QoE-Based Video Orchestration for 4G Networks

Marcos Carvalho*, Vinicius F. Silva*, Erik de Britto e Silva*, Daniel F. Macedo*, Henrique C. C. de Resende†, Johann M. Marquez-Barja†, Cristiano B. Both‡, Augusto Zanella Bardini§, Juliano Wickboldt§,
*Universidade Federal de Minas Gerais, Brazil | {marcoscarvalho, viniciusfs, erik, damacedo}@dcc.ufmg.br
†University of Antwerp - imec, Belgium | {henrique.carvalhoderesende, johann.marquez-barja}@uantwerpen.be
‡University of Vale do Rio dos Sinos, Brazil | cbboth@unisinos.br
§Federal University of Rio Grande do Sul, Brazil | {azbardini, jwickboldt}@inf.ufrgs.br

***Abstract*—Quality of Experience (QoE) should be the driver for network orchestration in 4G networks. At the same time, the network must be able to cope with high bandwidth requirements from applications such as video streaming, while dealing with a large number of users. This paper proposes a network orchestrator that adjusts network parameters to improve QoE of video streaming. The orchestrator uses Device-to-Device (D2D) communication to improve user's QoE, also reducing the demand on 4G network. The use of D2D is triggered by a machine learning engine. Experiments made in a physical testbed show an improvement on the mean horizontal video resolution from 768 to 1280 pixels, as well as a decrease of around 90% at the impact on the QoE, considering the number of video resolution changes. Finally, the demand on the network backhaul is decreased by around 38%.**

***Index Terms*—Machine Learning, Quality of Experience, Video Streaming, Device-to-Device**

I. Introduction

The 4G cellular networks are lacking on bandwidth to supply all the connected devices [1]. One type of bandwidth-hungry applications is video streaming. This type of application requires a stable network bandwidth to not compromise the user's Quality of Experience (QoE). Some standards such as the Dynamic Adaptive Streaming over HTTP (DASH) [2] have risen trying to fix the unstable and low network bandwidth experienced nowadays. DASH can change the video quality in situations of network instability, enabling the application to keep streaming the video without compromising the users' QoE. However, with the constant increase in the number of mobile subscribers, as well as new applications that demand intensive network data consumption, like online gaming and video sharing [3], there is a great possibility of the video be transmitted with lower resolutions, which can affect the users' QoE. Therefore, different network communication technologies such as Device-to-Device (D2D) must be researched to provide better service quality to the end user.

The D2D Proximity Services 3GPP standard defines as D2D communication the *ad hoc* communication between devices close to each other, with little or no intervention of the base station [4]. This technology enables to unburden the core of the network by moving the data traffic to the edge of the network, a process known as *mobile data offloading* [5]. With the use of D2D, different users requesting the same content can share it among themselves, reducing the need for connections with the same content at the core network. A use case for this scenario is a large group of people in a stadium watching a game replay from their phones. Considering the traditional 4G network, the requests for the same video will compete for bandwidth. However, when D2D is applied, the same video could be transmitted only once till the edge of the network, and then it would be shared between users using the wireless interface for D2D communication. As a consequence, we can offload data traffic from the core network [6] and increase the possible bandwidth to provide a better user experience [7]. Besides lowering the demand at the network backhaul, the D2D connectivity can lower the energy consumption of the network infrastructure [8].

There are several D2D works that help us to understand the pros and cons of the technology [4], [6]–[8]; however, how can D2D networks dynamically enhance video streaming QoE? In this work, we study how an orchestrator can manage D2D groups increasing the QoE of video streaming. The orchestrator was developed with Machine Learning (ML) and the video streaming technology used was DASH. It is important to highlight that this proposal does not require changes to the User Equipment (UE) enabling its implementation in real computer networks. Results on an experimental platform show that, with the orchestrator, the mean horizontal video resolution increased from 768 to 1280 pixels. At the same time, considering the number of resolution changes, there's a decrease of about 90% at the impact on the QoE. Finally, the required backhaul throughput decreased by up to 38%.

The rest of this paper is organized as follows: Section II presents the related work. Section III presents the ML-based orchestrator's overall architecture. Section IV presents the experimental setup, describing the hardware/software used. Section V presents the obtained results with the use of the proposed orchestrator. Finally, Section VI concludes this work.

