Contextual Router: Advancing Experience Oriented Networking to the Home

Ilker Nadi Bozkurt
Duke University
ilker@cs.duke.edu

Theophilus Benson
Duke University
tbenson@cs.duke.edu

ABSTRACT

Home networks often deliver poor application performance. However, recent attempts to improve performance expose low-level configurations or are optimized for individual applications. These approaches are limited in applicability or in effectiveness.

In this work, we argue that the home network should incorporate user specified preferences, the context of user interactions and application models in determining resource allocations. Furthermore, the home network should dynamically adjust allocations to reflect changes in user behavior. To this end, we present the design goals for Contextual Router, a management framework for home networks that enables optimal network utility. We developed a prototype implementation of Contextual Router. Our evaluations show that Contextual Router significantly improves page load times and reduce buffering events and buffering time.

CCS Concepts

•**Networks** → **Network management; Home networks;** *Routers;*

Keywords

Software Defined Networking (SDN); Home Networks; Bandwidth management; Quality of Service (QoS)

1. INTRODUCTION

Recently, there has been a surge in interest in management of home networks. This increased interest is due to a number of trends including the following two: first, an increase in the number of networked devices connected to the home, both due to the emergence of the Internet of Things (IoT) and the increase in the number of personal electronics; and second, a manifesto by content providers to deliver richer, more bandwidth intensive content to the end user. Despite the increase in bandwidth demands, home networking speeds have remained largely constant. As a result of these trends, application performance in home networks has become highly unpredictable with many home users suffering from poor quality of experience.

Contemporary efforts to improve performance have focused on application specific optimizations [16,33] or on queuing and scheduling disciplines for home routers [23, 28]. These efforts are either limited in scope, static, or agnostic to the user's expectations. In general, they ignore a very important fact, namely, that a user's expectations of quality are contextual in nature and highly variable across devices, e.g. watching Netflix on a cell phone versus on a smart TV or when watching Netflix versus performing an Operating System update. Further, these expectations of quality are relative, and rarely a strictly monotonic function. For example, a user watching a video on a cell phone may strongly prefer 420 bit rate over 240. Yet, this same user may see little value in using 720 over 420.

Motivated by these observations, we revisit the design of traditional home networks. We argue that home networks should be context aware; where a context is defined as an abstract construct that captures how a user is interacting with a specific application. In this paper, we loosely refer to a context as a tuple consisting of the application and the device on which the application is being run. [1]

Context aware home networks, or *Contextual Router*, are different from traditional networks in that they (1) proactively engage users to determine the user's expectations under various contexts; (2) monitor the user to determine her current contexts; (3) periodically calculate bandwidth allocations to the different contents such that the network's utility is maximized; and (4) reconfigure the network to dynamically enforce these allocations over time. Each of these steps represents a research challenge. The most pressing challenges involve designing efficient algorithms for allocating bandwidth to applications while maximizing end user experience and developing a practical framework for home networks.

Contextual Router builds on the adoption of Software Defined Networks (SDN), specifically SDN's fine-grained control and ease of automation which allows us to: (1) scalably and efficiently monitor individual flows and thus capture the user's contexts and (2) to enforce bandwidth allocations for individual applications. Unlike prior approaches, Contextual Router builds on the well-established quantitative models for capturing application utility as a function of network bandwidth.

In this paper, we present one of the first formulations for home networks that simultaneously determines bandwidth allocations and priorities in a dynamic fashion. Unlike prior approaches to resource allocation in home networks that assign priorities or allocations in an ad hoc manner [14, 21, 23, 28, 33], our formulation uses existing utility functions, user preferences, and contexts to make principled and systematic decisions. Our formulation allows us to holistically reason about optimality within home networks.

In this paper, we take the first step towards defining the design principles for Contextual Router and propose a straw man architecture that explores a point in the design space of Contextual Routers.

[1]The context can be easily extended to include other aspects of the user-device interactions if available.

SOSR '16, March 14-15, 2016, Santa Clara, CA, USA

DOI: http://dx.doi.org/10.1145/2890955.2890972

Our architecture and the prototype realizing its implementation illustrate the benefits of Contextual Router and are an existence proof of a practical Contextual Router. In this paper, we make the following contributions:

- **Framework for Contextual Router:** We define an architecture for supporting Contextual Router that integrates programmatic control of SDNs with recent advances in modeling and capturing application specific utility functions.

- **Resource Allocation Algorithm:** We present a formulation for systematically allocating resources to different applications in the home based on contexts.

- **Implementation and Evaluation:** We present an initial prototype and demonstrate the feasibility of employing Contextual Router within the home. Our experiments show that the Contextual Router can improve the QoE in typical home networks.

Roadmap. In § 2, we describe the characteristics of modern home networks and the design goals for Contextual Router. Then, we present the design space for Contextual Router in § 3, formulate the resource allocation problems in § 4, and discuss a prototype implementation in § 5. § 6 includes a basic evaluation of our prototype implementation. Related works are examined in § 7 and open issues are discussed in § 8. We present concluding remarks and future works in § 9.

