Software Defined Wireless Networking
Technical Report for Course Project

Shashank Rao

***Abstract*— This technical paper presents an overview of software defined wireless network architecture and concepts. It then shows how SDN can be incorporated into existing wireless technologies such as LTE, to improve performance and efficiency. Some of the existing implementations of Software Defined Wireless Network (SDWN) technologies are discussed in detail and analyzed for their contributions as well as flaws.**

I. Introduction

A number of trends are leading network providers and users to re-evaluate traditional approaches to network architecture. These trends can be grouped under demand, supply and traffic patterns. Increasing demand can be attributed to:

- Cloud computing
- Big data
- Mobile cloud
- The Internet of Things (IoT)

Increasing complexity of network traffic patterns have also added to difficulties in configuring network architecture. These complexities have risen from the following developments:

- Client/server applications generate horizontal traffic between servers and vertical traffic between servers and clients.
- Network convergence of voice, data and multimedia traffic.
- Unified communications (UC) strategies involve heavy use of multiple servers.
- Heavy use of mobile devices, resulting in user access to corporate content and applications from any device at any time.
- Widespread use of public clouds.
- Application and database servers virtualization

The current traditional distributed and autonomous approach was developed when networks were predominantly static and end systems at fixed locations. The Open Networking Foundation (ONF) cites the following four general limitations of traditional network architectures:

1) Static, complex architecture: To respond for demands such as high and fluctuating traffic volumes, and security requirements, networking technology has grown more complex and difficult to manage. This results in a number of independent protocols, each of which addresses a portion of networking requirements. An example of this arises when new devices are added or removed; the network management staff must make configuration changes in multiple switches, routers, firewalls, web authentication portals, and so on. It also includes changes in access control lists (ACLs), Quality of Service (QoS) settings in numerous devices, and other protocol-related adjustments.
2) Inconsistent policies: In a large network, when a new virtual machine is activated, it can take hours or even days to reconfigure ACLs across the network.
3) Inability to scale: Adding more switches and transmission capacity, involving multiple vendor equipments, is difficult due to the complex, static nature of the network.
4) Vendor dependence: If enterprises needs to deploy new capabilities or services involving new equipment, they will be severely limited by the relatively slow product cycles of vendor equipment.

All these shortcomings lead to the following requirements to be present in a modern networking approach - network adaptability, automation of policy changes, maintenance with minimal disruption of operations, model level network management rather than reconfiguration of individual network elements, mobility, integrated security, and on-demand scaling. SDN and NFV are designed to meet these evolving network requirements.

ONF defines SDN as - The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices. Some features of SDN are:

- DIRECTLY PROGRAMMABLE - because it is decoupled from forwarding functions.
- AGILE - administrators can dynamically adjust network-wide traffic flow to meet changing needs.
- CENTRALLY MANAGED - Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network.
- PROGRAMMATICALLY CONFIGURED - lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs.
- OPEN STANDARDS-BASED AND VENDOR-NEUTRAL

A. Traditional vs. SDN Networking Approach

The traditional distributed network architecture is shown in Figure 1. The solid line represents packet flows and dotted lines mark the interaction between the controllers of different switches. The control plane functions (routing, security, loop avoidance, failure recovery, etc.) require complex cross-network interactions. In the worst case, a network with n nodes, in which every node must interact with every other node, requires $n^2 - n$ unidirectional interactions, as can be seen in the figure.

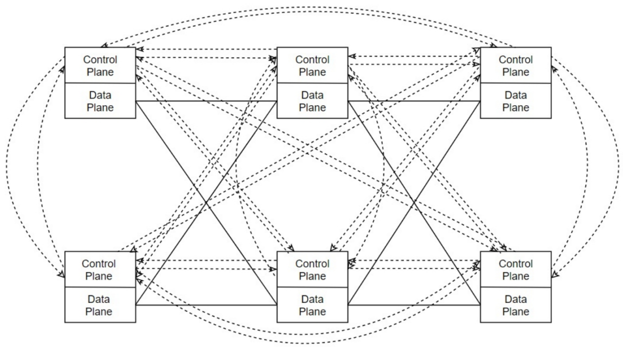

Fig. 1. Traditional Network Architecture - A distributed control plane requires complex node interactions

Figure 2 shows SDN approach taken for the same network as in figure 1. The SDN approach splits the switching function between a data plane and a control plane into separate devices. The data plane is simply responsible for forwarding packets, while the control plane provides the intelligence in designing routes, setting priority and routing policy parameters to meet requirements and to cope with shifting traffic patterns. Open interfaces are defined so that switching hardware presents a uniform interface regardless of the details of internal implementations and to enable networking applications to communicate with the SDN controller

