OpenFlow based Traffic Engineering for Mobile Devices

© Radhika Hirannaiah, PhD

Open vSwitch Fall Conference 2014

Current Challenges

Dynamic management of network connectivity

1. **Service Provider**
   - Limited flexibility to manage network connectivity on a mobile device
   - Network congestion
   - Monetization of peer-to-peer traffic
   - Increase bandwidth utilization

2. **Enterprise**
   - Management of data sharing and security on mobile devices by corporates for all its users

3. **Users**
   - Constrained network selection for applications due to device restrictions

Solution Concept

Traffic engineering from core to access device

1. **Service Provider**
   - Enable dynamic management with multiple control plane programmability
     - Deliver live stream video: Create a specific flow path from service provider to end device instead of using multicast

2. **Enterprise**
   - Ability to set security policy on devices
     - Sharing of corporate data: 2 employees are offsite and want to share a presentation. The device can be dynamically configured by IT to allow sharing, say, only over Bluetooth

3. **Users**
   - Select and resolve appropriate network connectivity
     - Today, one has to turn off say, WiFi, to make a VoIP call by using LTE. Applications are not provided flexibility to incorporate user input for network connectivity

Research

Proposed an architectural model to deploy cost effective Circuit Switching as an overlay over traditional Packet Switched IP network

1. Introduce OpenFlow based programmable switch to facilitate enhanced traffic engineering

2. Service providers, Enterprises and Users have the ability to program mobile data plane in a mutually exclusive way

3. Model flow movements across tables using finite state Markov chains

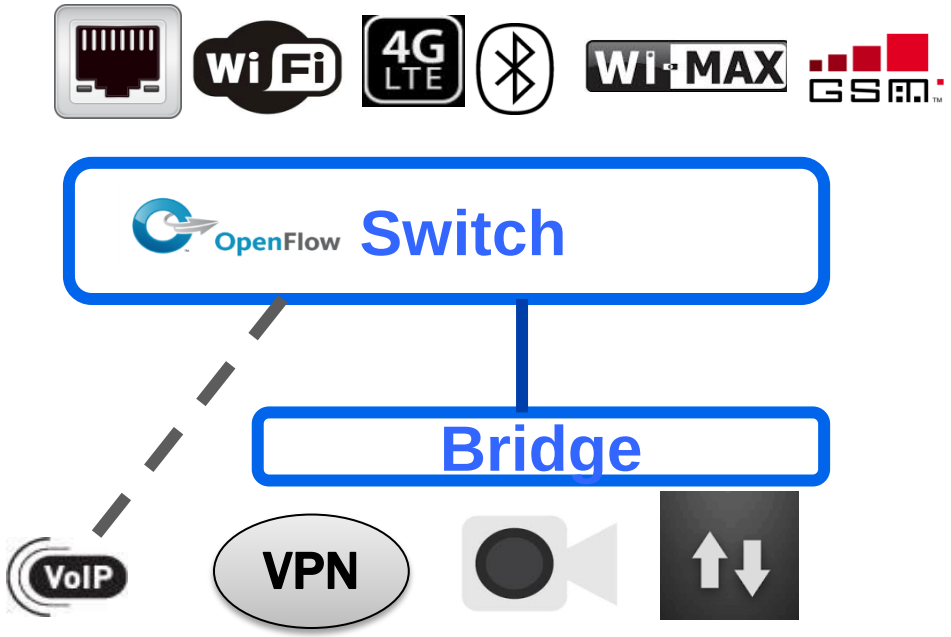
4. Model flow tables to controller assignments via a mathematical permutation algorithm

5. Validation of the solution concept though experimentation

Introduce OpenFlow based Programmable Switch

Improved traffic engineering with enhanced security and optimal use of device resources

