Putting Home Users in Charge of their Network

Yiannis Yiakoumis Te-Yuan Huang Kok-Kiong Yap
Sachin Katti Nick McKeown Ramesh Johari
Stanford University
{yiannisy,huangty,yapkke,skatti,nickm,rameshj}@stanford.edu

ABSTRACT

Policy-makers, ISPs and content providers are locked in a debate about who can control the Internet traffic that flows into our homes. In this paper we argue that the user, not the ISP or the content provider, should decide how traffic is prioritized to and from the home. Home users know most about their preferences, and if they can express them well to the ISP, then both the ISP and user are better off. To test the idea we built a prototype that lets users express high-level preferences that are translated to low-level semantics and used to control the network.

Categories and Subject Descriptors

C.2.0 [**Computer Systems Organization**]: Computer-Communication Networks—*General*

General Terms

Design, Economics, Human Factors, Management

Keywords

Network Slicing, QoS, Video-Streaming, Human Computer Interaction, Home Networks

1. INTRODUCTION

Home networks are an essential part of the modern household, and as the number of connected devices and applications grow, we grow more dependent on the quality of our home Internet connection. We expect the network to be fast, always-on, reliable and responsive.

Simply, we do not want the network to stand in the way of the applications we use at home. We place increasing demands on our home network, with video streaming, video chat, VoIP, gaming and cloud-based backup now being commonplace. In the past 12 months the amount of traffic on home networks has increased by 50% (from 7.0 GB per month to 10.3 GB). Our applications vie with each other for the last-mile connection, often causing congestion and bad user experience [4, 14].

Despite making great strides in the bandwidth delivered to the home, Internet service providers (ISPs, offering cable or DSL) still struggle with how to share the available bandwidth among users' applications. Recently, many ISPs have introduced controversial data cap plans [18, 8] and blocked traffic-intensive applications [1], sparking intense debate [2] and fueling fears that ISPs will try to constrain the applications we use.

In this paper we propose turning the debate on its head by placing the home user firmly in control, not the ISP. If the debate is about ISPs making bad decisions about which traffic to prioritize on our behalf, then let's place the decision in the hands of the user, who is the only one to know which applications they prefer. Instead of the current "one size fits all' approach, we propose allowing the user to choose the relative priority of their applications, and indicate their preference to the ISP, who then enforces the preference. We could use existing methods, such as RSVP [25], but we can go one step further and exploit recent trends in networking that make it even easier for ISPs to have more programmatic control over their networks [20], therefore making it easier for the ISP to implement the user's desire.

Therefore, in this paper we advocate that *user choice should guide the management of network traffic*—not only inside the home but also within the ISP.

We argue that user-driven network provisioning will not only better reflect the user's preferences, but will also improve the user-experience, and hence the retention rate of ISP customers—it is in the ISP's interest too. For example, consider a home user watching a streamed video, who is experiencing low bit-rate because of contention with an online backup application. If the ISP wants to best serve their customer, they need to: (1) Detect the video traffic, (2) Realize the user would prefer a higher bit-rate — this is not obvious, because the video service will deliberately pick a low and uncongested bit-rate; (3) Pick a rate, and provision the service accordingly.

If instead the user can simply express "My Netflix videos should stream at HD quality", or "I want Skype video calls to my mother to have the highest quality", then the home network and the ISP can better meet the user's needs and give them the service they want.

It will take two innovations to make such a service possible. First, we need a very easy to use and intuitive agent that allows non-technical users to express their choices (or to pick a profile), and translate them to network semantics (bit-rate, low-latency, priority, *etc.*). Second, the user preference (in network semantics) needs to be communicated to the ISP through a simple and stable abstraction, which in turn needs to be implemented in the network datapath.

UbiComp'12, September 5-8, 2012, Pittsburgh, USA.

In the simplest case, we could merely provision network bandwidth among different applications, which can be implemented by simply setting the weights on a Weighted-Fair-Queueing (WFQ) scheduler [9, 23] in the routers along the path. We propose going further and *slicing the control* of the network among applications, as well as the bandwidth, to make it easier to introduce innovative ways to control each service. For example, a video streaming service provider could provide new tools to improve the quality of routing from the ISP to the TV set in the home, even in the presence of wireless interference. Therefore we build on our previous work on slicing home networks [30] in our evaluation.

