Refactoring Network Infrastructure to Improve Manageability: A Case Study of Home Networking

Marshini Chetty and Nick Feamster
Georgia Tech
{marshini,feamster}@cc.gatech.edu

This article is an editorial note submitted to CCR. It has not been peer-reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

Managing a home network is challenging because the underlying infrastructure is so complex. Existing interfaces either hide or expose the networks underlying complexity, but in both cases, the information that is shown does not necessarily allow a user to complete desired tasks. Recent advances in software defined networking, however, permit a redesign of the underlying network and protocols, potentially allowing designers to move complexity further from the user and, in some cases, eliminating it entirely. In this paper, we explore whether the choices of what to make visible to the user in the design of today's home network infrastructure, performance, and policies make sense. We also examine whether new capabilities for refactoring the network infrastructure—changing the underlying system without compromising existing functionality—should cause us to revisit some of these choices. Our work represents a case study of how co-designing an interface and its underlying infrastructure could ultimately improve interfaces for that infrastructure.

Categories and Subject Descriptors: C.2.3 [Network Operations]: Network management, H.5 [Information interfaces and presentation]

General Terms: Measurement, Management, Human Factors

Keywords: home networking, management, monitoring, software defined networking

1. INTRODUCTION

Users depend on a functional "always-on" home network connection to use an increasingly diverse set of applications and services. Unfortunately, home networks are difficult to manage, troubleshoot, secure, and maintaineven for expertsbecause networks are such complicated systems [8, 21, 50]. Of course, much of this complexity is inherent: a home network must support a broad range of devices, users, applications, and usage scenarios, and the underlying protocols that provide this level of support are themselves complex. This innate complexity of the network and the difficulty of changing the underlying infrastructure [4] have forced home network interface designers to either expose or hide some network function to help users achieve their goals (e.g., exposing Internet Protocol (IP) when setting up a router or configuring defaults to hide complexity) [17].

Unfortunately, until recently, the difficulty of changing the underlying network infrastructure has constrained interface designers to have no more flexibility than choosing which shade of lipstick to put on a pig. In this paper, we argue that material improvements in network interfaces can result when designers work together with network architects to co-design the underlying network infrastructure and the interfaces that control the infrastructure.

Researchers have suggested that the ability to refactor the underlying network infrastructure—changing where function is placed within the modules of an existing system (as is often done with large software systems [34]) can ultimately improve home networking interfaces without crippling existing functions [7]. Fortunately, recent developments in software defined networking can make this refactoring possible [26, 33] without requiring a complete overhaul of the Internet. In this paper, we thoroughly explore opportunities for refactoring home network functions to improve system visibility, drawing on examples from existing literature. We also suggest how such refactoring network functions could ultimately improve user experience.

We make three contributions. First, we highlight where current home networking infrastructure, performance, and policies are either too visible or too hidden from end-users, and how this under- or over-exposure to complexity negatively affects users experiences. Second, we argue that designers are no longer limited by the constraints of the existing home network infrastructure. Rather than simply deciding whether to hide or display existing network complexity to the user, designers can work with networking researchers to refactor network functions to facilitate creating applications and interfaces that better accommodate the user. Third, we provide examples of how refactoring network functions can facilitate the design of interfaces that allow users to more directly express their intent, permitting what we call *intentional user interfaces*. In this paper, we focus on how refactoring home networking infrastructure can result in more usable interfaces for home network management, but we believe that co-designing infrastructure and interfaces can be applied to improve the manageability of other types of networks, as well. The notion of co-designing infrastructure and interfaces may also ultimately apply to a much broader set of problem domains.

2. VISIBILITY AND INFRASTRUCTURE

We first survey previous work that has explored the role of visibility in the design of infrastructure in general and introduce two conditions that should be true about an aspect of the infrastructure in order for the interface to expose it. We then describe previous work studying the role of visibility in the design of home networking interfaces and infrastructure. Finally, we present some background on software defined networking and generally describe how this emerging paradigm may create new opportunities for co-designing infrastructure and interfaces.

2.1 Making (Some) Infrastructure Visible

Infrastructure refers to the substrates, frameworks, and building blocks that support our everyday activities [48]; for example, it may refer to electricity grids, roads, classification systems, software systems, or business processes. Typically, infrastructure remains unnoticed until it breaks [6, 49]. This is particularly true for home network infrastructure, the systems we use to gain and share Internet connectivity in our homes, where even today's best home networks expose underlying network complexity when a failure or fault occurs. Designing interfaces that allow users to troubleshoot and fix problems in home networks without being exposed to the underlying network infrastructure and mechanics remains an open problem. In fact, given that users just want their networks to work, much like with other infrastructures, it is reasonable to ask whether anything at all should be made visible.

We argue that a totally invisible infrastructure is not feasible because some aspects of these complex systems will always require human intervention. In the case of home networking, for example, the network will never be able to intelligently execute a user's intent without some input or oversight. Although we agree that an infrastructure should invisibly support user tasks, as opposed to only being visible upon failure (as Star observed [48]), we believe information that provides a user insight into a system's everyday functioning is necessary to help users implement a wide range of policies and functions. In this paper, we examine how to improve what is currently visible and invisible during day-to-day functioning of the home network, and how these choices enable or hinder users' goals.

