Extending SDN Framework for Communication Networks

M. Saravanan(✉), Arud Selvan Sundaramurthy, Divya Sundar, and K. Hiba Sadia

Ericsson India Global Services Pvt. Ltd., Tamarai Tech Park, Chennai, India
{m.saravanan, arud.selvan.sundaramurthy, divya.sundar}@ericsson.com, hbsadia@gmail.com

Abstract. Software-Defined Networking (SDN) is an emerging new norm for networks which deals heavily on dynamic nature of higher bandwidth network applications to address the service velocity. SDN's features such as dynamic configuration of network elements, allowing appropriate open standards and centrally controllable tasks will make it suitable to introduce new applications for communication networks. In particular, by considering the voluminous telecom subscriber's transactions and supporting high performing applications in the event of network element failure, a manual Intervention is required for tuning of network elements. Even the recently introduced network resources for tuning could also fail due to unexpected flow of traffic without considering the exact load features dynamically. Any sudden failure in network functioning could bring huge revenue loss and also reduces considerable Quality of Experience of the service provider. In this paper, we have presented use cases relevant to Online Charging System (OCS) that highlight the integration of SDN with communication networks for managing optimized network utilization. Moreover, SDN with the use of machine learning techniques will take the proactive measure before the network node goes down. To support this feature of SDN, we have proposed Autonomous Resource Monitoring and Deployer application that monitors the continuous traffic flow in OCS, intelligently reroutes the traffic with the use of SDN controller by introducing new resources. Our approach handles specific network key performance indicators dynamically, which reduces maintenance and Operational Expenditure costs.

Keywords: Quality of Experience · SDN framework · Online Charging System (OCS) · Telecom applications

1 Introduction

Telecommunication service providers are witnessing a remarkable increase in volume, variety and velocity of subscriber's data due to technological advancements such as smart phones, social media, etc. This dynamic and rapid growth of today's telecommunication market demands new and innovative ways to compete and successfully introduce new technologies in the increasing competition. In the past decade, service providers with large networks and manpower ruled the telecom market. In the current scenario productive network administration and quick dispatch of new services is

B. Mandler et al. (Eds.): IoT 360° 2015, Part II, LNICST 170, pp. 539–550, 2016.
DOI: 10.1007/978-3-319-47075-7_58

expected rather than having sophisticated infrastructure. Thus the operators who know how to deal with their foundation and infrastructure all the more viably are considered to be more successful in business.

The recent big data projections which involve huge number of networking devices and humungous networking data traffic also demands smarter network management and control. The networking systems currently used consist of routers, switches, hubs etc., with limited software logic required to route the packets. Each and every router runs a distributed routing algorithm to enable packet forwarding. The present setup doesn't have the intelligence to reroute packets to an efficient path and maintain the heavy traffic due to large telecom transactions. One such application in telecom industry is the pre-paid charging system, which involves large volume of data transfer continuously taking place between Network Elements (NE). Due to high volume, the system is exposed to issues like network outage frequently which in turn degrades the overall (or specific location) network performance. These challenges have paved way for developing new norms for networking framework which is efficient and cost effective called Software Defined Networking (SDN). SDN allows network administrators to automatically and dynamically manage and control a large number of networking devices, service KPIs, topology, traffic paths and packet handling (i.e., Quality of Service) policies using high level language and APIs in a multitenant environment [1]. Thus by incorporating intelligence into this SDN controller with the use of machine learning techniques, the communication network can be made completely autonomous for future preemptive tasks.

SDN provides a dynamic enhancement to the current framework for the telecommunication industry. To meet the needs of exponentially growing network data scenario intelligently, integration of various IT technologies and virtualization techniques are needed. Telecom industries have to manage communications dynamically without any stoppage than any other industry. This data communication involves the location of customers, means of communication of the customers, handling Network Elements (NE) and methods of their business transactions. The data traffic involves bulk transfer of the network data, aggregation and partitioning of telecom subscribers data and control messages which are latency sensitive. SDN technology can be exploited to develop highly efficient applications. The features of SDN serve as a motivation to explore the design, architecture and framework details to build various applications in order to address typical network issues.