II. Related Work

Ullah and Hong [9] choose as the share point through D2D the UE with the highest signal-to-noise ratio in relation to the evolved Node B (eNodeB). In this proposal, an UE announces the files of interest and a neighbor UE will do the sharing, even if the sharing UE is still downloading the files. While in the state of the art the improvement of QoS is a consequence of the algorithm, not the end goal, in our proposal the orchestration is directly driven by the QoE. Besides that, the referred work was limited by simulations.

Essaili et al. [10] manage the transmission resources based on the buffer space from the UEs. The authors claim to improve the QoE because of the increase in throughput, since it steers more traffic to UEs with empty buffers. One of the main limitations of the referred work is that throughput improvements are obtained through traffic shaping. Consequently, it does not offload traffic from the 4G network, which is a disadvantage in comparison with the current work that proposes a QoE, ML-based orchestrator with a prediction module to decide by the use of D2D, which does offloading from the 4G network.

Doppler et al. [11] raised the downlink throughput through underlay D2D. The authors propose a D2D channel control between two users only, instead of groups of different sizes. The traffic through 4G of each UE is monitored for possible formation of a D2D channel between two UEs. The implemented system decides to use D2D communication when it detects a throughput gain with D2D in relation to 4G. The proposal still controls the transmission power between D2D transmitters to decrease the interference on the 4G UEs. Such work limits the offloading ratio to 50%, which is a disadvantage in comparison to the present work, since it allows the formation of groups with more than two UEs. Additionally, the referred work does not consider the QoE from the UEs.

Pyattaev et al. [6] calculated the viability of mobile data offloading with Wi-Fi Direct. The formation of D2D groups considers the device's location. The authors show that D2D clients obtain better transmission rates due to the shorter links of Wi-Fi. However, the referred work does not consider the QoE from UEs in order to define the D2D pairs, and the evaluation was done only through simulations.

As can be seen above, some works in the literature aim to improve the efficiency of the cellular spectrum or evaluate specific use cases (e.g. peer-to-peer). However, in most proposals the improvement in QoE is implicit from an improved QoS. Further, the studies are based on theoretical analysis or simulations [4], [12]. In this work the decision is driven by a QoE metric, and in order to provide more realistic results, our solution is evaluated in a physical testbed.

III. ML-Based Orchestrator

This section describes the proposed solution, with the network architecture and the orchestrator software.

A. Network architecture

Figure 1 shows the overall architecture of our solution. The architecture has three main entities: A central, fixed orchestrator with the prediction module for the D2D-based decisions; a 4G network composed by UEs which support D2D; and a video server.

The video server may be positioned in the cloud, being accessible by the 4G network through an optical link, or at the edge, i.e. close to the 4G network, if the optical link does not support the video demand, for example.

Finally, the 4G network monitors signal data from the UEs and sends it to the orchestrator in real time, which inputs it to the prediction module. The D2D decisions are sent by the orchestrator to the eNodeB, which forwards them to the UEs.

B. D2D Orchestrator

The orchestrator algorithm relies on a Machine Learning (ML) predictor, which predicts in real time

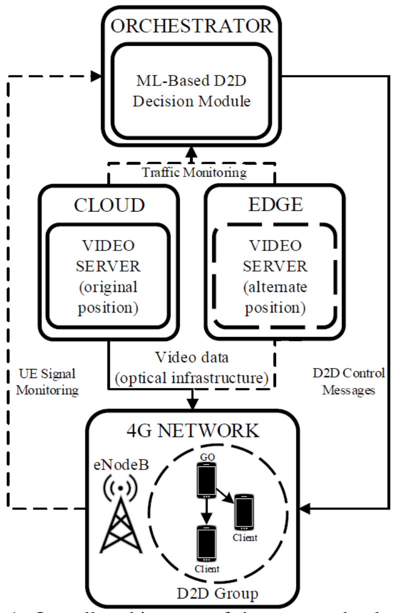

Fig. 1: Overall architecture of the proposed solution.

the horizontal video resolution (the class) for each UE watching a video. The predictor was modeled using Supervised ML. We assume that the horizontal resolution is the QoE goal, because higher resolutions improve the user's perception of the video quality. Since the resolution of the video is given by discrete values, we use a classifier instead of a regressor.

