Enabling a Carrier Grade SDN by Using a Top-Down Approach

Flávio Silva[1], **Caio Ferreira**[2], **Natal Neto**[1,2], **Alex Mota**[2], **Luiz Theodoro**[1,2], **João Henrique de Souza Pereira**[2], **Augusto Neto**[3], **Pedro Frosi Rosa**[1]

[1]Faculdade de Computação (FACOM) – Universidade Federal de Uberlândia (UFU)
Caixa Postal 593 – 38.400-902 – Uberlândia, MG – Brasil

[2]Algar Telecom - Uberlândia, MG – Brasil

[3]Dept. de Informática e Matemática Aplicada (DIMAp) - Universidade Federal do Rio Grande do Norte(UFRN) - Campus Lagoa Nova, Natal-RN - Brasil

{flavio,frosi}@facom.ufu.br, natal@mestrado.ufu.br

{caiocf,alex,lclaudio,joaohs}@algartelecom.com.br

augusto@dimap.ufrn.br

Abstract. *Software-Defined Networking (SDN) essentially decouples the hardware from the software that controls it. Currently, OpenFlow materializes some SDN abstractions and several OpenFlow controllers, based on different programming paradigms and architectures, are available. In this scenario, SDN community is using a bottom-up approach in order to build a SDN control layer that meets carrier grade requirements such as throughput, availability and scalability. Conversely, this work proposes a top-down approach for SDN controller architecture that is integrated with a carrier grade service level execution environment, based on the JAIN SLEE specification. The proposed approach reuses a mature component model, already deployed, thus extending SDN based services by integrating OpenFlow with several network resources and communication protocols providing a cross layer platform that can satisfy these telecom operators requirements.*

1. Introduction

Software-Defined Networking (SDN) [Open Networking Foundation 2012] essentially decouples the data plane from the control plane and provides a deep control of infrastructure by abstracting it from the network applications, thus, enabling innovation. Currently, OpenFlow [McKeown et al. 2008] materializes some concepts of SDN.

However SDN come up with some research challenges [Sezer et al. 2013] to the scientific community. A key challenge is related with the controllers that should meet carrier grade requirements [UNION 2006] that encompasses high availability, scalability, high performance, reliability, fault tolerance and manageability in order to foster the SDN adoption in mission critical environments, such as the ones handled by telecom operators.

Currently, several telecom operators have services [TeleStax 2013] deployed on top of a mature platform, known as JAIN SLEE. The JAIN (Java APIs for Integrated Networks) is a set of APIs dedicated to creating voice and data convergent services. The JAIN

SLEE (Service Logic Execution Environment) [Ferry and ODoherty 2008] is a component model that supports the deployment of event driven applications that requires carrier grade requirements [Gomez et al. 2009] [Femminella et al. 2011]. JAIN SLEE is available as commercial products, such as Rhino [OpenCloud 2013], and also as open source platform, such as Mobicents [MOBICENTS 2012].

At this moment, the SDN community is still in pursuit of a carrier grade SDN control layer that is suitable for the demands of such environments. The current efforts related to this area uses a bottom-up approach, which consists of creating the control layer components from scratch. Some examples of these ongoing efforts are the projects OpenDaylight [OpenDaylight 2013] and ONOS (Open Network Operation System) [ON.LAB 2013].

This work presents a contribution to this research area by using a top-down approach, which differs from the efforts currently undertaken by the SDN community in relation to this quest. This manner consists of constructing a control layer that reuses the component model defined by JAIN SLEE.

This paper briefly presents, at Section 2, the JAIN SLEE component model and the components created in order to integrate OpenFlow with the protocols used by telecom's voice and data services. Section 3 presents the status of this work, concluding remarks and future work.

2. Carrier Grade SDN Control Layer

The work, presented here, is deployed on top of the Java APIs for Integrated Networks (JAIN). The goal of these APIs is to abstract the underlying network, so those services can be developed independently of network technology. This approach couples with SDN abstractions.

The main components of JAIN SLEE are presented at Figure 1. The component model consists of two different layers: the Application Layer that represents the services which run at the JAIN SLEE Application Server and the Resource Adaptation Layer which abstracts underlaying protocol stacks and adapts them to the JAIN SLEE model.

At the Application Layer, the Service Build Blocks (SBB) contains the application and service logic. Each SBB can be composed of one or more children SBBs and they are organized as a graph. A Service is a deployed and managed artifact which specifies its root SBB and the default event delivery priority. A SBB registers to capture and fire events. An Activity is a related stream of events, such as a phone call, that are captured by SBBs entities. The state of these entities can be replicated in a clustered deployment. By using clustering, JAIN SLEE supports high availability and fault tolerance. In our approach, the services of the SDN Control Layer can be deployed as SBBs.

According to the JAIN SLEE specification, a *Resource* represents a system that is external to the JAIN SLEE. The Resource Adaptation layer (depicted at Figure 1) enables several control plane protocols, currently used at the telecommunication protocol stack to plug in at the JAIN SLEE component model. The approach presented here uses an OpenFlow compliant adapter that integrates OpenFlow with JAIN SLEE and fosters the development of new services and applications that can exploit SDN benefits.

The figure also presents two resource adapters created through this work, one in order to adapt the OpenFlow 1.0, the basis to our claims, and the other to integrate the IEEE 802.21 Media Independent Handover (MIH) [IEEE 2009] highlighting the flexibility of the presented approach to accommodate other protocols and build new scenarios.

