AL-FEC application on NGMN-Edge computing integrated systems

Christos Bouras, Nikolaos Kanakis
Computer Technology Institute and Press “Diophantus”, Patras, Greece
Computer Engineering and Informatics Department, University of Patras, Greece
bouras@cti.gr, kanakisn@cti.gr

***Abstract*—Edge computing or edge networking is an architecture that uses one or more collaborative multitude of end-user clients or near-user edge devices to carry out a substantial amount of communication, control, management or other operations. Edge Computing for mobile networks is a new technology which is currently under standardization providing an IT service environment and cloud-computing capabilities at the edge of the mobile network in close proximity to the mobile end users. The aim of this technology is to reduce latency, ensure highly efficient network operation and service delivery, providing improved user experience. All of these can be translated into value and can create opportunities for operators, application and content providers enabling them to better utilize the mobile broadband capabilities. Furthermore, edge computing enables a new value chain for end users but also for industries allowing to efficient deliver their applications over the mobile network providing fresh business opportunities and new use cases. FEC is a feedback free error recovery method where the sender introduces redundant data in advance with the source data enabling the recipient to recover from different arbitrary packet losses. Recently, the adoption of FEC error control method has been boosted by the introduction of the powerful RaptorQ Application Layer FEC (AL-FEC) codes. In this work we propose the integration of AL-FEC error protection application at the edge layer. We propose a novel AL-FEC application architecture scheme based on RaptorQ codes and we analyze the performance enhancements such an error control architecture can introduce on Next Generation Mobile Networks (NGMN)-edge computing integrated systems.**

***Keywords*-forward error correction, next generation mobile network, fog computing**

I. Introduction

Edge or Fog Computing [1] is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing, typically, but not exclusively located at the edge of network. Fog computing has recently emerged as a more practical solution to enable the smooth convergence between cloud and mobile for content delivery and real-time data processing [2]. The term Fog computing was first introduced by Cisco [3]. The idea of Fog computing is by placing light-weight cloud-like facility at the proximity of mobile users; the Fog therefore can serve mobile users with a direct short-fat connection as compared to the long-thin mobile cloud connection. More importantly, as deployed at localized sites, Fog computing can provide customized and engaged location-aware services which are more desirable to mobile users. Mini-clouds are getting deployed closer to the edge (to the user) via private clouds. Enhanced Packet Core (EPC) can easily be expanded to include their own mini clouds. Having a small cloud at the EPC can help to deliver services close to users (at the edge) and confine traffic with the help of Software Defined Networks (SDNs). The fog enables user devices to become the virtualisation platform themselves. As such, they can lease some computing/storage capacity for applications to run on them.

Forward error correction (FEC) is a method of obtaining error control in data transmission in which the source (transmitter) sends redundant data and the destination (receiver) recognizes only the portion of the data that contains no apparent errors. Because FEC does not require handshaking between the source and the destination, it can be used for broadcasting of data to many destinations simultaneously from a single source. With mobile devices limited in resources by nature, mobile applications typically need to outsource their computation jobs to the cloud, and expect real-time response.

In [4], the authors introduce the definition of edge computing, followed by several case studies, ranging from cloud offloading to smart home and city, as well as collaborative edge to materialize the concept of edge computing. They also present several challenges and opportunities in the field of edge computing. The authors of [5] present a novel approach to allow mobile app developers to easily benefit from the features of MEC. In particular, they present a programming model and framework that directly fit the common app developers' mindset to design elastic and scalable edge-based mobile applications. The work of [6] offers a comprehensive definition of the fog, comprehending technologies as diverse as cloud, sensor networks, peer-to-peer networks, network virtualisation functions or configuration management techniques. They highlight the main challenges faced by this potentially breakthrough technology amalgamation. Finally, in [7] a practical experience and experimental work is presenting demonstrating the feasibility of using even extremely constrained nodes as fog gateways. The reported results demonstrate that good scalability and limited overhead can be coupled, via proper configuration tuning and implementation optimizations, with the significant advantages of containerization in terms of flexibility and easy deployment, also when working on top of existing, off-the-shelf, and limited-cost gateway nodes.

In this work we concentrate on the potentials provided by

Edge Computing on migrating error control handling on the edge devices. We investigate the feasibility of delegating AL-FEC based error control operations on the edge architecture.

