Integration of IoT, Transport SDN and Edge/Cloud computing for Dynamic Distribution of IoT Analytics and Efficient Use of Network Resources

Raul Muñoz, *Senior, IEEE*, Ricard Vilalta, *Senior, IEEE*, Noboru Yoshikane, Ramon Casellas, *Senior, IEEE*, Ricardo Martínez, *Senior, IEEE*, Takehiro Tsuritani, Itsuro Morita.

***Abstract*— IoT requires cloud infrastructures for data analysis (e.g., temperature monitoring, energy consumption measurement, etc.). Traditionally, cloud services have been implemented in large datacentres in the core network. Core cloud offers high-computational capacity with moderate response time, meeting the requirements of centralized services with low-delay demands. However, collecting information and bringing it into one core cloud infrastructure is not a long-term scalable solution, particularly as the volume of IoT devices and data is forecasted to explode. A scalable and efficient solution, both at the network and cloud level, is to distribute the IoT analytics between the core cloud and the edge of the network (e.g. first analytics on the edge cloud and the big data analytics on the core cloud). For an efficient distribution of IoT analytics and use of network resources, it requires to integrate the control of the transport networks (packet and optical) with the distributed edge and cloud resources in order to deploy dynamic and efficient IoT services. This paper presents and experimentally validates the first IoT-aware multi-layer (packet/optical) transport SDN and edge/cloud orchestration architecture that deploys an IoT-traffic control and congestion avoidance mechanism for dynamic distribution of IoT processing to the edge of the network (i.e., edge computing) based on the actual network resource state.**

***Index Terms*—IoT analytics, edge computing, transport SDN, cloud orchestration**

I. Introduction

It is envisioned that the Internet of Things (IoT) will connect billions of heterogeneous devices (ranging from complex interactive systems to tiny sensors) to the transport network using widely deployed wireless access technologies (e.g., Bluetooth, ZigBee, Wi-Fi, LoRa), mobile technologies (e.g., GPRS/2G or eMTC and NB-IoT) or fixed access technologies (e.g., PLC, ADSL, optical Access, Ethernet). It will facilitate a wide variety of use cases from different vertical industries, such as automotive and mobility, healthcare, energy, media and entertainment, factories of the future, or energy.

Much like the fifth generation of mobile technology (5G), IoT also requires cloud computing and storage datacentres (DC) in order to perform IoT analytics from the data collected of sensors and actuators. For example, Siemens wind power develops machine learning to predict and diagnose potential problems in 9,000 wind turbines with 400 sensors each, sending data several times a second [1]. Another example is the connected car, that is expected to send 25 gigabytes of data (each car) to the cloud every hour, collecting telematics and driver behavior data to keep the vehicle's performance, efficiency, and safety in check [32]. However, collecting information from multiple sensors and analysing it using a relatively small number of core DCs does not scale, particularly as the number of IoT devices and volume of data is forecasted to explode [2].

On the transport network side, the generation of large number of flows (e.g., telematics, sensors) or huge aggregated volumes of data (e.g., remote monitoring, digital signage) from the edge of the network to the core DCs could congest the network. It will also be costly because transporting bits from the edge to core network actually costs money. Additionally, some mission-critical IoT applications with very stringent delay requirements may also require to perform IoT analytics in the edge in order to perform real-time actions (e.g., remote monitoring and telematics).

A solution recently proposed to address the new IoT requirements such as dense high traffic processing, large number of connections processing, peak traffic processing and low-latency processing is to distribute the processing of the IoT analytics from the core DC to micro-DCs and small-DCs at the network edge (edge computing), known as edge IoT analytics [3]. First analytics can be carried out on the edge cloud and only the necessary data or results are sent for further analysis (e.g.,

Part of this work has been performed in the framework of the H2020 project 5GCAR co-funded by the EU. Authors would like to acknowledge the contributions of their colleagues from 5GCAR although the views expressed are those of the authors and do not necessarily represent the views of the 5GCAR project. This research also has been partly funded by Spanish MINECO projects DESTELLO (TEC2015-69256-R).

This submission is an extended manuscript related to the ECOC 2017 paper [15]

R. Muñoz, R. Vilalta, R. Casellas, and R. Martínez are with the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Spain (e-mail: raul.munoz@cttc.es).

N. Yoshikane, T. Tsuritani, and I. Morita are with KDDI Research, Inc., Saitama, Japan (e-mail: yoshikane@kddi-research.jp).

big data) or storage in the core DC. Thus, the distribution of IoT analytics offloads the network and the DCs by creating a model that scales and releases bottlenecks. Thus, distributed lightweight computing resources (e.g., to fit in street cabinets), are needed in different network locations, closer to the network edge. It is in this sense that edge computing nodes (also known as fog nodes) are necessary in order to reduce the impact of the network latency and reduce bandwidth usage, as data is processed where it is generated, reducing the necessity of bulk data transmissions. Moreover, for an efficient distribution of IoT analytics and use of network resources it requires a tight coordination between the IoT analytics platform, the Transport SDN network and the cloud infrastructure.

This paper presents and experimentally validates the first IoT-aware multi-layer (packet/optical) transport SDN and cloud orchestration architecture that deploys an IoT-traffic control and congestion avoidance mechanism for dynamic distribution of IoT processing to the edge of the network (i.e., edge computing) based on the actual network resource state. Moreover, we also present and experimentally assess an edge node with cloudlet based on SDN-enabled containers (CT) for seamless integration with the IoT-aware SDN and cloud orchestration platform. In order to demonstrate the advantages of the proposed architecture, we also consider the use case of video analytics on a Close Circuit TV (CCTV) running at the edge of the network.

Specifically, in Sec. II we present the related work on IoT analytics and integration with the transport networks. Sec. III provides and overview of the requirements for IoT services and possible solutions. Sec IV presents the considered SDN-enabled edge node with cloudlet based on container technologies. Section V depicts the proposed IoT-aware SDN and cloud architecture for an efficient distribution of IoT analytics and use of network resources. Sec. VI presents the workflow for the provisioning of IoT analytics services, congestion detection and distribution of IoT analytics. Experimental validation and assessment of the proposed edge node and control architecture have been carried out in Sec. VII. Two proof-of-concepts (PoCs) are presented; provisioning of core IoT analytics and distribution to the edge, and; provisioning of distributed IoT analytics for video. Finally, concluding remarks are provided in Sec. VIII.

