Introducing Social Awareness to Next Generation Wireless Networks

Pavlos Kosmides, Konstantinos Demestichas, Angelos Rouskas, Evgenia Adamopoulou and Miltiades Anagnostou

***Abstract*—**Wireless communications have evolved rapidly during the last decades resulting to an environment of heterogeneous Radio Access Technologies (RATs). The coexistence of these RATs in a seamless manner is one of the most important issues arising in the development of next generation wireless networks. One of the most promising approaches includes the use of Software Defined Networks (SDN) in order to assist the abstraction of multi-RATs into software-based controllers. However, this path results to the increase of the amount of data that will be handled by the controllers. In this paper, in order to enhance software-based controllers in the decision making process, we introduce the use of Social Networks in order to collect necessary information about users that belong to each controller. We also present the proposed systems architecture including Business Processes and Services.

***Keywords*—**Heterogeneous Wireless Networks, Radio Access Technologies, Software Defined Networks, software-based controllers, Social Networks.

I. Introduction

DURING the last decades, wireless networks have been growing at high speed, passing through many stages of evolution like 2G, 3G and recently 4G networks. This evolution has resulted to the existence of many Radio Access Technologies (RATs) with different network characteristics. Similarly, there has been an explosive progression on users' mobile phones (e.g. smart phones), while by 2020 the traffic originating from mobile applications is expected to increase exponentially [1]. In addition, research community has recently started the effort for the development of the next generation (5G) of wireless networks [2]. A major challenge that arises is the concurrent operation of different RATs in a heterogeneous wireless environment.

An approach that has been recently proposed by various researchers is the use of Software Defined Networks (SDNs). One of the main objectives of SDNs, is to centralize the network intelligence in *software-based controllers* [3]. For example, the authors in [4] propose the use of SoftRANs, a software defined centralized control plane for radio access networks and present a preliminary design and architecture along with use cases and a feasibility analysis. Similarly in [5] the authors introduce OpenRAN, an architecture for software-defined RAN via virtualization.

The authors in [6] propose a centralized Software-defined network controller, or Software-defined RAN (SD-RAN), which abstracts all the resources made available by a pool of base stations into a single, large resource pool, namely the Virtual Cell (V-Cell). Finally, in [7] the authors argue that SDNs can simplify the design and management of cellular data networks, while enabling new services. They propose extensions to existing controller platforms, switches and base stations to enable software defined cellular networks.

However, there has not been any effort on introducing Social Network awareness to software-defined networks. In this paper, we combine SDNs with online Social Networks. Specifically, we propose an innovative system architecture that takes advantage of the software-based controllers and imports social awareness from online Social Networks, in order to assist the centralized controller in the decision making process, using Machine Learning techniques.

The following sections present in detail the designed architecture for the implementation of the abovementioned service, including foreseen components and the specified interactions among them. The architecture is designed using *ArchiMate*® [8] showing the application layer entities (application functions) and their relationship.

II. Social Network Connectivity

The *Social Network Connectivity service* is responsible for establishing connection of the software-based controller with available Social Networks. It uses the available APIs that are provided from each Social Network and after applying the required policies that are addressed from the Social Networks services, it links the software-based controller with the Social Network.

Fig. 1 states the main business processes identified for the *Social Network Connectivity service* and show how they map to application layer entities (namely application functions); such relationship is indicated by the "realization" arrow, according to *ArchiMate*® notation [8].

P. Kosmides, K. Demestichas, E. Adamopoulou, M. Anagnostou are with the Institute of Communication and Computer Systems of the National Technical University of Athens, Iroon Polytechneiou 9, 15772 Greece (phone: +30-210-7721495; e-mail: pkosmidis@ cn.ntua.gr).

A. Rouskas is with the Department of Digital Systems, University of Piraeus, Karaoli kai Dimitriou Str. 80, 18534 Piraeus, Greece.

ISBN: 978-1-61804-254-5

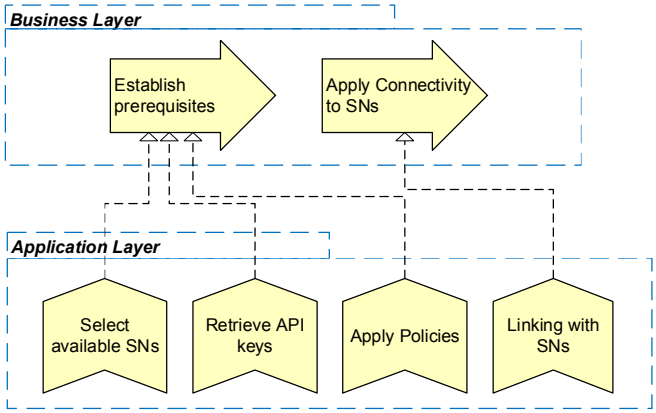

Fig. 1 Social Network Connectivity – mapping business layer to application layer.

