Blueprint for Introducing Innovation into the Wireless Networks we use every day

Kok-Kiong Yap, Rob Sherwood, Masayoshi Kobayashi, Nikhil Handigol,
Te-Yuan Huang, Michael Chan, Nick McKeown, and Guru Parulkar

Stanford University, Deutsche Telekom R&D Lab, and NEC

OPENFLOW-TR-2009-3

Abstract:

In the past couple of years we've seen quite a change in the wireless industry: Handsets have become mobile computers running user-contributed applications on (potentially) open operating systems. It seems we are on an unstoppable path towards a more open ecosystem; one that has been previously closed and proprietary. The biggest winners are the users, who will have more choice among competing, innovative ideas.

The same cannot be said for the wireless network infrastructure, which remains closed and (mostly) proprietary, and where innovation is bogged down by a glacial standards process. Yet as users, we are constantly surrounded by abundant wireless capacity and multiple wireless networks (WiFi and cellular), with most of the capacity off-limits to us. It seems industry has little incentive to change, preferring to hold onto control as long as possible, keeping an inefficient and closed system in place.

This paper is a "call to arms" to the research community to help move the network forward on a path to greater openness. We envision a world in which users can move freely between any wireless infrastructure, while providing payment to infrastructure owners, encouraging continued investment. We think the best path to get there is to separate the network service from the underlying physical infrastructure, and allow rapid innovation of network services, contributed by researchers, network operators, equipment vendors and third party developers.

We propose to build and deploy an open - but backward compatible - wireless network infrastructure that can be easily deployed on college campuses worldwide, that allows researchers to experiment with new network services directly in their production network.

October 12, 2009

Available at `http://OpenFlowSwitch.org/downloads/technicalreports/openflow-tr-2009-3-openflow-wireless.pdf`

Blueprint for Introducing Innovation into the Wireless Networks we use every day

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

ABSTRACT

In the past couple of years we've seen quite a change in the wireless industry: Handsets have become mobile computers running user-contributed applications on (potentially) open operating systems. It seems we are on an unstoppable path towards a more open ecosystem; one that has been previously closed and proprietary. The biggest winners are the users, who will have more choice among competing, innovative ideas.

The same cannot be said for the wireless network infrastructure, which remains closed and (mostly) proprietary, and where innovation is bogged down by a glacial standards process. Yet as users, we are constantly surrounded by abundant wireless capacity and multiple wireless networks (WiFi and cellular), with most of the capacity off-limits to us. It seems industry has little incentive to change, preferring to hold onto control as long as possible, keeping an inefficient and closed system in place.

This paper is a "call to arms" to the research community to help move the network forward on a path to greater openness. We envision a world in which users can move freely between any wireless infrastructure, while providing payment to infrastructure owners, encouraging continued investment. We think the best path to get there is to separate the network service from the underlying physical infrastructure, and allow rapid innovation of network services, contributed by researchers, network operators, equipment vendors and third party developers.

We propose to build and deploy an open - but backward compatible - wireless network infrastructure that can be easily deployed on college campuses worldwide, that allows researchers to experiment with new network services directly in their production network.

1. INTRODUCTION

There is currently much excitement in the air about openness in wireless and mobile computing. Users can choose from a thriving array of handsets and, in many countries, use their handset with a variety of commercial carriers. A burgeoning army of third-party developers are creating applications, games and content for mobile devices. And the Android operating system claims to be the "first truly open and comprehensive platform for mobile devices; all of the software to run a mobile phone, but without the proprietary obstacles that have hindered mobile innovation." [4]

Arguably these are all positive steps towards a more open ecosystem for the mobile world, creating more choice for users. We applaud the move towards openness and are great believers in the power of choice in the marketplace to bring innovation, efficiency and high quality service to the user. Industry benefits too. An innovative marketplace grows the business for everyone, and attracts new players eager to compete with incumbents. Openness creates choice which breeds innovation.

