Wireless Pers Commun
DOI 10.1007/s11277-016-3845-0

# Software-Defined Fog Network Architecture for IoT

**Slavica Tomovic[1] · Kenji Yoshigoe[2] · Ivo Maljevic[3] · Igor Radusinovic[1]**



**Abstract** Rapid increase in number and diversity of Internet-connected devices raises many challenges for the traditional network architecture, which is not designed to support a high level of scalability, real-time data delivery and mobility. To address these issues, in this paper we present a new model of Internet of Things architecture which combines benefits of two emerging technologies: software-defined networking and Fog computing. Software-defined networking implies a logically centralized network control plane, which allows implementation of sophisticated mechanisms for traffic control and resource management. On the other hand, Fog computing enables some data to be analysed and managed at the network edge, thus providing support for applications that require very low and predictable latency. In the paper, we give detailed insight into the system structure and functionality of its main components. We also discuss the benefits of the proposed architecture and its potential services.

**Keywords** Internet of things · SDN · Fog computing

✉ Slavica Tomovic
slavicat@ac.me

Kenji Yoshigoe
kxyoshigoe@ualr.edu

Ivo Maljevic
ivom@ieee.org

Igor Radusinovic
igorr@ac.me

[1] Faculty of Electrical Engineering, University of Montenegro, Podgorica, Montenegro

[2] Donaghey College of Engineering and Information Technology, University of Arkansas, Little Rock, Arkansas, USA

[3] TELUS Communications, University of Toronto, Toronto, Ontario, Canada

Published online: 24 October 2016

## 1 Introduction

Advances in wireless communications and micro-electro-mechanical systems (MEMS) have enabled the rapid evolution of smart devices connected to the Internet. This evolution inspired the idea about the Internet of Things (IoT)—a large-scale cognitive system in which wide variety of "things" could contribute. The definition of "thing" is very flexible, and may refer to: intelligent machines, drones, self-driving cars, sensor nodes measuring parameters such as temperature and humidity, actuators that turn on and off devices or make adjustments in real time, and much more. Thus, the exponential increase in the volume and variety of data is expected, creating a significant burden for the Internet architecture [1]. Adding more resources to provide enough capacity cannot be economically justified in the long-run. The major scale issue is not the volume of traffic, but the type and cadence of data delivery. IoT devices are often configured to send regular updates throughout the day, which may cause a tsunami of connections and data at periodic intervals. Regular surges of traffic may surpass baseline or average traffic by a significant multiple of existing traffic patterns, so high level of resource over-provisioning could be required for stable operation. To avoid this, service providers need mechanisms that can satisfy the bandwidth demand of IoT applications by efficiently utilizing the existing infrastructure. That is hard to achieve with distributed control plane in traditional network architectures, since a global view of the network state is lacking. Another challenge is low-latency handling of time-critical tasks such as analysis and decision-making. For example, network latency could badly affect traffic management application which requires real-time detection of the congested sites. On the other hand, emerging augmented reality applications (e.g. Google Glass, Sony SmartEyeGlass and Microsoft HoloLens) have to process real-time video, voice and sensor measurements in order to finally output informational content on displays [2]. Without almost deterministic response time utility, adoption of this technology is questionable. In general, because of a loosely controlled nature of the Internet many QoS (Quality of Service) issues are still unresolved.

In this paper, we propose the use of SDN (Software Defined Networking [3]) to alleviate resource contentions in IoT environment and improve overall IoT performance. SDN is a relatively new paradigm for communication networks, that implies separation of the forwarding and the control functions. Network intelligence is moved to logically centralized SDN controller, which maintains a global view of the network, interacts with data-plane devices and provides a programming interface for network management applications. The potential of this concept reflects in the fact that traffic engineering and resource management can be performed more efficiently in centralized system having insight into applications' requirements and all resources available. To address the applications that require mobility support and low-delay, the proposed IoT architecture integrates SDN with Fog computing [4]. Although the benefits of the both technologies have been widely recognized within the research community, there are still many challenges that hinder widespread acceptance of them. Therefore, this paper targets to explain how SDN and Fog computing could be efficiently combined together to compensate each other deficiencies. In particular, the proposed IoT architecture aims to solve the problem of Fog orchestration with SDN, as well as scalability issues of SDN with Fog computing.

The rest of the paper is organized as follows. Section 2 provides brief background on traditional IoT architecture and related challenges. Section 3 introduces the basic ideas of SDN and Fog computing and then describes the components and operation model of the

proposed architecture. A few use cases that would especially benefit from the proposed architecture are discussed in Sect. 4. Section 5 concludes the paper.

## 2 IoT: Current State and Challenges

The IoT paradigm is based on various kinds of smart devices with communication and networking capabilities, embedded in the environment around us. According to CISCO forecasts the number of Internet-connected devices overtook the human population in 2010, and will be about 50 billion by 2020 [5]. Although these devices mostly have low bandwidth demands, the overall result is enormous amount of data which have to be stored, processed and presented in a seamless, efficient, and easily interpretable form [6]. Figure 1 illustrates the common organization of today's IoT architecture from a high level perspective. Four main components involved are: sensing devices (things), the local communication network, the Internet Cloud and back-end IoT applications. Sensing devices gather data from the physical environment. These data are later used by IoT applications (e.g. smart transportation, healthcare, precision agriculture, video surveillance, etc.) to provide a desirable service to end customers. Since IoT devices are in general characterized by very limited memory and computational resources, IoT application usually takes advantage of services offered by the Cloud for data storage and processing. To reach the Cloud, sensing devices rely on different communication technologies. Those more powerful connect directly to cellular network (3G or 4G), or use Wi-Fi/Ethernet connectivity to the Internet gateway. However, these communication models require a fair amount of power that myriad devices cannot afford (e.g. battery operated devices). For the short range communication between energy-deficient devices, some other options are more convenient, such as Bluetooth, ZigBee or NFC (Near-Field Communication). Once the raw data generated by one or more sensing sources are appropriately processed in the Cloud, the useful information are finally delivered to the end user. That may be commercial or an industrial user, or another device in M2M (Machine-to-Machine) workflow.

