Coordinating heterogeneous IoT devices by means of the centralized vision of the SDN controller

Jaime Galán-Jiménez[1], Javier Berrocal[1], Jose Garcia-Alonso[1], Carlos Canal[2], and Juan M. Murillo[1]

[1] University of Extremadura, Spain
{jaime,jberolm,jgaralo,juanmamu}@unex.es
[2] University of Málaga, Spain
canal@lcc.uma.es

Abstract The IoT (Internet of Things) has become a reality during recent years. The desire of having everything connected to the Internet results in clearly identified benefits that will impact on socio economic development. However, the exponential growth in the number of IoT devices and their heterogeneity open new challenges that must be carefully studied. Coordination among devices to adapt them to their users' context usually requires high volumes of data to be exchanged with the cloud. In order to reduce unnecessary communications and network overhead, this paper proposes a novel network architecture based on the Software-Defined Networking paradigm that allows IoT devices coordinate and adapt them within the scope of a particular context.

Keywords: Internet of Things, Situational-Context, Software-Defined Networking

1 Introduction

During the last few years we have seen how data traffic has dramatically increased, and this trend will probably continue over the next few years. Some reports forecast [11] that the global IP traffic in 2020 will nearly triple the traffic in 2015. As these reports indicate, this increase is mainly due to the high penetration of smartphones and the massive use of cloud technologies [13].

This increase has a direct consequence: network infrastructure supporting this traffic needs to be improved in order to maintain acceptable levels of QoS (Quality of Service). To that end, new network equipment, higher bandwidth connections and wider mobile coverage are required [11]. Currently, most of the burden to improve this infrastructure rests on the telco operators side.

This improvement, until now, has been mainly funded with the users' quot. However, there are other actors who benefit from this infrastructure, such as applications and cloud service providers. For example, Google or Facebook are storing their users' information in order to be able to send them personalized advertisements, getting a huge profit with that [1]. Nevertheless, they are not paying operators to improve their network infastructure.

While this is currently a concern, it will become a real problem as IoT (Internet of Things) devices and WoT (Web of Things) systems are deployed. The connected smart devices are expected to be continuously sending and receiving information from cloud environments by nature. Therefore, the massive amount of data produced and exchanged by things in IoT networks brings to light the need to restructure and redesign both network and data storage systems [7].

There are new research lines trying to minimize the exhaustive tasks performed in the cloud, bringing the storage and computing of the data to near-user edge devices or even to the user's end device, such as Fog Computing [10]. Or, for example, in [16], a distributed mobile computing model is developed, in which the information is stored and computed in the user's mobile devices. Likewise, the Mobile Agents [9] are programs that can autonomously migrate between the nodes of a network, or different networks, resuming their execution there. In this way, passing from Cloud Computing to Edge-Fog Computing would reduce communication interactions among connected IoT devices.

During the last few years, the authors of this paper have been working on the Situational-Context [8]. This concept defines a proper way to deal with the expected increase in traffic. The idea behind it is to locally analyse the contextual information that exist at a particular time and place in order to predict, in real-time, the expected behaviour of IoT devices and WoT systems.

In order to apply this concept on a real networking environment, we propose to use the new Software-Defined Networking (SDN) paradigm. The centralized nature of SDN networks and the separation between data and control planes make this approach particularly appealing. However, a set of challenging issues must be faced when applying SDN on IoT networks (SDIoT, Software-Defined Internet of Things), especially the ones related to scalability (huge number of devices), heterogeneity (diversity in wireless technologies) and security (entry points for malwares).

This paper proposes the Context-SDIoT network architecture, where the Situational Context concept can be easily deployed on IoT networks by means of the SDN paradigm. In this architecture, each Situational-Context is composed of a set of IoT devices that share the same spatio-temporal area and a controller that is responsible of all control decisions within that area (context). The controller is able to know, at each time, specific information retrieved from each IoT device and identify the required strategies to meet the user's needs. With this approach, data traffic generated and exchanged among IoT devices can be computed within the scope of the proper context, avoiding unnecessary communications and coordination tasks with the cloud.