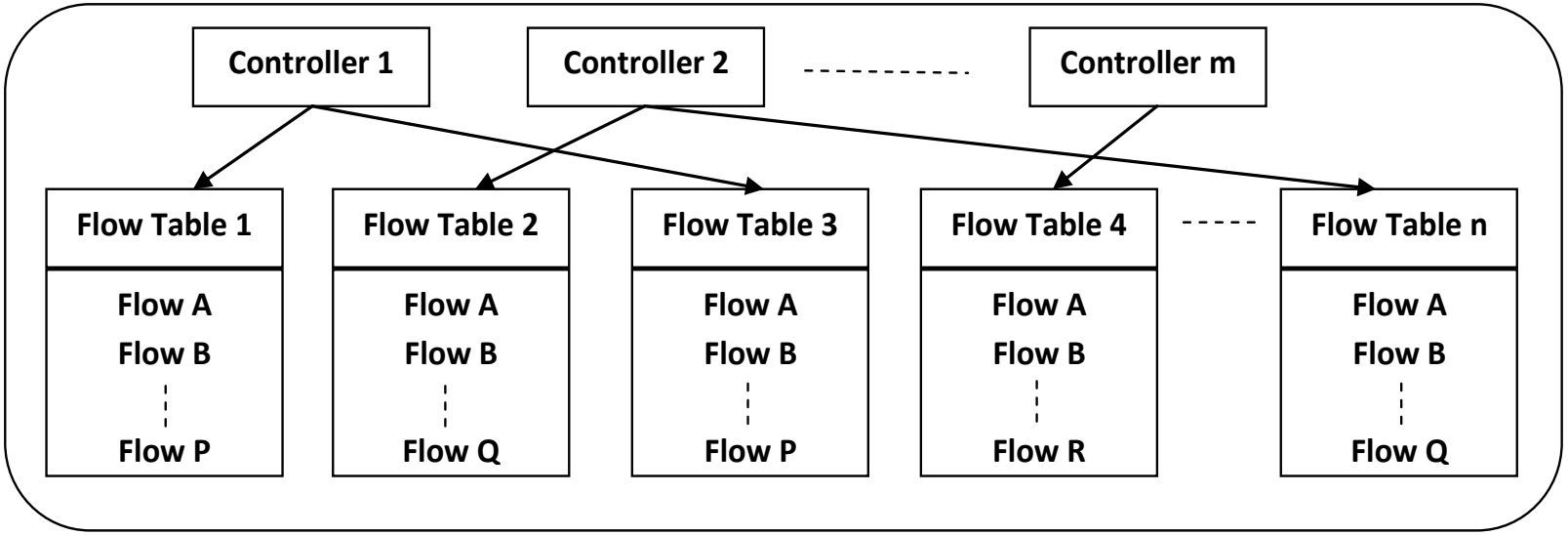
- Use of the network switch paradigm to hide the radios on the WAN side and applications on the port side
- Bridging the ports can enable existing applications to work seamlessly without any changes to the same
- Use OpenFlow based network switch to enable programmability



SP, Enterprises, Users are represented by Control Plane

Ability to program the User mobile data plane in a mutually exclusive way

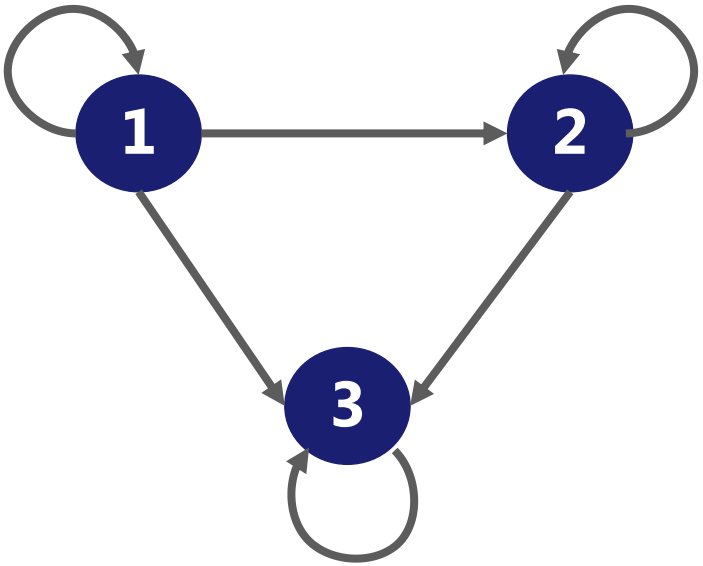
- Controllers run in “Equal” mode.
- Controllers manage flows on specific switch tables. Controller-switch table assignment is by out-of-band agreement and external to the system
- Standard forward direction packet processing pipeline
- Tables managed by a single controller are identical, except for the last one which has a different GoTo table entry



Application of Finite State Markov Chain

Flow movements across Flow tables are modeled as steady state transitions using concept of Markov chains

- Flow movement from one flow table to another is considered as a "transition" that is represented via a transition matrix
- The move from current state to next state is considered as the steady state and is evaluated using a conditional probability
- Simulated trials using R statistical software package



$$P = \begin{pmatrix} p_{11} & \cdots & p_{1j} \\ \vdots & \ddots & \vdots \\ p_{i1} & \cdots & p_{ij} \end{pmatrix}$$

i = initial state; j = final state

Mathematical Model for Controller Assignments

Controller assignments and actions are pre-determined and represented via the permutation algorithm