But despite all the progress, there are still real structural barriers to openness – barriers that industry and government will not break down on their own – requiring technical innovation. This is the domain of university researchers. In the spirit of enabling choice and innovation through openness, we describe here how we, as a research community, can set out to break down two large technological barriers:

1. **The inaccessible and closed wireless capacity around us.** Today, if we stand in the middle of a city, we can likely "see" multiple cellular and WiFi networks. But, frustratingly, these infrastructures are not available for us to use. Cellular companies restrict us to use their network; most private WiFi networks require authentication, and are effectively inaccessible. Although we are often surrounded by abundant wireless capacity, almost all is off-limits; our choice is almost non-existent. We believe users should be free to travel in a rich field of wireless networks with access to all infrastructure around them. Openness doesn't mean free; here it means a healthy market-place with lower-cost connectivity and broader coverage. In the ex-

treme, if all barriers to fluidity can be removed, users could connect to multiple networks at the same time, opening up enormous capacity and coverage.

2. **A network infrastructure that is closed to innovation.** Cellular networks increasingly use IP. IP has been tremendously successful in bringing choice and innovation to the end user: Arguably its greatest feat is enabling innovation at the edges. IP is simple, standardized, and provides universal connectivity. But we believe that as-is, IP is not the right choice for the future mobile Internet: It is ill-suited to support mobility and security; and it is hard to manage. Its architecture is fixed, allowing little room to add new capabilities. Today cellular providers feel the pain from poor support for mobility, security, and innovations in general. If we tweak IP to solve these problems, we will find new limitations. We need a network that allows continued innovation, for services we can't yet imagine, while allowing existing applications to work unchanged.

So as we look to the future, we want a network that will allow any mobile computer to connect to any network, and to move freely and seamlessly from one network to another. The logical next step is for a handheld to connect to any network around it – regardless of who owns the network and what radio technology it uses, as exemplified in Figure 1. While there are obvious non-technical barriers that stand in our way, e.g., economic and regulatory, we believe a new network architecture is needed to break down these barriers.

In this new architecture, there will exist lots of service providers, lots of radios, and lots of types of radios, all tied together by lots of wired networks. There will be diversity at all levels: diversity in network (many networks to choose from), channels (more spectrum will become available), antennas (more MIMO), radios (a handheld will contain many radios). Whereas today's phones commonly have three or four radios (e.g. GSM, WiFi, Bluetooth), in future they will have more. Shrinking geometries and energy-efficient circuit design will lead to mobile devices with ten or more radios, with several of the same type. A handheld may connect to several networks at once, for robustness and increased signal quality. If users are to move freely among many networks, the service provider needs to be separate from the network owner. Service provider should handle the mobility, authentication and billing for their users, regardless of the network they are connected to. To a limited extent, this is happening: Some cellular companies allow MVNOs[1] to provide services over their network. And in WiFi networks, when we login to a hotel or airport network, a third party provides authentication and billing services. We assert that all these will allow streaming applications to operate seamlessly as we walk, drive, or fly.

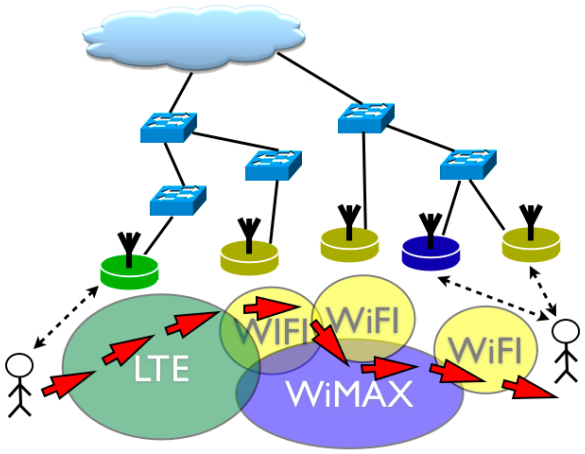

Figure 1: Vision for Future Wireless Mobile Internet: Choice of providers, networks and technologies.

Our general approach is to open up that which has been closed – to help industry move towards an open network architecture. As cellular providers make the transition to IP, we would like to enable them to innovate in their own network. We would like them to be able to research and experiment with new security models (e.g. new approaches to access control, and user authentication), and with new, more scalable alternatives to mobile IP. An open architecture allows a new industry of suppliers to provide improved features in the network; and it will help an open-source community to grow, which in turn will provide contributions to all users.