In the rest of the paper, we describe a design for a home network where users specify their choices and signal them to the ISP. We consider the division of labor between users and ISPs, and describe *user-agents*, which translate high-level user intentions to low-level network semantics (§2). We built and evaluated a prototype of the user-agent, and the control plane for the ISP (§3). We describe related work (§4) and conclude with a broader discussion of the pros and cons of our approach (§5).

2. DESIGN

Network control starts with the expression of a high-level intent from the user and completes with the appropriate configuration of the network to enforce the desired behavior. To divide the necessary work between the user and the ISP, we follow a simple principle.

The user should define ***which*** *traffic gets* ***what*** *type of service, and* ***when*** *this happens; while the ISP figures out* ***how*** *and* ***where*** *in the network, provisioning is implemented.*

2.1 Architecture

2.1.1 Intents and User Agents

Our suggestion to give control to the users might sound counter-intuitive. Many argue that home users have neither the expertise nor the interest to manage the network. It really shouldn't be their responsibility! However, the user has unique knowledge of the context beyond what the network (or an application) can deduce. He knows which applications are useful for him and whether they should be prioritized by the network. A user might be willing to watch HD movie from a specific content provider, but not for any video in the web. Sometimes, his desires might seem unreasonable as long-term policies but make sense under the current context. For example, an upload is typically considered background traffic. However, a student uploading a large submission for an impending deadline considers minimizing the flow completion time of that upload the most important task of the network.

To act upon the user's intents, we need to translate them into quantitative network semantics. For this, we introduce the notion of user agents. User agents bridge the gap between what users experience (e.g. frustration for bad-video experience), and what is conveyed to the network (e.g. reserve more bandwidth).

Intents, preferences, and comprehension vary wildly and are likely to change over time, and so we want user agents to be flexible, easy to install and use, and rapidly evolvable. Different parameters can be taken into account while designing a user agent. What is the best place to capture user intents (e.g. application, host or network-wide)? How does it interact with the network (e.g. statically or dynamically)? How does it interact with the user? Is it interactive or does it implicitly infer what to do from the broader context and past experience? Should it seek user's feedback to further finetune actions? How can it detect the targeted traffic and required network functionality?

We believe that there is no one-size-fits-all answer to these questions. In the next section, we explore the design space for user-agents by demonstrating three use-cases which vary in (i) where the agent is located, (ii) how it interacts with the network, (iii) how it interacts with the user. Further exploring and understanding of options and trade-offs in this area is important and requires work in the crossings of HCI and networking.

2.1.2 User-ISP Interface

In order to convey user preferences to the infrastructure we need an interface between the user (or his agents) and the ISP. We expect this to be generic and stable so that a plethora of agents can be developed on top of it.

The basic primitive exposed by the interface is a mapping between network traffic and network properties. This interface is application and content-agnostic, allowing users to prioritize any traffic they want and also facilitating the rapid integration of new types of applications (such as cloud-based backup). By providing application-agnostic interface, we shift the task of traffic characterization from the network to the user agent. As content delivery networks and load-balancers make network address space ephemeral we need more fine-grained and dynamic coordination. As a result, address-based application characterization becomes harder for ISPs; on the other hand, user agents can exploit the broader context to characterize the traffic.

Different network properties can be provided by the ISP, such as bit-rate, low-latency, low-loss, or combinations of them. These properties can be provided in various degree of granularity by the ISP. The user agent can then flexibly and dynamically manage the network for the user.

Following our suggestion for richer user-control beyond network provisioning, the user can also influence how the traffic is routed, by asking the ISP to delegate control to a third provider. This can enable future innovation where application providers can optimize the network for their specific needs in order to better serve the users.

Besides the basic traffic-service mapping, the interface should support options like discovery and authentication mechanism, query statistics, error reporting etc. The details of implementing this interface (e.g. in-band vs out-of-band, exact API calls, etc) are beyond the scope of this paper.