As shown, the following functions have been identified for the application layer:

- *Select available SNs*: this application function is responsible for selecting which of the available Social Networks are going to be used in order to crawl necessary data.
- *Retrieve API keys*: this application function retrieves the necessary API keys from the Social Networks developer guides.
- *Apply Policies*: this application function is responsible for applying the policies that are defined by each Social Network.
- *Linking with SNs*: this function establishes the connection with the appropriate Social Network.

Fig. 2 illustrates application functions and components together with relevant interfaces, main data objects and information flows for the *Social Network Connectivity service*.

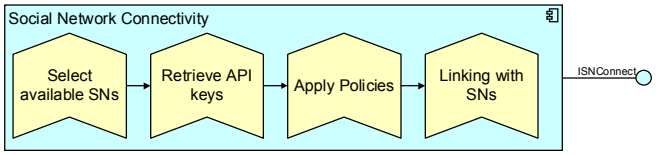

Fig. 2 Social Network Connectivity – main application components, functions, interfaces and data objects.

III. Data Determination

The *Data Determination service* is responsible for collecting information from both Base Stations and connected Social Networks. From the Base Stations, details like the location and each BS's range are defined, while the crawling of data from Social Networks is limited to the users that are located in each BS's range.

Fig. 3 states the main business processes identified for the *Data Determination service* and show how they map to application layer entities (namely application functions).

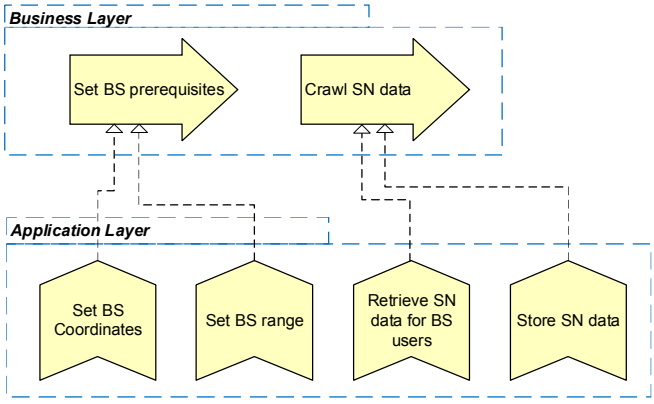

Fig. 3 Data Determination – mapping business layer to application layer.

As shown, the following functions have been identified for the application layer:

- *Set BS Coordinates*: this application function declares the coordinates of the corresponding Base Station.
- *Set BS range*: this application function is responsible for estimating the range of the corresponding Base Station.
- *Retrieve SN data for BS users*: this function limits the crawled data from the Social Networks for users that are located inside the BSs' range.
- *Store SN data*: the retrieved data that refer to the specific BS and its users are stored in the SNDB (Social Network Database) through this application function.

Fig. 4 illustrates application functions and components together with relevant interfaces, main data objects and information flows for the *Data Determination service*.

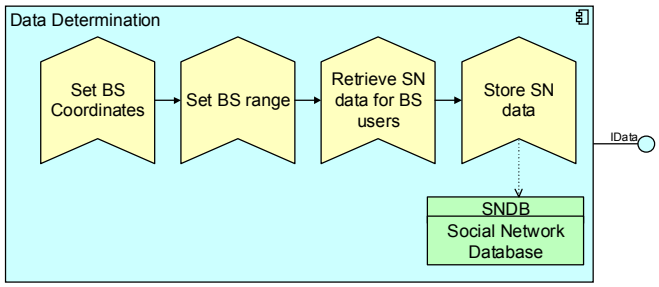

Fig. 4 Data Determination – main application components, functions, interfaces and data objects.

IV. Machine-Learning Engine Training

The *Machine-Learning Engine Training service* is responsible for the centralized training of machine-learning engines which will be used by Intelligent Decision Making service (described in Section V). The centralized training is made on the software-based controller side.