The idea of improving visibility in infrastructure has been raised before, specifically, in terms of system intelligibility and accountability. Belotti and Edwards [3] suggest that providing visibility into a system's internals may enhance the users understanding of why the system is taking particular actions and improve system intelligibility. Similarly, to improve users' understanding of a system, accountability refers to having systems expose reflective metaphors that indicate what the system is doing and how it is working [15]. For example, a file transfer may fail for a number of different reasons, such as dropped connectivity between two machines but a progress bar alone does not indicate why a failure occurred. Both of these attributes help users form better conceptual models of how systems work to better use the system as discussed with the example of home heating controls [28].

When (and how) should infrastructure be made visible? In this paper, we focus on the following design question: How much information in infrastructure should be made visible (and when) to help a user better understand a system and reach their end-goals; and how can the infrastructure support this type of visibility?

In the case of home networks, we offer an answer to this critical question that guides our design discussion throughout the paper: **Infrastructure should be made visible if and only if it (1) improves situational awareness and (2) provides actionable information**. By *improving situational awareness*, we refer to information that makes the user more aware of the system environment in ways that aid with decision making towards his or her end goals (see Endsley for a detailed definition [19]). *Actionable information* refers to information that a user can or should do something about. For example, information about the presence of an unauthorized device on the home network both improves situational awareness and is actionable (i.e., the user could take steps to block that device's access to the network or better secure the network). On the other hand, a router Web page that shows information about IP address assignments or hardware addresses of individual devices generally satisfies neither of these criteria, at least in the common case. Ideally, the parts of the network infrastructure that do not improve situational awareness or present actionable information would "disappear" in everyday use [53].

2.2 Home Networking and Visibility

Unfortunately, current home network interfaces expose complex (and often irrelevant) arcane details about infrastructure, protocols, and services (e.g., "SSID" to name a wireless network and "MAC address", the unique hardware identifier for a device). On the other hand, information that improves situational awareness and is actionable is often not visible enough to end-users. For example, although it is difficult for a homeowner to accrue a giant water bill without noticing a leaky faucet (or a giant puddle on the floor), it is entirely possible for a home user to exhaust the bandwidth cap on their home Internet service without realizing they have done so [8].

In fact, previous work has emphasized the difficulties that users face with existing home networks [24, 25, 29, 52]. Many of these are rooted in the design of the network infrastructure, which was not created with home users or usage scenarios in mind [4]. Some researchers have developed point solutions to help users with specific problems (e.g., making users aware of security issues [12]). Still, we believe that usability problems are serious and widespread enough to warrant both a refactoring of the underlying infrastructure and a redesign of the interfaces.

Researchers have also proposed approaches to address home networking difficulties, including both a "bandage" approach, which designs interfaces to mask network complexity; and a "clean slate" approach, which advocates a complete overhaul of the Internet protocols to be more user-centric [44]. Because redesigning the network from scratch is not tenable, designers have mostly applied the bandage model (e.g., the Eden system, which helps users configure network via direct manipulation interfaces [55]). Edwards *et al.* introduce the notion of constrained possibilities [18], suggesting that an infrastructures design can constrain its uses and interfaces in the first place. They suggest that altering some of these constraints may require changes to the infrastructure itself.

We argue that software defined networking can finally allow us to overcome constrained possibilities for home networking interfaces. Refactoring the underlying network makes a "co-design" approach possible, whereby the design of the infrastructure and the interface take each other and the user into account, as with the user-centered design approach suggested for creating software infrastructures [34].

2.3 Software Defined Networking: Enabling Refactoring

Software defined networking is an emerging trend in network protocol and infrastructure design, whereby the behavior of network devices and protocols are controlled from high-level software programs, rather than "baked into the hardware" [26, 33]. In short, software-defined networking refactors the intelligence of the network so that the logic that defines network behavior resides in a software "controller" that is sometimes logically centralized. The behavior of switches, routers, and other network devices are defined by the commands that it receives from the controller. Although the controller may be physically or logically centralized and is sometimes separate from the devices that forward traffic, this separation need not be the case and may simply refer to the placement of logic or control in a software element that resides on the same device. For example, in the case of a home network, software control might reside on the home router or even "in the cloud" outside of the home [57].

In the specific case of home networking, software-defined networks enable a radical refactoring of the infrastructure. Whereas previously network devices in the home (e.g., home routers and access points) were considered immutable, the behavior of any of these devices can now be controlled from a high-level software program. As a result, many of the network functions that were previously considered unchangeable—such as the protocols used to make decisions about whether and how to forward network traffic, how performance is measured and represented, and so forth—can now be refactored.

This goal of refactoring the network is similar to those of many context-aware application toolkits for supporting intelligibility and control [13], which expose information about existing software programs functioning and distributed state so that they can be appropriately reflected to the user. Unlike supporting accountability, which is "a reflective representation that an interactive system can offer of its own activity" [16], software defined networks facilitate new aspects of network monitoring and control. We argue that the ability to refactor network functions makes it easier to expose the appropriate level of complexity, thereby making it possible to design interfaces that expose only information that improves situational awareness and is actionable by the end-user and hid or eliminate cumbersom details.

In the following sections, we point to existing case studies and previous work to highlight where current home networking infrastructure is either too visible or too hidden from end-users. Next, we explain how refactoring the network can create possibilities for new co-designed user interfaces. We examine network infrastructure, performance, and policies.

3. INFRASTRUCTURE

Home networking infrastructure comprises (1) the networking devices in homes (e.g., access points, routers), (2) the devices that connect to each other and to the Internet through them (e.g., desktops, laptops, mobile phones, set-top boxes), and (3) the technologies that connect devices to each other (e.g., cables).

