Edge-assisted Content and Computation-Driven Dynamic Network Selection for Real-Time Services in the Urban IoT

Sabur Baidya and Marco Levorato
The Donald Bren School of Information and Computer Science, UC Irvine, CA, US
e-mail: {sbaidya, levorato}@uci.edu

***Abstract*—Supporting city-wide exchange of information in Urban Internet of Things (IoT) systems using existing communication infrastructures is extremely challenging especially when traditional services operate in the same network resource. Additionally, the most advanced Urban IoT services focus on real-time data processing, which shifts the perspective and goal of the network when transporting data. In this paper, the notion of Quality of Computing (QoC) is introduced to capture the level of support the communication infrastructure provides to this family of computation applications. In this context, we propose a dynamic network selection mechanism based on Software Defined Networks (SDN) designed to provide QoC in Urban IoT scenarios where the heterogeneous network resources are shared. The proposed mechanism dynamically assigns portions of data from IoT streams over licensed and unlicensed bands to guarantee QoC while minimizing cost of operations and licensed band occupation. Instrumental to our technique is the recently proposed edge-computing architecture, where computational resources placed at the edge of wireless access networks enable the interconnection of network management to processing. We consider a real-time monitoring scenario, where sensors transmit a video stream which is processed to identify and classify objects. The supporting wireless infrastructure consists of WiFi that operates in unlicensed frequency bands and cellular communication technology, Long Term Evolution (LTE) operating in licensed bands. We demonstrate the performance by means of real-world experiments on a testbed with WiFi and LTE networks built with hostapd and OpenAirInterface.**

I. Introduction

In Urban Internet of Things (IoT) systems [1] a large number of devices interconnect and interoperate to provide services to the citizens. Supporting this city-wide exchange of information using existing communication infrastructures is extremely challenging especially when IoT services coexist on the same network resource with traditional services, such as data and voice over cellular networks. Additionally, many relevant Urban IoT services, such as transportation and vehicular networks, are computation-driven, where data are processed in a, possibly multiscale, architecture to provide feedback and control signals in real-time. The Quality of Service (QoS) constraints imposed by such services, besides being often more stringent than those where human users consume data, are specific to the computational-task and data structure.

To address this scenario, we introduce the notion of Quality of Computing (QoC), where the performance of the network is measured in terms of quality of the output of algorithms processing data. In this paper, we propose a dynamic network selection mechanism based on Software Defined Networks (SDN) [2] designed to provide QoC in Urban IoT scenarios, where IoT data streams coexist with other services on the same network resource. The proposed mechanism dynamically assigns portions of the IoT stream to available licensed and unlicensed network resources to guarantee QoC while minimizing cost of operations and licensed band occupation. Instrumental to our technique is the recently proposed edge-computing architecture [3], where computational resources placed at the edge of wireless access networks enable the interconnection of network management to processing. Thus, data processing at the edge processor informs network management, which in the considered case takes the form of network selection for data stream portions with different relevance to the computational goal.

The framework and technique proposed in this paper find wide applicability. Herein, we consider a real-time monitoring scenario, where sensors transmit a video stream which is analyzed at the edge processor to identify and classify objects. The supporting wireless infrastructure consists of a WiFi network, which operates in unlicensed frequency bands, and a cellular communication technology, Long Term Evolution (LTE) in our case-study, which operates in licensed bands. In our setup, uplink licensed LTE provides high QoC, where packets experience a "near interference-free" environment which comes at the cost of utilizing bandwidth that could be allocated to competing licensed services. Conversely, transmission over WiFi is assumed cost-free, but is subject to interference from other data streams, which may cause temporally localized congestion. In the proposed framework, the edge-assisted SDN architecture dynamically selects the network resource to be used by the data stream, or specific portions of it, to trade-off licensed bandwidth usage and QoC.

We illustrate the performance of the proposed framework via experiments conducted on a real-world testbed emulating WiFi and LTE networks using hostapd [4] and OpenAirInterface [5,6]. In the considered setup, QoC at the edge processor is periodically affected by bursts of traffic on the same channel. The SDN-based system we propose mitigates this effect by reacting to congestion and opportunistically moving parts of the data stream to the LTE network. To this aim, we leverage the specifics of spatial and temporal video encoding to separate the video stream into sub-streams with different relevance to the recovery operated at the edge processor. Numerical results show that the dynamic selection technique improves QoC while minimizing LTE usage.

