# Re-Architecting the Packet Core and Control Plane for Future Cellular Networks

Ali Mohammadkhan*, and K.K. Ramakrishnan†
University of California, Riverside, U.S.
*amoha006@ucr.edu, †kk@cs.ucr.edu

***Abstract*—With the rapid increase in the number of users and changing pattern of network usage, cellular networks will continue to be challenged meeting bandwidth and latency requirements. A significant contributor to latency and overhead in cellular networks is the complex control-plane involving many message exchanges across multiple components in the packet core, base station, and user equipment.**

**We propose CleanG, a new packet core architecture *and* significantly more efficient control-plane protocol, that exploits the capabilities of modern-day Network Function Virtualization (NFV) platforms. In CleanG, we have consolidated the core components into a set of virtual network functions on an NFV platform. With the elastic scalability offered by NFV, the data and control sub-components of the core functions can scale, adapting to workload demand. CleanG eliminates the use of GPRS Tunneling Protocol (GTP) Tunnels for data packets and the associated complex protocol for coordination across multiple, distributed components for setting up and managing them, as specified in the 3rd Generation Partnership Project (3GPP) architecture and protocol standard, while retaining similar essential functionality for security, mobility, and air-interface resource management.**

**Measurements on our testbed show that CleanG substantially reduces both control *and* data plane latency, and significantly increases system capacity.**

## I. Introduction

Cellular networks have evolved from just providing voice communication between people, to ubiquitous data, video and voice connectivity for people as well as supporting machine-to-machine (M2M) and Internet of Things (IoT) communication. However, cellular networks continue to face significant capacity, latency and scalability challenges. While some of these concerns are being addressed in the 5G networks currently being deployed, fundamental problems remain. We believe it is highly desirable to take a new look at the architecture and associated protocols with the goal of improving performance to meet upcoming challenges. The architecture for the Long Term Evolution (LTE) cellular network, which has been widely deployed worldwide, called System Architecture Evolution (SAE), is shown in Fig. 1. The Evolved Packet Core (EPC) is normally implemented as a number of separate hardware components partly because of the need to scale control and data plane independently. This distribution of functionality among a set of distributed components results in significant protocol overhead, especially for setting up GTP tunnels. To transition from fourth generation (LTE) to fifth generation (5G) cellular networks, in what is termed as Release 14 of the 3GPP specification, the Control and User Plane Separation of EPC nodes (CUPS) architecture was introduced. This architecture takes the separation between the control and data plane one step further. The control part of the SGW and PGW are separated from their data plane counterparts and new interfaces are defined between these new components. The proposed 5G core architecture is based on this CUPS architecture. The greater degree of separation between the control plane and data plane components and more granular dedication of the tasks to components may simplify each component. However, it increases the number of components involved in serving users' requests, the coordination required and the number of messages exchanged between them.

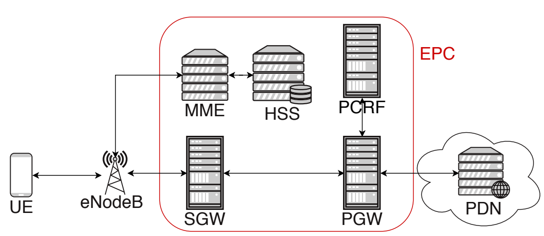


Fig. 1: LTE system architecture

978-1-7281-2700-2/19/$31.00 2019 © IEEE

In this paper, we re-evaluate the role of Software Defined Networking (SDN) and NFV for the next generation cellular networks. We propose an architecture that takes advantage of NFV's features, including scalability, service chaining, and shared memory. In addition, we demonstrate how the control protocol can be improved by using different techniques such as re-ordering of messages, consolidation, or using shared memory. We show that the data plane can also be more efficient by using simple GRE tunneling instead of GTP tunnels, as it eliminates the need to exchange tunnel identifiers in advance. The current fields in IP packets can be used for providing different classes of service. Finally, we have implemented all three alternatives (CleanG, EPC, and CUPS-based architecture similar to 5G) to show how CleanG performs in comparison to alternatives.

