Journal of Network and Computer Applications 115 (2018) 94–102

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

5G Virtualized Multi-access Edge Computing Platform for IoT Applications

Han-Chuan Hsieh[a,*], Jiann-Liang Chen[a], Abderrahim Benslimane[b]

[a] Department of Electrical Engineering, National Taiwan University of Science & Technology, Taipei, Taiwan, ROC
[b] Department of Computer Science, University of Avignon, France

ARTICLE INFO

Keywords:
Virtualized Multi-access Edge Computing (vMEC)
Network Function Virtualization (NFV)
Container-based Virtualization Technology (CVT)
Software Defined Networking (SDN)
Internet of Things (IoT)
Virtual Network Function (VNF)

ABSTRACT

The next generation of fifth generation (5G) network, which is implemented using Virtualized Multi-access Edge Computing (vMEC), Network Function Virtualization (NFV) and Software Defined Networking (SDN) technologies, is a flexible and resilient network that supports various Internet of Things (IoT) devices. While NFV provides flexibility by allowing network functions to be dynamically deployed and inter-connected, vMEC provides intelligence at the edge of the mobile network reduces latency and increases the available capacity. With the diverse development of networking applications, the proposed vMEC use of Container-based Virtualization Technology (CVT) as gateway with IoT devices for flow control mechanism in scheduling and analysis methods will effectively increase the application Quality of Service (QoS). In this work, the proposed IoT gateway is analyzed. The combined effect of simultaneously deploying Virtual Network Functions (VNFs) and vMEC applications on a single network infrastructure, and critically in effecting exhibits low latency, high bandwidth and agility that will be able to connect large scale of devices. The proposed platform efficiently exploiting resources from edge computing and cloud computing, and takes IoT applications that adapt to network conditions to degrade an average 30% of end to end network latency.

1. Introduction

The development of fifth generation (5G) mobile communication systems enables the integration of mobile devices and Internet of Things (IoT) applications on larger-scale sensor networks (Ratasuk et al., 2015). 5G is a collective name for technologies and methods enable support future networks that satisfy extreme capacity and performance demands. The communications industry aspires to meet the low-latency and gigabit throughput expectations to users of 5G. 5G provides optimized control during high density connection, which can support connectivity and management of networking devices on a large scale, at low cost, with low power consumption (Iwamura, 2015). IoT such as sensors or wearable devices for cloud computing platforms will grow rapidly with 5G technology and applications. IoT applications include Wireless Sensor Network (WSN), bridges monitoring, earthquake monitoring, tsunami alerts, autonomous cars, intelligent vehicles … etc., all of which will support the demand for low-latency and real-time services that become increasingly intelligent (Xiao et al., 2017; Al-Fuqaha et al., 2015; Raafatet al, 2017; Habib and Marimuthu, 2011).

The emerging demand for 5G mobile networks led to a complete network paradigm refurbishment that involves leveraging Network Function Virtualization (NFV) and Virtualized Multi-access Edge Computing (vMEC) (Mach and Becvar, 2017; Lyuet al, 2017). NFV is a concept that is supports network applications of Software Defined Networking (SDN), and replaces traditional networking hardware with software solutions (Cau et al., 2016). Many investigations of the virtualization investigations of traditional networking hardware, such as load balancers, seek to improve traditional network architecture. Researchers have presented various network structures that are based on SDN. The vMEC application platform, operated along Service-oriented Architecture (SOA) guidelines, seeks to provide communications services, interface design for middleware and application services. Communication services support applications through an Application Programming Interface (API) that is defined by mutual communication with application platforms. The interfaces are not affected by transmitted messages. Owing to an increase in the use of application services, the edge computing technology is presented to improve network resources efficiency. In September 2014, European Telecommunications Standards Institute (ETSI) developed the concept of edge computing technology, and could effectively reduce the burden on network equipment, and support the creation of a unique mobile experience (Salman et al., 2015). Fig. 1 presents vMEC as a new IoT applications framework that helps the service provider to offer an improved user experience and a network platform that reduces network loading (Dusia et al., 2015). Cloud computing technology is associated with boundary data processing, storage, and host-level management close to the end

* Corresponding author.
E-mail addresses: D10007501@mail.ntust.edu.tw (H.-C. Hsieh), abderrahim.benslimane@univ-avignon.fr (A. Benslimane).

https://doi.org/10.1016/j.jnca.2018.05.001
Received 31 October 2017; Received in revised form 25 March 2018; Accepted 6 May 2018
Available online 10 May 2018
1084-8045/

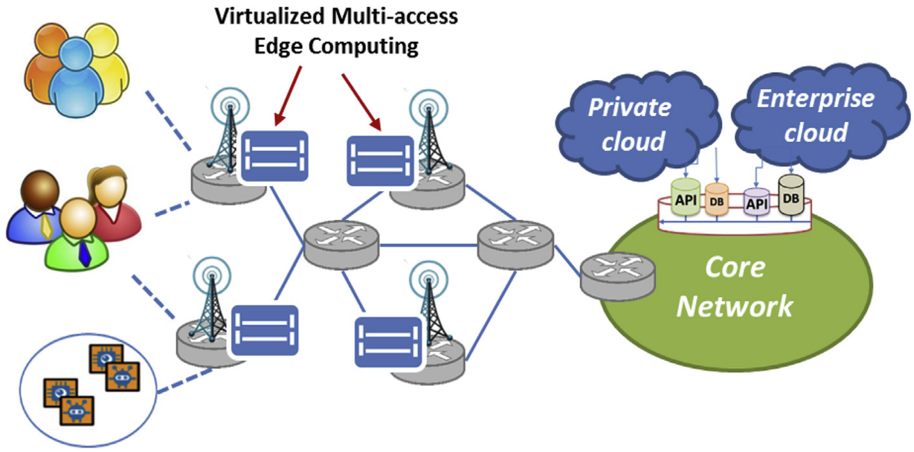


Fig. 1. vMEC service framework.

user node. vMEC communicates directly with the end users in response to content staging of network applications. The end users thus obtain content directly from vMEC. System resources can be effectively saved in network loading, significantly improving network performance, and shortening the response time and increasing the throughput capacity.

For this work, IoT scenarios based on virtualized vMEC platforms for flow control mechanism development for performance analysis, which is expected to reduce network congestion, reduce application latency, and give users a better QoS. Open source software and hardware platforms can be used to build virtualize networks efficiently. In order to make use of the app-centric Container-based Virtualization Technology (CVT), easy deployed features can be used to improve configuration management application service platforms, but efficiency of networking applications must be improved on a virtualized platform (Ismail et al., 2015; Jararweh et al., 2016).

Section 2 presents the background of IoT and virtualization technologies for the proposed framework. Section 3 discusses the system model and specifies the traffic control mechanism. Section 4 designates the proposed IoT gateway platform architecture. Summary of virtualized IoT gateway platform development and performance analyses are included in Section 5.