To describe the proposed model and the benefits it provides, this paper is organized as follows. Section 2 motivates the proposed architecture. Section 3 details the proposed Context-SDIoT network architecture. Finally, Section 4 contains our conclusions and future works.

2 Motivation

This section details the Situational Context concept, its main open issues and how SDN networks could be used to address some of them.

2.1 Situational Context

During the last few years, different approaches, such as Ambient Intelligence [12] and Context-Aware [17], have been working on identifying the users' needs and preferences. This information is then used to semi-automatically adapt the applications' behaviour to the users' context, improving the users experience.

In the last few years, the increased computing and storage capabilities of smartphones and smarthtings, allowed us to propose a new context-aware computing model. This model is called Situational-Context [8]. The Situational-Context is a way to analyse the conditions that exist at a particular time and place in order to predict, at run-time, the expected behaviour of WoT systems. This model exploits the smart devices' capabilities to gather, store and locally compute the contextual information in order to construct its virtual profile, and the virtual profile of its owner. Thus, the surrounding devices can reuse it to adapt themselves to the user's preferences. To that end, the Situational-Context defines that the virtual profile of an entity (thing or person) should contain at least the following information:

- A *Basic Profile* containing the dated raw contextual information with the entity's status, the relationships with other devices and its history.
- *Social Profile*. This profile contains the results of high level inferences performed over the Basic Profile.
- The *Goals* detailing the status of the environment desired by the entity. These Goals are deducted from the Basic and Social Profiles at run-time.
- The *Skills* or capabilities that an entity has to make decisions and perform actions capable of modifying the environment and aimed at achieving Goals.

Considering environments in which there are different entities and each of them has a virtual profile, the Situational-Context can be defined as the composition of the virtual profiles of all the entities involved in a particular situation. Once the Situational-Context is composed, the ways in which the entities will better satisfy the goals are identified from the Situational-Context itself. Therefore, the Situational-Context provides a higher level of automation of smart things with people.

However, this concept have some important issues that should be solved. Some of the most important ones are how the virtual profiles are shared between smart devices (since this implies the transfer of a large amount of data) and where the Situational-Context should be computed (i.e. in the end devices, in a server, in a network device) without this entailing a great overload of the network. In order to give an answer to all these issues, we propose to use SDN and SDIoT networks. Next subsection presents some of the most important challenges of using SDN over IoT environments.

2.2 Software-Defined Networks

Software-Defined Networking (SDN) is an emerging networking paradigm that gives hope to change the limitations of current network infrastructures. First, it breaks the vertical integration by separating the network's control logic (control plane) from the underlying routers and switches that forward the traffic (data plane) [20]. Second, with the separation of control and data planes, network switches become simple forwarding devices and the control logic is implemented in a logically centralized controller, simplifying policy enforcement, network (re)-configuration and evolution [18].

A fundamental characteristic of SDN is the logically centralized, but physically distributed controller component. The controller maintains a global network view of the underlying forwarding infrastructure and programs the forwarding entries (actions to be done) based on the policies defined by network services running on top of it [5]. A standardized programmable interface, namely OpenFlow [2], was adopted by the industry in order to program multiple types of forwarding devices. Next section describes the main challenges of applying the SDN paradigm on the network environment considered in this paper, i.e. IoT networks.

2.3 Challenges of SDIoT Networks

SDIoT networks are able to perform customized computations according to specific network requirements. A SDN controller, which is responsible of applying the network logic in the SDIoT scenario, could be used to control the data transfer among devices and the computation of the Situational Context. Although programmability and flexibility allowed by the emerging SDN paradigm [20] on wired environments can be easily exploited on wireless networks, there are challenging tasks that must be carefully studied to let this transition be viable. The most important issues are:

- **Scalability and heterogeneity.** One of the key points regarding the scalability is the one related to the number of coordinated controllers that are needed to satisfy the IoT demands [19]. The huge number of connected devices and their heterogeneity requires the use of a distributed scheme coordinating different physical controllers. Therefore, the proposal of a distributed-centralized control plane could be a feasible option to deal with the inherent dynamic nature of IoT networks [19].
- **Security.** IoT networks are more sensitive to security than traditional networks. Although the inclusion of centralized controllers with a global network view could lead to improve network performance, it can also attract the attention of attackers. In this way, novel security mechanisms must be proposed [15]. The flow-based nature of SDN forwarding, where all the packets belonging to a flow follow the same path for each source-destination pair, strengthens the importance of defining complex device-access rules based on access control lists, advanced firewalls and developing algorithms to detect specific security threats.