3.1 Setting up the Network

Installation: too much visibility. Installing and configuring a home network requires users to manipulate low-level networking protocols. For example, when a user first installs a home router, they must know the difference between the wired connection to the Internet, or the wide-area network, and the wired connections inside their home, or local-area network. Many users cannot make this distinction and mistakenly connect their home router incorrectly [41]. A user who succeeds in connecting the home router to the Internet must then grapple with the confusing terminology of wireless channel encryption for security purposes, and Dynamic Host Configuration Protocol (DHCP) configuration and IP addressing for achieving connectivity between devices and the Internet. These technicalities often cause users to simply install the network router with default settings; in extreme cases, they may even return the device altogether [5, 27].

Clearly, the mechanisms for establishing basic connectivity are fundamentally too complex, which makes it incredibly difficult to design a usable interface for home network setup in the first place. Existing systems, such as Network Magic [37] and NetPrints [1], attempt to hide this complexity from users by using wizards to guide users through setup, making configuration easier using physical interfaces, or automatically configuring devices with little user involvement. In fact, performing nearly any task ultimately requires a user to go "under the hood" to deal with underlying arcane network protocols and frameworks.

Refactoring basic connectivity to reduce visibility: By refactoring the infrastructure, the interfaces we design can be significantly less constrained. Refactoring network infrastructure to make the home router "smart", by moving some of the complexity into the router itself can enable all network devices to connect over a flat Ethernet network, obviating the need to configure and perform IP routing (e.g., assigning IP addresses to devices). The notion of a flat network has already been applied in other network domains. Designers of data center networks have realized that configuring a network of servers as a single flat network reduces configuration complexity and makes it easier to move servers from one portion of the network to another [36]. In another case, designers of large wireless networks (e.g., campus or enterprise networks) also set up their network as a single large "Virtual Local Area Network (LAN)" so that devices can move seamlessly from one network point to another [58]. Applying this approach in the home could allow us to come closer to the vision of a "plug and play" network [44], where users do not have to configure individual devices or the main router. Better interfaces could then provide users with a much simpler conceptual model of this mode of connectivity.

3.2 Maintaining the Network

Maintaining connectivity: not enough visibility. After users set up the home network, they may need to determine what devices are connected to the network, to ensure it is functional and secure; and what level of performance these devices are receiving to determine how their tasks will be affected. Today's interfaces make these tasks incredibly difficult: although most router Web pages do enumerate the connected devices and status of the users connection to the ISP, this information is not necessarily actionable. Current interfaces may not show whether a device can reach the Internet, or if a device is connected to the wrong network, particularly when neighbors have open wireless connections [11]. Fortunately, because the router is a central control and monitoring point in the home, all traffic to the Internetand, in many cases, all internal trafficpasses through the router. The router thus has a wealth of information that, if better presented, could improve situational awareness and provide actionable information.

Refactoring the network to improve visible connections: By changing the infrastructure to provide the router with additional functions to expose information about both connectivity and performance, a user interface for the home network can expose more intuitive abstractions that provide situational awareness and actionable information for both internal and external connectivity. An example of improving situational awareness in the home network to benefit end users is the Kermit study. This study showed that users could more easily manage their home network infrastructure if they could associate devices on their home network with recognizable pictures, and if those devices were displayed only when they are connected to the network [10]. This interface allowed users to see how and whether their devices were connected to the network and when unauthorized devices were connected. A user who sees an unauthorized device could then block access to that device. Refactoring the infrastructure by moving functions into the router can enable other interfaces that improve situational awareness or expose actionable information.

Another system called Home Watcher [9] revealed bandwidth "hogs" on the home network; in this system, devices were automatically added to an interface display, improving situational awareness. The system also provided actionable information, allowing users to better troubleshoot connectivity problems. In this type of system, if a device is connected to the router but that device is experiencing connectivity problems, then a user could infer that the problem was not inside the home and report the problem to the ISP.

Changing or upgrading the home network infrastructure: too much visibility. Because of the difficulty in managing home network configuration, users spend many hours performing "digital housekeeping" [52]. Furthermore, since home network configuration is so brittle, users may be reluctant to change or upgrade existing infrastructure (e.g., access points, routers, end-user equipment), especially given the risk of breaking a working configuration [45]. Although software development has well-defined processes for reverting faulty code, networks have only limited capabilities for reversing a configuration or infrastructure change. The inability to revert network device or software configuration is exacerbated by the inability to revert a physical network configuration (e.g., the network topology or wiring) to a previous state when an upgrade fails.

Refactoring the network to reduce connectivity complexity and track state: Fortunately, refactoring the network as a single "flat" network can also facilitate configuration changes and network upgrades. Because, in a flat network, all devices can communicate with every other device, a user can make changes to the physical topology—moving devices around and change the way they connect wires—without fear of triggering unknown dependencies (e.g., IP address subnets). In the absence of topological hierarchy, the only changes to the network itself are changes to the configurations of the devices themselves, which can be managed at a central location (such as a home router). Once all network changes are reduced to configuration or policy changes at a centralized control point (e.g., the home router), a user can easily revert the network state to a previous version by reverting the state of that single device.

Once network changes are recorded and revertible, interfaces can make these changes (and "versions" of the network) more visible to the end-user. This task, of course, raises questions concerning how to expose this new functionality to users using intuitive interfaces that improve situational awareness and enable user action (e.g., one might imagine using sliders and a timeline, with an evolving network representation of some kind), rather than using a standard software version control metaphor.

