ISBN 978-981-14-1455-8
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
WCSE_2019_SPRING
Yangon, Myanmar, February 27-March 1, 2019, pp. 11-16
doi:10.18178/wcse.2019.03.003

Prototype Framework of Low-Latency Collection of Multi-Point Video Stream over Multi-Hop Multi-Channel Wireless Network

Aung Myo Htut[1+], Juseong Kim[2], JongWon Kim[2] and Chaodit Aswakul[1]

[1] Wireless Network and Future Internet Research Unit, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand,

[2]Networked Computing Systems (NetCS) Laboratory, School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea.

Abstract. Big cities are facing heavy road traffic congestion. The main reason is the increasing number of vehicles beyond the road network capacity, which forces traffic police at times to be incapable of controlling the inevitable traffic jams. To resolve those traffic jams, traffic police need the near real-time information of the road network e.g. from traffic cameras. In this paper, a prototype framework of low-latency collection of multi-point video streams over a multi-hop multi-channel wireless network is proposed. Video frames are captured by Raspberry Pi cameras and sent by the wireless link to the herein-called IoT Concentrators. IoT-Cloud Hub then pulls and stores those video frames from the IoT Concentrators. Users (traffic police) can watch traffic videos from the display screen of the IoT-Cloud Hub. As a proof-of-concept implementation, two Pi Cameras have been set up with Apache Kafka Producers and two additional Raspberry Pi's have been set up as the IoT Concentrators, which run the leader and follower Apache Kafka Brokers. To complete our system, an Intel® NUC as IoT-Cloud Hub and another Intel® NUC as Orchestration-Hub have been also set up. We assume the available wired communication linkage between the Orchestration-Hub and the other devices. However, the links between the Pi Cameras, IoT Concentrators and IoT-Cloud Hub are wireless with Wi-Fi. To achieve a high performance, multiple Wi-Fi channels can be dynamically configured by the Orchestration-Hub to steer the VDO traffic (based on TCP) away from the congested channels. Based on the implementation prototype, the average TCP throughputs over all the wireless links have been improved by 56.046% on average, compared with the case without the dynamic Wi-Fi channel selection.

Keywords: Apache Kafka, Raspberry Pi, video streaming, wireless

1. Introduction

Big cities face inevitable congested road traffic conditions. The main reason for this problem is that the roads are built with the limited space and capacity [1]. However, this problem can be solved when the traffic police can control the traffic flow smoothly. To control, a real-time information of the road traffic network condition is required for the traffic police [2]. Big cities are facing heavy road traffic congestion. The main reason is the increasing number of vehicles beyond the road network capacity [1], which forces traffic police at times to be incapable of controlling the inevitable traffic jams. To resolve those traffic jams, traffic police need the near real-time information of the road network [2].

Conventionally, the road network condition relies on the inductive loop coil detectors, which count the passing vehicles per lane. However, from the viewpoint of traffic police, such information shown in numbers or graphs is not insightful enough. Preferably, to enhance their ability to manage the vehicle flows, the traffic police want to watch traffic videos from e.g. Closed-Circuit Television (CCTV) system. However, the conventional traffic CCTV system is highly costly to deploy at a large scale. For instance, the system needs to use wired links for the communication from the camera location to the nearest control points. Due to such

+ Corresponding author. Tel.: + 66 949 313 677.
E-mail address: [1+]aung.my@student.chula.ac.th, [2]jskim@nm.gist.ac.kr, [2]jongwon@smartx.kr, [1]chaodit.a@chula.ac.th.

costs, this traffic CCTV system, despite of its potential benefits, would be adoptable for only the developed or well-economic cities.

In this paper, a prototype framework of low-latency collection of multi-point video streams over a multi-hop wireless network is proposed. Enabling the system with wireless linkages is expectedly reducing the overall operational and building costs. Further, we design each camera with a cheap electronic board, e.g. Raspberry Pi with solid-state cameras. The video frames are sent from the Raspberry Pi's to IoT Concentrators in every 0.2 sec to mimic the near-real-time video playback while not overloading the wireless link capacity. Video Frames are played back at the IoT-Cloud Hub for the users. Since the system has been designed with the scalability target, for multi-point video streaming, the accumulated video data can be large. To deal with that scalability concern, in this paper, a big data framework, namely Apache Kafka [3] is used for all the message (or video frames) streaming.