The predictor's input data is collected in the eNodeB (the 4G network features) and the horizontal resolution of the video is logged by the UEs for evaluation purposes. The following data is collected from the eNodeB:

- UE's bit rate (Download/Upload)
- RNTI - Radio Network Temporary Identifier
- CQI - Channel Quality Indicator
- MCS - Modulation and Coding Scheme
- BLER - Block Error Rate
- SNR - Signal-To-Noise Ratio
- PHR - Power Headroom

Algorithm 1 decides when to create a D2D group using the output of the predictor. A group is created if the use of D2D will increase the resolution of the video received by some UE, i.e. if there is an UE watching a video that has a resolution lower than the maximum resolution supported by the server. The chosen Group Owner (GO) is the UE that receives the video with the highest resolution (line 5). Therefore, the interface change (from 4G to D2D) will occur when the UE with the highest horizontal resolution remains stable during a 10 second interval, in order to guarantee a more stable resolution.

It is possible that the devices connected to the D2D network return to the 4G network. This occurs when a group is formed and the expected resolution for one of the client devices is higher in 4G than in the current D2D connection (line 9).

Algorithm 1 D2D orchestration algorithm

1: **procedure** D2DDECISION( )
2: $MaxResolution \leftarrow max(Predictor(X))\ \forall X \in RNTI$
3: $RntiGO \leftarrow \{X \in RNTI \mid X.Predictor(X) = MaxResolution\}$
4: **if** $\{\exists\, X \in RNTI \mid X.resolution < MaxResolution\}$ **then**
5: $createGroupD2D(RntiGO)$
6: **procedure** 4GDECISION(RNTI)
7: $PredictedResolution \leftarrow Preditor(RNTI)$
8: **if** $PredictedResolution > RNTI.CurrentResolution$ **then**
9: $GoBackTo4G(RNTI)$

IV. Experimental Setup

This Section presents the experimental setup, with the implemented network architecture based on Figure 1, and the hardware/software used for the evaluation.

Network Architecture: The orchestrator and the D2D decision module run on a VM located at the UFMG's FUTEBOL[1] testbed. The video server runs in a VM at the UFRGS's FUTEBOL testbed, acting as the *cloud* in this case. On the other hand, the alternate position of the video server (i.e. the edge) is located in a VM at the UFMG's FUTEBOL testbed.

EPC and eNodeB: The 4G LTE Evolved Packet Core (HSS + MME + SP-GW) and the eNodeB run on two separate Docker containers at the UFMG's FUTEBOL testbed. We use Core i7@2.8GHz PCs with 16GB of RAM from the same testbed to emulate the 4G infrastructure, which also have second-generation Ettus USRP software defined radios connected by USB (B200 and B210 models). We used srsLTE[2] to emulate the EPC and the eNodeB.

Video Clients: For the video clients (the UEs), we used five LG Nexus 5X smartphones, which also support D2D communication through Wi-Fi Direct. We developed an Android application to download video from a DASH server, at the same time logging the video horizontal resolution (for evaluation purposes, since the ML predictor does not require data from the UE) and receiving control messages from the 4G infrastructure, which instructs the UEs to switch between the 4G and the D2D interfaces. For simplification, the caching UE is always the Group Owner (GO) of the D2D Wi-Fi Direct group, however, other UEs could also act as cache.

[1] http://www.ict-futebol.org.br
[2] https://github.com/srsLTE/srsLTE

Video Server: For the video server, in UFMG we use PCs with the same hardware setup as the ones used for the EPC and the eNodeB, while in UFRGS we use Core i7@2.4GHz PCs with 4GB of RAM. The video is split in chunks that can be downloaded independently from *nginx*[3]. The server provides seven different horizontal resolutions for the video *Big Buck Bunny*, from 320 to 1920 pixels.

V. Results

A. Training the Predictor

The training instances were obtained from UEs watching videos at the UFMG's FUTEBOL testbed, with the data generated by the eNodeB as the input (see Section III-B), and the logged horizontal resolution from the video, at each UE, as the output.

The collection was done in decreasing rounds from five UEs to one, receiving the video only through the 4G network. Each round had three repetitions. The main goal in this phase was to cover the highest amount of behavioral possibilities from the network.

