Slicing Home Networks

Yiannis Yiakoumis Kok-Kiong Yap Sachin Katti Guru Parulkar Nick McKeown
Stanford University
{yiannisy,yapkke,skatti,parulkar,nickm}@stanford.edu

ABSTRACT

Despite the popularity of home networks, they face a number of systemic problems: (i) Broadband networks are expensive to deploy; and it is not clear how the cost can be shared by several service providers; (ii) Home networks are getting harder to manage as we connect more devices, use new applications, and rely on them for entertainment, communication and work—it is common for home networks to be poorly managed, insecure or just plain broken; and (iii) It is not clear how home networks will steadily improve, after they have been deployed, to provide steadily better service to home users.

In this paper we propose *slicing* home networks as a way to overcome these problems. As a mechanism, slicing allows multiple service providers to share a common infrastructure; and supports many policies and business models for cost sharing. We propose four requirements for slicing home networks: bandwidth and traffic isolation between slices, independent control of each slice, and the ability to modify and improve the behavior of a slice. We explore how these requirements allow cost-sharing, out-sourced management of home networks, and the ability to customize a slice to provide higher-quality service. Finally, we describe an initial prototype that we are deploying in homes.

Categories and Subject Descriptors

C2.1 [**Computer Systems Organization**]: COMPUTER-COMMUNICATION NETWORKS—*Network Architecture and Design*

General Terms

Design, Experimentation, Management

Keywords

Home Networks, Software-Defined Networks, Network Slicing, OpenFlow

HomeNets'11, August 15, 2011, Toronto, Ontario, Canada.

1. INTRODUCTION

Broadband connectivity, and a network inside the home, are essential ingredients of a modern household. A large variety of home devices connect to the Internet, and high bandwidth Internet applications such as video and audio streaming, high quality video conferencing, file sharing and backup are now commonplace. But despite huge investments in broadband, and over a decade of experience with home WiFi, home networks still face a number of systemic challenges:

Expensive to deploy: Broadband connections to the home require huge investment for initial buildout, and continued high investment to improve data rates. Governments and industry are investing many billions of dollars in broadband connectivity [2, 6]. Yet in almost every case, the cost is borne by a single investor—there are no effective ways to amortize the cost and share the burden among several providers. Almost everyone loses: Investors must take on enormous risk; service providers fear becoming "dumb bit pipes", unable to reap the rewards of their huge investment (and hence fueling concerns about network neutrality); most home owners can only choose between one or two providers, giving users little choice. Even if several providers would like to share the cost of the broadband connection (e.g. an Internet provider such as Verizon, and a utility provider for reading meters), there are no mechanisms to isolate the traffic and bandwidth between providers.

Hard to manage: Most home users lack the technical know-how (or the desire) to manage their home network. Networks typically operate with the default settings shipped with the access router. Worse, a large number of access routers are returned because users could not get them to function[1]. While there has been plenty of discussion of outsourcing the management of networks inside the home, it is not widespread. There is no common interface to home networking equipment to allow it to be remotely controlled.

Prone to failure: Networks in the home often suffer unpredictable failures (e.g high packet-loss and disconnection). Reasons for failures are numerous, ranging from poor AP placement, to interference with neighboring WiFi APs, to misconfigured APs and network settings. Presumably a service provider with access to more expert resources and technical know-how can better diagnose these failures and manage the network to deliver predictable and good performance. However, in spite of the strong incentives (e.g. Net-

[1]25% of wireless networking equipment is returned (of which 90% is fully functional)—one of the highest return rates among consumer electronics [3].

flix would be motivated to carefully manage the home network to deliver good streaming quality), current home networks lack the affordable mechanisms for the service provider to innovate and improve the behavior of the home network to increase the quality of the product they deliver.

Motivated by these challenges, we argue that future home networks should provide two capabilities. First, the capability to virtually split a physical network into multiple *slices*, and second, the capability to allow independent programmatic control of each slice. We believe such home network *slicing* can help solve the problems described above. Slicing can help reduce the cost of deployment, because it allows multiple providers to share the cost of deploying the physical network, and then co-exist on the deployed network substrate. For example, an internet service provider and a utility company might share the cost of deploying broadband to the home, and simultaneously use the network for providing broadband connectivity and smart metering, respectively. In the particular sliced networks we propose, the control plane is separated from the dataplane, and therefore users can out-source network configuration and management. Trusted third parties can programmatically control slices to better manage WiFi configuration, improve routing and implement access control (e.g. configure WiFi channel and power to minimize interference and/or set parental controls). We can take it one step further, and allow an application provider (e.g. Netflix) to control their own slice, and customize it to provide higher quality user experience.