2. MOTIVATION

In this section, we describe the relevant characteristics of home networks relative to application performance and use them to motivate the need for Contextual Router.

Large Number of Networked Applications: Home users own a multitude of devices, with each device running some networked applications: several of which run in the background with little interaction with the user. Regardless, all applications compete for the network's resources as well as the user's attention. Fortunately, users have limited attention and can only engage with a limited number of applications. Yet, the network equally divides resources amongst the applications.

We argue, as in [23], that the network should divide resources in a manner that aligns with the user's attention: applications being interacted with should get more resources than those in the background.

Diverse Expectations: To further complicate matters, users have different expectations across different devices. For example, higher quality is expected on smart TVs, while lower quality is tolerated on mobile devices. Yet, many applications are unaware of these expectations and attempt to deliver the same, high quality, across all platforms.

We argue that the network should coordinate across all applications to ensure that each application delivers acceptable or high QoS without suffering from diminishing returns.

Complex and Inadequate Knobs: Although application performance and QoS is dear to the heart of many users; these users are helpless to control either. Specifically, home networks provide users with low-level knobs that most users are unable to succinctly express an application's complex requirements regarding priorities, bandwidth or port bindings.

We argue, that user should express qualitative metrics, rather than quantitative values, that appropriately capture their expectation; for example, relative priorities and minimum or maximum bandwidth bit rate preferences, or video resolutions for Netflix rather than bandwidth requirements and absolute priorities.

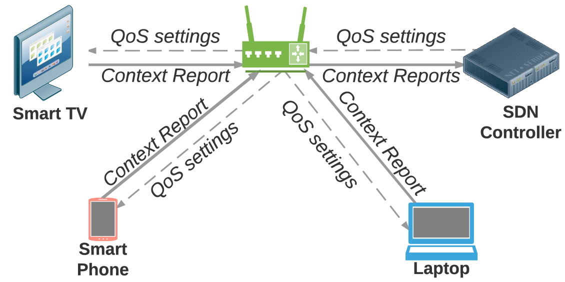

Figure 1: Architecture of a Contextual Router

Takeaway. In general, home users run a multiple of networked applications and expect them to deliver varying QoS depending on the context. Further, user expectations are such that there is a notion of diminishing returns for certain contexts. While home networks provide knobs for controlling application performance, these knobs are too low level and are inadequate for defining these contexts or expressing the rich space of user expectations. Finally, care must be taken to ensure applications are allocated resources (i.e. bandwidth) so that wasting resources is avoided.

3. ARCHITECTURE

The Contextual Router architecture, shown in Figure 1, consists of the following four components: *Contextual Portal* (not shown), a web GUI that captures a user's QoS preferences and priorities; *Contextual Monitors* which detects an application's footprint and periodically reports them to the Controller as "context reports"; *Contextual Router*, an SDN-enhanced home gateway that enforces resource allocations (e.g. Bandwidth limits), and, a *Contextual Controller* which periodically solves the optimization problem and reconfigures the network or applications to reflect the resulting resource allocations.

3.1 Contextual Portal

As discussed earlier, since users' preferences of applications and their quality expectations in different contexts vary, it is important to provide a simple graphical configuration interface to the end user. For a Contextual Router, the user might provide his priorities of different applications, limit maximum bandwidth to a traffic class or a specific application in a device, or ensure minimum bandwidth guarantees to certain applications. Figure 2 shows a simple example of how such an interface might look like. Note that this is just a mock interface, and designing a good interface which not only provides easy to use knobs for most users but also more advanced features for the more tech savvy home users is challenging and interesting on its own.

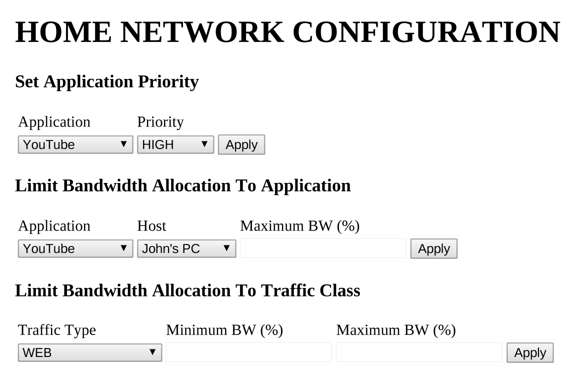

Figure 2: Graphical User Interface Design

3.2 Contextual Routers

Contextual Routers (referred to as Router) are enhanced home-gateways with SDN and potentially NFV functionality. The sole purpose is to enable enforcement of rate-limits and priorities. To realize this enforcement, the contextual routers include primitives for enqueuing packets and for rate limiting the different queues.