2. Background

Research into IoT seeks to develop commercial products that have an immediate impact human life. IoT applications can be defined as unique objects that are connected to each other over the internet to perform information exchanging, object identification, location updating, and security monitoring. IoT service providers are working on vMEC, where the computing, storage and networking resources are integrated and hosted at the edge of a network. These objectives would be difficult and expensive to realize without bringing the cloud closer to the edge of the network and to the users.

2.1. Virtualized Multi-access Edge Computing

Virtualized Multi-access Edge Computing (vMEC) platform architecture comprises of three parts, which are the host infrastructure, the application platform and application services (Li et al., 2016; Nunnaet al, 2015). The proposed hardware devices are disaggregated by NFV can be divided into hardware resources and a virtualization layer architecture, which provide computing hardware, storage, and control functions. In the virtualization layer are associated with virtualized computing processing, temporary storage, exchange and an MEC network hardware equipment management system. MEC can deliver application services that support external connection management (Joy, 2015; Moldován and Varga, 2012). An intermediary layer of middleware, applications and services may be required to provide network traffic control, a wireless messaging network, communications and application services that perform intermediary software registration and other functions (Pfaff et al., 2015; Mahkonen et al., 2015; Alfano et al., 2016). vMEC with networking gateways that integrates with open source software and hardware platform to provide NFV is shown in Fig. 2. The virtualization manager supports a flexible and efficient, multi-tenancy, and hosting environment for applications by providing Infrastructure as a Service (IaaS) facilities as an IoT gateway.

Two applications of ETSI MEC for Traffic Offloading Function (TOF) and Radio Network Information Service (RNIS) are establish in the Linux embedded system. The two main incentives for ultra-low latency and high-availability of IoT application platforms are as follow (Dusia et al., 2015; Vilaltaet al, 2017; Wang et al., 2017):

- TOF with traffic priority judgment and path selection can direct custom authorization. TOF support the following two application service modes, as shown in Fig. 3:
  - Through Mode: the uplink and/or downlink U-plane traffic is delivered to an application which can monitor, change or shape it and then direct back to the initial Packet Data Network (PDN)

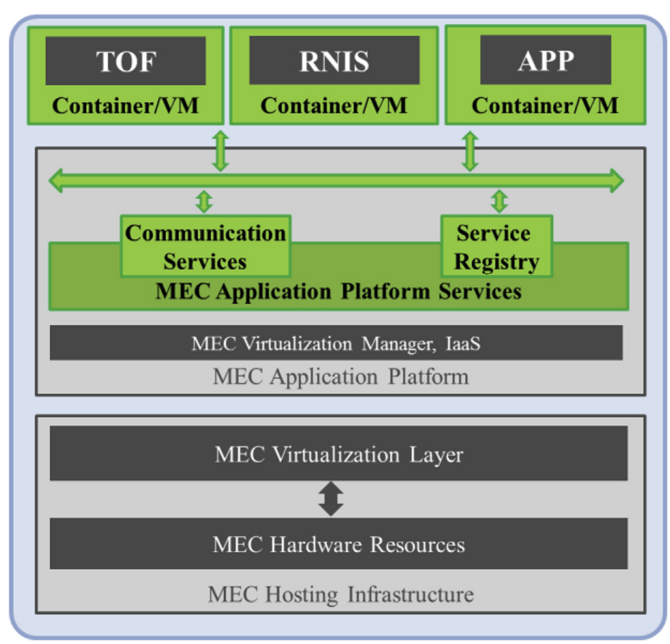


Fig. 2. Proposed ETSI-compliant vMEC architecture.

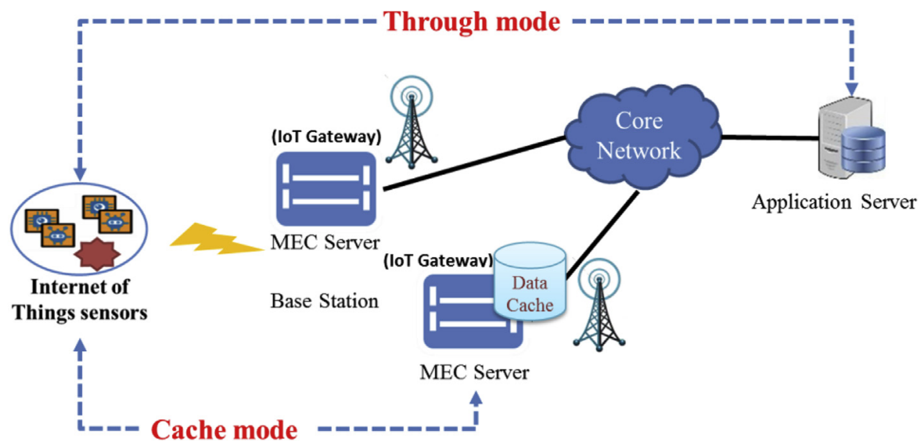


Fig. 3. Traffic offloading modes.

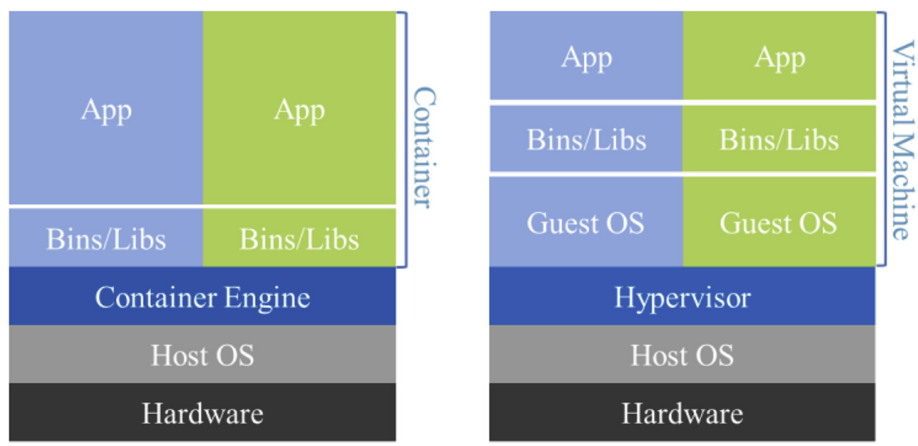


Fig. 4. Comparison of CVT and traditional VM technology.