Streaming is the sending and receiving process of data in a continuous flow [4]. Apache Kafka (hereinafter called Kafka) [3] is one of the big data streaming platforms. Kafka is a publish-subscribe messaging system where data is sent as Kafka messages to a Kafka topic at a Kafka broker. Any kind of data can be sent with Kafka with the data formatted to the allowable data sizes. Too large data must be divided into serval small Kafka messages. Apache Zookeeper [5] (hereinafter called Zookeeper) is a centralized service for providing distributed synchronization and grouping service. In this work, Zookeeper is used for managing the Kafka Brokers in the IoT Concentrators and Zookeeper can detect the addition and removal of Kafka Brokers. It also manages the Kafka topics in the Kafka Brokers. Although Kafka has been built originally intentionally for the ordered-logged messaging system, often in texts. Recently, researchers have tried also to apply Kafka for the video streaming system [6, 7] with the video data that must be modified efficiently.

In this paper, each Raspberry Pi's with a camera serves as a Kafka producer, which publishes video frames as Kafka messages to the available two IoT Concentrators, which run the leader and follower Apache Kafka Brokers, respectively. Video playback application as Kafka consumer residing at the IoT-Cloud Hub can read the video frames by consuming Kafka messages from the IoT Concentrators. To achieve a high performance, multiple Wi-Fi channels per each wireless link have been here proposed to be adjustable dynamically by the Orchestration-Hub. The purpose is to steer the VDO traffics (based on TCP) away from the congested channels. This design framework has been proved by a small in-laboratory scaled implementation prototype and the testbed measurement results are herein reported.

2. Proposed Framework of Low-Latency Collection of Multi-Point Video Stream over Multi-Hop Multi-Channel Wireless Network

Unlink the earlier work [6], which uses the Industrial, Scientific and Medical (ISM) 5-GHz band, in this work, the ISM 2.4-GHz band has been selected to reflect the main capability of switching dynamically those channels upon their emerging congestion conditions. Video frames are captured at the Pi Camera and stored temporarily at the local storage of Pi Camera before sending to the IoT Concentrator 1 and 2 when the links between the Pi Camera and IoT Concentrators become full or subject to too much signal interferences with the other Wi-Fi connections. As a result, the channel can be blocking potentially to Kafka messages. To avoid the much-interfered channels, the Orchestration-Hub can reassign the Wi-Fi channels between the Pi Camera and IoT Concentrators. For simplicity, the Orchestration-Hub is assumed to have the available wired communication linkage between the Orchestration-Hub and the other devices.

In this work, Raspberry Pi Camera serves as the Kafka Producer. This is a cost-effective hardware because it has enough computing power to run all the necessary software modules for capturing, storing temporarily and sending to the IoT Concentrators. Kafka broker is installed at the IoT Concentrator and Raspberry Pi is used as an IoT Concentrator. An Intel® NUC is used at the IoT-Cloud Hub which is serving as the central traffic video repository as well as running the video playback applications as Kafka consumers. Another Intel® NUC is used as the Orchestration-Hub. The Kafka Zookeeper is installed at the Orchestration-Hub for controlling topics inside the Kafka brokers and also Kafka brokers inside the IoT Concentrators.

Our prototype is designed with four Raspberry Pi's and two Intel® NUCs with the detailed specification summarised in Table I. The proposed prototype framework is shown in Figs. 1 and 2. In streaming data from Pi Cameras to IoT Concentrators 1 and 2, Wi-Fi Channel a and Channel b are used. From IoT Concentrators to the IoT-Cloud Hub, Channel x and Channel y are used. These channels a, b, x and y are selected from the 2.4-GHz Wi-Fi channels 1 to 11. To enable multiple channels between a node pair, two Wi-Fi interfaces are installed here at each Pi Camera and each IoT Concentrator as well as at the IoT-Cloud Hub.

Table I: Prototype Device Specification.

Device	Quantity	Aim
Intel® NUC5i5MYHE	2	1 for IoT-Cloud Hub and 1 for Orchestration-Hub
Raspberry Pi 3 model B	4	2 for Pi Cameras and 2 for IoT Concentrators
Wi-Fi USB dongle with omnidirectional antenna	10	External dongles for Intel® NUC5i5MYHEs and Raspberry Pi's

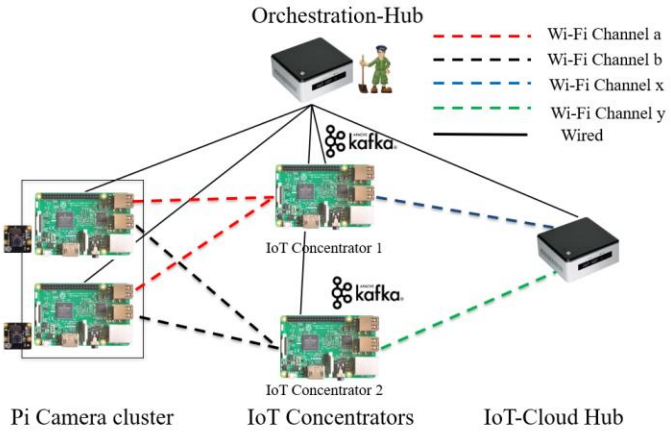

Fig. 1: Proposed prototype framework for video streaming.

