Wireless Software Defined Networks

Ayaka Koshibe, Akash Baid and Ivan Seskar

WINLAB

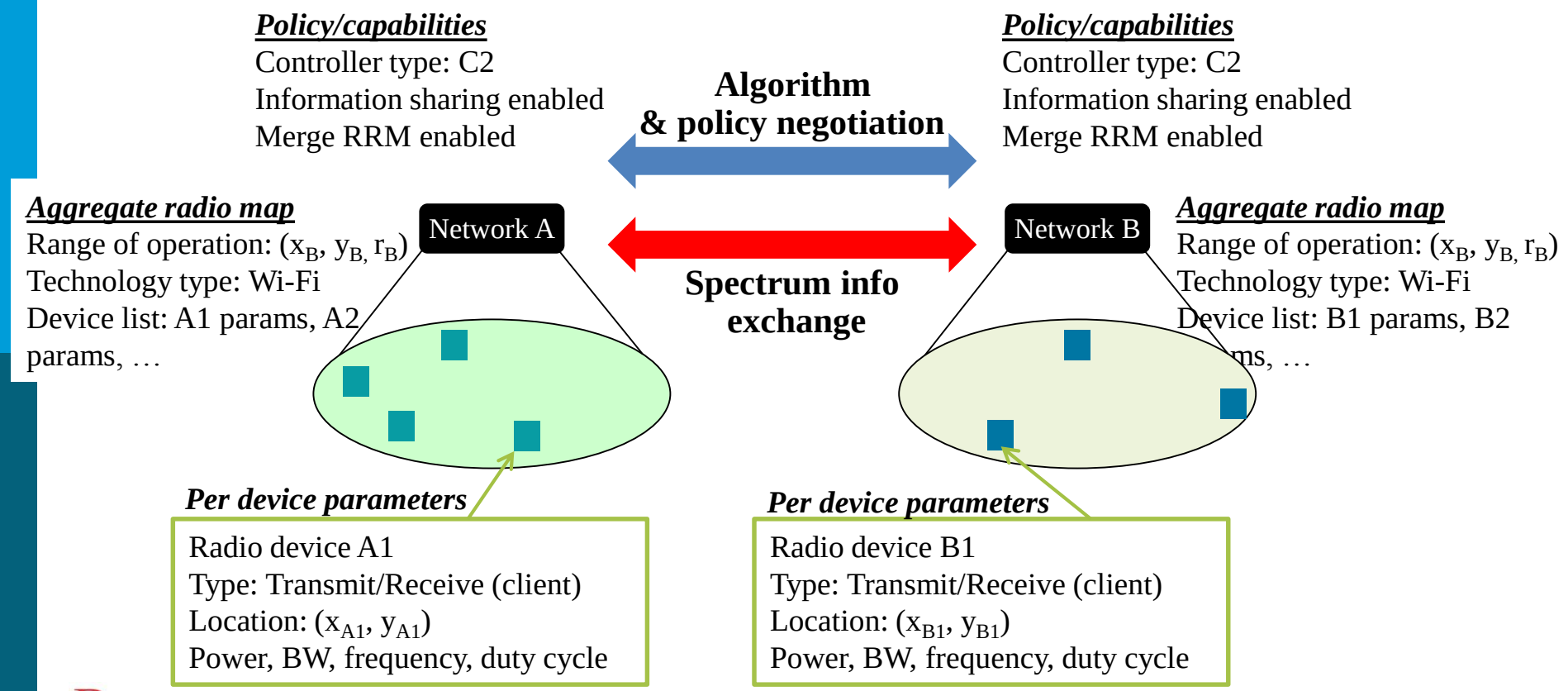
Rutgers University

2014 Fall IAB

December 12th, 2014

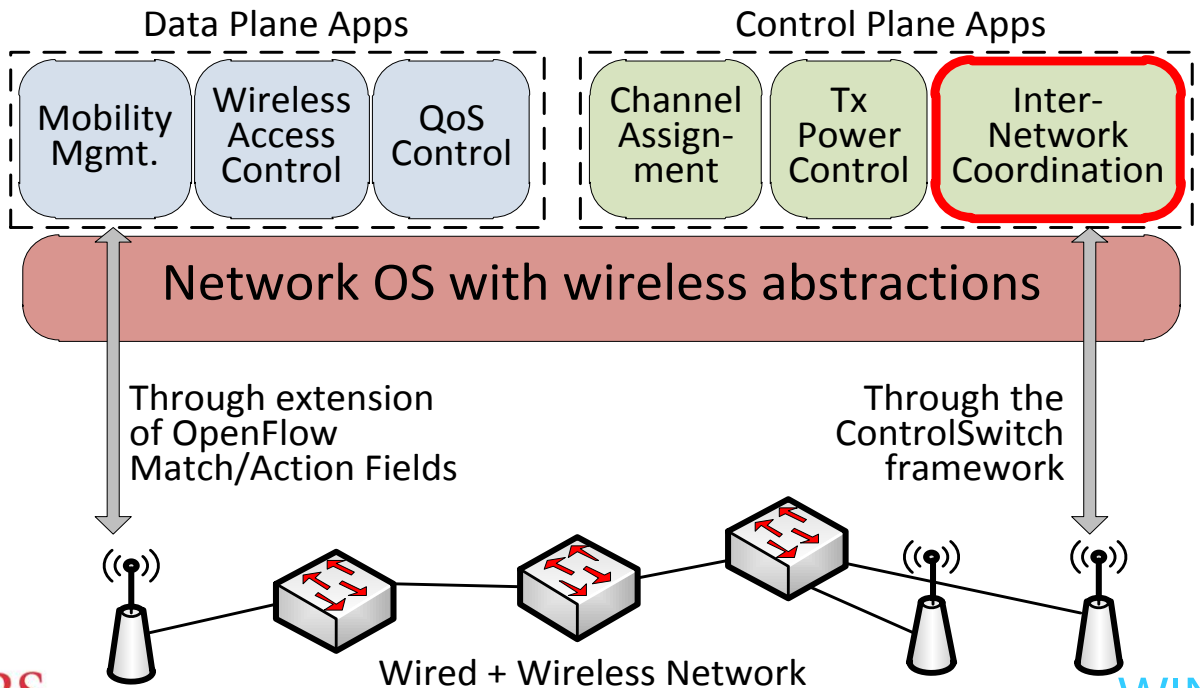
Need for Inter-network Cooperation

- Interaction between managed wireless networks over the back-end wired link for making more efficient use of the spectrum