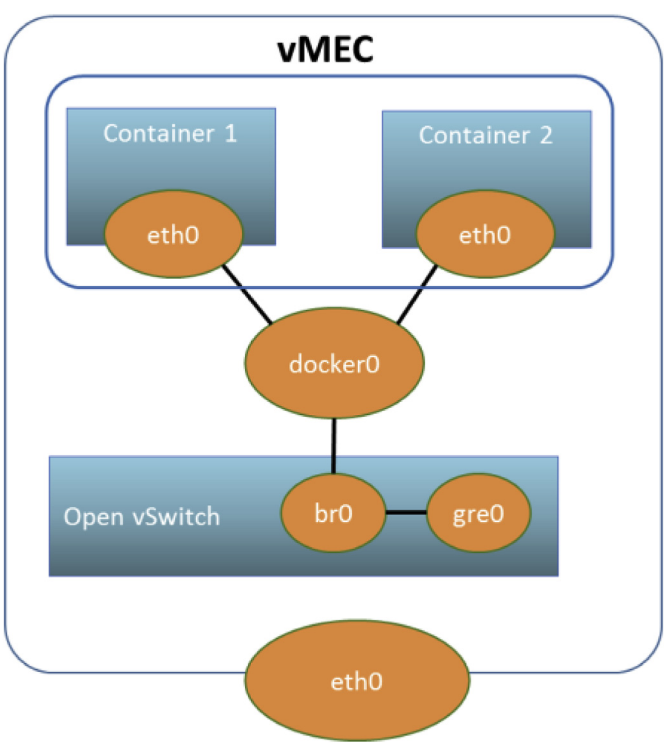


Fig. 5. The OVS as part of virtualized architecture.

connection.

- Cache Mode: the data flows computed by the internal virtualized application server can effectively back-up resources from core network and application servers to knowingly truncate the end to end latency time.

- RNIS obtains the underlying radio network information - cell-ID, user location, cell load status and wireless signal strength - between users and base stations to shorten policy decision time and improve platform utilization. The RNIS is a key feature, as it enables vMEC Applications to adjust the data transmission capacity for a particular user flow based on radio network information.

2.2. Container-based Virtualization Technology

Network Function Virtualization (NFV) drives lightweight application-centric with Container-based Virtualization Technology (CVT) is a lightweight virtualization technology based on container VNFs, and it used in an isolated space, running on the same Host Operating System (HOS), so the image capacity is less than that of Virtual Machine (VM) architecture. And then, it is easier to achieve the deployment and significant savings in computing resources, such that fewer are used. Traditional virtualization technology through the operating system layer is separated, as shown in Fig. 4, to exploit implementation of independent execution environment of the entire operating system, commonly referred to as a VM (Mao et al., 2016; Li et al., 2017; Simmhan et al., 2013). CVT is directly to code application that required libraries, environment configuration files are packaged together to establish an independent execution environment, which is produced independently of the application container technology. The HOS module

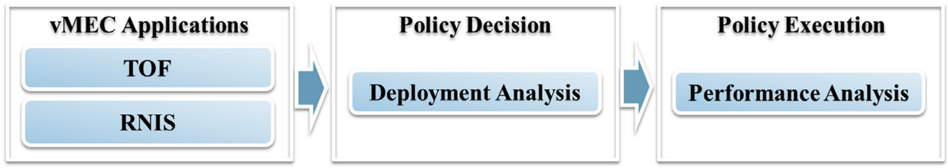


Fig. 6. Flow control mechanism.

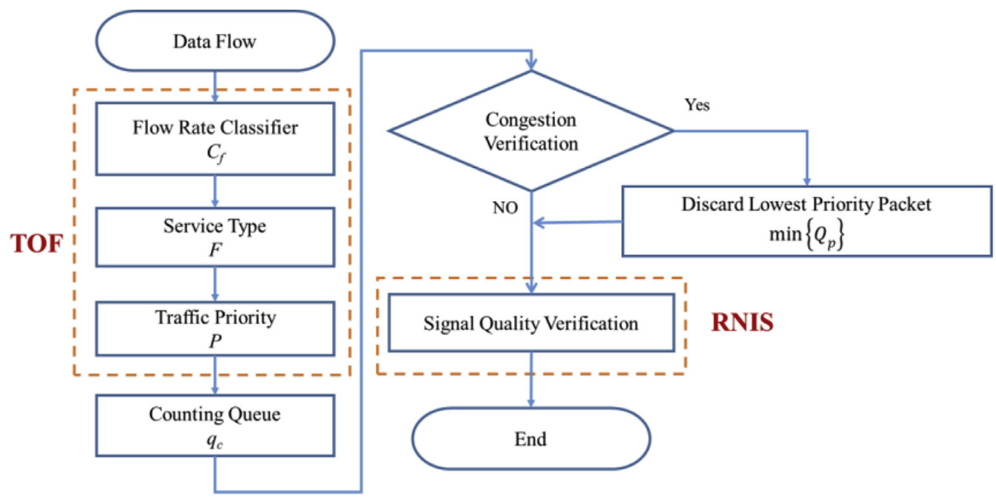


Fig. 7. Flow control procedure chart.

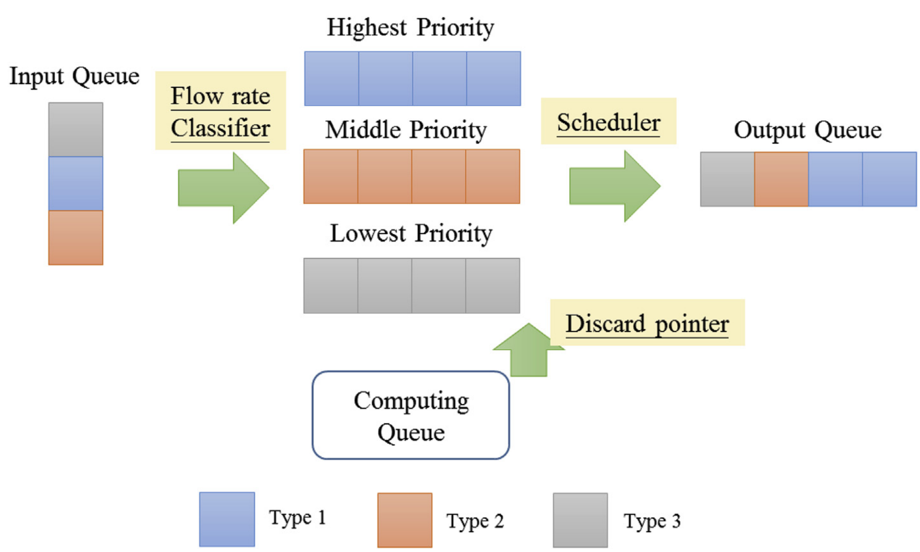


Fig. 8. Traffic processing architecture.

of CVT does not have to perform the Guest Operating System (GOS) installation that deployed in VM, thus establishing container without waiting for the boot time, it is only a few seconds to enable and faster than the traditional VM. Following the lightweight structure, its image file is smaller and more easily transmitted. In CVT, each container is isolated and running on the same HOS, and with independent virtual network interface (Perng et al., 2013; Papagianni et al., 2013; Preeth et al., 2015).