Our proposed approach poses several new challenging requirements for the home network; this paper merely raises these concerns and suggests initial ideas on how to tackle them:

- *Isolation of traffic:* To safely share a network among multiple slices, the slicing mechanism has to ensure that data should not leak from one slice to another.
- *Isolation of bandwidth:* A slice should not be able to "starve" another slice of its bandwidth. This may include bandwidth on links, or in network processing elements, such as routers and access points.
- *Independent control:* Each provider should be able to control and manage its own slice as if it owns the underlying physical network. Separation of control gives each provider the opportunity to impove reliability, while allowing clear accountability.
- *Ability to customize and modify:* A provider should be able to modify and customize a slice to optimize the application according to its specific needs. For example it should decide which packets are blocked, how packets are routed, the power level of a wireless channel, and so on.

Our focus in this paper is on the design of the slicing mechanism itself, and not on the policies that govern how slicing is used or the individual services within a slice. As discussed above, the capability to slice enables a number of interesting economic models, from network sharing to outsourced network management to guaranteed QoS for specific content providers. Some of these models also have implications for regulatory policies such as network neutrality. Our goal however is modest: specifically, we aim to separate mechanism from policy, and design a mechanism that supports a number of policies and enables their independent evolution.

In the following section, we discuss in some detail a few representative applications of slicing. In particular, we will examine sharing the investment in broadband connections between Internet providers, smart-grid utilities and cellular companies; outsourcing network management and guest WiFi; and dedicated slices for video streaming. In §3 we describe one possible design for a sliced home network; in §4 we report on a prototype we are building and deploying to experiment with slicing in home networks. We refer to related work and conclude in §5 and §6.

2. APPLICATIONS

We pick three classes of application to motivate our exploration of slicing home networks. For each, we consider the importance of our four requirements: traffic isolation, bandwidth isolation, individual control, and the ability to modify network behavior.

2.1 Sharing the Physical Network

Slicing would allow multiple providers to share one physical network and amortize their deployment costs. For example, consider smart metering for "smart grids". Electricity and gas suppliers are rolling out a parallel network to the home to enable measurement and metering, and in some cases control of home devices to improve energy efficiency and load prediction. Deploying, managing, and maintaining a separate physical infrastructure is expensive. Instead, if the broadband connection and the home network are sliced, the utility provider can share the cost, or rent a slice from the broadband provider. If the network is sliced inside the home, the slice could extend right to the home devices.

Such a slice would have strict requirements. First, to guarantee reliable billing, and clear accountability, the slice must be tamper-proof. Billing and accounting traffic should be isolated to stay private within the slice, and should keep out traffic from other slices. Second, if devices are to be remotely controlled, then the slice may need bandwidth and latency guarantees. There needs to be a way for the utility to express the bandwidth and latency requirements of a slice (or subset of the slice's traffic), and for the slicing mechanism to enforce it. If the utility wishes to enhance the behavior of the slice—e.g. to make it more secure or more reliable over time—then the utility should be able to provide and modify the control plane of its slice.

Cellular providers are already expanding their coverage by placing micro- and pico-cells in homes, and offloading cellular traffic to their customers' home and broadband networks. Today, the traffic is carried best-effort by the broadband network. If instead the cellular providers could rent a slice (or share the broadband deployment cost), they could potentially control the slice all the way to the micro-cell basestation. As before, traffic isolation is important for privacy and billing; and bandwidth isolation is important to guarantee good user experience. If the cellular providers can provide their own control plane for their slice, they can control routing and handoff, and could even support and route between multiple micro-cells per property, or use one micro-cell to serve multiple customers. In the extreme, in an urban setting, the cellular operator could obtain multiple slices and aggregate them to create (or augment) a city-wide wireless network from home WiFi deployments. A standardized slicing mechanism would allow multiple ways for such a network to be built and paid for. Deploying an open mechanism to-

day, would allow new policies and business models to evolve over time.