SDN Approach to Wireless Control Plane

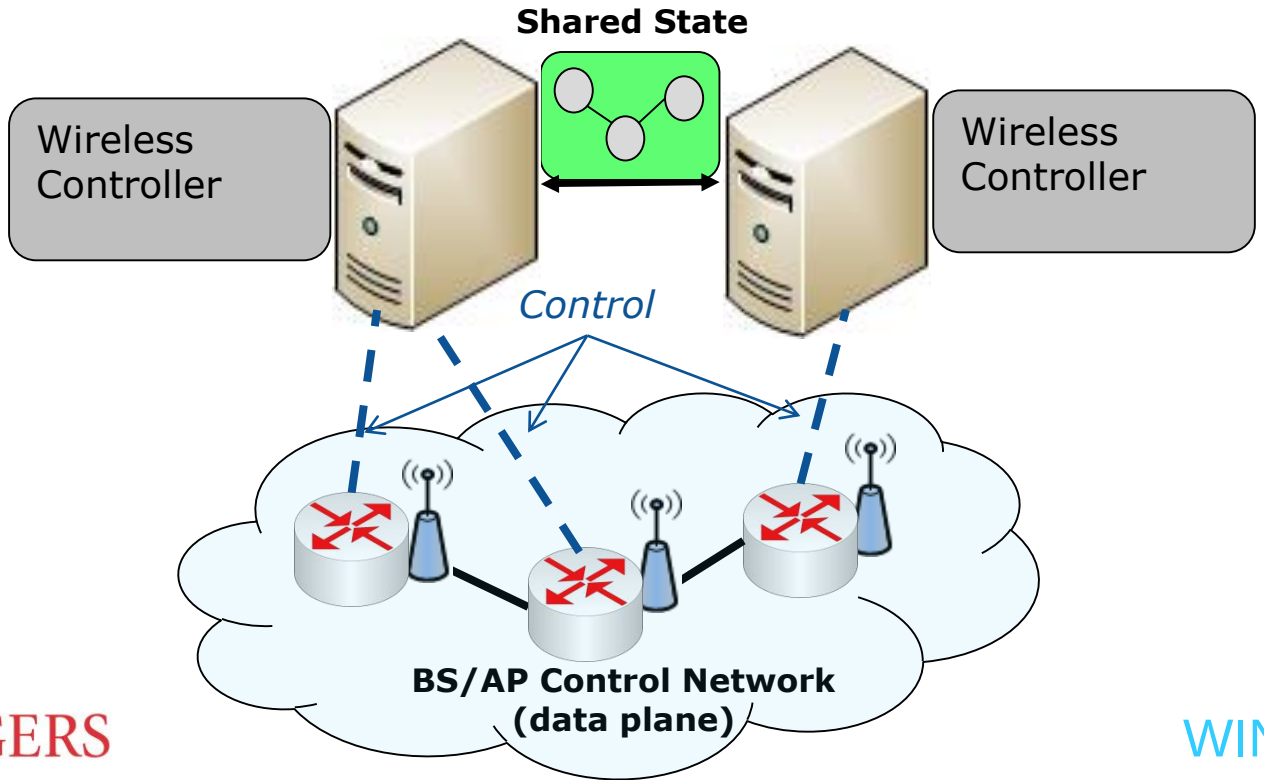
- Introducing flexibility in the wireless control plane by leveraging software defined networking techniques
- Inter-network cooperation translates to inter-controller interactions and setting of flow-rules



A: Distributed Control Plane

Extension of traditional Enterprise Controller:

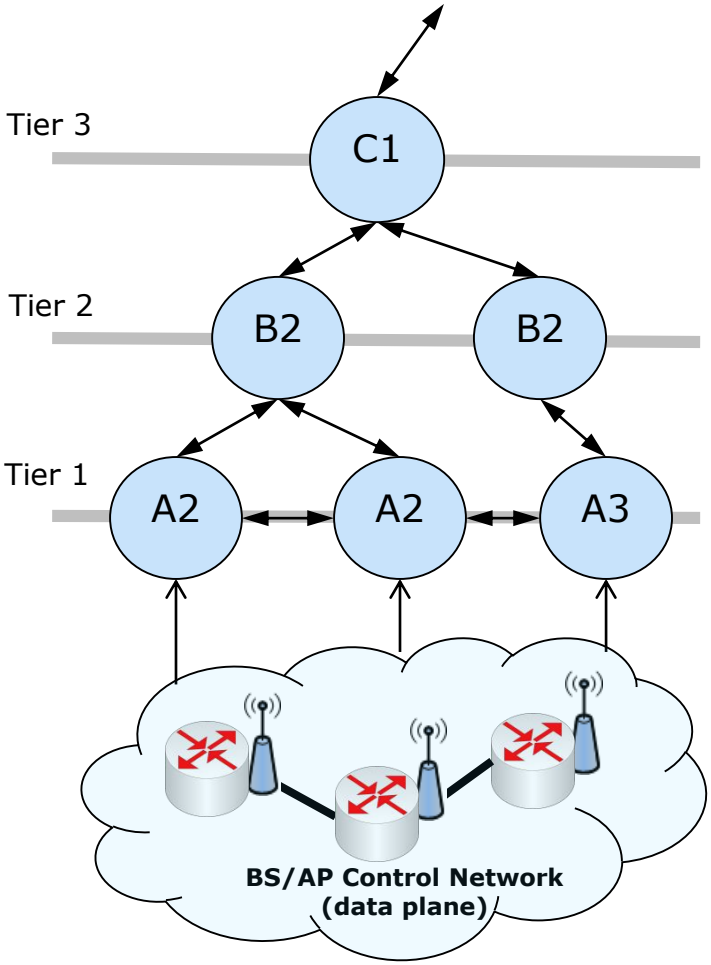
- Multiple copies of wireless controllers (WC) with mechanisms to cooperate, scattered throughout SDN based control plane
- Reduced distance between device and a controller – reduced flow setup times (reduced control latency)