The training base comprises 450 minutes of video playback and 10348 instances, being 771 for class 320 (the horizontal resolution of 320 pixels), 447 for 480, 1662 for 640, 3795 for 768, 1105 for 1024, 1245 for 1280 and 1323 for class 1920.

The learning procedure was done using Weka[4]. We performed experiments with five classification algorithms. Then, we evaluated the two algorithms with the highest hit rate (number of correct predictions) in *scikit-learn*[5]. For each evaluation we applied *cross-validation*, where 80% of the data were used for training and 20% for tests and validation.

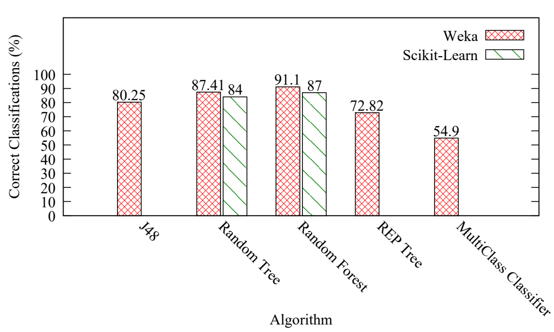

Fig. 2: Evaluation of the classification algorithms.

Figure 2 presents the correct classification rate for each ML algorithm evaluated with *Weka*, as well as the two best algorithms in *scikit-learn*. As can be seen in the Figure, *Random Forest* obtained the highest classification rate in both ML tools. Therefore, this algorithm was chosen for the experimental evaluation.

The model checking after training allows us to explain the behavior of the predictor and the 4G network. Considering the relevance of each feature collected from the 4G network, the inputs SNR, downlink BLER and downlink BRATE obtained the best values, 0.2, 0.15 and 0.17 respectively, while the other inputs obtained values ranging from 0.05 to 0.1.

Table I shows the confusion matrix obtained with *Random Forest*, with the number of samples associated to the correct class. Our model is precise for class 1920, but strives to separate class 768 from classes 640 and 1280. The precision (ratio between samples classified correctly and the total number of samples) and recall (ratio between samples classified correctly and the number of times the class appears in the test set) surpassed 70% and 80% respectively for all classes. For class 1920, the precision was of 98%.

TABLE I: Confusion matrix.

Actual \ Predicted	320	480	640	768	1024	1280	1920
320	130	0	9	28	0	4	0
480	3	68	10	4	5	5	0
640	2	1	256	54	3	1	0
768	9	0	12	699	3	21	0
1024	3	2	11	6	179	12	0
1280	0	0	5	35	16	198	6
1920	0	0	0	0	0	1	273

B. QoE Improvement with the ML-Based Predictor

Our main goal in this phase was to measure the QoE improvement achieved with our orchestrator and the predictor from the training phase. The experiment was run ten times, and considers a confidence interval of 95%. Our setup is composed by five UEs, with the orchestrator instructing them to form D2D groups.

Figure 3 shows the mean horizontal resolution for each UE, with and without the orchestrator. For all the UEs, the scenario with the orchestrator overcame the one without it. For UE 3, the mean horizontal resolution increased from 625 to 1559, an increase of about 150%. This was already expected, since D2D will be used only if it is found an UE with the video at a higher resolution than that of the other UEs. It can be also observed in Figure 3 that the mean horizontal resolution was the same for all the UEs. This occurred since only one D2D group was formed with all the UEs, during all the experiments. In this way, all the UEs kept the same video resolution. Further, since there is a single video download on

[3]https://www.nginx.com/
[4]https://www.cs.waikato.ac.nz/ml/weka/
[5]https://scikit-learn.org/stable/

the 4G network, the bandwidth demand is reduced, a factor that contributed to achieve higher resolutions.

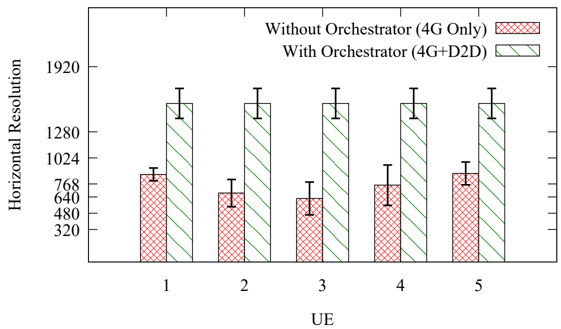

Fig. 3: Mean horizontal video resolution of each UE.