2.3. Software Defined Networking

Software Defined Networking (SDN) is a Linux-based bridging alternative accelerated by Open vSwitch (OVS) solution. Its multilayer software switching programming can be extended by large-scale network automation (Dong et al., 2013). OVS incorporates SDN-enabled virtual switches, which connect applications, are fully compatible with OpenFlow, integrate with SDN solutions, and are easy to manage (Sarzyniec et al., 2013). OVS performance in terms of throughput for smaller packets is must less than of line rate of the interface. To overcome this limitation, OVS has been ported to Data Plane Development Kit (DPDK) (Shanmugalingam et al., 2016). The virtualized architecture of CVT with the OVS network configuration is shown in Fig. 5. By the flexible allocation of resources, OVS can be programmed networking routers with high resource utilization using virtual technology, thereby reducing the cost of IoT applications. It also runs as a software switch in the management of routing program and is deployed directly to the IoT devices as a hardware control layer (Liu et al., 2011). OVS also supports a range of standard management interfaces, such as Netflow, Sampled flow (sFlow), Remote SPAN (RSPAN), Encapsulated RSPAN (ERSPAN), Command Line Interface (CLI) and others (Suárez-Varela and Barlet-

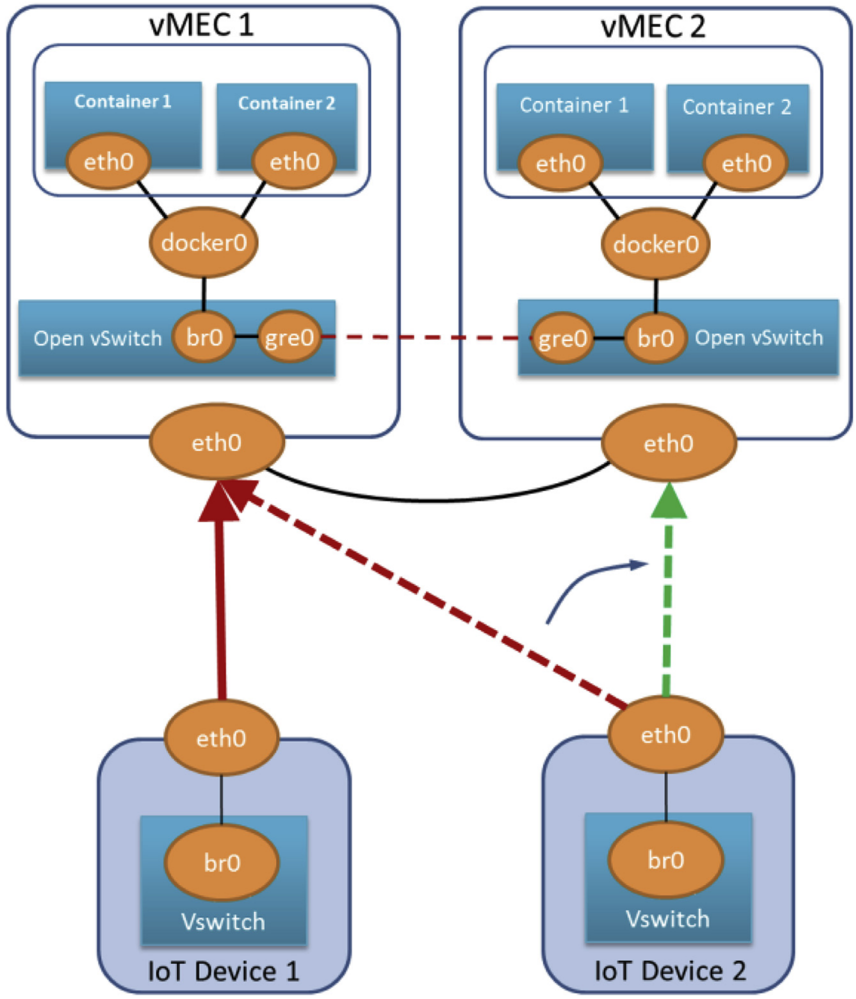


Fig. 9. Proposed vMEC with OVS platform.

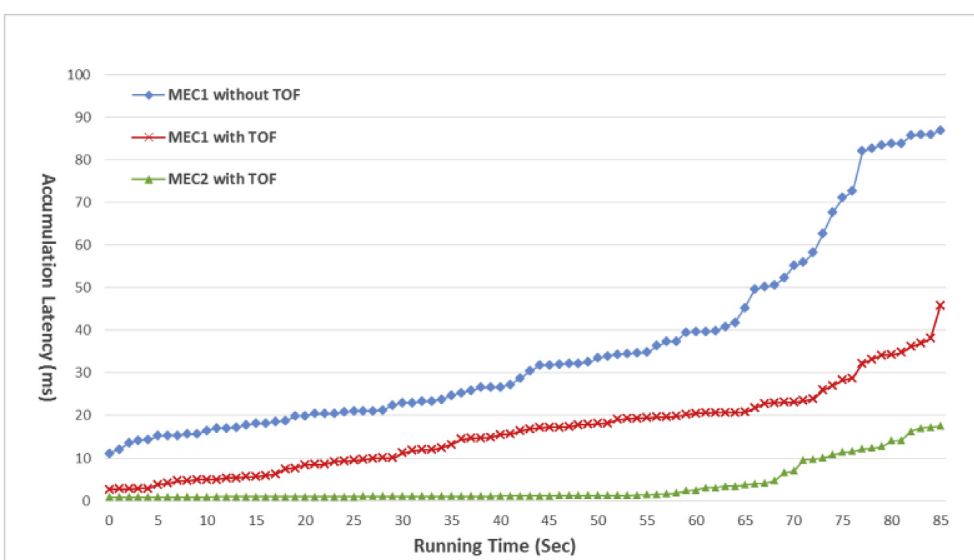


Fig. 10. Through mode with TOF function.

Ros, 2017).

3. System modeling

Following the above discussion, flow-based traffic engineering technique can be applied to system modeling. The flow control

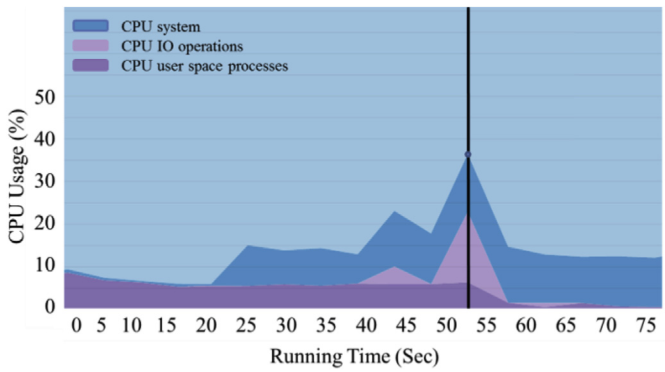


Fig. 11. CPU Utilization of vMEC Platform.

mechanism has four functions: TOF, RNIS, Policy Decision and Policy Execution as shown in Fig. 6. Analysis of networking environmental information and loading rate of IoT gateway will deliver the Policy Decision function for traffic offloading deployment and Policy Execution function for performance analysis. In effect, the analysis will significantly reduce the phenomenon of network congestion, latency, and improve Quality of Experience (QoE).