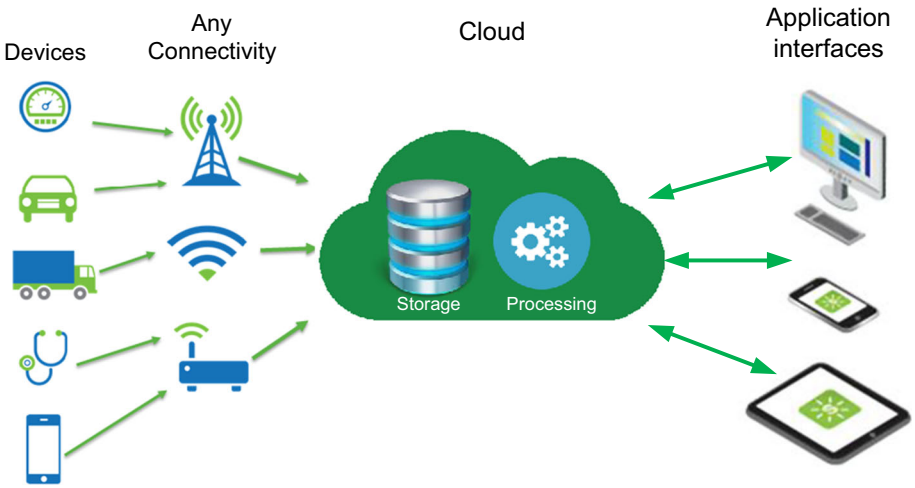


**Fig. 1** Traditional IoT architecture

Cloud offers scalable and cost effective solution to deal with data produced by IoT. Its virtually unlimited resources compensate technological limitation of IoT devices (e.g. storage, processing and energy) and enable analysis of unprecedented complexity [7]. Also, "pay-as-you-go" Cloud computing model is a more convenient alternative to owning and managing private data centres (DCs) if consider deployment and operational expenditures. However, existing Cloud services are originally designed for classic Web applications, which are not significantly affected by the distance between the edge devices and DCs. On the other side, many emerging IoT applications require real-time interaction and mobility support (e.g. smart traffic lights and target tracking systems), which makes network latency an important limiting factor. Latency introduced in the network is not only a consequence of long distance between IoT devices and the Cloud. It is also caused by queuing delay, which is non-negligible on the congested links. The impact of queuing delay could be reduced if traffic load is evenly distributed over the network. Unfortunately, dynamic routing is considered rather risky than beneficial in existing Internet architecture characterized by the distributed control plane [8]. Moreover, the simple shortest path routing model is still dominantly deployed. Dedicated mechanisms for connectivity control are also lacking. This must be resolved in order to fully exploit the opportunities offered by heterogeneous access networks in IoT environments.

## 3 SDN-Based Model of IoT System

In order to address previously discussed challenges, in this Section we propose a new model of IoT architecture based on two emerging technologies: SDN and Fog computing. We will briefly present basic ideas of these networking paradigms, and then explain how they are combined together and envisioned to operate in the proposed system model.

### 3.1 Software Defined Networking

In traditional telecommunications networks the control and the data plane are implemented in each networking device. The control plane carries signalling traffic, performs route calculation, system configuration and management. It contains all the logic that controls the behaviour of the network. On the other hand, the data plane is focused only on the transport of packets towards their next destination. The network itself basically could be seen as a distributed entity that connects diverse independent and autonomous devices. Network management is done at very low level, and once forwarding policy is defined, the only way to make an adjustment to the policy is via manual configuration of the devices. This limits ability to introduce new services in the network or to adapt the network behaviour to varying application requirements and load condition [9, 10]. Novelty introduced with SDN is clear separation of the control and the data plane. SDN control plane is placed on a logically centralized controller, which maintains a global view of the network, interacts with simple forwarding devices and provides a programming interface for network management applications (Fig. 2). In this way, SDN allows network managers to configure and optimize network resources dynamically via automated programs [3].

The communication between SDN controller and the data plane devices is commonly achieved via OpenFlow protocol [11]. The controller pro-actively or reactively instructs the data plane devices (OpenFlow switches) how to identify and treat different traffic flows in the network. When OpenFlow switch receives the instructions for specific traffic flow, it