In the telecommunication industry, SDN has been trying to provide a flexible way of delivering quality in terms of bandwidth utilizations, efficient cloud based network optimizations and effectively handling subscriber's data [2]. In addition, SDN requires satisfying the core telecom demands in terms of providing proactive and optimized way of managing and handling network traffic thereby avoiding unnecessary traffic bursts [3]. In this paper, we have analyzed the capabilities of the enhancements of SDN in addressing packet loss due to network congestions and compared it with the capabilities of traditional networking systems. We have also provided an insight into integrating the Online Charging System (OCS) with SDN framework to identify the network variations in the flows and prevent failures occurring due to network outburst and congestions when similar flows are encountered. The failures in the charging system network elements or sudden deviations in the bandwidth utilizations of network will result in unnecessary spikes in CPU causing traffic congestion which results in loss of transactions in the network and reduces the operator's knowledge in Quality of Experience (QoE). This can be addressed by bringing up a new instance of network element and routing traffic to it in order to autonomously balance the load. In our experiment the threshold (i.e. No. of packets received on a particular port) is pre-fixed (i.e. already derived using machine learning), which are monitored by SDN controller. When the threshold is reached, a new host is added dynamically, which minimizes the packet loss when compared with early methods where network outage is detected by SDN and then new node is deployed. Implementation of the proposed idea on a real-time telecom environment straightaway is quite challenging considering the practical difficulties. Therefore, we have used the network emulator called MININET [4] to perform relevant experiments and to address similar network traffic issues. Hence, our paper focuses on the discussion in detail about the work flow of the proposed extension in SDN framework with machine learning, relevant use cases, and the experimental results. Finally we conclude with the merits and demerits of the proposed framework.

2 Related Studies

Telecommunication industry has moved from 3G to 4G mobile networks and it is progressing towards 5G to control and effectively handle the large volume transactions. The volume of data generated is increasing, thus industries which were earlier handling multiple terabytes of data are now handling data in petabytes. Moreover the advent of IPv6 would create enormous number of IP addresses which in turn would lead to the exponential growth of internet of connected devices [2]. This increased volume of data in the telecom sector demands new operational challenges. There are various kinds and format of data flooding into the telecom industry due to the advent of social media, cellular devices, location sensors, etc. It involves heterogeneous data which includes structured data in the form of traditional databases and unstructured data comprising of audio, video, text documents, stock ticker data, email and financial transactions. The most challenging task is to manage, merge and govern these data items, including the signaling data in OCS. In addition to this, the number of subscribers using mobile phones, especially smart phones is exponentially increasing, which in turn increases the stress in network elements due to high signaling data load in OCS. Signaling data in OCS systems are latency sensitive, which in turn are affected by high volume of data load. OCS signaling data delay degrades network performance and must be handled in a timely manner. Communication service providers have a tough time in handling data and service velocities in the present load. All these aspects together prevent the service providers to provide best service to their subscribers.

In comparison to the evolution of data systems to handle telecom service provider's data, network management and control systems should also be enhanced as the current systems are not efficient enough to transfer data which are real time and latency sensitive. It has also become extremely difficult to configure huge OCS networks, which might span in multiple locations in service providers' network. SDN is an evolving technology which can successfully address these issues. Wang et al. [5] provided an insight of incorporating SDN with telecom applications and have analyzed

SDN's architecture in a telecom operator's view. Ferrer et al. [6] have proposed an approach to address the complex Virtual Network Functions scheduling problem in the SDN domain. From another explorative study [7], we noted down the positional changes of SDN and NFV under a complementary and unified framework especially in future carrier networking domain. We also understood the capabilities of combining SDN controller with optical switching to explore tight integration of application and network control [8].

Open Networking Foundation (ONF) looks SDN as a new networking framework which originates or invents OPENFLOW protocol to simplify and generate new standard in networking [9]. It has been mainly developed in order to change routing policies on the fly, to have an easier programming framework and to serve as an operating system for networks. Figure 1 specifies the existing components of SDN framework.