B: Heterogeneous Distributed Control Plane

- Builds the control plane as a network of different controllers
  - Each controller is a part of a control stack
  - Controllers communicate by message passing
  - Multiple controllers process each event
    - SDN control plane can pick and choose services for each event, avoiding conflict
- Wireless Control Stack (WCS) realized as a complex interaction between controllers rather than (single) monolithic application

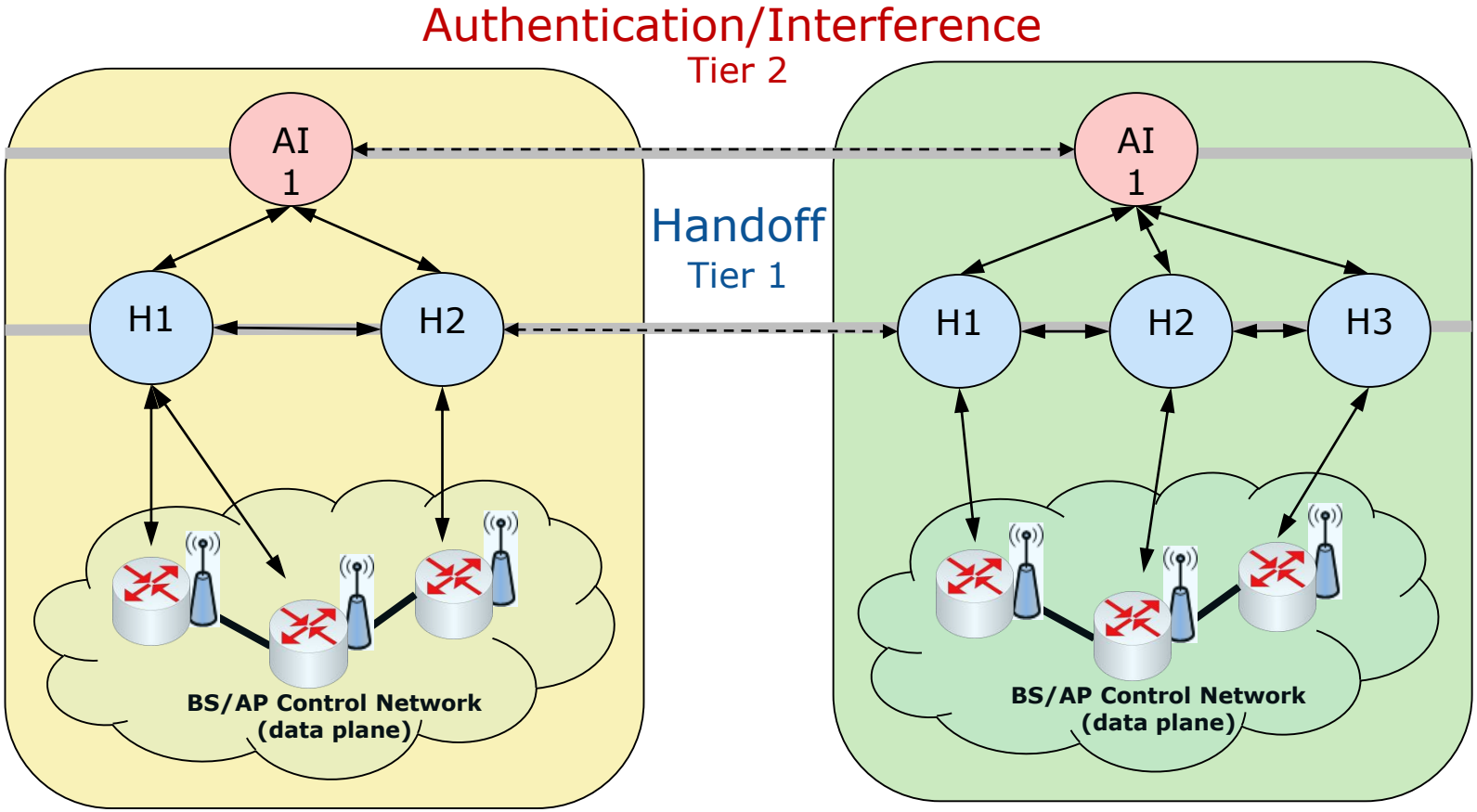
C: Heterogeneous Hierarchical Control Plane



- **Groups of functional WCS controllers arranged in tiers**
  - Higher tier events = more global and less frequent
  - Supports natural hierarchical inter-controller relations
- **Controllers connected with SDN inter-controller links**
  - Joins pieces of wireless control stacks together through SDN
  - Additional benefit: dynamic provisioning and routing for events to be moved between controllers and/or tiers (if they can't be handled at a tier)

Heterogeneous Hierarchical Control Plane

- Example: Pair of enterprises with heterogeneous decomposed controllers



Distributed SD(W)Ns: What's Missing?