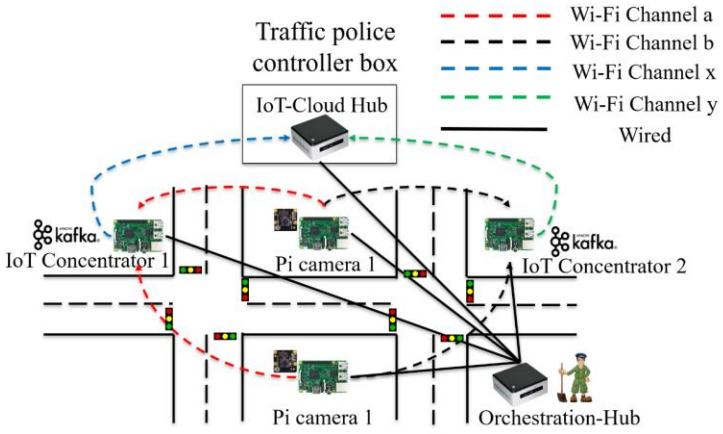

Fig. 2: Proposed prototype framework for video streaming on the road traffic network.

3. Preliminary Proof-of-Concept Evaluation

In the preliminary evaluation, a small testbed is established by using one Pi Camera, two IoT Concentrators, one IoT-Cloud Hub and one Orchestration-Hub as shown in Figs. 3 and 4.

In the test, Pi Camera captures 1080 × 720 resolution video frames and stores the images or video frames at the local storage. One video frame is averaged of 40 Kbytes in size. As the local storage of the Raspberry Pi, has the limited storage space, 16-GByte Secure Digital (SD) card is used as the local storage device of the Raspberry Pi. Video frames are removed 5 sec after their creation. The video frames are converted to Kafka messages and sent to the IoT Concentrator 1 and 2 with ISM 2.4 GHz Wi-Fi channels a and b, respectively. At the beginning of the test, channels a and b are assigned to Wi-Fi channels 3 and 8, respectively. Two

leaders (and follower) Kafka brokers are run at the IoT Concentrators 1 and 2, respectively. IoT-Cloud Hub pulls Kafka messages from IoT Concentrators 1 and 2 with Wi-Fi channels x and y, assigned to Wi-Fi channels 11 and 12 at the beginning of the test. Kafka messages are converted to video frames and show on the screen for the users as shown in Fig. 5. Video frames are stored at the local storage of the IoT-Cloud Hub.

While streaming the video, Wi-Fi Channels are configured to be highly subject to additionally inserted other Wi-Fi interfering sources. As a result, from Pi Camera to the IoT Concentrators 1 and 2, the originally assigned Wi-Fi channels have been subsequently changed dynamically by our system to less interfering Wi-Fi channels 11 and 2, respectively. Likewise, the Wi-Fi channels from IoT Concentrator 1 and 2 to IoT-Cloud Hub are also changed to Wi-Fi channels 6 and 5, respectively. Before and after such Wi-Fi changing instance, Iperf [8] is run to test the average TCP throughput of the link between the Pi Camera, IoT Concentrators and IoT-Cloud Hub. Comparison of average TCP throughput between before channel changing and after channel changing is shown in Table II.

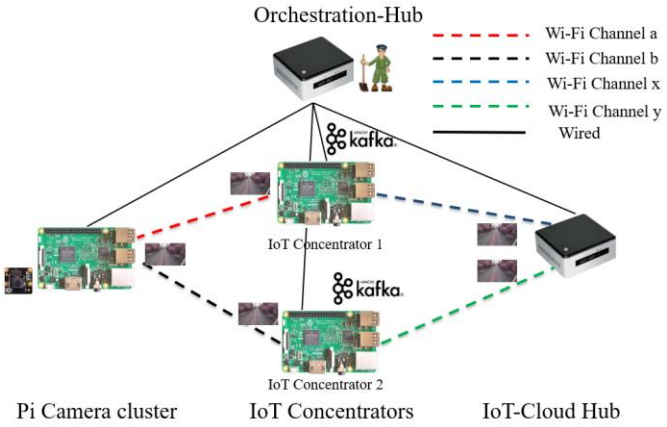

Fig. 3: Preliminary proof-of-concept evaluation testbed for video streaming.