The software defined networking which consists of OPENFLOW switches is controlled by the high powered centralized server called the SDN controller. SDN Applications can be developed by exploiting this framework and they interact with the controller using northbound APIs while the controller controls the underlying network using southbound APIs. SDN has the capability of creating a programmable network by considering both next generation systems and existing infrastructure making them more dynamic and flexible [10]. It is efficiently achieved by integrating essentially different systems and technologies together under a common monitoring and management. SDN is directly programmable, agile, centrally managed and open standards-based vendor

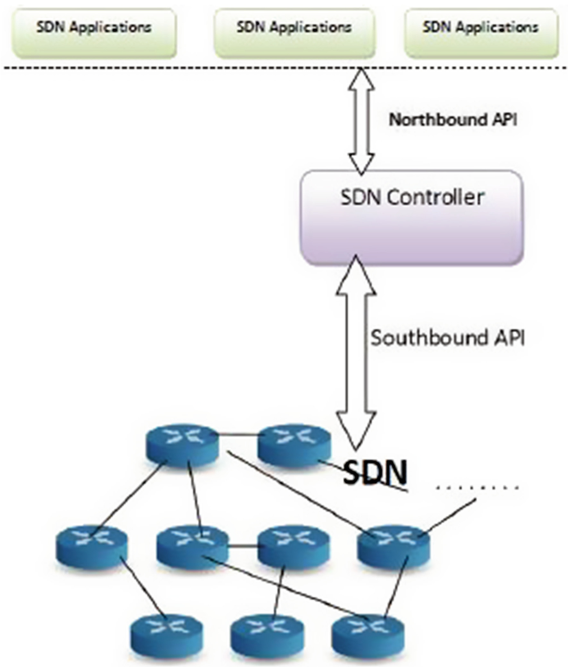

Fig. 1. Existing SDN framework

neutral architecture [11]. Certain benefits of SDN include virtualization, orchestration, dynamic scaling, automation, visibility, performance, multi-tenancy, openness and service integration [9]. In this paper, to utilize benefits of SDN usage, we developed a new application called ARMD to address unexpected traffic outbursts/congestions in OCS attached to SDN controller.

3 Telecom Domain Use Cases and Relevant Solutions

In this paper, we are providing an insight on the integration of SDN with OCS telecom network architecture to introduce two new practices specifically.

3.1 New Practices

First practice is to introduce network element monitoring to intelligently reroute the traffic and the second practice deals with the specific type of patterns and related intelligence incorporated in SDN to bring up a virtual instance of the network node dynamically during peak hours. To streamline the first practice, the traffic flow level and the network usage over a period of time needs to be monitored continuously. The changes with respect to the traffic disturbances should be alerted and handled proactively without causing service discontinuity and providing seamless service to the subscribers.

3.2 Execution Procedures

The traffic flowing through the charging system networking nodes could be continuously monitored by SDN controller. We would like to include all OCS network functions, which are monitored by SDN controller for network usages like bandwidth, load on NE, etc. SDN controller acts as an orchestration of OCS Network Function Virtualization (NFV). NFV uses software implementation of network functions called Virtual Network Functions (VNF) to deploy virtual network nodes into the underlying network [12]. SDN keeps monitoring load spikes in the OCS network functions. Usually, online charging traffic will increase during holidays or weekends or festival seasons due to many customers initiating enormous number of voice calls, text messages and MMS messages. This will create an operationally high volume of stress in OCS network nodes, which causes telecom service disruptions. To avoid this, an application can be developed to monitor the traffic load pattern in OCS network elements on a regular schedule. Based on the load pattern witnessed the application built over the controller instructs OCS network through the SDN controller, to deploy additional OCS network elements dynamically at the operator's site, thereby reducing the service outage during peak hours. Since the network elements are dynamically deployed on need basics, it will reduce online charging service unavailability and increase operator's revenue and also reduces OPEX cost. To introduce new practices, we have developed an Autonomous Resource Monitoring and Deployer (ARMD) application above (but connected to) the SDN controller which monitors the network and deploys new hosts into the network based on the requirements.

4 Workflow of SDN Controlled ARMD Application

This section describes the workflow of SDN controlled ARMD Application integrated with SDN. In Fig. 2 various stages of the application features are specified.

