Service Function Chaining Using SDN

Sushritha S
Assistant professor, Dept. of CS&E
Channabasaveshwara Institute of Technology,
Gubbi, Tumkur

Vedashree N
Assistant Professor, Dept. of IS&E
Channabasaveshwara Institute of Technology,
Gubbi, Tumkur

Abstract:-Network service chaining allows composing services out of multiple service functions. Traditional network service functions include, e.g., firewalls, TCP optimizers, web proxies, or higher layer applications. Network service chaining requires flexible service function deployment models. Related work facilitating service chaining include, e.g., the network service header proposal discussed at the IETF or the complementary work on network function virtualization(NFV)atETSI.

Keywords:-software define network, service function chaining.

I. INTRODUCTION

Network is a group of two or more computers linked together. Examples for computer networks are local area network, wide area network and metropolitan area network. Over past decades networking principles remained unchanged. Switches and routers are used to assemble the networks. Devices are developed by number of vendors commonly using proprietary operating system and interfaces. An institution has to apply a specialist on every router brand for building a heterogeneous network on devices from distant vendors. Because of this probability of configuration mistakes also increases while configuring different systems. So a new technology has to be addressed to make networks more scalable, to allow easily managing of networks devices from distinct vendors. To fulfil these needs, a new programmable network called Software Defined Network (SDN) is used.

SDN is a new technology which decouples the data plane and the control plane of network. This decoupling leads to a new architecture. Switches are used for basic packet forwarding devices consists of flow tables populated with the localized flow rules . Rule indicates how incoming packets are handled based on the matching fields. These are managed by the remote "controller" in the SDN. Control plane indicates how the packets should be moved. Communication between controller and the switches will be in a secure manner using a standard and open interface like OpenFlow protocol. It consists of internal flow table and a standardized interface to add or remove flow entries. This architecture allows for a range of considerably more flexible and effective network management solutions. A logically centralized controller provides application developers with a unified programmable interface on which to deploy software and higher level of application. Its approach for providing flexible network programmability. It provides real time configuration, operation and monitoring of a network.

SDN encompass of three layers and their interactions are shown in figure 1.1.If there large-scale and wide-area region network, then we may use more than one SDN controller. Through network policies control layer always balances the network states in either a distributed or centralized manner. Because of dynamic flow of activities network policies should be updated timely, for the unrestricted access to global network elements and resources. The SDN applications are present in the application layer of the SDN. The communication between the application layer and the control layer are supported by a set of application programming interfaces such as north bound open APIs, for the enabling of common network services like routing and access control, traffic engineering, bandwidth management, QoS, DDoS mitigation, usage of energy etc. Forwarding packets in the data layer can be employed by programmable OpenFlow switches via OpenFlow Controller and communication of switches with the controller through south bound APIs (ex. OpenFlow protocol).

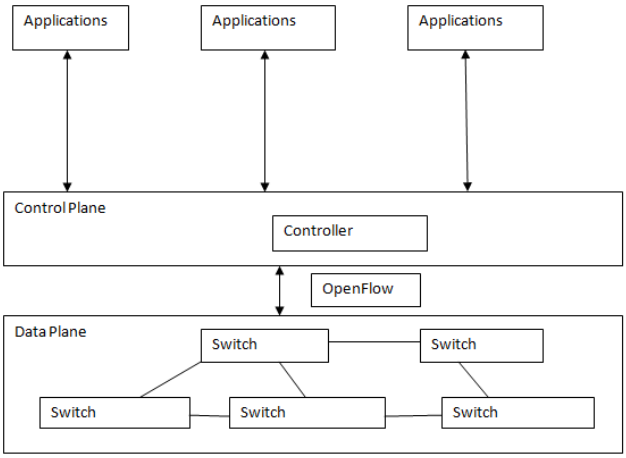

Fig. 1. SDN Architecture

The interaction among these layers, the SDN allows a whole and global view of the network, and also includes a powerful control platform for the network management over traffic flows. Simplifying of the network management, reducing of operating costs, promote innovation and evolution in present and future network are all will be done in the SDN.