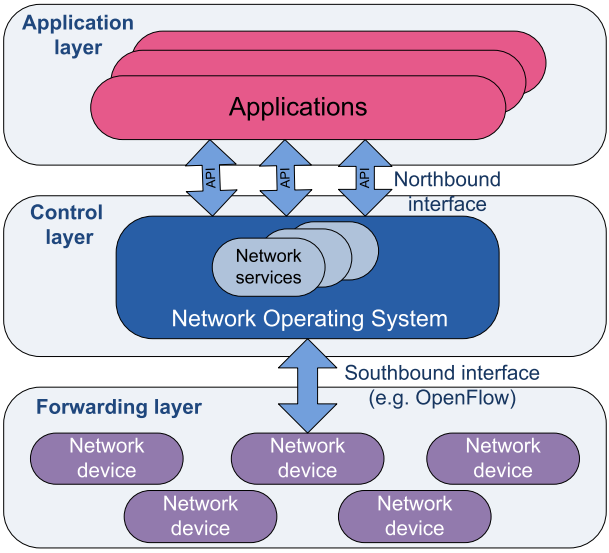


**Fig. 2** SDN architecture

is able to handle packets belonging to the flow without further interaction with the controller until validation time for the instructions expires. Note that SDN/OpenFlow architecture described above is originally designed for DC and WAN (Wide Area Network) networks. However, the need for similar technology in 5G mobile networks has been widely recognized by the research communities [9, 12, 13].

### 3.2 Fog Computing

Fog computing is an emerging technology that brings data processing, storage and analytics closer to the network edge. It has many mechanisms and attributes in common with Cloud computing, however, Fog can be distinguished from Cloud by its proximity to end-users, the dense geographical distribution and its support for mobility [4]. Figure 3 illustrates the position of Fog computing in IoT systems. It is obvious that Fog cannot substitute the Cloud, but complements its services by introducing a new intermediate layer composed of geo-distributed Fog nodes. Each Fog node is highly virtualized platform hosted on dedicated computing node equipped with communication interface, or resource-poorer device such as set-top box, access point, router, switch, etc. Data collected by IoT sensing devices are not send directly to the Cloud server for processing. Instead, they are sent to nearby Fog node in order to obtain fast and high-rate service. However, Fog node can filter out non-actionable data (e.g. regular sensor measurements) and send them to the Cloud for long-term storage and batch analytics. The Cloud is a natural place to run global analytics on data collected from widely distributed devices over long periods (months, years) [1, 14].

Fog computing does not only enhance QoS (Quality of Service) for a large number of IoT applications, but also may significantly reduce bandwidth consumption in the backbone network. Consequently, the users could be benefited from the reduced service costs [15]. This vision has been recently made possible by Cisco IOx platform, which combines open-source Linux and Cisco IOS network operating system together in a single networked device (router, switch or IP camera). Linux transforms the underlying device to mini-

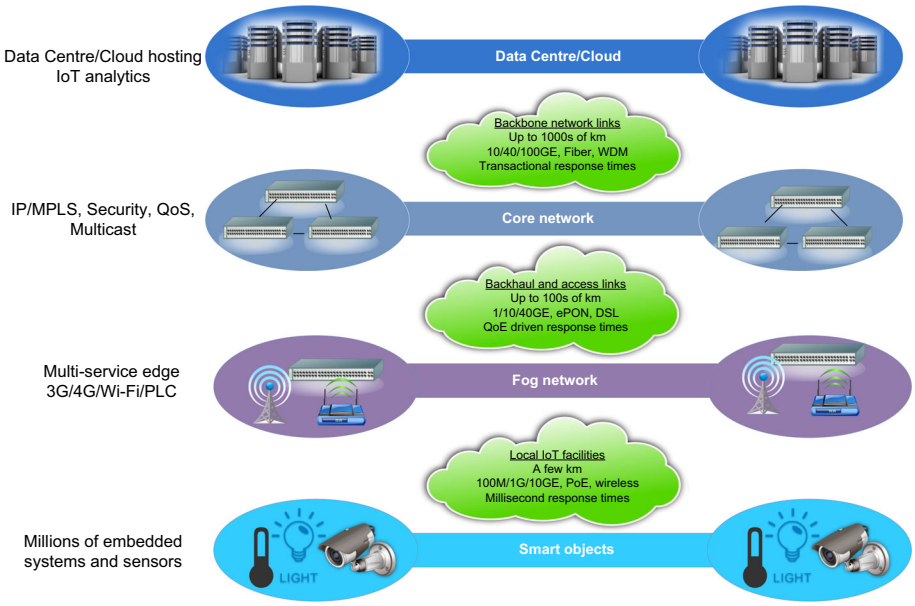


**Fig. 3** The role of Fog layer in IoT architecture

computer that can host third party applications on virtual containers and analyse data locally in real time [16].

### 3.3 The Proposed System Design

This section describes a model of IoT architecture which takes advantage of SDN and Fog computing paradigms. The proposed solution is inspired by recent works on these topics [1, 17–20]. However, while each of them is either focused only on one of the technologies or considers their application in VANET (Vehicular Ad-hoc NETwork) networks, we analyse the generic IoT scenario where features of both technologies are combined together in one integrated system.

Figure 4 shows the system structure which involves: end devices with multiple wireless communication solutions, SDN controllers, heterogeneous Fog infrastructure (virtualized servers, routers, access points, etc.) and Cloud in the network core. Since IoT applications may be geospatially distributed, we assumed hierarchical deployment of Fog network. As illustrated in Fig. 5a, Fog nodes expose a set of APIs (Application Programming Interfaces) for application deployment and development, resource management and control. These APIs allow seamless access to hypervisors, various operating systems and service containers on a physical machine [1]. Also, they enable remote monitoring and management of physical resources such as CPU, memory and network interfaces. Development of IoT applications using hierarchically deployed and heterogeneous Fog resources could be simplified by adopting Mobile Fog programming model [20]. Mobile Fog runs the same application code on various devices of the heterogeneous Fog infrastructure. The application consists of multiple processes that perform different tasks with respect to the device capabilities and position in the network hierarchy. For example, tasks of large-scale video surveillance application may be organized in three levels: motion detection at IP camera,

