is a new distributed computing architecture that can break up complex computational workloads into small elements and perform them on local edge devices/nodes, instead of offloading large volume data and workloads on a remote cloud data center. Here, the edge devices/nodes can be any device with computing, storage, and network connectivity, located at the frontiers of the Internet. For example, they are not only the data producers such as the sensors, surveillance cameras and smartphones, but also the communication gateways (*e.g.* switches, routers, access point and base station).The core of edge computing is to offload computational tasks in the proximity of data source, either on the devices where the data is generated, or on the nearby gateways. Doing so can eliminate the data transmission over the Internet, saving the network bandwidth; thus alleviating the Internet traffic burden. Without large scale communication cost, the edge computing based applications enable the edge nodes/devices to react locally and significantly reduce the system latency. Edge computing is suitable for time sensitive applications, especially for those taking the resource-constrained devices' readings as input, meanwhile, depending on complex data processing/algorithms to obtain actionable and immediate output.

Note that edge computing is not mutually exclusive with cloud computing. They can coexist within a same application, in which the edge is responsible for processing the time-sensitive functions while the cloud is responsible for less time sensitive operations, such as historical analysis and long-term storage. Whether we can depend on the cloud server totally depends on the application's requirement of response time. In particular, given a specific application with data from various sensors, the developer can consider the hybrid model if some data needs to be processed and acted on with very small tolerable latency (*e.g.* a few seconds), and some can wait minutes or even a few hours for action.

3 SMART FIREFIGHTING

The key of smart firefighting is to leverage the IoT to integrate all data. According to the report in [9], smart firefighting contains three elements: (1) data gathering from a range of sources; (2) data processing, analysis and prediction; and (3) effective dissemination of processed results to the fire responders or other related stakeholders.

Smart firefighting contains many applications. In this paper, we limit our focus on the real-time applications with data collection and processing, the research areas that might benefit from using edge computing. To be specific , they are *Situational Awareness*, *Intelligent Safety Decision-making*, and *3D Fire Ground modeling*. More research topics or applications could be found in [9].

3.1 Situational Awareness

Situational awareness is comprised of many aspects including the localization and tracking, hazards detection and firefighter health condition monitoring. Improved situational awareness will enable the fire responders to recognize and avoid potential hazards, and enhance fire personnel's safety. A series of sensors and field devices are the source of data to improve situational awareness.

3.1.1 Localization and Tracking. Indoor firefighter localization and tracking has been explored since 1999. The main challenge is to pinpoint the fire fighter's location within a few meters over a few seconds, with significantly higher demand in both the precision and timeliness. Many solutions are radio-ranging based, already demonstrating success in the traditional indoor environment. The radio technology contains a localization infrastructure with some "anchors" deployed at known positions, intermittently emitting radio signals to radio transmitters worn by each firefighter to range the distance between the firefighter and the known anchors and then determine the firefighter's position through different localization algorithms (*e.g.* triangulation, fingerprinting or proximity). Given the extreme conditions of such environment, all these solutions suffer from low precision, leading to the spate of multi-sensory fusion calibration scheme, which involves a large scale sensor readings analysis and system integration[8][13]. As the more precise enhanced system usually includes highly sophisticated algorithms, the cloud architecture is widely adopted. The limitation of cloud-based solution is the longer system response time, produced by transmitting large volume of sensor data over the Internet. Furthermore, due to the prevalence of IoT, an unprecedented volume and variety of data are entering into the Internet, severely making the Internet traffic much more crowded than ever before, and thus bringing a huge challenge for the time-critical cloud- based localization.

3.1.2 Hazards Detection & Occupant Counting. Anything can happen on the fireground, and the lack of hazard awareness would add to the danger. Identifying the hazard's location and number of occupants can speed rescue. For such applications, smart firefighting will rely on video analytics to distill invaluable information from body cameras and surveillance cameras in on the firegrounds. For example, while the rescue team is in the search mission, correctly detecting the flash-over and toxic gas, then quickly broadcasting its location to all fire fighters may allow them to avoid risky areas. Some other types of hazards are falling ceiling, collapsing wall, or even chemical gas emission. This data can aid in routing fire teams. Moreover, the images or videos from the surveillance cameras are sources for discovering how many occupants are trapped and even where they may be. Similarly, video and image analytics require large computing power, which requires a cloud-based service. Extensive video data, significantly much larger than the regular sensor readings, will go through the Internet and may adversely affect the real-time performance.