Upon accepting requests from an agent, the ISP needs to perform the necessary accounting and admission control to ensure that the request is valid and realizable given the current network state. Further coordination (e.g. through user-feedback or predefined priorities) might be necessary when different agents have conflicting interests. Part of this functionality can be placed within the home network itself to mediate requests originating from competing users or applications and their agents.

2.1.3 Infrastructure

The main role of the infrastructure is to enforce the specified provisioning and control. Besides the low-level mecha-

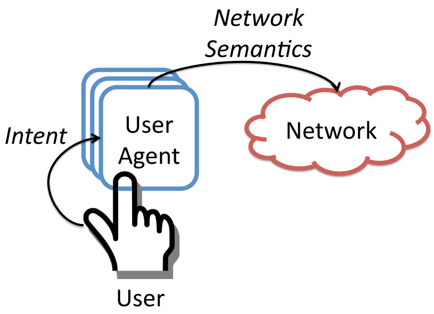


Figure 1: Basic Workflow. Users express their intents to user-agents which translate them to network-semantics and forward them to the ISP. The ISP performs appropriate accounting and enforces the desired provisioning in the network.

nisms for that (e.g. WFQ), network-wide co-ordination will probably be needed to provide the desired services. Existing mechanisms—such as reservation protocols (RSVP), or marking specific bits on a packet's header (e.g. DSCP)—could be used. An ISP can decide where provisioning should take place; it could be limited to the last-mile connection, or all the way from the border router to the home Access-Point (AP).

Following trends in networking [20], the infrastructure can expose richer functionality beyond basic provisioning, to allow further optimization. For example, routing for a video stream could be optimized to achieve better quality, or support multicast to stream to multiple devices. In previous work we suggested slicing as a way towards this direction. A slice consists of an isolated set of resources and network traffic. By mapping traffic to the appropriate slice we can ensure desired properties, according to the provisioned resources. More important, a slice comes with its own control, allowing further customization of the network for this traffic based on the needs of each application. Control primitives are delegated to a remote controller per-slice, allowing multiple slices to co-exist under the arbitration of the user. We refer the reader to [26] for a better discussion of the slicing mechanism and possible applications.

2.2 Basic Workflow

Armed with our basic design, we outline the skeletal workflow for a user-driven network, reinforcing the interactions between the various pieces (as shown in Figure 1). More specifically, we will consider the basic steps needed to convert a high-level user intent to an enforceable low-level network configuration.

1. The network provisioning process begins with an expression of intent by the user. This intent is captured on a user-facing device, where its target might be inferred on by the user agent. The user target would then convert this high-level intent into low-level network semantics. For example, a request to "boost" a video stream can be translated into a provisioning for guaranteed bandwidth. The resulting low-level network semantics must now be communicated to the ISP where it is implemented or enforced.

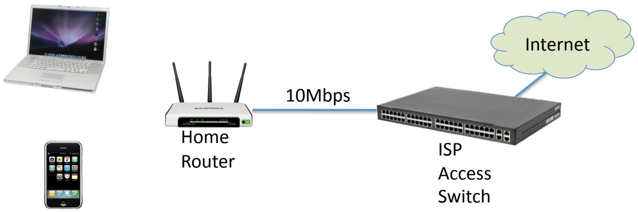


Figure 2: Prototype Setup

2. Once the request is received, the ISP has to mediate requests from different users, perform necessary accounting, checks and balances. This is necessary to ensure the request is valid and the network is not overloaded—which is keystone to the ISP's capability to maintain a functional network at all times. In this process, the ISP may also provide feedback on whatever the intent is admitted or denied, and possibly suggest alternatives to a denied intent.

3. Once the intent is admitted into the system, this intent is then enforced throughout the network. This can be done through low-level network provisioning at one or more points in the network.

3. PROTOTYPE AND EVALUATION

We now describe a prototype that showcases user-driven network control. The goal of the prototype is to demonstrate the technical feasibility of the proposal and present our exploratory foray of the design space. We focus mostly on user-agents and how to capture user intents, and refer the reader to [26] for details on the network slicing mechanism.