Enforcing priority and rate-limits for uplink traffic, traffic out of the home, is relatively straightforward – dropping packets and using queues. Unfortunately downlink traffic, traffic into the home, is harder – dropping downlink traffic at the home network does not ensure isolation of the bottleneck resource (the link between the gateway and the Internet service provider (ISP)). To enforce policies on downlink traffic, there are several design choices:

Transport Level: The router modifies TCP parameters, such as receive window, to force the source of traffic to slow down. This requires the least amount of cooperation but is limited to TCP flows and takes, at least, an RTT to enforce policies.

Network Level: The router could offload policy enforcement of downlink traffic to the ISP – the ISP can perform policy enforcement upstream of the bottleneck link. This requires the ISP to provide appropriate abstractions and raises questions about incentives and costs.

Application Level: Orthogonally, the enforcement can be performed by cooperating applications rather than the router. This approach requires modification and trust of the cooperating applications.

3.3 Contextual Monitors

The contextual monitor (referred to as monitors) captures "contextual reports" (referred to as reports) summarizing each user's interaction with the applications running on the various devices in the home. The granularity and coverage of these reports are impacted by the specific design choice and the amount of cooperation between the network and the applications.

User Library for Applications: At one extreme, the applications cooperate with network directly, through a set of libraries. The applications can specify and express all available services and their bandwidth requirements. For example, a video application can specify available bit rates and bandwidth requirements. Given this level of cooperation, more information can easily be obtained: e.g. is the user actively interacting with the application? Unfortunately, this approach requires rewriting the different applications.

A Network Function: At the other extreme, there is absolutely no cooperation between the application, devices and the network. In this situation, we employ deep packet inspection (DPI) to classify flows to applications and require specialized algorithms to differentiate background behavior from foreground behavior. This approach has the benefit of being the most general approach but provides coarse-grained information and little meta-information about the user interaction.

An OS daemon: In the middle, the OS rather than the application cooperates with the network and a daemon is installed on the user's device. This daemon captures both the set of applications and also annotates applications as either foreground or background applications. Moreover, this approach allows the monitoring agents to accurately map applications to network flows. Unfortunately, this approach is limited to the set of open devices in which applications can be installed; e.g. laptops, cell phones, and tablets.

3.4 Contextual Controller

Contextual Controller (referred to as the controller) is an SDN application which manages all traffic in the home network. The controller periodically receives reports, extracts contexts, and runs a resource allocation algorithm to determine the priority and bandwidths for the different contexts (application-device pair).

The resource allocation algorithm must account for the design choices made in the design of the router or the monitor. Specifically, the granularity of policy enforcement by the router and of meta information collection by the monitor. Regardless, the controller needs to install appropriate flow rules and change QoS settings in the Contextual Router for enforcing the bandwidth allocations. The resource allocation algorithm presents many interesting theoretical challenges as we discussed in Section 4.

4. RESOURCE ALLOCATION FOR HOME NETWORKS

In this work, we approach the problem of improving application performance as a general resource allocation problem and formulate an optimization problem that assigns network upload and download bandwidth to different applications based on user preferences and application demands. The objective of the formulation is to maximize the total utility delivered by all applications. Next, we discuss the inputs, formulation, and outputs for the optimization problem.

Formulation Input: The resource allocation algorithm takes as input:

- **Priorities:** A set of priorities; user defined priorities and QoS-preferences, and the priorities captured by the Contextual Monitors (low for background applications and high for foreground applications). The user defined priorities (Low,Medium,High) and inferred priorities (Low, High) are translated into empirically determined coefficients $(1.0, 1.1, 1.2)$ and $(1.0, 1.1)$ respectively. We combine the different priorities for each context by multiplying the user's coefficients with the Monitor's to get an aggregate value of priority.
- **Utility Functions:** The utility functions for each application capturing the bandwidth needs of applications [29]. To support contexts, we specifically pre-define different utility functions for different contexts (e.g. mobile device, PC) for a set of applications, such as, Web browsing, video streaming, VoIP and so on. Figure 3 shows two defined utility functions for file transfer (elastic type) and Netflix (stepwise type) applications. While defining the utility functions, we approximated the commonly known utility functions (such as for rate adaptive and elastic applications) with piecewise linear functions. Moreover, we used recommendations of various application/-content providers for high quality user experience [25, 30], and knowledge of the bit rates of encodings for video streaming applications. Obviously the use of utility functions for bandwidth allocation is not a new idea. Similar to our approach, a recent work uses utility functions and solves an optimization problem to achieve fairness among competing video streams [13]. Piecewise linear utility functions were used in [7] to ensure fairness among competing applications in an ABR service.

