OpenRoads: Empowering Research in Mobile Networks

Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Nikhil Handigol,
Te-Yuan Huang, Michael Chan, and Nick McKeown
Stanford University, NEC System Platforms Research Labs, and Deutsche Telekom Inc., R&D Lab
yapkke@stanford.edu, m-kobayashi@eo.jp.nec.com, robert.sherwood@telekom.com,
{nikhilh,huangty,mcfchan,nickm}@stanford.edu

1. INTRODUCTION

Many researchers are working on mobility management. Continued innovation and development in this area can naturally be expected. We anticipate rapid progress will be made in the areas of handover between different technologies, separation of "virtual" wireless service providers from the physical infrastructure, lossless handovers and multihoming solutions. There is no lack of good ideas. However, testing and verifying these ideas at a significant scale has proven to be difficult. The community will therefore benefit from a common platform, in which they can reuse and build upon each other's work. To this end, we developed *OpenRoads*, a platform for innovation in mobile networks.

The goal of OpenRoads is to create an *open* platform in which other researchers and ourselves can experiment and explore different mobility solutions, network controllers and/or routing protocols. OpenRoads provides researchers control of the network through two means, namely, control of the datapath using OpenFlow [2, 3] and control of the device configuration using SNMP.

Via control given by OpenFlow and SNMP, we further provide abstract events and functions for network management in OpenRoads. The abstraction in turns allows us to manage with ease heterogeneous network devices, namely different wireless technologies such as WiFi and WiMAX. Further, sharing the vision of a virtualized network, such as in [1, 5], OpenRoads also presents our attempt to share network control among different researchers.

Through our poster, shown in Fig. 1, we present OpenRoads to the community at large. Our long term goal is to develop OpenRoads into a robust and effective platform for research in mobile networks, that is adopted by other campuses, for experimentation in their production networks.

2. CONTROLLING A NETWORK SLICE

A major distinction between today's cellular network and IP-based network is the control plane. IP has the control plane embedded in each and every router, resulting in an inflexible infrastructure with glacially slow innovation. On the other hand, today's cellular network separates the control plane and rapid innovations in the control plane are common.

A typical Ethernet switch consists of the flow table in hardware and the control logic in software, all of which is enclosed in a single device. Adopting a similar philosophy as the cellular network, OpenFlow separates the control logic of a switch from the physical device, exporting it to an external controller. By controlling the flow table of a switch from an external controller, OpenFlow allows rapid innovation to the datapath of the network. To deploy OpenFlow in our closet (and closets of other researchers), we are adding OpenFlow as feature to commercial switches and wireless devices, such as WiFi APs and WiMAX base-station.