## II. Proposed CleanG Architecture

We propose CleanG as a simple, efficient, and scalable architecture for next generation cellular core networks. CleanG architecture is shown in Fig. 2. This architecture is based on the OpenNetVM framework [1], a high performance NFV platform on top of the Data Plane Development Kit (DPDK)

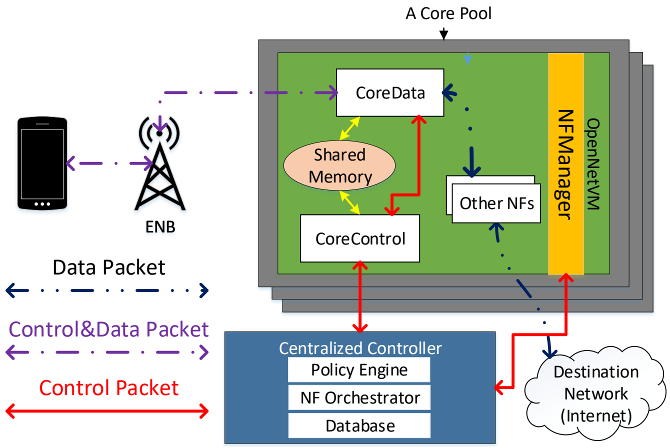


Fig. 2: CleanG architecture

library. A primary goal of the CleanG system architecture is to introduce minimum delay for an update in the cellular control plane to be reflected in the necessary changes to the cellular data plane. The CleanG architecture is built on the basis of a tightly knit control and data plane for the cellular core network, running on the OpenNetVM framework. CleanG instances can be instantiated in the edge, central offices, data centers, or where and when demand exists. The two main components, the CoreData and CoreControl implementing the cellular data and control planes respectively thus share data and state using the OpenNetVM shared memory, buffers, and queues thus resulting in minimal delay, as desired. In our design, the CoreData data plane component receives all packets from network interfaces, encapsulates/decapsulates and forwards data packets based on the rules provided by CoreControl as a Flow table (Hash table using DPDK's cuckoo hash [2]) in shared memory. If CoreData does not have a rule for a packet (e.g., control packets), it forwards these packet to the CoreControl component. The forwarding between these components takes place without data movement of the packet, as it is achieved by adding a pointer to the packet buffer to the receive queue of the CoreControl NF. The decision to delegate the responsibility of reading all the packets from interfaces by CoreData saves resources (mainly CPU cycles) in our design. While it is normally done by the *flow director* NF in the OpenNetVM framework, those responsibilities are merged for efficiency in this CoreData design.

## III. Proposed CleanG Protocol

Leveraging NFV enables us to consolidate the various components of the cellular packet core on to a single server and derive performance improvements. First, we retained the original 3GPP protocol framework and implemented it within the CleanG system architecture. However, as we show in our evaluations, the improvement is somewhat limited, as measured by the task completion times for control events. We see a significant opportunity to dramatically improve performance when we are able to conflate the improvements from the architectural consolidation with a careful re-design of the control plane protocols to take advantage of the new architecture. There are two main opportunities we take advantage of to improve the cellular control plane protocol. The consolidation of the cellular core components eliminates the need to keep state synchronized among different components and the consequent need for a number of additional messages to be exchanged to confirm the state update. A second major improvement comes from eliminating the process of setting up the GPRS Tunneling Protocol (GTP) tunnels by using the simpler Generic Routing Encapsulation (GRE) tunnels. We also use the IP packet header fields to provide the information needed for deciding how to forward packets for different classes of the service.

In addition to the techniques we already mentioned, we use the following measures to improve the control plane protocol:

- We optimize the protocol for the typical scenario and then address exceptions instead of burdening the common case with unnecessary messages. For example, by knowing the security algorithm used by the user previously, we take advantage of this soft state for the user when he uses the same algorithm for the next connection. If however the user changes the algorithm for the new connection, an extra message is exchanged to change the selected algorithm.
- The central controller is only used for high-level monitoring and policy enforcement. The controller is not involved in the exchange of each and every control message.
- Where appropriate, we take advantage of changing the order of the messages exchanged to reduce the number of messages. For example, mutual authentication needs a minimum of three messages, but by initiating from the client, we can reduce the number of messages exchanged.
- We can merge information across what would otherwise have been carried across multiple messages where appropriate.
- Where appropriate, we delegate responsibility to other network components. For instance, we allow the eNodeB to participate in the authentication thus reducing the need to send an additional message back to the CoreControl.
- By taking advantage of shared memory between the cellular core components, we reduce the need for the exchange of several messages, and the shared data structure allows for synchronization and sharing of information.
- Based on the user service required (e.g., delay tolerant or delay sensitive), events are handled differently. For handover of a delay sensitive stream, packets are duplicated. But, they are not duplicated for delay tolerant streams.
- If not necessary, control message exchanges are de-serialized. For example, location can be updated in the HSS while the attach acknowledgment is sent to the UE in parallel.

### A. Forwarding data packet in CleanG

Forwarding in the 3GPP protocol used in LTE & 5G is described in detail in [3], [4]. For each class of service of each user, a distinct tunnel is established between the entities (SGW, PGW, and eNB in 4G, and between UPF and gNB in 5G) forwarding the data packets. Tunnel IDs are exchanged between these entities, mediated by the control plane components. While we retain QoS treatment and admission control for the data plane, there are a number of significant differences in CleanG. For downstream flows, based on the destination IP address, the UE's IP, a Traffic Flow Template (TFT) of the user is fetched. The packet header is matched against the TFT rules and based on that, the class of the service for the flow is selected. Besides that, we obtain the IP of the destination eNodeB, which then is used to encapsulate the packet with a Generic Routing Encapsulation (GRE) header. Based on their class, packets will be forwarded with the appropriate priority by CoreData, with the Differentiated Services Code Point (DSCP) field set accordingly. DSCP is used by routers in the backhaul network to forward packets and the serving eNodeB to select the proper radio bearer. In eNodeB, based on the IP address of the user and the DSCP bits, the proper radio bearer is chosen and the packet forwarded to the user. Upstream forwarding resembles the downstream one, but the details are omitted for the sake of brevity. To sum up, in CleanG, instead of creating separate GTP tunnel headers (tags) for each user and class of service, and exchanging a number of messages between the different components to set up the tunnels across different components, we use an encapsulating IP header and DSCP bits to provide a similar functionality, without having to set it up between these components. This is far more efficient, without needing *any message exchanges.*

## IV. Summary of Evaluation

We compare our proposed CleanG architecture and protocol with the current 3GPP protocol and implementation of cellular networks, where the key EPC components are separate and distinct entities. We also compare to a CUPS-based architecture with an SDN controller between the data and control entities. Currently, there is no open source implementation for 5G and existing open source implementations of the EPC for LTE cellular networks have limitations in terms of performance, including OpenAirInterface [5]. Thus, to perform an apples-to-apples fair comparison to our proposed CleanG system, we implemented each of the three variants as carefully and fairly as possible on the high performance OpenNetVM DPDK-based framework. All of the three architectures and protocols were therefore evaluated over the OpenNetVM framework. As a consequence we are able to clearly show the performance improvement of CleanG is based on the architecture and protocol improvements and not because of the use of a faster platform. In our implementation, we included all the key messages and the primary information fields that are key to the protocol operation.

The protocol and system implementation used C (about 20K lines of code) to get the highest performance we could achieve. We use one server to generate the user workload traffic at scale. Based on reasonably representative user behavior, we are able to scale the system up to support millions of users.

We generated a number of representative user events (e.g., UE connecting/disconnecting to/from the network, the UE going to idle state, moving causing handover, etc.) for each UE emulated. These events are generated and arrive at the NF representing the eNodeB. The workload generator maintains state for each UE to generate the appropriate messages and the UE changes state when a control plane event happens.

### A. *Overview of Evaluation Results*

We compare three alternative architectures and protocols, and we observed the highest capacity and lowest delay with the CleanG architecture and protocol. With only running one instance of the components, CleanG marginally outperforms the other two alternatives. However, more importantly, with similar amounts of resources (CPU cores), CleanG supports a dramatically more significant number of users (Up to 3 times). The bottleneck for system capacity in the CUPS-based architecture was the SDN controller, and in the 3GPP EPC, the SGW becomes the bottleneck. We observed that the completion time of different control plane tasks (such as attach, handover, or idle-to-active) is much higher in the CUPS-based approach because of the delay caused by SDN controller and it is higher in the 3GPP EPC because of the higher load on each component and the larger number of messages exchanged in comparison to CleanG. Packets going through the CleanG core also observe a much smaller delay in comparison to the 3GPP EPC approach, but they are comparable (slightly better) to the CUPS-based approach because both are not affected by the load of the control plane components and in handling a smaller number of messages.