Problem Formulation: Using the utility functions, holistic priorities, and additional constraints obtained from the web portal (e.g. per-device or per-user bandwidth limits), the formulation of a typical optimization problem is as follows:

$$\max c^T u$$

subject to:

$$u_i = f_i(b_i), \forall i$$
$$\sum_i b_i \leq B,$$
$$\sum_{j \in C} \leq B_C$$

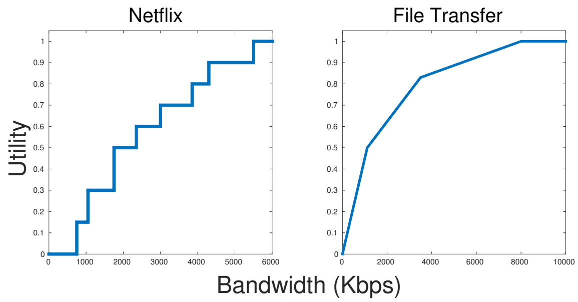

Figure 3: Defined Utility Functions for Netflix and File Transfer

where B is the total available bandwidth, u_i is the utility of application i, b_i is the bandwidth to be assigned to application i, $f_i()$ is the utility function i and $\vec{c}$ is the vector of weights(priorities). The last constraint bounds total bandwidth allocation to traffic class C to B_C. Note that the LP formulation is flexible and can encode other types of constraints from the web portal. Such constraints might be used to limit the bandwidth used by a device, or a set of devices belonging to a single user in the home.

Formulation Output and Postprocessing: The output of the LP is a set of bandwidth allocations to each application (based on the contexts). network. Unfortunately, home networks can have tens of such applications and home routers are not able to support as many queues. Thus, to scale, we aggregate applications into traffic classes by summing up their bandwidth allocations: VIDEO, WEB, FILE_TRANSFER and VOICE. Also, we specify minimum bandwidth constrains to the aggregate traffic classes to ensure that there is available bandwidth to prevent starvation.

5. PROTOTYPE IMPLEMENTATION

Next, we present our prototype implementation and discuss the design choices made and their implications when appropriate.

5.1 Contextual Monitor

The monitors are developed as OS level daemons (1041 lines of Python code) which are configured to send periodical reports to the home gateway, which in turn forwards the reports to the controller. We chose to have the monitors send reports to the gateway rather than the controller to minimize configuration of the monitors. For the latter, the monitors need to discover the address of the controller, whereas all devices discover the address of the gateway as part of DHCP.

The Context reports are generated by the monitors every second and they include: (1) the set of applications actively using the network and their flows; (2) interactivity information about each application (whether in the foreground or background); and (3) multimedia utilization of each application (whether the application is using the audio drivers or graphics drivers) – this information ensures these applications are not treated as background traffic even when their windows are minimized.

5.2 Contextual Router

The contextual routers are built atop Bismark routers [31]: each contextual router is an OpenWrt router running OpenVSwitch [26]. Unfortunately, the OpenFlow version running on Bismark does not support control over bandwidth allocation and priorities

To enable programmatic control over bandwidth allocation and priorities, we developed an application (277 lines of C code and

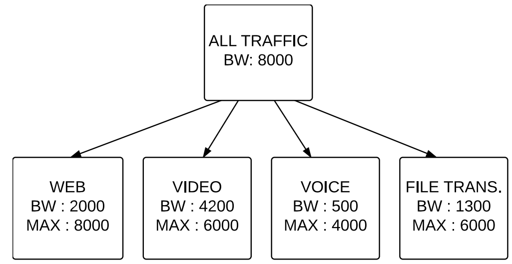

Figure 4: HTB classes and example bandwidth allocations

139 lines of shell script), `qos_manager`, that runs on the routers along side OpenVSwitch. This application uses Linux **tc** (traffic control) utility and HTB (Hierarchical Token Buffer) scheduler to enforce bandwidth limits (through rate-limiting) and priorities (through queues). To ensure work conservation, the HTB discs are configured with token-buckets which allows different traffic classes to temporarily use spare capacity that is currently being wasted by other traffic classes.

`qos_manager` consists of two modules: The first, the shell script, which runs during router initialization and sets up an HTB qdisc (queuing discipline), creates HTB classes on the queues corresponding to the supported traffic classes and attaches the queue to the WAN interface of the router. The second, a C program, listens for QoS settings from the controller and appropriately reconfigures **tc** to change bandwidth allocations and HTB qdisc to change priorities. For example, in our evaluations we setup four traffic classes: WEB, VIDEO, VOICE and FILE_TRANSFER traffic and in Figure 4, we show the HTB qdisc configuration for the different classes.

From this figure, we observe that the web class is allocation 2000KB but allowed to use up to 8000KB when the other classes are idle. Each HTB class corresponding to different types of traffic are added as a child of a root class, which enables this bandwidth borrowing. Note that the HTB class hierarchy is work conserving and thus better utilizes unused bandwidth than prior approaches that use two virtual switches to enforce bandwidth limits [3,28]. Also the HTB clases are flexible because we can easily scale up and down the number of internal nodes to reflect the number of different devices and users. The HTB class hierarchy allow us to capture complex custom constraints from the configuration portal, map them into LP constraints and enforce them at a fine granularity.

5.3 Contextual Controller

We implemented Contextual Controller as an SDN application on the Floodlight controller [1] (3521 lines of Java code). The SDN application operates in two modes. In start-up mode, the controller retrieves user's preferences and settings from the web-portal; and installs default forwarding rules and rules for forwarding Contextual reports to the controller.