The rest of the paper is organized as follows. Section II discusses related work proposing network selection and multi-network frameworks. In Section III, we describe the real-time case-study scenario considered herein, and illustrate the impact of interference on the output of the processing algorithm. Section IV presents the network architecture and software defined system and the edge-assisted content and computation-aware network selection algorithm at the core of the proposed framework. Section V presents and discusses the experimental setup and the numerical results and evaluations. Section VI concludes the paper.

II. Related Work

Recently, network selection techniques aiming at improving throughput and bandwidth utilization have attracted considerable attention. Stream Control Transmission Protocol (SCTP) [7], Concurrent Multipath Transfer (CMT) [8] and other analogous protocols have been proposed to select alternate network paths to improve reliability. Most of these innovative architectures are based on connection-oriented and stateful protocols, which often do not match the needs of real-time services. The Multipath TCP (MPTCP) protocol [9,10] proposes to use multiple paths available in the network to build a single connection session, where packets are distributed across the paths proportionately to their respective quality. However, similar to TCP, the protocol prescribes retransmission and even an individual slow path may cause issues such as buffer overflow and delayed packet reordering at the receiver. This limits its applicability in time-sensitive applications. Network selection in multi-radio networks for IoT environments is proposed in [11] and [12]. However, the techniques proposed in these papers are based on traditional throughput requirements and involve core network components with unpredictable propagation delays. Different from these previous contributions, we propose a framework where network selection is aware of the content transported by the stream and the global computation task. To this aim, our framework is edge-assisted, thus creating an innovative connection between content and network management, and avoiding the, possibly large and unpredictable, delay introduced by cloud- and core-network based architectures. Additionally, since network selection is controlled by the edge processor and local network managers, our technique minimizes impact on coexisting services. Specific content-based Interference control can be added to the proposed framework using approaches analogous to those in [13,14].

III. Case-Study Scenario

Since processing architectures based on cloud-computing, such as [15], often cannot meet the stringent delay constraints imposed by real-time urban IoT services, herein we assume that data processing occurs at the wireless network edge. In particular, we assume that sufficient computational power, i.e., an edge processor, is available at one, wireless, hop from the sensors acquiring the data. We focus our study on a scenario where video input data are streamed by sensors over the wireless network to the edge processor, whose objective is to detect and classify objects within the captured area.

We assume that the overall bandwidth of the network infrastructure, which is composed of multiple individual networks and technologies, is sufficient to support the data stream. We focus on WiFi [16] and LTE [17] networks, which are widely available in urban environments, supported by most devices and predominant technologies for mid and long-range communications. The two networks have different characteristics. The LTE network has an access cost, which conceptually could even correspond only to the reduction in the bandwidth available to other licensed users, but operates under Quality of Service (QoS) guarantees with dedicated uplink channel. The WiFi network, conversely, operates in the unlicensed band and does not incur an "operational cost", but is subject to congestion and interference from other, possibly bursty, data streams.

The video streaming application encodes the data using H.264 encoder [18] and then transmits using Real Time Protocol (RTP) [19]. The sequence of video frames is spatially and temporally compressed generating Group of Pictures (GoP), where each GoP contains a reference frame (full picture) followed by a series of differential frames, which transport differences with respect to the reference frame. Conversely, damage to a differential frame causes a quickly recovered corruption [13,14], which is also mitigated by temporal interpolation. The encoded video frames are further divided into transport stream (TS) packets and encapsulated into the payload of data packets for video streaming.

A. Quality of Computation Metrics

Metrics such as packet latency and jitter have been traditionally used to measure the performance of video streaming for real time services. However, packet-based metrics which average over all packets may not fully capture the performance of the data stream, especially in relation to the output of the processing algorithm which is heavily dependent on spatially and temporally local properties of the received video. For instance, due to spatial and temporal encoding, if packet loss occurs within a reference frame, the resulting corruption spreads over the entire GoP creating persistent artifacts which may be erroneously detected as objects. Thus, the impact of packet loss, either due to packet delivery failure or excessive delay, is not uniform over the video stream.