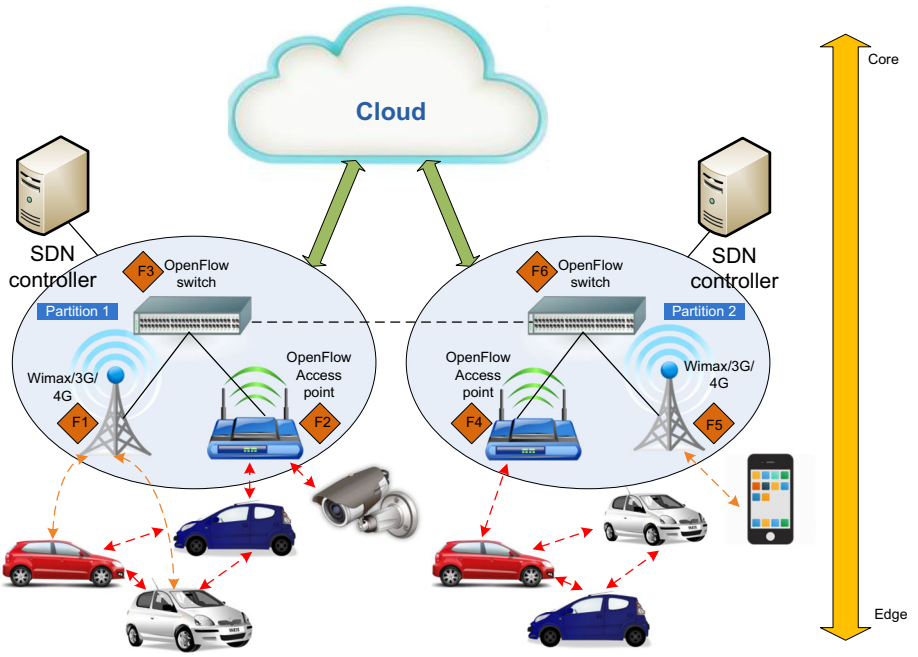


**Fig. 4** SDN architecture for IoT based on Fog computing

face recognition at edge Fog nodes and aggregation of identities at Cloud server [20]. It is assumed that each of the devices has information about its geophysical location. Thus, although all of them run the same code, each one is aware of its particular tasks.

A major challenge imposed by Fog concept refers to service orchestration. The orchestration involves automated instantiation, replication and migration of service instances on a large volume of Fog nodes with a wide range of capabilities. As discussed earlier, many IoT applications deal with dynamic workload due to periodic or event-driven data delivery models. In an ideal case, applications should be transparently scaled at the runtime without resource over-provisioning. In order to achieve that, we propose logical centralization of orchestration functionality at SDN controller. The design of SDN controller is modified compared to traditional one used in DC networks. As illustrated in Figure 5b, its role in IoT system is threefold:

1. Fog orchestration.
2. Injection of routing logic into SDN-enabled network elements.
3. Optimal selection of access points for IoT devices (i.e. radio access network management).

To perform above tasks efficiently the controller needs an up-to-date view of the system. For this reason, it collects and maintains information about:

– Features of Fog nodes in the controlled domain, such as: available RAM, secondary storage, running Operating Systems and software applications [1].
– Capabilities, state and interconnectivity of the network elements, including: wireless technology of the access points (e.g. 3G/4G, LTE, Wi-Fi etc.), links capacity and residual bandwidth, the flow table content and neighbour list of each network node.


**Fig. 5** Structure of the Fog node (**a**), SDN controller (**b**) and wireless end device [19] (**c**) in the proposed architecture

– Characteristics of the connected smart devices, such as: supported radio access technologies and types of services the devices are requesting. In order to obtain up-to-date information about topology of the ad-hoc connected devices, link layer mechanism in each device could be used to periodically broadcast beacon messages for learning neighbour's information. Beside neighbour lists, for the purpose of high-mobile vehicular applications, the controller may also store road map of the environment and information regarding the position and speed of the vehicles involved [19].

The Fog orchestration is performed according to business polices defined by application service providers. For example, the policies may specify: requirements in terms of computing and memory resources, requirements in term of bandwidth and delay for different classes of subscribers, thresholds for load balancing, privacy rules etc. [1]. These policies are stored in SDN controllers and the Fog nodes hosting the provider's application. The end-device connects to application process running on the Fog computing instance that covers the location of the end-device. If the end-device is mobile and enters the new

region, it will connect to the new Fog node that runs the same application process on the same level of network hierarchy as the previous one. SDN controller provides dynamic, policy-based management of Fog services. It can track the moving devices and predict their potential destinations in the near future. This enables seamless handover to a new Fog node at the network edge. Also, there is a possibility that the Fog computing instance located in the new region cannot meet QoS requirements of the end-user. In that case, new computing instance needs to be instantiated on a platform with the matching capabilities. Procedure of creating a new computing instance is far from trivial, because it requires: resource reservation, copying of application data, setting-up the instance configuration and dissemination of the new flow rules in the network. Therefore, the role of SDN controller is crucial to timely detect and react to threat of policy violation based on the up-to-date view of the system state. High workload can also trigger dynamic scaling of the application if that is specified in the business policy. For example, when a load balancing threshold in terms of maximum number of users, connections or CPU load is reached, new on-demand Fog instances could be created. To distribute the workload over them, SDN controller splits the coverage region of the overloaded application process in multiple smaller sub-regions. The number of sub-regions corresponds to the number of newly created Fog instances of the same process. Similarly, when these nearby processes at the same network hierarchy level become under loaded, their coverage regions are merged together into a single coverage area, and all the processes except one for the merged coverage are terminated [20].

