Report on

On

Towards a Software Defined Network based controller for Management of Large Scale WLAN

Submitted in partial fulfillment of the requirements
of the degree of

Master of Technology

by

Aniruddh Rao K
Roll No. 133079005

Supervisor:

Prof. Abhay Karandikar

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

October, 2015

To Amma & Appa

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Abhay Karandikar for the invaluable guidance, opportunities and infrastructure support provided by him over the last two years. I would like to thank my team Mahak, Navneet, Mayank, Akshatha, Nishanth sir and Ojas for their support in doing this project. I would also like to thank all my lab mates for their support.

October 4, 2015 **Aniruddh Rao K**

Abstract

Recent growth in demand for Internet data and mobile communications, calls for innovation in communication technology to increase coverage, reachability, capacity and also use existing technologies effectively. Deploying WLAN(Wireless LAN) on a large scale is one of the solutions which telecommunication operators are keen to implement to cater to the increased demand for high speed data services. Management of large scale deployment of WLAN is a challenge and is best addressed by centralized management. Standards such as CAPWAP(Control and Provisioning of Wireless Access Points), TR-069(Technical report 069) provide a handle to centralized control and management of WLAN. A controller using CAPWAP standard to manage WLAN is implemented as part of this project.

Large scale deployment of WLAN would need APs(Access Points) of different vendors to be managed by a single controller but the existing standards have interoperability issues. The exisiting standards also have mutiple issues in managing the WLAN effectively. A protocol stack is proposed to be standardized with TSDSI(Telecommunication Standards Development Society in India) to address these issues and manage WLAN efficiently.

Also the existing WLAN controllers have very less ability to change policies on the go and control network dynamically. An emerging technology in the field of networking called SDN(Software Defined Networking) makes modifying the policies easier and uniform. It also gives view of entire network and assists in effiecient management of the WLAN. An SDN based architecture for dense WLAN controller is also proposed in this thesis which would enable dynamic management of the network.

Contents

List of Figures

Abbreviations

UHF	Ultra High Frequency
WLAN	Wireless Local Area Network
SDN	Software Defined Networking
AP	Access Point
ONF	Open Networking Foundation
CAPWAP	Control And Provisioning of Wireless Access Points
RFC	Request For Comment
WTP	Wireless Termination Point
MAC	Medium Access Control
CPE	Customer Premise Equipment
WAN	Wide Area Network
CWMP	CPE WAN Management Protocol
TR069	Technical report 069
SOAP	Simple Object Access Protocol
ACS	Automatic Configuration Server
AC	Access Controller
FSM	Finite State Machine
DTLS	Datagram Transport Layer Security
UCI	Unified Configuration Interface
STA	Station
TSDSI	Telecommunications Standrads Development Society in India
ISP	Internet Service Provider
TSP	Telecom Service Provider
QOS	Quality Of Service

Chapter 1

Introduction

An initiative called **Digital India** is started by the Government of India in July 2015 to increase the internet connectivity and online infrastructure in India. The initiative also includes connecting rural India with high speed networks. In areas with irregular landscapes and sparse population, rolling out optical fiber cable to provide a wired broad band would be difficult and expensive. Recent research in the use of TV UHF band or TV white spaces to provide broad band connectivity would address this issue. It is demonstrated by testbeds laid out at Palghar district, Maharastra by my colleagues of infonet lab,IIT Bombay, and at srikakulam district, Andhra by Microsoft (with Ernet) that wireless broadband can be provided by deploying WiFi access points as access network and using TV white space to back-haul.

Also there is a plan to increase internet connectivity in public places by deploying WiFi on a large scale at bus stands, railway stations, airport etc. Google has proposed to provide WiFi connectivity at 500 railway stations. A plan is proposed by Delhi Government to deploy WiFi throughout the Capital of India. Thus deployment of WiFi in a large scale is one of the major steps that will be taken to increase internet connectivity across India.

1.1 Central management of large scale WLAN

In a large scale deployment of WLAN, thousands of APs are deployed at a site. Configuring each AP individually is a tedious task. In case APs need to be updated with new firmware or configurations, it becomes a very tedious task and brings in inconsistency in management of WLAN. To optimize the WLAN performance i.e address interference, divide load etc. there needs to be some inter AP communication or a centralized information database that provides a handle to monitor the entire WLAN deployed and clients connected to the WLAN. Thus a centralized management scheme for WLAN would address these issues and make the WLAN management efficient. It can also be used to centrally manage client connections to each AP in the WLAN.

1.2 SDN paradigm for WLAN controller

Software Defined Networking (SDN) is a paradigm that re-organizes the network architecture by seperating the forwarding plane and control plane. It removes computational load from the end routers and places it in a centralized controller. This simplifies the end devices and reduces the network infrastructure cost. SDN also gives the network administrators a complete view of the network and helps in making better routing decisions. ONF (Open Networking Foundation) defines the SDN architecture [9] as

1. Directly programmable: Network control is directly programmable because it is decoupled from forwarding functions.

2. Agile: Abstracting control from forwarding lets administrators dynamically adjust network-wide traffic flow to meet changing needs.