A metric which is widely used to summarize video quality is Peak Signal to Noise Ratio (PSNR) [20], which measures the average Mean Squared Error (MSE) over all the video frames using as reference the uncorrupted video, that is:

$$\mathrm{PSNR} = 10 \log 10 (K_{\mathrm{bps}} / \mathrm{MSE}_{\mathrm{avg}}), \tag{1}$$

where K_{bps} is 2^B, B is the number of bits per pixel, and MSE_{avg} is the average MSE over all the frames of the video. However, two videos with same PSNR can have

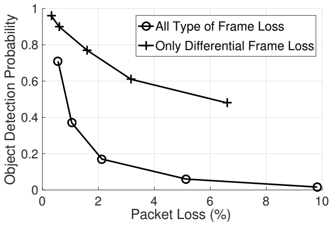

Figure 1: Object Detection to Packet Loss Mapping

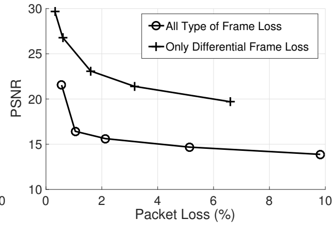

Figure 2: PSNR vs Packet Loss.

entirely different characteristics, especially for the purpose of computation.

Herein, we use a metric of quality which is directly dependent on computation. More specifically, we directly use correct object detection rate as a metric to measure the quality provided by specific packet loss patterns and guide network selection. Then, we define

$$P_{obj\_det} = \frac{N_{rcvd}}{N_{ref}} \tag{2}$$

where N_{rcvd} and N_{ref} are the total number of detected local features, corners and edge pixels of all matching surface objects in the received and reference video, respectively, such that $\{N_{rcvd}, N_{ref}\} \in N$ where N is the total number of local features of surface objects detected in the uncorrupted video.

B. Preliminary Study

In order to understand the impact of packet loss within the video stream on the quality of the computation output, we perform a numerical study on real-world video. Specifically, we generate synthetic packet loss patterns and measure the quality of the decoded corrupted video in terms of object detection. We first test “content-agnostic” packet loss patterns, where packets are dropped with uniform probability over the stream. Then, we test a “content-aware” pattern, where reference frames are protected and packets are dropped only in differential frames. In our experiments, packet loss is distributed according to an exponential distribution. Figure 1 depicts object detection probability as a function of packet loss for these two cases. It can be observed that concentrating packet drops in differential frame considerably improves object detection for a given percentage of packets drops. For instance, an object detection probability of 0.6 corresponds to 1% packet loss in the content-agnostic case and 4% packet loss in the content-aware case. Thus, content-aware packet loss patterns can tolerate 4 times the packet loss achieving the same QoC. Analogous observation can be made when measuring PSNR, see Fig. 2. We build our network selection framework on these observations, where the edge processor evaluates patterns of packet drops and informs network selection. Specifically, the edge processor matches recent packet loss patterns to QoC, and generates feedback used by the sensors to determine the network used to transport different portions of the video.

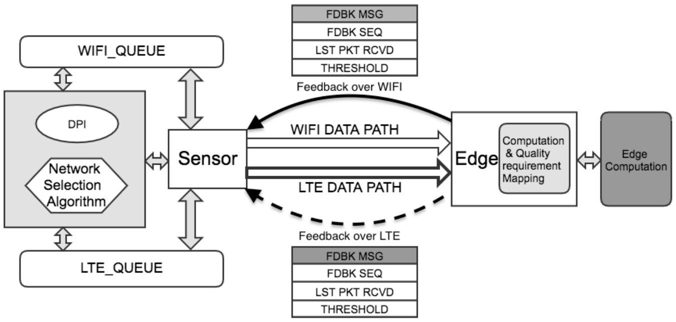

Figure 3: Edge Assisted urban IoT Architecture

IV. Architecture

The considered edge-assisted urban IoT architecture is composed of an edge processor connected to the WiFi and LTE base stations via a high-capacity link, and a set of sensors connected wirelessly. Figure 5 shows the components of the architecture and the flow of information. Sensors have two queues to maintain the state of packet flows on WiFi and LTE, where one of the data paths can be used at any point of time for upstreaming data to the edge processor. The “state” of the data path associated with each individual network is based on feedback generated from the edge processor, which evaluates the pattern of received packets in relation to the specific computational goal. The output of the evaluation is sent to the sensors using reverse paths over all available networks in the form of feedback messages. This to avoid long delays and possible timeouts in case one of the two paths is currently congested, and improve reaction time of the system in selecting different networks in response to paths’ state. Feedback messages are also used to probe paths, which is mostly effective when up and downstream links use the same channel resource. However, the consistency of feedback-based probing with the real state of the network is limited by the small size of feedback messages.