- Scalable and resilient, but not readily available
  - proprietary, experimental, or early-stage
- Focuses on uniformity, expects controllers to be homogeneous
- No standard mechanism to handle interaction of network stacks (SDX?)
  - Relies on specialized mechanisms
    - network hypervisors - virtualize network
    - network compilers - resolve flow conflicts
    - semi-intelligent switches - delegate some work to switches
    - SDN-to-IP network peering applications

Implications

A complex control plane structure requires:

- Orchestration of event handling across multiple controllers
  → An inter-controller process chain
- Path selection across control plane
  → A control plane routing mechanism
- Allow controllers to learn of available services/topology
  → An information propagation scheme

Enabling Coordination

Servers configure **client process chains** via subscription messages containing:

- SID and events handled by server
- Packet process chain behavior directives for the client
  - Services may want clients to behave in specific ways → A client must wait for an ACL's result for an event before further handling
  - Four types: DENY, ALLOW, DIVERT, SPLIT
- Service reachability information
  - Route to service by hop count
  - Hops increased as messages propagate towards data plane
  - Allow picking of smallest hop routes to each service

Controller Architecture - Layered Model

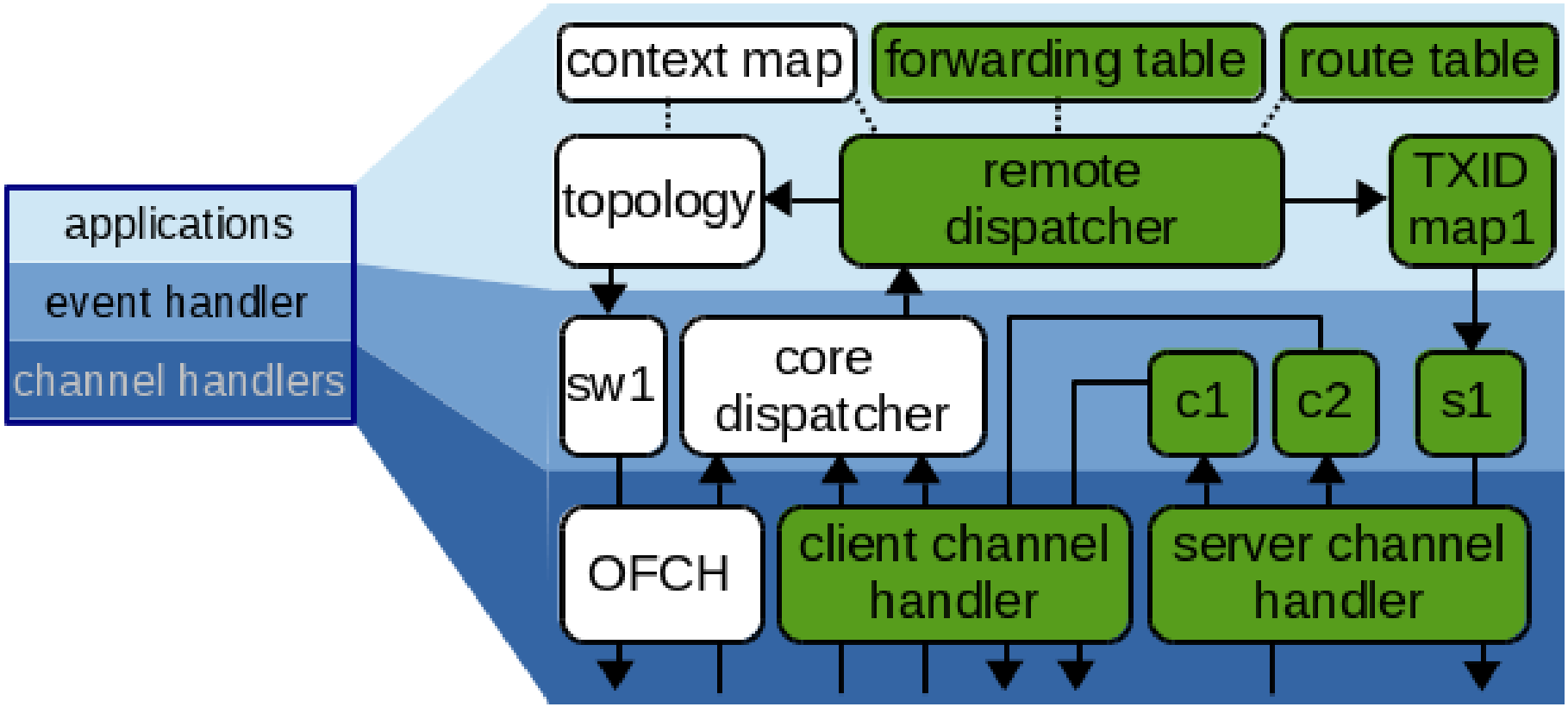
How is a controller implemented?

A controller is composed of three layers:

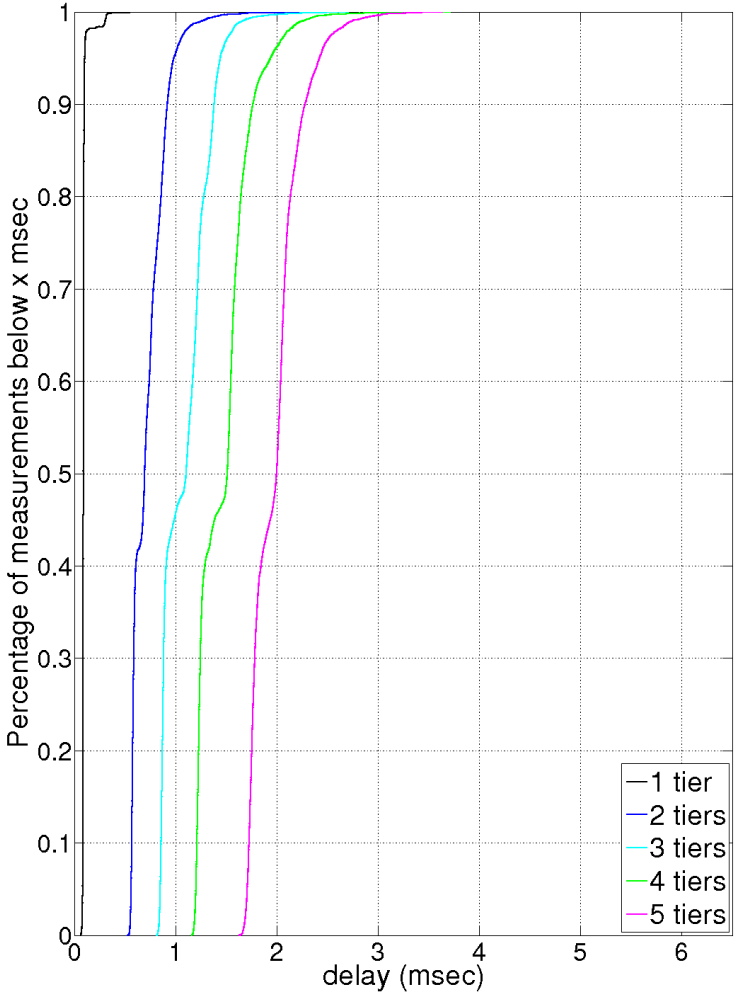
- Control channel handler: Sending/receiving control messages from the data plane
- Event dispatcher: Conversion between control messages and events, distribution of events to services
- Applications: Implementation of services, event handlers and interfaces for usability

