Towards Software Defined Networks to Manage Large Scale WLAN

Presented by Aniruddh Rao

Date: 19th October, 2015

Supervisor: Prof. Abhay Karandikar

Motivation

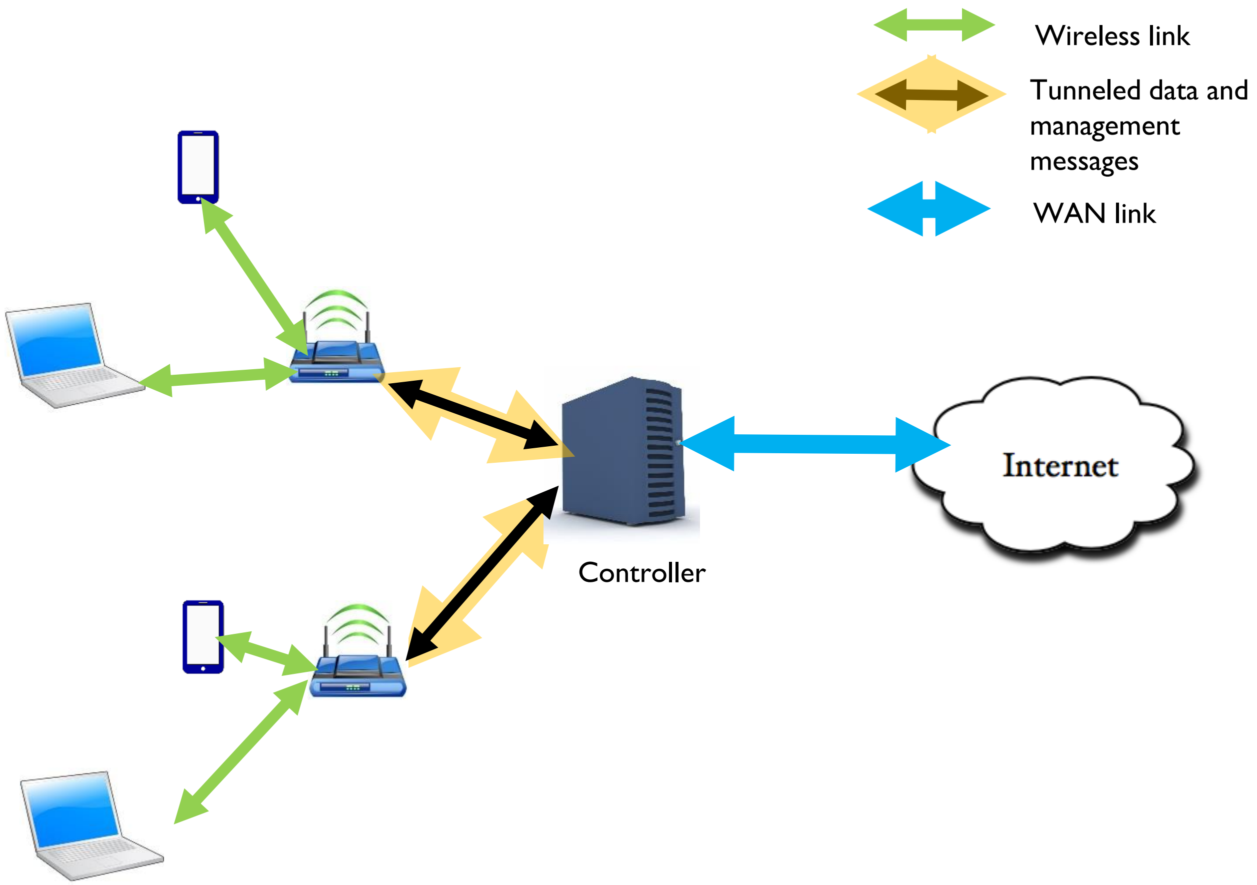
Wi-Fi for additional capacity and coverage

- Unlicensed band
- Offload cellular traffic
- Rural Broadband access network
- Large scale deployment

Management of large scale deployment

- Centralized control
- Flexibility in configuration and policy handling
- Interoperability

Central Management of Large Scale WLAN



Wireless link

Tunneled data and management messages

WAN link

IITB Wireless, IITB Guest

- Features: Roaming, multiple networks , access control etc.