II. BACKGROUND

NFV and dynamic network service chaining are still new and evolving topics. The terminology is not yet standardized completely, even more so for their implementations. An overview on the terminology used in this work is depicted in Figure 2. It is partially based on [1] and has been extended where necessary. The basic concept is a service function, which defines in an abstract manner the treatment of packets related to network flows.

Examples include firewalls, HTTP proxies or bandwidth limiters. A service instance is an operational soft- or hardware instance that is deployed in the network, delivering the treatment specified by the associated service function. Software service instances are deployed in services nodes. Besides providing a runtime environment, service nodes comprise facilities for attaching service instances to the network. A service chain specifies a network service that is implemented through the applications of a series of service functions. Like a service function, it is an abstract specification that is implemented by service chain instances. Furthermore, service chain instances rely on service function instances and their interconnection for delivering a network service to a given users.

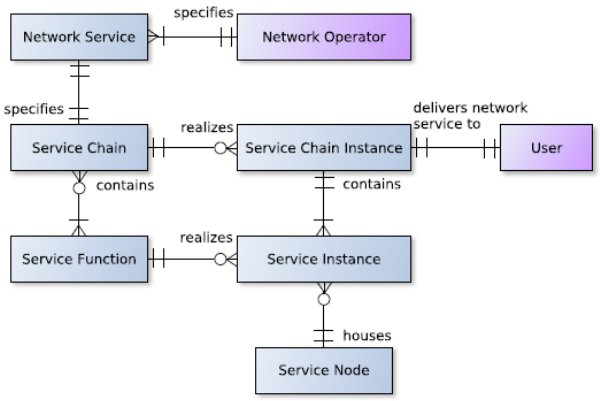

Fig. 2. High-Level Entity Relationship Model for Service Chaining

A service chain instance embodies an instantiated service chain capable of delivering a concrete network service to a concrete user. While a service chain defines an order of abstract service functions, an active service chain instance requires the setup, parameterization, and availability of service instances needed to realize the service chain's expected behaviour to the user. Additionally, it pronounces the need of building up the vital activity courses (streams) between the client, benefit examples, and the suitable supplier edge switch. The movement treatment a client's activity encounters is called organize benefit. Its usefulness can be either noticeable to or controllable by the client, or implemented and alloted to specific clients or client bunches by the system administrator. The system administrator gives and oversees a systems administration framework and determines the accessible system administrations to convey to its clients. At long last, a client is an extraordinary organized end-client gadget asking for access to various kinds of system administrations.

III. SERVICE CHAINING CONCEPT

The interconnection of administration occurrences, named benefit fastening, works by mapping bundles to benefit chains at the edges and sending them between benefit occations. OpenFlow and uses layer 2 addresses to identify service instances as discussed later. The basic approach for integration of per-user service instances in a service chain is depicted in Figure 2. The arrow points delineate the heading of the parcel stream, though sessions are started by the client's gadget. Each administration example is associated with the system by two connections and advances movement between these. This prerequisite can be satisfied by for all intents and purposes each system benefit. For instance, a HTTP intermediary requires just a single interface, however takes a shot at frameworks with two interfaces also.

The entrance connect faces toward The client, while the departure interface faces from the client. These connections are basic reference focuses for the tying approach. By recognizing the related connection and bearing of movement enteringor leaving the OpenFlow organize, it is mapped to a client,its administration chain example, and the course of sending.Bundles entering the OpenFlow organize are altered to meet the sending prerequisites, i.e. bundle headers must contain the required data for sending the parcel to the following administration work. Parcels leaving the OpenFlow organizeare adjusted to meet the prerequisites of the administration occasion, **i.e.** they should contain suitable layer 2 addresses. In addition, the disconnection of administration occasions is accomplished by separating system control convention activity, for example, ARP parcels. Linux namespaces1 have been connected here to influence the utilization of one administration to occasion for each client attainable for ISP-scale 3 applications under Linux.