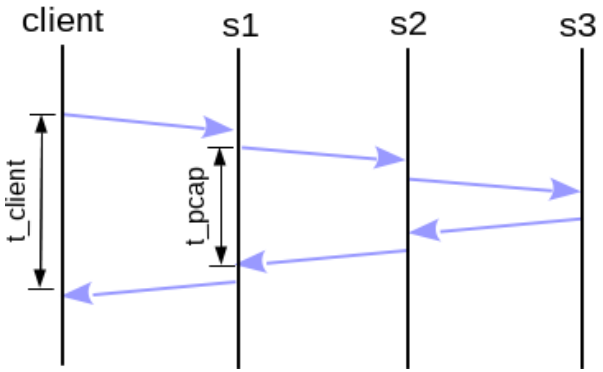
Implementation: A Hierarchical Controller With Floodlight

- Base controller extended with modules, channel handlers, and dispatchers



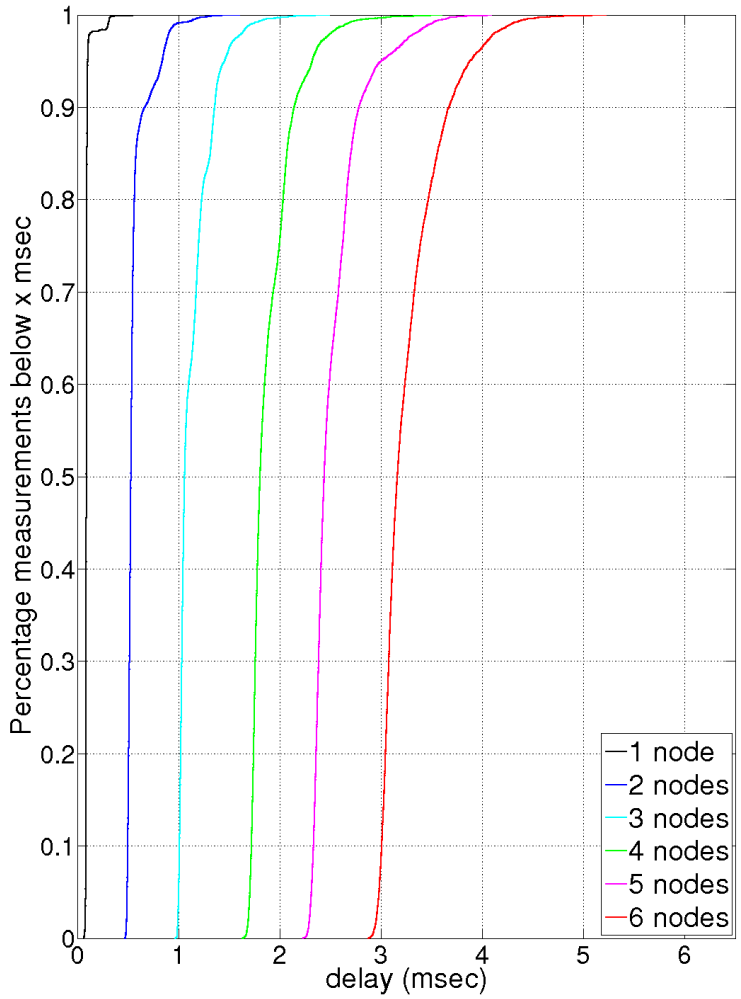
Processing times, multiple tiers

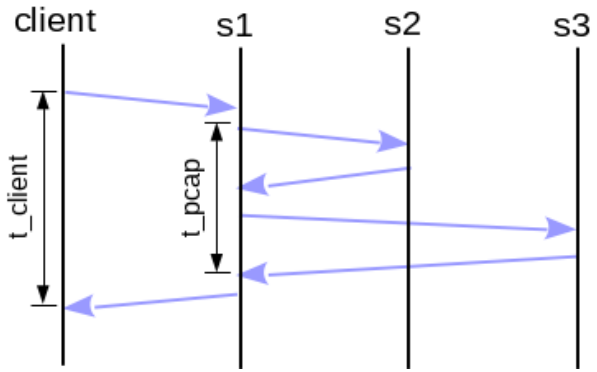




hops	mean(msec)
1	0.09
2	0.72
3	1.09
4	1.47
5	1.98

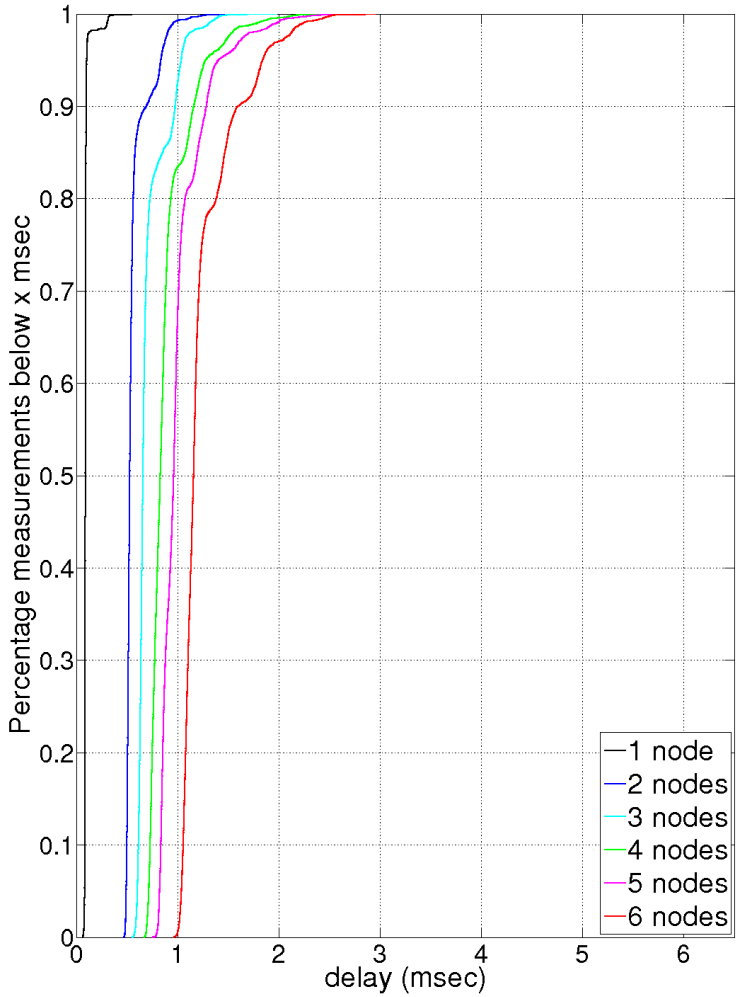
Processing times, multiple DIVERT

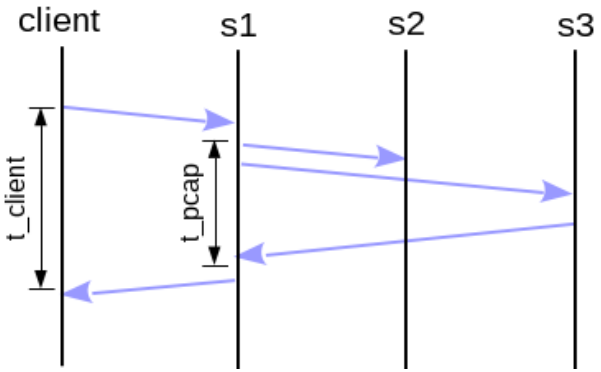




hops	mean(msec)