The rest of this paper is organized as follows: In Section II we provide a description of the Fog/Mobile Edge computing on NGMN mobile networks and in Section III we present the concept of the applying FEC on the application layer and the most recent and valuable FEC codes. In Section IV we provide a very thorough description of the basic Edge network components. In Section V we present and analyze the proposed architecture migrating the application of AL-FEC error protection on the Edge gateways and we also provide a performance evaluation of the proposed schemes under several perspectives. Finally, in Section VI we provide a discussion on the proposed scheme and the results presented and possible future directions that could be followed in order to extend this work.

II. Mobile Edge computing

Mobile Edge Computing (MEC) [1] is a new technology which is currently being standardized in an ETSI Industry Specification Group (ISG) of the same name. Mobile Edge Computing provides an IT service environment and cloud-computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and in close proximity to mobile subscribers. The aim is to reduce latency, ensure highly efficient network operation and service delivery, and offer an improved user experience. Mobile Edge Computing is a natural development in the evolution of mobile base stations and the convergence of IT and telecommunications networking. Based on a virtualized platform, MEC is recognized by the European 5G PPP (5G Infrastructure Public Private Partnership) research body as one of the key emerging technologies for 5G networks (together with Network Functions Virtualization (NFV) and Software-Defined Networking (SDN)) [8]. In addition to defining more advanced air interface technologies, 5G networks will leverage more programmable approaches to software networking and use IT virtualization technology extensively within the telecommunications infrastructure, functions, and applications. MEC thus represents a key technology and architectural concept to enable the evolution to 5G, since it helps advance the transformation of the mobile broadband network into a programmable world and contributes to satisfying the demanding requirements of 5G in terms of expected throughout, latency, scalability and automation. MEC is based on a virtualized platform, with an approach complementary to NFV: in fact, while NVF is focused on network functions, the MEC framework enables applications running at the edge of the network. The infrastructure that hosts MEC and NFV or network functions is quite similar; thus, in order to allow operators to benefit as much as possible from their investment, it will be beneficial to reuse the infrastructure and infrastructure management of NFV to the largest extent possible, by hosting both VNFs (Virtual Network Functions) and MEC applications on the same platform. The environment of Mobile Edge Computing is characterized by low latency, proximity, high bandwidth, and real-time insight into radio network information and location awareness. All of this can be translated into value and can create opportunities for mobile operators, application and content providers enabling them to play complementary and profitable roles within their respective business models and allowing them to better monetize the mobile broadband experience. Mobile Edge Computing opens up services to consumers and enterprise customers as well as to adjacent industries that can now deliver their mission-critical applications over the mobile network. It enables a new value chain, fresh business opportunities and a myriad of new use cases across multiple sectors. The intention is to develop favorable market conditions which will create sustainable business for all players in the value chain, and to facilitate global market growth. To this end, a standardized, open environment needs to be created to allow the efficient and seamless integration of such applications across multi-vendor Mobile Edge Computing platforms. This will also ensure that the vast majority of the customers of a mobile operator can be served.

III. AL-FEC Reliability Control

Polar codes [9] are a major breakthrough in coding theory. They can achieve Shannon capacity with a simple encoder and a simple successive cancellation (SC) decoder when the code block size is large enough. Polar codes have brought significant interests and a lot of research work has been done mainly on code design and decoding algorithm. One of the most important decoding algorithms is the SC-list decoding which can perform as well as the optimal maximum-likelihood (ML) decoding with a list size of 32 for moderate code block sizes. A lot of performance simulations show that Polar codes concatenated with cyclic redundancy codes (CRC) and an adaptive SC-list decoder can outperform turbo/LDPC (Low Density Parity Check) codes for short and moderate code block sizes. Polar code has better performance than all the codes currently used in the 4G LTE systems, especially for short code length, thus it is considered as a candidate for the FEC module in 5G air interface design.