Fig. 5 summarized the main business processes identified for the *Machine-Learning Engine Training service* and shows how they map to application layer entities.

ISBN: 978-1-61804-254-5

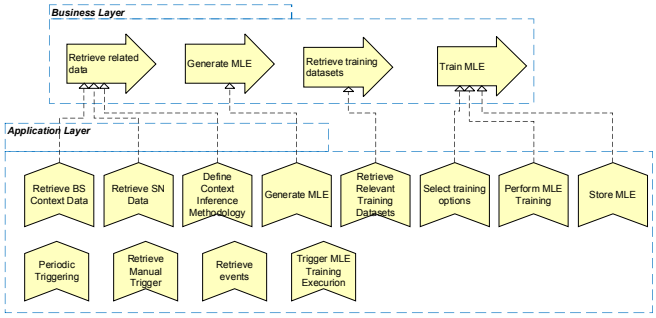

Fig. 5 Machine-Learning Engine Training – mapping business layer to application layer.

As observed from the diagram presented, the application functions can be divided into two categories. The following application functions have been identified for the first category:

- *Retrieve BS Context Data*: this application function is responsible for retrieving context data related to the specified Base Station.
- *Retrieve SN Data*: the data that were stored in the SNDB (Social Network Database) and are related to the specified Base Station, are retrieved through this application function.
- *Define Context inference methodology*: this function is responsible for applying inference mechanisms in order to infer knowledge from the retrieved data.
- *Generate MLE*: this application function is responsible for the creation of Machine-Learning Engines that will be used for the training process.
- *Retrieve Relevant Training Datasets*: this application function retrieves the relevant dataset that will be used for the training process from the MLTDB (Machine Learning Training Database).
- *Select training options*: this application function selects and defines the training options for training the datasets.
- *Perform MLE Training*: MLE training is performed on this application function with regard to the retrieved training datasets and the options that were defined in the previous application functions.
- *Store MLE*: the resulting MLEs are stored in the MLEDB (Machine Learning Engine Database) through this application function.

The above mentioned application functions are implemented by the application component named *MLE Training Execution*.

The application functions of the second category are:

- *Periodic Triggering*: this application function periodically sends messages to trigger the Machine-Learning Engine Training.
- *Retrieve Manual Trigger*: this application function allows manual triggering of the Machine-Learning Engine Training.
- *Retrieve events*: this application function triggers the Machine-Learning Engine Training based on events.
- *Trigger MLE Training Execution*: this application function is responsible for triggering the *MLE Training Execution* component, after it receives the appropriate message from the above functions.

These application functions are implemented by the application component named *MLE Training Scheduler*. The main purpose of this component is to initiate the generation of new MLEs and its application functions are not mapped to any business layer processes. Fig. 6 illustrates application functions and components together with relevant interfaces, main data objects and information flow.

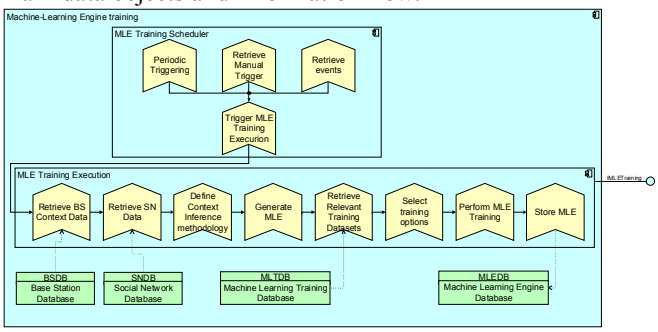

Fig. 6 Machine-Learning Engine Training – main application components, functions, interfaces and data objects.

V. Intelligent Decision Making

The *Intelligent Decision Making service* is responsible for taking into account the current status of the software-based controller and using the trained MLEs reaches to decisions such as resource allocation, energy efficiency etc.

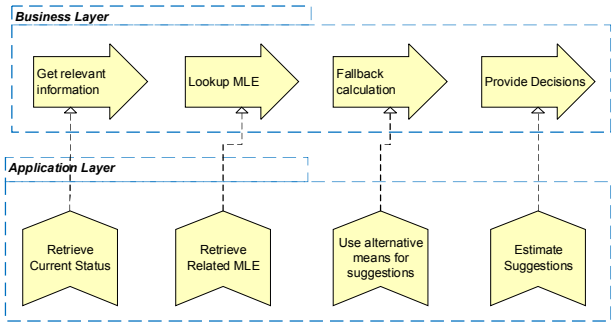

Fig. 7 Intelligent Decision Making – mapping business layer to application layer.