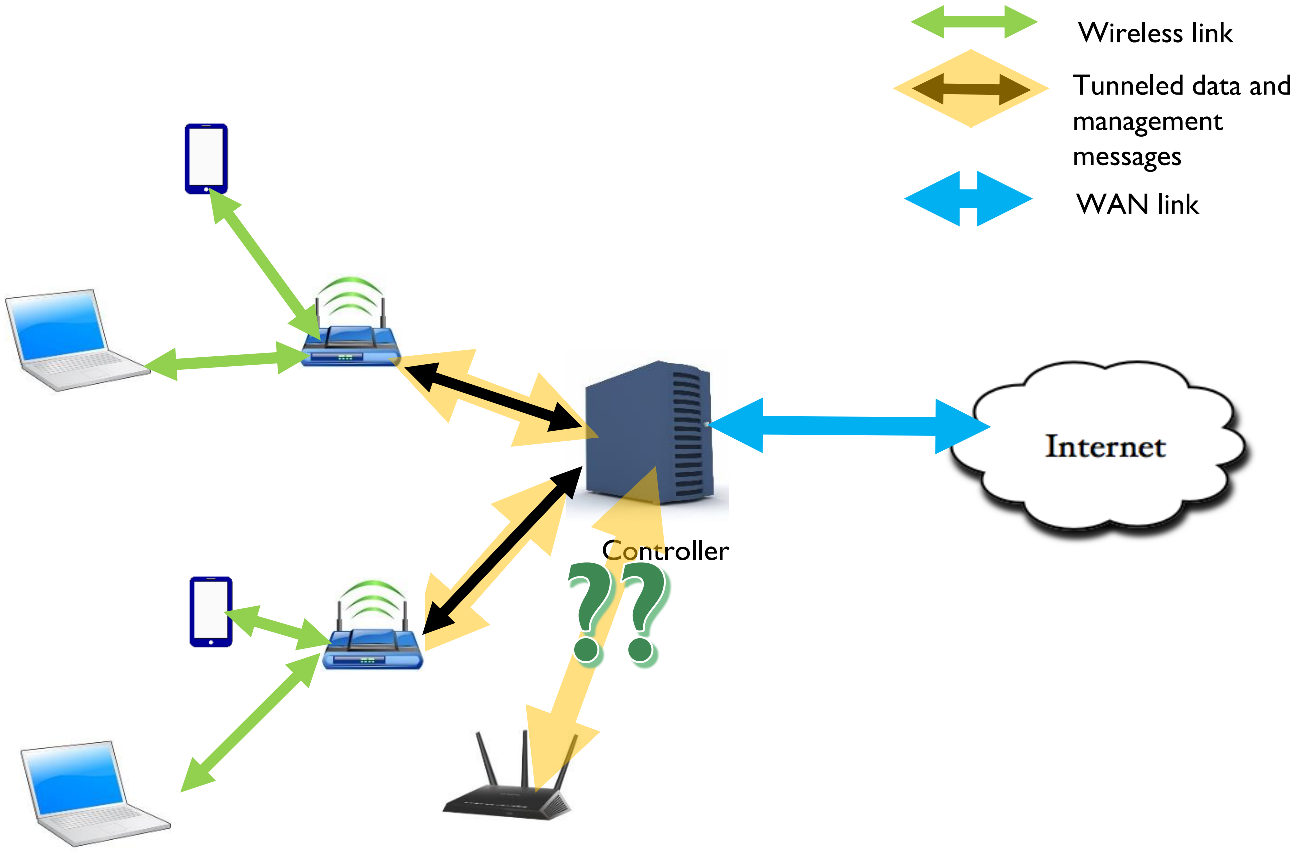
Standardized or Proprietary?

- Partially Standard, mostly proprietary and closed implementation based on that
- CAPWAP or TR069 based
- Expensive solutions.

Is the solution scalable?

- Why is controller loaded with user data?

Central Management of Large Scale WLAN



Can same controller manage different vendor APs?

- No. They cant understand each other.

How to address these issues?

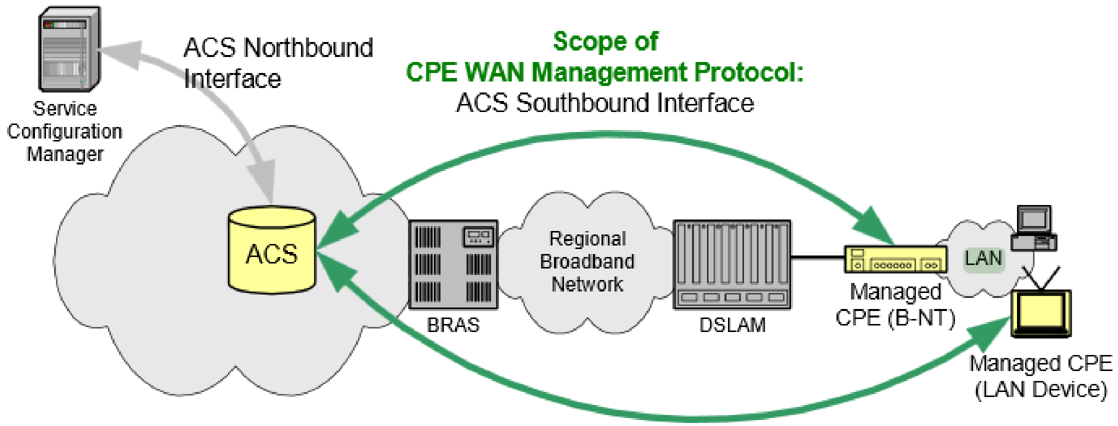
- All feature implementation must be **standardized**
- Can we do better? YES.