During normal operation, the controller solves the LP for bandwidth allocations, aggregates bandwidth for different traffic classes and reconfigures the network with the new QoS settings by exchanging two types of custom messages with the router.

- *FILTER* messages include the name of a traffic class and information identifying a flow (source, destination IP addresses, and port numbers), and is used to define **tc** filters in the router, so that each flow will use the appropriate queue for its corresponding class.
- *BW* messages result in changing the bandwidth allocation of each traffic class (an example allocation is shown in Figure 4).

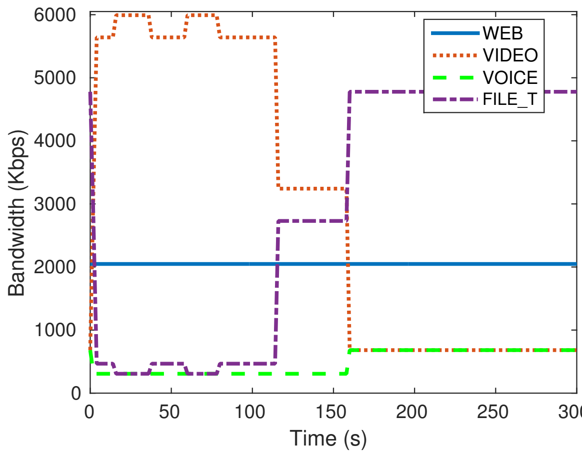

Figure 5: Change of bandwidth for different traffic classes

6. EVALUATION

Next we present a preliminary evaluation of Contextual Router and, focus, specifically, on quantifying the improvements in QoE for common applications run in home networks.

6.1 Testbed and Experimental Setup

We evaluated Contextual Router on a small testbed consisting of two laptops and a Contextual Router. The home network's uplink is limited to 8 Mbps for download and 4 Mbps for upload.

We run three applications in this network. At the beginning of the experiments, the first laptop runs a background file transfer to a server outside of the home network (emulated using `iperf`) and a browser playing an HD quality YouTube video. Simultaneously, the second laptop plays another HD quality YouTube video. 15 seconds and 30 seconds into the experiment, the second laptop opens two browsers to load the Wikipedia and Facebook's home pages.

We automated the tests on both laptops using shell scripts and instrumented the laptops to record QoE metrics for the different applications. Specifically, for video we used a Google-Chrome extension called YouSlow to record YouTube QoE data [24], for web browsing (page load time) we captured the HAR file using Firebug with Firefox, and for the bulk download we recorded `iperf` logs. Due to a limitation with YouSlow, we do not compare video bitrates but rather resolution sizes.

We compare the Quality of Experience results from the different applications in a network employing Contextual routers against a network with traditional routers. For the network with Contextual routers, we assumed YouTube traffic is given high priority by the end users, and all other applications have medium priority. The contextual monitor places the file transfer (`iperf`) as low priority because it is in the background.

6.2 Bandwidth Allocations and Reallocation

In Figure 5 we present the change in total bandwidth allocation to different classes of traffic over time by the Contextual Router. We observe an instantaneous increase in the bandwidth allocation to the VIDEO class to 6Mbps (red line) – this occurs because two video sessions were started in the beginning. Furthermore, we observe a sharp decrease in the bandwidth allocations to the low priority file transfer (`iperf`) flow.

Unlike the VIDEO traffic which benefited from a huge ramp up, we observe that with the WEB traffic there is no increase in bandwidth allocation despite having both laptops load multiple websites. The WEB traffic experiences this different behavior because it has lower priority than the VIDEO traffic and the higher priority YouTube sessions prevented WEB traffic to obtain more bandwidth. We note that the WEB traffic maintain a constant rate of 25% because the system is designed to allocate a minimum of 25% of the available bandwidth to the WEB traffic. We make this exception for the WEB traffic because any flow which does not match any **tc** filters is classified as WEB traffic, which will happen until the controller receives a context report which includes that flow.

At around minute two, the first video session ends and the controller reconfigures the network and allocates more bandwidth to the File transfer application. Similarly, after the second video ends around the 160-second mark, more bandwidth is further allocated to file transfer.

Finally, we observe the network allocates bandwidth to the voice class despite the fact that our experiments include no voice application. Fortunately, this is not a problem as Contextual Router is designed to be work conserving and thus other traffic classes will be able to use the bandwidth allocated to the VOICE class.

	Number of Buffering Events	Total Buffering Time(s)	Number of Resolution Changes
Traditional			
Video 1	4	14	6
Video 2	5	23	7
Contextual Router			
Video 1	0	0	1
Video 2	0	0	1

Table 1: QoE with and without Contextual Router.

6.3 QoE Improvements

In this section, we compare the QoE metrics for the web (classified as WEB) and video streaming (classified as VIDEO) in Contextual Router against a traditional home network (Traditional).