- **Mobility.** Traditional implementations of SDN technology on wired networks are not very close for handling the dynamic needs of pervasive IoT applications, especially the ones related to mobility. Inherent ubiquitous nature of IoT devices results in frequent changes between access networks. Continuous negotiations between controllers as IoT devices enter and leave the network could lead to an increase in network overhead. In order to tackle the mobility issues in SDIoT networks, each controller in the network must have a global view of the mobility of these devices [14].
- **Quality of Service.** Maximizing the utilization of SDIoT networks requires fine-grained QoS support for differentiated application requirements. Traffic classification, prioritization and analysis of different multimedia applications for streaming become challenges to propose solutions providing acceptable values of QoS [6]. In order to satisfy these distinct QoS requirements, distributed controllers could offer smart routing, scheduling, and virtualization solutions. In this way, IoT use-case with conflicting requirements can be isolated and treated independently, providing specific network resources for each of them.

Once the main challenges of applying the SDN paradigm on IoT environments have been introduced, next section describes the proposed architecture for implementing the Situational Context by means of the SDN paradigm.

3 Context-SDIoT Architecture

This section proposes a novel network architecture to apply the concept of Situational-Context on a SDIoT environment. Three different approaches can be considered:

- *Fully-centralized solution.* In this situation, a single controller is the responsible of performing all the tasks associated with a specific context. This approach allows the controller to have a (near) real-time view of the network state. Nevertheless, network performance must be analysed in terms of network overhead and controller overload.
- *Fully-distributed solution.* This case considers that all the computation tasks are carried out by end devices in a distributed manner. This option has the drawback of the extremely limitation of end devices in terms of processing and storage capabilities (packet buffering), especially if they are sensors. It is then doubtful whether they can support such a level of programmability.
- *Hybrid solution.* The latter case is to divide the set of actions to be done into two different subsets. Low complexity tasks could be directly carried out by end devices, whilst more complex tasks would be relegated to be done by the controller. This selection of tasks could be performed adaptively depending on the device's resources and capabilities.

With the aim of starting to evaluate the feasibility and effectiveness of these approaches, this paper focuses on the fully-centralized solution where a single controller is able to manage a context. In order to describe it, a set of concepts related to the main components of the architecture must be previously defined:

- **IoT device:** General IoT device that has, at least, one wireless network interface, independently of the specific nature of the wireless technology. Each IoT device is able to create its Basic Profile (BP) and its Social Profile (SP). Besides both profiles, a set of Goals (G) and a set of Skills (S) are respectively inferred and provided by the IoT device.
- **Context-LAN**: Local Area Network (LAN) intrinsically associated to a specific geographic area which is composed of a set of switches, access points and IoT devices that all together form a particular context.
- **SDIoT controller**: Centralized device that belongs to a particular Context-LAN and is responsible of all control decisions among the set of IoT devices connected to the proper Context-LAN. The controller has a global network view and is able to request information about IoT devices and retrieve statistics from switches both in a proactive and in a reactive way.

Figure 1 depicts the proposed Context-SDIoT network architecture. A set of heterogeneous IoT devices (e.g. smartphone, air conditioner and lighting equipment) are connected to the Context-LAN depicted by the blue area using specific wireless technologies. Moreover, the logically centralized SDIoT controller is able to request and obtain specific information from the set of IoT devices that are connected to the Context-LAN it manages. In reality, physical connections are made between the controller and the different access points (e.g. WiFi, WiMAX, cellular) and switches in the Context-LAN. Green dashed lines represent secure channels connecting the SDIoT controller and the network equipment and OpenFlow protocol [2] is used for this purpose.