We believe the research community has a big part to play in bringing this new open architecture to fruition, and there are many interesting research problems to be solved along the way. For example, in service of end-users, we need to figure out how to cleanly cleave the network service from the underling network infrastructure. A clean separation of "service from infrastructure' across different networks (cellular providers, home networks, enterprise networks,

[1]MVNO: Mobile Virtual Network Operator. In the US, Virgin is an MVNO in Sprint's network; Sprint owns the radios and wired network, and Virgin provides branded AAA and billing services for its customers. In some countries, notably Holland, hundreds of MVNOs compete over a small number of physical networks.

coffee-shop networks, etc.) and across different types of wireless network (e.g. GSM, WiFi, WiMAX, LTE) would give us access to a lot more wireless capacity, and more competition among providers. Other research includes how to create a personalized mobility manager that lives in the cloud in service of one or more customers. A personal mobility manager can implement a user's preferences for routing, network selection and pricing. An open network will enable new experiments with location-aware services. And it will enable experiments with new, large scalable directory services for a population of billions of mobile users and services. Finally, we can research ways to improve measurement and instrumentation of the network, to allow users to compare service quality from different providers and in different networks.

To support this vision and research along the way, we present OpenRoads (§ 2), a blueprint for this open network architecture. We also describe the current deployment of OpenRoads (§ 3) on our campus, and how it has already been used in classrooms to enable new research in wireless networks (§ 3.3). We conclude the paper with a call to action for the research community to join in our expedition towards this vision of openness (§ 4).

2. OPENROADS: BLUEPRINT FOR AN OPEN WIRELESS NETWORK

In support of our vision – and as a first step towards engaging the broader research community – we propose OpenRoads: a mobile wireless network platform enabling experimental research and realistic deployments of networks and services. Figure 2 provides an overview of OpenRoads. OpenRoads uses OpenFlow to separate control from the datapath through an open API, FlowVisor [11] to create network slices and provide isolation among them, hence allowing multiple experiments to run simultaneously in *production* wireless network, and SNMPVisor to mediate device configuration access among experiments. These components relate directly back to our vision for future wireless Internet design, in terms of decoupling mobility from physical network (OpenFlow), and allowing multiple service providers to concurrently control (FlowVisor) and configure (SNMPVisor) the underlying infrastructure.

2.1 OpenFlow

OpenFlow [6] is a feature added to switches, routers, access points (APs) and basestations, allowing these datapath devices to be controlled through an external, standardized API. OpenFlow exploits the fact that almost all datapath devices already contain a flow-table (originally put there to hold firewall ACLs),

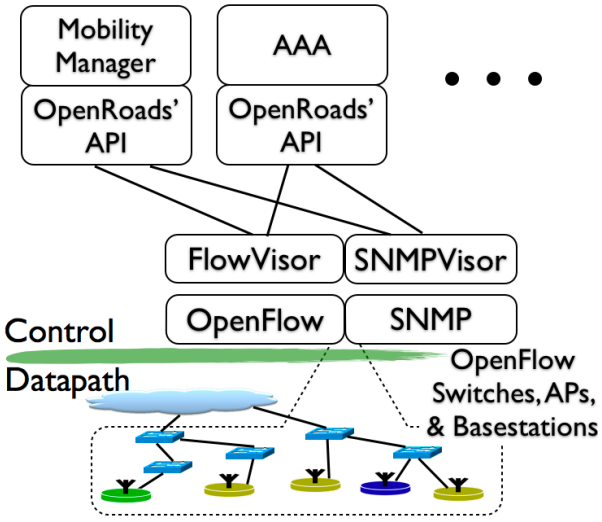

Figure 2: OpenRoads Architecture

although current switches and routers don't have a common external interface. In OpenRoads, we add OpenFlow to WiFi APs and WiMAX basestations as well by modifying their software; and in principle the same thing could be done for LTE and other cellular technologies.

In OpenFlow - and therefore in OpenRoads - the network datapath is controlled by one or more remote controllers that run on a PC. In our network we use a freely available open-source controller NOX [9] from Nicira; but in principle any controller is possible, so long as it speaks the OpenFlow protocol. The controller manages the flow-table in all the datapath elements and gets to decide packets are routed in the network. In this manner, the datapath and its control are separated, and the controller has complete control over the operation of the datapath. The controller can define the granularity of a flow. For example a flow can consist of a single TCP session or any combination of packet headers (Layer 1-4) allowing aggregation.[2]

As an example of control and datapath separation, a "mobility manager" in OpenRoads can be implemented as a NOX application. NOX provides network-wide visibility of the current topology, link-state and flow-state; and all other network events. The mobility manager can choose to be made aware of every new application flow in the network, and can pick the route it takes. When the user moves, the mobility manager is notified, and can decide to

[2]More information about OpenFlow can be found at `http://OpenFlowSwitch.org`, including reference systems, specifications, and a list of commercial switches supporting the OpenFlow protocol.

re-route the flow. Because OpenFlow is independent of the physical layer (i.e., whether the wireless termination point is running WiFi or WiMAX), vertical handoff between different radio networks is transparent and simple.