The edge processor connected to the base stations performs real-time computation and keeps track of recent QoC and packet loss patterns. Based on pre-computed mapping and the QoC requirement, the edge processor determines the packet loss threshold to be included in feedback messages. The sensor performs Deep Packet Inspection (DPI) to determine the content of packets, which informs network selection in response to the feedback and network state.

We make the architecture described above specific to the real-time video monitoring scenario at the center of our study. In this context, content differentiation corresponds to different types of video frames, as illustrated in the preliminary results discussed in the previous section. Once the sensor receives the tolerable packet loss limit, it determines the frame type of the packet to be transmitted and tunes the threshold for offloading the traffic to the LTE or WiFi network accordingly. In order to select the network, the sensor first measures the long term average packet reception quality at the edge from the feedback messages sent from edge. The edge sends application sequence number of the last packet received in every feedback to the sensor. Upon reception of every feedback message, the sensor updates the state queues for the WiFi and LTE accordingly. Then before selecting the network, the sensor reads the most recently updated queue and check if the corresponding network provides sufficient QoS to packets to match the threshold requirement. If so, it selects the current network otherwise, it offloads the traffic opportunistically.

The network selection algorithm also considers the cost of operation. To this aim, once data are offloaded to the LTE network, the protocol will try to re-route the traffic trough WiFi whenever a pattern is detected that can support the QoC requirement. Since the WiFi channel is bidirectional, performance measured using feedback roughly correspond to those of the upstream data. Thus, the protocol extracts useful information from the feedback, where good performance of the feedback channel is used to trigger a re-assignment of traffic to WiFi. Then, the protocol compares the sequence number of feedback message over WiFi with that that over LTE and, if they are comparable for a time window, it presumes the WiFi channel quality returned to a good state.

The basic functions of the network selection algorithm are summarized in Algorithm 1. We study two variants of the protocol, namely content-agnostic and content-aware protocols. In the former case, the system treats all the packets in same way, i.e. uses a common packet loss threshold for network selection. In the latter case, the threshold value will be applied to the non-reference packets only, whereas the packets from reference frames will be transmitted using the LTE network, which corresponds to zero packet loss tolerance.

Algorithm 1: Content & Computation aware Network Selection Protocol

```
Function CollectFeedback()
Data: Collect Feedback control messages from Edge:

update QUEUE_lte
update QUEUE_wifi
if FDBK_lte > FDBK_wifi then
|   select QUEUE_lte
else
|   select QUEUE_wifi
end

Function DeepPacketInspection()
if Packet_type ϵ ReferenceFrame then
|   packet_type = 1
else
|   packet_type = 0
end

Function NetSelect()
Data: Read queue Q from CollectFeedback()
Call DeepPacketInspection()
if packet_type == 1 then
|       /* Edge assigned threshold for I frame for QoC
|   requirement */
|   Loss_thresh = T1
else
|       /* Threshold for non-reference frame */
|   Loss_thresh = T2
end
Data: Q[i]: Packet number received at edge in ith slot

if |Q[start] − Q[end]| − Q_size < Loss_thresh then
|   if Q == QUEUE_wifi then
|   |       /* back to wifi if it is good */
|   |   set Network = WIFI
|   else
|   |       /* Keep the same network */
|   end
else
|   set Network = LTE
end
```

V. Experiments & Numerical Results

We implemented the network selection algorithm on a fully functional testbed, where we use hostapd [4] to create software access point (AP) for WiFi and OpenAirInterface [5] as open-source LTE emulator. We use a real-world monitoring video of 641 frames, which are encoded with H.264 and converted to a TS packet stream using ffmpeg [21] codec. The TS packets are, then, encapsulated within UDP packets and transmitted over the network to the edge processor. To illustrate the general performance of object detection, we use Speeded Up Robust Feature (SURF) based object detection [22] as a measure of computation quality. The received video packets are first decoded using ffmpeg decoder and then fed into the object detection process.