Finally, the confidence intervals in Figure 3 show larger variation in the resolution of some UEs in the case without the orchestrator. Such variation, and the differences of the mean resolution between the UEs from the case without the orchestrator, can be explained by the high demand at the 4G network, in comparison to the case with the orchestrator, which had only one UE downloading video through 4G. The high concurrence causes frequent resolution changes that can harm the QoE. Such harm occurs mainly if the video is at lower resolutions, which was the case without the orchestrator, where the horizontal resolution ranged between 480 and 768 pixels. With the orchestrator, although the variations were closer to the ones without the orchestrator, they were constant for all the UEs, showing that D2D has a more predictable behavior. Also, the variation with the orchestrator occurred at horizontal resolutions superior to 1280 pixels, reducing the harm to the QoE, since variations at this level are less easily perceived by the users.

Figure 4 shows the resolution behavior for each UE, with the use of the orchestrator. At the left side of the vertical line, the downloads occur with 4G, while the right side shows the downloads with D2D. During the downloads with 4G, the resolution changes constantly, due to the high concurrence at the 4G network. After some time, UE 2 surpasses the other UEs with higher resolutions and keeps a high difference in comparison to them. One possible reason is that UE 2 had a better 4G connection, which favored the download of higher resolutions. This was also observed by the prediction module, that decided in time 00:20 to use D2D and choose UE 2 to be the GO. Therefore, the other UEs started downloading the video from him, following the GO's video resolution pattern. During the use of D2D, there are variations at the resolution delivered to the GO, however, such resolution remained high in relation to the interval where all the UEs are connected through 4G, as shown in Figure 4. For the sake of simplicity, this variation was hidden from time 00:41.

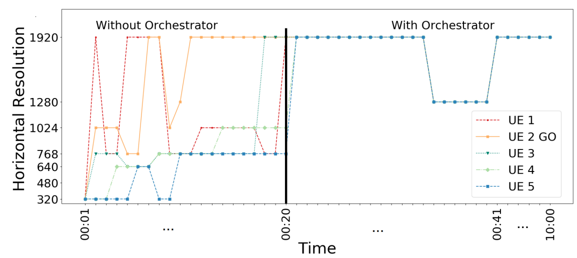

Fig. 4: Network behavior with the orchestrator.

As stated before, frequent resolution changes can harm the QoE. In order to address this issue, Yin et al. [13] proposed Equation 1 to evaluate the impact on the QoE based on the number of video resolution changes between chunks. The higher the result, the higher is the impact.

$$\frac{1}{K-1}\sum_{k=1}^{K-1}|q(R_{k+1})-q(R_k)| \tag{1}$$

According to Equation 1, K is the sum of video chunks, k is one downloaded chunk, and R_k is the bitrate reached with chunk k. Finally, function $q(R_k)$ maps the bitrate reached with chunk k to the video quality perceived by the user. Considering the network behavior from Figure 4 and Equation 1, the use of D2D from time 00:20, in relation to the interval where D2D is not used (between times 00:00 and 00:20), decreases the impact on the QoE from 1940878 to 186976, a decrease of about 90,4%.

C. Usage of the Optical Link

In this phase we evaluate the demand on the optical infrastructure. In our experiments, the video server is positioned at the edge (the UFMG's FUTEBOL testbed) or at the cloud (the UFRGS's FUTEBOL testbed, 1500km far from the UEs). We measure the capacity of the orchestrator to cope with video servers that are far away from the UEs, even though the predictor was trained in a local network. We consider as performance metric the amount of bytes sent by the video server. The experiment was run ten times, and considers a confidence interval of 95%.

Figure 5 presents the mean amount of bytes transmitted by the video server to the UEs, when positioned at UFRGS and at UFMG. There is a clear reduction of the demand in the optical link with the use of the orchestrator. Considering the video server located at UFRGS, the orchestrator managed to decrease the amount of bytes from 105.59 MB to 65.25 MB, a decrease of about 38.2%. The decrease is almost the same when the server is located at UFMG: From 112.08 MB to 67.35 MB (about 39,9%). Such improvement was only possible because the orchestrator left only one UE downloading with 4G and the optical link (the caching UE), while the other UEs started downloading from the caching UE with D2D.