3.2 Intelligent Safety Decision Making

The rapid advances of communication systems, sensing technologies and IoT are giving the fire responder access unprecedented large volumes of data, which is not limited to fireground data. The data can also come from the Internet, such as demographic reports, building blue-prints, public social media postings and so on. As a result, the data might be overwhelming and may increase the risk of distracting the fire responders and even causing them to make wrong and unsafe decisions.

A more fully integrated type of application tries to make the IoT data actionable and useful, usually depending on a series of machine learning and artificial intelligence to analyze the volume of data, turn them into actionable knowledge, and ultimately form real-time recommendations and decisions for the fire responders. Imagine the next generation fireground: various wearable sensors and devices in the firefighter's clothes can sense the position, health condition, the presence of dangerous chemical gases, environmental heat and much more; drones can see the fireground's aerial imagery; robotics with cameras and sensors can enter into dangerous areas to see debris and report other important environmental parameters such as heat, smoke density and others. This data coupled with an automated intelligent safety decision system can help the firefighter find exits, identify hazards, warn them about surrounding temperatures, and estimate the probability of explosion.

AI based research for fire responders is still at the early stage. So far there is only one ongoing project from NASA JPL, also referred as AUDREY[16], which is exploring the application of artificial intelligence to help fire responders make safe and split-second recommendations in dangerous situations. Apparently, AUDREY is highly time-sensitive. Yet the AI-based large-scale data analysis and reasoning determines AUDREY can only perform core computation on a remote cloud, which in turn extends the system response time.

3.3 fireground 3D Modeling

fireground 3D modeling enables the IC to know the situational information in 3D space. Coupled with a localization system, it can display each firefighter's position in a more meaningful way (*i.e.* on which floor and which room). Unfortunately, 3D structure modeling requires many input parameters such as the building's height, number of floors, shape and the inside floor plan and so on, while this may be unavailable for the firefighting scenario with little or even no prior knowledge about the burning structure. Our ongoing project FAST[3] attempts to the building's dimensions and the inside floor plan from the aerial images taken by drones and the open data portal for city construction.3D structure modeling requires significant computing power. Rather than purely depending on the cloud model, we propose to add edge computing into our computing architecture.

4 EDGE COMPUTING ENABLED SMART FIREFIGHTING

In this section, we first discuss the potential opportunities for smart firefighting, then propose the overall architecture for edge computing based smart firefighting. Last, the challenges of using edge computing in the firefighting context are discussed.

4.1 Opportunities

The edge computing aims to reduce system latency and improve system's reactive speed by offloading computational task to one or more resource-appropriate edge nodes, which are in the proximity of the data source. Obviously, this new computing model shines bright lights on applications with higher requirements on both accuracy and timeliness. By applying edge computing to smart firefighting, we anticipate two main opportunities for fire responders:

Timeliness: As the field collected data will be processed or analyzed on geographically local devices, instead of a distant cloud center, the system response time could be enormously reduced by saving the data transmission cost and other extra time consumption in the cloud.

Accuracy: Many efforts in the edge computing community are focused on enhancing the edge nodes' computing capacity. Some, like the local distributed system, focus on software solutions while others work on developing more powerful processing chips. Even though edge computing is still in the early phase, all these efforts will definitely enable the edge server either on single or multiple devices to execute complex algorithms and employ data analytics to obtain more accurate results.

4.2 System Architecture

The architectural depiction of the edge computing enabled smart firefighting is illustrated in the Figure 1. The system consists of the fireground sensing components, routing network and the remote cloud server. For the first two elements, there are many different types of devices, which have the potential of offloading workload. The cloud server is kept to store large-scale data for future historical analysis and conduct off-line model training or other latency tolerant applications.