II. Related work

IoT analytics have been largely studied in the literature, and in particular, edge IoT analytics have been comprehensively studied in the last years. For example, [21] presents a flexible architecture for IoT data analytics using the concept of edge computing that facilitates the automated transitions between edge and cloud depending on the dynamic conditions of the IoT infrastructure and application requirements. Similarly, [22] depicts an IoT platform that supports reliable and guaranteed data dissemination and analysis (both at the edge and cloud) for rural or remote areas where the wireless infrastructure is sparse and requires long-range multi-hop connectivity to remote sensors and actuators. Some other works are more focused on the specific IoT data analysis that is required to be performed in the edge in order to reduce the cloud storage requirement, the energy consumption. For example, [23] presents a fog-level IoT analytic system with an efficient knowledge extraction technique to reduce the amount of data transfer to the cloud and to help simplify the process. Similarly, [24] presents the general models and architecture of fog data stream processing and analytics, by analyzing the common properties of several typical applications, such as video mining and event monitoring, which are very popular in the cloud, but have not been comprehensively investigated in the context of fog architecture. Additionally, there are some other papers that apply similar IoT data analytics architectures to some specific use case such as smart cities [25], public transportation [26], e-health [27], smart cities [28].

However, the integration of the IoT analytics platforms with the optical transport networks (or in general with transport networks between the edge computing and the cloud) has not been properly addressed. All the above works are only focused on the sensor networks, and the edge and cloud computing infrastructure for performing the data analytics. The transport network connecting the edge computing with the cloud infrastructure is considered as a commodity providing static pipes. In [29] the authors presented for the first time an SDN/NFV-enabled edge node providing virtual machines for IoT applications that was integrated in the network and cloud orchestration platform provided by the ADRENALINE testbed [30]. In this work, the IoT gateway can dynamically request cloud or edge computing resources to the network and cloud orchestrator. Then, the orchestrator dynamically provisions a virtual machine in the cloud or edge node, and provisions the required connections between the virtual machines and the IoT gateway. More details about the orchestration of network and cloud resources, identifying the limitations of the existing implementations is presented in [7]. In [3], the authors extended the SDN/NFV-enabled edge node to support a new virtualization technology based on containers. The IoT applications running in the containers can trigger on-demand connectivity services to the cloud. This architecture was validated with the use case of MEC video analytics on a Close Circuit TV (CCTV). In [15], the authors presented for the first time the integration of an IoT analytics platform with the global network and cloud orchestrator. In the previous work, the services were directly triggered by the distributed IoT gateways or the IoT applications, but not by any centralized IoT analytics platform that manages all IoT services. This integration aims at performing an efficient use of network resources by providing an IoT-traffic control and congestion avoidance mechanism that enables the dynamic distribution of IoT processing to the edge of the network based on the monitoring of the actual IoT flows in the network. This paper extends the work carried out in [15], by providing the following additional contributions:

- Integration of the SDN/NFV edge nodes in the experimental validation of the proposed IoT-aware multi-layer (packet/optical) transport SDN and edge/cloud orchestration architecture. It enables to also manage containers in the edge of the network, in addition to the virtual machines provided by the small DCs considered in [15].

Requirements	Corresponding use cases	Possible Solutions
Low-latency	Real-time sensor, Remote monitoring etc.	Distributed data processing etc.
High speed traffic	Remote monitoring, Telematics etc.	Increase capacity of IoT slice Distributed data processing etc.
Large capacity traffic	Remote monitoring, Digital signage etc.	Increase capacity of IoT slice Distributed data processing etc.
Massive connections	Sensor, Telematics etc.	Increase capacity of IoT slice Distributed data processing etc.

Tab.1. SDN-enabled container-based architecture for the edge node

- Deployment of the real use case of video analytics running at the edge of the network to validate the provisioning of edge IoT analytics services through the proposed architecture, in addition to the core IoT analytics and distribution to the edge presented in [15].
- More elaborated description of the proposed IoT-aware network and cloud architecture, experimental scenarios, and further details of the joint setup.

III. Requirements for IoT services and possible solutions

Before the IoT era, in which communication was mainly done through human-to-human and human-to-machine, use cases of communication networks such as telephone call and e-mail were determinative and requirements for the networks such as data speed and data amount per device were basically fixed. In the IoT era, on the other hand, various kinds of devices are connected to the networks. Therefore, use cases and requirements for IoT services become diversified greatly. Table 1 summarizes network requirements for IoT services and possible solutions.

- As for low-latency, expected corresponding use cases of IoT services are real-time sensor, remote monitoring and so on. In this case, distributed data processing such as the proposed SDN-enabled container-based edge node would be effective for the requirement.
- Regarding high speed traffic, expected corresponding use cases of IoT services are remote monitoring, telematics and so on. In this case, possible solutions are capacity increase of IoT slice and distributed data processing.
- With respect to large capacity traffic, expected corresponding use cases of IoT services are remote monitoring, digital signage and so on. In this case, possible solutions are capacity increase of IoT slice and distributed data processing.
- About massive connections, expected corresponding use cases of IoT services are sensor, telematics and so on. In this case, possible solutions are the same as the high speed traffic and large capacity traffic processing cases (i.e. capacity increase of IoT slice and distributed data processing).

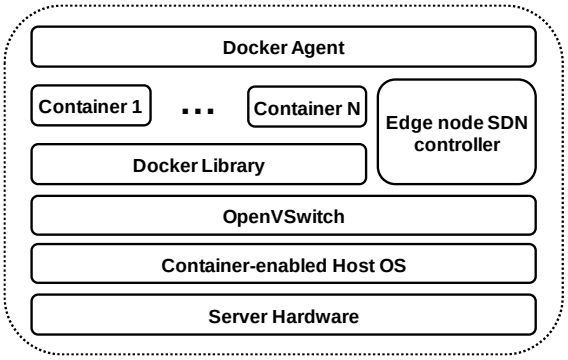

Fig.1. SDN-enabled container-based architecture for the edge node

Since the proposed SDN-enabled container-based edge node has the capability of distributed data processing, the proposed method can satisfy all the requirements for IoT services shown in table 1.

IV. SDN-enabled container-based edge node

As the compute resources at the edge are commonly limited, typically running in lightweight industrialized servers (e.g., to fit in street cabinets), we propose to use SDN-enabled Linux containers as the preferred compute virtualization solution for micro-DCs located at the network edge, thus reducing performance overhead below 5% [5]. Docker is a de-facto standard for container management. Its integration of container networking with SDN has been previously demonstrated [5]. Docker containers are able to run generalized lightweight applications, thus allowing to dynamically run on-demand virtual applications.