A related, but simpler scenario is to create an isolated "Guest WiFi" slice in the home network, to allow friends and guests to share Internet access, or enable a community WiFi network. The slice could be managed by a local or remote control plane—perhaps managed remotely by a third party—for AAA service (association, authentication and accounting). The slice would need bandwidth isolation to prevent hogging of the WiFi or uplink bandwidth, and traffic isolation would be needed to protect the home network from attack. The home owner could choose to create multiple slices, one for friends, and rent others to 3rd party service providers. Again, one slicing mechanism could support a variety of business models and policies.

2.2 Outsourcing Network Management

Home networks are getting more complicated. In our homes, we attach a growing variety of devices (smartphones, laptops, TVs, heating controllers, loudspeakers, sensors etc), use applications with diverse needs (video streaming, browsing, VoIP, etc), and then operate them over unreliable WiFi networks. It is hardly surprising that many consumers are confused and frustrated when trying to manage their home network, and trying to make it secure. While managing such a network is not inherently difficult for a sophisticated network administrator, the typical home user lacks the technical know-how and desire to do so. It seems likely that – if they could – many would opt to delegate and out-source the configuration and management to a trusted third-party expert. Inspired by recent work on outsourcing home network security [10], we envision users delegating control of a slice of their home network, or even the entire physical network. A network management company might remotely set power levels and channels to reduce interference, suggest AP location, provide parental controls, set firewall rules, set the right priorities and QoS levels for a TV set or VoIP call, and so on.

A "one-stop-shop" management company might offer to manage the entire physical home network, including configuring network devices inside the home, controlling their day-to-day operation, and managing upgrades. The management company might manage slicing, too. For example, they could slice the network for the home-owner, creating slices for different applications, such as cellular offload and guest WiFi. They might also manage, and collect revenue from a utility company that was sharing the cost of the network. Alternatively, some home-owners will choose to manage the physical network themselves, and create slices for different applications. The home-owner then cedes control of a slice to an application provider, who then provides a dedicated control plane to manage traffic and bandwidth within its slice. By outsourcing control we can decouple high level, user-defined policies from the actual implementation, hiding technical details from the user.

2.3 Customizing Slices for Applications

Slicing can enable a new model of application delivery to the home, where an application provider can obtain a slice of the home network with certain resource reservations and the ability to modify it to meet an application's requirements. For example, an application provider such as Netflix could obtain a slice with sufficient bandwidth guarantees to support the high datarates of the video stream. Further, Netflix could customize the slice to handle packet loss and caching in a way that is optimized for video delivery, to ensure a smooth and high quality user experience. Such a model is in contrast to the current model, where applications are passive participants and are at the mercy of the network. Infrastructure owners can leverage this capability to further subsidize the cost of network deployment and operation from application providers.

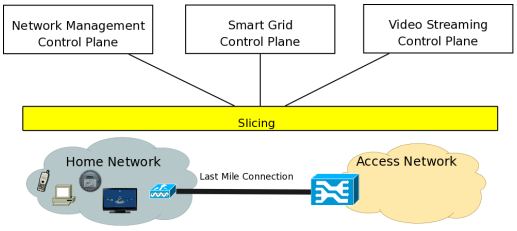


Figure 1: Slicing on a Home Network : Multiple service providers share the same infrastructure. Each provider controls a given slice, and a slicing layer enforces isolation between different slices.

Extending the slice within the home network, the content provider can now provision video delivery in an end-to-end manner, i.e through the last-mile link and up to the end-device playing-back the video. Given the ability to modify how forwarding takes place, the application provider can implement his own routing mechanism, use replication and retransmission methods to optimize video streaming over wireless, minimize latency for user interaction (streaming control operations like video seeking and rewind), or even use special congestion control protocols like WCP to improve the user experience [18].

To enable such new models of application delivery, the slicing mechanism has to support both bandwidth and traffic isolation, as well as allow the slice to be modified and customized independently by each provider.

3. DESIGN

3.1 Overview

Motivated by our requirements for slicing, we sketch out a potential design for slicing home networks. We leverage recent work on FlowVisor, a slicing mechanism for OpenFlow-based networks [17].