Refactoring the network to shift management tasks to third parties: Emerging technologies such as network virtualization allow a single physical network infrastructure to be "sliced" into multiple virtual networks. In other words, slicing allows us to conceive of the network as many separate entities, each with its own configuration, services, and management capabilities [22]. The ability for a single physical infrastructure to be divided and shared by multiple users or service providers enables a variety of new functions. One interesting possibility is the notion that a home user could grant the ISP access to a slice of the home network to allow them to remotely setup, manage, and troubleshoot the user's home network [57]. Slicing could provide an ISP enough visibility to remotely manage the network, while still limiting the ISPs access to private information in the user's home network. In this case, complexity is moved away from the user totally, but interfaces must be co-designed to enable remote troubleshooting and setup of the virtual slice according to user preferences.

4. PERFORMANCE

Network performance affects the usability of various networked applications. When setting up their home network, users must compare information concerning the advertised network performance of different Internet service providers. During everyday operation, users expect the network to perform well enough to support a variety of applications, ranging from email to Web browsing to streaming video (often simultaneously). Home networks currently provide inadequate visibility into both advertised and achieved network performance; refactoring the network may improve situational awareness about network performance and also make this information actionable.

4.1 Setting up the network for optimal performance

Choosing an Internet service plan: too little visibility. A user who is setting up a home network must first choose an Internet service plan from available ISPs that may offer a diverse set of options. Today, a user must select a service plan from an ISP based solely on advertisements concerning only two metrics: the speed of the connection (i.e., the amount of data that the users connection to the ISP can support in a given time interval) and cost. Unfortunately, many aspects of ISP performance that can perceivably affect the user experience of specific online applications are not visible in broadband advertisements, and auditing a plan after purchase is not currently possible. Even if other aspects were visible, current home networking infrastructure does not support monitoring the performance of the network over time or reporting those statistics to the user or even a third party governing body. Although exceptions exist, such as performing one-off "speed tests" or flashing a router with custom firmware [54], these approaches are not intuitive (and often not actionable).

Refactoring the network to be allow better measurements: We can refactor the network infrastructure to support monitoring that accurately and intuitively reflects ISP performance and make that information visible over time to improve situational awareness. For example, OpenWrt modules running on the home router can measure network performance continuously (and more accurately) [40]. If users could measure and understand a wider variety of performance metrics, ISPs could advertise their access network performance in home broadband advertisements using a more comprehensive—and intuitive—set of metrics. After users have purchased a particular service plan, the home network could provide a means for auditing these metrics, making the information actionable, since a user could change their plan if it does not perform as advertised.

The challenge in providing a comprehensive, yet intuitive, measure of network performance involves (1) enumerating the network metrics that can ultimately affect performance in visible ways, and (2) mapping these metrics to representations that users can understand. Networking researchers have taken the first step, by proposing a set of "raw network metrics" that can affect application performance [50]; the next step involves mapping these metrics to more intuitive representations.

One approach might be to map these low-level performance metrics to more meaningful application performance metrics. For example, researchers have proposed a nutrition label that presents more user-centric metrics that are derived from low-level metrics [51]. For example, a typical user may have difficulty understanding how packet loss can introduce severe performance degradations in voice stream; on the other hand, one might imagine profiling a user or household to identify the common applications used in the home, and presenting a report for the performance of each of those applications, based on the observed the low-level metrics (e.g., Skype performance may be poor, without necessarily telling the user it is because packet loss is higher than the application expects.) A novice user need only understand that Skype performance is poor; a more expert user, however, might gain additional insight from knowing the underlying cause. Deriving high-level representations from low-level performance metrics that improve situational awareness and present actionable information remains an open problem.

An alternate approach might be to map low-level performance metrics to intuitive visual representations that a user can understand. For example, the downstream speed of the ISP connection might be represented by the width of a pipe, akin to how the Kermit system used the thickness of a line to indicate how much bandwidth a user was using [10]. Packet loss (i.e., traffic that does not make it to its destination) might be shown as certain traffic not making it all the way through the pipe; latency (i.e., the time traffic takes to reach its destination) could be illustrated with pipes of different lengths. There may be more intuitive representations that we have not yet envisioned; ultimately, any mapping of performance metrics to alternative representations must be co-designed for users at different levels of expertise.

4.2 Auditing and troubleshooting performance

Isolating the source of a performance problem: too little visibility: Users and regulatory agencies are interested in determining whether ISP performance matches advertised rates [20, 38]. Unfortunately, the performance of both the home network and the ISP network is currently mostly invisible to users, meaning users lack a situational awareness of the network and have less actionable information about how the network is running at any given moment [10]. Rather, they only see an "end-to-end" view of application performance (i.e., measured from the end-user device to a server on the other end of the Internet path), which makes it difficult to isolate a problem to any device, application, or portion of the network [9]. Due to this lack of actionable information, users may be unable to diagnose the cause of performance degradation. ISPs themselves may face service calls from consumers who are experiencing poor performance [41], such as the Internet being "slow", even when the degradation results from the user's own network configuration, activity, or devices. Without a view into the home network, however, even ISPs may be unable to troubleshoot performance problems.

Refactoring the network to provide continuous visibility of ISP and home network performance: Refactoring the network function so that the home router, rather than end devices, perform performance measurements could facilitate collecting and reporting of performance metrics to both ISPs and consumers. This approach both improves situational awareness and provides actionable information, by helping the user isolate performance problems better. If these measurements are performed from the router, they are also likely to be more accurate and not affected by properties of an end-user device [2], a problem that users of speedtest.net face [47]. An example system that refactors the network in this way is Project BISmark [42], an open platform for continuously measuring home network performance from home routers. Clearly, performance measurements of the home network could compromise user privacy. Studies have already shown that network metrics can tell household members whether children are doing homework or up beyond their bedtime for instance [9]. HCI researchers must understand what information users are comfortable sharing, and how aggregating data and limiting access can improve privacy both with other users in the home and with the ISP.