Figure 2 shows currently available edge devices and edge nodes for firefighting and their corresponding distances to the field data

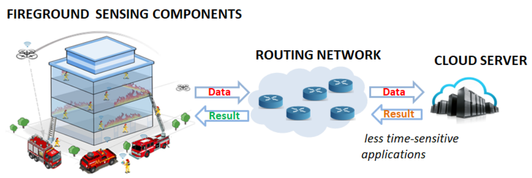


Figure 1: The architecture of edge computing enabled smart firefighting.

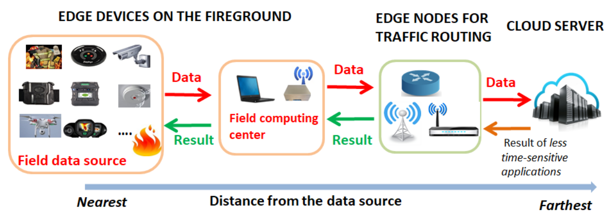


Figure 2: Involved devices in the edge computing enabled smart firefighting system.

source. The objective for edge-enabled smart firefighting is to harness all computing resources close to the data sources, including the original data producer, the field edge devices on the fire vehicle as well as the routing nodes/mobile base stations located at the frontiers of core Internet. The computational task requiring immediate reactions will be deployed on either the edge nodes or field edge devices, physically close to the data source with only one-hop or two-hop away.

Field edge devices: The edge devices either worn by the firefighter or newly installed around the fire scene are field data sources. For example, the tiny sensors embedded in the protective gear, or other physiological sensors (*e.g.* Zephyr BioHarness[7]) are responsible for continuously tracking each firefighter's heath data, such as heart rate, heart rate variability, and other physiological factors. The location unit[8] is the source of firefighter's indoor location data. The infrared body cam and toxic gas sensor can be used to detect the hazardous events on the fireground. Furthermore, existing surveillance cameras could be used to estimate the number of trapped occupants and their locations.

Additionally, on the fire vehicle, there is a local centralized data center, also referred as base station, usually deployed on a laptop, operated by the IC, providing the user interface for monitoring and tracking. Most fire vehicles also contain a mobile broadband router, with the capacities of converting the broadband cellular signals (*i.e.* 4G/LTE) to WiFi and creating a local WiFi hotspot. Compared with the field sensors, they have much more capacity both in computation and storage. Thus, we refer them as the field computing center in Figure 2.

Edge nodes: These devices include the routers, base stations, switches, as well as their corresponding capacity-added nodes in storage or computation. Besides the traditional Internet traffic routing, in the edge computing model, they are also responsible for processing complex data analysis and algorithms for time-sensitive applications.

Cloud server: In the edge computing enabled smart firefighting, the field data used for historical analysis will be transmitted to the cloud center when the Internet connection is available. And the less time-sensitive applications such as the dynamic model training, can deploy on the cloud server as well.

Wireless data communication: The wireless communication network is another essential component of edge-enabled smart firefighting. Two types of wireless networks are widely used. First is the *infrastructure-less* networking, in which field devices can form a network in ad-hoc manner and communicate with each other without any pre-existing infrastructure. This network will handle the data transmission between the field devices and the base station. Second is the *infrastructure-based* communication network between the base station and the remote cloud server. Currently, the base station can depend on the broadband cellular network (*i.e.* 4G/LTE). Furthermore, with the rapid development of smart cities, the free public WiFi is ubiquitous, making the IEEE 802.11 radio-based WiFi as another option for Internet access.

4.3 Challenges

Despite of the superiorities of edge computing for time-sensitive applications, there are some challenges when using it in the smart firefighting context.

Challenge 1: Where is the edge server?

The edge server includes a set of edge devices/nodes that are chosen to locally execute computational tasks, which are traditionally deployed on the remote cloud server. In theory, the network bridge nodes (*i.e.* edge nodes in Figure 2) are the ideal options for edge server due to their more powerful capabilities in computation and storage, compared with the extreme edge devices on the fireground. These nodes are usually located one-hop or two-hop away from the field sensors.

In the extreme application context, however, there are many challenges for deploying these nodes as edge servers. Unlike other real-time applications, the network connectivity between the field sensors and such gateway nodes is unreliable regardless of what kind of Internet access method (*e.g.* WiFi or 4G/LTE) is used. The power outage often associated with structure fires might break down the WiFi infrastructure. The available cellular network's bandwidth would be dynamic and unpredictable with wireless radio interference, building materials and geographic location.