Fig.1. shows the proposed architecture of an SDN-enabled container-based edge node. On top of an industrialized server hardware, a container-enabled host operative system is executed. In order to offer the necessary internal and external connectivity (e.g., inside the edge node and towards the edge network) to the virtualized compute resources, a software switch (such as OpenVSwitch) is used. Docker Library and daemon allows us to dynamically offer isolated guest containers and interconnect the containers towards the software switch. An edge node SDN controller is responsible for handling the network connectivity and establishing the necessary flows in the software switch. Finally, we have introduced a container agent (docker-agent) [6] in order to provide a YANG-based RESTconf API, which offers the necessary container and network services to both the cloud/edge and network orchestrators.

V. Proposed IoT-aware SDN and Cloud orchestration architecture

Fig. 2 depicts the proposed IoT-aware multi-layer transport SDN and cloud architecture. At the infrastructure layer, it is composed of several packet and optical transport domains (access, metro and core) providing connectivity to core-DCs and micro & small-DCs (located at the network edge, and referred as edge nodes) providing computing and storage

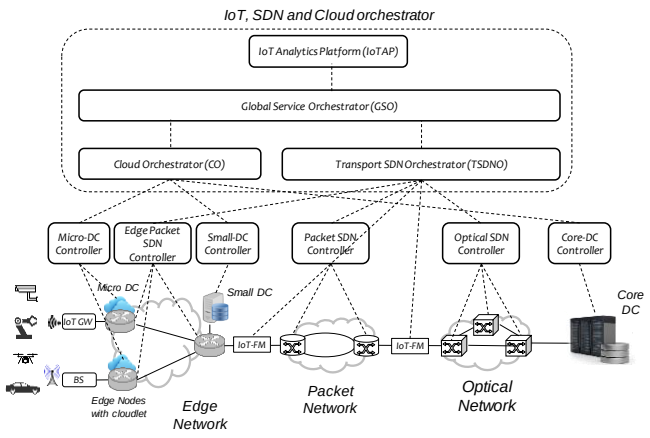

Fig.2. Proposed architecture for IoT, SDN and cloud orchestration

resources. The proposed solution deploys IoT Flow Monitors (IoT-FM) at the edge of the packet transport network domains. The IoT-FMs are responsible of monitoring the average bandwidth of the aggregated IoT traffic.

At the control layer, each micro, small and core DC can have a dedicated DC controller (e.g., OpenStack). On top of the multiple DC controllers we deploy a cloud orchestrator (CO) [7] that enables to deploy federated cloud services for multiple tenants across private distributed DC infrastructures (micro, small, core) as well as public clouds (e.g. Amazon web services, Microsoft Azure). Private distributed DC infrastructures can be controlled with different software implementations (e.g., OpenStack, CloudStack or OpenNebula). There are two OpenStack projects aiming at developing a hierarchical OpenStack architecture. It would enable to develop a cloud orchestrator based on OpenStack (e.g., Trio2o [8] and Tricircle [9]) and use the OpenStack API as both the southbound interface (SBI) with the OpenStack controllers as well as the northbound interface (NBI). This solution is under development, and is limited to the use of OpenStack in all the multiple DCs.

On the transport side, each packet or optical transport domain can have a dedicated SDN controller (e.g., OpenDaylight, ONOS, Ryu). On top of the SDN controllers, we deploy an IoT-aware Transport SDN orchestrator (TSDNO) that acts as a unified transport network operating system (controller of controllers) [10]. The TSDNO allows providing end-to-end connectivity services, at a higher, abstracted level, of heterogeneous network technologies regardless of the specific control plane technology employed in each domain through the use of the common Transport API defined in [20] and experimentally validated by the authors in [11]. The Transport API enables to abstract a set of control plane functions used by an SDN Controller, allowing the TSDNO to uniformly interact with heterogeneous control domains. This abstraction enables the TSDNO to virtualize the network, that is, to partition the physical infrastructure and dynamically create, modify or delete multiple co-existing virtual tenant networks (VTN), independent of the underlying transport technology and network protocols.

In this paper we have extended the TSDNO to provision packet flows tagged as IoT, and to monitor the IoT-FMs in order to detect and prevent IoT-traffic congestion. More specifically, the TSDNO can define for each monitored link the maximum IoT-traffic bandwidth threshold and the time over threshold (ToT) allowed to avoid the generation of alarms for peak traffics above the bandwidth threshold. Then, the TSDNO requests statistics about the IoT traffic in the IoT-FMs to the SDN controllers on a periodic basis. When the TSDNO detects that a defined bandwidth threshold has been exceeded for a duration longer than the allowed ToT, the TSDNO identifies all IoT flows going through the congested link and notifies the IoT-aware global service orchestrator (GSO) to trigger the distribution of IoT analytics to the edge.

The IoT-aware GSO is deployed on top of TSDNO and the CO. It is responsible to provide global orchestration of end-to-end services by decomposing the global service into cloud services, and network services, and forwarding these service requests to the CO and TSDNO. Thus, GSO can dynamically provide network services by coordinating the instantiation and configuration of groups of cloud services (i.e., virtual machines –VMs- /containers – CTs- instances) and the connectivity services between them and the service end-points. The GSO is also responsible to serve application requests and sent notifications through a northbound interface (NBI).

In this paper, we consider the IoT Analytics Platform (IoTAP) on top of the GSO to request the provisioning of edge and core IoT analytics services. The IoTAP is able to deploy and manage the lifecycle of IoT applications running on top of VMs/CTs at the edge nodes (edge computing) and/or the core DC (cloud) that are connected among them and with the service end-points (e.g. IoT gateways). The IoT-aware GSO notifies the IoTAP about all IoT flows going through a detected IoT-traffic congested link, and requests to the IoTAP the distribution of some of the affected IoT analytics services to the edge. Thus, The IoTAP is responsible for selecting the IoT analytics services that will be distributed to the edge, based on the characteristics of the IoT applications. We have developed two IoT applications: a) a video analytics application [12], which is responsible for processing the camera image feed by detecting movement (using Gaussian blur filters, differential image and contour detection), requesting connectivity towards the DC and storing the suspicious videos, and b) the video storage application running in a remote DC.