1	0.56
2	1.13
3	1.89
4	2.52
5	3.27

Processing times, multiple SPLIT





hops	mean(msec)
1	0.56
2	0.71
3	0.89
4	1.01
5	1.25

Scalability Testing - Hierarchical

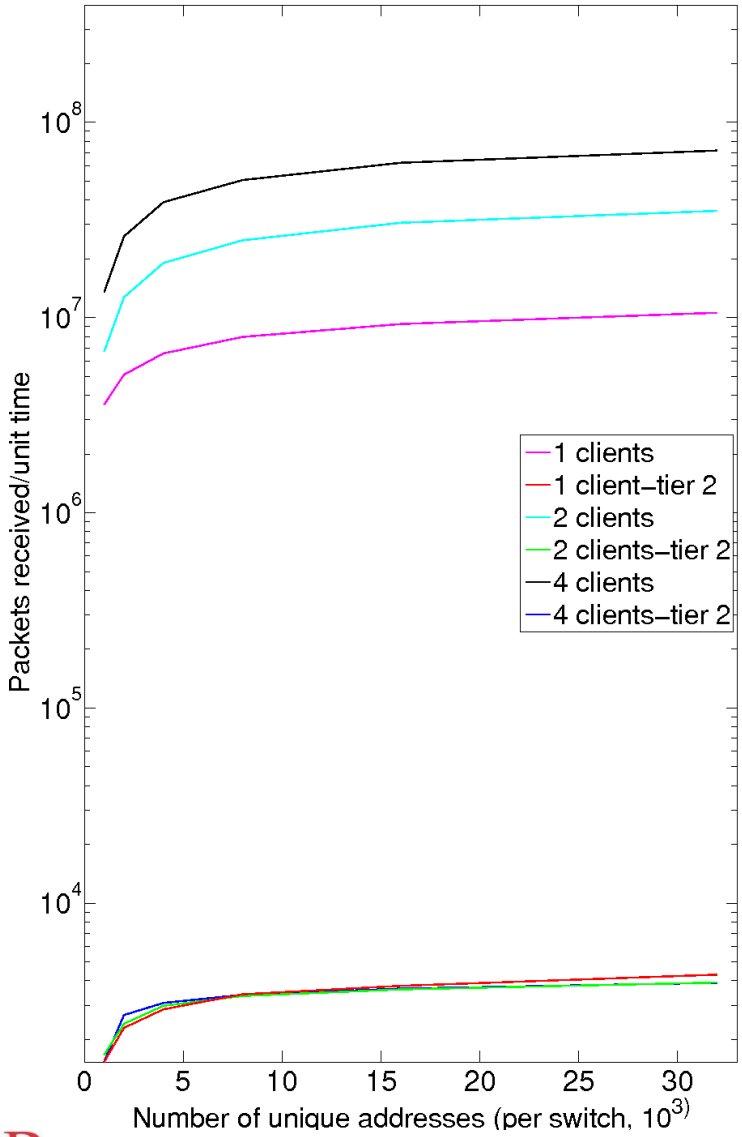
Two-tiered control plane

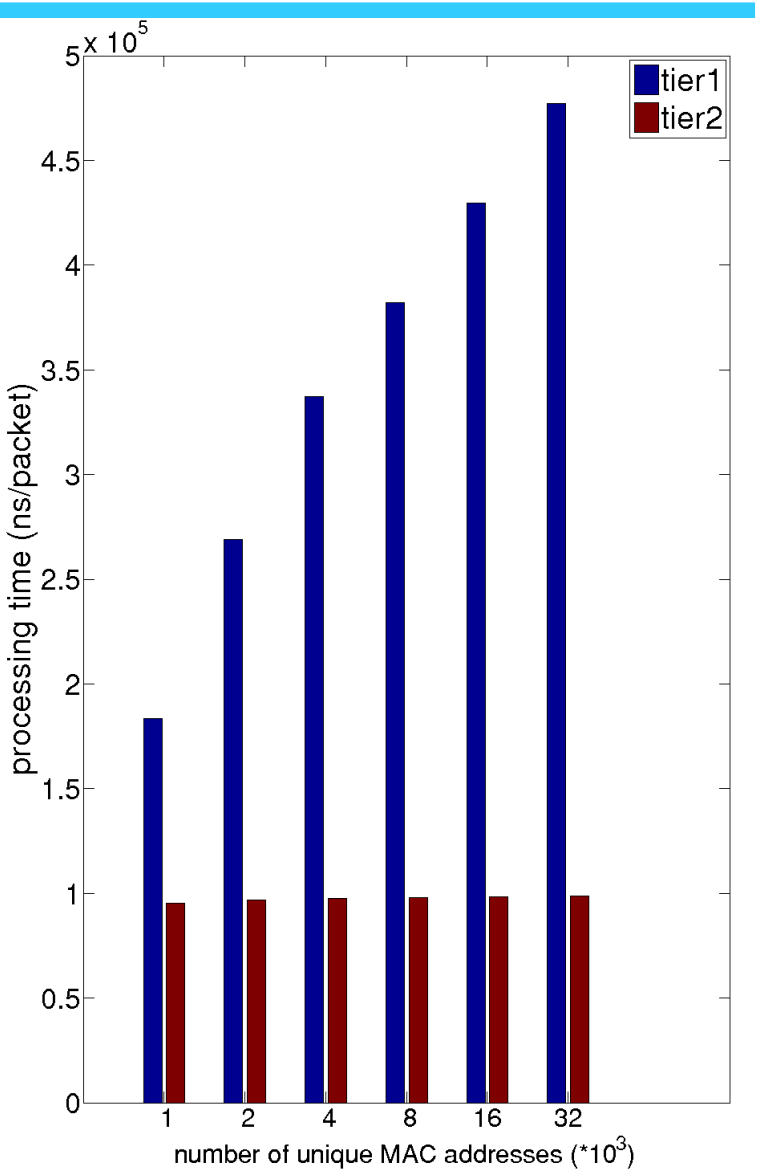
- One, two, and four tier 1 controllers
- Fixed at four virtual datapaths/controller
- Tier 2 controller subscribed to new-device event triggers

We expect:

- Tier 2 subscriptions cut event escalations to manageable levels