A. Testbed setup

For the testbed emulation, we used laptop computers as sensor and edge machines. These machines run ubuntu 14.4 operating system with Linux kernel version 3.19. The WiFi software AP and also the LTE base station runs on the same machine as the edge processor to simulate a high capacity link. We configured the software AP using open-source hostapd software with configurations conforming with IEEE 802.11 standard. For LTE, we used open-source based OpenAirInterface (OAI) module installed on the machines that connect to radio interfaces USRP B210 with omnidirectional antennas separately for uplink and downlink. The sensor machine runs the code for LTE UE and the edge machine runs the LTE eNodeB code. We used licensed band for our experiment within allowable duration for academic research. As we consider edge computing, we do not use the EPC core network for LTE and only use the evolved universal terrestrial radio access network (EUTRAN) [23].

B. Numerical Results

We performed the experiment with our testbed in a scenario where the WiFi network is affected by bursts of traffic competing with the video stream. We used controlled UDP flooding with the iperf [24] tool to produce the congestion in the WiFi channel. We generated the external traffic over WiFi on the same channel with packet bursts whose arrival is distributed according to an exponential distribution of fixed rate, and the burst size is 3 seconds. Bursts congest the network and result in localized sequences of packet drops.

From our preliminary study, we built a map between the QoC level and the packet loss rate based on which the edge processor sets the maximum packet loss threshold. In the high interference regime we used in our testing, the WiFi channel drops about 25% packets on average due to interference.

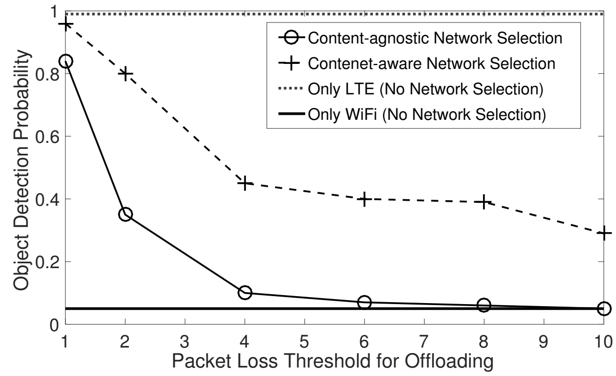

Figure 4: Object detection probability as a function of packet loss threshold

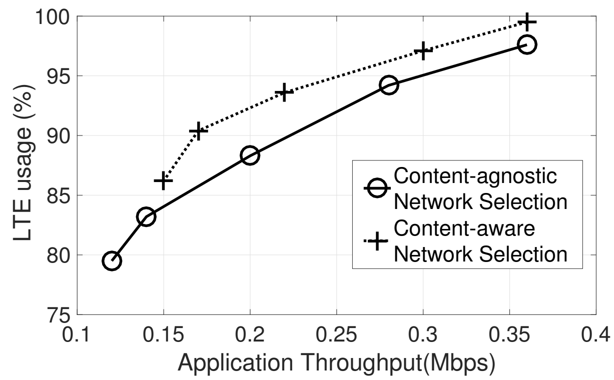

Figure 6: LTE usage (network operation cost) as a function of application throughput.

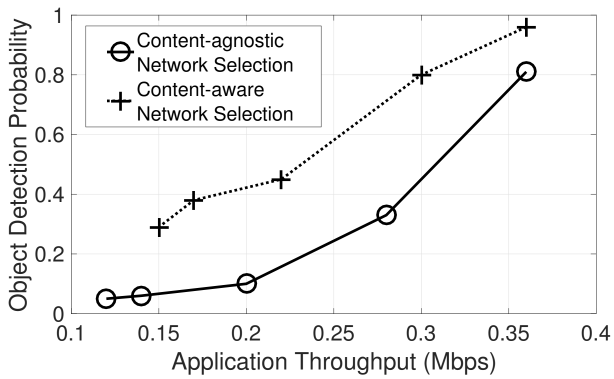

Figure 5: Object detection probability as a function of application throughput

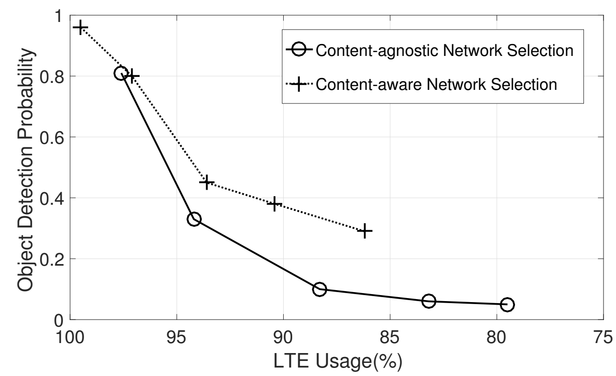

Figure 7: Object Detection Probability as a function of network Operation Cost measured in terms of LTE usage.

