Contextual Networks: Advancing Experience Oriented Networking to the Home

Author List

ABSTRACT

In a home network, there are multiple users each running different applications interacting with the network. To enhance the experience of each user, prioritization of various network applications is important.

Previous solutions to this problem assigned priorities in a static manner. Even though there has been some efforts to assign priorities dynamically, these solutions only used interactivity of the application to prioritize traffic. We present Contextual Router, which achieves better prioritization by detecting all the flows generated in a home network and assigning priorities in a dynamic manner using various features of flows collected from each user's machine.

1. INTRODUCTION

Recently, there has been a surge in interest on management of home-networks. At the root of this interest are the following two trends: first, an increase in networked devices, 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 provider 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 [14, 31].

Recent efforts to improve performance have focused on application specific optimization [15] or on home router queuing/scheduling disciplines [22, 26]. These efforts are either limited in scope, are static, or are agnostic of 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 SmartTV or when watching Netflix versus performing an operating system update. Further, these expectations of quality are relatively 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 the set of applications and device pairs that are actively using the network.

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

We present one of the first formulations for home networks that simultaneously determines bandwidth allocation and priorities in a dynamic fashion. Unlike prior approaches to resource allocation in home networks [13, 20, 22, 26, 33] that assign priorities or allocations in an ad hoc manner, our formulation uses existing QoE models based on utility functions and contexts to make principled and systematic decisions. This formulation allows us to reason holistically about optimality within home networks.

Our framework builds on a number of interesting trends: first, the adoption of Software Defined Networks (SDN), specifically the fine-grained control and automation provided by the SDN paradigm, and second, the recent development of quantitative models for capturing QoE for different applicationd. SDNs allow 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. The application model allows us to accurately determine the bandwidth requirements, allocations, and priorities.

In this paper, we take the first step towards defining the design principles of Contextual networks and propose a strawman architecture that explores one area in the design space of contextual networks. Our architecture and the prototype realizing its implementation illustrate the benefits of contextual networks and are an existence proof of a practical contextual network. In this paper, we make the following contributions:

- Framework for Contextual Networks: We define an architecture for supporting contextual networks that integrates programmatic control of SDNs with recent advances in modeling and capturing QoE.

- 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 employing Contextual Networks within the home. Our experiments show that the Contextual Networks are able to improve the QoE in typical home networks.

2. RELATED WORK

Understanding Home Networks: Recent works have focused on understanding home networks [9, 14, 19, 30], specifically understanding: the performance characteristics of broadband networks; the impact of wireless routers on home users traffic; the usability of BW rate limits in home networks; and the source of web-performance bottlenecks. The insights gained in these studies motivates us to re-examine the design of home networks and more specifically to provide usable knobs that explicitly capture user preferences rather than bandwidth limits.

Home Traffic Prioritization: The most closely related works [18, 26] allow users to specify priorities for different applications and then uses deep packet inspection to characterize flows into application classes for prioritization. Others [6, 7, 13, 20–22, 33] attempts at prioritization either require applications to come with built-in priorities or infer priority by monitoring end-user interactions. Contextual Networks 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 [6, 7, 18, 21, 22, 26], Contextual Networks forces applications to behave in a globally optimal fashion by placing bandwidth limits on them.

Another recent work uses utility functions and solves an optimization problem similar to our work for achieving fairness among competing video streams [12].

Capturing User Contexts: Our approaches for inferring user context by monitoring user interactions and by inspecting network traffic advances on a long line of research for characterizing user traffic [18, 22, 26]. We extend these approaches to account for the complex user contexts that arise due to user interaction with multi-media devices; such as videos or teleconference. More over, we overcome visibility issues introduced by TLS encryption to existing approaches by running agents at the end-points.

QoE: Our work builds on recent efforts [4, 16] to quantify and measure QoE for video, web, and real-time multimedia and uses their models as input for our formulation (section 4). Recent attempts to improve QoE by developing application specific algorithms [12, 15, 16, 34] and design new internet architectures [17] attack an orthogonal space. These approaches attempt to improve individual applications where as Contextual Networks attempts to improve the user's overall viewing experience across all applications.

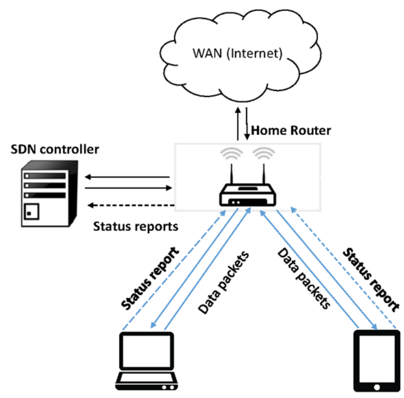

Figure 1: System Architecture

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 [11], capturing special logs to enable troubleshooting [8], or by introducing software defined middleboxes [28, 32]. Yet, others [10] have focused on managing complexity of orchestrating tasks across multiple devices by developing a home operating system.

3. DESIGN GOALS

Based on the problems of previous efforts addressing traffic management in the home, we aimed to reach the following goals while designing Contextual Router.

- **Minimal interaction with the user:** The system should require minimal interaction with the user. During an initial setup, the user might provide preferences for different applications but the system should not rely on any user action during operation.

- **Achieves dynamic prioritization:** Depending on the running applications, the bandwidth reserved for certain applications should be changed dynamically. The system must make use of the available bandwidth in an efficient manner.

- **Small overhead:** The system should use minimal network and computational resources. The overhead of running Contextual Router in the home should be negligible.