  SDN can make things better

Existing Standards: CAPWAP and TR069

CAPWAP - Control And Provisioning of Wireless Termination Points

- By IETF (Internet Engineering Task Force)
- UDP based protocol - RFC 5415
- Bindings are written for 802.11 WLAN Networks RFC 5416



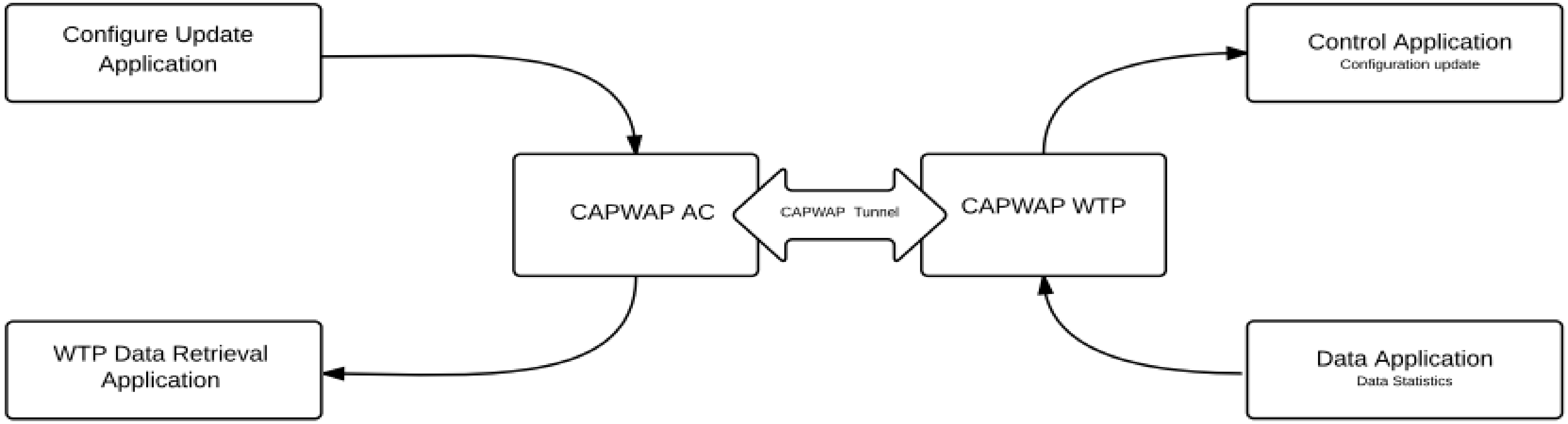
Typical Positioning of ACS. (courtesy [1])

TR069 – Technical Report 69; CWMP – CPE WAN Management Protocol

- By Broadband Forum
- Protocol to configure CPE (Customer Premise Equipment) from remote ACS (Auto Configuration Server)
- http/SOAP based protocol for configuration