3. Centrally managed: Network intelligence is (logically) centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch.

4. Programmatically configured: SDN lets network managers configure, manage, secure, and optimize network resources very quickly via dynamic, automated SDN programs, which they can write themselves because the programs do not depend on proprietary software.

5. Open standards-based and vendor-neutral: When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols.

SDN paradigm to centrally manage a large scale WLAN would make management simpler and efficient. It would remove processing load from end access points, making the access points light low cost radio devices and also would enable dynamic control of the flow of traffic. In summary, it would give all the advantages that a SDN gives in a wired network and in addition enables easier management of WLAN by a centralized controller.

In this thesis, we present a implementation of WLAN controller based on CAPWAP protocol. The shortcomings of existing standards in managing WLAN are listed and a new extensible protocol is proposed. Moving towards SDN paradigm, a study of various architectures and implementations of SDN controller for WLAN like Odin [3], ethanol [4] and OpenRoads [2] (Openflow Wireless) is presented. Also a SDN controller architecture is proposed for management of large scale WLAN deployment.

Chapter 2

Central WLAN Management

2.1 Large scale WLAN

2.1.1 Existing standards:CAPWAP and TR069/CWMP

There are standard protocols for centralized management of WLAN. CAPWAP(Control And Provisioning of Wireless Access Points) is one such standard protocol defined under IETF. TR-069(Technical Report 069) is another standard. It is a technical specification published by broadband forum and is entitled CWMP(CPE WAN Management Protocol).

CAPWAP

CAPWAP Protocol, is a standard (RFC 5415 [7]), inter-operable protocol that enables an AC to manage a collection of WTPs. CAPWAP protocol is defined to be independent of L2 technology. Traditional protocols for managing WTPs are either manual static configuration via HTTP, proprietary L2 specific or non-existent. CAPWAP assumes a network configuration consisting of multiple WTPs communicating via the IP to an AC. The CAPWAP protocol transport layer carries two types of payload, CAPWAP Data messages and CAPWAP Control messages. CAPWAP Data messages encapsulate forwarded wireless frames. CAPWAP Control messages are management messages exchanged between a WTP and AC.
RFC for the CAPWAP protocol states the following goals.

1. Centralize the authentication and policy enforcement functions for a wireless net-

work.

2. Remove processing load from WTP by moving it away towards AC except for time critical functions.

3. Provide extensible protocol not bound to a specific technology.

As the CAPWAP protocol is not bound to a specific wireless technology, bindings are written to support the use of CAPWAP protocol with IEEE802.11 WLANs in RFC 5416 [8]. Bindings specify the implementation of CAPWAP for a specific wireless technology. CAPWAP protocol supports two modes of operation: Split MAC and Local MAC. In split MAC mode all the wireless data and management frames are encapsulated via the CAPWAP protocol and exchanged between AC and WTP along with CAPWAP control messages. While in local MAC mode, the data frames are either locally bridged or tunneled as 802.3 frames and management frames are processed by the WTP and then forwarded to the AC.

TR069/CWMP

TR069 [6] is the specification document published by the Broadband Forum that specifies the CPE WAN Management Protocol (CWMP). CWMP is intended for communication between a CPE and Auto-Configuration Server(ACS). CWMP defines a mechanism that encompasses secure auto-configuration of CPE, and also incorporates other CPE management functions into a common framework. It supports a variety of functionalities to manage a collection of CPE which includes:

1. Auto-configuration of CPE and dynamic service provisioning

2. Software/firmware image management of CPE

3. Status and performance monitoring

4. Diagnostics

TR069 defines a SOAP/HTTP based protocol that provides communication between CPE and ACS with typical positioning of ACS and CPE as show in figure 2.1. The protocol vastly addresses the issue of automatic configuration and management of different internet access devices like modems, routers, gateways etc by a remote server i.e. ACS.