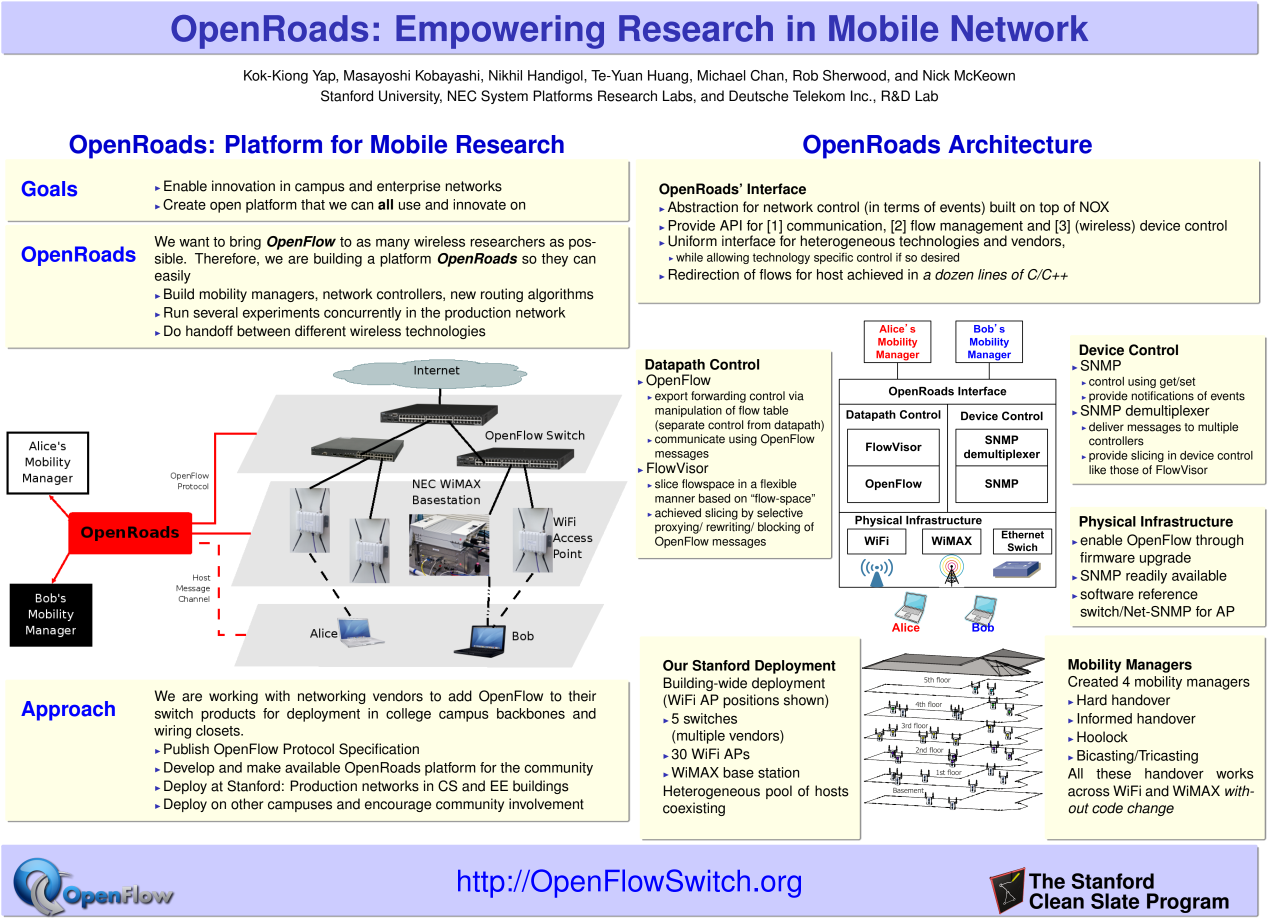
Building on OpenFlow, we developed a platform OpenRoads, which allows researchers to control the network via OpenFlow and SNMP. Our resulting architecture is as shown in right half of Fig. 1. While OpenFlow provides control of forwarding, SNMP provides control of device configuration. Being a widely deployed interface, SNMP allows researchers to configure the switches and wireless access points in a network. Configuration is especially critical in wireless devices, where parameters like transmit power can directly impact on the datapath performance. Further, controllers can also be promptly informed of network events such as a host joining a WiFi AP.

To further share this control of the network among multiple researchers, we use the FlowVisor [4] and SNMP demultiplexer to flexibly divide the control. This is achieved through appropriate proxying, rewriting and policing of the control messages. More specifically, each experiment is given a "flow-space" (a range of header values). By ensuring that experiments cannot exert control beyond their designated "flow-space", FlowVisor provides isolation between experiments. Also, by appearing as a controller to network devices and as devices to the controllers, FlowVisor (and SNMP demultiplexer) allows researchers to develop as if they have complete over the entire network. This allows the scale of the experiment to be expanded without any change in code.

Finally, to allow for rapid development on OpenRoads, we provide abstract high-level control interfaces, such as redirection of a flow in the network and retrieving all flows associated with a host. These interfaces can be categorized as communication (i.e., messaging between devices), flow management (e.g., install flow entries for specific route), and device control (e.g., set power level of WiFi APs).

3. PROGRESS

OpenFlow has been ported as a feature in many switches from multiple vendors, including NEC and HP. Many of these devices already supports SNMP. We have also extended OpenFlow and SNMP into WiFi APs built with PCEngine boxes running Linux, and a NEC WiMAX basestation. Armed with these devices, we have deployed a wireless access network consisting of 5 switches, 30 WiFi APs and a WIMAX basestation. With this physical infrastructure, we are able to validate the usefulness of OpenRoads.



