

Towards Rapid Innovation in Mobile Networks

[Extended Abstract]

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}@stanford.edu, nickm@stanford.edu

ABSTRACT

Today’s mobile network presents an oxymoron. While innovative services are sought after by ISPs, deployment of novel services by small businesses and researchers are often met with great resistance. This stems from the coupling of infrastructure and service providers, preventing multiple services to coexist in a single infrastructure. To explore for a remedy to this, we design, build and deployed a wireless network in which multiple services can coexist within a single infrastructure. In this work, we share our design and experience gained.

1. INTRODUCTION

Mobile networks today presents a peculiar scenario. On one hand, Internet service providers (ISPs) are compelled to offer innovative and attractive services to the customer. On the other, small businesses and researchers are unable to innovate and provide novel services on that infrastructure. Among the phelthora of innovations, those compelling service(s) that ISPs are looking for are likely to be found. Wouldn’t it be great if small businesses and researchers can innovate in the mobile networks?

This awkward situation arises from the split “personality” of the ISPs. Beyond being the infrastructure builder and owner, ISPs seek to be the service provider. This is often a mismatch due to the end-to-end nature of services, and localized nature of infrastructure. For example, mobile users desire pervasive connectivity. However in today’s network, we can typically only connect to a single ISP. Regardless of how poor the connection is, we cannot communicate via that nearby base-station belonging to the other ISP. Ironically, if we subscribed to a foreign ISP, we could roam between these networks.

To address this issue, we seek a “virtualized” infrastructure, such as Cabo [1], to decouple the service and infrastructure providers. In such an network, service providers (which could be the ISPs, small businesses or even researchers) can co-

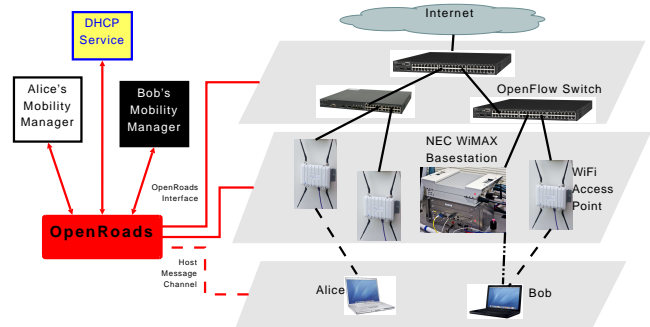


Figure 1: Multiple Services on OpenRoads

exists, and deliver truly end-to-end services to mobile users. Building such an infrastructure is not without its challenges. On *OpenRoads*, we made our first cut attempt to build such a network, within which multiple service providers can coexist within a single physical infrastructure, as illustrated in Fig. 1.

To accomplish this, we need to provide isolation in midst of interaction. With reference to the example in Fig. 1, Alice’s mobility manager should not be able to affect Bob’s traffic. More importantly, if Bob’s mobility manager is to fail, Alice’s connectivity should not be affected. Further, both Alice and Bob could be served by the same DHCP service. This can be achieved by enforcing a consistent policy layer in our system. Our objective here is to share our design and experience with the community.

2. SLICING THE NETWORK

To allow multiple controllers within a single network, *OpenRoads* divides the control into datapath and device, as shown in Fig. 2. Datapath control refers to the ability to redirect traffic in the network, for which we adopted OpenFlow [2, 3]. OpenFlow separates the control logic from the switch itself, allowing us to manipulate the traffic transversing production devices in the network. This allows us to process the packets at line rate, while maintaining a large degree of control over the traffic. Meanwhile, device control refers to configuration of the devices. Here, an arbitrary choice of SNMP is made. We note that control of devices complements the datapath control, provided by OpenFlow. For example, we might want to change the transmit power of the WiFi AP, which in turns affects the reliability of a particular link in the network.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2009 ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