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

CAPWAP based Controller



Architecture of CAPWAP based WLAN controller

A CAPWAP tunnel is setup between AC and WTP

Applications are written on these main threads for CAPWAP 802.11 bindings

- An application to set configurations
- An applications to retrieve statistics

Setting up CAPWAP Tunnel

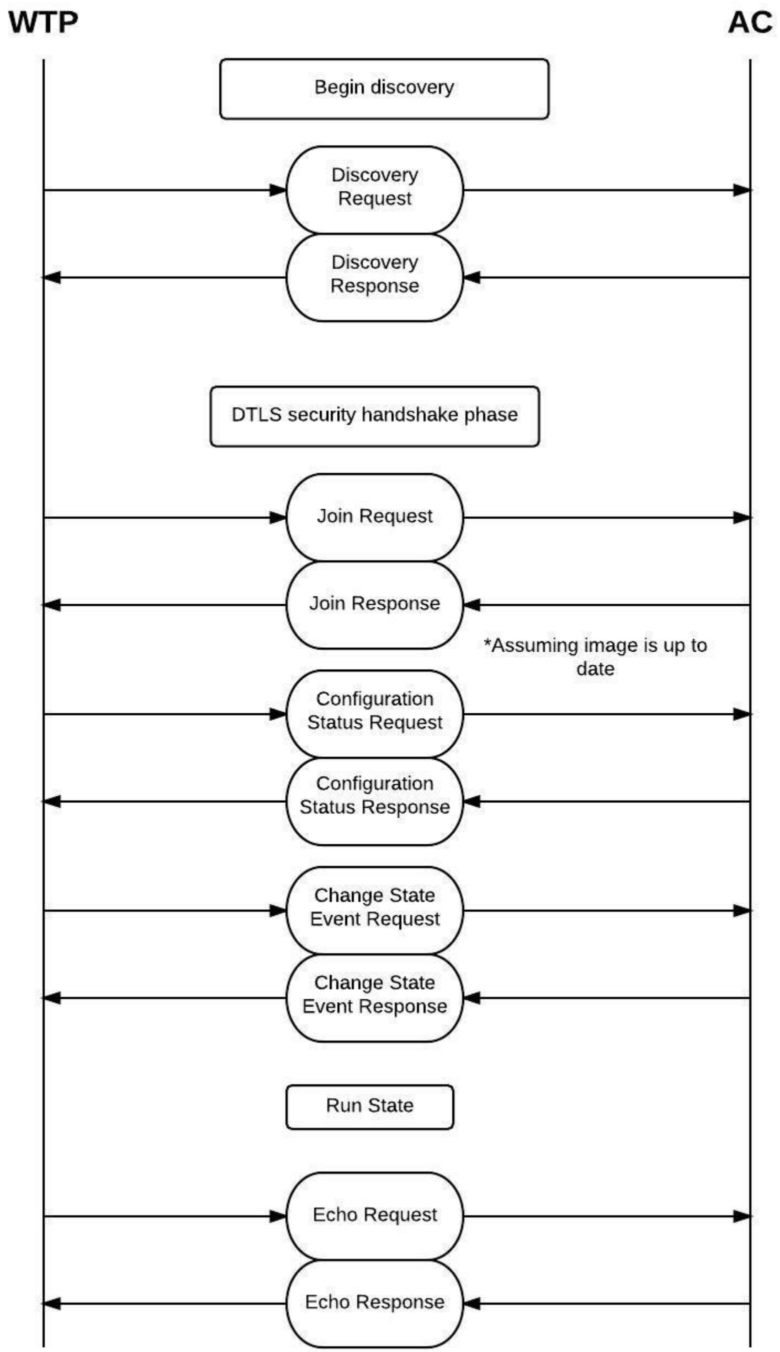
The tunnel setup is based on OpenCAPWAP implementation by M.Bernaschi et.al. , IAC-CNR Rome, Italy[2]

UDP based tunnel with a retransmission technique

OpenSSL implementation of DTLS protocol

OpenWRT ported WTPs

- To ensure interoperability i.e. manage different vendor WTPs
- Setting of configurations done through UCI
- Used net link library to get statistics



WTP
AC
Begin discovery
Discovery Request
Discovery Response
DTLS security handshake phase
Join Request
Join Response
*Assuming image is up to date
Configuration Status Request
Configuration Status Response
Change State Event Request
Change State Event Response
Run State
Echo Request
Echo Response

[2] 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

IEEE 802.11 Bindings for CAPWAP

802.11 bindings specified in RFC 5416[3]

- At configuration update application
  - Change channel of operation in 2.4GHz band: IEEE 802.11 OFDM control (Bi Directional)

Byte 0	Byte 1	Byte 2	Byte 3
Radio ID	Reserved	Channel	Band support
TI Threshold			

  - Change Tx power : IEEE 802.11 TX Power

Byte 0	Byte 1	Byte 2	Byte 3
Radio ID	Reserved	TX Power (in mW)	

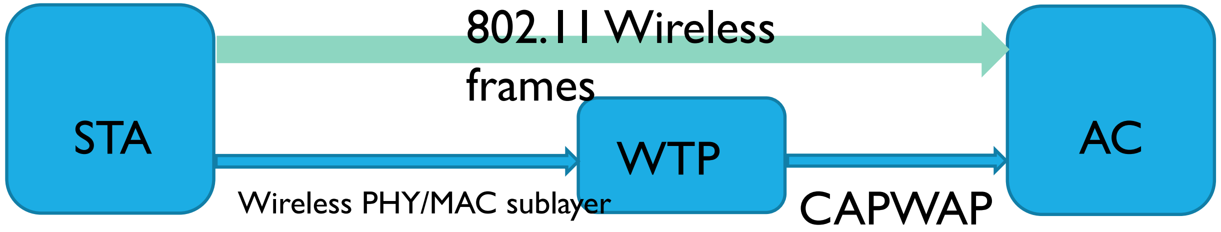
- At Statistics retrieval application
  - Station Dump/ STA statistics: IEEE 802.11 statistics
- More bindings for more features
  - Architecture needs only addition of message structures at applications. CAPWAP tunnel remains intact