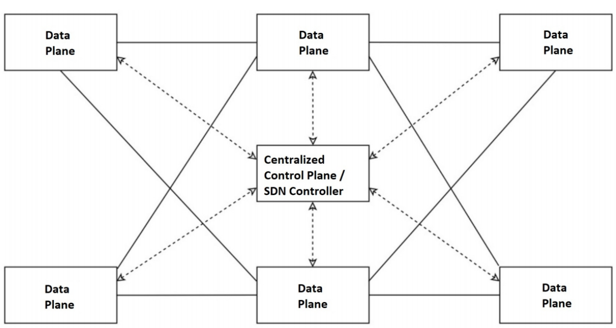

Fig. 2. SDN Approach - A centralized control plane eliminates cross-network interactions

B. SDN Architecture Overview

Figure 3 shows the high level detail of the SDN architecture. The data plane consists of physical and virtual switches. In both cases, switches are responsible for forwarding packets. The vendor can decide on the internal implementation of buffers, priority parameters and other data structures related to forwarding. However, each switch must implement an abstraction of packet forwarding that is uniform and is open to the SDN controllers. This abstraction is defined in terms of an open API between the control plane and data plane (Southbound API). Some of the communication means between control and data planes include - OpenFlow, NETCONF, Extensible Message and Presence Protocol (XMPP), BGP, Open vSwitch Database Management Protocol (OVSDB), and Ciscos onePK. The most prominent example of such an open API is OpenFlow, developed by the Open Networking Foundation (ONF). OpenFlow specification defines both a protocol between control and data planes as well as an API by which control plane can invoke the OpenFlow protocol.

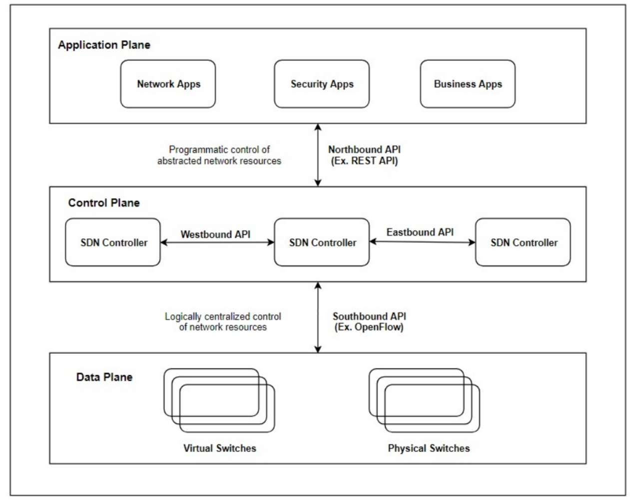

Fig. 3. Software Defined Networking Architecture

SDN controllers can be implemented directly on a server or on a virtual server. An open API is used to control the switches in the data plane. In addition, controllers use information about capacity and demand obtained from networking equipment through which traffic flows, in order to determine the paths taken by the packets in the data plane. SDN controllers also expose Northbound APIs, which allow developers and network managers to deploy a wide range of off-the-shelf and custom-built network applications, many of which were not feasible before the advent of SDN. The most common open Northbound API offered by vendors is Representational State Transfer (REST) APIs which provide a programmable interface to their SDN controller.

The horizontal APIs (Eastbound and Westbound) enables communication and cooperation among groups of controllers to synchronize network state for high availability.

There are a variety of applications at the application plane that interact with the SDN controller. These applications convey their network requirement and desired network behavior to the SDN controller via a Northbound API. Examples are energy-efficient networking, security monitoring, access control, and network management.

Some key features of SDN architecture are:

- The control plane is separated from the data plane. Data plane devices become simple packet forwarding devices.
- The SDN controller has a centralized view of the networks under its control.
- Open interfaces are defined between devices in the control plane and those in the data plane.
- The network is programmable by applications running

on top of the SDN controllers.

C. Overview of Long Term Evolution (LTE)

In mobile networks, the demand for higher data rates, lower delays and seamless mobility is fueling rapid deployment of 4G Long Term Evolution (LTE) and LTE-Advanced technologies around the world. In LTE systems, Radio Access Network (RAN) incorporates OFDMA (Orthogonal Frequency Division Multiple Access) and advanced antenna techniques to maximize the efficient use of spectrum. Moreover, LTE is the first all-Internet Protocol (IP) network technology to enable seamless delivery of real-time packet data applications and services with reduced deployment and operational costs and complexity. The LTE architecture as shown in Fig. 4 is comprised of two components: Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC).