Internet of Things (IoT) Networking platform achieved from sensor nodes, and the IoT gateway, and to cloud server can deliver IoT networking applications and analysis of the flow control mechanism (Mach and Becvar, 2017). In Fig. 7, the IoT gateway integrates with TOF and RNIS functions, while following the wireless network status as weight table for determination of drop function to provide QoS and packet data set to conduct by triage mechanism. Flow control mechanism conducts TOF process by first obtaining IoT traffic information A, which contains data type F, traffic priority P and number of packets μ, in order to determine TOF traffic schedule and combined processing load of IoT gateway.

The length of flow rate counting vector is correlated to the TCP background flow. Fig. 8 indicates n background flows, in which

$$\text{Service Type} = 1,\quad \text{Text Packet} = 1 \ldots \mathrm{n}_1,\quad \text{Traffic Pri.} = \text{High},\quad \mu_1 = 1 \tag{1}$$

$$\text{Service Type} = 2,\quad \text{Voice Packet} = 1 \ldots \mathrm{n}_2,\quad \text{Traffic Pri.} = \text{Med.},\ \mu_2 = 2 \tag{2}$$

$$\text{Service Type} = 3,\quad \text{Video Packet} = 1 \ldots \mathrm{n}_3,\quad \text{Traffic Pri.} = \text{Low},\quad \mu_3 = 3 \tag{3}$$

In this program, counting matrix of flow rate C_f, counting queue $\mathrm{q_c}$, weight rate W, counting rate of flow which every packet in the queue r_i, set of Service type A, number of packet B_i, set of corresponding every flow types and discard pointer d_p.

$$C_f = \left\{ C_{f_{ij}} \middle| \mathrm{i} \varepsilon \mathrm{A},\ \mathrm{j} \varepsilon B_i, \text{ Service type i has } B_i \text{ Packet} \right\} \tag{4}$$

$$q_c = \{ r_{ij} | \mathrm{i} \varepsilon \mathrm{A},\ \mathrm{j} \varepsilon B_i, \text{ Service type i has } B_i \text{ Packet} \} \tag{5}$$

If we suppose discard pointer d_p indicates packet loss of packet of type 3, then

$$W(i,\ j) = \mu_i \tag{6}$$

$$d_p \to \{ Packet(i, j) | \min W(i, j),\ \mathrm{i} \varepsilon \mathrm{A},\ \mathrm{j} \varepsilon B_i \} \tag{7}$$

$$\min W(i, j) = 1,\ d_p = \{ Packet_{n3} \},\ \text{packet of type 3} \tag{8}$$

Detecting the wireless network interface of the IoT gateway, the set of connected wireless sessions are N, the connections status is f, the latency time *is* L. The network weight value W is calculated from f to obtain the signal strength level. By using wireless signal strength to determine whether the flow control application should be implemented, if the wireless signal strength is at its lowest, the lowest priority packet $\min\{\mathrm{Q_p}\}$ will be discarded that time cost D.

Flow Control Mechanism, as defined in Fig. 6, is obtained from traffic information on TOF process, which contains the data type F, packet quantity μ and priority P for traffic classification, in order to determine whether the current network is congested. In which, TOF $\coprod$ RNIS $\Rightarrow f_{ToF} \coprod g_{RSSI}$, the two functions are independent.

$$f_{ToF}(X) = \begin{cases} P_{ToF}, & \mathrm{X} = 1,\ \text{Congestion} \\ 1 - P_{ToF}, & \mathrm{X} = 0,\ \text{Un} - \text{Congestion},\ \text{Scan in } \mathrm{t_1} \end{cases} \tag{9}$$

$$L_{ToF}(X) = \begin{cases} 0, & \mathrm{X} = 1 \\ {}^{t_1}\!/_{v_1}, & \mathrm{X} = 0,\ {}^{t_1}\!/_{v_1} \text{ is congestion time cost under } \mathrm{t_1} \end{cases} \tag{10}$$

If we suppose the network is congested, the lowest priority packet $\min\{Q_p\}$ is selected. RNIS will proceed to confirm all wireless network connection N and determine the wireless network status. If the wireless network status is lower than the minimum wireless signal strength, it will execute the flow control mechanism after re-confirming the wireless network status.

$$g_{RSSI}(Y) = \begin{cases} P_{RSSI}, & RSSI \geq \text{minimum signal quality},\ \mathrm{Y} = 1 \\ 1 - P_{RSSI}, & RSSI < \text{minimum signal quality},\ \mathrm{Y} = 0 \end{cases} \tag{11}$$

$$H_{RSSI}(Y) = \begin{cases} 0, & Y = 0 \\ {}^{t_2}\!/_{v_2}, & Y = 1,\ {}^{t_2}\!/_{v_2} \text{ is congestion time cost under } \mathrm{t_2} \end{cases} \tag{12}$$

$$g'_{RNIS}(Z) = \begin{cases} P_{RSSI},\ Z = 0 \Rightarrow g_{RSSI\,(Y=1)} \\ (1 - P_{RSSI})(P_{RSSI}),\ Z = 1 \Rightarrow g_{RSSI\,(Y=0)} \times g_{RSSI\,(Y=1)} \\ (1 - P_{RSSI})^2,\ Z = 2 \Rightarrow g^2_{RSSI\,(X=0)} \end{cases} \tag{13}$$

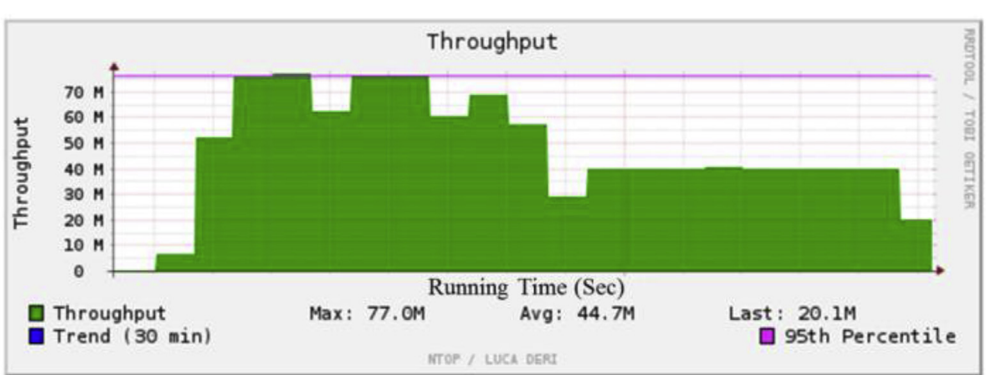


Fig. 12. Throughput rate.