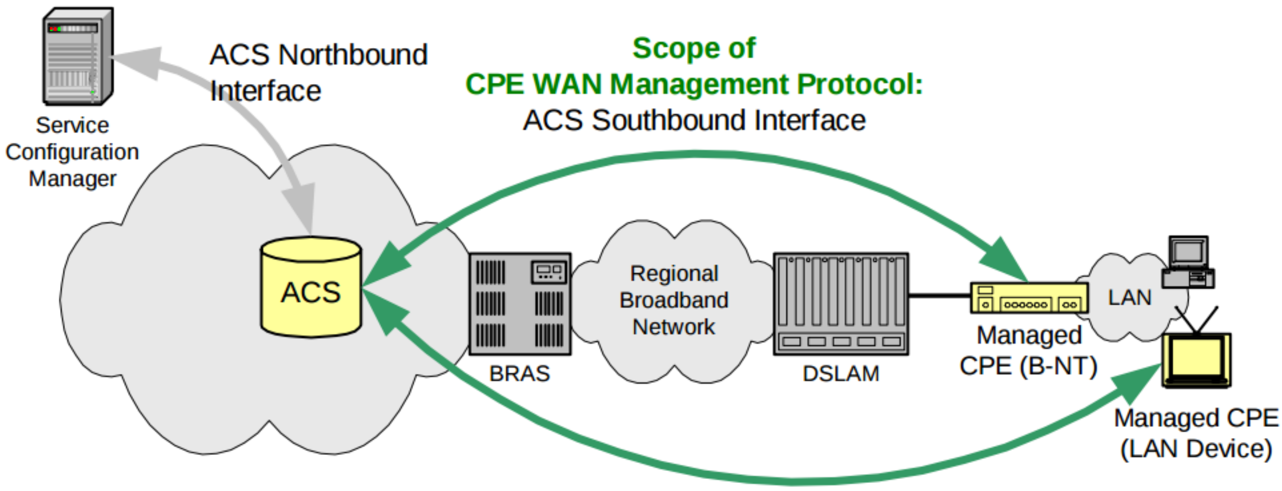

Figure 2.1: Positioning of components in Auto configuration Architecture using CWMP to configure CPE [6]

2.2 WLAN Controller based on CAPWAP protocol

A CAPWAP based open source WLAN controller is implemented by a team of 5 of us: Navneet, Mayank, Aniruddh, Mahak and Pravesh.

The objectives of the implementation are:

1. Setup a CAPWAP tunnel between AC and WTP as specified in RFC 5415.

2. Manage multiple vendor APs by same AC. i.e. make the implementation interoperable.

3. Demonstarate a subset of configuration and management of WLAN by writing IEEE 802.11 bindings as specified in RFC 5416 [8].

Our implementation of CAPWAP controller is based on a opensource implementation of CAPWAP called OpenCAPWAP [5] by M.Bernaschi et.al. , IAC-CNR Rome, Italy. The CAPWAP based controller is implemented as a user space program in linux written in C language. An application written in C is installed at WTP to enable communication with AC.

To control WTPs/APs of different make or APs of different vendors i.e. APs having different WiFi chipsets by the same client application, APs are ported to OpenWRT and

CAPWAP client application is installed in each AP. The reason to port APs to Openwrt is that, each vendor uses a different driver based on the chipset of WiFi module to configure the radio interface and makes it difficult to configure different vendor APs using same CAPWAP client application. OpenWRT gives a command line interface called UCI which can configure APs across different chipsets and enables the CAPWAP controller to manage APs of different vendor or chipset through same client application.

2.2.1 CAPWAP engine

The engine for setting up CAPWAP tunnel follows the finite state machine as specified by CAPWAP RFC 5415 except that there is no state for firmware update. Firmware update is not addressed in this implementation. The FSM is shown in figure 2.2.