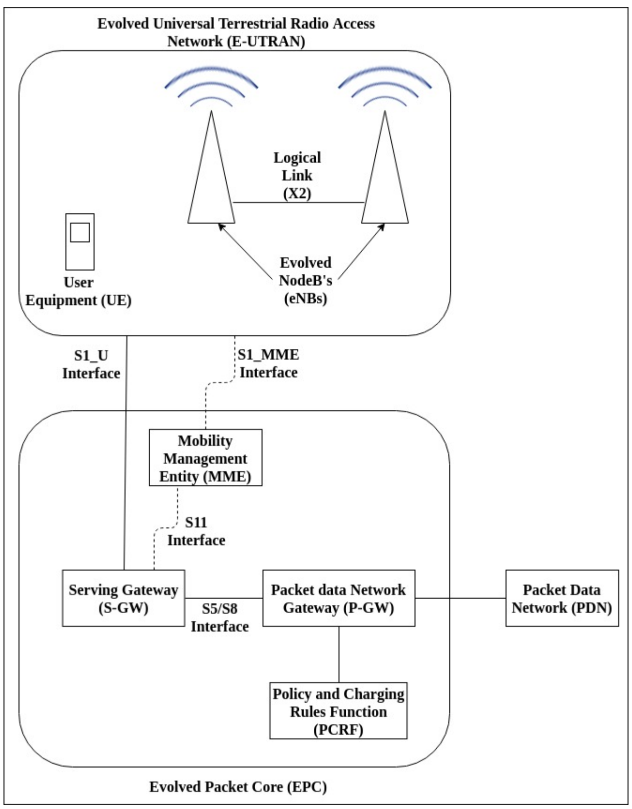

Fig. 4. LTE System Architecture

The access network of LTE - E-UTRAN - consists of a network of Evolved NodeBs (eNBs) which support OFDMA and advanced antenna techniques to connect to User Equipment (UEs). Inter eNB communication to handle interference management and hand-overs are maintained through X2 logical link interface. The E-UTRAN handles the following tasks:

- Radio resource management functions related to radio admission control and radio resource scheduling to UEs, both during uplink and downlink; radio mobility control and radio bearers control.
- Interference management, ciphering, handover management and power control.

The Evolved Packet Core (EPC) is responsible for overall control of UEs and establishment of bearers. The main components of EPC include:

- Mobility Management Entity (MME): It acts as the control plane node in EPC and is responsible for authentication of UEs, establishing bearers, Non Access Stratum (NAS) mobility management, and inter-working with other 3GPP access systems like GSM and UMTS.
- Serving Gateway (S-GW): It acts as the user plane node in EPC and is responsible for maintaining user data tunnels between base stations and P-GW. It also acts as mobility anchor for other 3GPP technologies.
- Packet data network gateway (P-GW): It also acts as the user plane in EPC and is the default router for UE which want to connect to the Internet and intranets. It is used to assign IP addresses to UEs and acts as mobility anchor for non-3GPP access technologies like CDMA2000 and Wi-Fi.
- Policy and Charging Rules Function (PCRF): It interacts with P-GW for enforcing policies and charging rules for the UEs being used.

The eNBs of E-UTRAN are connected to MME via S1_MME interface and to S-GW via S1_U interface. EPC is responsible for the overall control of UEs and establishment of the bearers. Bearers are used to provide Quality of Service (QoS) to UE traffic.

II. SDN Role in Wireless LTE Networks

Handling of mobile networks incurs greater Capital Expenditure (CAPEX) and Operating Expenditure (OPEX) as compared to other wireless networking technologies like Bluetooth and Wi-Fi. As an example, deploying an LTE network involves a variety of eNBs (i.e., Macro, Micro, Pico and Femto), S-GWs, P-GWs, MMEs and Middle BoXes (MBXs) like firewalls, deep packet inspectors, and proxy servers. Deployment and managing these network entities require meticulous network planning and management. Proprietary firmware or closed software solutions of these MBXs and GWs restrict the options available to mobile network operators (MNOs) as they cannot innovate or change these closed box solutions. MNOs also cannot replace their existing network entities easily with newer generation of entities for coping up with the exponential increase in demand and traffic due to deployment issues and higher CAPEX and spectrum licensing costs. MNOs are desperately looking for flexible, manageable and scalable network architectures that could offer innovative services to users with low CAPEX and OPEX. SDN with its key network components (SDN-switches and OpenFlow controllers) offers flexible, manageable and scalable solutions to traditional wired networks. Recent advancements in SDN help us to design and deploy flexible and dynamic SDN based mobile network architectures.

Traditional LTE network divides tasks in the following way:

- Control Plane tasks include access control, bearers management, mobility control, load balance, interference management, radio resource sharing and interworking. The entities responsible for these tasks are eNBs, MMEs, S-GWs and P-GWs.
- Data Plane tasks include routing, policy enforcement, Quality of Service and pricing. The entities responsible for these tasks are S-GWs and P-GWs.

Thus, it can be seen that S-GW and P-GW handle both control and data plane functions. Most of the data plane functions are implemented in P-GW, and as a result, it leads to higher delay and congestion in the network. Also, the control plane functions are distributed across various entities. High centralization of data plane and low centralization of control plane cause inflexibility and scalability issues in the LTE networks. SDN paradigms can help to create highly flexible, adaptive, and scalable mobile networks.

Various interesting SDN based mobile architectures have been proposed in the literature and this paper will discuss and analyze some of these proposals. SDN concepts applied in the LTE-EPC component has resulted in proposals such as SoftCell, ProCel, SoftMoW, SMORE, MobileFlow, etc. SDN concepts applied in the LTE-RAN component has resulted in proposals such as SoftRAN, RadioVisor, WiSA, SoftMobile, OpenRoads, etc.