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

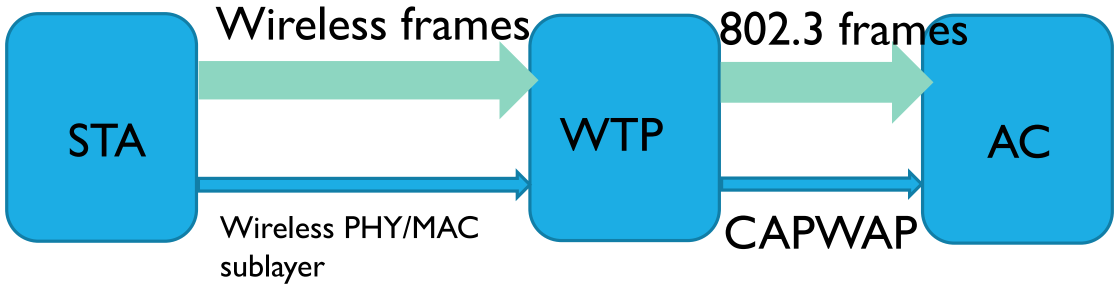
Issues with Existing Standards

Local bridging of Data in CAPWAP

- Split MAC
  - Control, beacon and probe frames processed locally
  - All data and other management frames forwarded to AC



- Local MAC
  - All frames processed locally
  - Data and management frames forwarded as 802.3 frames



Issues with Existing Standards

Why should data be forwarded to AC?

- Not scalable

Should the AP/WTP really process all the wireless frames? Can we do better?

- Makes AP heavy

No support for IEEE802.11 r, k, u or any recent amendments

- CAPWAP doesn’t specify a technique for roaming/handover
- Key caching technique is used
- Pre-authentication not supported

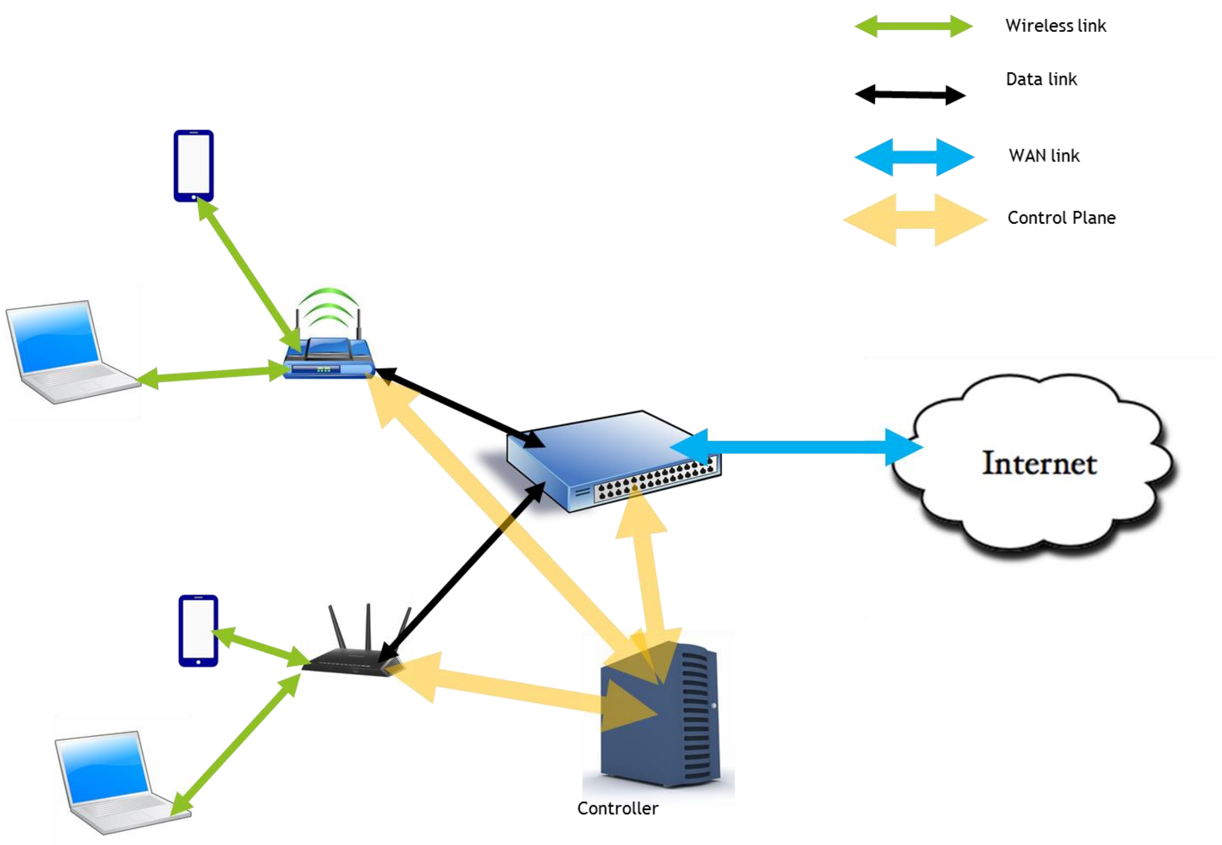
Interoperability issues

- No standard interfaces defined

Multiple Networks on same AP and access control

- Not featured

SDN for WLAN Management



What is SDN? How does it help

- SDN: Software Defined Networking. Enables dynamic programming of network
- Separates Control plane and data plane
- Provides standard interfaces or APIs for features. Implementation may differ below this level

Is it really needed?