Fountain Codes are a new class of codes designed and ideally suited for reliable transmission of data over an erasure channel with unknown erasure probability [10]. A fountain code has properties similar to a water Fountain which can be thought as an infinite supply of water drops. Anyone who wants to collect the water drops holds a bucket under the fountain. When enough water is collected, the bucket is removed. Similarly with a digital source, a client gets encoded packets from one or more servers and packages once enough are obtained, the client can reconstruct the original file, which packets are obtained should not matter. They are rate-less in the sense that for a given message, the encoder can produce potentially infinite number of output symbols. Output symbols can be bits or more general bit sequences. However, random linear Fountain Codes have encoding complexity which makes them impractical for nowadays applications. Luby Transform (LT) codes have been proposed in [11] to reduce the encoding and decoding complexity of random linear Fountain Codes while maintaining the small overhead. With a good choice of degree distribution, i.e. the distributions of the edges in the Tanner graph, LT codes can come arbitrarily close to channel capacity with certain decoder reliability and logarithmically increasing encoding and decoding costs. In order to reduce the complexity even more, we can decrease the reliability of the decoder. Thus, we would have a reduced degree distribution resulting linear time encoding and decoding complexity. However, the decoder cannot decode all the input symbols with the lower degree distribution for the same overhead constraint. Therefore, utilizing an erasure correcting pre-code would then correct the erasures arising from the weakened decoder. If the pre-code is a linear time block code, like an LDPC code, Raptor Codes provide marvelous encoding and decoding speeds while providing near optimal performance for the BEC. Raptor codes are an extension of the other part of LT codes combined with a system of pre-coding. The design and degree distribution pre-coding is the heart of Raptor codes. Instead, the media data is protected using the application layer FEC with Raptor codes.

The new RaptorQ code [12] is a significantly more efficient AL-FEC code than its predecessor Raptor code. It provides superior flexibility and improved error protection and coding efficiency. The encoding process of RaptorQ code is mostly identical with that of Raptor code. However, RaptorQ code introduces certain design that ensure higher performance compared with the older Raptor code. A key difference between the two AL-FEC codes is that the RaptorQ code operates over larger finite fields that allows to overcome the performance limitations of Raptor code and achieve successful recovery with lower reception overhead. Some other important aspects of the enhanced properties of RaptorQ code are the increased number of possible source symbols and the increased number of generated encoding symbols. The expanded range of the encoding parameters simplifies the application of the AL-FEC protection and offers higher flexibility to RaptorQ.

IV. Edge Network Architecture

It is imperative for the 5G networks to be more than just a communication infrastructure. Computation and storage services, if supplied by the network, close to the devices, will allow applications to take benefit of low latency radio to provide very fast end-to-end response time. This will highly benefit both the customers (by giving timely responses) and the provider (by alleviating the load on the backbone network). This descent of processing from the cloud to the edge forms the definition of fog computing, and it would not be wrong to say that 5G networks cannot fulfill its promises without fog computing. Fog computing is not a feature, as most view it, but a necessary requirement for 5G networks to be able to succeed.

For an application to be called fog-ready, it must be designed to harness the full potential of the fog. Typically, a fog-ready application should have the following components.

A. Device component

The Device component is bound to the end devices. It performs device level operations, mostly, power management, redundancy elimination, etc. At times, when the end-device is not just a light client, it also hosts application logic demanding very low latency responses as this component is executed on the device itself. However, due to the resource constraints of the underlying device, this component should not contain heavy processing tasks.

B. Edge component

The edge component performs tasks that are critical in terms of latency and require such processing power that cannot be provided by end-devices. Furthermore, since the fog component is meant to run on fog devices close to the edge, the coverage of this component is not global. Thus this component should host logic that requires only local state information to execute. The edge/fog component is not bound to a particular kind of device. It is free to reside in any kind of device between the edge (consisting of end-devices) and the cloud. The mapping of the fog components to devices depends on the points of offload in the path from the edge to the cloud. Depending on the geographical coverage and latency requirements of the application, the fog component can be hosted on any of these points of offload. Mobile Edge Computing servers can be deployed at an aggregation point which may also be at the edge of the core network. The deployment of Edge components may depend on a number of factors, including scalability, physical deployment constraints or performance criteria.

C. Cloud component

Cloud component is bounded to the cloud servers in the core network. It contains logic for long-term data collected from the lower layers and for operations that do not have any sort of latency constraints. Applications tasks requiring large processing power and storage are suitable to be placed in the cloud component, so that they can harness the infinite resources of the cloud. Moreover, since the cloud layer is located at the apex of the network, it receives information from all devices and hence has a global knowledge of the entire system. Thus, application logic requiring knowledge of the global state of the system should be placed in the cloud component of the application.

V. Edge AL-FEC Protection

In this section we present the proposed scheme of this work and some performance results indicating the improvements that the AL-FEC protection on the edge can introduce.