Web Browsing: For WEB traffic, we examine the page load times and observe that when the traffic is not prioritized, Facebook and the Wikipedia landing pages took more than 45 seconds and 29.83 seconds to load respectively[2]. However, with Contextual Router, page load times were 3.83 and 4.90 seconds for Facebook and Wikipedia respectively — a significant improvement in page load times by a factor of 12X and 6X respectively.

Online Video Stream: Next, for online streaming, we examine two well-known QoE metrics: buffering events and the number of resolution changes (lower numbers are better). Table 1 presents these two QoE metrics for the two different scenarios. We observe a dramatic difference the values for these metrics – with Contextual Router allowing significantly more bandwidth to the VIDEO traffic and thereby reducing the number of buffering events and resolution switches down to 0 and one respectively. In the Contextual Router scenario, the single observed resolution change in both videos improve the quality: for Video 1 the quality goes from small to medium and for video two the quality goes from tiny to large. The traditional scenario provided significant degradation with Video One constantly flapping between tiny and small resolutions and Video 2 flapping between medium, large, small and tiny resolutions.

Takeaways For two of the most common applications in home networks, video and web traffic, we observe that Contextual Router is capable of improving relevant QoE metrics.

[2]we note that the Facebook page timed out in our orchestrator

7. RELATED WORK

Understanding Home Networks: Recent works have focused on understanding home networks [8–10, 15, 19, 31], specifically understanding: the performance characteristics of broadband networks; the impact of wireless routers on home users traffic; the usability of bandwidth rate limits in home networks; and the source of web-performance bottlenecks. The insights gained in these studies motivate us to reexamine the design of home networks, and more specifically to provide usable knobs that explicitly capture user preferences rather than bandwidth limits.

Home Traffic Prioritization: Prioritizing traffic in the home has been studied for a long time. The solutions proposed in the pre-SDN era include systems that work in the end hosts [4, 5], and systems relying on applications setting IP TOS field [22]. Martin and Feamster proposed a user-driven approach [23] to the problem with a prototype implemented in Click [20]. Priority is determined based on the activity of the user and the interactivity of the application; based on this, hosts and applications are classified as active or inactive. This work is similar in spirit to ours but it seems to be abandoned at early stages.

The work of Yiakoumis et al. [33] provides users with an option for statically provisioning their network for different applications and two specific add-ons to improve Skype and video streaming. A work with similar flavor to ours is FlowQoS, in which the users specify which applications should get high priority [28]. FlowQoS relies on packet inspection for flow classification, but the main difference with our work is again the static nature of priority assignments. The works [14, 21] also rely on the end user for prioritization, and they defined an API to negotiate the bandwidth allocation with the ISP. These works provide easy to use knobs to the end user, but the resulting prioritization methods do not adapt to changes in the traffic, i.e. they are static.

The recent work OpenSDWN leverages SDN to programmatically control wireless networks and in one of their use cases the authors showed that it can be used to prioritize traffic in a home network [27]. For this use case, OpenSDWN uses end user input for choosing high priority applications and uses deep packet inspection to identify flows belonging to those applications.

Contextual Router presents a logical evolution by allowing user to define more expressive priorities, i.e. contexts, that automatically adjust to changes in network conditions. Unlike prior approaches on prioritization, Contextual Router force applications to behave in a globally optimal fashion by placing bandwidth limits on them.

Capturing User Contexts: Our approaches for inferring user context by monitoring user interactions and by inspecting network traffic advances on previous research for characterizing user traffic [23, 27, 28]. We extend these approaches to account for the complex user contexts that arise due to user interaction with multimedia devices; such as videos or teleconference. Moreover, we overcome visibility issues introduced by TLS encryption to existing approaches by running agents at the end hosts.

QoE: Our work builds on recent efforts [2, 16, 17] to quantify and measure QoE for video, web, and real-time multimedia, and uses their models as input for our formulation (section 3). Recent attempts to improve QoE by developing application specific algorithms [13, 17, 34], and design new Internet architectures [18] attack an orthogonal space. These approaches attempt to improve individual applications where as Contextual Router attempts to improve the user's overall viewing experience across all applications.

Other Works on Home Networks: Orthogonal to home performance, others have looked at ways to simplify and improve security within home networks by: delegating to the cloud [12], and capturing special logs to enable troubleshooting [6]. Yet, others [11] have focused on managing complexity of orchestrating tasks across multiple devices by developing a home operating system.

8. DISCUSSION

Configuring Contextual Router Contextual Router's optimization formulation requires some parameters: e.g., coefficients and bandwidth constraints for different traffic classes. Ideally, these parameters will be specifically tuned for each user. Unfortunately, this approach is cumbersome. Instead, we observe that the number of different Internet service plans is limited – we suspect that parameters may be tuned for such groups. The user can then try out a couple and determine which fits his specific home setup.

Creating and Maintaining Utility Functions. Our application utility functions are empirically created by analyzing applications. Unfortunately even modest changes to application code and configurations may easily change the utility functions. Furthermore, given the diversity of applications, there will need to be a large number of these utility functions. We envision that the utility functions will be created and maintained in a marketplace by a collection of third parties in a similar to how antivirus signatures are created and maintained by a third party.