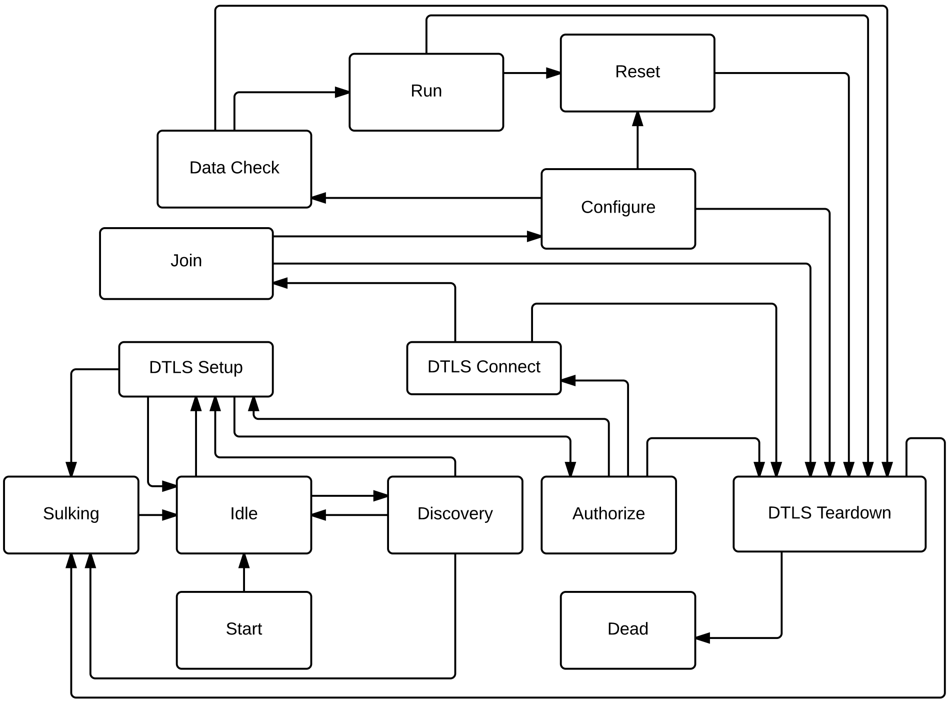

Figure 2.2: Finite State Machine of CAPWAP Protocol

As specified by the RFC, both AC and WTP follow the same FSM but some transitions occur in only one of them. The state transitions are different in both AC and WTP as WTP communicates to a single AC while an AC communicates to multiple WTPs. A heart-beat message exchange is setup between AC and WTP to monitor the connectivity

of WTP to AC and its availability in WLAN. The CAPWAP protocol implemented is local MAC mode. All the data frames are locally bridged. All the 802.11 management frames except probe and beacon frames are processed and forwarded to AC in CAPWAP tunnel while all the 802.11 control frames are locally addressed at WTP.
Each AC runs three threads:

1. Discovery thread: The AC's Discovery thread is responsible for receiving, and responding to, Discovery Request messages from WTP.

2. Listener thread: The AC's Listener thread handles inbound DTLS session establishment requests. Once a DTLS session has been validated, which occurs when the state machine enters the "Authorize" state, the Listener thread creates a WTP session-specific Service thread and state context.

3. Service thread: AC's service thread handles the per WTP states and one such state exists per-WTP connection. This thread is used for any configuration or communications between AC and WTP. When communication with the WTP is complete, the Service thread is terminated and all associated resources are released.

The session starts with WTP in the START mode and after initialization it changes to IDLE mode. In IDLE mode it either connects to a fixed AC if it is explicitly instructed to connect to it or it enters the DISCOVERY phase to find an AC by sending a discovery request. After the DISCOVERY state, it proceeds to DTLS SETUP state where it establishes a secure connection between the access point and the controller. After the DTLS SETUP, it moves to AUTHORIZE state where the DTLS stack needs authorization for the session establishment. After successful authorization it enters the DTLS CONNECT state and after connection establishment, it changes to JOIN state where the WTP and AC communicate with each other and CAPWAP session begins. The AC then sends a successful join response message to the WTP. After receiving the join response message, the WTP is configured through a set of commands in the CONFIGURE state. The WTP reaches the RUN state after confirmation of successful configuration in the DATA CHECK state. From the RUN state, it may be asked to update the configuration in the CONFIGURE state or may lose its connectivity with the AC which will change the state to SULKING. After SULKING state, the WTP goes back to the DISCOVER state after temporary transition to the IDLE state.

These state transitions setup a DTLS CAPWAP tunnel between AC and WTP. All configuration message exchanges are addressed by the bindings written as applications at both AC and WTP.

2.2.2 IEEE 802.11 bindings

CAPWAP protocol is not bound to a wireless technology and bindings are written to support its use for a specific wireless technology. RFC 5416 specifies bindings for 802.11 WLAN and a subset of bindings are implemented here.

We follow an architecture where the CAPWAP bindings are written as applications on either sides i.e. at AC and WTP. The architecture is shown in figure 2.3. This architecture would make addition of new bindings easier as it needs addition of a message structure only at the applications with no change in the main CAPWAP engine.

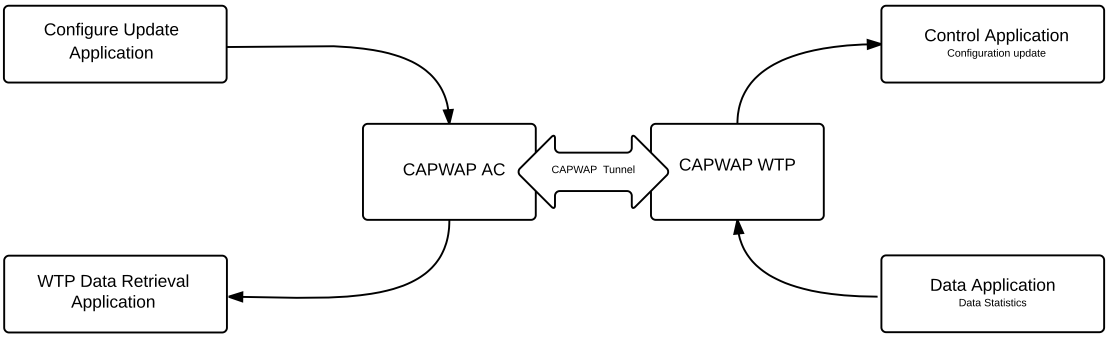

Figure 2.3: Architecture of IEEE 802.11 binding applications

Applications at AC for IEEE 802.11 bindings

1. Configure Update Application: This application is used to build and send configuration messages to WTP. The application builds a message according to IEEE 802.11 bindings defined in RFC 5416 and sends it to AC. AC encapsulates the message in a CAPWAP header and forwards it to WTP.