A. Proposed AL-FEC Protection Scheme on the Edge

According to Fig. 1 the main concept of the proposed architecture is to move the AL-FEC application to the fog/edge gateways. While in traditional 4G deployments the AL-FEC was applied on the MBMS gateway, in this work we migrate

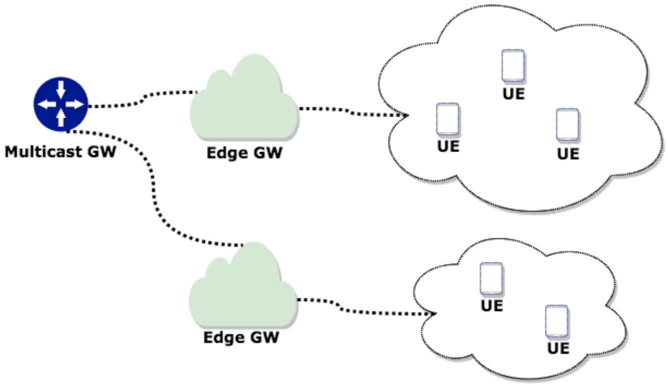

Fig. 1: AL-FEC on the edge gateways architecture

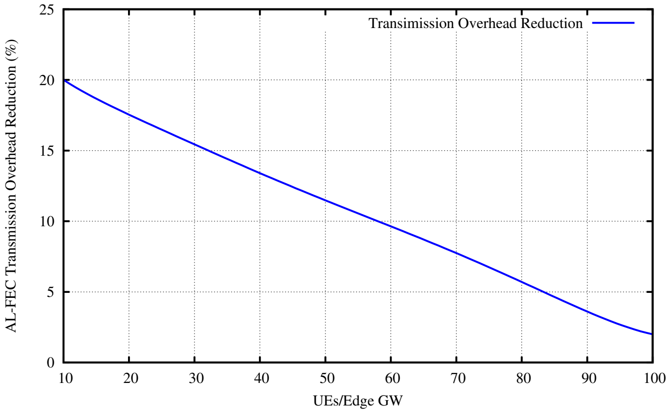

Fig. 2: AL-FEC Transmission Overhead vs UEs/Edge GW

the AL-FEC encoding on the edge gateways responsible for a cluster of end-user UEs.

While in previous architectures the multicast gateway was responsible to decide on the appropriate AL-FEC overhead to introduce on the transmitted object with the edge gateways and the delegation of AL-FEC overhead on this layer we can achieve a more coarse grained analysis and selection of the appropriate AL-FEC transmission overhead. Thus, this choice introduces increased efficiency on the AL-FEC application since we enable the reduction of the introduced transmission overhead and hence the network load according to the current reception conditions of the cluster of UEs. Apart from this, monitoring functions utilized on the AL-FEC overhead selection can also be migrated to the edge gateways level reducing further the cost of the deployments.

The transmitted content will be forwarded to the MBMS GW which is responsible to prepare the content for delivery to the end UEs. The MBMS GW is able to decide if AL-FEC protection should be introduced to the transmitted object towards the edge GWs according to the connection type between the MBMS GW and the edge GW. On the edge GW the forwarded transmitted object will be AL-FEC encoded and delivered to the end devices. The edge GW will be able to select an efficient amount of AL-FEC overhead during the AL-FEC encoding by implementing network functions on the network reception conditions on the end devices cluster. Furthermore, as in any other case, this error protection scheme will not be the only error control method applied in the same layer but will be augmented by conventional techniques. 3GPP [13] already defines a post-delivery procedure to provide file repair features for multicast delivery where a UE is able to determine which source symbols should have been received but have not and is able to send a request message for unreceived symbols through a point-to-point (ptp) delivery.

B. Performance Evaluation

In this part we provide a theoretical performance evaluation of the proposed AL-FEC protection scheme applied on the edge, presenting results on the impacts that the proposed scheme has on the average introduced AL-FEC transmission overhead and on the average network traffic with respect to both packets transmitted towards the UEs or packets transmitted on the backwards channels for control operations such as requesting the retransmission of lost packets from the MBMS GW.

1) AL-FEC Transmission Overhead: In this paragraph we present the impacts of moving the AL-FEC application protection on the edge to the Edge GWs on the amount of the transmission overhead introduced in average to the delivery of a transmitted object with respect to the transmission overhead applied from conventional approaches, where the transmission overhead was computed and applied on the MBMS GW, in order to achieve the same delivery outcome to the UEs participating on the reception of the object in both cases. On the first case we evaluate the impacts on the transmission overhead from the number of UEs allocated to an Edge GW perspective while, on the second case we evaluate the impacts on the introduced transmission overhead with respect on the delivery conditions of the network i.e. the average packet loss rate on the application layer.