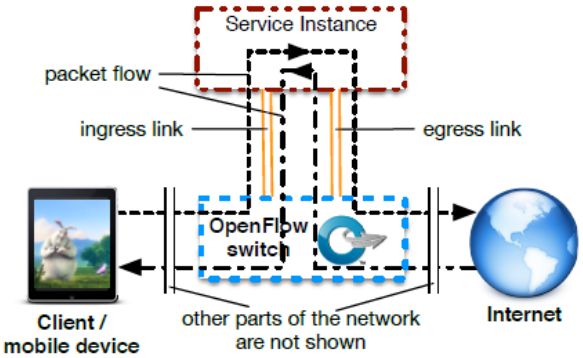

Fig. 3. Service integration into a service chain.

Linux namespaces present an adaptable and effective virtualization approach. Rather than executing an undeniable virtualization condition or a totally virtualized working framework, a few partsof the working system can be virtualized independently with Linux namespaces. This permits to blend and-match he required virtualization highlights for each utilization case. Linux namespaces empower, e.g., client, process, and system namespaces.

The sending idea is propelled by the administration binding approach presented in 5 [4]. As delineated in Figure 3, layer 3 tending to is utilized for recognizable proof of clients and arrangement of parcels at the edges. The switch interfacing the versatile system to the Gi-LAN2 OpenFlow arranges is called entrance edge switch or ingress switch. The switch that interfaces the Gi-LAN to the Internet is called departure edge switch or departure switch. Sending inside the OpenFlow Network is entirely in view of layer 2 addresses. While different fields of the bundle header could be utilized for encoding the sending data, numerous current OpenFlow

switches bolster layer 2 header field 6 changing proficiently in equipment, which is regularly not the situation for layer 3 fields. Reference focuses for the sending are the MAC address and port of both the entrance and the departure switch. For each enlisted client, an OpenFlow run coordinates the IP source address of approaching parcels from the entrance switch.

The MAC address of the following administration occasion that the coordinating parcels should pass is composed to the MAC goal address field. Sending through the OpenFlow organize depends on the goal MAC address. At the point when a bundle touches base at the last OpenFlow switch before the administration occasion, the goal MAC deliver is revised to the one required by the administration occurrences and sent through the related port. After the bundle is prepared by the administration occasion, it leaves the departure port of the administration occurrence and enters the OpenFlow arrange once more. The approaching port of the OpenFlow switch distinguishes the related administration chain example of the parcel.

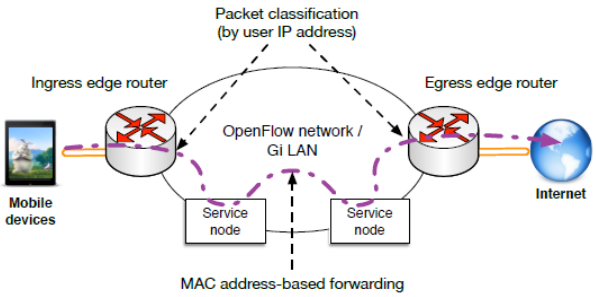

Fig. 4. Overview on the forwarding concept.

Again, the MAC address of the next service instance is written to the destination MAC address field and the packet is forwarded. After the last service instance processed the packet, the MAC address of the egress router is written to the destination MAC address field and the packet is forwarded. Processing packets of flows in the opposite direction is done accordingly**.** Packets incoming from the egress router are associated with a service chain instance by matching the destination IP address of the user. While MAC addresses are used for identifying interfaces of service instances, the addresses are generally not learned by employing the well-known approach of a learning switch. Instead, the MAC address of each interface of each service instance are learned and disseminated bythe chaining control system.

A. Service Instance Interface

Handling packets to and from service instances is done uniquely in contrast to in [5]. Doing without the utilization of MAC address learning in the sending approach enables sifting of control activity to and from benefit cases and in this way empowers their confinement: they work in a totally disconnected system condition. Administration examples are not ready to speak with other administration occasions or hubs in the system, the Address Resolution Protocol (ARP) and comparative control 8 conventions are sifted.

The confinement of the benefit examples is valuable, as an intelligible system tending to plot that traverses the entire administration anchoring framework isn't required. All administration occurrences utilize indistinguishable IP addresses, which disentangles the arrangement procedure. For instance, each administration occasion can utilize 172.16.30.2/30 and next jump 172.16.30.1 for its entrance, and 172.16.30.4/30 and next bounce 172.16.30.5 for its departure interface. Macintosh addresses are found and spread in the framework as required by the control plane.