In this prototype, we recreated a minimal user-ISP infrastructure and showcase how users can drive network provisioning to meet their needs. We focus on user intents and how they get translated to network semantics. More specifically we show (i) a web-based agent for static provisioning; (ii) an agent integrated with Skype for low-latency VoIP (Skype+); and (iii) an interactive agent for on-demand bandwidth for video-streaming (MyBoost). Our agents differ in the following dimensions: where are they implemented (network-wide, browser extension or application-embedded), how they interact with the network (static or dynamic), and whether they dynamically interact with the user.

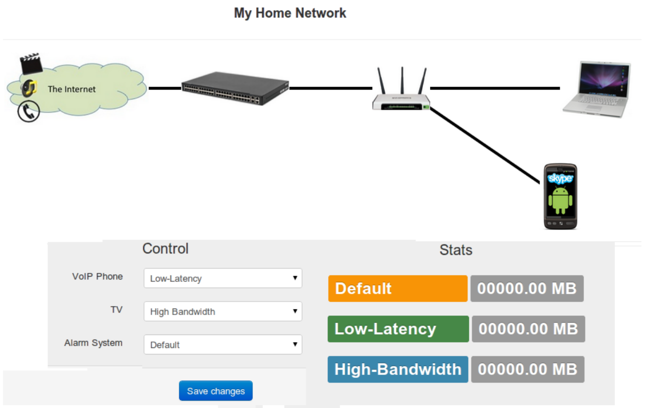
We evaluate our prototype, focusing on how user intents can improve user experience. We use widely available applications for video streaming and VoIP, and evaluate user-experience through metrics highly correlated to it, namely video playout bitrate and latency respectively. More specifically we study the impact of Skype+ and MyBoost on a Skype call and a movie from Netflix respectively.[1] We introduce background traffic using Dropbox, an HTTP download and another video stream, and examined how the above-mentioned applications behave before and after expression of user intents.

3.1 Infrastructure

Our infrastructure represents a minimal user-ISP setup (Fig. 2). The home network consists of an access point (TP-

---

[1] We do not show results from the web-based agent as they are similar with the other two.



VoIP Phone: Low-Latency
TV: High Bandwidth
Alarm System: Default

Default: 00000.00 MB
Low-Latency: 00000.00 MB
High-Bandwidth: 00000.00 MB

Figure 3: Web-based management console where users can statically provision their network.

Link WR1043ND) and a set of devices (laptops and phones) that run multiple applications. This network in turn connects to the Internet through a 48-port Pronto switch (the ISP's edge switch) via a 10 Mb/s link, which acts as the last-mile connection. In our prototype, provisioning takes place at the last-mile; this is clearly not representative of how and where an ISP could implement network provisioning but suffice for our demonstrations. We use minimum-rate queues to enforce low-latency and/or high-bandwidth. We define three different services: (i)best-effort, (ii) a 5 Mb/s guaranteed bandwidth service, and (iii) a 500 kb/s low-latency service. User-agents communicate with the infrastructure using an out-of-band messaging scheme, through which they define a set of flow to service mapping. On the infrastructure side, the messages are received by an OpenFlow controller which enforces the mappings to appropriate queues.

3.2 User-Agents

3.2.1 *Static Network-Wide Provisioning*

We implemented a web-based management console where the user can perform static network provisioning. Fig. 3 shows a snapshot of the interface. Using the console a user can ensure that a particular device (e.g. a TV) gets good quality service. He can directly select the device and map it to the high-bandwidth service. Depending on the setup, the device can be identified by its MAC/IP address which are communicated to the ISP. In our case we use MAC address to identify traffic coming from/destined to our device of preference, and all this traffic gets mapped to the appropriate queue. In cases where NAT is present, finer-granularity co-ordination might be needed.

3.2.2 *Skype+: Dynamic App-based Provisioning*

Real-time communications like VoIP can suffer from high-latency. This is not unusual, often caused by bloated buffers in the network [14]. How could we provision our network to eliminate high-latency symptoms for an application like Skype? Skype works using a combination of protocols and methods (UDP/TCP, direct peer-to-peer/supernodes overlay, etc), and therefore statically configuring the network is not an option. To that end, we implemented Skype+ as a simple extension to Skype. Skype+ integrates a user-agent which dynamically co-ordinates with the network. Whenever a new call is initiated, Skype+ asks the network to map the specific flow to the low-latency service. In doing so, it can use the user's credentials to authenticate with the network provider.