Fig. 1 shows the basic architecture of our sliced system. The network substrate consists of network elements (e.g. Wireless Access Point at home, Ethernet switch in the Access Network). The network elements are *programmable*, in the sense that they can be remotely controlled, allowing us to modify and customize the network behavior. The network elements connect to multiple providers through a slicing layer. Each slice is independent of the rest, and defined by (1) reserved network resources (e.g. bandwidth, forwarding table entries, router CPU), (2) the traffic it can carry, and (3) control logic that defines how packets are controlled and routed in the network.

3.2 Programmable Network Elements

Programmable network elements are needed both for building the slicing mechanism itself, as well as providing the flex-

ibility on top. Ability to *control and modify* a slice should include arbitrary forwarding policies, mapping flows to queues, configuration of wireless parameters etc. In our example of outsourcing network management, the provider needs to block malicious traffic, or give VoIP calls priority over P2P traffic. Similarly, a smart-grid provider might need to connect metered equipment to an encrypted wireless network. And a video content provider might want to mark data to implement a customized congestion control mechanism. A *low-level network API* is therefore essential for our design.

The control logic that specifies a slice's behavior can be flexibly placed within the network equipment, or in a controller either at home or at the provider's premises. The decoupling allows different slicing models as we describe next.

3.3 Slicing Layer

The slicing layer enforces isolation between different slices so they can safely coexist. The slicing layer sits on top of the programmable network elements, and orchestrates the independent control of each slice by a slice-specific control plane. Our current design provides the four different isolation guarantees we sketched out earlier, namely (i) *isolation of bandwidth*, (ii) *isolation of traffic*, (iii) *isolation of control* and (iv) *ability to independently modify*. However, in practice different applications may need only a particular subset of the four isolation guarantees above.

To isolate traffic, a subset of all traffic is allocated to each slice. For example, all video traffic from Amazon might be under the control of one slice; all smart-meter traffic part of another; and all remaining traffic may be part of a default slice. The slicing mechanism must ensure that a slice's control plane can only control traffic belonging to its slice.

To isolate bandwidth between slices, every networking component must provide the means for bandwidth isolation. A guest WiFi user should not be able to affect the home user's experience by exhausting the available bandwidth. In the packet world this implies queueing, scheduling or policing; or using wavelengths or frequency bands, if the last-mile sharing takes place at a lower layer. Other network resources (e.g. router CPU and forwarding table entries) should also be divided and isolated.

To isolate control, the slicing layer intercepts messages between the control logic of each slice and the underlying network substrate. Each slice should be able to control only its related traffic, using its available network resources.

Note that we have intentionally left the definition of what a slice is unspecified. In our architecture, a slice may consist of a single slice, a combination of multiple slices, or even a subdivision (or delegation) of an existing slice. Such flexibility is needed to accommodate different policies with regards to privacy, ownership and control of the infrastructure. Fig. 2 highlights an instance of the suggested design where a home network subscribes to multiple providers using different policies.

Finally, while the slicing layer provides the basic scaffold on top of which several providers can coexist, additional application specific functional blocks may be needed. For example, to enable "advertisement" of and subscription to available services, a service broker may translate high-level policies to lower-level network semantics, define the division of duty among slices and detect potential conflicts. To address privacy concerns while outsourcing control of the network, it might be necessary to aggregate and/or anonymize

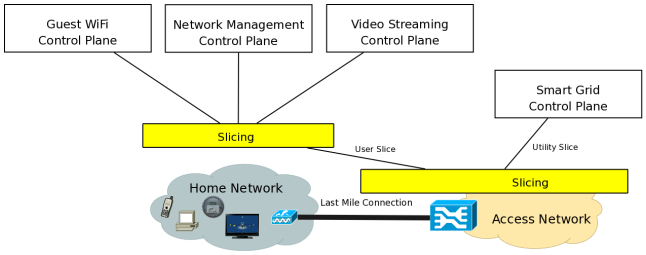


Figure 2: Multiple Service Providers on a Sliced Home Network: The access network owner gives a slice of the last-mile infrastructure to the user, and rents another one to a utility company. The user trusts a network management company to operate his network for security, configuration, and basic QoS management. He delegates resources and control for video traffic to a content provider. He shares a small portion of his network with guests and neighbors, outsourcing AAA to a GuestWiFi service.

data before we send them to the controller without hindering its usefulness. We believe our slicing design is sufficiently extensible to accommodate such additions.