The considered transport architecture (i.e., SDN controllers and TSDNO) can be mapped to the Abstraction and Control of Traffic Engineered Networks (ACTN) architecture [17] defined in the Internet Engineering Task Force (IETF). ACTN defines the requirements, use cases, and an SDN-based architecture, relying on the concepts of network and service abstraction, detaching the network and service control from the underlying data plane. The architecture encompasses Physical Network Controllers (PNCs), responsible for specific technology and administrative domains, orchestrated by Multi-Domain Service Coordinator (MDSC) which, in turn, enables underlay transport resources to be abstracted and virtual network instances to be allocated to customers and applications, under the control of a Customer Network Controller (CNC). The PNCs are the SDN controllers and the MDSC is the TSNO. Some previous works

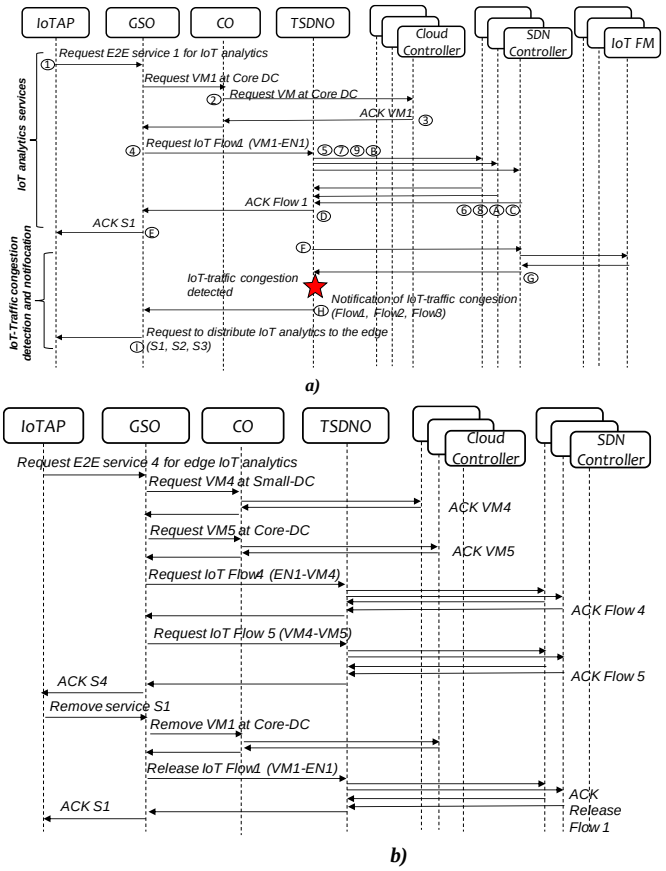

Fig. 3. Orchestration workflows. a) Provisioning of IoT analytics and congestion detection; b) distribution of IoT analytics services

presenting the application of the ACTN architecture are provided in [18][19].

VI. Provisioning of IoT Analytics Service, Congestion Detection, and Distribution of IoT Analytics

When the IoTAP needs to deploy a new core IoT analytics service (i.e. S1) as depicted in Fig. 3.a, it requests to the GSO the provisioning of a cloud computing resource and an IoT flow with the required bandwidth between the VM and the edge node where the IoT flow is originated (service end-point). First, the GSO requests to the CO the provisioning of a VM at the Core-DC. The CO forwards this request to the specific core-DC controller responsible of the actual provisioning (i.e., VM1 in Fig.3.a). Second, the GSO requests the provisioning of an IoT flow (i.e. IoT Flow1 in Fig.3.a) to the TSDNO between the provisioned VM and the edge node specified by the IoTAP (i.e. EN1 in Fig.3.a). The TSDNO is responsible to compute an end-to-end multi-layer under QoS constraints (e.g., bandwidth), to split the computed path into domain segments, and to request the actual provisioning of the path segments to the involved domain SDN controllers. The flows are identified with the IoT tag inserted as an OpenFlow cookie. This process is repeated any time the IoTAP requires to setup an IoT analytics service. Cookie is one of the core fields of OpenFlow FlowMod message. It is OpenFlow specification way of identifying flows for modify/delete operations. Principal goal is to identify flows because there is no concept of FlowId or FlowName in

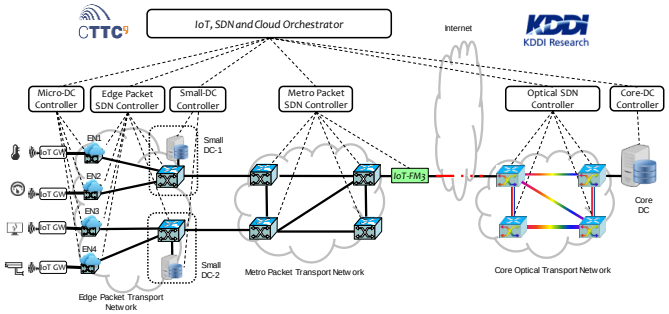

Fig.4. Experimental scenario

OpenFlow. Cookie can be considered as the OpenFlow counterpart of SDN's FlowID, only that it need not be unique. Switch will not use the cookie field when matching data packets in flow table so it doesn't need to reside in hardware. Cookie is used only as a filter for get-stats, flow modifications and flow deletions.

In parallel, the TSDNO monitors the IoT traffic from the IoT-FMs. Once the TSDNO detects an IoT-traffic congested link, it generates an alarm to the GSO, identifying all IoT flows going through the congested link (i.e., flow1, flow2, flow3 in Fig.3.b). After the notification, the GSO identifies all E2E services involved (i.e., S1, S2, S3 in Fig.3.a) and requests to the IoTAP the distribution of some of the affected IoT analytics services to the edge.

The IoTAP is responsible for selecting the IoT analytics services that will be distributed to the edge, based on the characteristics of the IoT applications. Once selected the services to be distribute (i.e. S1 in Fig.3.b), first the IoTAP requests to the GSO the provisioning of a new edge IoT analytics service (i.e., S4 in Fig.3.b) composed of a cloud computing resource (i.e., VM4 at core-DC in Fig.3.b) connected with an edge computing resource (i.e., VM5 at small-DC Fig.3.b) and connected with the service end-point (i.e., EN1 Fig.3.b). Then, the IoTAP proceeds to release the established IoT Analytics service, by specifying to the GSO the identifier of the E2E service (i.e., S1 Fig.3.b).