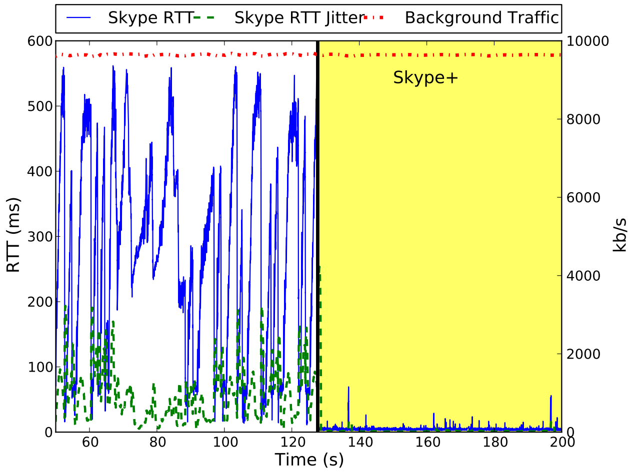


Figure 4: Skype+ achieves low-latency under high background load.

Fig. 4 shows how Skype+ can eliminate high-latency problems during a call. As shown in the figure, due to background traffic Skype suffers from high-latency. TCP flows fill up the buffer at the bottleneck link leading to high latency and jitter (median of 280 ms and 50 ms respectively). When enabling Skype+, the quality of the call dramatically improves and latency median goes down to 4 ms (with 1 ms jitter). Note that since Skype produces low-volume traffic, the overall bandwidth used by the other applications is not affected.

3.2.3 *MyBoost: On-Demand Bandwidth*

The need for network provisioning might not always be inferred or predictable. Think of a video stream which plays on a low bitrate, preventing the user from enjoying a movie. Input from the user can be extremely useful. An interactive user-agent can help the user to simply express his frustration and act upon that. To show the idea, we developed MyBoost, a single-button browser extension which allows users to interactively ask for more bandwidth. When the user clicks the button, MyBoost detects all traffic generated within the current browser tab, and then asks the ISP to route this traffic through the high-bandwidth service.

We implemented MyBoost during a 48-hour coding competition and we encourage the reader to look at the relevant demonstration [3]. In our view it highlights the potential value of decoupling network semantics from user-intents, bringing the latter closer to the users and developers.

Fig. 5 highlights the improvement that MyBoost brings to a Netflix stream. To operate in different network conditions, Netflix adapts its video rate depending on the available bandwidth. The video rate ranges from 3.6 Mb/s (High-Definition) down to 235 Kb/s. From the figure we can see that at the beginning—without background traffic—Netflix streams HD video at 3.6 Mb/s. When background traffic is introduced it cannot continue streaming HD and the video rate goes down to 560 kb/s at lowest. Most of the band-

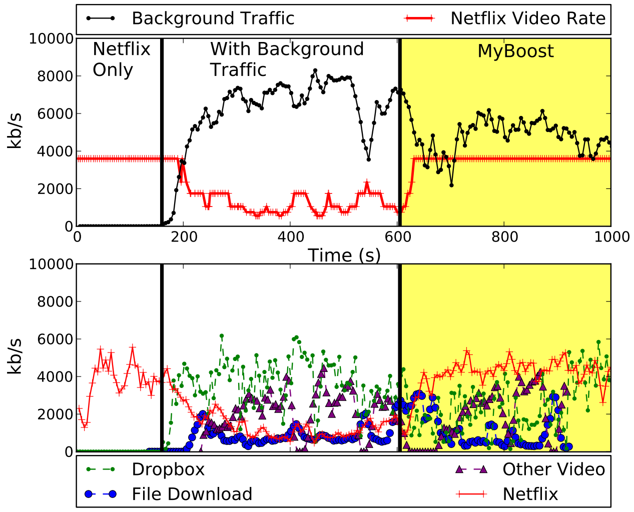


Figure 5: On-Demand bandwidth reservation. After using MyBoost the video stream returns to High-Definition.

width is consumed by applications which are not critical for the user, as the bottom figure in Fig. 5 shows. The user expresses his frustration using the MyBoost button which provides Netflix with sufficient bandwidth. Shortly after, the movie goes back to HD streaming, while the rest of the available bandwidth is shared among the remaining applications.