4. PROTOTYPE AND DEPLOYMENT

To better understand how to slice home networks, and to gain practical experience, we created a prototype sliced home network, and deployed it in several homes. To date, we manage seven home networks, but are steadily increasing the population of users. In this section, we describe our prototype. An immediate goal is to implement and experiment with interesting applications on top–allowing us to present practical functional examples on top of slicing.

Our prototype exploits prior work on OpenFlow [15]—an open API to remotely control forwarding in network elements, and FlowVisor [17], a slicing mechanism for OpenFlow networks. OpenFlow separates control logic from the datapath of the network. A remote controller defines the forwarding logic of a network switch (data plane) by installing flow entries. A flow entry consists of a match, a set of actions, and a set of counters. The switch looks-up incoming packets against installed flow entries, and performs the corresponding actions. If no match is found, an event is sent to the controller, which in turn decides how to forward the packet and caches this decision as a flow entry into the switch. FlowVisor slices an OpenFlow network by policing and proxying OpenFlow control messages, allowing it to *isolate traffic and bandwidth* (together with queuing in the datapath).

Each OpenFlow controller (e.g., NOX [12]) runs on top of the FlowVisor, and is given *independent, programmatic control* of its slice. OpenFlow provides the low-level control so that slices can *be customized*, to some extent. We use SNMP to configure the wireless access-points (SSID, WiFi encryption, queues, etc) [13]. To interoperate with NATs and firewalls typical in the home environment, we run SNMP over a TCP tunnel (i.e., UDP-in-TCP tunneling).

For our prototype we ported OpenFlow 1.0 to OpenWrt, a Linux distribution for low-cost home WiFi routers. Using low-cost home routers allows us to deploy more of them; and allows us to explore how a simple AP can be controlled

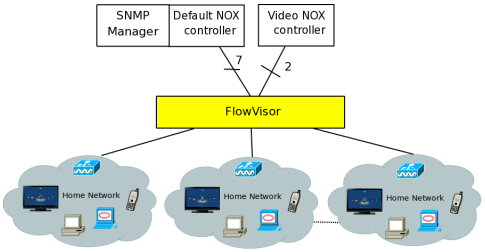


Figure 3: Prototype Deployment. Seven homes connect to a default controller through FlowVisor. A separate controller manages traffic from video set-top boxes.

by a sophisticated control plane. This is in contrast to the more capable, and more expensive boxes used in other deployments [7]—for which slicing would work just as well.

In our initial deployments, we make use of the TP-Link WR1043ND AP, with 400 MHz CPU, 16 MB flash memory, 4 Ethernet ports and an Atheros 802.11n WiFi chipset. The OpenWrt driver for the Atheros chipset supports multiple SSID on a single radio interface. The APs are deployed in seven homes (three on campus and four off campus), with different broadband providers (campus Ethernet, DSL, cable and WiMAX). Inside the homes are a variety of laptops, desktops, smart-phones, game consoles, and video setup-boxes. All homes have a default slice which is managed by a controller that runs at Stanford. The controller is implemented using NOX, and an SNMP manager to configure the APs, and each slice connects to the controller through FlowVisor (Fig. 3).

To experiment with slicing, we create a "video slice" for all video devices. In our deployment, these include GoogleTV and Netflix boxes. All of the traffic in the video slice (defined as all the traffic to and from the MAC addresses of these devices) is controlled by a separate controller. To control the video slice, we run a dedicated video controller at Stanford, also based on NOX. Currently this slice is configured manually, but we plan to create a web-based platform—to configure the FlowVisor via XML-RPC—so that home owners can configure their slices more easily.

At the time of writing, our deployment is in its infancy, and we have run it for one month. Over time, we aim to build practical experience slicing and controling home networks. Moving forward, we will enhance the slice controllers (video and network management), integrate an existing GuestWiFi application, and expand our population of users. So far, our deployments do not allow us to control the last-mile infrastructure. However, we are exploring several alternatives that would allow slicing back into the broadband network.