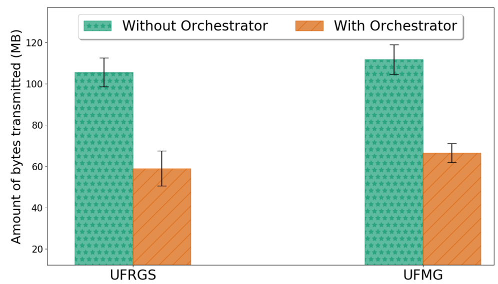

Fig. 5: Optical link demand with and without the orchestrator.

Further, a higher amount of bytes was measured at UFMG in relation to UFRGS, when comparing the scenarios with and without the orchestrator separately. This occurs because a video server closer to the 4G network is less subject to noise and packet losses, which allows UEs to request higher resolutions, resulting in a larger amount of bytes being transmitted.

Finally, for the scenario with the orchestrator, the difference of bytes transmitted between the video server at UFRGS and at UFMG is smaller than the difference observed at the scenario without the orchestrator. This shows that the orchestrator brings stability to the expected performance of video transmissions, independently of the position of the video server.

VI. Conclusions

This work proposed an orchestrator that uses D2D communication to improve the QoE of UEs connected to a 4G network, as well as to reduce the demand on the network backhaul. Differently from the state of the art, where the improvement of the QoE is the consequence of a higher QoS, the QoE metric is considered as the main goal of the orchestration decisions. The orchestrator uses machine learning to predict the video resolution at the UEs, and switches back and forth from D2D to 4G based on that information.

The proposal was evaluated on a testbed spanning more than 1500km, covering cases of video servers in the cloud or at the edge (near to the UEs). Results showed that the use of D2D improves the QoE by means of video transmissions with better resolutions. The mean horizontal resolution improved from 768 to 1280 pixels. Further, the impact on the QoE is decreased by around 90% considering the number of resolution changes, since D2D reduces the number of users downloading video through the 4G network, which favors the stability of the resolution. Finally, D2D relieves the backhaul links by up to 38%, since less video flows need to be transmitted over them.

As future work, we intend to evaluate the predictor in networks with background traffic, since it will affect the delivery quality. Further, UE mobility will be taken into account when transmitting data using D2D.

References

[1] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, March 2017.

[2] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP–: Standards and Design Principles," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*. ACM, 2011, pp. 133–144.

[3] V. Thrimurthulu and N. M. Sarma, "Device-to-device Communications in Long Term Evaluation-Advanced Network," in *Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on*. IEEE, 2017, pp. 818–823.

[4] A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1801–1819, Fourthquarter 2014.

[5] H. Zhou, H. Wang, X. Li, and V. C. M. Leung, "A Survey on Mobile Data Offloading Technologies," *IEEE Access*, vol. 6, pp. 5101–5111, 2018.

[6] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "3GPP LTE Traffic Offloading onto WiFi Direct," in *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, April 2013, pp. 135–140.

[7] X. Chai, T. Liu, C. Xing, H. Xiao, and Z. Zhang, "Throughput Improvement in Cellular Networks via Full-Duplex Based Device-to-Device Communications," *IEEE Access*, vol. 4, pp. 7645–7657, 2016.

[8] Y. Xu, S. Jiang, and J. Wu, "Towards Energy Efficient Device-to-Device Content Dissemination in Cellular Networks," *IEEE Access*, vol. 6, pp. 25 816–25 828, 2018.

[9] S. Ullah and C. S. Hong, "Outband D2D Communication for Layered Video Delivery in ICN Enabled Cellular Network," *Proceedings of the Korean Information Science Society*, pp. 1370–1372, 2016.

[10] A. E. Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-Based Traffic and Resource Management for Adaptive HTTP Video Delivery in LTE," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 988–1001, June 2015.

[11] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-Device Communication as an Underlay to LTE-Advanced Networks," *IEEE Communications Magazine*, vol. 47, no. 12, 2009.

[12] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-Device Communication in LTE-Advanced Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 1923–1940, Fourthquarter 2015.

[13] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 325–338, 2015.