The openness of the controller makes it is easy to add or change the functionality of the network. For example, a researcher can create a new mobility manager (e.g., one that does faster or lossless handoff) by simply modifying an existing one. In our prototype deployment (§3.3), we have already seen this happen many times, as researchers and students exchange code and build on each others work. In this way, we believe rapid innovation is possible. Further, by separating the datapath and its control, we can reap many benefits of centralized control (see below, *A Trend Towards Centralized Control of WiFi Networks*). Anecdotally, we have found network administrators receptive to a centrally managed network that is easily monitored.

Taken to the extreme, an application could be an entire mobility service, akin to the cellular service we buy from companies like AT&T, Vodafone, Orange, etc. An application can be written to implement AAA, billing, routing, directory services and so on... all running as a program on an OpenFlow controller. And because the controller is simply a program running on a server, it can be placed anywhere in the network - even in a remote data center. In the short term, we expect applications and experiments to be much simpler; but, in the long-term, an open network can grow to support a wealth of new services.

A Trend Towards Centralized Control of WiFi Networks

Independently, campus WiFi networks are rapidly becoming more centralized. A central controller manages many wireless access points across a campus, allocating non-overlapping channels, setting power levels to reduce interference, and authenticating users in a single consistent way. Mobile users can roam freely across a campus without changing IP address, and without dropping TCP connections. Such a network is "software-defined", with the entire network operation determined by code running in the controller. When new features are needed in the networks (e.g. new routing algorithms, new authentication policies, etc.) the manufacturers simply update the central controllers. This trend is typified by a range of products from Cisco, Aruba, Meraki, Meru [2, 1, 8, 7] and a set of standards [5]; together they show a trend towards central control, and show it is viable (although commercial controllers are proprietary and closed, and are designed to work with specific products in a particular market segment). Our experience so far is that network administrators like centralized control of their wireless network: it provides a single point of management, network configuration (e.g., access control) is more likely to stay consistent, and the network is easier to upgrade. OpenRoads provides this in an open way, building on reliable hardware, presenting (probably the first) practical way to open up the wireless network we all use everyday. We contend the synergy between administrators' trend towards centralized control and researchers' desire to work with production systems provides a powerful "carrot" for deploying OpenRoads as campus networks.

2.2 Hosting Multiple Simultaneous Experiments

Although we've explained how we can run a new experimental service in the OpenRoads network, it still begs the question of how we can have *multiple* competing services running at the same time in the same network. And how one service could allow its users to roam freely across multiple physical networks. The trick here is to *slice* the network, allowing multiple controllers to co-exist, each controlling a different slice of the network. A slice may consist of one user or many users; one network or many networks; one subset of traffic or all traffic. So, we use the *FlowVisor*: an open-source application created specifically to slice OpenFlow networks.

FlowVisor slices the network by delegating control of different flows to different controllers. As shown in Figure 2, FlowVisor is an additional layer added between the datapath and controllers. Because the FlowVisor speaks the OpenFlow protocol to the datapath, they believe they are controlled by a single controller (the FlowVisor); and because the FlowVisor speaks OpenFlow to the controllers above, they think they each control their own private network of switches; i.e. FlowVisor is a transparent proxy for OpenFlow. The trick is to correctly isolate the flows according to a policy, and hence create one slice per experiment with its own private "flow-space" (a range of header values). FlowVisor works by deciding which OpenFlow messages belong to each slice, and pass them to the controller for that slice. If, for example, Controller A is responsible for all of Alice's traffic, then the FlowVisor passes all control messages relevant to Alice to Controller A. Therefore, FlowVisor separates slices according to a policy, defined by the network manager, by providing a strict communication isolation between slices.

A direct consequence of slicing the network is that we can safely run experiments in a production net-

work. The FlowVisor allocates "flow-space" by default to the production network, which can be routed using legacy protocols. Each experiment is assigned its own slice, defined by the "flow-space" and topology, and implemented by the FlowVisor. Because real users are already connected to the production network, it makes opt-in relatively simple. If the network is sufficiently large, then experiments can be run at the same scale as, say, a campus wireless network. Or could even be run over multiple networks on multiple campuses.

