Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 680 – 684

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

Developing a distributed software defined networking testbed for IoT

Olivier Flauzac, Carlos Gonzalez, Florent Nolot

University of Reims Champagne-Ardenne, Laboratory CReSTIC, 51100 Reims, France

Abstract

The rapid proliferation of the Internet of Things (IoT) has led to growth in the ad-hoc networking scenario. With the recent upcoming technologies of network programmability like SDN may be effectively integrated to create a communication platform. In this work, we present the details of our preliminary study of how to determine the effectiveness of an approach to build a cluster network using Software-Defined Networking (SDN). We will provide an overview of optimum path routing protocol with cluster interaction. Our proposed scheme is a starting point for some experiments providing perspective over SDN deployment in a cluster environment for IoT. With this aim in mind, we propose a routing protocol that manages routing tasks over Cluster-SDN. By using network virtualization and OpenFlow technologies to generate virtual nodes, we simulate a prototype system controlled by SDN. Our designed testbed, is a real openflow protocol evaluation environment comprising networks and applications, which provide flexible control of framework support for large scale experiments.

Peer-review under responsibility of the Conference Program Chairs

Keywords: IoT, Software Defined Networking, Ad-Hoc networks, Cluster

1. Introduction

Today's network technology allows things and hence users, to be connected at any time at any place. With the increasing development of the internet, security threats constantly appear and the protection of data transmission has become an issue to be solved. Currently, there are more objects connected to the internet than humans in the world[1], and these generate an enormous amount of traffic (i.e., voice, video, data, etc.). All of these factors increase considerably the cost pressure on network operators, due to the emerging mobile devices and applications[2]. One of the greatest challenges concerns the security of the Internet of Things (IoT) , since it will include every object or device able to connect to wireless or wired networks[3].

In this article, we present a model to control and secure information exchanges for the IoT, based on the SDN architectures[4,5]. Firstly, the proposed model was designed to establish and secure both ad-hoc and IoT networks, in

* Corresponding author. Tel.: (+33) 3 26 91 32 15 ; fax: (+33) 3 26 91 33 97.
E-mail address: olivier.flauzac@univ-reims.fr, carlos.gonzalez-santamaria@etudiant.univ-reims.fr, florent.nolot@univ-reims.fr

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.151

order to include objects such as: sensors, tablets, smartphone, etc. Secondly, we extended the proposed architecture, and explain how flows can be routed between controllers. We demonstrate the implementation of our prototype system and evaluation results. The major contributions of this paper are:

- our work is a novel exploration of the SDN architecture to optimize the interconnection of ad-hoc and IoT applications.
- we introduce the concept of the SDN Clustered Head (SDNCH) to distribute routing functions and security rules to each edge controller.
- we introduce an implementation model with an Opendaylight controller to manage and monitor traffic from the end-users in IoT environments.

Our model is discussed later in this article, and we conclude with an outline of our vision for the SDN based solutions for ad-hoc and IoT networks.

2. SDN Cluster architecture for IoT and Ad-Hoc Network

Previously, we have proposed a Software Defined Clustered Sensor Networks (SDCSN)[10]. Clustering consists in organizing the network into groups of nodes following a hierarchical structure. Each cluster is managed by a cluster head. To develop this architecture we place the SDN controller in the cluster head. Different clustering solutions have been proposed in the literature. Some solutions propose building 1-hop clusters[14,15,17]. In those solutions, each node is at a most a distance of 1 from the cluster head, and the maximum diameter of each cluster is 2. Other solutions build k-hop clusters[18,19,20]. In k-hop cluster solutions, each node can be located at a distance at most of k from the cluster head and the maximum diameter of clusters is 2k.

Based on the approach of Model-driven Service Abstraction Layer (MD-SAL) Clustering solution active/active mode, we propose Multiple SDN Controller architecture for IoT and Ad-Hoc Networks[16]. An SDN-based architecture involves: (i) legacy interfaces (the physical layer); (ii) the SDN-compatible virtual switch (programmable layer); (iii) the SDN controller operating systems and their applications (OS layer). Ad-Hoc users will connect with other nodes through their embedded SDN-compatible switch.

Normally, a large network can not operate efficiently without some organized structure. For this reason, we propose to cluster the network and assume that each cluster head is a controller. A node state[7,8,20] can be : Simple Node (SN), Gateway Node (GN) or Cluster Head (CH). In our approach, Cluster Heads (CH) in SDCSN architecture are called SDN Cluster Heads (SDNCH). Each cluster is called an SDN Domain which is defined by :

- SDNCH is the coordinator of the domain.
- Gateway is a bridge node between Sensor Nodes and SDNCH.
- Sensor Nodes are groups of nodes in a domain together with their gateway nodes.