VII. Experimental Assessment

A. *Experimental scenario*

Two proof-of-concepts (PoCs) of the whole IoT-aware multi-layer (packet/optical) transport SDN and edge/cloud orchestration architecture has been evaluated and validated in a joint experimentation between the CTTC ADRENALINE Testbed [13] in Barcelona (Spain) and the KDDI Research Testbed in Saitama (Japan) as depicted in Fig.4.

The ADRENALINE testbed provides an edge and metro packet transport networks for traffic aggregation and switching of flows, and distributed edge computing platform. The KDDI Research testbed provides a core-DC and an optical core network. Both infrastructures are connected using OpenVPN tunnels on top of internet.

The packet transport network leverages on cost-effective OpenFlow switches deployed on commercial off-the-shelf (COTS) server and using Open vSwitch (OVS) technology. There are a total of ten OpenFlow switches distributed in the

edge (access) and metro (aggregation) network segments. The metro packet transport network is composed of 4 OpenFlow switches. The edge packet transport network is composed of four edge nodes (providing connectivity to IoT access gateways) and two OpenFlow switches located in the central offices. The edge nodes are lightweight servers developed using an Intel NUC 6i7KYK2 (including an i7 processor), with 32Gb RAM and 512 Gb SSD. Several USB to Ethernet port converters have been included in order to extent the node switching capabilities. The edge node follows the presented architecture described in Fig.1, including Docker, OpenVSwitch, and a docker agent [6].

The distributed core and edge cloud platform is composed by by one core-DC, two small-DCs, and four micro-DCs, leveraging virtual machines (VM) and container (CT)-based technologies. Specifically, VM-centric host virtualization is used for the core-DC and small-DCs, and CT-based technology, less secure but lightweight, for micro-DCs. The core-DC is composed of a compute node (HPC servers with a hypervisor to deploy and run VMs), as well as the small-DCs deployed into servers with 2 x Intel Xeon E5-2420 and 32GB RAM. The four micro-DCs are integrated in the edge nodes, together with the OpenFlow switch. The distributed edge cloud platform is controlled using three OpenStack controllers, one for the core-DC, another for the two small-DCs, and the third one for the four micro-DCs. Within a DC, OpenStack networking service (Neutron) manages networks and IP addresses, providing flat networks or VLANs to separate traffic between hosts. For example, the OpenStack Neutron service enables to configure the virtual switch (e.g., OVS) within a compute node (e.g. creation of new ports connecting new VMs/CTs, configuration of forwarding rules) through an SDN controller. The OpenStack Neutron service assumes that between the compute nodes there is a network that does not require any configuration (e.g., Layer 2 network), and therefore, it just focuses on the configuration of the L2 virtual switches at the compute nodes. The configuration of the OpenFlow switches of the edge (access) and metro (aggregation) network segments are performed by two SDN controllers based on Ryu.

The implementations of the IoT, SDN and cloud orchestrator (i.e., CO, TSDN, GSO, IoTAP) use Python. Several internal REST interfaces are offered between the different components. It is located in CTTC premises.

B. Provisioning of distributed IoT analytics

We consider the use case of video analytics on a CCTV running at the edge of the network, as shown in Fig.5.a. It is based on the ETSI MEC PoC defined in [14]. This use case enables to reduce the network bandwidth utilization up to 90% by deploying video analytics at the network edge, as stated in [14] and [32]. Standard video analytics require a constant bitrate of 240 Kbps towards a DC running the necessary video surveillance applications [14]. The overall usage of CCTV cameras easily justifies the need for Gbps for transmission of raw data. We move the video analytics applications to the network edge nodes. This allows us to process the video stream closer to the CCTV sources, and only transmit data to the core-DC in case an event is detected by the application, as shown in

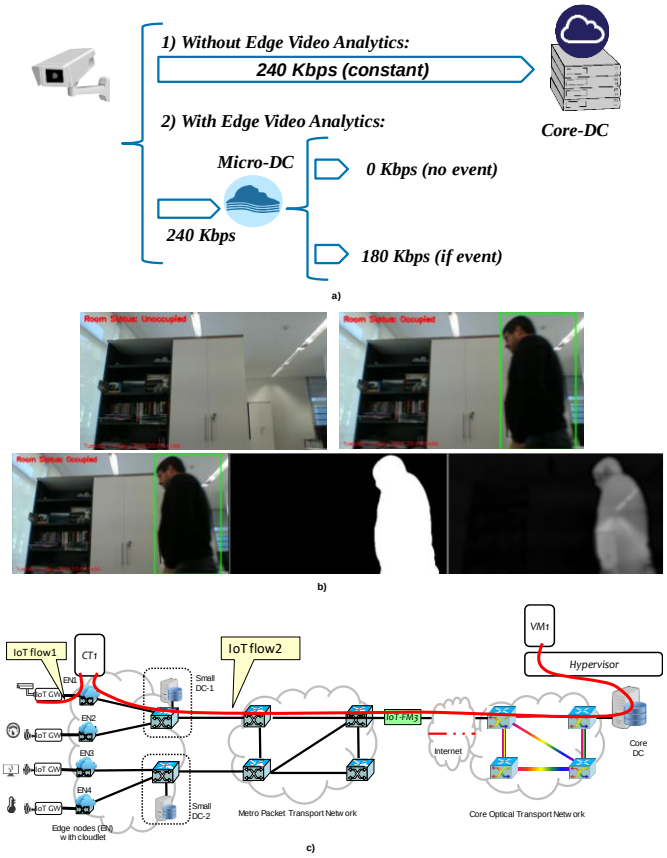

Fig. 5. Proof-of-concept #1. a) video analytics use case ; b) video analytics application for movement detection; c) Provisioning of a core IoT analytics service for video analysis at the edge node

Fig.5.b. A video storage application is run in the core data center. When the video analytics app detects a movement, it sends the video in real-time to the core DC in order to be stored and processed.

Firstly, the IoTAP requests to the GSO the provisioning of a new edge IoT analytics service composed of a cloud computing resource (i.e., VM1 at core-DC) connected (i.e., flow2) with an edge computing resource (i.e., container CT1 at micro-DC in EN1) and connected (i.e., flow1) with the service end-point (i.e., the CCTV camera), as shown in Fig.5.c. The GSO requests the provisioning of the VM and CT1 to the CO, and the provisioning of the IoT flow1 and flow2 to the TSDNO, following the workflow described in Fig.3.b. The actual provisioning of VM1/CT1 is performed by the cloud controllers selected by the CO, and of flow1/flow2 by the SDN controllers selected by the TSDNO. The VM is used to deploy the video storage application, and the container for the video analytics application. Once the video analytics is running in the micro-DC in the edge node, and a suspicious movement is detected, the video analysis application selects and processes the video frames which are suspicious and transmit them into the video storage application running on the VM at core-DC using SCP (with SSL encryption).