Instead of purely relying on the gateway nodes, the applications for firefighting can deploy the edge server on the field devices, including different types of sensors, drones, robotics, body cameras, IC's laptop, mobile hotspot and other devices with computing capacities. The edge server could be on a single edge device/node or a virtual computing center across multiple devices. For example, FAST [3] proposes to deploy the 3D structural modeling on the drone.

Note that, unlike the drones, laptops or cameras, the field sensors have very little computing power, to make them support local real-time analytics and data processing, the researches can rely on the Apache Edgent[2], which is a lightweight programming model and micro-kernel style run-time which could be embedded in small sensors to accelerate the small sensor's processing power. Take the application of detecting gas emission as an example. Traditionally, the relevant sensors worn by the fire responders continuously sense and stream the sensors' readings to the base station to analyze.Using the Apache Edgent, the small sensor can locally run the data analysis algorithm. Additionally, it also includes a remote computing center feature to process complex data. As the network connectivity to the Internet is unreliable, the researchers can tailor or custom the base station center as the remote data center.

Applications involving complex machine learning or artificial intelligence can depend on the local distributed system, in which the local field devices will first form a resource pool, then a local centralized management center execute the task divisions, then offload all subtasks on the resource-appropriate edge devices and at last assemble the final result based on each subtask's output. Another option is to choose hardware containing advanced chips to support machine learning and deep learning. For example, the GPU from the NVIDA could be used for the training task. Once the training is done, the CPU from many mainstream vendors can conduct the reasoning and inference tasks, like the Intel's Xeon and FPGA, Qualcomm Snapdragon, Google's TPU, NVidia's Jeston[5].

Challenge 2: Load balancing and energy conservation

Moving the computing server further close to the data source has the benefit of saving communication cost and reducing system latency. However, the biggest concern is the associated energy consumption. In the fireground, all edge devices are battery-operated. The chosen edge server's battery will drain quickly due to the extra computation workload from other edge devices. For this problem, dynamic edge server election might be a solution, which will allow the qualified edge devices, based on their current residual energy level, to automatically enter/exit the resource pool to contribute their compute resources.

Challenge 3: Lightweight data processing and algorithms

Compared with the cloud server, the computational ability of edge devices/nodes is still limited. This is because they all have their own workloads and the priority of running as an edge server is usually less than their primary task. For instance, the router, if allowed to be configured as an edge server, will assign limited resources or limited time (*e.g.* during off peak hour) to execute computational task from other devices, causing minimum impact on its traditional workloads in data routing and transmitting. Therefore, if the researchers would like to enhance the real-time performance using edge computing model, they need to furthermore optimize their algorithms to maximally reduce the computational complexity.

Challenge 4: Task partitioning and work offloading

For the extreme field edge device, it is hard for a single device to satisfy the computational demands of the complex tasks, such as machine learning based decision making. In such case, the edge devices that qualify as an edge server can form a local distributed system to cooperatively work on a specific task. Unlike the task partitioning for regular edge applications discussed in [10, 12, 15, 17, 18], the targeted server is energy and resource constrained. In addition, the edge server is highly dynamic as the edge server needs to exit the resource pool to conserve energy for its own workload. So, in the firefighting context, the question is how to effectively partition a task from the energy perspective and offload each subtask on the resource appropriate edge server.

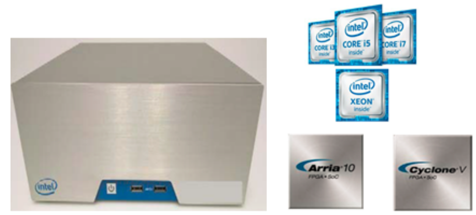


Figure 3: Intel's FRD and its core processing components.

5 PRELIMINARY VIDEO ANALYTICS

In this Section, we will investigate the performance of video analysis on a local edge server and a remote cloud center, respectively. Here, the video means the real field data from the firefighters' body cam, one most important data source for enhancing the situational awareness.