Beside Fog orchestration, SDN controller performs traffic control as OpenFlow controller and connectivity management for IoT devices. Since IoT application in large-scale network environments (e.g. smart city applications) generate enormous amounts of data flows, the SDN control plane needs to be partitioned among multiple physical controllers to avoid scalability and reliability implications. As can be seen from Fig. 5, single SDN controller covers region with multiple Fog nodes because it is placed on a higher level of network hierarchy. The network partitions are interconnected by OpenFlow switches to enable exchange of data between controllers. This is necessary for scheduling traffic flows between IoT devices located in different partitions. For robustness reasons, we assumed that some control tasks may be delegated to local SDN agents running on Fog nodes and SDN-enabled IoT devices, as proposed in [18]. For example, Fog nodes at the network edge may control IoT device to IoT device multi-hop wireless communication in their coverage region based on their local knowledge and policy rules obtained from the controller. On the other side, the controller calculates the other routes in the system, such as inter-region routes and routes towards the other autonomous systems for data intended for the Cloud. Note that support for time-critical IoT tasks requires flow rules installed in advance, which reduces controller's efficiency in resource allocation. However, granular traffic control offered by OpenFlow can be exploited to separate emergency traffic and delay-insensitive traffic. Therefore, the first class could be scheduled always with the highest priority over the proactively installed routes, while the route calculation and resource reservation for the other traffic class could be done in a reactive manner.

The edge Fog nodes are envisioned to regularly inform the SDN controller about the capabilities and position of IoT devices that are being served. In this way, controller can build an entire connectivity graph and periodically run optimization algorithms to provide better utilization of network resources and enhance QoS. For example, if SDN controller discovers that the network load had become unbalanced because proactive routing results in traffic focusing on some selected nodes, it can start a rerouting process to improve network utility and reduce congestion. In addition to routing optimization, the controller is

supposed to perform connectivity management on the time-window basis. Algorithm for optimal access point selection in multi-network IoT environment has been proposed in [22]. Such an algorithm can be implemented on SDN controller to perform access point assignment for set of newly joined devices based on: the current multi-network capacity in the controlled partition, the supported radio access technologies and the types of services the devices are requesting. The assignment process is triggered at the end of each time-window, and aims to find a set of access points that can satisfy bandwidth requirements of the IoT devices and guarantee optimal system performance. If the algorithm outputs the assignment configuration different than the existing one, SDN controller starts the suggested handover procedures between the heterogeneous access networks.

An architecture of SDN-enabled sensor node is shown in Fig. 5c. Main component introduced in the architecture is a software OpenFlow switch. Application traffic always goes through this component before being sent, which allows the control plane elements (i.e. Fog nodes at the edge and SDN controllers) to identify the access of traffic flows into the network [19]. The number of wireless interfaces on the sensor node depends on services for which node is intended. Note that we considered only IP addressed sensor nodes as the elements of SDN data-plane. However, there are other solutions for implementing SDN in WSN networks as well [23, 24]. Beside SDN module, each sensor node runs a daemon process which implements some of the traditional ad-hoc routing protocols (e.g. AODV—Ad hoc On-Demand Distance Vector [25]). Both, the SDN control entity and the AODV daemon, have access to the node's flow table. AODV process is activated when no route to the SDN control entity is known. That is the case when sensor node joins the network or when control communication is lost due to bad conditions on the wireless channel.

## 4 Use-Cases

Rapid evolution of the IoT brings a number of use cases of interest that could benefit from the concepts of SDN and Fog computing. In this Section we chose some of them to illustrate the potential of the proposed system architecture.

### 4.1 Smart Transportation

Smart transportation represents one of the markets of the primary importance for IoT. The rationale for the adoption of IoT in this field are several, from social to economic. Some common services refer to traffic management, safety and infotainment. Traffic management services analyse traffic behaviour and events in order to optimize overall road capacity, reduce travel time and minimize the ecological footprint by smartly routing vehicles and coordinating traffic lights. Safety services aim to reduce accidents for pedestrians and vehicle occupants. The infotainment services are focused on providing classic IP informative and entertaining applications like Web browsing, video streaming, e-mail and social networks [20].

Different classes of transportation services often use the same data sources (e.g. sensors on the roads, traffic cameras, passing vehicles and so on) and networking infrastructure. However, importance of each service is not the same for the end user. Therefore, when the network is congested traffic flows should not be treated in the same manner. Without mechanisms for service differentiation and ability to provide real-time delivery today's IoT architecture limits efficiency of the existing services and hinders implementation of new

ones. On the other side, with SDN and Fog computing both of the mentioned requirements could be met. We will elaborate this on the example of STL (Smart Traffic Light) system. STL systems use a large number of distributed sensors to measure distance, speed and direction of vehicles and detect the presence of pedestrian and cyclist crossing the street. Collected sensor measurements are mainly used for three purposes: (a) accident prevention; (b) detection and offloading of congested sites; (c) long-term analysis of the system efficiency. These three tasks are significantly different in terms of delay-sensitivity. The last one can tolerate delays in data delivery, while the first two require immediate or near real-time reaction. If all the measurement data are sent to the Cloud for processing, support for real-time services is very questionable. When STL application detects risk of vehicle collision with pedestrians and cyclist, to be effective, the control system has to send notification to the approaching vehicles within milliseconds [1]. The need for fast decision making at the network edge (i.e. Fog computing) is strongly emphasized in this scenario, and has been discussed in detail in [1]. Beside Fog computing, the presented system model can also exploit benefits of SDN to dynamically assign higher priority to some traffic flows in emergency situations, and hence guarantee low-latency.