Hence, in order to reduce the packet loss and fulfill the QoC requirements, the network selection may need to re-direct part of the stream to LTE in periods where the WiFi is affected by congestion. The frequency of offloading traffic to LTE depends on the threshold value set by the edge processor, which corresponds to the QoC requirement.

We performed the experiments by varying the QoC threshold and measured the object detection and throughput both for content-aware and content agnostic algorithms. Fig. 4 shows that when WiFi is used, in the considered interference regime object detection probability is close to zero due to almost one fourth of packet lost due to congestion. It can be observed that if the whole stream is assigned to LTE, then object detection probability is close to 1. When network selection algorithm is employed, the lower the QoC threshold, i.e., lower packet loss tolerance, the higher the object detection probability. The figure also shows that when content-aware network selection is employed, i.e. always using LTE to transfer reference frame and WiFi to LTE offloading for other packets when WiFi has average packet loss smaller than the threshold, then the overall performance of the system improves with respect to the content-agnostic network selection. However, the content-aware network selection may result in larger usage of LTE resources, and requires DPI.

In addition to QoC metrics, we measure the output throughput and object detection for both content-aware and content-agnostic selection. Fig. 5 shows that as the throughput increases, object detection probability increases as expected. It also shows that content-based network selection leads to greater improvement in terms of object detection with respect to throughput improvement compared to the content-agnostic case. We then measure the cost of operation in terms of LTE usage for the same packet loss threshold range, and plot it against the application throughput in Fig. 6. It can be observed that throughput improvement comes at the price of an increased cost of network operation. In the considered setup, the cost is larger for content-based network selection than in the content-agnostic network selection mechanism.

Fig. 7 depicts object detection probability as a function of network operation cost in terms of LTE usage. It can be observed as the network cost decreases, object detection probability also decreases in both content-based and content-agnostic cases. It can also be observed that when the cost is very high ($> 90\%$), i.e., a small packet loss tolerance is used to trigger an offload to LTE, the object detection probability is high and comparable in the two cases. However, as the packet loss threshold is small, for a given cost of operation the content-aware network selection obtains larger object detection probability compared to the content-agnostic case.

Finally, we measure the average application latency and

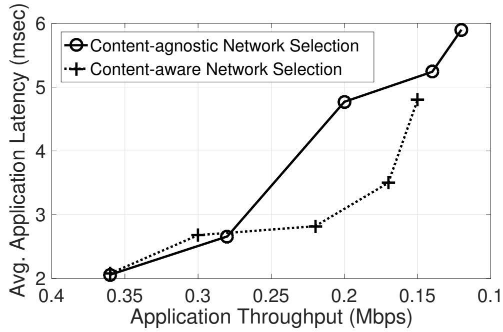

Figure 8: Application latency as a function of application throughput

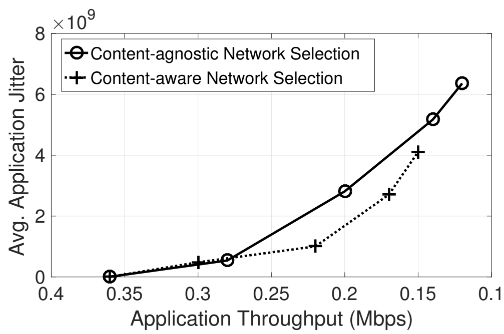

Figure 9: Application jitter as a function of application throughput

jitter. Fig. 8 shows that the average latency increases sharply when the throughput decreases, i.e. when the packet loss threshold is large. It also shows that when the achievable throughput is close to its maximum – that is, 0.35 Mbps – content-aware and content-agnostic selection result in similar latency. Thus, even if suffering a decrease in throughput, the content-aware protocol suffers a smaller penalty in terms of latency. Similar considerations can be made for application jitter as shown in Fig. 9.