2. WTP Data retrieval application: This application will retrieve the information data from different WTPs. AC will receive information data periodically from all connected WTP. The data packet will contain WTP-ID to identify which WTP

has sent the data. Present implementation only processes the connected Stations information from every WTP.

Applications at WTP for IEEE 802.11 bindings

1. Control application: Handles the configuration update requests from AC. WTP CAPWAP client thread on receiving a configuration update message, forwards it to control application. Control application parses it and updates the configuration using UCI.

2. Data application: This application sends various statistics from WTP to AC. Data application periodically fetches statistics from the radio, formats as per the bindings and forwards it to WTP main thread that forwards the data to AC.

Bindings currently implemented

- SET CHANNEL: Configuration message to set channel of operation of WTP. Binding specified in section 6.10 RFC 5416.

- SET TX POWER: Configuration message to set transmit power of WTP. Binding specified in section 6.18 RFC 5416.

- STATISTICS: Data sent by WTP to AC. Contains statistics of data forwarded to stations from WTP. Binding specified in section 6.16 RFC 5416

Chapter 3

Extensible Protocol for Panagement of Large Scale WLAN

3.1 Limitations of existing standards

The existing standards viz CAPWAP and TR069 employ the use of some sort of Controller architecture which helps in managing a dense/large network of Wireless APs/CPEs. While these standards do spell out a lot of grammar for addressing the management of such networks, there are a few essential gaps which these standards fail to address. The following section enumerates those in the context of CAPWAP and TR069.

3.1.1 Gaps in CAPWAP and TR069

Local bridging of data

In CAPWAP, there are two types of MAC implementation based on division of labor between AC and WTP on handling the different IEEE 802.11 frames. Details are in section 2.2 of RFC 5416 [8]

1. Split MAC: In split MAC distribution and integration functions services reside on the AC. All user data is tunneled to AC. All real time 802.11 services (802.11 Control frames) including beacon and probe responses are handled by WTP. All remaining 802.11 management frames are handled by AC. All user data will be tunneled as native 802.11 frames.

2. Local MAC: In Local MAC, all the 802.11 frames are processed at WTP. User data may be locally bridged or forwarded to AC or tunneled as 802.11 frames or 802.3 frames.

Current Split MAC feature would give the advantage that it would make WTP free from processing various 802.11 frames but would add processing load on AC as it has to process all the user data. Local MAC would give the advantage of bridging data locally which would reduce processing load on the AC. There is no support for Split MAC with a local break out of data in CAPWAP. This architecture would need WTP to process a limited set of frames and also removes the load of processing data frames on AC. It would also make the WLAN extensible as support for handling of any new message frames introduced by 802.11 amendments only need to be processed at AC.

Fast Roaming

In CAPWAP, re-association follows either of these procedures

1. A full association procedure with all the authentication steps, security and policy message exchanges (to new WTP) followed by a dissociation procedure (to old WTP).

2. Association of STA to new WTP is assisted by access controller which caches the authentication keys avoiding the 802.1X authentication step and would reduce handover time.

There is no support for IEEE 802.11r in CAPWAP which would facilitate fast handover where as TR069 does not address the mobility of STA in WLAN at all.

Interoperability

The existing CAPWAP standard has only a limited set of functionalities specified. There is no specifications regarding client authentication, Virtual access points, slicing the network and many other recent features. Manufacturers of APs and controllers would do proprietary implementations of these recent features that they would like to provide but not standardized. This would lead to interoperability issues i.e. a single controller will not be able to control different vendor APs. Thus there is a need to develop an extensible protocol and standardize it.

3.2 Proposal for a new extensible protocol

A work item is proposed by Prof. Abhay Karandikar with TSDSI(Telecommunications Standards Development Society in India) to develop a protocol for large scale WLAN management and address the above mentioned issues in the existing standards. The work item is approved and a work group is formed to develop the standard. We, a team of 3 people representing IIT Bombay, are working with telecom operators like Tata tele services, Reliance and telcom vendors like Ericsson, Huawei and Intel to develop a protocol for management of large scale WLAN. Currently the work is in requirement analysis phase.

Requirement analysis is done by listing out use cases. We list out use cases that are not addressed by the existing standards and develop protocol accordingly. Two such use cases that were discussed in the work group are explained below. some typical scenarios wherein a large number of Wireless Access Points are deployed are taken into consideration and use cases are listed out.

Use case 1

An ISP/TSP has WLAN deployments in various residential apartments and office buildings which are spread across a township. The WLAN provides connectivity to the residents/employees both within and outdoors. A new office building has recently been constructed and needs similar WLAN deployment. The ISP/TSP is looking to deploy the same network equipment in the new office but cannot find the same network equipment that is being used in the earlier deployments as those models are no longer being manufactured by the vendor. The ISP/TSP invites a quote from a different vendor and ends up installing its equipment in the new location. The ISP/TSP would like to lay the network such that the new WLAN deployment is also managed by the controller that is used to manage all other deployments in the township.