This paper will incorporate in-depth analysis of the following two proposals - SoftCell for SDN applied in LTE-EPC component and SoftRAN for SDN applied in LTE-RAN component.

III. Critical Analysis of SoftCell

SoftCell is a scalable architecture that supports fine-grained policies for mobile devices in EPC, using commodity switches and servers. High-level service policies of the MNOs are realized as direct traffic through a sequence of Middleboxes (MBXs) based on subscriber attributes and applications. SoftCell uses Multi-Dimensional Aggregation (the dimensions being - the service policy, the base station, and the mobile device at different switches in the network), in order to minimize the size of the forwarding tables. SoftCell performs fine-grained packet classification at the access switches, next to the base stations, where software switches can easily handle the state and bandwidth requirements. Even in the presence of mobility, SoftCell guarantees that packets belonging to the same connection traverse the same sequence of MBXs in both directions.

Cellphone providers rely heavily on subscriber attributes (such as cell-phone model or M2M device type, the OS version, billing plan, parental controls, total traffic usage cap, user roaming, etc.) and application types (such as video and web traffic, or specific applications for which the developers pay the carrier on the user's behalf) in order to create customized policies. In order to route traffic and perform fine-grained packet processing, MNOs rely on proprietary S-GWs and P-GWs communicating using GPRS Tunneling Protocol (GTP). These are highly inefficient, inflexible and complex.

The paper proposes a network design which consists of a fabric of simple core switches, with most functionalities moved to low-bandwidth access switches placed closer to base stations and a distributed set of MBXs that the MNOs can expand as needed to meet demands. These MBXs could be dedicated appliances, VMs, or packet-processing rules running directly in the switches. A logically centralized controller can then be used to route traffic through the appropriate MBXs, via efficient network paths, to realize high-level service policy. However, implementing such a strategy introduces unique scalability challenges in cellular networks - one of them being the fact that commodity switches can only store a few thousands to ten thousands of rules; second being that all traffic in cellular networks goes to and from the Internet, thus placing heavier bandwidth and state requirements on gateways; third, additional state is required to ensure seamless connectivity while the device is mobile. To address these challenges, SoftCell employs two novel techniques: Multi-dimensional Aggregation and smart access edge with dumb gateway edge.

A. SoftCell Architecture

SoftCells goal is to support numerous fine-grained policies in a scalable manner for cellular core networks (or EPCs). SoftCell connects unmodified UEs to the Internet as shown in Fig. 5. SoftCell does not require specialized network elements (e.g., S-GWs and P-GWs) or point-to-point tunneling (e.g., user-level GTP tunnels) generally used in LTE networks.

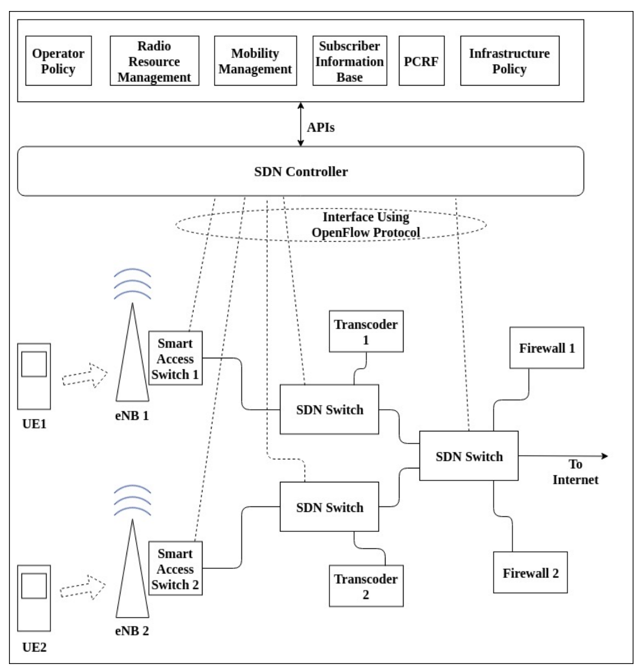

Fig. 5. SoftCell Network Architecture

<u>Controller:</u> Controller implements high-level service policies by directing traffic through the MBXs (transcoder, firewall, etc.) by installing rules in the SDN switches. To compute service policies, controller has access to mostly static

attributes of each UE (eg. billing plan and phone model). To support unmodified UEs, SoftCell controller implements the LTE signaling protocols used between UEs and their MMEs. These protocols handle connection establishment and disconnection, tracking area update, paging, etc.

Smart Access Switches: Access switches are placed at each base stations which perform fine-grained packet classification on traffic from UEs. These can be implemented as software switches implemented on commodity servers. The servers also runs a local agent that caches packet classifiers for attached UEs, so as to reduce interaction with central controller.

Core SDN switches: The rest of the network consists of SDN switches, including a few gateway SDN switches that connects to the Internet. These switches forward traffic through the appropriate MBXs. They can perform arbitrary wildcard matching on IP addresses and TCP/UDP port numbers, or can cache flat rules after processing wildcard rules locally in software. SoftCell gateway switches are much cheaper than P-GWs; they just perform packet forwarding, and relegate sophisticated packet processing to MBXs.