In Fig. 2 we evaluate the reduction on the average amount of AL-FEC transmission overhead introduced on the delivery in terms of percentage again the number of UEs assigned to each Edge GW of the evaluated deployment with the average packet loss rate on the network fixed at 20%.

We can immediately note from the results that as the number of UEs per Edge GW are increased the required average transmission overhead is increased leading in decreased values of the transmission overhead reduction. In more details for 10 UEs per Edge GW we have almost 20% of transmission overhead reduction, for 50 UEs per Edge GW around 10% of reduction and for 100 UEs per Edge GW we have less than 5% reduction on the transmission overhead. This is something anticipated since as the number of UEs per Edge GW increases more UEs with different reception conditions are added to the control of the Edge GW and the amount of the average transmission overhead that the Edge GW should introduce to

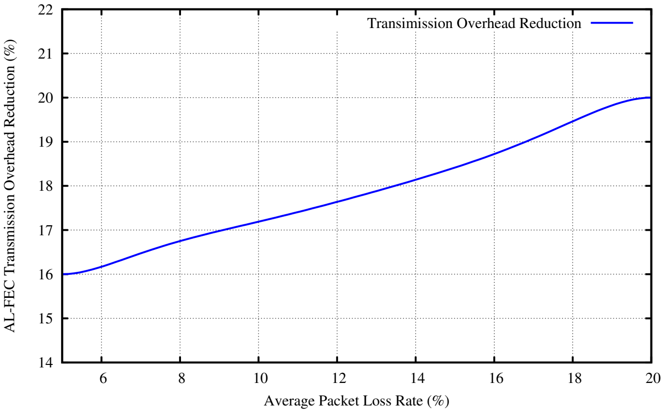

Fig. 3: AL-FEC Transmission Overhead vs Packet Loss Rate

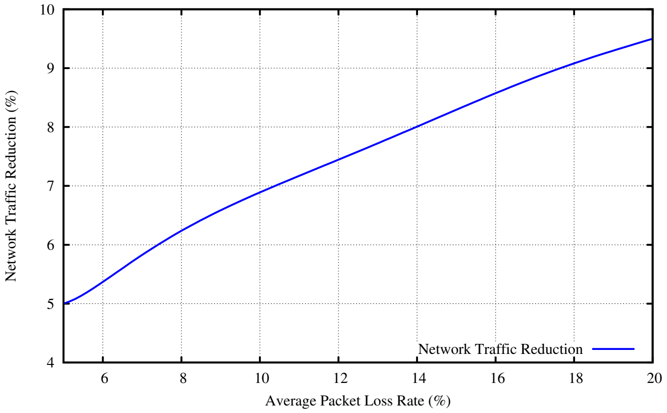

Fig. 4: Network Traffic vs Packet Loss Rate

the delivery gets increased. However, the reduction on the introduced AL-FEC transmission overhead is even in the case of the 100 UEs per Edge GW.

In Fig. 3 we evaluate the reduction on the average AL-FEC transmission overhead of the delivery against different values of average packet loss rate in the range of 5% to 20% providing an indication on how the proposed scheme operates on different reception conditions of the network with the number of UEs per Edge GW fixed at 10.

The trend of the curve indicates that as the average packet loss rate of the network is increased the reduction on the average introduced AL-FEC transmission overhead is increased too having 16% of reduction with 5% of packet loss rate and almost 20% of reduction when the average packet loss rate reaches 20%. This behaviour is somehow expected since reduction is computed against a centralized approach of computing the necessary transmission overhead on the multicast GW and since the proposed scheme introduces a sense of clustering on how the transmission overhead is computed the Edge GW is able to handle more efficient UEs with high packet loss rate.

2) Network Traffic: In this second part of the evaluation results we present the impacts of the proposed error protection scheme applied on the edge components on the network traffic load of the mobile network, i.e. the total amount of packets transmitted towards the mobile devices and on the backwards channels from the mobile devices where for instance a UE can request the retransmission of lost packets through a post-delivery repair session in order to reconstruct the partially delivered object.