The required time to create a VM is above some few seconds (e.g, 2-8s), and the container is created within 400ms. Finally, the connectivity service through the underlying networks domains requires 730ms, due to the fact that optical connections where previously established. The optical connections are

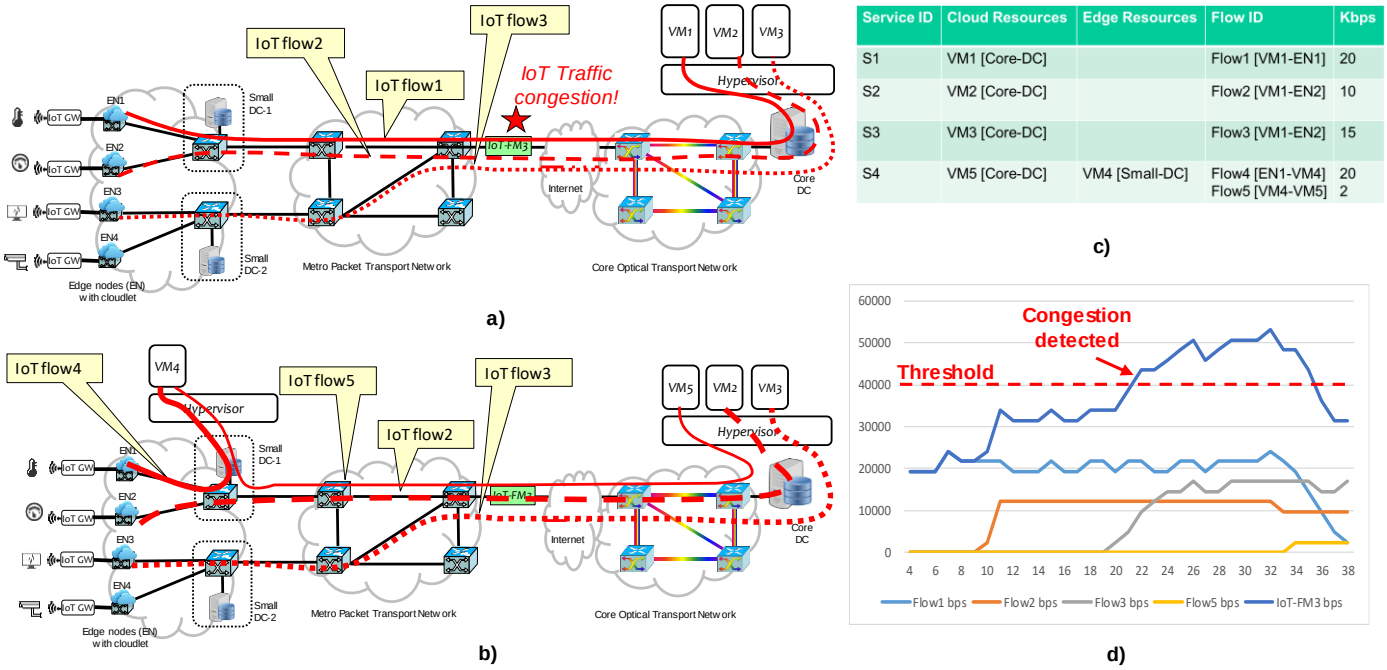

Fig. 6 Proof-of-concept #2 a) Provisioning of core IoT analytics services and congestion detection, b) Distribution of IoT analytics services to the edge, c) requested services, d) IoT traffic monitored by IoT-FM3

dynamically configured using the optical SDN controller the first time of the experimentation. Once established, they are used to transport the provisioned packet flows. For the sake of simplicity, details are skipped in the description of the workflow. For detailed information, please refer to the work [16] from the authors.

C. Provisioning of core IoT analytics, congestion detection and distribution to the edge

For the PoC, the IoT-FM has been integrated in the OpenFlow switches for simplicity. In the experimentation, the maximum IoT-traffic bandwidth threshold is configured to 40 Kbps and the TOT to 10s. Individual flow statistics information is requested with the OFPST_FLOW stats request type. The TSDNO requests the byte_count of the IoT flows on the output port of the OpenFlow switch where the IoT-FM should be placed with a periodicity of 1s. Then, the TSDNO computes, each second, the average IoT bandwidth (bw) for each IoT flow in the last 5 seconds, that is, the bandwidth at time n (tn) is (byte_count (tn) - byte_count (tn-5)) / 5. The aggregated IoT traffic is the sum of the average bw of all IoT flows.

First, we provision three E2E services (i.e., S1, S2 and S3 in Fig.6.a) for deploying core IoT analytics according to the parameters shown in Fig.6.c. Fig.7.a shows the network traffic capture at the GSO for the provisioning of an E2E service, showing all the communications workflows between the different involved systems (the numeration in the left side is used to map with the workflow of Fig3.a). After the provisioning of each service, we use a traffic generator to generate packets with a constant bitrate, according to Fig3.c (20Kbps, 10 Kbps and 15 Kbps). Fig.6.d shows the IoT traffic monitored by IoT-FM3 located between the metro and optical networks. IoT-FM3 shows the average IoT bandwidth employed by flow1, flow2 and flow3, as well as the overall aggregated traffic. We can see that at time 21s the maximum IoT bandwidth (40Kps) is exceeded and therefore at time 31s the TSDON notifies about the IoT-traffic congested link, as shown in the network traffic capture of Fig.7.b.

Then, the IoTAP decides to provision an edge IoT analytics service (S4 in Fig.6.b) and remove the core IoT analytics service S1. From Fig.6.d, we can appreciate that at time 34s, the flow1 (20Kbps) from S1 starts to decrease, and the new flow5 (2Kbps) from S4 appears. Flow4 is not shown because VM4 is located in the small-DC, and therefore does not cross Iot-FM3.

VIII. CONCLUSION

We have presented and experimentally assessed the first IoT-aware SDN and cloud architecture. It deploys IoT flow monitors and integrates IoT traffic-congestion avoidance techniques in the control and orchestration platform. It enables to offload the transports networks by dynamically and efficiently distributing the processing of IoT analytics from core datacentres to the network edge.