Refactoring the network to improve performance visibility for ISPs to ultimately benefit users: To improve an ISP's visibility into the home network, we can refactor the network to allow limited diagnostics and performance measurements from a vantage point inside the home. Because users may be continually changing the infrastructure and configuration of the home network itself [43], an ideal location for such a diagnostic platform is also on the home router itself. From such a platform, an ISP could perform simple diagnostics, such as: (1) measuring speed of the wireless network to various devices in the home; (2) directly measuring the performance of its own network, as seen from user's home router (i.e., without the presence of other devices or factors in the home that might affect performance). An ISP might even be able to capture traffic traces for specific applications and determine the location (and source) of other performance problems, such as packet loss.

Providing the ISP sufficient situational awareness and actionable information without compromising user privacy is challenging: a user may not wish to let the ISP know the types of devices he or she has connected to the home network, and the user certainly may not want to give the ISP access to those devices. On the other hand, knowledge about the types of devices that are connected to the network might also help either the user or the ISP determine the likely source of a problem. Striking the right balance between usability and privacy involves exposing sufficient actionable information to help the user help themselves, without sending private information to the ISP.

5. POLICY

Policy refers to the rules that control how traffic flows through the network. Policy may encompass a variety of factors, including access control (who or what is allowed to access network), prioritization (which application should receive better service), rate limits (how fast any user, device, or application can send traffic), and usage caps (how much data any user, device, or application should be allowed to send over a period of time). Policies can be either internal (i.e., applied within the home to devices, users, and applications) or external (i.e., applied by an ISP as it traverses its network). In this section, we argue that the current home networking infrastructure provides too much visibility during the setup of internal network policies, but not enough visibility into either how ISPs establish and implement external policies or the extent to which traffic inside the home actually conforms to the established internal policies. We argue that moving many policy-related functions to a central controller can allow for a co-design of better interfaces for setting up and maintaining policy.

5.1 Setting up network policies

Setting up internal home network policies: too much visibility. Users react positively to having control over limiting, throttling, or prioritizing connections [9, 10] but setting up this functionality is too complex, because current home network interfaces do not separate policy from mechanism. To specify a high-level policy, users must specify and configure the low-level mechanisms that implement them; there is no way for a user to specify policies at a higher level of abstraction. Unfortunately, because today's mechanisms for setting up these policies are so complex [32], users typically never bother to configure them in the first place, even though there are a wide variety of policies that they would like to enact. For example, a home network user may want to allow network guests

Internet access [57], but to restrict access to the devices and their files on the home network. They might also want to exercise parental controls [56] or specify which types of activities should take priority (e.g., never let my file backups interfere with the performance of streaming video). Some aspects of control even require a user to flash their router with custom firmware, a task that is far beyond the call of duty for the average user.

Many of the existing mechanisms for specifying and implementing policy are so complicated that users often never touch them in the first place, simply resorting to "all or nothing" access policies (either a user doesn't have the wireless password and cannot get on the network; or they do, in which case they can do anything). Previous designs have attempted to expose some of this complexity to users, to improve awareness of the traffic on the network [12] or help them set access control policies [55]; however, exposing these complex, low-level mechanisms to users will simply result in complex interfaces. Developing more usable interfaces ultimately requires simplifying the underlying mechanisms, as well.

Refactoring the network to make policy setup more intuitive, by separating policy from mechanism: Ideally, users would be able to specify network policies at a higher level of abstraction, effectively specifying what they want to do, as opposed to how the infrastructure should implement it. Refactoring the underlying infrastructure to separate the control plane (i.e., functions that decide how traffic should be forwarded and implement policy) from the data plane (i.e., functions that simply forward the traffic) should make it easier to design an implement such a language. Researchers initially proposed separating the control and data planes in ISP networks to make network management easier for network operators; the OpenFlow protocol, an instantiation of software defined networking [39], enables this change. Applying OpenFlow to achieve a similar refactoring of the control and data planes in the home network could also simplify network management in homes by enabling users to specify higher-level policies.

One recent example of a system that could enable this type of refactoring in the home is Resonance, which uses OpenFlow to allow users to specify high-level policies in terms of states and actions [30]. For example, a user might enumerate various actions (e.g., block a user, limit traffic from a certain application or device) that the network should take based on various events or states (e.g., time of day). In this example, although the current framework for specifying policies centralizes and abstracts policy specification from low-level mechanisms, the current policy language is likely still too complex for an average user. More research can help determine the types of languages and interfaces that users are comfortable with for specifying network policies; some researchers already have a head start on tackling this issue [35]. For example, users may be more comfortable specifying actions such as "block this person from my Internet", "throttle this application", "prioritize this person's device", or "let the kids access the Internet before supper but not after bedtime".

5.2 Monitoring and adjusting network policies

Monitoring internal network policies: too little visibility. After specifying policies, users need interfaces that help them monitor usage and enforce these policies. Users may want to monitor network activity to ensure that network usage and activity conforms to the policies that they have specified. For example, a user who is on a capped bandwidth plan may wish to monitor how much of the cap they have used and ensure that no user exceeds their allocation [8]. They may also wish to see the current network usage, so that they can determine whether ongoing network traffic is subject to the specified policies (e.g., Is traffic for a backup to the cloud actually affecting the performance of the video being streamed simultaneously from Netflix?).