Owing to the mature digital video technology, currently, body cams are being used by many fire departments. In fact, there are plenty of helmet-mounted cameras on the market, equipped with WiFi interface and large volume memory card, with the prices ranging from \$100 to \$400. These cams have the capability to transmit live video, audio, high quality snapshots, along with GPS location in real-time to the base station. Due to the limited bandwidth typically available at fire scenes, body cams are currently just used for on-site video recording, post-incident analysis, investigation and training.

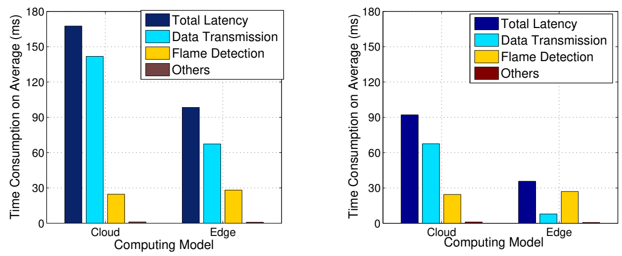
5.1 Experimental Set Up and Assumption

Our main goal is to prove the feasibility of using edge computing in firefighting, especially its potential in improving the response speed. So we set up a simulated environment, using two laptops at different positions to represent the data center on the fire vehicle and the fire fighter's body cam on the fireground, respectively.

The computational task we focus on is the real-time video flame detection, which is usually computationally intensive and not appropriate to execute on the power and resource limited edge devices.

The algorithm we chosen is Support Vector Machine based, requiring a group of prior videos as training to learn a model [6]. We deploy this algorithm on a local edge device and a remote cloud center, which are discussed in detail below.

Edge server: We employ Intel's Fog Reference Design (RFD)[4], a test bed demonstration for fog and edge computing in a self-contained enclosed chassis as shown in Figure 3. RFD is still in the early prototype phase, with limited distribution to the OpenFog university members through Intel's Altera University program. To execute complex computational tasks locally, RFD equips with high compute performance with Intel Core i3/i5/i7 or Xeon Processor. It also contains Intel's FPGA solutions and tools to provide programmable logic for specialized functions. Its software includes the Ubuntu 16.04 Desktop AMD64 , Open Source BIOS, video analytics, DL/ML, hardware acceleration, time sensitive networking, and many others. In addition to the powerful ingredients, RFD is very small and can be put on the fire vehicle to serve as a local edge server.



(a) Using public WiFi . **(b) Using 4G/LTE.**

Figure 4: Comparison of time consumption.

Cloud server: We rely on Amazon Elastic Compute Cloud (Amazon EC2) for US East (Ohio) Region, geographically located in New Albany, Ohio. To be specific, we deploy the fire flame detection algorithm on the t2.2 xlarge compute instances, which features 32 GB of memory and 8 vCPU [1].

Assumption: We assume that the fire occurs in Detroit, USA, around 200 miles from the cloud center and the FRD is physically located close to the local data center within USB cable communication distance. We further ignore such cable based transmission delay as it provides much higher transmission speed than the wireless network, producing a very small or even negligible delay. In other words, the simulated body camera will stream video to FRD directly. The video we used is the test fire video clips from [6].

As the communication infrastructure, the open free public WiFi and 4G/LTE are both considered, with their corresponding download/upload speed as around 11Mbps/5Mbps and 24Mbps/21Mbps. We assume the body cam is equipped with built-in WiFi interface and can only communicate with other devices through WiFi connection. So we configure FRD as a WiFi hotspot for 4G/LTE testing case.

5.2 Experimental Results

Figure 4 illustrates the average latency for edge and cloud using two different wireless communication infrastructures. The time consumption for different operations are explored as well. Here, the operations we consider include the data transmission, flame detection, frame decoding and processing (denoted as Others in the Figure 4). As expected, the edge computing model behaves significantly better than the cloud model in terms of total latency. On the other hand, the edge's performance gains are more obvious in the case of using 4G, with around 61% latency reduction from 92.02ms for the cloud model and 35.7ms for the edge model.

In spite of the higher bandwidth of WiFi, surprisingly, the performance of using 4G/LTE overall outperforms that of using WiFi. The cellular network's advantages actually benefits from the 1-hop direct communication between the simulated body cam and the FDR (*i.e.* configured WiFi hotspot). In addition, regardless of what wireless infrastructure is used, we also note that the time duration for detecting a flame is very close for cloud and edge, with the former performing slightly better than the latter. This demonstrates that Intel's FDR is powerful enough to process the computation-intensive tasks, and thus is a very good fit for the edge computing model.