We have also proposed an SDN-enabled container-based edge node and have presented its integration in SDN control architecture, both at cloud and network resources. We have presented a use case for reducing network bandwidth utilization up to 90% by deploying video analytics at the network edge.

REFERENCES

[1] Remote diagnostics informs wind turbine O&M, Siemens The Magazine https://www.siemens.com/customer-magazine/en/home/energy/power-transmission-and-distribution/remote-diagnostics-informs-wind-turbine-o-and-m.html

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper.

[3] R. Vilalta, et al., End-to-End SDN/NFV Orchestration of Video Analytics Using Edge and Cloud. In Proc. of Optical Fiber Conference (OFC), March 2017.

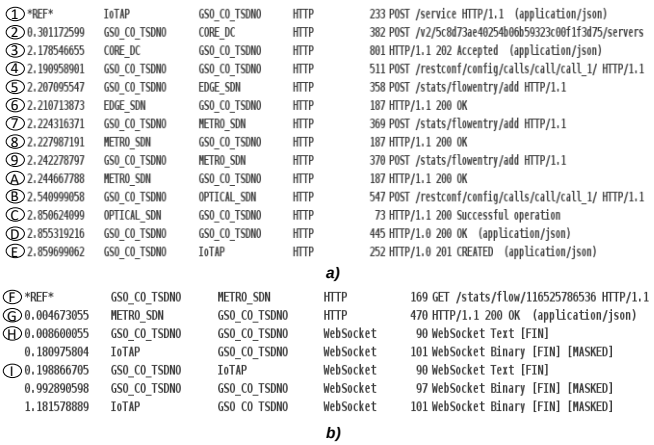

Fig. 7. Wireshark captures. a) Provisioning of a core IoT analytic serive; b) congestion detection

[4] N. Yoshikane et al., "First Demonstration of Geographically Unconstrained Control of an Industrial Robot by Jointly Employing SDN-based Optical Transport Networks and Edge Compute.", PDP 1.1, OECC/PS 2016.

[5] Y. Nakagawa, et al. "Dynamic virtual network configuration between containers using physical switch functions for NFV infrastructure." Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on. IEEE, 2015.

[6] https://github.com/rvilalta/docker-agent

[7] R. Muñoz et al, "Integrating Optical Transport Network Testbeds and Cloud Platforms to Enable End-to-End 5G and IoT Services", 19th International Conference on Transparent Optical Networks (ICTON 2017, July 2017.

[8] Openstack Trio2o project (https://wiki.openstack.org/wiki/Trio2o).

[9] Openstack Tricircle project (https://wiki.openstack.org/wiki/Tricircle)

[10] R. Muñoz et al., Transport network orchestration for end-to-end multilayer provisioning across heterogeneous sdn/openflow and gmpls/pce control domains," Journal of Lightwave Technology, vol. 33, no. 8, pp. 1540–1548, 2015.

[11] A. Mayoral, et al. "First experimental demonstration of distributed cloud and heterogeneous network orchestration with a common Transport API for E2E services with QoS." Optical Fiber Communication Conference, 2016.

[12] https://github.com/rvilalta/motion_detector

[13] R. Muñoz, et al. "The Need for a Transport API in 5G networks: the Control Orchestration Protocol", in Proceedings of Optical Fiber Conference (OFC), March 2016"

[14] PoC 8 Video Analytics, ETSI MEC, September 2016. http://mecwiki.etsi.org/index.php?title=PoC_8_Video_Analytics

[15] R. Muñoz et al, "IoT-aware Multi-layer Transport SDN and Cloud Architecture for Traffic Congestion Avoidance Through Dynamic Distribution of IoT Analytics, European Conference on Optical Communications (ECOC), September 2017.

[16] R. Muñoz, R. Vilalta, R. Casellas, R. Martínez, F. Francois, M. Channegowda, A. Hammad, S. Peng, R. Nejabati, D. Simeonidou, N. Yoshikane, T. Tsuritani, V. López, A. Autenrieth, Transport Network Orchestration for end-to-end Multi-layer Provisioning Across heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains , Journal of Lightwave Technology, Vol. 33, No. 8, pp. 1540 - 1548, April 2015.

[17] Daniele Ceccarelli, Young Lee, Framework for Abstraction and Control of Traffic Engineered Networks, Internet Engineering Task Force (IETF), work in progress, draft-ceccarelli-teas-actn-framework, October 2017

[18] R. Casellas, R. Vilalta, R. Martínez, R. Muñoz, H. Zheng, Y. Lee, Experimental Validation of the ACTN architecture for flexi-grid optical networks using Active Stateful Hierarchical PCEs , within the 19 International Conference on Transparent Optical Networks (ICTON2017), 2-6 July, 2017, Girona, Spain, July 2017.

[19] R. Vilalta, Y. Lee, H. Zheng, Y. Lin, R. Casellas, A. Mayoral, R. Martínez, R. Muñoz, L. Miguel Contreras, V. López, Fully Automated Peer Service Orchestration of Cloud and Network Resources Using ACTN and CSO, in Proceedings of SDN & NFV Demo Zone at International Conference on Optical Fiber Communications (OFC), 19-23 March 2017, Los Angeles (USA).

[20] ONF technical recommendation, Functional Requirements for Transport API, ONF TR-527, June 2016.

[21] Pankesh Patel; Muhammad Intizar Ali; Amit Sheth, On Using the Intelligent Edge for IoT Analytics, IEEE Intelligent Systems, Vol. 32, No. 5, pp. 64-69, 2017

[22] Glenn Daneels; Esteban Municio; Kathleen Spaey; Gilles Vandewiele; Alexander Dejonghe; Femke Ongenae; Steven Latré; Jeroen Famaey, Real-Time data dissemination and analytics platform for challenging IoT environments, 2017 Global Information Infrastructure and Networking Symposium (GIIS), 2017.

[23] Hazem M. Raafat; M. Shamim Hossain; Ehab Essa; Samir Elmougy; Ahmed S. Tolba; Ghulam Muhammad; Ahmed Ghoneim, Fog Intelligence for Real-Time IoT Sensor Data Analytics, IEEE Access, Vol.5, pp 24062 – 24069, 2017.