4.1 The Cost of Slicing

A natural question about slicing is "what is its overhead"? While this has been previously quantified for the campus network [17], we iterate and interpret the results in the context of home networks.

From the user's experience perspective, remote control of the network and more specifically RTT between the home router and the FlowVisor/controller seems to dominate performance penalties. In our deployment, the median RTT between the controller and the homes is 16 ms (with CDF of latency shown in Fig. 4(b)). It's expected that "proximity" between the home and FlowVisor/Controller should be considered through slicing and control delegation.

A significant portion of the traffic in home networks can be proactively managed ((Fig. 4(a))—with about 20% being DNS and ARP, reducing both the initial delay a new flow suffers and the load on FlowVisor and controllers.

In terms of scalability, FlowVisor scales linearly to the network of switches and number of flows. In our deployment of 7 homes where a single slicing layer exists, FlowVisor is running with an average load of 4.07 flows per second with a maximum load of 331 flows per second (and CDF is shown in Fig. 4(c)). This means we can expect to scale up our deployments by over 200 times with our current setup. In the context of home networks, both FlowVisor and OpenFlow controllers can easily perform better by demultiplexing matching of slice rules and network state based on the individual residency—which will allow for efficient load-balancing.

A detailed description of how to optimize slicing and control for the home networks is beyond the scope of this paper.

5. RELATED WORK

Slicing has been proposed by Sherwood et al [17] to enable networking experiments on a production network. In our prototype we heavily borrow ideas and implementation efforts from this prior work.

Others have previously identified the need for applications and services within the home to cope with increasing complexity and heterogeneity. Dixon et al suggested an operating system for the home (HomeOS and app store) in which users deal with applications and high-level policies to deal with integration and management of their network [9]. In [19] the authors use an OSGI-based framework to install applications on a residential gateway.

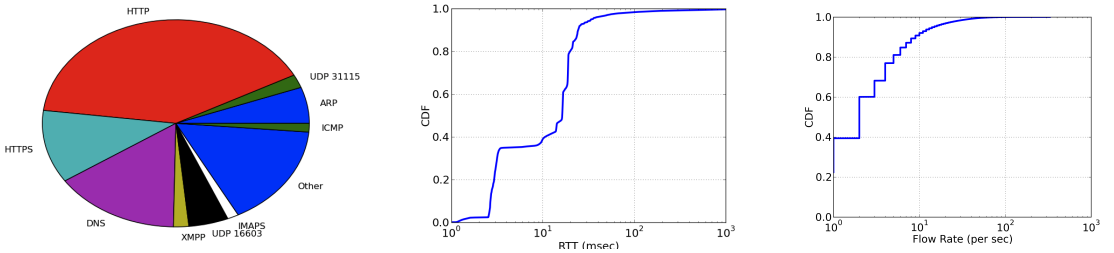
Feamster proposed outsourcing of network security to an offsite controller that can both detect Internet-wide coordinated activity coming from homes and automatically take corrective action on behalf of unskilled home users. The architecture suggested in this paper is the closest to a slice of our design. Other applications such as 3G/WiFi offloading [5, 14] and video streaming over wireless [20] have been explored in their own context.

Sharing of the last-mile infrastructure has been mostly discussed under the terms of multiple ISPs [1]. The most popular way today is to terminate the link on ISP-specific data-link layer equipment (e.g. DSLAM, OLT), while others suggest to share the physical medium or enforce separation in the network layer [6, 16]. This gives users more options, but there is still a single provider present at the home.

Significant work has been done to automate detection and diagnosis of faults in home networks, and to define the appropriate interaction and interfaces between the users and tools to manage and configure the home network [4, 7, 8, 11]. We seek to build intuitive and meaningful interfaces for the user to subscribe and configure new services, and we plan to leverage on existing work in this field.

6. CONCLUSION

The home network faces growing challenges. It is the meeting point for home automation, smart-grid, seamless connectivity, online entertainment, ubiquitous access to data; all of which require a working home network, customized for



(a) Traffic Breakdown

(b) CDF of RTT between controller and homes

(c) CDF of Flow Request Rate

Figure 4: Measurements from current deployments in seven homes

their needs. Our long-term goal is the architecture of home networks to allow all of these new applications; plus the ability to evolve and improve the home network to support new applications as-yet unthought of. We propose slicing as a starting point, and our prototype allows us to experiment with different policies and applications. Through this exercise, we hope to gain a better understanding of the needs and trade-offs between performance, privacy, security and flexibility. Moving forward we want to make our prototype platform available to the community and so others can build upon our work.