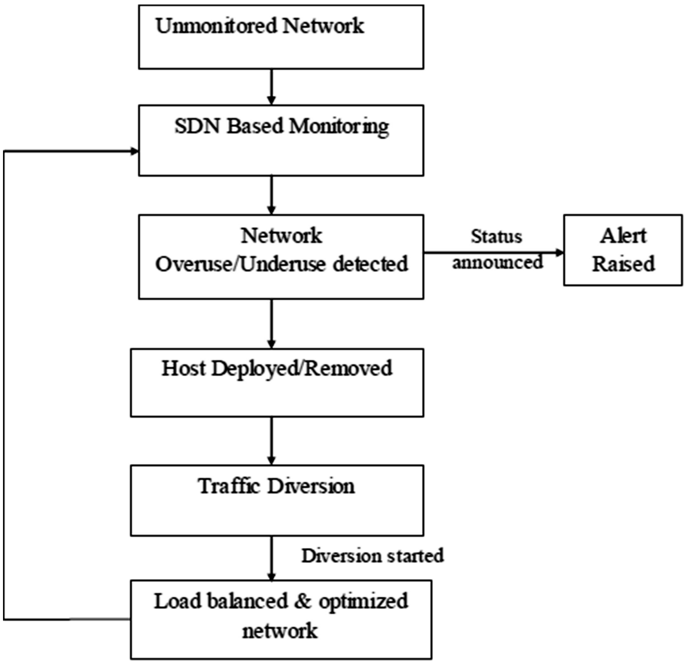

Fig. 2. ARMD application workflow

4.1 Unmonitored Network

The unmonitored network here refers to the network which is not controlled or monitored by the SDN controller. The network unless being changed manually, remains static. It is prone to packet loss and congestions. SDN based monitoring along with the incorporation of ARMD application will enable these capabilities and make the network intelligent enough to handle congestions.

4.2 SDN Controller Based Monitoring

SDN controller visualizes the global network view of the underlying network and understands the current scenario. The abstract view of the underlying network is presented to the ARMD application. The application is capable of capturing the packets received count on the ports of every switch present in the network.

4.3 Network Overuse/Underuse Detected

Every host present in the network is capable of handling only certain load after which the host may fail or will not be in a position to handle client requests. This scenario may result in packet loss which in turn might affect the operator's QoE. On the contrary, if the network is deployed with nodes greater than required, it makes the network underused which may result in unnecessary resource wastage. To avoid such discrepancies in the network, the ARMD application adds or removes hosts to or from the network based on certain threshold value (This value is learned based on the experiments). If the packets received count of a particular port exceeds the upper-bound, network overuse is detected and similarly if the packets received count of a port is found to be below the lower-bound, network underuse is detected. The alert is raised to explicitly intimate the internal counter-action taken by the application.

4.4 Host Deployed/Removed

After congestion or network overuse detection on a particular port, a virtual node/host is deployed dynamically through the controller to balance the load. In the same way if network underuse is detected, the corresponding host which has no traffic for a considerable amount of time will be removed from the network. Thus using ARMD application, hosts are being dynamically added to or removed from the network based on the network usage.

4.5 Traffic Diversion

Once the hosts are added to the network or removed from the network, traffic should be diverted. Traffic diversion is accomplished by reassigning the IP addresses of the hosts being removed or added. By enabling appropriate traffic diversion, there could hardly be any packet loss in the network.

4.6 Load Balanced and Optimized Network

The network always remains load balanced and optimized due to the use of ARMD application. Hosts are deployed dynamically when needed and removed if not required. In addition to load balancing, optimal use of resources is also ensured. Thus by using ARMD application with the SDN controller we can explore optimal, cost-effective and dynamic networks based on the operator requirement.