Slicing also allows "versioning" in the production network, where new features can gradually be incorporated into the production slice. Different slices can be dedicated to different versions, some more stable than others, as new features are carefully rolled out in stages. In this way, new features can be deployed and tested quickly, then gradually made available network-wide, and even shared with the owners of other campus networks. Such an ecosystem allows for the survival of the fittest, bringing the best to users. Also, legacy clients can be supported on a separate legacy slice, and the network can now evolve without being held-back by backward compatibility.

Finally, slicing allows delegation. Network administrators can cascade FlowVisors to further delegate (or slice) the flow space allocated to them. Repeated delegation makes sense in networks with a hierarchy of control; for example, in our network the campus network manager delegates a slice of the network for research experiments in our group, which we in turn slice (using another FlowVisor) among different experiments.

2.3 Configuring the Datapath

While OpenFlow provides a means to control the OpenRoads datapath, it doesn't provide a way to configure the datapath elements: e.g. to set power levels, allocate channels, enable and disable interfaces. This job is normally left to a command line interface, SNMP or NetConf. Although simple in principle, configuration is tricky in a sliced network, as we want to configure each slice independently. For example, we might wish to disable a certain network interface in one slice, without disabling the same physical interface shared by another slice.

We slice datapath configuration using a "SNMPVisor", that runs alongside the FlowVisor, to allow an experimenter to configure his slice. FlowVisor slices the datapath, and SNMPVisor slices the configuration by watching SNMP control messages, and sending them (and possibly modifying them) to the correct datapath element. Similar to FlowVisor, SNMPVisor acts as a transparent SNMP proxy between the datapath and controllers, providing the same features of versioning and delegation.

But sometimes it's hard or impossible to slice the configuration: For example, setting power levels for different slices on a WiFi AP. If slices share a channel, then we want to set different transmit power levels for the flows in each slice - something not possible on existing APs. We follow the general mantra of slicing where we can, and exposing non-sliceable configuration parameters to the user via feedback and error messages.

3. AN OPENROADS DEPLOYMENT

To gain some early experience with OpenRoads, we built a prototype – based on our blueprint – and deployed on our campus. Along with the details of our implementation and deployment, we also describe an often overlooked but crucial component, our measurement infrastructure. We implemented OpenRoads on top of NOX OpenFlow controller, through which we control the switches, APs and basestations; and we slice the network using FlowVisor. We also extended SNMP into NOX to let us control power, frequency, data rate, SSID, etc., and to capture wireless events (like when hosts associate with an AP). All of these tools are freely available under open-source licenses [10], and we encourage the community to help us improve them over time.

3.1 Datapath elements: Access Points, Basestations, and Switches

Our initial deployment consists of 30 WiFi APs, 2 WiMAX basestation and 5 Gigabit Ethernet switches in our wiring closets.

WiFi: Our WiFi APs, an ALIX PCEngine embedded computer running Linux, have two radios and cost about $200. We will shortly make available a lower-cost OpenWRT-based AP as well.

WiMAX basestation: We added OpenFlow to an NEC WiMAX basestation, and placed it in our network under an FCC research spectrum license. The basestation essentially operates as a dumb WiMAX AP running OpenFlow.

Switches: We use HP and NEC Ethernet switches in our wiring closets:, NEC IP880 24/48-port GE and HP 5406 chassis GE switches, with firmware upgrades to support OpenFlow.

For future work, we plan to (and hope the community will also) experiment with more exotic hardware. For example, we could attach experimental programmable radios to our deployment, e.g., GNURadio [3] and WARP [12], add OpenFlow interface to them, and then exploit their extra programmability through SNMP or NetConf.

3.2 Logging and Measurement

To better ensure that the results of our experiments are accurate and repeatable, we implement a comprehensive measurement component as part of the OpenRoads deployment. With NOX's event logs, we record all network events including changes in topology, link-state, flow-state, etc. We have created a number of visualization, monitoring tools and GUIs running on top of NOX to help users instrument the state of their slice, and draw temporal correlations between network events. We believe that the completeness of the measurement infrastructure will further ease innovation in open wireless networks, exposing important problems, trends and interactions in the network.