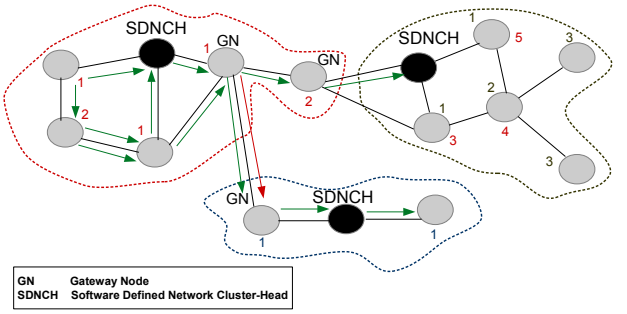

Fig. 1. Data Communication Software-defined wireless sensor networks

To deploy this architecture, SDNCH has not only to manage the domain network[6,11,12,13]. It also has to monitor and efficiently secure the domain to prevent outside and inside attacks. Moreover, this approach acts as a security guard on the edge of the domain to ensure the domain safety. Therefore, the proposed cluster can not solve problems with routing processes in distributed SDN-architecture. But by combining routing protocols and SDN, a new effective controller-routing method is proposed in the next section.

3. Routing Protocol for Distributed Cluster SDN in IoT

The main issue is the gateway node between each cluster. In an IoT network[22], a thing may not have routing capabilities, so we can not have a thing which is a node with few resources as a gateway. For the clustering architecture, we propose an SDN controller in each cluster. This controller manages and controls all traffic of nodes connected only in its domain, so-called intra-domain. In this environment, the controllers communicate with others via an inter-domain link. Previous work[23,24] propose hierarchical architecture for SDN to optimize and distribute control functions. We propose not to distribute control functions on multiple controllers but to distribute routing functions on each SDNCH.

Software defined information centric networking (SD-ICN) has already been proposed[23]. The scalable area-based hierarchical architecture (SAHA) has been used for the controller deployment in SD-ICN, in which a centralized root-controller has a global network view and the area controllers only know their local area view. The problem of zoning for distributed network optimization has been discussed[24] and the proposed approach consists of a number of slave controllers to be in charge of the zones separately, under the coordination of a master controller. Its design consists of using the clustering heuristic, the partitioning heuristic, and the assignment heuristic solutions which allows the communication and load processing between controllers.

The deployment of this architecture is based on the perspective of an Opendaylight MD-SAL Akka-based clustering solution. To select an appropriate path between nodes connected on the cluster, the process of routing flows has been indicated in (Fig. 2a). The main process flow is as follows:

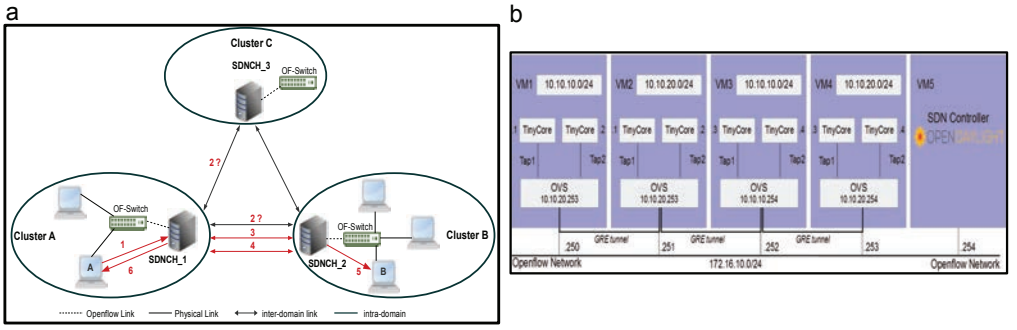

Fig. 2. (a) Distributed Routing Cluster for SDN; (b) Network design of the IoT based SDN testbed

1. Node A joins the node B, and node A sends a request to the SDNCH1;
2. SDNCH1 sends the same request to the neighbor controllers connected on the network;
3. The ones which know node B, send a replies to SDNCH1, for this example SDNCH2;
4. The flow may be installed on the SDNCH1. Node A can use the communication path via SDNCH1 for taking out of cluster the packet;
5. The routing information between SDNCH1 and SDNCH2 being set up on the inter-domain path;
6. The messages will be exchanged between both nodes.

We focus in the proposal of a new Routing Protocol for a Distributed Cluster SDN which can be used to support SDN-based inter-domain collaboration. The goal is to allow an automated routing setup of inter-domain clusters. As a result, a node can interact with other nodes located in different clusters through an inter-cluster routing protocol.