5 Developing Application in SDN Framework

To develop new application in SDN framework, we require OPENFLOW switches, Telecom nodes (hosts), an SDN controller, Southbound APIs and Northbound APIs. The SDN controller has the capability to monitor only the SDN aware OPENFLOW switches. Thus, building an application for the real-time SDN framework is challenging

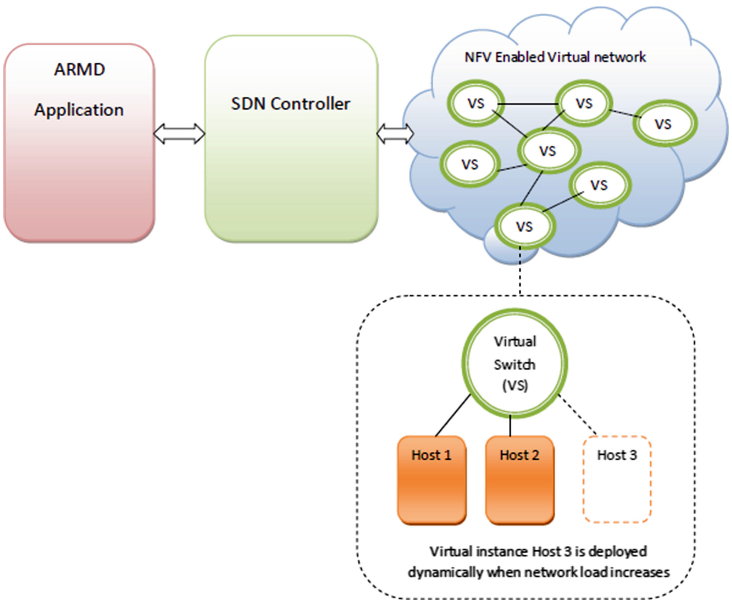

Fig. 3. ARMD for virtual network controlled SDN framework

and has some practical difficulties. Therefore, we have used a network emulator called MININET [4] for our experiments. Figure 3 depicts the virtual network created by MININET and controlled by SDN controller through the decisions made by the ARMD application.

5.1 ARMD Application

The ARMD *Application* is an intelligent module written to establish explicit, direct and programmatic communication with the SDN Controller via Northbound APIs (REST APIs). It communicates with the controller to provide awareness about the network requirements and to achieve desired network behavior. In addition, it requires an abstracted view of the network for its internal decision making purpose. It consists of telecom use case relevant logic along with one or more NBI Drivers.

5.2 SDN Controller

The SDN Controller is a logically centralized entity in charge of (i) translating the requirements from the SDN Application layer down to the SDN Data paths (logical network device) and (ii) providing the SDN Applications with an abstract view of the network which may include statistics and events. Control decisions are taken by the controller which includes dynamic deployment or removal of network nodes and traffic diversion through other nodes. An SDN Controller consists of more than one NBI

Agent, the SDN Control Logic, and the Control to Data-Plane Interface (CDPI) driver. The controller establishes connection with the underlying network via Southbound APIs such as OPENFLOW.

5.3 Virtual Network

The logical network devices referred to as Virtual Switches (VS) are being introduced and controlled by the SDN controller that can be emulated by MININET [4]. It has the capability to create realistic virtual network with simple commands. Users can also create custom topologies by running user-defined python scripts. MININET CLI enables the users to interact with the network. It creates virtual hosts with separate IP addresses. However it is possible to create desired number of hosts. It creates OPENFLOW software switch VS with ports. Different switch topologies can be created and these virtual switches created by MININET are capable of being monitored by the controller. Moreover, it connects the OPENFLOW switch with each virtual host via a virtual Ethernet cable and each host is assigned with a MAC address corresponding to its IP address. It also configures the OPENFLOW switch created, to connect to a remote controller. The inputs from the ARMD application makes the controller take dynamic decisions which are enforced on the virtual network. Figure 3 depicts OCS running on two hosts, Host 1 and Host 2. Whenever the threshold count is reached, the ARMD application observes and intimates it to the SDN controller which in turn dynamically deploys a new virtual instance Host 3 proactively. These decisions taken on the fly make the underlying network evolve gradually to handle network congestions autonomously.