Tackling Performance Anomalies The Controller improves application performance by allocating the home gateway's uplink optimally between competing applications. This approach works when the uplink is the bottleneck. In certain conditions, applications performance is impacted by bottlenecks is other portions of the network, e.g. Wifi [27]. We envision Contextual Router will be run alongside other tools for diagnosing performance problems in home networks [32]. These tools can be used to help inform the bandwidth allocation decisions.

9. CONCLUSION AND FUTURE WORK

In this work, we argue for more principled approaches to improving the QoE of applications within home networks. We explore the use of utility functions to capture bandwidth requirements of applications and develop a formulation that combines these utility functions with user preferences and inferred application priorities. We developed an initial prototype of Contextual Router and demonstrated that our approach improves QoE with negligible overheads.

Our work presents an attempt to apply sound theoretical principles to the improvement of home networks. As part of ongoing work we are (1) exploring more intuitive approaches to capturing user preferences, and contexts, (2) analyzing our prototype under diverse settings, (3) designing abstraction to enforce control over downstream traffic, and (4) investigating different deployment scenarios and their implications on the scalability of the system – specifically understand how increasing the number and heterogeneity of the applications impacts the run time of our LP and reactiveness of our prototype.

Acknowledgements

We thank Nick Feamster for providing us the Bismark router and M. Said Seddiki for his valuable technical support for getting started with the router configuration. We thank Yilun Zhou for the implementation of Contextual monitors. We also thank the anonymous referees for their constructive comments.

10. REFERENCES

[1] Floodlight v1.0. http://www.projectfloodlight.org/floodlight/.

[2] BALACHANDRAN, A., SEKAR, V., AKELLA, A., SESHAN, S., STOICA, I., AND ZHANG, H. Developing a predictive model of quality of experience for internet video. In *Proceedings of the ACM*

SIGCOMM 2013 Conference on SIGCOMM (New York, NY, USA, 2013), SIGCOMM '13, ACM, pp. 339–350.

[3] BOZKURT, I. N., ZHOU, Y., AND BENSON, T. Dynamic prioritization of traffic in home networks using software-defined networking. In *CoNext Student Workshop* (2015).

[4] BRANDT, S., NUTT, G., BERK, T., AND HUMPHREY, M. Soft real-time application execution with dynamic quality of service assurance. In *Quality of Service, 1998. (IWQoS 98) 1998 Sixth International Workshop on* (May 1998), pp. 154–163.

[5] BRANDT, S., NUTT, G., BERK, T., AND MANKOVICH, J. A dynamic quality of service middleware agent for mediating application resource usage. In *Real-Time Systems Symposium, 1998. Proceedings., The 19th IEEE* (Dec 1998), pp. 307–317.

[6] CALVERT, K. L., EDWARDS, W. K., FEAMSTER, N., GRINTER, R. E., DENG, Y., AND ZHOU, X. Instrumenting home networks. *SIGCOMM Comput. Commun. Rev. 41*, 1 (Jan. 2011), 84–89.

[7] CAO, Z., AND ZEGURA, E. Utility max-min: an application-oriented bandwidth allocation scheme. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Mar 1999), vol. 2, pp. 793–801 vol.2.

[8] CHETTY, M., BANKS, R., HARPER, R., REGAN, T., SELLEN, A., GKANTSIDIS, C., KARAGIANNIS, T., AND KEY, P. Who's hogging the bandwidth: The consequences of revealing the invisible in the home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 659–668.

[9] CHETTY, M., AND FEAMSTER, N. Refactoring network infrastructure to improve manageability: A case study of home networking. *SIGCOMM Comput. Commun. Rev. 42*, 3 (June 2012), 54–61.

[10] CHETTY, M., HASLEM, D., BAIRD, A., OFOHA, U., SUMNER, B., AND GRINTER, R. Why is my internet slow?: Making network speeds visible. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, ACM, pp. 1889–1898.

[11] DIXON, C., MAHAJAN, R., AGARWAL, S., BRUSH, A., LEE, B., SAROIU, S., AND BAHL, P. An operating system for the home. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)* (San Jose, CA, 2012), USENIX, pp. 337–352.

[12] FEAMSTER, N. Outsourcing home network security. In *Proceedings of the 2010 ACM SIGCOMM Workshop on Home Networks* (New York, NY, USA, 2010), HomeNets '10, ACM, pp. 37–42.

[13] GEORGOPOULOS, P., ELKHATIB, Y., BROADBENT, M., MU, M., AND RACE, N. Towards network-wide qoe fairness using openflow-assisted adaptive video streaming. In *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking* (New York, NY, USA, 2013), FhMN '13, ACM, pp. 15–20.

[14] GHARAKHEILI, H. H., BASS, J., EXTON, L., AND SIVARAMAN, V. Personalizing the home network experience using cloud-based sdn. In *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on* (2014), IEEE, pp. 1–6.