Middleboxes (MBXs): SoftCell supports commodity MBXs such as dedicated appliances, virtual machines, or packet-processing rules on switches. Each MBX function (e.g.,firewall) may be available at multiple locations.

SoftCell does not require any change to the RAN hardware at the base station, or common LTE-RAN functions. SoftCell only changes how the base stations communicate with the core network, by having the base stations coordinate the controller to enforce service policies. Similarly, SoftCell does not require changes to commodity MBXs, or any support from the rest of the Internet.

The low level details like ephemeral network identifiers, the locations of MBXs and switches as well as the application identification are handled by the controller. A service policy contains multiple clauses that specify *which* traffic (called predicate) is handled in *what* way (called action).

Predicate: This is a boolean expression for subscriber attributes and application types.

Action: This is a sequence of MBXs, along with Quality of Service (QoS) and access control specifications. The action does not specify that a particular instance of MBXs should be used, which gives controller freedom to select MBX instances and network paths to reduce latency and load.

Priority: Priorities are used to differentiate between overlapping predicates.

B. Challenges in designing SoftCell

- One challenge faced during SoftCell design was that of supporting fine-grained policies by using small switch tables. Supporting such policies in large networks can lead to an explosion in the dataplane state (or the number of paths that can be taken) needed to direct traffic through the correct MBXs. This challenge was solved in SoftCell design by using multi-dimensional aggregation. This technique combines location based routing and tag based routing to scale to large networks with large service policies.
- The second challenge was to support fine-grained packet classification in asymmetric topology. Packets are generally classified at the network edge in order to determine which service policies to apply to them. However, in a cellular network, there are thousands of base stations at the access edge for UEs, while there are only a few gateway edges facing the Internet to service traffic from all the UEs from thousands of base stations. Hence, classifying packets at the gateway edge at a constant rate is very difficult. A solution to this challenge was by using a smart access edge at the base stations and a dumb gateway edge. Since all traffic is initiated by the UEs, SoftCell classifies packets only at the access edge. The source IP and port number are embedded during the initial classification of the packet at the access point, so that the gateway edge can easily direct the return traffic by examining the destination and port.
- The third challenge was to handle scaling of dynamic network changes. Cellular networks are dynamic in nature due to UE's mobility. Mobility management generally shows only how to minimize packet loss during handoff, but does not show how to address service policies. This challenge was solved using smart local agents at base stations. The access stations act as mobility anchors for attached UEs. Upon handoff, the ongoing flows continue to use the old access station via the old path by leveraging the large number of access switches. The new flows traverse the new access switch, and new policy paths, to minimize path lengths.

C. Concepts behind Multi-Dimensional Aggregation

Traditional IP networks forward traffic based on destination prefix, and operators align IP prefixes with the topology to enable aggregation of contiguous prefixes. However, for a cellular network, destination-based routing is not flexible enough and forwarding is done on the basis of subscriber attributes or application types. Generally, VLAN tags or MPLS labels are used, which scales poorly as it results in flat-routing and removes the ability to aggregate contiguous entries, even if the destination is the same. Adding multiple tags or labels in the packet headers also incurs large overhead.

SoftCell combines benefits of location-based routing and tag-based routing, by using the ability of the SDN switches to selectively match different fields in the packet. SoftCell uses three dimensions to aggregate rules:

1) Aggregation by policy (policy tags): To minimize the number of rules in SDN switches, a policy tag is used to aggregate flows which are on the same policy path. This tag is associated at the access switch, thus allowing SDN switches to use a particular tag to forward packets along a particular path.

2) Aggregation by location (hierarchical IP address): Destination information are included in the UE addresses, so that the traffic can be aggregated by IP prefix. Thus, each base station is assigned an IP prefix, called as base station ID, and IDs of nearby stations can be further aggregated into larger blocks. This aggregation can be further increased by combining policy tags and IP address. If two policy paths going to two different base stations share a common long path segment, then a single policy tag can be assigned for that common path until the branching point, and at the branching point, traffic can be divided on the basis of IP prefix.
3) Aggregation by UE (UE ID): Packets are assigned an UE identifier (UE ID) at the base stations that enable differentiation between other UEs at the same base station. The base station prefix and the local UE ID form a hierarchical location-dependent address (LocIP) for the UE. This LocIP is used for routing in the PCE and Internet, but not RAN. When the UE attaches to the network, it is allocated a permanent IP address by the DHCP, while the LocIP changes when the UE moves between base stations. Translations between permanent IP and LocIP are performed by the Access Switches.

The Multi-Dimensional aggregation is performed by a greedy online algorithm, given a stream of policy paths as the inputs. It is performed online since policy paths can be dynamically installed or removed due to policy changes or MBX balancing.