OpenRoads: Empowering Research in Mobile Network

Kok-Kiong Yap, Masayoshi Kobayashi, Nikhil Handigol, Te-Yuan Huang, Michael Chan, Rob Sherwood, and Nick McKeown
Stanford University, NEC System Platforms Research Labs, and Deutsche Telekom Inc., R&D Lab

OpenRoads: Platform for Mobile Research

Goals

- Enable innovation in campus and enterprise networks
- Create open platform that we can **all** use and innovate on

OpenRoads

We want to bring ***OpenFlow*** to as many wireless researchers as possible. Therefore, we are building a platform ***OpenRoads*** so they can easily

- Build mobility managers, network controllers, new routing algorithms
- Run several experiments concurrently in the production network
- Do handoff between different wireless technologies

Approach

We are working with networking vendors to add OpenFlow to their switch products for deployment in college campus backbones and wiring closets.

- Publish OpenFlow Protocol Specification
- Develop and make available OpenRoads platform for the community
- Deploy at Stanford: Production networks in CS and EE buildings
- Deploy on other campuses and encourage community involvement

OpenRoads Architecture

OpenRoads' Interface

- Abstraction for network control (in terms of events) built on top of NOX
- Provide API for [1] communication, [2] flow management and [3] (wireless) device control
- Uniform interface for heterogeneous technologies and vendors,
  - while allowing technology specific control if so desired
- Redirection of flows for host achieved in *a dozen lines of C/C++*

Datapath Control

- OpenFlow
  - export forwarding control via manipulation of flow table (separate control from datapath)
  - communicate using OpenFlow messages
- FlowVisor
  - slice flowspace in a flexible manner based on "flow-space"
  - achieved slicing by selective proxying/ rewriting/ blocking of OpenFlow messages

Device Control

- SNMP
  - control using get/set
  - provide notifications of events
- SNMP demultiplexer
  - deliver messages to multiple controllers
  - provide slicing in device control like those of FlowVisor

Physical Infrastructure

- enable OpenFlow through firmware upgrade
- SNMP readily available
- software reference switch/Net-SNMP for AP

Our Stanford Deployment

Building-wide deployment (WiFi AP positions shown)

- 5 switches (multiple vendors)
- 30 WiFi APs
- WiMAX base station

Heterogeneous pool of hosts coexisting

Mobility Managers

Created 4 mobility managers

- Hard handover
- Informed handover
- Hoolock
- Bicasting/Tricasting

All these handover works across WiFi and WiMAX *without code change*

http://OpenFlowSwitch.org

The Stanford Clean Slate Program

Figure 1: OpenRoads' Poster

Using our OpenRoads platform, we were able to create multiple mobility managers (some of which were created as part of a class) and run them concurrently on a single physical infrastructure. The mobility managers includes hard handover, informed handover, bicasting/tricasting and Hoolock. In informed handover, the client is provided with information to aid handover decisions. While in bicasting/tricasting, flows are multicasted to the client over multiple interfaces. Hoolock is a novel handover scheme which also exploits multiple interfaces on a device to achieve lossless handover, without the need to reorder at the client. All of the above mobility managers are able to perform handover between WiFi and WiMAX without any code change. In these mobility managers, OpenRoads' interface allows us to redirect flows associated with a host in 12 lines of C/C++.

While running these mobility managers, we have a heterogeneous pool of hosts coexisting in the network. For example, the bicasting mobility assumes all devices have more than one network interfaces, while hard handover assumes the contrary, i.e., all devices has one interface. These devices coexists in our network, and are handled by their respective mobility managers without conflict.

We have created a network in which multiple experiments can be run concurrently over the same physical infrastructure. In the long run, we hope OpenRoads will inspire the creation of virtual service providers that are independent of the infrastructure, allowing for seamless handover between different wireless technologies.

4. REFERENCES

[1] N. Feamster, L. Gao, and J. Rexford. How to lease the internet in your spare time. *SIGCOMM Comput. Commun. Rev.*, 37(1):61–64, 2007.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, April 2008.

[3] The OpenFlow Switch Consortium. http://www.openflowswitch.org.

[4] Rob Sherwood, Michael Chan,Glen Gibb, Nikhil Handigol,Te-Yuan Huang,Peyman Kazemian,Masayoshi Kobayashi, David Underhill, Kok-Kiong Yap, and Nick McKeown. Carving research slices out of your production networks with OpenFlow. In *Proceedings of ACM SIGCOMM (Demo)*, Barcelona, Spain, August 2009.

[5] A virtual network infrastructure. http://www.vini-veritas.net.