Support for real-time decision-making provided by Fog computing could bring autonomous cars soon in reality. Tesla Motors and Google are investing a lot of efforts in developing a software that will allow "hands free" operation of the car [26]. These cars will not require human involvement in control of the primary driving features such as brakes, acceleration and steering. With Fog infrastructure at the network edge, the cars and sensing devices deployed along the road will be able to interact in real-time [17]. When it comes to delivery of infotainment services in VANET networks, it could be greatly enhanced with SDN technology. The most popular infotainment applications include various forms of audio/video streaming. Delivery of this kind of data is very challenging not only due to strict delay requirements, but also due to high level of mobility and topology fluctuations [19]. Based on road map and detailed information about the vehicle position, speed and moving direction, SDN controller can make timely decisions about deployment of new service instances at the Fog nodes. In this way, impact of mobility on application performance could be minimized. On the other hand, the proposed system architecture can help in reducing duplication of the streaming traffic. For example if blue and red car in Fig. 6 are requesting the same data stream, SDN controller has enough knowledge to make optimal routing decision, i.e. to use only one traffic flow from the streaming server to the red car, and then to serve blue car from the red car.

### 4.2 Video Surveillance

Video surveillance is an important component of smart cities. Largely-distributed cameras in a city or along the road bring security services on a higher level, providing strong sense of assurance to the public. There are two advantages of using the proposed IoT architecture for video surveillance. The first is that SDN controller can make resource allocation and routing decisions based on QoS requirements. Thus, it will always seek for routes that can meet bandwidth requirements of video flows. The second advantage is that local, resource rich Fog nodes, can provide real-time processing of video frames and send notification to end-user when some event is detected. This significantly reduces bandwidth consumption in the network, while increases application efficiency (e.g. detection of criminal activity). In [21], authors proposed the hierarchical design of Fog application for vehicle tracking. Their application is organized in three processes which take place at different levels of the network hierarchy. IP camera runs the first process. If the camera is capable to perform

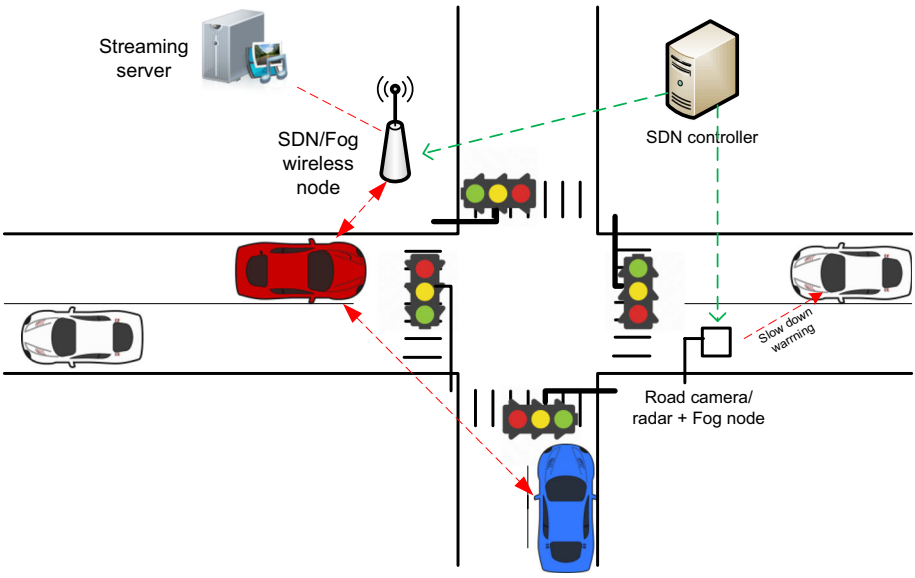


**Fig. 6** Use-case of smart transportation

motion detection, it will not send the video to the parent Fog node unless motion is detected. The parent Fog node is placed at higher level of network hierarchy, and hence have much wider scope. It runs process that identifies vehicles in the video scenes and detects their position and licence plates. If a police had issued a search for the detected vehicle, information about vehicle location is recorded into database and notification is sent to application process at the highest level of hierarchy. However, if the video image is not clear enough to enable detection of the licence plate number, the Fog node may send pan-tilt-zoom (PTZ) command to camera. Interaction with PTZ module requires very low latency, which cannot be provided if video processing is done at remote Cloud server.

### 4.3 Precision Agriculture

Precision agriculture takes advantage of advanced information and communication technologies to address today's agricultural issues such as the need to balance productivity with environmental concerns. The applications of this scenario are facilitated by ad-hoc wireless sensors and actuator networks (WSANs), deployed to measure/monitor specific parameters of the environment and enforce control decisions. The data obtained from sensor nodes are used by the Cloud application to make intelligent control decisions that should yield better and more crops through optimal application of water, pesticides and fertilizers. With Fog node deployed at the network edge, the local application instance can process the collected data, control the measurement process itself, the stability and the oscillatory behaviors and issue commands to actuators (e.g. to irrigation valves) in real-time. The Fog application can also decide to reject packets carrying redundant information, and send the aggregated data to the Cloud for long-term analysis. Local SDN controller could be run as one of the application on the Fog node. This application would be responsible for automatic configuring of WSN and optimal management of energy constrained sensor nodes with limited communications abilities [27].