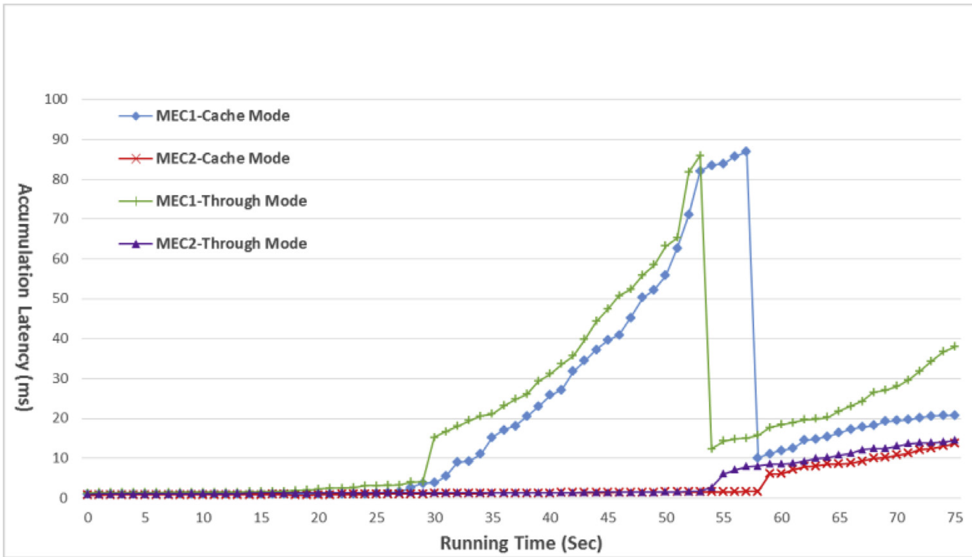


Fig. 13. Traffic offloading modes comparison.

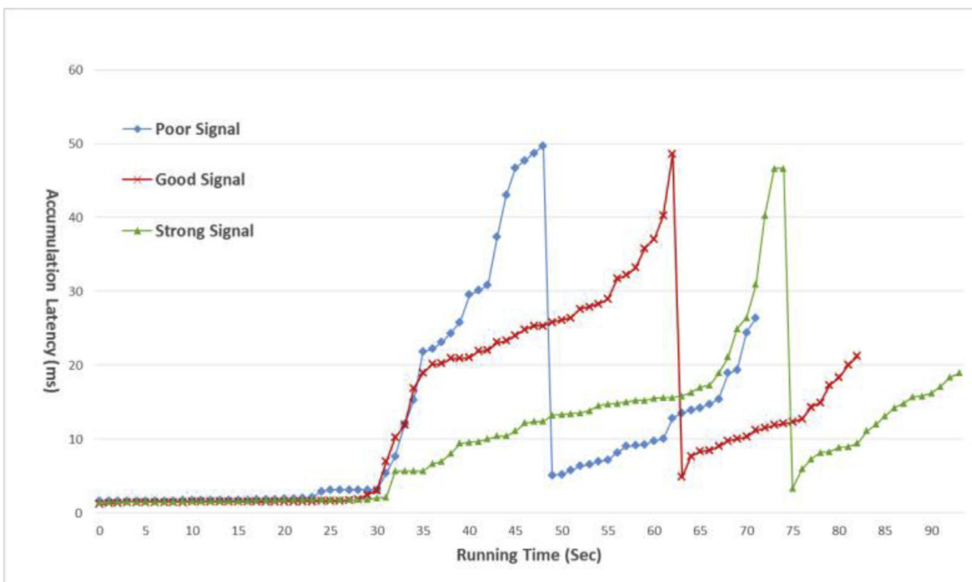


Fig. 14. Through mode-with ToF and RNIS.

$$H'_{RNIS}(X,Z)=\begin{cases}0, & X=0,\ Z=0\\ {}^{t_2}/_{v_2}, & X=0,\ Z=1\\ {}^{t_2}/_{v_2}, & X=1,\ Z=1\\ 0, & X=1,\ Z=2\end{cases} \tag{14}$$

$$D(X,Z)=\begin{cases}0, & X=0,\ Z=0\\ 0, & X=1,\ Z=0\\ 0, & X=1,\ Z=1\\ t_3, & X=1,\ Z=2\end{cases} \tag{15}$$

4. Implementation and analysis

This section delivers some considerations and background information associated with the proposed vMEC platforms with CVT for IoT applications.

4.1. Implementation

The proposed ETSI MEC established with a network virtualized platform has two major functions, which are TOF for session control and RNIS for wireless network monitoring. The virtualized platform with TOF and RNIS functions is integrated as flow control mechanism, which offers more flexibility and deployability as shown in Fig. 5. The proposed deployment of resource scheduling contributes to the service mode and traffic control strategies of applications, which improve the delivery networking latency and user experience. The IoT gateway that is proposed herein aims to achieve resource computing based on NFV architecture and create TOF and RNIS applications as services that are implemented by TOF for traffic offloading and efficient utilization of the wireless signal strength through RNIS.

4.2. Performance analysis

This section adopts IoT scenarios based on vMEC platforms for flow control mechanism and performance analysis. The open source code of

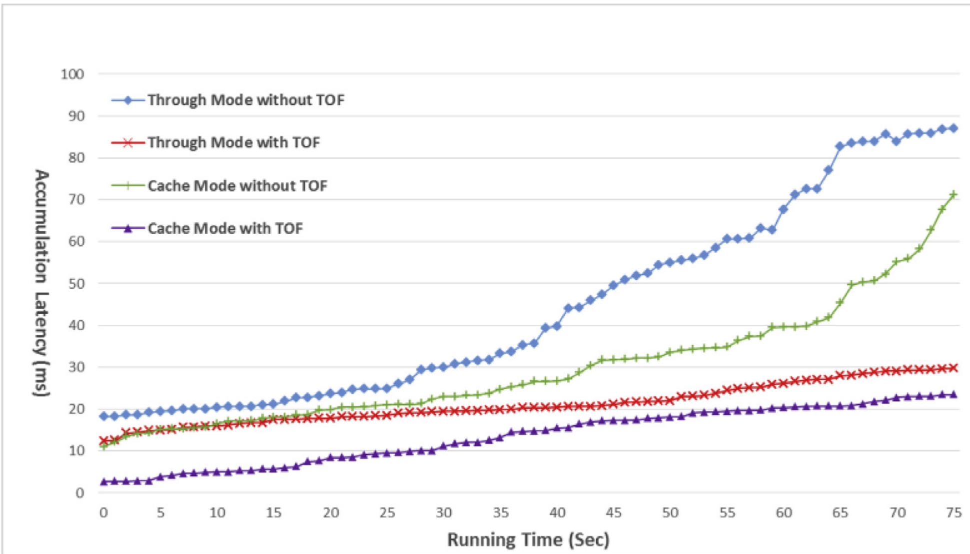


Fig. 15. Latency of Through Mode vs Cache Mode.

vMEC and hardware platform are rapidly built through NFV platform with CVT. Implementation of the flow control mechanism on an IoT gateway is expected to reduce the phenomenon of network congestion and latency of applications, and latency measurements achieved with the Iperf network monitoring tool. The proposed vMEC with OVS platforms for traffic control mechanism in Through Mode and Cache Mode, TOF and RNIS service applications, and OVS for flow-based switching are shown in Fig. 9. While the control mechanisms is implemented on the proposed vMEC platforms and the network throughput rate in full load or over warning threshold, then the flow control mechanisms will execute and OVS enabled to switchover the application sessions from vMEC1 to vMEC2.