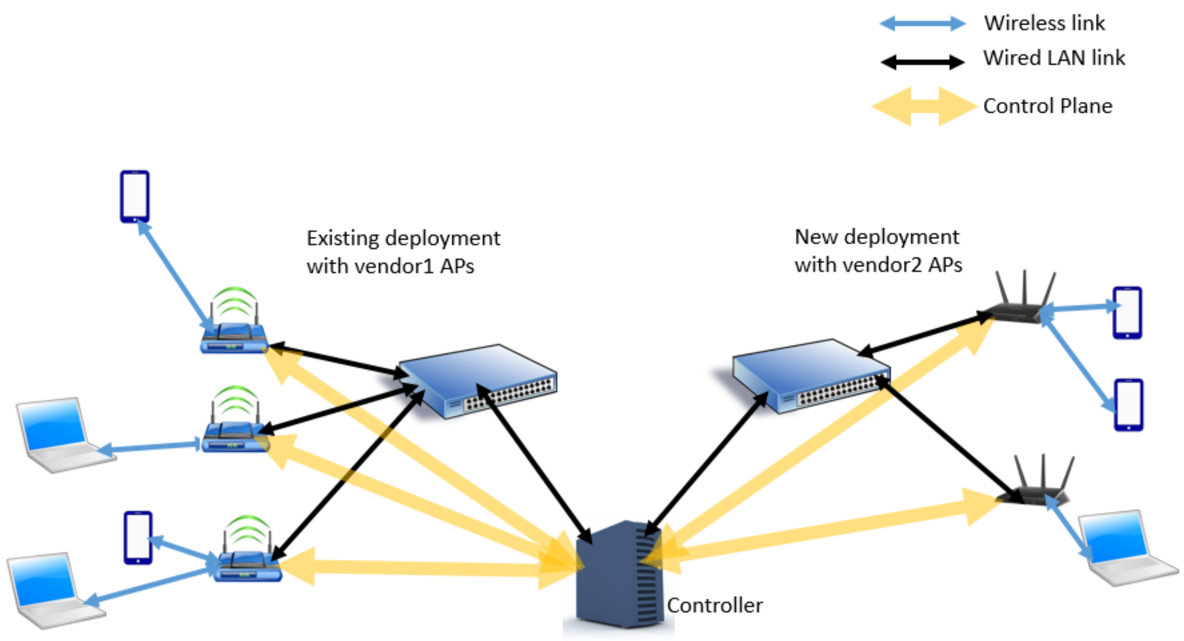
Wireless link
Wired LAN link
Control Plane
Existing deployment with vendor1 APs
New deployment with vendor2 APs
Controller

Figure 3.1: USECASE 1: Different model/vendor APs managed by same WLAN controller

This use case highlights the need for the interoperability issue to be addressed by the protocol. The protocol developed must enable interoperability i.e. same controller must be able to manage different vendor APs as shown in figure 3.1.

Use Case 2

Community WiFi networks allow service providers to leverage unused capacity on existing WiFi infrastructure to offer WiFi network access to visitors and passersby. An operator can also use this excess capacity to offer services to retail and roamingpartner operators subscribers. The residential subscribers accessing the network from inside their homes have prioritized access to the WiFi resources. The residential WiFi infrastructure is configured in a manner that allows for a secure and independent access channel to retain service quality, safety, and privacy for both residential and visitor customers. Roaming users are only allowed to use the WiFi network capacity that is not currently used by the subscriber at home.

Basically, the wireless AP in the home will provide two networks: a private one for the home owner/subscriber, and a community network for on-the-go subscribers passing through the neighborhood. While the user is at home, all of their WiFi devices (smart phone, tablet, etc.) should automatically connect to the private network. When the user travels outside the vicinity of their APs coverage area, and passes in range of another AP operated by the same service provider, their client devices will be able to connect to the public network.

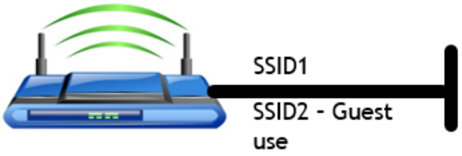

Figure 3.2: USE CASE 2: Dividing WLAN into multiple virtual networks

As shown in figure 3.2, this use case explains the need for the protocol to divide WLAN into multiple networks and manages QOS, access control to users connected.