Refactoring the network to make policies more visible: Refactoring the network to have a measurement model for performance could also reveal network policies in the home. Policy-related concerns force designers to confront another question concerning visibility aside from providing situational awareness and being actionable: Which users on the home network should be able to see this information, bearing in mind user privacy? Past studies of network performance prototypes in users homes [9, 10] assumed that all users in the house should be able to see aggregated information about which devices are using the network, because information about the current network usage within the home could help a user isolate possible causes of a performance problem (e.g., if another user on the network is using a significant fraction of the bandwidth). These studies report that social and power structures in the home may imply that certain users, such as parents, want access to the information and rights to control resources, while others do not [9]. In all cases, interfaces should be flexible enough to designate who has access to the information. To preserve user privacy, usage information could be aggregated, to hide data about which sites a user is visiting.

Monitoring ISP and external network policies: too little visibility. The current network infrastructure provides users little visibility into Internet service provider policies that may affect the performance of different applications [21]. In recent years, ISPs have implemented policies to intentionally block or de-prioritize certain application traffic, such as Comcast blocking BitTorrent traffic in 2007 [14]. Users and regulators are increasingly interested in measuring the performance that their ISPs deliver [20, 38]; furthermore, with the advent of usage caps from providers such as Comcast and AT&T [46], users may also wish to monitor how users, applications, and devices consume the usage allocation. Currently, ISPs have no obligation to make these policies visible to users, so users generally are left to their own devices to discover that an ISP is implementing policies that intentionally degrade the performance of specific applications. Unfortunately, the current network infrastructure makes it difficult for users to discover these policies.

Refactoring the network to make external policies more visible: Like measuring ISP performance, one of the reasons that inferring ISP policies is difficult is that there is currently no infrastructure for a user to continuously monitor the ISPs side of the home network. The measurement model we proposed in the previous section on performance might also permit policy-monitoring interfaces, by providing information on ISP network operation. This information will improve situational awareness and allow users to act on policy violations.

6. INTENTIONAL HOME NETWORK INTERFACES

The previous sections explained how changes to underlying network infrastructure enable design decisions concerning visibility that could improve interfaces for tasks relating to infrastructure, performance, and policy. From our examples, it becomes clear that the property of networking infrastructure that lends itself to co-design with user interfaces and applications is programmability. When refactoring an infrastructure such as the home network does not compromise its existing functionality, then user applications can either build on its main interfaces and abstractions or use refactoring to support additional monitoring and control. Refactoring infrastructure provides control over the underlying system behavior to support application requirements including intelligibility, accountability, or visibility.

In this section, we provide examples where interfaces and networks could be co-designed. The interfaces that we describe below could provide users actionable information and improve situational awareness; we call these interfaces *intentional*, since they allow more direct, declarative expressions of intent with respect to the behavior of the underlying system.

Infrastructure: A Home Network Looking Glass As we previously described, network virtualization ("slicing") can provide the ability for an ISP to perform remote troubleshooting or maintenance on behalf of its users. Although the refactoring that we described could provide the functions for remote troubleshooting, the interfaces for performing these types of tasks must still be designed. Specifically, network operators need usable interfaces that allow them to quickly isolate performance problems for home users from a remote vantage point [41]. Determining what information should be collected and presented to the operator—and how to balance the home user's privacy with the need for an ISP operator to see the configuration, topology, and devices on the home network to quickly and accurately isolate performance problems—is an area for future work in co-designing interfaces and the underlying architecture to support it.

Performance: Crowdsourced ISP "Weather Maps" Home users, ISPs, and regulators need better ways to visualize the performance of broadband access networks to balance power relationships between these stakeholders [21]. When a user is experiencing a certain level of performance, their first question is often "Are other customers experiencing the same problem?" An interface that could allow users not only to monitor their own performance but also to compare their performance to other users with the same ISP service plans or geography could help users answer this question. The infrastructure that we described in earlier sections, whereby each home network continually measures and reports performance information to a home user, could form the basis of a crowdsourced Internet service provider weather map. Similar concepts have been proposed in the past [23], but they have faced technical problems due to the fact that measurements were performed from end user devices, as opposed to the home router. More importantly, any such application needs to be co-designed to help users intuitively understand whether other similar users are experiencing the performance that they observe or whether a different ISP is providing better service to their particular geography.

Policy: Usage Meters and Bandwidth Brokers Usage quotas (or "caps") limit how much users can download per billing cycle before they are disconnected, throttled, or billed at a higher rate. Exploratory studies of usage and anecdotal evidence from popular media suggest that users need better ways to monitor and control their usage over the course of a billing cycle [8, 31]. A smarter home router that could monitor the utilization of different devices, applications or users [35] and throttle or limit network access based on utilization [30] to allow users to more easily regulate consumption of usage caps. Co-designing an interface to allow users to easily monitor and manage their usage caps (e.g., limiting the usage of a particular application, trading caps with other users in the house) is an exciting application design problem.

Applying the Refactoring Approach to Other Domains Although we have focused on how refactoring infrastructure may ultimately improve interfaces in home networks, there are other areas of computing where a refactoring of the infrastructure might merit making different design choices about user interfaces. For example, many conventional desktop applications are migrating to the cloud, yet the interfaces that users must interact with often do not reflect the fact that the infrastructure has been refactored. For example, a cloud-backed word processing program exposes an interface that roughly corresponds to its desktop-application counterpartit does not offer the user better tools for managing disconnection from the Internet, periods of poor network performance, data provenance and privacy, and so forth. Of course, as the underlying infrastructure for cloud applications continues to evolve and mature, user interfaces must also adapt to reflect these changes. In light of the lessons of this paper, however, one might also ask whether more design-inspired infrastructure decisions (e.g., caching some content on the local desktop, providing better primitives for encryption or privacy) can ultimately result in better interfaces.