4. RELATED WORK

There has been a lot of interest recently for home networks, both in terms of performance and interaction with the user. [15, 27] focused on performance issues of current broadband access links. Other researchers looked into the details of traffic engineering—such as PowerBoost—deployed by ISPs [5, 17], or traffic characteristics in home networks [19, 24]. The studies show that there is a huge diversity between users, and while each ISP optimizes different performance metrics, it's hard to find one ISP that best fits all users.

Various other proposals, both from the system networking and HCI perspectives, focus on making network management an easier task for the user. HomeOS [12] provides a centralized and programmable platform to simplify device management in the home. Mortier et al. [21] proposed interactive network mangement by redesigning home routers to provide an interactive interface to the users. [30] proposed a combination of slicing and outsourcing to allow users control their network using functionality from third-party providers. There is also other work on automated problem diagnosis in home networks [10, 11]. Beyond home networks, there are also many efforts on understanding network performance on user experience for various applications, such as video streaming [13], VoIP [6] and online gaming [7]. All the above highlight a drive towards user-centric home networks—providing better user experience and promoting more intuitive user interaction.

Our approach has been motivated by this trend. Our work differs from previous proposals in that we focus on the user-ISP interaction and ways to make it more dynamic and useful for the user. Our focus on high-level intents, looks beyond abstracting existing network policies, suggesting new type of services such as short-term, on-demand bandwidth.

Our network provisioning assumptions build on several existing ideas—from the more classical proposals like RSVP [25] to more recent proposals like Serval [22], SFNet [29] and intentional networking [16]. Our focus on how to integrate user preferences to network provisioning is orthogonal to the actual mechanism used to request resources.

The application and type agnostic interface between user and ISP has been suggested in [28] which examines in detail the trade-offs for different choices.

5. DISCUSSION

Home networks served as the starting point for our proposition of user-driven networks. Even though we looked at the interaction between users and ISPs, the same mechanism could be used to configure the home network itself. We believe that the idea is more generally applicable to other contexts. For example, mobile networks presents very similar circumstances, where user intent can inform network provisioning within the mobile carrier's network. An in-depth exploration in different contexts will be left to future work.

User-driven control requires fine-grained management of the network. We believe this is feasible. ISP networks already maintain per-user state for rate limiting and accounting, and protocols for cable and 4G wireless networks (like DOCSIS and LTE) already support Quality of Service (QoS) primitives. While the edge of the network seems more appropriate to enforce fine-granularity policies, ISPs can then map flows to appropriate traffic groups to maintain the provisioning as traffic goes deeper inside the network.

In the meantime, trends like Software-Defined Networking (SDN) bring in a more programmable network, with which more sophisticated network management tools can be built.

Network-provisioning in access networks is hotly debated in net-neutrality arguments. By placing control back to the users, we can sidestep the concerns raised. Further we believe that our design gives ISPs the opportunity to differentiate by supporting a wide range of network abstractions for users (and their agents).

Opening a feedback loop between users and network administrators can provide new insights for network management. Currently, users use the networks, and administrators configure it according to what they think is meaningful and important. Giving the capability to interact with the network, operators can better learn, predict, and serve the needs of their users.

Much more work is needed to figure out the most appropriate way to capture user intents and how to translate them into appropriate network semantics. We hope this work helps to seed this research. We look forward to seeing new ways to capture users' intents and how these can be translated to network policies.

6. ACKNOWLEDGEMENT

This research is supported in part by the NSF POMI (Programmable Open Mobile Internet) 2020 Expedition Grant 0832820, ONRC (Open Networking Research Center) and the Mr. and Mrs. Chun Chiu Stanford Graduate Fellowship. The authors also like to thank Masayoshi Kobayashi and Ali Al-Shabibi for helpful feedback.