4. Experiments

We implemented an Opendaylight SDN controller using OpenFlow 1.3 protocol[21,25]. When host A wants to send an IP packet over Ethernet to host B by IP address, it needs an ARP reply from B. When a new flow from host A arrives at an OpenFlow switch, the switch forwards the packet to the controller SDN. If there is not match with the flow entry installed, the packet will be rejected. Upon receiving the reply, the routing in the controller will send the flow to the switch connected to the source host and the switch connected to the destination host. Once, the link is established, a shortest route will be set for the switch. In this way, the switch is in charge of routing the packets between host A and host B.

The implemented testbed in this paper includes a virtualized testing environment combined with vmware vsphere, qemu system, TinyCore, software-based OpenFlow switches (OVS) and Opendaylight controller. The experimental platform was developed based on SDN and that enabled users easily to design network topology via web applications. The framework of the testbed would then allocate resources to the experiment which is able to create routes between nodes, monitor requests sent to the controller and also decide policies delivered to each device. Our approach on the testbed provides an environment for IoT and ad-hoc network deployment. It allows nodes to be connected with others via one SDN-compatible OVS switch and at the same time this switch is controlled by an SDN controller.

To evaluate the performance of Openflow, we first describe the experimental scenario deployed on the SDN testbed. It consists of Opendaylight Helium-SR4 and five virtual machines (VM) connected to the same network. The PC running the controller uses Ubuntu 14.04 operating system and was provided with 2 virtual CPUs and 16GB of RAM. Each VM had 8 virtual CPU and 16GB of RAM, using the Debian 8.0 image with OVS Open vSwitch pre-installed. We use KVM for machine virtualization over the VMs, to run TinyCore Linux 3.16.6 with 48MB of RAM.

This testbed is an excellent way to setup OpenFlow test platform. We have developed a shell script to simplify work in such an environment through an SDN controller to control behavior of the whole network. It is important to emphasize that this script enables control of each device and communicates with them via OpenFlow. In this way, each virtual machine engages one script to create the required network over the SDN environment. The script consists in launch hosted virtual machine monitor and the instructions to connect OVS with flows rules. It provides a means for remote access and the ability to handle VMs running on Hyper-V hosts. Once the setup and configuration were completed, we installed a TinyCore image into the VMs to use when starting Hyper-V hosts. For Ethernet connection all computers get an IP address through DHCP shell script configured on the start image. Whenever a PC turns on, it sends a DHCP request to acquired an IP address based on the MAC address assigned by the first script. Now, we have ability to connect this IP address by Virtual Network Computing (VNC). Once the setup and configuration are completed, they can be used to program the flow-table in an OpenFlow network emulation Testbed with a fully-virtualized environment.

Our simulations aim at studying the interaction between the openflow protocol and the transport layer using our designed implementation to scale to large network configurations. Our goal is that the SDN Controller manages the switches, in such a way that the nodes become forwarding devices only. At this point, each OVS created acts as a regular layer-2 learning switch, it automatically creates a learning table based on the source MAC address of incoming frame and places it on an incoming port then either forwards the frame to the appropriate output port destination.

The routing decisions for all the switches including the forwarding operations at flow or packet level are therefore processed by the controller pushing rules into hardware devices calling the ovs-ofctl add-flow command as well as ARP resolution between hosts and edge switches. As mentioned earlier, if the switch receives a packet that does not match any entry in the flow table, the Openflow switch forwards the packet to the controller by an ARP request. When such request is received by the controller, it will be ignored because the controller does not know the NextHop IP address needed to handle all traffic forwarded to an another subnet. To configure the OVS's to learn the NextHop, an IP addresses is assigned to them. The packets routed will be sent out via the OVS local port to establish the connection between all nodes.

As shown in (Fig. 2b) we have built a testbed to experiment with our presented protocol, and it represents a cluster. Future work will address extending the SDN cluster concept to multiple SDN domains. We have virtualized OpenFlow compatible switchs (OpenVSwitch version 2.3.0) and small Linux TinyCore. From this testbed, we will be able to experiment our protocol with more than 500 things, in a first attempt. Each thing is a TinyCore Linux system.

Acknowledgements

This work was supported by SENACYT-Panama, Secretaria Nacional de Ciencia, Tecnologia e Innovacion.

5. Conclusion

In this paper, we have proposed a cluster management system based on OpenFlow. The system manages communication between clusters by an SDN cluster head managed in an SDN controller. We have also built a prototype system to emulate a real Openflow network. Our next set of experiments is in progress, wherein we are working to set up a routing cluster protocol dynamic over SDN. This work shows promise for achieving one of the goals of software defined networking, which is to provide better routing paths for objects in cluster scenarios. Future work includes expanding our simulations to use IPv6 and routing protocols for MD-SAL Clustering environments. In addition, more simulations are required to evaluates the performance of the proposed architecture implemented for IoT networks.