VI. Conclusions

The primary contribution of this paper is a novel edge-assisted network selection mechanism for real-time services in urban IoT. We introduce the notion of Quality of Computing, where the edge processor, and the computational task, inform network selection. The network selection protocol differentiates control decisions based on content to further improve performance over cost of network operation given QoC requirements. An experimental evaluation is performed by transmitting a real surveillance video on a testbed built with open-source based WiFi and LTE networks. Numerical results show that in periods where the WiFi network suffers congestion and is unable to provide the desired QoC, the data can be effectively offloaded to LTE in real time to support computation.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[3] E. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the internet of things," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3816–3825, Oct 2013.

[4] J. Malinen *et al.*, "hostapd: Ieee 802.11 ap, ieee 802.1 x," WPA/WPA2/EAP/RADIUS Authenticator. online: http://hostap. epitest. fi/hostapd, Tech. Rep., 2014.

[5] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "Openairinterface: A flexible platform for 5g research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.

[6] N. Nikaein, R. Knopp, F. Kaltenberger, L. Gauthier, C. Bonnet, D. Nussbaum, and R. Ghaddab, "Demo: Openairinterface: an open lte network in a pc," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 305–308.

[7] R. R. Stewart and Q. Xie, "Stream control transmission protocol (sctp)," 2001.

[8] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using sctp multihoming over independent end-to-end paths," *IEEE/ACM Transactions on networking*, vol. 14, no. 5, pp. 951–964, 2006.

[9] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp." in *NSDI*, vol. 11, 2011, pp. 8–8.

[10] S. H. Baidya and R. Prakash, "Improving the performance of multipath tcp over heterogeneous paths using slow path adaptation," in *IEEE International Conference on Communications (ICC), 2014*. IEEE, 2014, pp. 3222–3227.

[11] S. Andreev, M. Gerasimenko, O. Galinina, Y. Koucheryavy, N. Himayat, S.-P. Yeh, and S. Talwar, "Intelligent access network selection in converged multi-radio heterogeneous networks," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 86–96, 2014.

[12] M. Gerasimenko, N. Himayat, S.-p. Yeh, S. Talwar, S. Andreev, and Y. Koucheryavy, "Characterizing performance of load-aware network selection in multi-radio (wifi/lte) heterogeneous networks," in *Globecom Workshops (GC Wkshps), 2013 IEEE*. IEEE, 2013, pp. 397–402.

[13] S. Baidya and M. Levorato, "Content-based Interference Management for Video Transmission in D2D Communications Underlaying LTE," *IEEE International Conference on Computing, Networking and Communication (ICNC) 2017*, 2017.

[14] S. Baidya and M. Levorato, "Content-based Cognitive Interference Control for City Monitoring Applications in the Urban IoT," *IEEE Global Communications Conference, GLOBECOM 2016, Washington DC, USA, December 4-8*, 2016.

[15] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "On the integration of cloud computing and internet of things," in *International Conference on Future Internet of Things and Cloud (FiCloud), 2014*. IEEE, 2014, pp. 23–30.

[16] S. Tozlu, M. Senel, W. Mao, and A. Keshavarzian, "Wi-fi enabled sensors for internet of things: A practical approach," *IEEE Communications Magazine*, vol. 50, no. 6, 2012.

[17] B. Furht and S. A. Ahson, *Long Term Evolution: 3GPP LTE radio and cellular technology*. Crc Press, 2016.

[18] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[19] V. Jacobson, R. Frederick, S. Casner, and H. Schulzrinne, "Rtp: A transport protocol for real-time applications," 2003.

[20] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[21] F. Bellard, M. Niedermayer *et al.*, "FFmpeg," *http://ffmpeg. org*, 2012.

[22] H. Baya, A. Essa, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding, Volume 110, Issue 3, June 2008, Pages 346-359*, June 2008.

[23] T. Lunttila, J. Lindholm, K. Pajukoski, E. Tiirola, and A. Toskala, "Eutran uplink performance," in *2nd International Symposium on Wireless Pervasive Computing, 2007. ISWPC'07*. IEEE, 2007.

[24] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf," 2006.