7. REFERENCES

[1] Comcast vs. your torrents : A recap. `http://gigaom.com/video/comcast-vs-your-torrents-a-recap/`.

[2] http://www.huffingtonpost.com/news/net-neutrality.

[3] MyBoost demonstration video. `http://yuba.stanford.edu/~yiannis/myboost.html`.

[4] S. Akhshabi, A. C. Begen, and C. Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. ACM MMSys '11, New York, NY, USA, 2011. ACM.

[5] S. Bauer, D. Clark, and W. Lehr. Powerboost. HomeNets '11.

[6] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying Skype user satisfaction. In *SIGCOMM '06*.

[7] K.-T. Chen, P. Huang, and C.-L. Lei. How sensitive are online gamers to network quality? *Commun. ACM*, 49(11), Nov. 2006.

[8] M. Chetty, R. Banks, B. Brush, J. Donner, and R. Grinter. "you're capped!" understanding the effects of bandwidth caps on broadband use in the home. In *CHI'12*.

[9] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *SIGCOMM CCR '89*.

[10] L. DiCioccio, R. Teixeira, M. May, and C. Kreibich. Probe and pray: Using upnp for home network measurements. In *PAM '12*. 2012.

[11] L. DiCioccio, R. Teixeira, and C. Rosenberg. Impact of home networks on end-to-end performance: controlled experiments. In *ACM HomeNets'10*, 2010.

[12] C. Dixon, R. Mahajan, S. Agarwal, A. Brush, B. Lee, S. Saroiu, and P. Bahl. An operating system for the home. In *NSDI '12*.

[13] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. ACM SIGCOMM 2011.

[14] J. Gettys and K. Nichols. Bufferbloat: Dark buffers in the internet. *Queue*, 9(11):40:40–40:54, Nov. 2011.

[15] O. Goga and R. Teixeira. Speed measurements of residential internet access. In *PAM*, pages 168–178, 2012.

[16] B. D. Higgins, A. Reda, T. Alperovich, J. Flinn, T. J. Giuli, B. Noble, and D. Watson. Intentional networking: opportunistic exploitation of mobile network diversity. In *ACM MobiCom '10*, pages 73–84, New York, NY, USA, 2010. ACM.

[17] P. Kanuparthy and C. Dovrolis. Shaperprobe: end-to-end detection of isp traffic shaping using active methods. IMC '11.

[18] M. Lasar. It could be worse: data caps around the world. `http://arstechnica.com/tech-policy/2011/04/how-internet-users-are-disciplined-around-the-world/`.

[19] G. Maier, F. Schneider, and A. Feldmann. NAT usage in residential broadband networks. In *PAM '11*.

[20] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Mar. 2008.

[21] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, A. Moore, A. Koliousis, and J. Sventek. Control and understanding: Owning your home network. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1 –10, jan. 2012.

[22] E. Nordstrom, D. Shue, P. Gopalan, R. Kiefer, M. Arye, S. Y. Ko, J. Rexford, and M. J. Freedman. Serval: An end-host stack for service-centric networking. In *USENIX NSDI*, April 2012.

[23] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Trans. Netw.*, 1:344–357, June 1993.

[24] A. Reggani, F. Schneider, and R. Teixeira. An end-host view on local traffic at home and work. In *PAM '12*.

[25] RFC 2205: The resource reservation protocol (rsvp). `http://tools.ietf.org/html/rfc2205`.

[26] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar. Can the production network be the testbed? In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, OSDI'10, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association.

[27] S. Sundaresan, W. de Donato, N. Feamster, . Teixeira, S. Crawford, and A. Pescapè. Broadband internet performance: a view from the gateway. *SIGCOMM CCR '11*.

[28] B. Van Schewick. Network neutrality: What a non-discrimination rule should look like. 2010.

[29] K.-K. Yap, T.-Y. Huang, B. Dodson, M. S. Lam, and N. McKeown. Towards software-friendly networks. In *ACM APSys '10*, New York, NY, USA, 2010. ACM.

[30] Y. Yiakoumis, K.-K. Yap, S. Katti, G. Parulkar, and N. McKeown. Slicing home networks. ACM HomeNets '11.