7. CONCLUSION

Home networking remains challenging for users because of the inherent complexity of the underlying infrastructure needed to support a variety of network functions. Recent developments in software defined networks facilitate a refactoring of the underlying network without compromising existing functionality. In this paper, we showed where current aspects of home networking expose too little or too much detail about the underlying network to help home users achieve their needs. We also showed that refactoring the network creates the possibility for interfaces that improve a home network user's situational awareness with actionable information. Finally, we presented examples of intentional interfaces that build on a refactored home network to allow co-design of the interface and underlying system.

The examples we presented in this paper illustrate how the programmability that software defined networking provides can facilitate the co-design of infrastructure and interfaces to make networks more manageable. Evaluating the architectural changes that we suggest is an subject of ongoing research. Additionally, some of the refactoring approaches we propose may conflict, depending on the context: at some times, it makes sense to hide certain aspects of the infrastructure, and at others, it is helpful to expose more detail. Determining how to resolve these potential conflicts as part of a holistic architecture for home network management poses a significant and important challenge, and is a rich direction for future research.

ACKNOWLEDGMENTS

This work was supported by the the National Science Foundation through awards CNS-1059350, CNS-0643974, and a generous Google Focused Research Award. We thank Ken Calvert, Sam Crawford, Keith Edwards, Ilda Ladeira, Hyojoon Kim, Andrea Parker, Bethany Summer, Srikanth Sundaresan, and Susan Wyche for feedback and discussions that helped improve this paper.

References

[1] B. Aggarwal, R. Bhagwan, T. Das, S. Eswaran, V. N. Padmanabhan, and G. M. Voelker. Netprints: diagnosing home network misconfigurations using shared knowledge. In *6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 349–364, Boston, MA, 2009. USENIX Association.

[2] S. Bauer, D. Clark, and W. Lehr. Powerboost. In *ACM SIGCOMM HomeNets Workshop*, Toronto, Ontario, Canada, Aug. 2011. ACM.

[3] V. Bellotti and W. K. Edwards. Intelligibility and accountability: Human considerations in context-aware systems. *Journal of Human-Computer Interaction*, 16(2-4):193–212, 2001.

[4] M. S. Blumenthal and D. D. Clark. Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. *ACM Trans. Internet Technol.*, 1(1):70–109, 2001.

[5] S. Bly, B. Schilit, D. McDonald, B. Rosario, and Y. Saint-Hilaire. Broken expectations in the digital home. In *CHI Extended Abstracts*, pages 568–569, 2006.

[6] G. Bowker and L. Star. *Sorting Things Out: Classification and Its Consequences*. MIT Press, 1999.

[7] K. L. Calvert, W. K. Edwards, and R. E. Grinter. Moving toward the middle: The case against the end-to-end argument in home networking. In *ACM SIGCOMM Workshop on Hot Topics in Networking (HotNets)*. ACM, 2007.

[8] M. Chetty, R. Banks, A. J. Bernheim Brush, J. Donner, and R. E. Grinter. Under development: While the meter is running: computing in a capped world. *interactions*, 18:72–75, 2011.

[9] 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. Atlanta, GA, May 2010.

[10] M. Chetty, D. Haslem, A. Baird, U. Ofoha, B. Sumner, and R. Grinter. Why is my internet slow?: Making network speeds visible. Vancouver, BC, Canada, May 2011.

[11] M. Chetty, J. Sung, and R. E. Grinter. How smart homes learn: The evolution of the networked home and household. In *Ubicomp*. Springer-Verlag, 2007.

[12] S. Consolvo, J. Jung, B. Greenstein, P. Powledge, G. Maganis, and D. Avrahami. The wi-fi privacy ticker: improving awareness and control of personal information exposure on wi-fi. In *Ubicomp*, pages 321–330, Copenhagen, Denmark, 2010. ACM.

[13] A. Dey and A. Newberger. Support for context-aware intelligibility and control. In *ACM CHI*, Boston, MA, May 2009.

[14] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi. Detecting Bittorrent blocking. In *Internet Measurement Conference*, pages 3–8, Oct. 2008.

[15] P. Dourish. Accounting for system behaviour: Representation, reflection and resourceful action. In *Computers in Context*, Aarhus, Denmark, 1995.

[16] P. Dourish, A. Adler, and B. C. Smith. Organising user interfaces around reflective accounts. In *Reflection*, San Francisco, CA, 1996.

[17] W. Edwards, V. Belotti, A. Dey, and M. Newman. Stuck in the middle: The challenges of user-centered design and evaluation for infrastructure. In *ACM CHI*, Ft. Lauderdale, FL, 2003.

[18] W. K. Edwards, M. W. Newman, and E. S. Poole. The infrastructure problem in hci. pages 423–432, Atlanta, GA, May 2010.

[19] M. R. Endsley. Measurement of situation awareness in dynamic systems. 37(1):65–84, 1985.

[20] Connecting America: The National Broadband Plan, 2010. Federal Communications Commission.

[21] Measuring Broadband America: A Report on Consumer Wireline Broadband Performance in the U.S. Technical report, 2011. Federal Communications Commission.

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

[23] Grenouille. `http://www.grenouille.com/`.