One step further, Figure 1 also shows the full process that is carried out when a new IoT device joins a particular Context-LAN. When a new IoT device enters an area that is managed by a SDIoT controller ①, there is an association between the proper IoT device and one of the different types of access points belonging to the Context-LAN ②. Once this association has been performed, an OpenFlow notification message is generated by the access point and sent back to the controller in order to let it know that a new device has been connected. Therefore, an unique id must be included in the payload of such message ③. Finally, the SDIoT controller is able to gather information from the newly joined device to know its virtual profile, as well as its goals and skills. In order to do this, the controller creates a new OpenFlow message directed to the same access point requesting information related to the IoT device with the same id that was included in the previously received message ④.

Note that mobile IoT devices are prone to move quickly between Context-LANs, leaving the one they are connected to and entering new ones, in a short period of time. To reduce network overhead and save the communication cost, when a previously associated IoT device enters its Context-LAN again, the controller does not need to initiate another joining process and request information from that device. Instead, the controller can localize the corresponding stateful information using fast lookup on its state table and update it only if necessary.

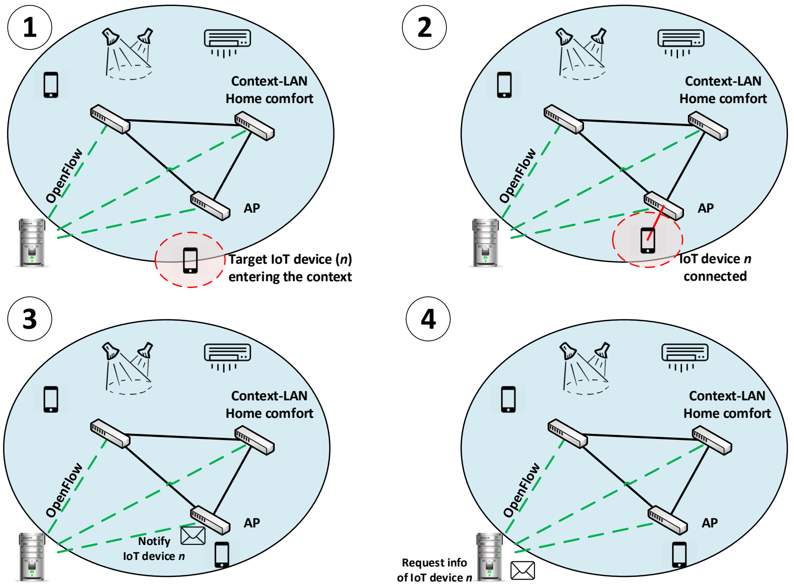

Figure 1: Context-SDIoT network architecture. IoT device joining process.

At this point, we define a threshold value, namely *assoc_thresh*, that refers to the maximum time that the stateful information of a particular device remains in the controller table. After expiring, this entry is removed and if the IoT device re-enters the Context-LAN again, it would be necessary to start a new joining process. The value for *assoc_thresh* is dynamically adjusted by the controller based on the history of the particular Context-LAN it is responsible for.

With the present work, our aim is to position an initial approach where the concept of Situational-Context is included in the IoT networking scope. Through the use of the emerging SDN paradigm, the SDIoT controller will be able to satisfy the set of goals of each IoT device within the proper context. For this purpose, both the set of goals and the set of skills of each IoT device must be known by the controller at each time. In this way, specific applications according to these skills and goals can be optimally implemented and executed by the controller when necessary. In the following, the generic problem of managing a Context-LAN by a SDIoT controller is formalized and a use case related to home comfort is finally explained.

3.1 Problem Formulation

Let us consider a generic Context-LAN, $L = (D, C)$ with a set of $d_i \in D$ connected IoT devices managed by a single SDIoT controller, C. Each IoT device, $d_i \in D$, is a 4-tuple of type $d_i = \{BP_i, SP_i, G_i, S_i\}$, with BP_i as the *Basic Profile* of i-th IoT device, SP_i as its *Social Profile*, $G_i = \{g_1^i, g_2^i, ..., g_k^i\}$ as the set of

Goals that device d_i pursues and $S_i = \{s_1^i, s_2^i, ..., s_p^i\}$ as the set of *Skills* that d_i is able to perform, respectively. Note that, although each device has, at least, a BP_i, any of the three remaining components could not be required by a specific IoT device, i.e. $SP_i = \emptyset$, $G_i = \emptyset$, $S_i = \emptyset$.