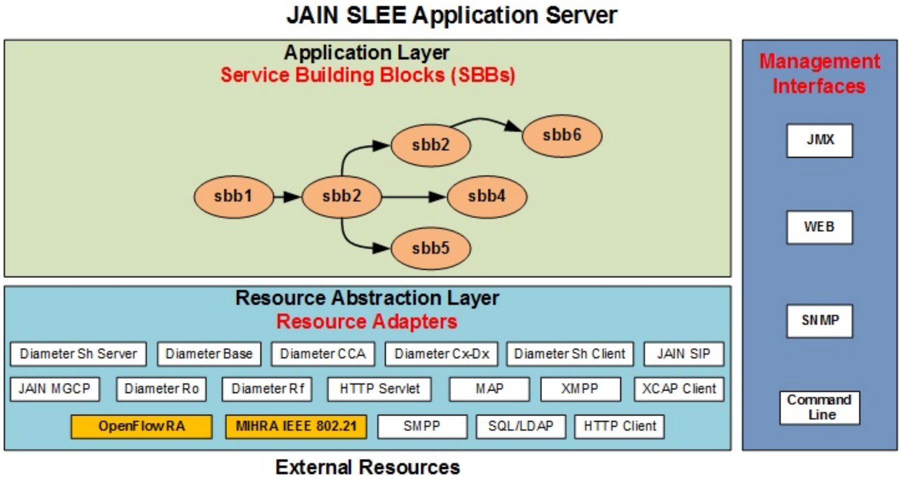

Figure 1. JAIN SLEE Architecture Main Components.

Considering all these features, JAIN SLEE is a platform suitable to handle carrier-grade throughput, latency and fault tolerance requirements over a general purpose IT infrastructure [Gomez et al. 2009].

3. Concluding Remarks and Future Work

This work proposes a new SDN controller architecture, which is integrated with a carrier grade service level execution environment, based on the JAIN SLEE specification. Such architecture can abstract several different protocol stacks and provides a common component model where new services and applications could be deployed. JAIN SLEE current implementations are adopted and being used by several telecom operators worldwide.

The approach proposed here was applied and in order to couple JAIN SLEE with SDN and an OpenFlow 1.0 resource adapter was created. Moreover, using the flexibility of the presented architecture, another resource adapter was created enabling the use of both protocols in order to optimize handover between different networks [Guimarães et al. 2014].

The carrier grade approach presented in this work is aligned with most current trends regarding a SDN control layer, but on the contrary of other proposals, it enables an integration of the protocols that controls the network hardware, such as OpenFlow, with the ones the controls the applications, thus enabling new types of network services. Moreover, the extensible model can accommodate new protocols and future initiatives, thus preserving investments and becoming an interesting outcome that can be exploited by telecom operators.

As future work, the proposed, and constructed, carrier grade SDN control layer will be tested under different scenarios in order to demonstrate the fault tolerance and scalability. In addition, an OpenFlow 1.3 compliant resource adapter, based on the ONF OpenFlow driver (a.k.a. *libfluid*), will be deployed and plugged into the architecture.

The innovative approach presented under this work shows to the research community a SDN control layer that is suitable to meet carrier grade requirements and is a viable alternative to bring SDN into the telecom infrastructure, capable of fostering SDN adoption by this industry.

References

[Femminella et al. 2011] Femminella, M., Francescangeli, R., Maccherani, E., and Monacelli, L. (2011). Implementation and performance analysis of advanced IT services based on open source JAIN SLEE. In *2011 IEEE 36th Conference on Local Computer Networks (LCN)*, pages 746–753.

[Ferry and ODoherty 2008] Ferry, D. and ODoherty, P. (2008). JAIN SLEE (JSLEE) 1.1 specification, final release.

[Gomez et al. 2009] Gomez, M., Torres, E., Chamorro, J., Hernandez, T., and Mendez, E. (2009). On the integration and convergence of IN and IP mobile service infrastructures. In *International Conference on Telecommunications, 2009. ICT '09*, pages 143–148.

[Guimarães et al. 2014] Guimarães, C., Corujo, D., de Oliveira Silva, F., Rosa, P. F., Augusto Jose Venancio Neto, P. D., and Aguiar, R. L. (2014). IEEE 802.21-enabled entity title architecture for handover optimization. In *IEEE WCNC'14 Track 3 (Mobile and Wireless Networks) (IEEE WCNC'14 Track 3 : NET)*, Istanbul, Turkey. Unpublished.

[IEEE 2009] IEEE (2009). IEEE standard for local and metropolitan area networks- part 21: Media independent handover. *IEEE Std 802.21-2008*, pages c1 –301.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74. ACM ID: 1355746.

[MOBICENTS 2012] MOBICENTS (2012). Mobicents JAIN SLEE. http://www.mobicents.org/slee/intro.html.

[ON.LAB 2013] ON.LAB (2013). ONOS - open network operating system.

[Open Networking Foundation 2012] Open Networking Foundation (2012). Software-defined networking: The new norm for networks.

[OpenCloud 2013] OpenCloud (2013). Rhino SLEE carrier grade system.

[OpenDaylight 2013] OpenDaylight (2013). OpenDaylight technical overview.

[Sezer et al. 2013] Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., and Rao, N. (2013). Are we ready for SDN? implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7):36–43.

[TeleStax 2013] TeleStax (2013). TeleStax – open source cloud communications - success stories.

[UNION 2006] UNION, I. T. (2006). *The carrier grade open environment reference model*. SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS. International Telecommunication Union.