Fig. 10 displays result of the pre-setting IoT applications on proposed vMEC platforms in application sessions full load. Voice, data and video sessions in Through Mode with and without the flow control mechanism are compared. After the flow control mechanism is performed, the latency exchange rate drops from 90 ms to 45 ms. It is obvious that the proposed vMEC platforms with flow control function are effectively reducing latency. Fig. 11 indicates CPU utilization of the proposed vMEC platforms that is caused by enhanced application sessions. The flow control mechanism contributes to the reduction of the CPU utilization and measures the networking loading rate.

Fig. 12 shows the network throughput before and after implementation of the flow control mechanism. When the master vMEC platform is fully loaded, the throughput of network interface is as high as up to 77 Mbps. When the flow control mechanism is implemented, the network throughput is downgraded; indicating that the application sessions is not congested after enabling the OVS function to share the load to slave vMEC platform, as the throughput rate becomes 41 Mbps. As shown in Fig. 13, the proposed vMEC platforms implements on Cache Mode, while the control mechanisms is not implemented and the network throughput rate in full load or over warning threshold, result in latency is up to 80 ms. After control mechanisms is implemented and OVS enabled to switchover the sessions from vMEC1 to vMEC2, the latency is reduced to 19 ms.

Next, the TOF is integrated with RNIS to speed up the response time of flow control with poor wireless signal strength. As indicated in Fig. 14, the analysis of diverse wireless signal strength using network interface yields the blue curve as the strong signal, the red curve as the good signal, and the green curve as the poor signal. Once traffic loads are implemented to the poor signal after 30 s, the latency increases rapidly. After flow control mechanism is implemented on the proposed vMEC platforms to latency more than 25 ms at 36 s, the signal strength is confirmed poor and the proposed vMEC platforms will drop session within 5 s. The latency will continue to increase until the flow control mechanism detects network congestion at 43 s, where it will drop sessions at 48 s.

As shown in Fig. 15, implementation of Cache Mode and Through Mode is initially, and the maximum difference of latency is at 30 ms. Flow control mechanism is then implemented and the latency falls to 10 ms, indicating that Cache Mode outperforms Through Mode.

5. Conclusion

In conjunction with SDN and NFV, vMEC plays significant role in effecting low latency, high bandwidth and trillions of devices worldwide. In this study, vMEC standard architecture proposed by ETSI to achieve establishment of configuration management consistent with NFV platform next-generation 5G network demand, provision of increased strengthening IoT application platform, and implementation of flow control mechanism. Different service configurations were used in different applicational situations. A better platform with greater flexibility and deployability is thus obtained. The traffic control strategies proposed in this study effectively reduce IoT delay, improve QoE and reduce network congestion with IoT application service, while using different split mode.

Open source software and a hardware platform were used herein to build a network virtualization platform. The easy deployment features of app-centric CVT is then used to improve configuration management application service platform and allows for swift establishment of networking application services on a virtualized platform. The flow control mechanism proposed in this study, in the case of application services in full, applies various different traffic control strategies to varying services and control strategies, in order to reduce the latency of IoT application services by an average of 30% and improve service quality.

Acknowledgement

This paper is a partial result of project no. MOST 107-2911-I-011-503 (Project Title: New Information and Communications Technologies for the Intelligent Transportation Systems and the Internet of Things) conducted by National Taiwan University of Science and Technology under the sponsorship of the National Science Council, ROC.

References

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M., 2015. Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials 17 (4), 2347–2376 Taiwan.

Alfano, G., Garetto, M., Leonardi, E., Feb. 2016. Content-centric wireless networks with limited buffers: when mobility hurts. IEEE/ACM Trans. Netw. 24 (1), 299–311.

Cau, E., Corici, M., Bellavista, P., Foschini, L., Carella, G., Edmonds, A., Bohnert, T.M., 2016. Efficient exploitation of mobile edge computing for virtualized 5G in EPC architectures. In: Proceedings of the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (IEEE MobileCloud 2016), pp. 100–109.

Dong, Y., Yun, Y., Xiao, L., Wenhao, L., Lizhen, C., Meng, X., Jinjun, C., 2013. A highly practical approach toward achieving minimum data sets storage cost in the cloud. IEEE Trans. Parallel Distr. Syst. 1234–1244.

Dusia, A., Yang, Y., Taufer, M., 2015. Network quality of service in docker containers. In: Proceedings of the IEEE International Conference on Cluster Computing (IEEE CLUSTER 2015), pp. 527–528.

Habib, S.J., Marimuthu, P.N., Sept. 2011. Data aggregation at the gateways through sensors' tasks scheduling in wireless sensor networks. Wireless Sensor Syst. 171–178 IET.

Ismail, B.I., Goortani, E.M., Karim, M.B.A., Tat, W.M., Setapa, S., Luke, J.Y., Hoe, O.H., 2015. Evaluation of docker as edge computing platform. In: Proceedings of the IEEE International Conference on Open Systems (IEEE ICOS 2015), pp. 130–135.

Iwamura, M., 2015. NGMN view on 5G architecture. In: Proceedings of the 82nd IEEE International Conference on IEEE Vehicular Technology (IEEE VTS 2015), pp. 1–5.

Jararweh, Y., Doulat, A., AlQudah, O., Ahmed, E., Al-Ayyoub, M., Benkhelifa, E., 2016. The future of mobile cloud computing: integrating cloudlets and mobile edge computing. In: Proceedings of the 23rd IEEE International Conference on Telecommunications (IEEE ICT 2016), pp. 1–5 Thessaloniki.

Joy, A.M., 2015. Performance comparison between linux containers and virtual machines. In: Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications (IEEE ICACEA 2015), pp. 342–346.

Li, H., Shou, G., Hu, Y., Guo, Z., 2016. Mobile edge computing: progress and challenges. In: Proceedings of the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (IEEE MobileCloud 2016), pp. 83–84 Oxford.

Li, S., Yu, Q., Maddah-Ali, M.A., Avestimehr, A.S., Oct. 2017. A scalable framework for wireless distributed computing. IEEE/ACM Trans. Netw. 25 (5), 2643–2654.