- Event handling capacity of Tier 1 scales with the number of controllers

Event reception/handling





OPEN BTS

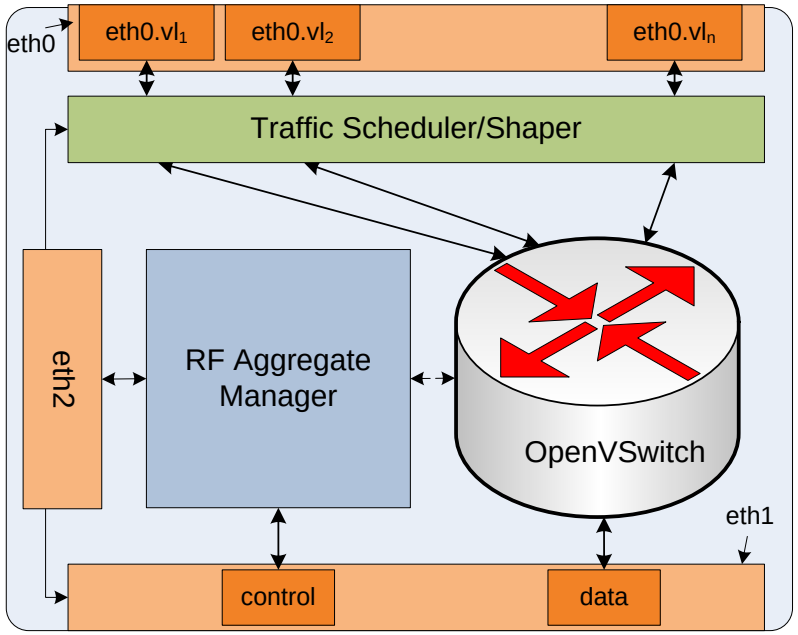
GENI Project – Open WiMAX BTS

- Exposed all controllable parameters through API
- Removed all default IP routing, simplified ASN controller*
- All switching purely based on MAC addresses
- Implemented the datapath virtualization and *VNTS* shaping mechanism in click/OpenVSwtich for slice isolation