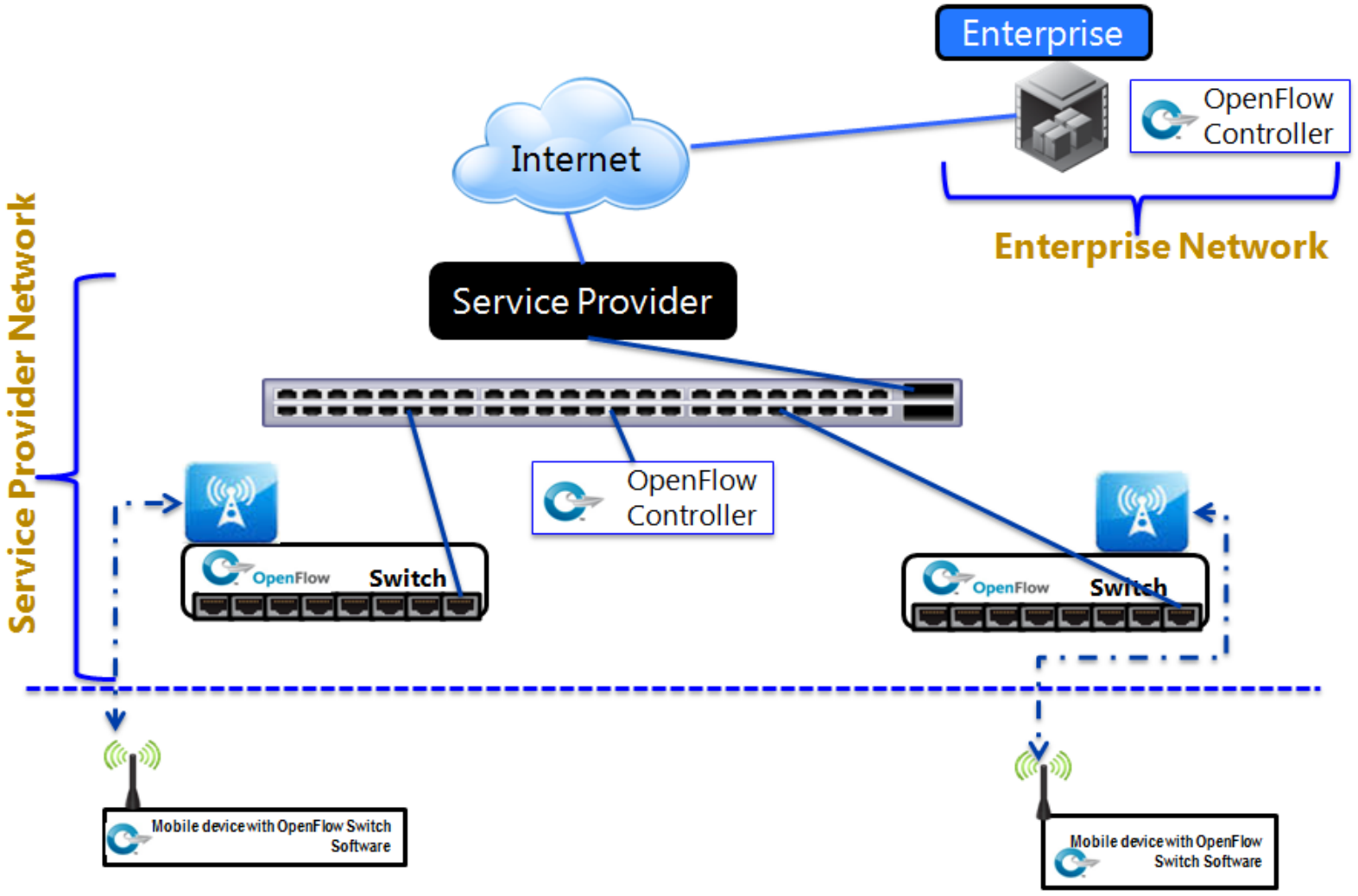
- The order of application of rules processing is pre-defined by either Service Provider, Enterprise or User
- Order of arrangement of these controllers is based on permutation
- Mathematically, Flow table to controller association is represented by

$$n(n-1)+1 \text{ for any } n \geq 1$$

where n is number of controllers

Experimentation

Testing programmability from core to end device using pre-defined flow actions



radhikamh at gmail dot com