D. Asymmetric Edge Design for Scalable Packet Classification

An appropriate policy-tag and and LocIP address is associated with each packet that arrives at the base station. SoftCell performs the key functionalities at the access edge so as to reduce overheads at the data-plane (to apply packet-classification rules) as well as the control plane (to fetch the rules).

Every packet that arrives at the base station is embedded (not encapsulated) with the packet-classification result within it. The policy tag, base station ID, and UE ID are embedded in the packet header. This ensures that return traffic from the Internet carries those fields. The policy tag is embedded as part of the source port. Due to this embedding mechanism, the three identifiers are implicitly piggybacked in return traffic and the gateway switch can simply forward the incoming packets based on the destination IP and port number. The following figure shows the format after the embedding mechanism:

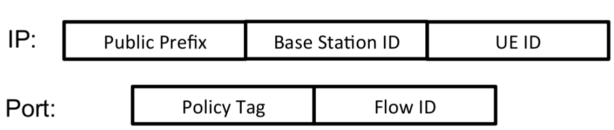

Fig. 6. Embedding location and policy information in source IP address and source port number.

In order to address security concerns which may be caused by malicious Internet hosts spoofing policy tags and congesting network links or MBXs, SoftCell uses Network Address Translation (NAT) as packets arrive from the Internet. NAT function picks up a different IP address and/or port number for every flow, regardless of whether or not the UE moves. As such, SoftCell provides the same level of security and privacy protection as a normal cellular network.

In order to reduce the overhead between access switch and controller, a local software agent running at each base station is used to scale the control plane. Each UE's list of packet classifiers are computed (based on service policy) when the UE first arrives at the base station and is then cached at the by the local agent at the behest of the controller. When UE initiates a new flow, the local agent consults these packet classifiers to determine the correct policy tag for the packets, thus reducing need for controller intervention for each new flow from UE.

E. Policy Consistency during Mobility

- Differentiating between old and new flows: Packets coming in for the old flow have destination IP address corresponding to UE's old location, so these packets traverse the old sequence of MBXs to reach the base station. The packets are merely redirected to the new base station, which then remaps the old LocIP to the UE's permanent address.
- Efficiently rerouting the old flows: SoftCell maintains long-lived tunnels between nearby base stations. These tunnels can carry traffic for any UEs that have moved between base stations. This 'triangle routing' ensures policy consistency and minimizes packet loss.

SoftCell handles controller failure by maintaining a distributed, consistent copy of the control-plane state. SoftCell handles local agent failure by simply restarting the local agent, which fetches the related state from the controller once again.

Performance evaluation of SoftCell was performed analyzing real LTE workloads and performing micro-benchmarks on the prototype controller as well as large scale simulations and it was found that scalability and flexibility of EPC was improved by using SoftCell.

F. Major contributions of SoftCell

- SoftCell EPC requires only simple core SD-switches and commodity servers for its implementation.
- MBXs can be easily deployed over SD-Switches using flow rules based solutions, or over VMs and the flows are directed to the respective VMs using the centralized controller.
- Dynamic nature of the cellular network and issues associated with handovers and flow mobility are also addressed in detail.
- SoftCell has been tested with results from real-time LTE logs; this gives credibility to its successful working.

G. *Open Issues and challenges with SoftCell architecture*

- Location to deploy MBXs to assure smooth UE traffic flows has not been discussed. In building SoftCell design, it has been assumed that all the MBXs have been placed at an ideal locations which result consistent UE traffic flow.
- No discussions have been made regarding minimizing access delays and reducing contentions in either EPC or RAN.
- No clear explanation has been given on the organization and distribution of multiple controllers over the entire network to avoid failure of controllers. It has only been stated that distributed controllers will be present, but there is no clear idea presented for such an implementation.
- Handling of QoS enforcement, mobility and security has only been briefly stated; these do not accompany standardized control protocols on the SDN controller.
- Predicate language rules for converting high level service policies into flow rules has not been presented.

IV. Critical Analysis of SoftRAN

The Radio Access Network (RAN) provides wide-area wireless connectivity to mobile devices. RAN solves the problem of how the limited spectrum available is managed, to ensure this connectivity. When the wireless deployment becomes dense with large number of mobile nodes and limited spectrum, allocation of radio resources, handover implementations, interference management, load balancing between cells, etc. becomes a difficult task to handle. SoftRAN is a fundamental rethink of the radio access layer. It is a software defined centralized control plane for RAN that abstracts all base stations in a local geographic area as a virtual big base station comprised of a central controller and radio elements (which are individual physical base stations). Basically, this is a way of saying that all the physical base stations in a geographical location are controlled by a central controller. By doing this abstraction, a local geographical network can efficiently perform load balancing and interference management, while maximizing throughput.

Increasing mobile traffic and limited spectrum has resulted in networks to become dense in order to increase capacity (known as cell splitting). However, due to the limited spectrum available, neighboring base stations within a dense deployment have to operate on the same channel, which is known as deploying networks with frequency reuse factor of 1. Handling dense networks with frequency reuse one is very difficult due to close interdependence in control plane decision making between nearby base stations, i.e., radio resource management decisions made at one base station will impact neighboring base stations and vice-versa. These dense deployments lead to clients of one base station to experience significant interference from neighboring base stations, thus degrading capacity. Also, the smaller coverage areas lead to rapid load fluctuation due to user mobility.