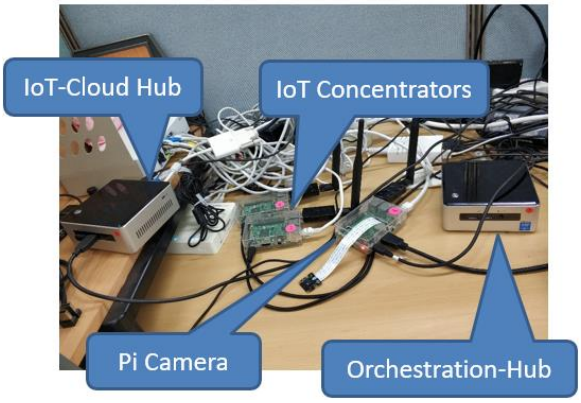

Fig. 4: Preliminary proof-of-concept evaluation tested at the laboratory.

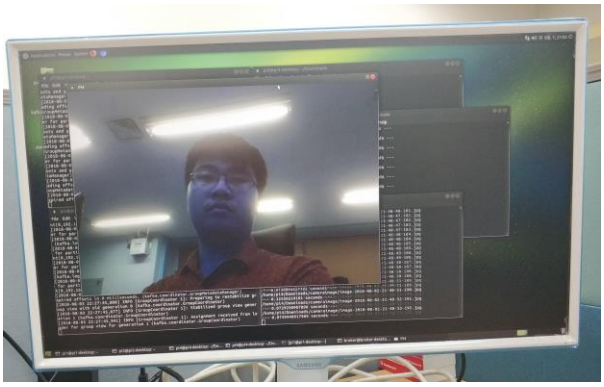

Fig. 5: Snapshot example during testing for preliminary proof-of-concept evaluation.

Table II: Average TCP throughput before channel changing and after channel changing

From	To	Average TCP Throughput (Before channel changing)	Average TCP Throughput (After channel changing)
Pi Camera	IoT Concentrator 1	Channel 3	Channel 1
		8.70Mbit/s	12.65Mbit/s
Pi Camera	IoT Concentrator 2	Channel 8	Channel 10
		7.25 Mbit/s	16.88 Mbit/s
IoT Concentrator 1	IoT-Cloud Hub	Channel 11	Channel 6
		9.54 Mbit/s	15.30 Mbit/s
IoT Concentrator 2	IoT-Cloud Hub	Channel 2	Channel 5
		18.86 Mbit/s	16.14 Mbit/s

From these results, the throughput of the links is improved after Wi-Fi channel changing expectedly Average video streaming latency depends on the average TCP throughput as Kafka uses TCP for streaming. After channel changing, the throughput is significantly increased and, therefore, the average latency is also decreased. This should help improve the overall quality of video streaming service for our intended application.

4. Conclusion and Future Work

In this paper, a prototype framework of low-latency collection of multi-point video stream over multi-hop multi-channel wireless network is designed for road traffic monitoring system with the enabler technology of Apache Kafka. To have a high performance for wireless linkage, an Orchestration-Hub is introduced to control the Wi-Fi channel selection between Pi Camera, IoT Concentrators and IoT-Cloud Hub. A preliminary evaluation is tested and reported here on the proof-of-concept evidence of this design framework. It is confirmed that, with the proposed scheme, the obtainable average TCP throughput and consequently latency can be greatly improved. This is our ongoing research work. Currently, built on our early test successes, a larger scale of system prototype will be implemented with more Pi Cameras and also IoT Concentrators and future results will be further reported.

5. Acknowledgements

The first author gratefully acknowledges to Prof. JongWon Kim and lab mates at Networked Computing System (NetCS) Laboratory, School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, and Assoc. Prof. Chaodit Aswakul and colleagues at the Wireless Network and Future Internet Research Unit, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University for their kind assistance and supports throughout this research.

6. References

[1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," Proceedings of the IEEE, vol. 91, no. 12, pp. 2043-2067, 2003

[2] K. Nellore and G. P. Hancke, "A survey on urban traffic management system using wireless sensor networks," Sensors, vol. 16, no. 2, pp. 157-177, 2016.

[3] G. Wang et al., "Building a replicated logging system with Apache Kafka," Proceedings of the VLDB Endowment, vol. 8, no. 12, pp. 1654-1655, 2015.

[4] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the Internet: approaches and directions," IEEE Transactions on circuits and systems for video technology, vol. 11, no. 3, pp. 282-300, 2001.

[5] “Apache Zookeeper.” [Online]. Available: http://zookeeper.apache.org/

[6] A. M. Htut, S. Y. Htet, K. Leevangtou, K. Kawila, and C. Aswakul, “Testbed Design of Near Real-time Wireless Image Streaming with Apache Kafka for Road Traffic Monitoring,” in 33rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC) 2018, pp. 188-191, 2018.

[7] X. Guo, Y. Cao, and J. Tao, "SVIS: large scale video data ingestion into big data platform," in International Conference on Database Systems for Advanced Applications, pp. 300-306, 2015: Springer.

[8] “Iperf.” [Online]. Available: https://iperf.fr/