6 Implementation of ARMD in SDN Framework

Due to practical difficulties in the implementation of our procedures in real-time network traffic, we have simulated the network congestion manually, by explicitly setting a valid threshold as per the experience gained. Once the threshold count is reached, the application gives instruction to the network emulator (MININET) to make changes to the current network via the SDN controller to minimize packet loss. The network changes are imposed by detecting the port that is overloaded (No. of packets received on a particular port increases the given threshold), addition of a new host to the network and reassigning the IP of the overloaded/failed host to the newly added host. We are explicitly making the host go down to illustrate the congestion in the network. In real-time setup the host will automatically fail due to high network traffic.

For this entire process to take place programmatically (with the help of SDN controller and the ARMD application developed above the controller), maximum of 3 packets are lost. However in certain cases there is lesser packet loss. But when this process is done manually, the packet loss is phenomenally high. On an average 40 to 45 packets are lost for these changes to take place even if congestion is detected immediately and this loss count may vary based on the factors like time taken to detect the congestion in the network, time taken to deploy a new host and re-assign IP and time

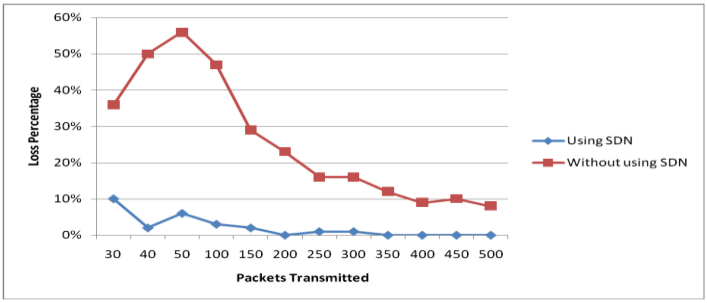

Fig. 4. Comparison of loss percentage in network with and without using SDN

taken to divert traffic flow to the newly added host. Figure 4 depicts the packet loss in the network in comparison with traditional system. It clearly shows the distinction of packet loss during the stages of packet transmission after extending the SDN framework with machine learning for this application.

After successful implementation of the proposed idea with MININET emulator, we would like to extend this setup to real-time networks by addressing the practical difficulties. Each and every virtual component created should be replaced by exactly same physical component. SDN controller is capable of monitoring only OPENFLOW switches, thus virtual switches should be replaced by real ones and they should be configured to address the procedures. After performing the physical setup, when MININET is initiated, SDN controller has a view of physical network of switches functioning in real-time.

One main limitation of using MININET is its inefficiency or lack of performance particularly at high loads. It is uncertain that a host that is prepared to send packets would be scheduled correctly and the rate at which the switches transmit would be the same. Currently, MININET has the capability to run only on a single machine. All the switches, hosts and Ethernet links are emulated on a single operating system. This limitation restricts all the network elements to share the same resources which turns out to be a great disadvantage to span the system on a large scale. Although not appropriate for substantially large scale simulations, MININET is a great network emulator for prototyping and testing network traffic.

7 Possible Extensions in Telecom Sector Using Machine Learning

SDN framework can be extended for various applications in the telecom domain. In this section, we have suggested an insight on how new applications could be built by extending the SDN framework with the use of Machine Learning techniques. We expect this possible extension can introduce more autonomous, security and intelligence in communications networks.

7.1 Suggesting the Use of Machine Learning Techniques

A learning agent could be built above the SDN controller which enables dynamic reconfiguration of network based on traffic estimates. The agent gets information about the underlying network, learns the network behavior and stores the details about the network congestions. Thus the agent contains the previous history of congestions. Machine Learning techniques could be employed to understand the dynamic nature of the networks and train the system accordingly. As a result, the learning agent gradually learns about network, stores all the learnt information and predicts discrepancies in the network before they occur based on the past learning experiences. The agent interacts with the SDN controller to dynamically configure the monitored network and also help to fix few needed thresholds.