Appropriate service instance network behavior is either implemented by statically configuring the routes and MAC addresses for the egress and ingress interfaces or by implementing a dynamic routing mechanism and ARP handling module in the adaption layer using OpenFlow. instance of the corresponding service type could be shared by multiple users when using the flow identification scheme of the described system. Additional service function types could be added, the list is not exclusive. A necessities of administration occasions can shift. Five kinds of 9 benefit examples are distinguished as laid out in Table I. The system conduct and data misfortune for every parcel amid the preparing is distinctive for each sort and recorded in the section "User!Internet: changed fields". The prerequisites for approaching activity are recorded in column"In: necessities". For representation, the bundle alterations that are connected on parcels got by the OpenFlow change from the examples departure interface are recorded in column"Egress out: connected adjustments". Relating changes are connected to parcels sent to and from the entrance interface. The administration affixing framework has to know the administration work kind of each administration occurrence.

This sending mode guarantees that passing bundles are not adjusted in the administration example; this write is called L2. Notwithstanding, if an approaching bundle is tended to at the MAC address of the switch, it isn't sent however prepared locally and in this manner dropped. In this manner, the sort requires approaching movement to not be tended to at the MAC address of one of the interfaces of the administration occurrence; they are revised to the one of the edge switch that ought to be passed next while in transit to the goal. A firewall administration can be actualized in light of directing. Approaching bundles are tended to at the layer 2 address of the approaching interface, separated, and sent to the next jump as arranged in the steering table.

The layer 3 information remains unchanged, but both the source and destination MAC address are rewritten during the forwarding process; this type is called L3. For this service type, as well as for types L4src,dst and L7 the requirement is that the MAC address of the incoming packet matches the one of the incoming interface. However, when a packet leaves the service instance, it contains the wrong source and destination MAC address. The destination MAC address is rewritten after every service instance as required by the forwarding system; the source MAC address is rewritten to the one of the opposite edge router.

Redirection changes the destination IP address and layer 4 destination port, the corresponding type is called L4dst. No information that is relevant for the forwarding system is changed; hence no processing in addition to the one of type L3 is required. When using Network Address Translation (NAT), the layer 3 source address and layer 4 source port in the packet header are modified. This service instance type is called type L4src, while the service instance has to be treated like one of type L7. Service instances implementing an application layer function such as a web proxy are based on routing, too. In addition to that, the transport layer connection coming from the user's device is terminated by the proxy and a new connection to the requested web server is created. Therefore, after being processed by the web proxy, the packets layer 2, 3, and 4 headers are changed; this service instance type is called L7. Packets leaving the egress port contain the correct destination IP address, but not the source IP address required by the forwarding system. Hence, the IP address of the user is written to the source address field.

IV. EXISTING SYSTEM

Enablers are network services which are deployed as hardware appliances that are physically connected. These devices play a vital role in network operators achieving the security and performance they desire. Examples include the following devices:

- Firewalls
- Traffic Optimizers
- Network Address Translation (NAT) devices
- Web Proxies
- Load Balancers

These gadgets are utilized to help an assortment of uses. In the static administration chain display, all activity should course through every one of the empowering influences, despite the fact that lone a subset of these administrations might be required. This implies the different apparatuses actualizing these administrations should have enough ability to deal with the full activity pipe, regardless of whether they will simply need to give the stream a chance to go through without handling it. This likewise implies each administration capacity should have its own interior ability for choosing whether a movement stream must be handled or dropped.

There are a few confinements to this approach:

• All administrations must be outlined with a cynical approach, building them for the most extreme conceivable limit.

• As the movement prerequisites increment, all administrations should increment in limit, paying little respect to their real utilization; this could imply that occasionally even ceased or low-use administrations should be scaled-up.

• There is practically no granularity in how the activity is marked and in how benefits are connected to particular streams.