Fig. 7 states the main business processes identified for the *Intelligent Decision Making service* along with their mapping to the application layer entities.

The applications that can be identified from the presented diagram are the following:

- *Retrieve Current Status*: this function retrieves details about the current status of each BS which is part of the software-based controller.
- *Retrieve Related MLE*: the MLE that has been trained from the *Machine-Learning Engine*

training service is retrieved from the MLEDB (Machine Learning Engine Database).

- *Use alternative means for suggestions*: in case there is any problem retrieving data required for successfully estimating the suggestions list, such as no relevant MLE having been created yet, this function uses a fallback mechanism to provide estimations with alternative means.
- *Estimate Suggestions*: This function finally provides the estimated list with suggestions that will assist the decision making process.

Fig. 8 illustrates application functions and components together with relevant interfaces, main data objects and information flows for the *Intelligent Decision Making* service.

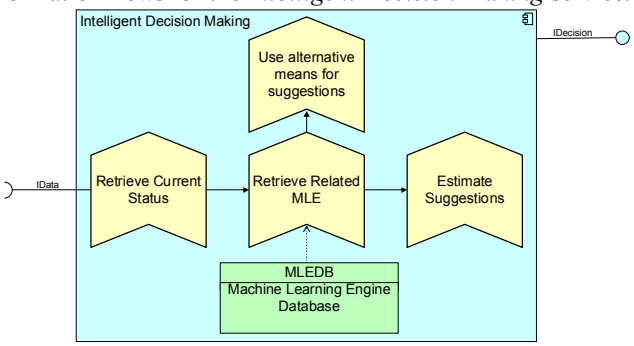

Fig. 8 Intelligent Decision Making – main application components, functions, interfaces and data objects.

VI. Conclusions

In this paper, we have discussed a novel approach for extracting knowledge from online Social Networks, in order to assist software-based controllers that rely on SDNs. The application and business layers of the identified architecture, including foreseen components and their interactions, were presented in detail. Further research activities include the testing and validation of the discussed approach, proving the concept's wide degree of feasibility.

Acknowledgment

This work has been performed under the Greek National project WikiZen (11ΣΥΝ_10_1808), which has received research funding from the Operational Programme "Competitiveness & Entrepreneurship" of the National Strategic Reference Framework NSRF 2007-2013. This paper reflects only the authors' views, and the Operational Programme is not liable for any use that may be made of the information contained therein.

References

[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5Gs," *IEEE Communications Magazine*, vol. 52(2), pp. 74-80, Feb. 2014.

[2] W. H. Chin, Z. Fan, and R. Haines, "Emerging technologies and research challenges for 5G wireless networks," *IEEE Wireless Communications*, vol. 21, no. 2, pp. 106–112, Apr. 2014.

[3] B.A.A. Nunes, M.A.S. Santos, B.T. de Oliveira, C.B. Margi, K Obraczka and T. Turletti, "Software-Defined Networking Enabled Capacity Sharing Software-Defined Networking Enabled Capacity Sharing in User Centric Networks", *IEEE Communications Magazine*, vol 52(9), pp. 28-36, Sept. 2014.

[4] A. Gudipati, D. Perry, L.E. Li and S. Katti, "SoftRAN : Software Defined Radio Access Network," in *Proc. of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, Hong Kong, 2013, pp. 25–30.

[5] M. Yang, Y. Li, D. Jin, L. Su, S. Ma and L. Zeng, "OpenRAN : A Software-defined RAN Architecture Via Virtualization," in *Proc. of the ACM SIGCOMM*, Hong Kong, 2013, pp. 549–550.

[6] R. Riggio, K. Gomez, L. Goratti, R. Fedrizzi, and T Rasheed, "V-Cell: Going beyond the cell abstraction in 5G mobile networks," in *Proc. of IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–5.

[7] L.E. Li, Z.M. Mao and J. Rexford, "Toward Software-Defined Cellular Networks," *in Proc. of 2012 European Workshop on Software Defined Networking*, 2012, pp. 7–12.

[8] ArchiMate® 2.1 Specification, retrieved on May 2014 from URL: http://pubs.opengroup.org/architecture/archimate2-doc/