## V. Related Work

In recent years, improving the cellular network has been the subject of numerous efforts both in research and industry [6], [7], especially as 5G is being deployed. However, there has been only limited focus on simplifying the cellular protocols. One work that has focused on the simplification of control plane is [8], which rightfully points to the delays caused by control plane complexity and suggests improvements to it. However, we believe the potential of the optimization is not completely unlocked if possible improvements to the architecture and data plane forwarding are not also considered. A large body of recent research has focused on the separation of data and control plane and with and without the intervention of the SDN controller. [9] suggests using an SDN controller to shorten the path between UE's using P2P services. While this technique applies to the CleanG architecture as well, we observe that we can achieve similar or better performance by instantiating Core instances closer to the users. In [10] the EPC-edge is introduced as a termination for GTP tunnels, in addition to the separation. However, it still suffers from the complexity of the 3GPP protocol and architecture. Other work in the same vein includes [11], and [12].

Another direction has been to introduce packet cores close to the edge of the network (e.g., at the telephony-related local central offices) [13]. CleanG can use a similar approach for locating the CleanG Core Pools. Recently, [14] investigated the effects of unreliability in the virtualized core network and suggested using a proxy to mask these failures from the core message exchanges, as it can significantly hamper performance. Because of the use of a reliable underlying transport protocol and the fact that the number of messages is reduced and components are consolidated, we mitigate this effect in CleanG. Moreover, we can take advantage of reliability approaches for NFV platforms [15], [16].

A short motivation and introduction to CleanG's architecture were described in [17]. [18] follows the approach outlined in [17] for the design of the NFV-based cellular core by having the separation between the data and control NFs, with techniques to make the state tables more efficient. However, it retains the 3GPP protocol and suffers its inherent inefficiencies. Another closely related work is [19], which seeks to understand the bottlenecks in virtualizing cellular core network functions. Their observations are in sync with what we observed as well to motivate CleanG. Recently [20] has proposed an approach to use a streaming framework to implement specific cellular core components such as the MME efficiently, while retaining the 3GPP protocol. While the streaming framework is indeed useful, we believe a greater opportunity is also to see how the changes in both the architecture and protocol can be combined as in CleanG. Finally, techniques similar to [21] can be used to balance the load between different cores and pools in CleanG.

## VI. Conclusion

CleanG provides a simplified and easily scalable architecture and protocol for future cellular networks by intelligent adoption of NFV and SDN and we studied its performance in a number of scenarios. While we have not addressed all the possible scenarios and corner cases in this paper, we believe our design principles will be valuable even in a complete, production architecture and implementation of a cellular network that handles all possible exceptions. We also observed while it is possible to improve the scaling of the data plane by leveraging the separation between control and data plane, as in SDN, in the cellular network it has significant consequences. We will make the code for CleanG and our implementation of EPC and CUPS-based approaches open source on Github, to facilitate further research, evaluation and development work on the cellular network protocol and architecture.

## Acknowledgment

This work was supported by US NSF grants CNS-1522546, CRI-1823270 and CNS-1763929. The work was also partially supported by the ARO DURIP grant W911NF-15-1-0508, Department of the Army, US Army Research, Development and Engineering Command grant W911NF-15-1-0508.

## References

[1] W. Zhang, G. Liu, W. Zhang, N. Shah, P. Lopreiato, G. Todeschi, K. Ramakrishnan, and T. Wood, "Opennetvm: A platform for high performance network service chains," in *Proceedings of the 2016 Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, HotMIddlebox '16, (New York, NY, USA), pp. 26–31, ACM, 2016.

[2] Intel Corp., "Intel data plane development kit: Programmer's guide," 2019.