3.3 Early Mobility Experiments on OpenRoads

As a first foray into creating experiments on OpenRoads, we charged students in a 12 week project-based class to design and deploy their own novel mobility manager, then deploy it into the network. Some interesting designs resulted: For example, one group designed a mobility manager to perform lossless handoff for fast-moving users. Another group created a "high-reliability" service by n-casting packets in the network (using OpenFlow) so that every client received multiple copies over different paths and radios. And another group used network state information from NOX to predict which channel they should use during a handover, to minimize the hunt time. In each project, the students demonstrated the manager working in our actual production network, running simultaneously in their own slice. Due to OpenRoads' design, all of the mobility managers were immediately able to do handover between WiFi and WiMAX; and they all worked without modifying end host software or applications.

The point here is not that the mobility managers were radically new; it is that each one was written by non-experts in less than 4 weeks by building on top of a growing base of open-source code and a steadily improving platform. While we have many things to improve still - this is just a first prototype - we were surprised to find that each mobility manager could be written in approximately 200 lines of C++. We take these experiences as preliminary validation that a system like OpenRoads will indeed be useful to the research community by easing the innovation process in wireless networks.

4. CONCLUSION

The architecture of wireless networks is going to change significantly in the coming years, with a slow convergence of the cellular and WiFi networks. Without us, the industry will stay closed and based on proprietary equipment. Our role as the research community, is to help open up the infrastructure - to allow multiple ideas to co-exist in the same physical network - and therefore allow innovation to happen more freely and more quickly. Opening a closed infrastructure might seem like a naive pipedream; but recall the change that Linux brought to the computer industry by a dedicated community of open-source developers. We believe the best place to start opening the wireless infrastructure is on our own college campuses, by replacing our wireless networks with a more open (and backwardly compatible) alternative. We call this new network "OpenRoads".

OpenRoads builds right on top of OpenFlow (which is, itself, making progress in wireline networks). The main technical additions are the ability to slice the network using the FlowVisor (to slice the datapath) and SNMPVisor (to slice configuration); and the ability for users to opt-in to experiments. As a whole, OpenRoads forms a complete production network that can be sliced – by the existing network administrator – to create isolated slices for new experiments or new versions of features.

The blueprint we present here is a starting point. We hope our platform and this paper serve as a "call to arms" to the networking research community to come together and adopt, build, deploy and use the OpenRoads mobile wireless network architecture, building the system using existing access points, stripped down cellular base stations, wireline switches, and placed under the control of an open system that allows multiple isolated experiments to run concurrently in the network. We propose that the resulting network be widely deployed as our campus production wireless network. Done right, we believe that the community can share code, build on each other's work, and eventually create a common infrastructure in which researchers can safely try out new ideas, and network administrators can pick the best ideas to deploy in their production network and thus showcase "the future" on campuses before it is realized broadly.

5. REFERENCES

[1] Aruba Networks. `http://www.arubanetworks.com`.
[2] Mobility - Cisco Systems. `http://www.cisco.com/en/US/netsol/ns175/networking_solutions_solution_segment_home.html`.
[3] GNU Radio. `gnuradio.org`.
[4] Official google blog: Where's my gphone? `http://googleblog.blogspot.com/2007/11/wheres-my-gphone.html`.
[5] Internet-Drafts Database Interface. `http://ftp.ietf.org/drafts/draft-ohara-capwap-lwapp/`.

[6] 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.

[7] Enterprise Wireless LAN Networks: Indoor and Outdoor Wireless Networks By Meraki. `www.Meraki.com`.

[8] Meru Networks. `http://www.merunetworks.com/`.

[9] NOX: An OpenFlow Controller. `http://noxrepo.org/wp/`.

[10] OpenFlow Wireless. `http://www.openflowswitch.org/wk/index.php/OpenRoads`.

[11] Rob Sherwood, Michael Chan, Adam Covington, Glen Gibb, Mario Flajslik, Nikhil Handigol, Te-Yuan Huang, Peyman Kazemian, Masayoshi Kobayashi, Jad Naous, Srinivasan Seetharaman, David Underhill, Tatsuya Yabe, Kok-Kiong Yap, Yiannis Yiakoumis, Hongyi Zeng, Guido Appenzeller, Ramesh Johari, Nick McKeown, and Guru Parulkar. Carving research slices out of your production networks with OpenFlow. In *Proceedings of ACM SIGCOMM (Demo)*, Barcelona, Spain, August 2009.

[12] Rice University WARP - Wireless Open-Access Research Platform. `warp.rice.edu`.