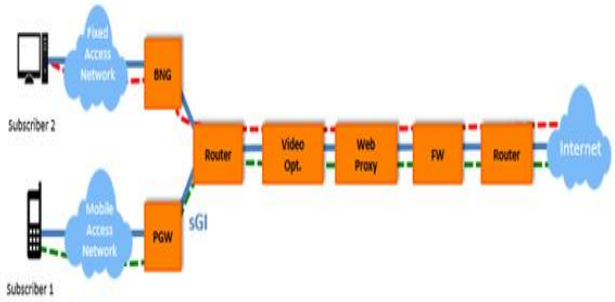

Fig. 5. Static Service Function Chaining

Intheexamplescenarioillustratedinfigure5,Subscriber1 wishes to access video content on their mobile device. The user wouldsimplyneedthevideooptimizationservice,as well as basic firewalling. However, the user's traffic will have to traverse the entire chain. Adding to this, services must often be applied in a specific order, which implies the need for complex routing techniques and VLANs to ensure that this performed correctly. This example highlights the sub-optimal use of network and compute resources, as the entire service chain has to be traversed, regardless of whether this is required or not
For a given service chain, which will be used to support a new application, the operator must first identify if existing appliances/topologies are able to cater for the application's needs. If not, a new topology must be considered; network devices must be purchased, physically connected and manually configured. For instance, in the case where theapplication's user base grows or there are 12 reasonable peaks where the application is being used more often, operators have to constantly overprovision their network to cope with traffic based on estimates, when in reality the volume 13 may never actually reach the expected level.

V. ARCHITECTURE

The system architecture is based on the OpenFlow 14 reference architecture [8] introduced by the Open Networking Foundation. As depicted in Fig. 4, the three major layers are the infrastructure layer, the control layer, and the application layer. While the infrastructure layer consists of the actual OpenFlow devices, the control layer contains the control plane logic. However, the layer should not process application logic data but strictly network- related information. The Service Function Chaining Router (SFCR) module is located in this layer, while the business logic is located in the application layer, where the Service Function Chaining Controller (SFCC) module is located. The high-level, modular architecture for the proposed SDN-based approach is depicted in Figure 6. The data flow is depicted as a double line, where connections originate at the mobile user, cross the wireless network as well as the Gi-LAN and are forwarded to the Internet. Service nodes are traversed if the selected network service requires the forwarding through the service instances that

are hosted on them. Packets arriving from the Internet traverse the path in the reverse direction.

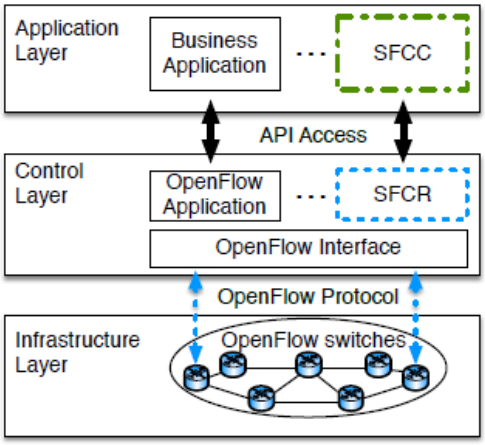

Fig. 6. Mapping of SFCR and SFCC to OpenFlow

Reference Architecture

The Service Function Chaining Controller (SFCC) [9] is the focal controlling unit in charge of controlling all framework segments, keeping up the framework state, taking care of client (de-)enlistment, benefit occasion assignment, and offering an abnormal state API as single interface to the administration binding framework. The SFCC keeps up a lasting association with the SFCR and all accessible Service Node Controllers (SNC). The SNCs are in charge of overseeing one administration hub and the administration occurrences which are worked there. At the point when a client interfaces with the remote system, it is identified and enlisted with the SFCC. A default benefit chain occurrence is then being made for the new client, therefore apportioning the required administration cases and asking for the SFCR to introduce the proper system stream rules. After a distinction occasion, the already settled administration chain case is annihilated by discharging all administration examples and asking for the SFCR to evacuate the made stream rules. The SFCC offers an abnormal state API that enables the system administrator to characterize, change, and evacuate benefit chains amid runtime. In the event that an administration chain is changed, e.g., by including an administration work, for each relating administration chain case a suitable administration occasion is designated and embedded. The SFCC orders the SFCR to refresh the administration chain directing by sending data on the entrance and departure port of the administration occurrence, the OpenFlow switch where it is associated, and the kind of the administration occasion. The SFCR at that point looks into the bundle preparing that is required for this administration example write and introduces the comparing OpenFlow rules. The SFCR manages arrange related deliberations, for example, ports, layer 2, and layer 3 tends to as it were.