Need for processing at the network edge has been more pronounced with the recent emergence of UAVs (Unmanned aerial vehicles) at farm market. So far, UAVs has been mostly used in agriculture to collect sensor data, scan plants for health problems, and locate disease out brakes. These information are than used by farmers to provide only the needed pesticide or nutrient to each plant. However, modern trends go towards multi-UAV deployment. Beside UAVs that perform monitoring of the crops, special types of UAVs are designed that can apply pesticides with very high accuracy according to spraying logistic [28]. Multiple simultaneously operating UAVs impose the need for centralized coordination of the individual tasks. In the proposed IoT architecture Fog node could play role of Internet gateway and UAV coordinator in the same time. It could process various types of multimedia and scalar data collected by UAVs, recommend optimal crop treatment and send control commands to aerial (UAVs robots) and ground actuators.

## 5 Conclusion

In this paper, we have proposed the architecture for IoT, which relies on two emerging technologies: SDN and Fog computing. The proposed architecture is designed in the way to support a high level of scalability, real-time data delivery and mobility. Fog computing platform is considered as the appropriate platform for IoT due to its capability to resolve problems related to latency for services that require fast analysis and decision-making. On the other hand, SDN introduces logically centralized control plane, which allows the implementation of sophisticated mechanisms for traffic control and resource management. Such a network design could be of vital importance to address increasing capacity demands in IoT environments where an enormous number of Internet-connected devices is expected. While the mentioned benefits have been widely recognized by the research community, this paper discusses the main challenges that hinder widespread adoption of each of the two technologies individually. The proposed IoT architecture addresses these challenges by combining SDN and Fog computing together in one system and adapting them to each other. In particular, the functionality of Fog orchestration is delegated to SDN controller in order to achieve higher efficiency, while SDN scalability issue is relaxed by delegating some controller's tasks to Fog nodes. Benefits of using our proposed architecture have been illustrated by several use-cases that range from theoretical visions to existing services.

Plan for future work is to design centralized control logic for orchestration of Fog services and to evaluate performance of the proposed solution in the appropriate simulation environment.

**Acknowledgments** This work has been supported by the EU FP7 project Fore-Mont (Grant Agreement No. 315970 FP7-REGPOT-CT-2013) and the BIO-ICT Centre of Excellence (Contract No. 01-1001) funded by Ministry of Science of Montenegro and the HERIC project.

## References

1. Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog computing: A platform for Internet of things and analytics. Big data and internet of things: A roadmap for smart environments. *Springer International Publishing*, *546*, 169–186.
2. Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: Concepts, applications and issues. In *Workshop on Mobile Big Data* (pp. 37–42).

3. Open Networking Foundation. Software defined networking: the new norm for networks. Web white Paper. Accessed 15 Feb 2016. https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.
4. Stojmenovic, I., & Sheng, W. (2014). The fog computing paradigm: Scenarios and security issues. In *Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 1–8).
5. Cisco. The Internet of Things: How the next evolution of the internet is changing everything. Web white paper. Accessed 25 Feb 2016. https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.
6. Fazio, M., Bessis, N., & Villari, M. (2015). Advances in service-oriented and cloud computing. Preface of CLIoT. *Springer International Publishing*, *58*, 73–75.
7. Botta, A., De Donato, V., Persico, V., & Pescape, A. (2014). On the integration of cloud computing and internet of Things. In *International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 23–30).
8. Tomovic, S., Prasad, N., & Radusinovic, I. (2015). Performance comparison of QoS routing algorithms applicable to large-scale SDN networks. In *International Conference on Computer as a Tool* (*EUROCON*) (pp. 1–6).
9. Tomovic, S., Pejanovic-Djurisic, M., & Radusinovic, I. (2014). SDN-based mobile networks: Concepts and benefits. *Wireless Personal Communications*, *78*(3), 1629–1644.
10. Tomovic, S., Prasad, N., & Radusinovic, I. (2014). SDN control framework for QoS provisioning. In *Proceedings of Telecommunications Forum TELFOR* (pp. 111–114).
11. McKeown, N., Anderson, T., Balakrishnan, H., Peterson, L., Rexford, J., Rexford, J., et al. (2008). Open-Flow: Enabling innovations in campus networks. *ACM SIGCOMM Computer Communication Review (CCR)*, *38*(2), 6974.
12. Trivisonno, R., Guerzoni, R., Vaishnavi, I., & Soldani, D. (2015). SDN-based 5G mobile networks: architecture, functions, procedures and backward compatibility. *Transactions on Emerging Telecommunications Technologies*, *26*, 8292.
13. Tomovic, S., Pejanovic-Djurisic, M., Yoshigoe, K., Maljevic, I., & Radusinovic, I. (2014). SDN-based concept of QoS aware heterogeneous wireless network operation. In *Proceedings of Telecommunications Forum TELFOR* (pp. 27–30).
14. Mohammad, A., & Eui-Nam, H. (2014). Fog computing and smart gateway based communication for cloud of things (pp. 464–470).
15. Luan, T. H., Gao, L., Xiang, Y., Li, Z., & Sun, L. (2015). Fog Computing: Focusing on mobile users at the edge. ArXiv:1502.01815v1 [cs.NI].
16. Peter, N. (2015). Fog computing and its real time applications. *International Journal of Emerging Technology and Advanced Engineering (IJETAE)*, *5*(6), 266–269.
17. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Workshop on Mobile Cloud Computing* (*MCC*) (p. 1316).
18. Truong, N. B., Lee, M. G., & Ghamri-Doudane, Y. (2015). Software defined networking-based vehicular ad-hoc network with Fog computing. In *IEEE International Symposium on Integrated Network Management* (pp. 1202–1207).
19. Ku, I., You, L., Gerla, M., Ongaro, F., Gomes, R. L., & Cerqueira, E. (2014). Towards software-defined VANET: Architecture and services. In *13th Annual Mediterranean Ad Hoc Networking Workshop* (pp. 103-110). MED-HOC-NET.
20. Hong, K., Lillethun, D., Ramachandran, U., Ottenwalder, B., & Koldehofe, B. (2013). Mobile fog: A programming model for large-scale applications on the internet of things. In *ACM SIGCOMM workshop on Mobile cloud computing* (pp. 15–20).
21. Zhijing, Q., Denker, G., Giannelli, C., Bellavista, P., & Venkatasubramanian, N. (2015). A software defined networking architecture for the Internet-of-Things. In *IEEE Network Operations and Management Symposium* (pp. 1–9).
22. Wu, D., Arkhipov, D., Asmare, E., Qin, Z., & McCann, J. (2015). UbiFlow: Mobility management in urban-scale software defined IoT. In *Proc. of the 34th IEEE Conference on Computer Communications* (pp. 208–216). INFOCOM.
23. Luo, T., Tan, H.-P., & Quek, T. (2012). Sensor OpenFlow: Enabling software-dened wireless sensor networks. *IEEE Communications Letters*, *16*(11), 1896–1899.
24. Costanzo, S., Galluccio, L., Morabito, G., & Palazzo, S. (2012). Software defined wireless networks: Unbridling sedans. In *European Workshop on Software Defined Networking* (p. 16). EWSDN.
25. Perkins, C., Belding-Royer, E., & Das, S. (2003). Ad hoc on-demand distance vector (AODV) routing. IETF. RFC 3561. Accessed 15 Feb 2016.
26. Tesla Motors. Web page. Accessed 15 Feb 2016. http://my.teslamotors.com/it_IT/forum/forums/elon-musk-talks-google-bring-driverless-tech-tesla-cars.