Liu, C.Y., Chuang, C.L., Chen, C.P., Chang, W.Y., Shieh, J.C., Lin, C.-H., Tseng, C.-L., Jiang, J.A., Nov. 2011. Development of an embedded system-based gateway for environmental monitoring using wireless sensor network technology. In: Proceeding of the 5th IEEE International Conference on Sensing Technology (IEEE ICST 2011), pp. 544–548.

Lyu, X., et al., Nov. 2017. Optimal schedule of mobile edge computing for internet of things using partial information. IEEE J. Sel. Area. Commun. 35 (11), 2606–2615.

Mach, P., Becvar, Z., Aug. 2017. Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutorials 19 (3), 1628–1656.

Mahkonen, H., Manghirmalani, R., Shirazipour, M., Xia, M., Takacs, A., 2015. Elastic network monitoring with virtual probes. In: Proceedings of the 1st IEEE Conference on Network Function Virtualization and Software Defined Network (IEEE NFV-SDN 2015), pp. 1–3.

Mao, Y., Zhang, J., Letaief, K.B., Dec. 2016. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J. Sel. Area. Commun. 34 (12), 3590–3605.

Moldován, I., Varga, P., 2012. A flexible switch-router with reconfigurable forwarding and linux-based control element. In: Proceedings of the 10th IEEE International Symposium on Electronics and Telecommunications, (IEEE ISETC 2012), pp. 217–220 Timisoara.

Nunna, S., et al., 2015. Enabling real-time context-aware collaboration through 5G and mobile edge computing. In: Proceedings of the 12th IEEE International Conference on Information Technology - New Generations (IEEE ITNG 2015), pp. 601–605 Las Vegas, NV.

Papagianni, C., Leivadeas, A., Papavassiliou, S., Maglaris, V., Cervello-Pastor, C., Monje, A., 2013. On the optimal allocation of virtual resources in cloud computing networks. IEEE Trans. Comput. 62 (6), 1060–1071.

Perng, C., Li, T., Chang, R., 2013. Cloud analytics for capacity planning and instant VM provisioning. IEEE Trans. Netw. Serv. Manag. 1–14.

Pfaff, B., Pettit, J., Koponen, T., Jackson, E., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., Amidon, K., Casado, M., 2015. The design and implementation of open vSwitch. In: Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation (NSDI 2015), pp. 117–130.

Preeth, E.N., Mulerickal, Fr J.P., Paul, B., Sastri, Y., Mar. 2015. Evaluation of docker containers based on hardware utilization. In: Proceedings of the 3rd IEEE International Conference on Control, Communication & Computing (IEEE ICCC 2015), pp. 697–700.

Raafat, H.M., et al., 2017. Fog intelligence for real-time IoT sensor data analytics. IEEE Access 5, 24062–24069.

Ratasuk, R., Prasad, A., Li, Z., Ghosh, A., Uusitalo, M.A., 2015. Recent advancements in M2M communications in 4G networks and evolution towards 5G. In: Proceedings of the 8th IEEE International Conference on Intelligence in Next Generation Networks (IEEE ICIN 2015), pp. 52–57.

Salman, O., Elhajj, I., Kayssi, A., Chehab, A., 2015. Edge computing enabling the internet of things. In: Proceedings of the 2nd IEEE International Conference on World Forum on Internet of Things (IEEE WF-IoT 2015), pp. 603–608.

Sarzyniec, L., Buchert, T., Jeanvoine, E., Nussbaum, L., 2013. Design and evaluation of a virtual experimental environment for distributed systems. In: Proceedings of the 21th IEEE Euromicro International Conference on Parallel, Distributed and Network-based Processing (IEEE PDP 2013), pp. 172–179 Belfast.

Shanmugalingam, S., Ksentini, A., Bertin, P., Jun. 2016. DPDK Open vSwitch performance validation with mirroring feature. In: Proceeding of the 23rd IEEE International Conference on Telecommunications (IEEE ICT 2016), pp. 1–6 Thessaloniki.

Simmhan, Y., Prasanna, V., Aman, S., Kumbhare, A., Liu, R., Stevens, S., Zhao, Q., July-Aug. 2013. Cloud-based software platform for big data analytics in smart grids. IEEE Comput. Sci. Eng. 15 (4), 38–47.

Suárez-Varela, J., Barlet-Ros, P., Oct. 2017. Towards a netflow implementation for openflow software-defined networks. In: Proceeding of the 29th IEEE International Conference on Teletraffic Congress (IEEE ITC 2017), pp. 187–195 Italy.

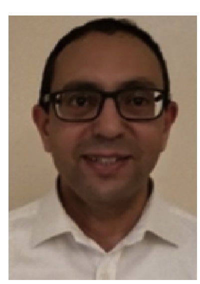
Vilalta, R., et al., 2017. TelcoFog: a unified flexible fog and cloud computing architecture for 5G networks. IEEE Commun. Mag. 55 (8), 36–43.

Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., Wang, W., 2017. A survey on mobile edge networks: convergence of computing, caching and communications. IEEE Access 5, 6757–6779.

Xiao, L., Li, Y., Huang, X., Du, X., Oct. 2017. Cloud-based malware detection game for mobile devices with offloading. IEEE Trans. Mobile Comput. 16 (10), 2742–2750.

Han-Chuan Hsieh was received a B.S. degree in Electrical Engineering from National Taipei University of Technology (NTUT), in 1998, and an M.S. degree in Communication Engineering from Tatung Institute of Technology, Taipei, Taiwan, in 2008. He received his Ph.D. degree in Department of Electrical Engineering of National Taiwan University of Science and Technology (NTUST) in 2018. His major interests are in Long Term Evolution-Advanced, Internet of Things, Software Defined Networking and Network Functions Virtualization in 5G.

Jiann-Liang Chen received the Ph.D. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 1989. Since August 1997, he has been with the Department of Computer Science and Information Engineering of National Dong Hwa University, where he is a professor and Vice Dean of Science and Engineering College. Prof. Chen joins the Department of Electrical Engineering, National Taiwan University of Science and Technology, as a full professor now. His current research interests are directed at cellular mobility management, digital home network, telematics applications, cloud computing and RFID middleware design. Prof. Chen is an IEEE Senior Member and UK BCS Fellow. He has published more than 150 papers in journals and conferences, and also holds several patents.



Abderrahim Benslimane is Professor of Computer-Science at the Avignon University/France since 2001. He has been recently a Technical International Expert at the French Ministry of Foreign and European affairs (2012–2016). He served as a coordinator of the Faculty of Engineering at the French University in Egypt. He obtained the French award of Doctoral supervisions (2017–2021) and he was attributed the French award of Scientific Excellency (2011–2014). His research interests are in development of communication protocols with the use of graph theory for mobile and wireless networks. Currently, he is working on the conception and performance evaluation of protocols for security in Internet of Things. Also, he works on multicast routing, inter-vehicular communications, Quality of service, energy conservation, localization, intrusion detection and MAC layer performance evaluation. He has more than 140 refereed international publications (book, journals and conferences) in all those domains.