Figure 5 describes about various modules of the learning agent which interact with each other to exert necessary changes to the network being controlled by the SDN controller. The knowledge base present in the learning agent has extracted knowledge about the current and previous network configuration states. Watchdog signal from the network elements represents various conditions of the network such as latency; link availability info etc., with this information the knowledge base provides feedback to the learning element. Considering the feedback and past learnt history, the learning element derives new interesting patterns or matches current pattern with the already learnt pattern. After pattern matching is performed it sets learning goals for the problem generator. Appropriate counter actions for the network congestion problems could be employed. The performance of the new decisions is evaluated by the performance element and the evaluated results are provided as a necessary base for new directions that will be considered in future. For example, in this application, a prediction-based technique is used to understand the exact threshold to introduce new host.

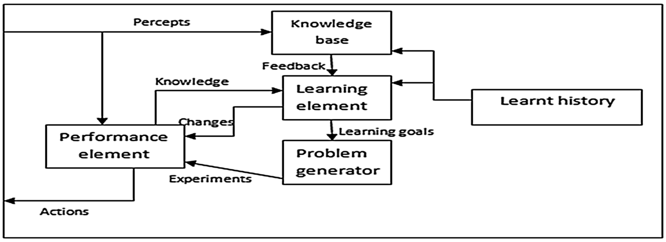

Fig. 5. Learning agent

8 Conclusion and Future Work

Our idea is an explorative work of integrating SDN framework in communication network by introducing new ARMD application. This will make the applications network aware rather than having traditional networking systems with application awareness. The telecom prepaid business models guarantee revenue assurance to operators, service assurance to subscribers and also quality of service for the products. Since the prepaid charging market is dynamic in nature, operators face a huge pressure in maintaining the QoS. It involves large OPEX cost to the operators. Hence the proposed enhancement in SDN framework with machine learning techniques introduces an automatic system to monitor the network elements and dynamically deploy additional nodes when the traffic is increased as well as remove the additional nodes in the network when the traffic decreases. The SDN controller in conjunction with the ARMD application helps in achieving this milestone by providing dynamic routing of traffic within network elements and reduces the packet loss significantly. Running the system in a completely parallelized architecture involving multiple controllers and streaming of data by the controller, data and service parallelization is also equally challenging. We are more interested to run the proposed framework in a real-time system by incorporating machine learning techniques to make the system completely autonomous and to achieve greater efficiency in handling high volume communication networks.

References

1. Humernbrum, T., Glinka, F., Gorlatch, S.: Using software- defined networking for real-time Internet applications. In: Proceedings of the International Multi-conference of Engineers and Computer Scientists, vol. I, IMECS 2014, Hong Kong, 12–14 March 2014
2. The real-time cloud. Ericsson White paper Uen 284 23-3219 Rev B, February 2014
3. http://www.radisys.com/solutions/telecom-solutions/telecom-cloud-sdn-nfv/
4. http://www.mininet.org
5. Wang, J.-Q., Fu, H., Cao, C.: Software defined networking for telecom operators: architecture and applications. In: 2013 8th International ICST Conference on Proceedings of the Communications and Networking in China (CHINACOM), pp. 828–833, 14–16 August 2013
6. Ferrer Riera, J., Escalona, E., Batalle, J., Grasa, E., Garcia-Espin, J.A.: Virtual network function scheduling: concept and challenges. In: 2014 International Conference on Smart Communications in Network Technologies (SaCoNeT), pp. 1–5, 18–20 June 2014
7. Haleplidis, E., Denazis, S., Koufopavlou, O., Lopez, D., Joachimpillai, D., Martin, J., Salim, J.H., Pentikousis, K.: ForCES applicability to SDN-enhanced NFV. In: 2014 Third European Workshop on Software Defined Networks (EWSDN), pp. 43–48, 1–3 September 2014
8. Software-Defined Networking: The New Norm for Networks. White paper, Open Networking Foundation, 13 April 2012
9. SDN Architecture Overview. Openetworking.org. Accessed 22 Nov 2014
10. Wang, G., Eugene Ngy, T.S., Shaikh, A.: Programming your network at run-time for big data applications. IBM T.J. Watson Research Center, Rice University, ACM, August 2012. 978-1-4503-1477-0
11. SDN and the Future of Service Provider Networks. White paper, Fujitsu Network Communications Inc. (2013)
12. Network Functions Virtualisation – Introductory White Paper. ETSI, 22 October 2012. Accessed 20 June 2013