Traditional RAN is considered as a group of independent base stations, each capable of making independent control plane decisions and loosely coordinating with one other via mechanisms like SON (self organizing networks). However, in dense networks, multiple coordinated control plane decisions need to be made between several neighboring base stations simultaneously, with as low latency as possible. Distributed coordination algorithms need to work large number of base stations and so are not able to scale efficiently. This causes poor performance and reduction in capacity due to poor balancing of load and interference. Such distributed algorithms also become complex.

Rather than looking at RAN layer as a set of independent base stations, all base stations in a geographical area is abstracted as a virtual big-base station made up of radio elements. Thus, all neighboring base stations are considered to be allocating from a fixed shared resource, i.e., radio resources are abstracted as a three dimensional grid of space, time, and frequency slots and allocated programmatically in a software defined method through a logically centralized radio access control plane. APIs are defined between radio elements and control plane which allow the individual radio elements to describe the global view of the network to the controller as well as allow the controller to communicate radio resource management decisions back to radio elements.

The greatest challenge in such a software defined RAN is the inherent delay between centralized controller and individual radio elements. Generally, latency between the links is somewhere between 5-10 ms. This latency means that individual radio elements have a more updated view of the local state, and, as a result, the controller cannot be expected to perfectly allocate resources over long time scales due to quickly changing channel conditions at the radio elements themselves. The radio elements might themselves, in certain conditions, be able to manage themselves better due to having the updated view. To address this challenge, SoftRAN refactors the control plane functionalities between the radio elements and the controller.

A. *Design of SoftRAN*

The RAN seeks to meet its objective of managing resources while maximizing throughput, minimizing delay and maintaining fairness by taking the following actions with regard to the base station's data plane:

- Performing handovers between base stations
- Allocation of group of resource blocks to each flow; LTE uses OFDMA, where radio resources are split into time and frequency slots, which are called resource blocks; these blocks must be smartly given to the competing clients since channel quality and interference for each client can vary
- Assignment of transmission power to each resource block at each base station

These actions form the control plane of RAN and can be termed as control decisions. Control decisions between neighboring base stations are interlinked with each other.

B. *Virtual Big Base Station Abstraction*

Instead of the distributed architecture which is used in LTE networks, SoftRAN proposes a centralized architecture which abstracts out all the base stations deployed in a geographical area as a virtual big-base station with individual radio elements having minimal logic. This logically centralized controller makes all the decisions for the individual radio elements in the geographical region. The controller of the big base station maintains a global view of the entire RAN and provides a framework on which control algorithms can be run.

The radio resources in the RAN are abstracted out as a 3D resource grid, with base station index, time and frequency as the three dimensions. Each time-frequency slot at each base station will require a decision to be made. The transmit power used and the flow that would be served for each block of the 3D grid, needs to be assigned by the controller. From the perspective of the network operator, all the radio elements in an area can be thought of as a big base station with a 3 dimensional resource grid; but from the perspective of UE, we cannot strictly accomplish a single base station abstraction without changes to the LTE standard. The UE (or client) will continue to sense multiple base stations and perform traditional handshakes with both the previous and new station. However, centralized control will lead to much smoother handovers as well as reduce dropped connections and multiple handovers between the same pair of base stations (ping-pong). This would result in the clients experiencing a much stable connection for cellular networks.

C. *Realization of Big Base Station*

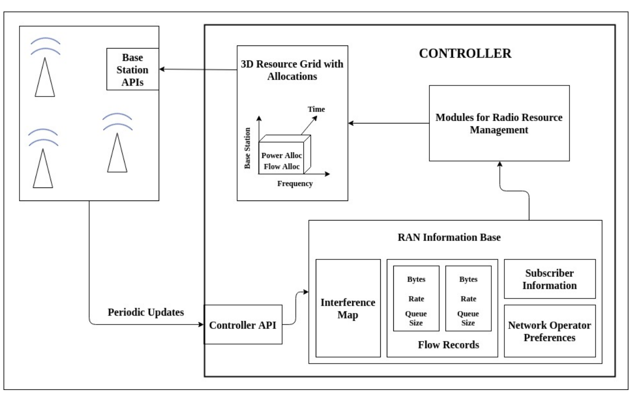

Fig. 7. SoftRAN Architecture

Fig. 7 shows how SoftRAN achieves the big base abstraction architecturally. There are two main challenges with realizing such architectures:

- Taking care that delay between controller and radio elements do not negatively affect performance.
- Designing a controller which can provide a framework for different control algorithms to operate on.