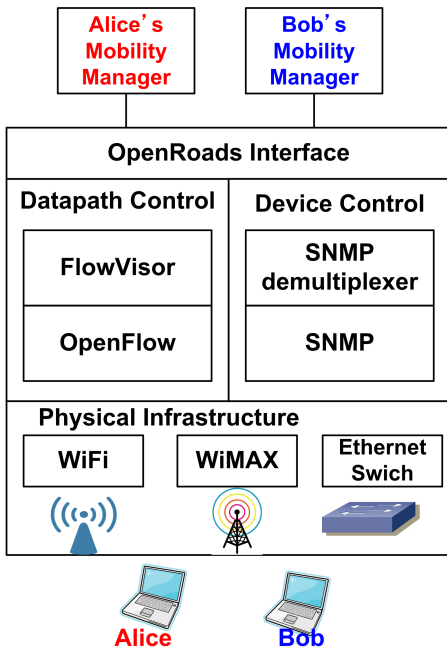


Figure 2: OpenRoads' Architecture

To slice the network, we exploit FlowVisor and SNMP multiplexer as a policy layer. This policy layer appropriately modify and disseminated control packets of OpenFlow and SNMP. Further, the layer allows enforcement of consistency in the network. As such, Alice's mobility manager cannot redirect Bob's traffic, since the command will be discarded by this policy layer, providing the desired isolation mechanism. Also, DHCP request from Alice can be sent to both the DHCP service and Alice's mobility manager. This allows Alice's mobility manager to maintain her network location through observing such traffic, while receiving her IP address from an independent DHCP service of her choice.

Further, we argue that technology and vendor agnosticism is an important factor in our system. Given that a service provider may operate over multiple network slices, spanning across multiple infrastructure providers, it would be impractical to cater for each and every vendor equipment separately. Therefore, a common interface to all devices performing the same functionality is required. In our system, that interface is OpenFlow and SNMP, abstracted as OpenRoads interface to the application programmer. This argument can be extended to technology, where a link over WiMAX or a link over WiFi, should not perturb mobility managers that primarily concerned with redirection of flows.

It should be mentioned that our system allow but do not require application programmers to be technology and vendor agnostic. For example, one might want to exploit a vendor specific feature on the switch or fine tune the wireless link's parameter for the application at hand. This can be done in our system, with the same interface. We contend that such flexibility while having a degree of uniformity is important for rapid innovation in the network.

3. SERVICES AND CHALLENGES

To gain a better understanding of our system, we built and deployed our design in Stanford's Gates building. Deploying 4 OpenFlow switches (from NEC and HP), 30 WiFi APs, and a NEC WiMAX base stations, we created an experimental testbed on which we can deploy new services.

To date, we have created 4 distinct mobility managers, which we run concurrently on our deployed network. Our system allowed us to simultaneously support an heterogeneous pool of clients, some of which have multiple interfaces, while others may have connections over WiMAX instead of WiFi. We see that this allows us to deploy new mobility managers, without worrying about existing clients. This in turn allow services to evolve over time, without worries of backward compatibility.

Also, since a separate IP address space has been allocated for clients on our wireless networks, we created a DHCP service that replies to DHCP requests from wireless connections only. By selective proxy of applicable requests to our DHCP server, and forwarding the rest to Stanford's, we have demonstrated the feasibility of hosting multiple services, such as DHCP, in the same physical network¹ without any hardware changes. We are also able to create other novel services on this infrastructure, e.g., a service that predict a laptop's next AP is created to help reduce scan times of WiFi.

From our experience, flexible slicing of the network have been crucial in our ability to host multiple services within the same physical infrastructure. Further, a strong isolation mechanism allows the numerous parties involved to share the network, without fear of bad interaction. Vendor and technology agnosticism have also accelerated the development of new and interesting applications in our network.

Many challenges remain in building our ideal system. We have yet to understand how our system can be scaled to multiple administrative domains, or how a service can be deployed on multiple network slices spanning across multiple physical networks. We see our work as the first steps towards creating an "app-store" of services in the network, a critical component towards rapid innovation in mobile networks.

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] OpenFlow Consortium. Openflow switch specification, version 0.8.9 (draft). <http://www.openflowswitch.org/documents/openflow-spec-v0.8.9.pdf>.

¹It is worth noting that both DHCP services runs in the same VLAN in our network.