Besides, as it is a latent, stateless part, it works on summons by the SFCC as it were. For the OpenFlow gadgets it goes about as a controller and executes fastening of administration examples. Each SNC keeps running on the benefit hub it oversees. It depends on an OpenFlow empowered programming switch and numerous administration occasions. The SNC is in charge of the creation, utilization, design, and decimation of administration occasions. Their 17 usage is in view of standard Linux highlights, for example, movement forming and firewalling or other reasonable programming. Linux arrange namespaces are utilized to isolate the system activity of various occurrences. For each administration example the product switch gives two system ports, building up an association between the two substances. For each administration work that ought to be accessible on a given administration hub, various cases are begun ahead of time. The system utilized for pre beginning cases can be adjusted to the working condition. The framework layer comprises of OpenFlow switches, both devoted equipment gadgets and programming switches 18 running on the administration hubs. Entrance what's more, departure switches are not required to be OpenFlow empowered. Be that as it may, their MAC locations and system ports through which they are associated should be known to the SFCC, either through arrangement or ARP snooping. The administration anchoring system is straightforward to the edge switches. For these, the framework is unclear from an immediate system connect. The versatile system registers and de-enrolls its clients and their IP deliver to the SFCC through its API

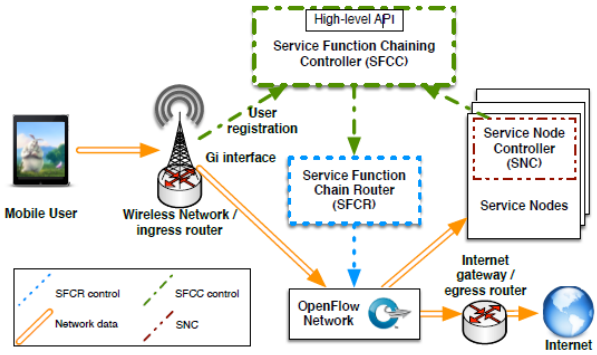

Fig. 7. High-level architecture of components and their interactions.

VI. CONCLUSION

In this paper, mainly concentrated on providing different network services in a chain manner.The services are selected on the basis of the particular operation. The reliability and flexibility can be achieved. This can be used to solve topological dependencies etc.

REFERENCES

[1] R. Bifulco,T. Dietz, F. Huici, M. Ahmed, J. Martins, S. Niccolini, and H.-J.Kolbe, “Rethinking Access Networks with High Performance Virtual Software BRASes,” in EWSDN Workshop, 2013.
[2] M. Chiosi et al., “Network Functions Virtualisation (NFV),” European Telecommunications Standards Institute (ETSI), Oct. 2013.

[3] W. John, K. Pentikousis, G. Agapiou et al., "Research Directions in Network Service Chaining," in IEEE SDN4FNS, 2013.
[4] B. Lantz, B. Heller, and N.McKeown, "A Network in a Laptop," in ACM HotNets, 2010.
[5] N. Leymann, "Flexible Service Chaining. Requirements and Architectures." in EWSDN. Presentation, 2013.
[6] A. Madhavapeddy and D. J. Scott, "Unikernels: Rise of the Virtual Library Operating System," ACM Queue, vol. 11, no. 11, pp. 1–15, 2013.
[7] NEC, "ProgrammableFlow PF5240 Switch," http://www.necam.com/ docs/?id=5ce9b8d9-e3f3-41de-a5c2-6bd7c9b37246.
[8] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," Apr. 2012.
[9] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "SIMPLE-fying Middlebox Policy Enforcement using SDN," in ACM SIGCOMM, 2013