1) Controller Architecture: The centralized controller receives periodic updates from all the radio elements regarding the local network state within the geographic region. These updates are used for maintaining and updating the global network state in a database, called here as RAN Information Base (RIB). The RIB conceptually consists of:

1) Interference Map: This is a weighted graph, where the weighted edges stand for the channel strength between the nodes and the nodes themselves represent a radio element or an active client in the geographic region.
2) Network Operator Preferences: These are used to prioritize certain flows over others; controlled by the network operator, and he/she can enter his/her preferences into the RIB.
3) Flow Records: These are records of relevant parameters of an ongoing flow, for eg., average transmission rate, number of bytes transmitted, number of packets queued, etc.

The network operator deploys various control modules for accessing RIB; these control modules take the decisions needed for radio resource management, i.e., they assign groups of resource blocks to clients, while simultaneously specifying transmission powers to be used by each of the individual physical base stations (radio elements) at each resource block.

2) Refactoring of Control Plane: Due to inherent delay between controller and radio elements, the radio elements are in a more updated local network state as compared to the controller. Hence, in spite of the coordination provided by the centralized controller, the control decisions that are dependent on the rapidly varying parameters of the network can be optimized only at the radio element. As a result, the control functionality between controller and radio element has to be refactored.

The following are the guiding principles for refactoring of control plane:

1) The control decisions which affect the decision making at neighboring radio elements must be made at the controller, since they need to be communicated across to the neighboring radio elements.
2) The control decisions that depend on continuously varying parameters should be made at the radio element, since the inherent delay between the controller and the radio element increases the response time to these continuously changing parameters.

The following tasks are grouped on the basis of these two principles:

- Handovers: Since handovers influence decision making at the neighboring radio elements by increasing the load at the neighboring radio elements, these control decisions will be taken the centralized controller.
- Transmission power of the radio elements: Transmission power influences the interference at the neighboring radio elements; as a result, transmission power allocated per resource block is maintained by the controller.
- Resource Block Allocation: Since the transmit powers used by the radio element are known, neighboring radio elements do not need to know which clients are being allocated resource block. Hence, on the down-

link, resource block allocation doesn't impact decision making in the neighboring radio elements. Also, channel measurements reported by the client will affect the resource block allocation among competing clients, which takes about 2ms to occur. As a result, it can be observed that the local radio elements are at the more updated network state, i.e., decisions will be based on continuously changing parameters; thus, at down-link resource allocation can be performed by the radio element. However, during up-link, the scenario is reversed. The client which will transmit on each particular resource block is decided by the up-link resource block allocator, which will affect the up-link interference from the perspective of neighboring radio elements. As a result, up-link resource block allocation is decided at the controller.

D. Major contributions of SoftRAN

- A good architecture has been proposed to design software defined RAN with Virtual Big-Base Station and a centralized controller.
- Detailed explanation regarding separation of radio access plane tasks between the controller and base stations.
- The proposal has many potential use cases like centralized interference management and load balancing.

E. Open Issues and challenges with SoftRAN architecture

- There is no communication protocol defined between eNBs and centralized controller. This protocol has to defined for SoftRAN to a viable product.
- Since there has been no practical tests, studies about practical limitations for deploying centralized RAN control algorithms are required to be done.
- Controller scalability has not been discussed in the paper.

V. Conclusion

During the course of the project, need for SDN paradigms in LTE architecture has been presented and how SDWN architectures can improve and simplify LTE cellular networks has been seen. Two different proposals - SoftCell (in which SDN concepts have been applied to LTE EPC component) and SoftRAN (in which SDN concepts have been applied to LTE RAN component) - have been discussed in detail. Studying such literature has provided great insights into the novel methods in which SDN can be used for wireless networks. It has enabled the reader to delve deeper into the vast and highly productive realm of Software Defined Networking.

References

[1] William Stallings, Foundations of Modern Networking - SDN, NFV, QoE, IoT and Cloud; Pearson Publications: November, 2015

[2] Doug Marschke, Jeff Doyle and Pete Mayer, SDN: Anatomy of OpenFlow; Lulu publications: March, 2015

[3] Jin, X., Li, L. E., Vanbever, L., & Rexford, J. (2013). Softcell: Scalable and flexible cellular core network architecture. In Proceedings of the ninth ACM conference on emerging networking experiments and technologies (pp. 163174). ACM.

[4] Open Data Center Alliance. Open Data Center Alliance Master Usage Model: Software-Defined Networking Rev. 2.0. White Paper. 2014.

[5] Gudipati, A., Perry, D., Li, L. E., & Katti, S. (2013). Softran: Software defined radio access network. In Proceedings of the second ACM SIGCOMM workshop on hot topics in software defined networking (pp. 2530). ACM.

[6] Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. ONF White Paper, April 13, 2012.

[7] Software Defined Wireless Networks: A Survey of Issues and Solutions; Anil Kumar Rangisetti & Bheemarjuna Reddy Tamma; Published online: 14 August 2017 Springer Science+Business Media, LLC 2017

[8] ISG NFV. Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges & Call for Action. ISG NFV white paper, October 2012.

[9] ETSI GS NFV Architectural Framework 002 v1.1.1, October-2013