[3] 3GPP, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," Technical Specification (TS) 23.401, 3rd Generation Partnership Project (3GPP), 2017. Version 15.3.0.

[4] 3GPP, "General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)," Technical Specification (TS) 29.281, 3rd Generation Partnership Project (3GPP), 2017. Version 15.2.0.

[5] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "Openairinterface: A flexible platform for 5g research," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 33–38, Oct. 2014.

[6] A. Gupta and R. K. Jha, "A survey of 5g network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[7] P. Rost, A. Banchs, I. Berberana, M. Breitbach, M. Doll, H. Droste, C. Mannweiler, M. A. Puente, K. Samdanis, and B. Sayadi, "Mobile network architecture evolution toward 5g," *IEEE Communications Magazine*, vol. 54, pp. 84–91, May 2016.

[8] Y. Li, Z. Yuan, and C. Peng, "A control-plane perspective on reducing data access latency in lte networks," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom '17, (New York, NY, USA), pp. 56–69, ACM, 2017.

[9] R. Saunders, J. Cho, A. Banerjee, F. Rocha, and J. Van der Merwe, "P2p offloading in mobile networks using sdn," in *Proceedings of the Symposium on SDN Research*, SOSR '16, (New York, NY, USA), pp. 3:1–3:7, ACM, 2016.

[10] S. Matsushima and R. Wakikawa, "Stateless user-plane architecture for virtualized epc (vepc)," *draft-matsushima-stateless-uplane-vepc-04 (work in progress)*, 2015.

[11] X. An, W. Kiess, and D. Perez-Caparros, "Virtualization of cellular network epc gateways based on a scalable sdn architecture," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, pp. 2295–2301, IEEE, 2014.

[12] J. Kempf, B. Johansson, S. Pettersson, H. Lüning, and T. Nilsson, "Moving the mobile evolved packet core to the cloud," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on*, pp. 784–791, IEEE, 2012.

[13] E. C. Perkins, "M-Cord:Re-architecting Mobile Infrastructure to Enable 5G Networks." https://www.opennetworking.org/solutions/m-cord/. [Online; accessed 19-May-2018].

[14] M. T. Raza, D. Kim, K. H. Kim, S. Lu, and M. Gerla, "Rethinking lte network functions virtualization," in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pp. 1–10, Oct 2017.

[15] J. Sherry, P. X. Gao, S. Basu, A. Panda, A. Krishnamurthy, C. Maciocco, M. Manesh, J. a. Martins, S. Ratnasamy, L. Rizzo, and S. Shenker, "Rollback-recovery for middleboxes," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, (New York, NY, USA), pp. 227–240, ACM, 2015.

[16] S. Rajagopalan, D. Williams, and H. Jamjoom, "Pico replication: A high availability framework for middleboxes," in *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, (New York, NY, USA), pp. 1:1–1:15, ACM, 2013.

[17] A. Mohammadkhan, K. Ramakrishnan, A. S. Rajan, and C. Maciocco, "Cleang: A clean-slate epc architecture and controlplane protocol for next generation cellular networks," in *Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking*, CAN '16, (New York, NY, USA), pp. 31–36, ACM, 2016.

[18] Z. A. Qazi, M. Walls, A. Panda, V. Sekar, S. Ratnasamy, and S. Shenker, "A high performance packet core for next generation cellular networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, (New York, NY, USA), pp. 348–361, ACM, 2017.

[19] Z. A. Qazi, P. K. Penumarthi, V. Sekar, V. Gopalakrishnan, K. Joshi, and S. R. Das, "Klein: A minimally disruptive design for an elastic cellular core," in *Proceedings of the Symposium on SDN Research*, SOSR '16, (New York, NY, USA), pp. 2:1–2:12, ACM, 2016.

[20] J. Cho and J. Van der Merwe, "Poster: A new scalable, programmable and evolvable mobile control plane platform," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom '17, (New York, NY, USA), pp. 540–542, ACM, 2017.

[21] A. Banerjee, R. Mahindra, K. Sundaresan, S. Kasera, K. Van der Merwe, and S. Rangarajan, "Scaling the lte control-plane for future mobile access," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '15, (New York, NY, USA), pp. 19:1–19:13, ACM, 2015.