Acknowledgment

The authors would like to thank Julius Schulz-Zander, Jiang Zhu and Bobby Holley for their initial port of OpenFlow to OpenWrt. We would also like to thank Te-Yuan Huang, Rob Sherwood and the anonymous reviewers for comments and suggestions.

7. REFERENCES

[1] Telecommunications act of 1996, pub. la. no. 104-104, 110 stat. 56 (1996).

[2] Verizon Fios Press Release. `http://investor.verizon.com/news/view.aspx?NewsID=773`.

[3] Accenture. Big trouble with no trouble found: How consumer electronics firms confront the high cost of customer returns, 2008.

[4] B. Aggarwal, R. Bhagwan, T. Das, S. Eswaran, V. N. Padmanabhan, and G. M. Voelker. Netprints: diagnosing home network misconfigurations using shared knowledge. In *NSDI '09*, pages 349–364, Berkeley, CA, USA, 2009. USENIX Association.

[5] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting mobile 3g using wifi. In *MobiSys '10*, pages 209–222, New York, NY, USA, 2010. ACM.

[6] A. Banerjee and M. Sirbu. *Towards a Technologically and Competitively Neutral Fiber-to-the-Home (FTTH) Infrastructure*, pages 119–139. John Wiley & Sons, Ltd, 2005.

[7] K. L. Calvert, W. K. Edwards, N. Feamster, R. E. Grinter, Y. Deng, and X. Zhou. Instrumenting home networks. *SIGCOMM CCR*, 41:84–89, January 2011.

[8] M. Chetty, R. Banks, R. Harper, T. Regan, A. Sellen, C. Gkantsidis, T. Karagiannis, and P. Key. Who's hogging the bandwidth: the consequences of revealing the invisible in the home. In *CHI '10*, pages 659–668, New York, NY, USA, 2010. ACM.

[9] C. Dixon, R. Mahajan, S. Agarwal, A. J. Brush, B. Lee, S. Saroiu, and V. Bahl. The home needs an operating system (and an app store). In *Hotnets '10*, pages 18:1–18:6, New York, NY, USA, 2010. ACM.

[10] N. Feamster. Outsourcing home network security. In *HomeNets '10*, pages 37–42, New York, NY, USA, 2010. ACM.

[11] R. E. Grinter, W. K. Edwards, M. W. Newman, and N. Ducheneaut. The work to make a home network work. In H. Gellersen, K. Schmidt, M. Beaudouin-Lafon, and W. Mackay, editors, *ECSCW 2005*, pages 469–488. Springer Netherlands, 2005.

[12] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: Towards and operating system for networks. In *ACM SIGCOMM CCR*, July 2008.

[13] K.-K. Yap, et. al. Blueprint for introducing innovation into wireless mobile networks. In *VISA '10*, pages 25–32, New York, NY, USA, 2010. ACM.

[14] K. Lee, I. Rhee, J. Lee, S. Chong, and Y. Yi. Mobile data offloading: how much can wifi deliver? In *Co-NEXT '10*, pages 26:1–26:12, New York, NY, USA, 2010. ACM.

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

[16] O'Donnell, Shawn. Broadband Architectures, ISP Business Plans, and Open Access. `http://dspace.mit.edu/handle/1721.1/1513`.

[17] R. Sherwood, et. al. Can the production network be the testbed? In *OSDI'10*, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association.

[18] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan. Understanding congestion control in multi-hop wireless mesh networks. In *MobiCom '08*, pages 291–302, New York, NY, USA, 2008. ACM.

[19] D. Valtchev and I. Frankov. Service gateway architecture for a smart home. *Communications Magazine, IEEE*, 40(4):126 –132, Apr. 2002.

[20] X. Zhu and B. Girod. Distributed media-aware rate allocation for wireless video streaming. *IEEE TCSVT*, 20(11):1462 –1474, 2010.