6 CONCLUSION

In this paper, we reviewed the IoT based smart firefighting and summarized several desirable time-sensitive applications for the fire responders. We then pointed out the edge computing enabled architecture, opportunities and challenges for smart firefighting. Last, we investigated the feasibility of edge computing through a video based fire flame detection. The total latency is significantly reduced when using the edge computing model. We expect this article to inspire researchers in smart firefighting field in order to reconsider their computing architecture to enhance the system quality in the perspective of timeliness and accuracy.

ACKNOWLEDGMENTS

The authors would like to thank Intel's support for providing edge/fog computing demonstration platform through the Altera University program. The authors would also like to thank other members of the project team: Dr. Zhifeng Yu and Sheng Tan for their helpful comments and suggestions.

REFERENCES

[1] [n. d.]. Amazon EC2 Instance Types. ([n. d.]). https://aws.amazon.com/ec2/instance-typess
[2] [n. d.]. Apache Edgent. ([n. d.]). http://edgent.apache.org/
[3] [n. d.]. FAST: Firefighting Assitant SysTem. ([n. d.]). http://mist.cs.wayne.edu/EdgeCOPS/index.html-FAST
[4] [n. d.]. Intel Fog Reference Design Overview. ([n. d.]). https://www.intel.com/content/dam/www/public/us/en/documents/design-guides/fog-reference-design-overview-guide.pdf
[5] [n. d.]. A Machine Learning Landscape: Where AMD, Intel, NVIDIA, Qualcomm And Xilinx AI Engines Live. ([n. d.]). https://www.forbes.com/sites/moorinsights/2017/03/03/a-machine-learning-landscape-where-amd-intel-nvidia-qualcomm-and-xilinx-ai-engines-live/49832358742f
[6] [n. d.]. A video-based multi-feature flame detection system. ([n. d.]). https://github.com/liberize/flame-detection-system
[7] [n. d.]. ZEPHYR performance systems for fire responders. ([n. d.]). https://www.zephyranywhere.com/users/first-responders
[8] 2016. NEON Personnel Tracker. (2016). http://www.trxsystems.com/personnel-tracker.html.
[9] Albert W. Jones Anthony P. Hamins, Nelson P. Bryner and Galen H. Koepke. [n. d.]. Research Roadmap for Smart Fire Fighting. ([n. d.]). Retrieved July 1st, 2017 from https://www.nist.gov/publications/research-roadmap-smart-fire-fighting?pub_id=918636
[10] Dimitrios S. Nikolopoulos Blesson Varghese, Nan Wang and Rajkumar Buyya. 2017. Feasibility of Fog Computing. (Jan. 2017). https://arxiv.org/pdf/1701.05451.pdf
[11] Fog Computing. [n. d.]. OpenFog Consortium. ([n. d.]). https://www.openfogconsortium.org/
[12] Cisco White Paper. [n. d.]. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. ([n. d.]). https://www.cisco.com/c/dam/en-us/solutions/trends/iot/docs/computing-overview.pdf
[13] Bob Scannell. [n. d.]. Sensor Fusion Approach to Precision Location and Tracking for First Responders. ([n. d.]). http://www.electronicdesign.com/embedded/sensor-fusion-approach-first-responder-precision-locationtracking
[14] Homeland Security. [n. d.]. Next Generation First Responder Apex Program. ([n. d.]). https://www.dhs.gov/science-and-technology/ngfr
[15] Qun Li Shanhe Yi, Cheng Li. 2015. A Survey of Fog Computing: Concepts, Applications and Issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*. ACM, New York, 37–42.
[16] AUDREY Fact Sheet. 2016. Assistant for Understanding Data through Reasoning, Extraction and Synthesis (AUDREY). (Aug. 2016). https://www.dhs.gov/sites/default/files/publications/Audrey2-fact-sheet-508.pdf
[17] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal* 3, 5 (October 2016).
[18] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S. Nikolopoulos. 2016. Challenges and Opportunities in Edge Computing. In *IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE, New York, 20–26.