Chapter 4

SDN based Controller for WLAN management

4.1 Need for bringing in SDN paradigm to manage WLAN

Most of the enterprise WLAN management solutions provided are closed and proprietary. The existing standard protocols do not provide messages or message formats for all the recent feature additions and IEEE 802.11 amendments, leading to their implementation by manufacturers in a non standard and proprietary manner. Thus, using different vendor equipments in a WLAN has become almost impossible. Also, the ability to configure network dynamically is a issue which needs to be addressed. This would make the network management more easier, efficient and would empower the network owners.
SDN in WLAN management would have the following advantages

- **Empower network owners and increase pace of innovation:** SDN would enable dynamic configuration of the network. It would allow the network owners to implement and test different algorithms for interference management, roaming, load balancing and various other policy management. It would enable the network owners to slice the network into production setup and test setup. The test setup slice can be used to experiment and test any new algorithms without affecting the production setup.

- **Diversify the supply chain:** Operating at a level of abstraction, SDN provides standard interfaces for communication. The implementation details below the level of abstraction need not follow a standard and different vendors may implement it differently. This would encourage more number of vendors as they need to provide only a standard interface as part of their product and still follow their proprietary implementation for the operations under the interface provided.

4.2 Study of existing work using SDN in WLAN

A lot of research has been going on in bringing SDN paradigm in wireless access networks and also wireless back-haul networks. Some of the work on SDN in WLAN management are presented in detail in this section.

OpenRoads [2] was a first step towards using openflow in wireless networks. Openroads proposes on how applications at a central controller can address mobility, load balancing etc. in a wireless network. Openroads architecture has three layers.

- Flow layer: In this layer, data flow in the network is managed by openflow and SNMP
- Slicing layer: This layer slices the data path and SNMP configurations for different access points in the network.
- Control layer: Applications at controller decide on how data should flow and configure access points.

Odin [3] proposes a SDN based enterprise WLAN management scheme. The objective of odin is to empower network owners to program and provide WLAN services and features as network applications. Odin controller has a master that speaks to switches and APs using openflow protocol. Odin addresses mobility efficiently, by creating a light virtual access point per client connected in the network.

Another prototype, ethanol [4] proposes a architecture where controller talks to switches using openflow and to APs using XML/HTTP based protocol. Ethanol demonstrates features like load balancing, QoS management as applications.

4.3 Hierarchical architecture for SDN based controller

We propose a hierarchical SDN based controller to manage WLAN. The typical positioning of controllers are shown in the figure 4.1. Local controller would manage time critical or delay constrained features like mobility management, localized features like load balancing etc and global controller would manage features like global policy management, QoS management, authentication etc.

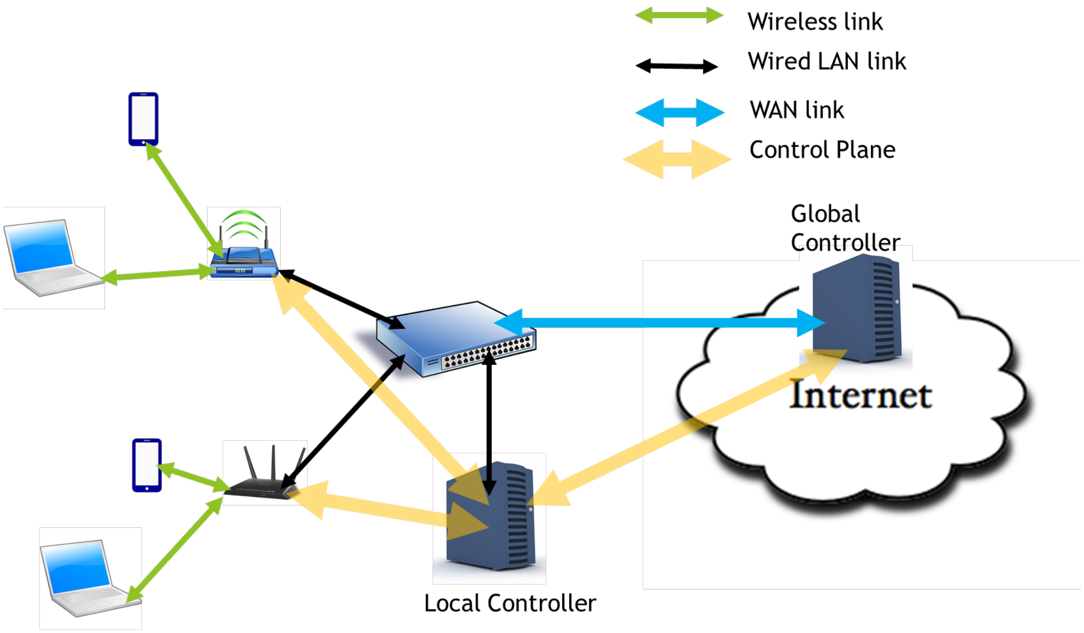

Figure 4.1: Hierarchical controller setup: Global and local controller

The proposed architecture for the controller is shown in figure 4.2. The controller has two modules within, an openflow controller and a wireless AP controller. There are two interfaces to the controller, a north bound interface and a south bound interface. Through south bound interface, openflow controller manages switches using openflow protocol and wireless AP controller manages APs. The protocol for communication between controller and wireless APs is yet to be designed based on various use cases. Applications implementing features like mobility management, load balancing etc. communicate to controller through north bound interface. Also a controller can talk to another controller that may be at a higher level or a lower level in the wireless network hierarchy.