[24] Shusen Yang, IoT Stream Processing and Analytics in the Fog, IEEE Communications Magazine, Vol.55, No.8, pp. 21 – 27

[25] Paula Ta-Shma; Adnan Akbar; Guy Gerson-Golan; Guy Hadash; Francois Carrez; Klaus Moessner, An Ingestion and Analytics Architecture for IoT applied to Smart City Use Cases, IEEE Internet of Things Journal, DOI 10.1109/JIOT.2017.2722378

[26] Dan Puiu; Stefan Bischof; Bogdan Serbanescu; Septimiu Nechifor; Josiane Parreira; Herwig Schreiner, A public transportation journey planner enabled by IoT data analytics, 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), 2017

[27] P. Dineshkumar; R. SenthilKumar; K. Sujatha; R. S. Ponmagal; V. N. Rajavarman, Big data analytics of IoT based Health care monitoring system, IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON), 2016.

[28] Ravi Ramakrishnan; Loveleen Gaur, Smart electricity distribution in residential areas: Internet of Things (IoT) based advanced metering infrastructure and cloud analytics, International Conference on Internet of Things and Applications (IOTA), 2016.

[29] R. Vilalta, A. Mayoral, J. Baranda, J. Núñez, R. Casellas, R. Martínez, J. Mangues, R. Muñoz, Hierarchical SDN Orchestration of Wireless and Optical Networks with E2E Provisioning and Recovery for Future 5G Networks , in Proceedings of the Optical Fiber Communication Conference and Exhibition (OFC), 20-24 March 2016, Anaheim, California (USA).

[30] R. Muñoz, L. Nadal, R. Casellas, M. Svaluto Moreolo, R. Vilalta, J. M. Fabrega, R. Martínez, A. Mayoral, F. J. Vilchez, The ADRENALINE Testbed: An SDN/NFV Packet/Optical Transport Network and Edge/Core Cloud Platform for End-to-End 5G and IoT Services , in Proceedings of European Conference on Networks and Communications (EUCNC 2017), June 12-15 2017, Oulu (Finland)

[31] R. Muñoz, R. Vilalta, R. Casellas, A. Mayoral, R. Martínez, Integrating Optical Transport Network Testbeds and Cloud Platforms to Enable End-to-End 5G and IoT Services , in Proceedings of 19th International Conference on Transparent Optical Networks (ICTON 2017), 2-6 July 2017, Girona (Spain).

[32] ETSI Webinar: MEC PoC#8 – Video Analytics http://www.etsi.org/news-events/events/1122-2016-09-webinar-mec-poc-8-video-analytics

[33] Connected cars will send 25 gigabytes of data to the cloud every hour https://qz.com/344466/connected-cars-will-send-25-gigabytes-of-data-to-the-cloud-every-hour/

BIOGRAPHIES

Raül Muñoz (SM'12) is graduated in telecommunications engineering in 2001 and received a Ph.D. degree in telecommunications in 2005, both from UPC-BarcelonaTech, Spain. He is Head of the Optical Networks and Communications Networks Division Manager at CTTC (Barcelona, Spain). Since 2000, he has participated over 40 R&D projects funded by the EC's Framework Programmes, the Spanish Ministries, and the industry. He has coordinated the and the H2020-MSCA-ITN ONFIRE project and the EU-Japan FP7-ICT STRAUSS project. He has published over 250 journal and international conference papers.

Ricard Vilalta (SM'17) has a telecommunications engineering degree (2007) and Ph.D. degree (2013), at UPC, Spain. He is senior researcher at CTTC, in the Communication Networks Division. His research is focused on SDN/NFV, Network Virtualization and Network Orchestration. He has been involved in

international, EU, national and industrial research projects, and published more than 170 journals, conference papers and invited talks. He is also involved in standardization activities in ONF, IETF and ETSI.

Noboru Yoshikane joined Kokusai Denshin Denwa Company Ltd. (currently KDDI Corporation), Tokyo, Japan in 1999, and since 2001, he has been working at their Research and Development Laboratories. He has been engaged in research on the design of submarine cable systems, highly spectrally efficient optical communication systems utilizing wavelength division multiplexing transmission, and designing and modeling of photonic networks. He is a Member of the IEICE.

Ramon Casellas (SM'12) graduated in telecommunications engineering in by the UPC-BarcelonaTech and ENST Telecom Paristech (1999). He worked as an undergraduate researcher at France Telecom R&D and British Telecom Labs, and completed a Ph.D. in 2002 at ENST, working as an associate professor. He joined CTTC in 2006, working in international and technology transfer research projects. His research interests include network control and management, traffic engineering, GMPLS/PCE, SDN and NFV. He has published over 180 papers, 4 IETF RFCs and 4 book chapters.

Ricardo Martínez (SM'14) received an M.Sc. degree in 2002 and a Ph.D. degree in 2007, both in telecommunications engineering, from the UPC–BarcelonaTech University, Spain. He has been actively involved in several EU public-funded and industrial technology transfer projects. Since 2013, he is Senior Researcher at CTTC in Castelldefels, Spain. His research interests include control and orchestration architectures for heterogeneous and integrated network and cloud infrastructures along with advanced mechanisms for provisioning/recovering quality-enabled services.

Takehiro Tsuritani received M.E. and the Ph.D. degrees in Electronics Engineering from Tohoku University, Miyagi, Japan, in 1997 and 2006, respectively. He joined Kokusai Denshin Denwa (KDD) Company, Limited (currently KDDI Corporation), Tokyo, Japan in 1997. Since 1998, he has been working at their Research and Development Laboratories (currently KDDI Research, Inc.) and has been engaged in research on high-capacity long-haul wavelength-division multiplexing (WDM) transmission systems and dynamic photonic networking. Currently, he is working as a Senior Manager of the Photonic Transport Network Laboratory in KDDI R&D Laboratories Inc. and is a Senior Member of IEICE. He is a recipient of the Best Paper Award of OECC 2000.

Itsuro Morita received B.E., M.E., and Dr. Eng. degrees in Electronics Engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1990, 1992, and 2005, respectively. He joined Kokusai Denshin Denwa (KDD) Co mpany, Ltd. (currently KDDI Corporation), Tokyo, in 1992, where he has been with the Research and Development Laboratories since 1994. He has been engaged in research on long-distance and high-speed optical communication systems. In 1998, he was on leave at Stanford University, Stanford, CA. He is currently an Executive Director of KDDI Research, Inc. and a Senior Member of IEICE.
Dr. Morita was a recipient of the Minister Award from METI in 2006 and the Hisoka Maejima Award from the Tsushinbunka Association in 2011.