Ongoing work – Open LTE BTS

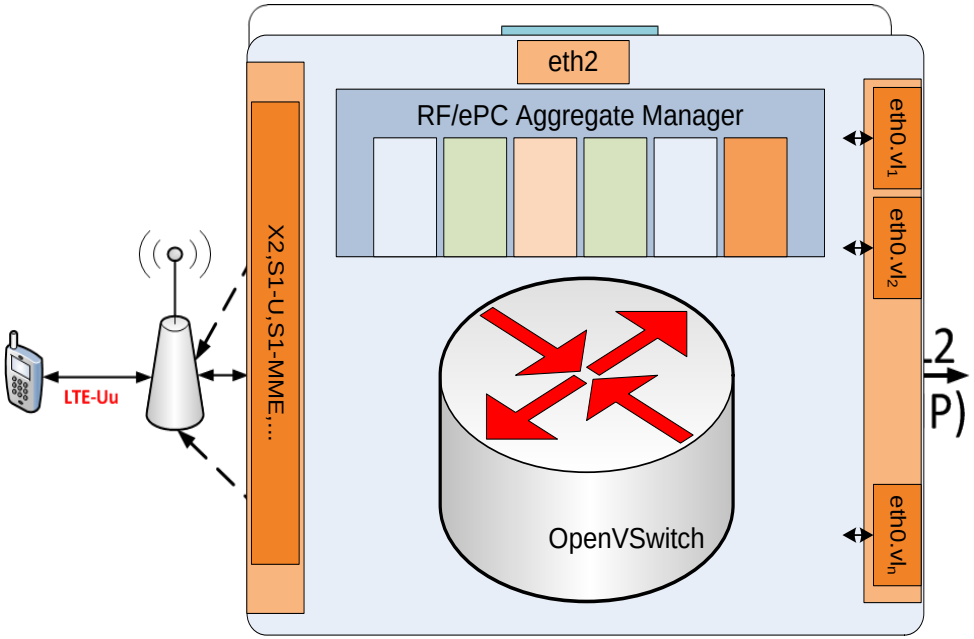
- Exposed all controllable parameters through the same REST based API
- Implemented the datapath with openvswitch
- *Currentl development: ePC replacement with open source aggregate manager (i.e. simplification/elimination of LTE control protocols)*

OPEN BTS: WiMAX & LTE

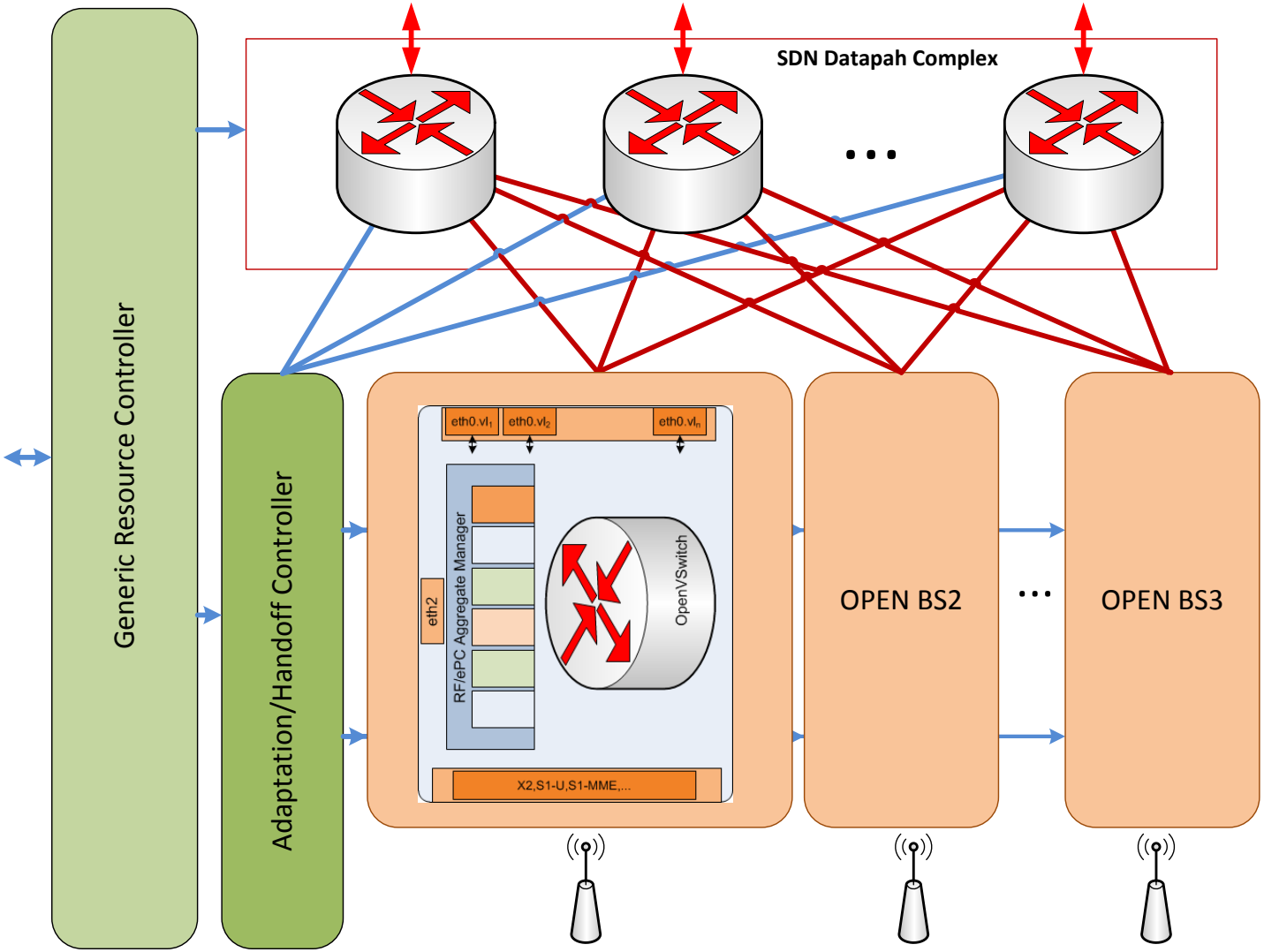


WiMAX

LTE



4G in GENI: Larger Picture



More Info @

www.orbit-lab.org

www.geni.net

www.winlab.rutgers.edu

wimax.orbit-lab.org