27. Tomovic, S., & Radusinovic, I. (2015). Performance analysis of a new SDN-based WSN architecture. In *Proc. of 23Rd Telecommunication Forum TELFOR* (pp. 99–102).
28. Robodrone. Web page. (2016) http://www.robodrone.com/. Accessed 15 Feb.

**Slavica Tomovic** was born born on 05.02.1991 in Montenegro. In 2012 she received BSc degree in Electronics, Telecommunications and Computer Science from Faculty of Electrical Engineering in Podgorica, University of Montenegro. From the same faculty she received MSc degree in Telecommunication (2015). Currently, Slavica is Ph.D. student on Faculty of Electrical Engineering, University of Montenegro. She is also teaching/research assistant on the same faculty. Her main research interests are in the areas of software-defined networking, quality of service (QoS) management and architectures, Internet of things (IoT) and 5G wireless network design.

**Kenji Yoshigoe** is a Professor and Chair of Computer Science at the University of Arkansas at Little Rock (UALR), the Director of UALR Computational Research Center, and the Director of NSA/DHS Designated National Center of Academic Excellence in Cyber Defense Education (CAECDE). He received his Ph.D. degree in Computer Science and Engineering from the University of South Florida. His current research explores the reliability, security, and scalability of various interconnected systems ranging from tightly-coupled high performance computing systems to resource-constrained wireless sensor networks to dynamically evolving social networks.

**Ivo Maljevic** received the B.Sc. degree from the University of Montenegro in 1991, the M.Sc. degree from the University of Belgrade in 1995, and the Ph.D. degree from the University of Toronto, Canada, in 2004, all in electrical engineering. Currently he is an Adjunct Professor at the University of Toronto and a senior member of TELUS' technology strategy team. His areas of expertise include LTE/WiMAX/CDMA radio access networks, signal processing, and digital communications theory.

**Igor Radusinovic** received the Telecommunications Engineering degree by the University of Montenegro in 1994. He received the MSc and Ph.D. degree by the University of Belgrade, Serbia, in 1997 and 2003 respectively. He is a Full professor in Telecommunications networks and Switching systems at the Faculty of Electrical Engineering, University of Montenegro. From 2009 to 2011 Prof. Radusinovic was Deputy Minister for Science, Research and Technological Development in the Ministry of Education and Science of the Government of Montenegro. He was Montenegrin representative to the Board of Governors of the Joint Research Centre of the European Commission (JRC EC) and European Research Area Committee (ERAC). Prof. Radusinovic was Chairmen of Montenegrin Council for Science and Research. He is a Member of the Managing Board of University of Montenegro and President of the Managing Board of. Montenegro Post He has published as an author or co-author more than 120 papers in international and national scientific journals and international and regional conferences. He participated in a number of international (FP7, COST action) and national research teams and projects, as well as bilateral research cooperation. Prof. Radusinovic has considerable industry and operating experiences working on many strategic studies, regulatory issues, development strategies and technical solutions in area of ICT. Research interest of Prof. Radusinovic is focused, but not limited, on: packet switching systems, telecommunications network theory, congestion control algorithms, software defined networking and 5G.