References

1. Evans, Dave. The Internet of Things How the Next Evolution of the Internet Is Changing Everything. CISCO white paper (2011).
2. Liotou, E.; Tseliou, G.; Samdanis, K.; Tsolkas, D.; Adelantado, F.; Verikoukis, C., *Multi-tenancy for Virtualized Network Functions*, in Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on. pp.1-2, May 2015.
3. Moreno Sanchez P, Marin Lopez R, Gomez Skarmeta AF. A Network Access Control Implementation Based on PANA for IoT Devices. Sensors 2013. p.14888-14917
4. A. Tootoonchianand Y. Ganjali, *Hyperflow: A distributed control plane for openflow*, in Proceedings of the Internet Network Management Conference on Research on Enterprise Networking. pp.3-3, 2010.
5. S. Scott-Hayward, G. OCallaghan, and S. Sezer, *SSDN security: A survey*, in Proceedings of the IEEE SDN for Future Networks and Services. pp.1-7, 2013.
6. T. Luo, H. Tan, T.Q.S. Quek, *Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks*, Communications Letters, IEEE, 2012, 16, (11), pp.1896-1899
7. P. Azad and V. Sharma, *Cluster Head Selection in Wireless Sensor Networks under Fuzzy Environment*, ISRN Sensor Networks, 2013
8. M. Ba, O. Flauzac, R. Makhloufi et al.: *A Novel Aggregation Approach Based on Cooperative Agents and Self-Stabilizing Clustering for WSNs*, The Seventh International Conference on Communication Theory, Reliability, and Quality of Service, CTRQ 2014
9. D. Zeng, T. Miyazaki, Song Guo et al., *Evolution of Software-Defined Sensor Networks*, Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference, 2013, pp.410-413
10. O. Flauzac, C. Gonzalez, and F. Nolot *SDN Based Architecture for Clustered WSN*. Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 9th International Conference on, Blumenau, 2015, pp. 342-347.
11. A. De Gante, M. Aslan and A. Matrawy, *Smart wireless sensor network management based on software-defined networking*, Communications (QBSC), 2014 27th Biennial Symposium 2014, pp.71-75
12. Y. Fan , V. Gondi, J.O. Hallstrom et al.: *OpenFlow-based load balancing for wireless mesh infrastructure*, Consumer Communications and Networking Conference (CCNC), pp.444-449, 2014.
13. El-Mougy, Amr; Ibnkahla, Mohamed; Hegazy, Lobna, *Software-defined wireless network architectures for the Internet-of-Things*, in Local Computer Networks Conference Workshops (LCN Workshops), pp.804-811, Oct. 2015.
14. N. Mitton, A. Busson, E. Fleury, *Self-organization in large scale ad hoc networks*, in MED-HOC-NET, 2004
15. O. Flauzac, B.S. Haggar, F. Nolot, *Self-stabilizing clustering algorithm for ad hoc networks*, ICWMC, 2009, pp.24-29
16. O. Flauzac, C. Gonzalez, A. Hachani and F. Nolot *SDN Based Architecture for IoT and Improvement of the Security*. Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on, Gwangiu, 2015, pp. 688-693.
17. C. Johnen and L. H. Nguyen, *Robust self-stabilizing weight-based clustering algorithm*, TCS, 2009, pp.581594
18. N. Mitton, E. Fleury, I. Guerin Lassous et al., *Selfstabilization in self-organized multihop wireless networks*, in ICDCSW, 2005, pp.909915
19. E. Caron, A. K. Datta, B. Depardon et al., *A selfstabilizing k-clustering algorithm for weighted graphs*, JPDC, 2010, pp.11591173
20. M. Ba, O. Flauzac, B. S. Haggar et al, *Self-stabilizing k-hops clustering algorithm for wireless ad hoc networks*, in: 7th ACM ICUIMC (IMCOM), 2013.
21. Network Functions Virtualization (NFV), Available: *OpenDaylight*, http://www.opendaylight.org/
22. P. Diogo and L.P. Reis and N. Vasco Lopes, *Internet of Things: A system's architecture proposal*, in 9th Iberian Conference on Information Systems and Technologies (CISTI), 2014 , pp.1-6, 18-21 June 2014
23. Shuai Gao; Yujing Zeng; Hongbin Luo; Hongke Zhang, *Scalable area-based hierarchical control plane for software defined information centric networking*, in 23rd International Conference on Computer Communication and Networks (ICCCN), 2014, pp.1,7, 4-7 Aug. 2014
24. Xu Li; Djukic, P.; Hang Zhang, *Zoning for hierarchical network optimization in software defined networks*, in IEEE Network Operations and Management Symposium (NOMS), 2014, pp.1-8, May 2014
25. OpenFlow Switch Specification, Open Networking Foundation, Available: http://www.opennetworking.org/