[15] GROVER, S., PARK, M. S., SUNDARESAN, S., BURNETT, S., KIM, H., RAVI, B., AND FEAMSTER, N. Peeking behind the nat: An empirical study of home networks. In *Proceedings of the 2013 Conference on Internet Measurement Conference* (New York, NY, USA, 2013), IMC '13, ACM, pp. 377–390.

[16] HUANG, T.-Y., HANDIGOL, N., HELLER, B., MCKEOWN, N., AND JOHARI, R. Confused, timid, and unstable: Picking a video streaming rate is hard. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference* (New York, NY, USA, 2012), IMC '12, ACM, pp. 225–238.

[17] HUANG, T.-Y., HUANG, P., CHEN, K.-T., AND WANG, P.-J. Could skype be more satisfying? a qoe-centric study of the fec mechanism in an internet-scale voip system. *Network, IEEE 24*, 2 (March 2010), 42–48.

[18] JIANG, J., LIU, X., SEKAR, V., STOICA, I., AND ZHANG, H. Eona: Experience-oriented network architecture. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2014), HotNets-XIII, ACM, pp. 11:1–11:7.

[19] KIM, H., SUNDARESAN, S., CHETTY, M., FEAMSTER, N., AND EDWARDS, W. K. Communicating with caps: Managing usage caps in home networks. *SIGCOMM Comput. Commun. Rev. 41*, 4 (Aug. 2011), 470–471.

[20] KOHLER, E., MORRIS, R., CHEN, B., JANNOTTI, J., AND KAASHOEK, M. F. The click modular router. *ACM Trans. Comput. Syst. 18*, 3 (Aug. 2000), 263–297.

[21] KUMAR, H., GHARAKHEILI, H. H., AND SIVARAMAN, V. User control of quality of experience in home networks using sdn. In *Advanced Networks and Telecommuncations Systems (ANTS), 2013 IEEE International Conference on* (2013), IEEE, pp. 1–6.

[22] LIU, G., ZHOU, S., ZHOU, X., AND HUANG, X. Qos management in home network. In *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on* (Nov 2006), pp. 203–203.

[23] MARTIN, J., AND FEAMSTER, N. User-driven dynamic traffic prioritization for home networks. In *Proceedings of the 2012 ACM SIGCOMM Workshop on Measurements Up the Stack* (New York, NY, USA, 2012), W-MUST '12, ACM, pp. 19–24.

[24] NAM, H., KIM, K.-H., CALIN, D., AND SCHULZRINNE, H. Youslow: A performance analysis tool for adaptive bitrate video streaming. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (New York, NY, USA, 2014), SIGCOMM '14, ACM, pp. 111–112.

[25] NETFLIX. Internet connection speed recommendations. https://help.netflix.com/en/node/306.

[26] PFAFF, B., PETTIT, J., KOPONEN, T., AMIDON, K., CASADO, M., AND SHENKER, S. e.a.: Extending networking into the virtualization layer. In *In: 8th ACM Workshop on Hot Topics inNetworks (HotNets-VIII).New YorkCity,NY(October 2009*.

[27] SCHULZ-ZANDER, J., MAYER, C., CIOBOTARU, B., SCHMID, S., AND FELDMANN, A. Opensdwn: Programmatic control over home and enterprise wifi. In *SOSR* (2015).

[28] SEDDIKI, M. S., SHAHBAZ, M., DONOVAN, S., GROVER, S., PARK, M., FEAMSTER, N., AND SONG, Y.-Q. Flowqos: Qos for the rest of us. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking* (New York, NY, USA, 2014), HotSDN '14, ACM, pp. 207–208.

[29] SHENKER, S. Fundamental design issues for the future internet. *IEEE J.Sel. A. Commun. 13*, 7 (Sept. 2006), 1176–1188.

[30] SKYPE. How much bandwidth does skype need? https://support.skype.com/en/faq/fa1417/how-much-bandwidth-does-skype-need.

[31] SUNDARESAN, S., BURNETT, S., FEAMSTER, N., AND DE DONATO, W. Bismark: A testbed for deploying measurements and applications in broadband access networks. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference* (Berkeley, CA, USA, 2014), USENIX ATC'14, USENIX Association, pp. 383–394.

[32] SUNDARESAN, S., FEAMSTER, N., AND TEIXEIRA, R. Locating throughput bottlenecks in home networks. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (New York, NY, USA, 2014), SIGCOMM '14, ACM, pp. 351–352.

[33] YIAKOUMIS, Y., KATTI, S., HUANG, T.-Y., MCKEOWN, N., YAP, K.-K., AND JOHARI, R. Putting home users in charge of their network. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (New York, NY, USA, 2012), UbiComp '12, ACM, pp. 1114–1119.

[34] YIN, X., SEKAR, V., AND SINOPOLI, B. Toward a principled framework to design dynamic adaptive streaming algorithms over http. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2014), HotNets-XIII, ACM, pp. 9:1–9:7.