- Yes. It makes APs light radios that forward data
- Gives global view, uniform policy management
- Enables interoperability by providing standard interface.

Is the solution scalable?

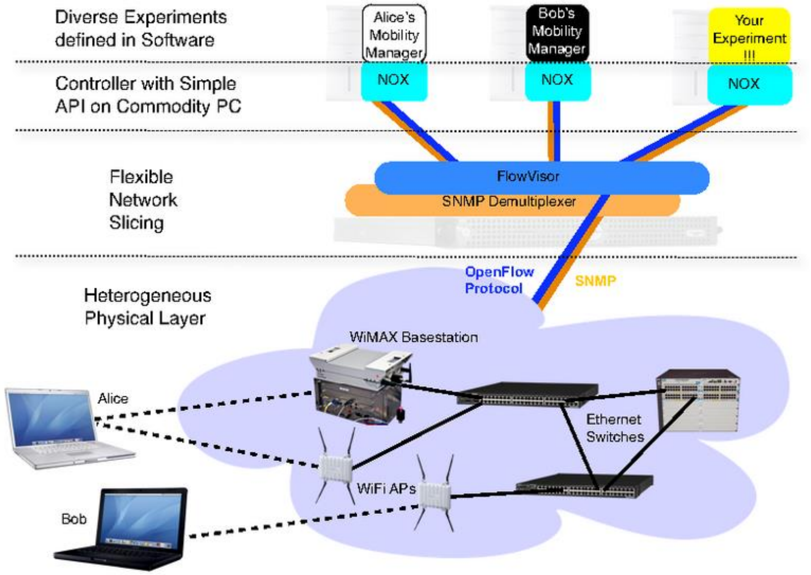
- Yes. As the controller is not loaded with data.

Existing work?

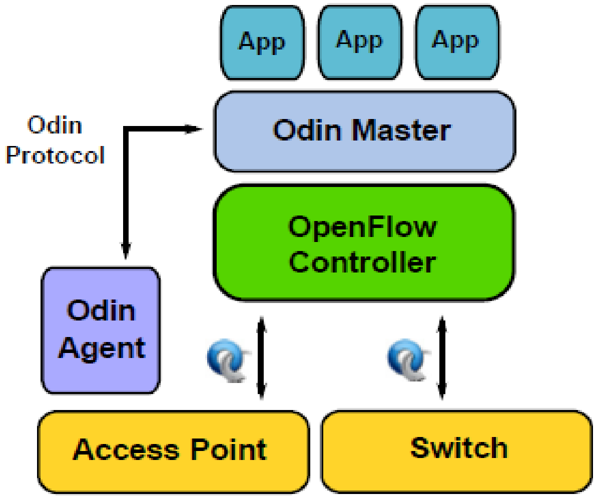
- Odin, OpenFlow wireless or OpenRoads, ethanol etc.

Various SDN Controllers for WLAN

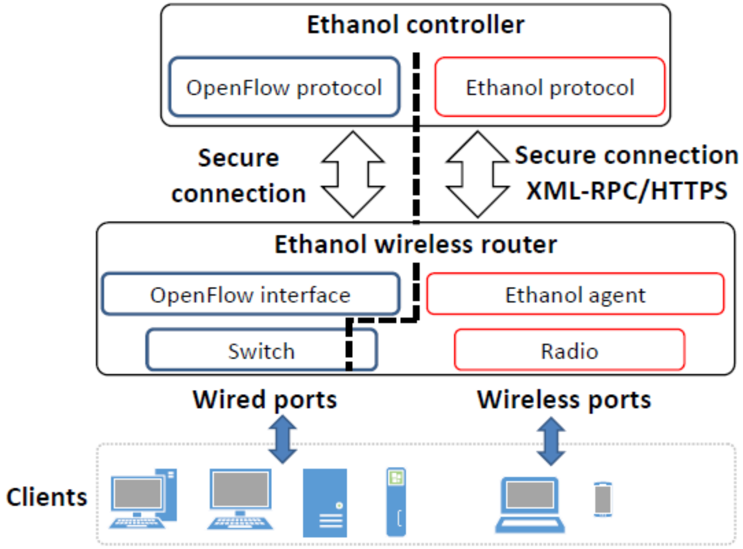
Architecture of OpenRoads, Odin and Ethanol



OpenRoads Architecture (Figure Courtesy [4])



Odin Architecture (Figure Courtesy [5])



Ethanol Architecture (Figure Courtesy [6])

[4] Kok-Kiong Yap et al. ”OpenRoads: empowering research in mobile networks”, SIGCOMM Comput. Commun. Rev. 40, 1 (January 2010), 125-126, 2010.