In Fig. 4 we provide simulation results evaluating the reduction on network traffic in terms of packets communicated against different values of average packet loss rate in the range of 5% to 20% and the UEs assigned on each Edge GW fixed at 10.

An immediate remark is that the network traffic reduction is increased with the average packet loss rate of the network increase. In more details, with 5% packet loss rate the scheme achieves more than 5% of network traffic reduction, with 12% packet loss rate the reduction reaches more than 7% and with 20% of packet loss rate the reduction is more than 9%. This behaviour is a direct consequence of the increased reduction on the introduced transmission overhead while the average packet loss rate is increased since less repair symbols are transmitted compared to the case of the centralized computation of the transmission overhead.

VI. Conclusions & Future Work

In this work we have presented a novel approach for applying error control on NGMN where the fog/edge layer is a newly introduced layer which can dramatically boost the capabilities of 5G/IoT deployments. We have described an architecture where the application of AL-FEC can be migrated to the edge devices of a network providing coarse-grained error control capabilities on the network. We have presented some performance results indicating the gains that this kind of deployment can introduce on the overhead introduced by the AL-FEC protection. Furthermore, such a scheme is able to provide significant reduction on the overall network load since is able to provide more efficient protection on the first hand with AL-FEC mitigating the need of utilizing conventional error control methods like ARQ. We have investigated the impacts of the proposed scheme on the efficiency of the AL-FEC application providing simulation results for different deployment scenarios and network conditions. The results indicate that the proposed scheme is able to provide reduction on the transmission overhead realizing the application of AL-FEC protection on the Edge.

The future steps that could follow and extend this work are based device-to-device communication descibed on edge model architectures. The IoT service architecture exposes to the IoT devices computational resources and an interface to request end-to-end connectivity between end IoT devices or between devices and services.

Device-to-device (D2D) communication, which utilizes mobile devices located within close proximity for direct connection and data exchange, holds great promise for improving the efficiency of mobile multimedia in NGMN networks. A communication between two UEs in proximity by means of communication path established between the UEs. When UEs are near each other it allows in communication high data rates and low end-to-end delay. Direct communication compared to the normal downlink and uplink saves energy and thus improves the radio resource utilization. Using direct path between UEs compared to infrastructure path offloads cellular traffic, reduces congestion and in that way benefits also other cellular UEs.

In the context of this D2D communication model the proposed scheme could be extended in order to utilize the D2D communication capabilities and create a collaborative AL-FEC protection deployment on the Edge and below where repair symbols could also be received from a UE in proximity and not only from the Edge GW augmenting the reconstruction phase of AL-FEC protection.

References

[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing?a key technology towards 5g.”

[2] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, “An overview of fog computing and its security issues,” *Concurrency and Computation: Practice and Experience*, 2015.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.

[5] G. Orsini, D. Bade, and W. Lamersdorf, “Computing at the mobile edge: Designing elastic android applications for computation offloading,” in *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*, Oct 2015, pp. 112–119.

[6] L. M. Vaquero and L. Rodero-Merino, “Finding your way in the fog: Towards a comprehensive definition of fog computing,” *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014. [Online]. Available: http://doi.acm.org/10.1145/2677046.2677052

[7] P. Bellavista and A. Zanni, “Feasibility of fog computing deployment based on docker containerization over raspberrypi,” in *Proceedings of the 18th International Conference on Distributed Computing and Networking*, ser. ICDCN ’17. New York, NY, USA: ACM, 2017, pp. 16:1–16:10. [Online]. Available: http://doi.acm.org/10.1145/3007748.3007777

[8] 5G Infrastructure Public Private Partnership, “5G Vision: The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services,” 5G Infrastructure Public Private Partnership, https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf, Tech. Rep., 2015.

[9] R. Pedarsani, S. H. Hassani, I. Tal, and E. Telatar, “On the construction of polar codes,” in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 11–15.

[10] D. J. MacKay, “Fountain codes,” in *Communications, IEE Proceedings-*, vol. 152, no. 6. IET, 2005, pp. 1062–1068.

[11] M. Luby, “LT codes,” in *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, 2002, pp. 271–280.

[12] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, “RaptorQ Forward Error Correction Scheme for Object Delivery,” RFC 6330, Internet Engineering Task Force, Aug. 2011. [Online]. Available: http://tools.ietf.org/rfc/rfc6330.txt

[13] 3GPP, “Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (Release 10),” 3rd Generation Partnership Project (3GPP), TS 26.346, 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/26346.htm