Being $S = \{S_1, S_2, ..., S_n\}$ the full set of skills retrieved from the n IoT devices connected in L, the controller must be able to assess $|S|$ different objective functions. If a generic skill j of a device d_i is selected at time t, s_j^i, the specific objective function is defined as follows:

$$min \quad \sigma \sum_{d_i \in D} g_j^i \tag{1}$$

where g_j^i is the goal j of device d_i related to skill s_j^i, taken as input. The objective function for a particular skill (eq. 1) is therefore to minimize the standard deviation among the values of the set of goals related to that skill in the network. The result is finally sent by the controller to the involved devices through the use of specific OpenFlow messages and the desired goals are achieved.

3.2 Use case: Home comfort Context-LAN

In the following, a particular use case is described. Consider the Context-LAN of Figure 1 related to home comfort. Network description could be given by $L = (D, C)$, with $D = \{d_1, d_2, d_3, d_4, d_5\}$:

- d_1: Smartphone 1 with only one goal $G_1 = \{(g_1^1 = \textit{Comfort temperature} = 22°C)\}$, and no skills, $S_1 = \emptyset$.
- d_2: Smartphone 2 with two goals $G_2 = \{(g_1^2 = \textit{Comfort temperature} = 20°C); (g_2^2 = \textit{Luminance} = 100 \quad lux)\}$, and no skills, $S_2 = \emptyset$.
- d_3: Smartphone 3 with two goals $G_3 = \{(g_1^3 = \textit{Comfort temperature} = 21°C); (g_2^3 = \textit{Luminance} = 300 \quad lux)\}$, and no skills, $S_3 = \emptyset$.
- d_4: Air conditioner with no goals, $G_4 = \emptyset$, and one skill, $S_4 = \{(s_1^4 = \textit{Temperature control})\}$.
- d_5: Lighting equipment with no goals, $G_5 = \emptyset$, and one skill, $S_5 = \{(s_1^5 = \textit{Luminance control})\}$.

As we can see, there are two different skill functions to be managed and executed by controller C: $S = \{S_4, S_5\} = \{s_1^4, s_1^5\}$. The first one, s_1^4, is provided by the air conditioner, d_4, and sets the adequate comfort temperature according to goal values, g_1^1, g_1^2, g_1^3, obtained from the three different smartphones. After performing the assessments, C sends the resulting value ($temp = 21°C$) to d_4, which finally sets the temperature in the Context-LAN L.

The latter skill, s_1^5, is managed by the lighting equipment and sets the average luminance required by smartphones 2 and 3 (d_2 and d_3), since d_1 does not have the goal of setting the luminance among its set of goals, G_1. As in the previous case, resulting luminance value (200 lux) will be sent by C to d_5 in order to set the desired luminance.

In order to ease the comprehension of the proposed architecture, the use case explained above employs the average value in the objective function for the two skills considered. Obviously, this simple metric can be used in specific situations, but real IoT environments would require more complex algorithms which would take into account the importance of each IoT device and the role of each user.

4 Conclusion and Future Work

This paper proposes the application of the Situational-Context concept on IoT environments by means of the SDN paradigm. The flexibility and programmability offered by SDN can be effectively exploited to reduce unnecessary interactions between IoT devices and the cloud.

Specifically, we propose the Context-SDIoT architecture, where each situational context is composed of a set of IoT devices that share the same spatio-temporal area and a controller that is responsible of all control decisions within that area (context). The controller is able to know, at a particular time, specific information retrieved from each IoT device (profile, goals and skills) and identify the required strategies to meet the user's goals. With this approach, data traffic generated and exchanged among IoT devices can be computed within the scope of the proper context.