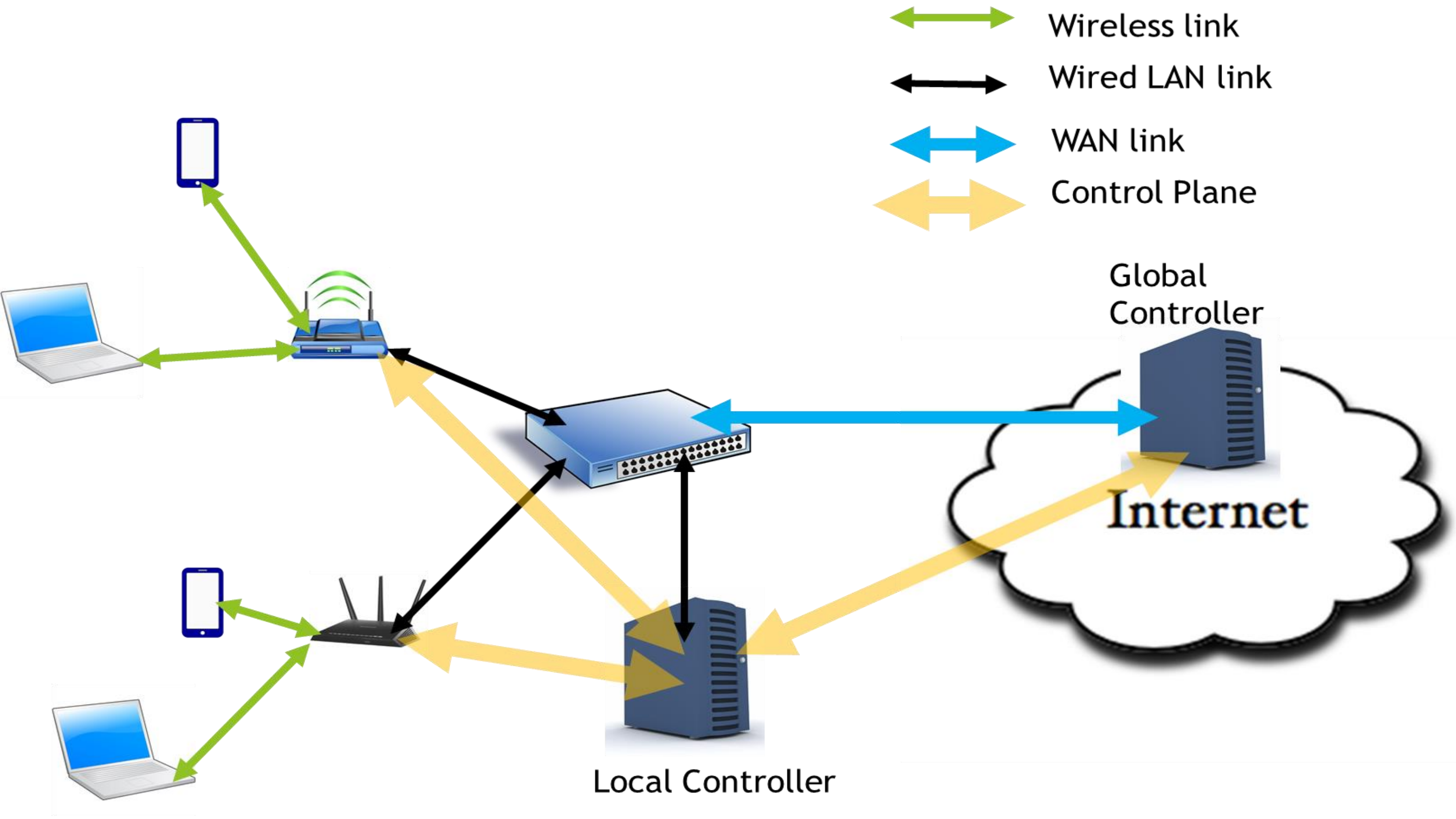
[5] Lalith Suresh et al. ”Towards programmable enterprise WLANS with Odin”, (HotSDN ’12). ACM, New York, NY, USA, 115-120, 2012.

[6] Moura, H et al., ”Ethanol: Software defined networking for 802.11 Wireless Networks”, Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium, 2015.