The protocol for communication between controller and wireless APs is designed such that it is wireless technology independent. Bindings are written to support a specific wireless technology. In a network with TV UHF band back-haul network and WLAN deployed as access network, the same protocol can be used to manage TV UHF APs and also WLAN APs. This enables same controller hierarchy to manage different wireless access technology APs.

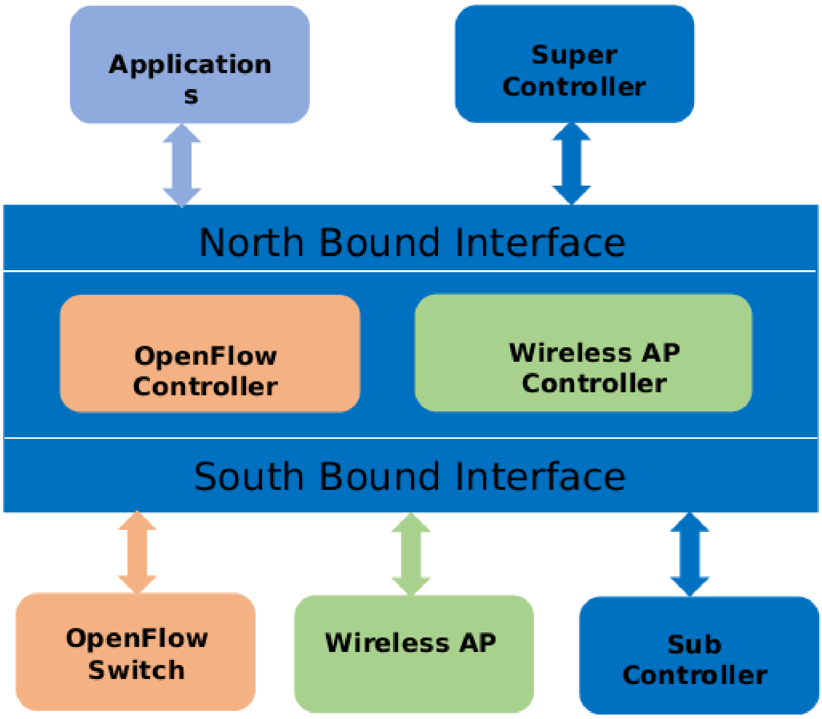

Figure 4.2: Architecture of SDN controller

Chapter 5

Future Work

Implementation of SDN based WLAN controller with a hierarchical architecture

After the study of various SDN WLAN controller architecture, we came up with a hierarchical architecture for SDN WLAN controller. This architecture would be implemented and verified in the next phase of this project.
Applications can be written for various features and performance study can be made on a prototype controller.

Extensible protocol for WLAN control and management

As mentioned in chapter 3, the work on standard protocol development for WLAN control and management is in requirement analysis phase currently. More use cases would be considered to come up with detailed requirements, discussed in the work group that includes various TSPs and telecommunication companies and a extensible protocol will be developed and standardized with TSDSI.

Bibliography

[1] Nachikethas A. Jagadeesan, Bhaskar Krishnamachari, "Software-Defined Networking Paradigms in Wireless Networks: A Survey". ACM Comput. Surv. 47, 2, Article 27, November 2014.

[2] Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Te-Yuan Huang, Michael Chan, Nikhil Handigol, and Nick McKeown, "OpenRoads: empowering research in mobile networks", SIGCOMM Comput. Commun. Rev. 40, 1 (January 2010), 125-126, 2010.

[3] Lalith Suresh, Julius Schulz-Zander, Ruben Merz, Anja Feldmann, and Teresa Vazao, "Towards programmable enterprise WLANS with Odin", In Proceedings of the first workshop on Hot topics in software defined networks (HotSDN '12). ACM, New York, NY, USA, 115-120, 2012.

[4] Moura, H.; Bessa, G.V.C.; Vieira, M.A.M.; Macedo, D.F, "Ethanol: Software defined networking for 802.11 Wireless Networks", Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium, 2015.

[5] M. Bernaschi, F. Cacace, G. Iannello, M. Vellucci, and L. Vollero, " OpenCAPWAP: An open source CAPWAP implementation for the management and configuration of WiFi hot-spots", Comput. Netw. 53, 2 (February 2009), 217-230, 2009.

[6] https://www.broadband-forum.org/technical/download/TR-069.pdf

[7] https://tools.ietf.org/html/rfc5415

[8] https://tools.ietf.org/html/rfc5416

[9] https://www.opennetworking.org/sdn-resources/sdn-definition