As future work, the proposed architecture is going to be implemented on wireless network simulators to evaluate its performance under different scenarios. In particular, we believe that the recently added OpenFlow module for OMNeT++ [3] makes this simulator particularly appealing to be used for a prototype implementation. One step further is to evaluate our proposal with realistic experiments by exploiting ORBIT wireless network testbed [4], or even customizing a set of Raspberry Pis to create and evaluate small-sized Context-LANs.

Acknowledgements

This work was supported by projects TIN2014-53986-REDT, TIN2015-67083-R and TIN2015- 69957-R (MINECO/FEDER, UE), by the Department of Economy and Infrastructure of the Government of Extremadura (GR15098), and by the European Regional Development Fund.

References

1. Google Financial Tables - Investor Relations (December 2016), `https://abc.xyz/investor/index.html`
2. OpenFlow Switch Specification. Version 1.5.0 (2016), `https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf`
3. OMNeT++ Discrete Event Simulator (2017), `https://omnetpp.org/`
4. Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) (January 2017), `www.orbit-lab.org`

5. Agarwal, S., Kodialam, M., Lakshman, T.V.: Traffic engineering in software defined networks. In: 2013 Proceedings IEEE INFOCOM. pp. 2211–2219 (April 2013)
6. Awobuluyi, O., Nightingale, J., Wang, Q., Alcaraz-Calero, J.M.: Video quality in 5g networks: Context-aware qoe management in the sdn control plane. In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. pp. 1657–1662 (Oct 2015)
7. Bernbo, S.: The Internet of Things Demands a New Data Architecture (2014), `http://compuverde.com/contact/news-room/the-internet-of-things-demands-a-new-data-architecture/`
8. Berrocal, J., Garcia-Alonso, J., Canal, C., Murillo, J.M.: Situational-context: A unified view of everything involved at a particular situation. In: Bozzon, A., Cudré-Mauroux, P., Pautasso, C. (eds.) Web Engineering - 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9671, pp. 476–483. Springer (2016)
9. Bobed, C., Ilarri, S., Mena, E.: Distributed mobile computing: Development of distributed applications using mobile agents. In: PDPTA. pp. 562–568 (2010)
10. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing. pp. 13–16. ACM (2012)
11. Cisco: Cisco Visual Networking Index: Forecast and Methodology, 20152020 (2016), `http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf`
12. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: Technologies, applications, and opportunities. Pervasive and Mobile Computing 5(4), 277–298 (2009)
13. Dinh, H.T., Lee, C., Niyato, D., Wang, P.: A survey of mobile cloud computing: Architecture, applications, and approaches. IWirel. Commun. Mob. Comput. 13(18), 536–550 (2013)
14. Dong, M., Li, H., Ota, K., Xiao, J.: Rule caching in sdn-enabled mobile access networks. IEEE Network 29(4), 40–45 (July 2015)
15. Gonzalez, C., Flauzac, O., Nolot, F., Jara, A.: A novel distributed sdn-secured architecture for the iot. In: 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS). pp. 244–249 (May 2016)
16. Guillen, J., Miranda, J., Berrocal, J., Garcia-Alonso, J., Murillo, J.M., Canal, C.: People as a service: A mobile-centric model for providing collective sociological profiles. IEEE Software 31(2), 48–53 (2014)
17. Hong, J.y., Suh, E.h., Kim, S.J.: Context-aware systems: A literature review and classification. Exp. Sys. with App. 36(4), 8509–8522 (2009)
18. Kim, H., Feamster, N.: Improving network management with software defined networking. IEEE Communications Magazine 51(2), 114–119 (February 2013)
19. Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., Shenker, S.: Onix: A distributed control platform for large-scale production networks. In: Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation. pp. 351–364. OSDI'10, USENIX Association, Berkeley, CA, USA (2010), `http://dl.acm.org/citation.cfm?id=1924943.1924968`
20. Kreutz, D., Ramos, F.M.V., Ver?ssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: A comprehensive survey. Proceedings of the IEEE 103(1), 14–76 (Jan 2015)