Comparison of Existing Controllers and Standards

Architecture	Load balancing and Mobility management	QOS management	Virtualization and slicing	Security and communication
Open Roads	Implemented test algorithms	Not mentioned	Network slicing using Flow Visor and SNMP Visor	Not mentioned
Odin	Done using split MAC and LVAP. Very smooth handovers.	Not mentioned: Mentioned as a drawback of Odin in Ethanol paper	LVAP: Access point virtualized per client	Security as in openflow . Authentication done through a AAA server
Ethanol	Done with 802.11 k, f and r (Every feature here is done according to a 80211 binding)	Like in Pantou: Using HTB	Virtual Aps and a flow entity based on openflow	https connection (XML-RPC). Improved security at controller : Localization
CAPWAP	Possible. As every data frame goes to controller.	Possible	Not possible at AP level. Not a feature	Secured DTLS tunnel
TR - 069	Not a feature mentioned (Possibility questioned as entire control framework lies outside network)	Not mentioned	Not mentioned (Needs changes at AP)	https connection for control. Authentication is through a different communication (Not through CWMP protocol)

Proposed Architecture for SDN Controller



Hierarchical Controller Architecture

- ○ Time critical operations in local controller
- ○ Global policy management in cloud controller

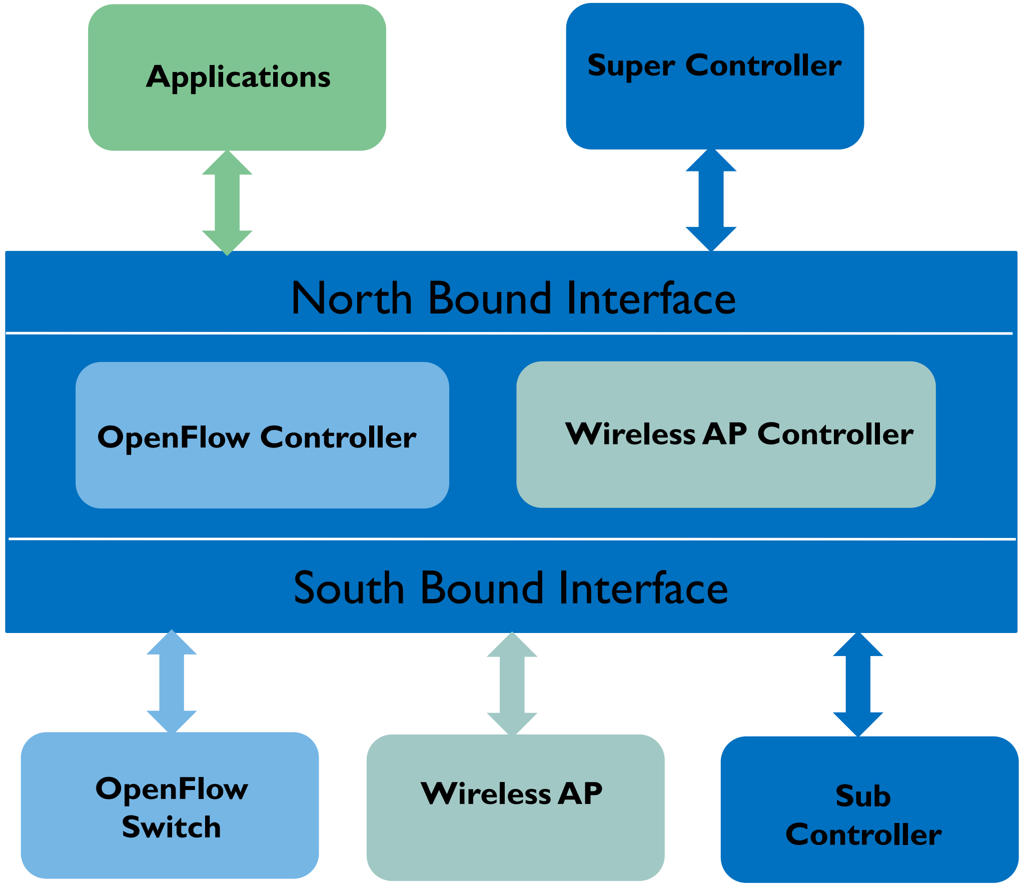
Access technology independent protocol

- ○ Manage Wi-Fi , Wi-Max deployments using same controller
- ○ Same controller to manage TVWS backhaul and also WLAN deployment

Standard Interfaces

- ○ Enables interoperability

Proposed Architecture for SDN Controller



Controller and switches: **OpenFlow** protocol

Controller and APs: TCP based protocol (Proposed for standardization)

- Wireless technology independent protocol
- Bindings written to support specific wireless technology

Controller to controllers: Openflow forwarded by flow visor

Current and Future Work

Exploring OpenMUL[7], a SDN controller

- Open source software written in C

Pantou[8] on an AP

- An openflow application over OpenWRT

Working on Protocol stack for management of WLAN with TSDSI

- Detailed gap analysis of existing standards
- Explore more uses cases

Implement a Controller as per the architecture proposed

- Write applications for Load balancing, roaming etc. and test performance.

[7] http://www.openmul.org/

[8] http://archive.openflow.org/wk/index.php/Pantou_:_OpenFlow_1.0_for_OpenWRT

THANK YOU