[24] R. E. Grinter, N. Ducheneaut, W. K. Edwards, and M. Newman. The work to make the home network work. In *ECSCW*, pages 469–488, Sept. 2005.

[25] R. E. Grinter, W. K. Edwards, M. Chetty, E. S. Poole, J.-Y. Sung, J. Yang, A. Crabtree, P. Tolmie, T. Rodden, C. Greenhalgh, and S. Benford. The ins and outs of home networking: The case for useful and usable domestic networking. *ACM Trans. Comput.-Hum. Interact.*, 16(2):1–28, 2009.

[26] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, 2008.

[27] J. Horrigan and S. Jones. When technology fails. Technical report, Pew Internet and American Life Project, 2008.

[28] W. Kempton. Two theories of home heat control. *Cultural Models in Language and Thought*, 1987.

[29] S. Kiesler, V. Lundmark, B. Zdaniuk, and R. E. Kraut. Troubles with the internet: The dynamics of help at home. *Human Computer Interaction*, 13:323–351, 2000.

[30] H. Kim, S. Sundaresan, M. Chetty, N. Feamster, and W. K. Edwards. Communicating with caps: Managing usage caps in home networks. In *ACM SIGCOMM (Demo)*, Toronto, Ontario, Canada, Aug. 2011.

[31] M. Lasar. It could be worse: data caps around the world. Apr. 2011.

[32] M. Mazurek, J. Arsenault, J. Bresee, N. Gupta, I. Ion, C. Johns, D. Lee, Y. Liang, J. Olsen, B. Salmon, R. Shay, K. Vaniea, L. Bauer, L. Cranor, G. Ganger, and M. Reiter. Access control for home data sharing: Attitudes, needs and practices. Atlanta, GA, May 2010.

[33] 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, 2008.

[34] T. Mens and T. Tourwe. A survey of software refactoring. *IEEE Transactions on Software Engineering*, 30(2):126–139, 2004.

[35] R. Mortier, B. Bedwell, K. Glover, T. Lodge, T. Rodden, C. Rotsos, A. W. Moore, A. Koliousis, and J. Svente. Supporting novel home network management interfaces with openow and nox. In *ACM SIGCOMM (Demo)*, Toronto, Ontario, Canada, Aug. 2011.

[36] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. In *ACM SIGCOMM*, pages 39–50, Barcelona, Spain, Aug. 2009.

[37] Network magic. `http://www.purenetworks.com/product/pro.php`.

[38] Ofcom. Uk broadband speeds, may 2010. the performance of fixed-line broadband delivered to uk residential consumers, 2010.

[39] OpenFlow. Openflow switch consortium. `http://www.openflowswitch.org`, 2008.

[40] Openwrt. `https://openwrt.org/`.

[41] E. S. Poole, W. K. Edwards, and L. Jarvis. The home network as a socio-technical system: Understanding the challenges of remote home network problem diagnosis. *Comput. Supported Coop. Work*, 18(2-3):277–299, 2009.

[42] Project BISmark. `http://projectbismark.net`.

[43] T. Rodden and S. Benford. The evolution of buildings and implications for the design of ubiquitous domestic environments. In *ACM CHI*, pages 9–16, Ft. Lauderdale, FL, 2003.

[44] E. Shehan and W. Edwards. Home networking and hci: What hath god wrought? In *ACM CHI*, 2007.

[45] E. Shehan-Poole, M. Chetty, R. Grinter, and K. Edwards. More than meets the eye: Transforming the user experience of home network management. In *ACM DIS*, Cape Town, South Africa, 2008.

[46] R. Singel. Shed a tear: The age of broadband caps begins monday. *Wired*, 2011.

[47] speedtest.net. `http://www.speedtest.net`.

[48] L. Star. The ethnography of infrastructure. *American Behavioural Scientist*, 43(3):377–391, 1999.

[49] S. Star and K. Ruhleder. Steps towards an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research*, 7(1):111–134, 1996.

[50] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescape. Broadband internet performance: A view from the gateway. In *ACM SIGCOMM*, Toronto, Ontario, Canada, Aug. 2011.

[51] S. Sundaresan, N. Feamster, R. Teixeira, A. Tang, W. Edwards, R. Grinter, M. Chetty, and W. de Donato. Helping users shop for isps with internet nutrition labels. In *ACM SIGCOMM Workshop on Home Networking*, Toronto, Ontario, Canada, Aug. 2011.

[52] P. Tolmie, A. Crabtree, T. Rodden, C. Greenhalgh, and S. Benford. Making the home network at home: Digital housekeeping. In *ECSCW*, Limerick, Ireland, 2007.

[53] P. Tolmie, J. Pycock, T. Diggins, A. MacLean, and A. Karsenty. Unremarkable computing. In *ACM CHI*, pages 399–406, 2002.

[54] Tomato firmware. `http://www.polarcloud.com/tomato`.

[55] J. Yang and W. Edwards. Eden: Supporting home network management through interactive visual tools. In *UIST*, pages 109–118, New York, NY, 2010.

[56] S. Yardi and A. Bruckman. Social and technical challenges in parenting teens' social media use. Vancouver, BC, Canada, May 2011.

[57] Y. Yiakoumis, K.-K. Yap, S. Katti, G. Parulkar, and N. McKeown. Slicing home networks. In *ACM SIGCOMM Workshop on Home Networking*, Toronto, Ontario, Canada, Aug. 2011.

[58] M. Yu, J. Rexford, S. Xin, S. Rao, and N. Feamster. A survey of virtual lan usage in campus networks. *IEEE Communications*, 49(7):98–203, 2010.