- **Usefulness:** Obviously our main goal is improving the users' experience in the home network. The QoE improvement should be visible.

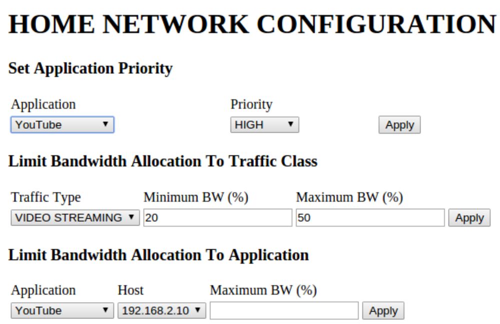

Figure 2: Graphical User Interface Design

4. SYSTEM DESIGN

We approach the traffic prioritization problem by treating it as a resource allocation problem. The resource to be shared is the available upload and download bandwidth of a typical home network. An optimization problem which maximizes the total utility of the home network is solved periodically to compute bandwidth allocations of active applications. The optimization problem takes user preferences, inferred relative priorities and the bandwidth needs of detected applications into account.

Contextual Networks architecture (Figure 1), which realizes our proposed solution, consists of 3 components : *Contextual Router*, an SDN-enhanced home gateway with the capability to enforce flow priorities, *Contextual Monitors* which periodically capture and report contexts, and, *Contextual Controller* which periodically recomputes bandwidth allocations based on the reported contexts, and reconfigures the network to reflect resource allocation decisions.

Since QoE is subjective and user's preferences of different applications vary, it is important to provide a simple graphical configuration interface to the end user. For 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.

4.1 Contextual Routers

These are enhanced home-gateways with SDN and potentially NFV functionality. The sole purpose is to enable enforcement of rate-limits and of priorities. To this extend, the contextual routers include primitives for enqueuing packets in queues and for rate limiting different queues.

4.2 Contextual Monitors

In principle, the contextual monitor captures information about the users' interaction with various devices in the home. At a coarse granularity, it captures the applications that the user is actively using on each device. Each application and device pair is then reported as a context. In practice, the contextual monitors can be realized in the following ways:

- **An OS daemon:** that is installed on all user devices. This allows the monitoring agent to both capture the set of applications and also to annotate 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 device in which applications can be installed; e.g. laptops, cell phones, and tablets.

- **A Network Function:** is a more general approach that is applicable to both open and closed devices. This approach uses network functions, e.g. DPI, to classify flows to applications and requires specialized algorithms to differentiate background behavior from foreground behavior. Moreover this approach requires special key exchanges to ensure that TLS encrypted traffic can be inspected by the network functions.

4.3 Contextual Controller

Contextual Controller (henceforth referred to as controller) is an SDN application which manages the traffic in the home network. Contextual reports are processed periodically to determine active network applications and their flows. The reports also include information to determine relative priority of each application which is fed to the resource allocation algorithm along with user preferences. The bandwidth allocation to different applications also takes bandwidth needs of distinct applications into account. Moreover, bandwidth requirements of applications also differ based on the type of the device the application is running on, so context is also taken into account.

Our design allows enforcing bandwidth allocation on a per application basis, but it is also possible to aggregate bandwidth for a small number of traffic classes for the same purpose. Either way, the controller needs to install appropriate flow rules and change QoS settings in the Contextual Router for enforcing the bandwidth allocations. Different choices are possible for both tasks with pros and cons. We present our concrete implementation in the next section and explain our choice by comparing it with an alternative method which we used in our earlier prototype implementation.

5. PROTOTYPE IMPLEMENTATION

We next describe each component of our prototype implementation in detail.

5.1 Contextual Monitor

Out of convenience, we have prototyped the contextual monitors as agents running on users devices (989 lines of Python code). The monitors periodically (every second) send context-reports to the contextual router. Context reports include the following information:

- Active network applications and their flows (source, destination IP addresses and ports)

- The window on focus, minimized applications and percentage of the visible window size of each non-minimized application. This helps classify foreground and background traffic and detect the active application the user is interacting with.
- Applications accessing sound card and video camera. This information ensures these applications are not treated as background traffic even when their windows are minimized.

5.2 Contextual Controller

We implemented Contextual Controller as an SDN application atop Floodlight controller [2] (2990 lines of Java code). On start, the controller retrieves user's preferences and settings from the graphical user interface, installs default forwarding rules and also installs rules for the contextual reports so that the Contextual Router will forward the reports to the controller.

During normal operation, received reports are processed, bandwidth allocations are recomputed and the network is reconfigured periodically (currently every second). We describe bandwidth allocation next.

5.2.1 *Bandwidth Allocation*

Bandwidth allocation algorithm makes use of both application priorities and the bandwidth needs of each distinct application. Each detected application is classified as low priority or high priority based on the processed report. The coefficients 1.0 and 1.1 are used for low and high priority classes in our prototype. This is combined (by multiplication) with the user preferences for the application. Three different choices are given to the user with coefficients $1.0, 1.1$ and 1.2, corresponding to low, medium and high priorities respectively, with medium priority chosen for all applications for which the user did not make his preference known. Our current prototype does not yet utilize a user interface, so user preferences are simulated via configuration parameters in the Contextual Monitors.

We used utility functions to capture the bandwidth needs of applications [27] and pre-defined different utility functions for different contexts (e.g. mobile device, PC) for a set of applications such as Web browsing, file transfer, YouTube and so on. The appropriate utility function is chosen for each detected application for bandwidth allocation. In defining the utility functions, we approximated the commonly known utility functions with piece-wise linear functions as is done in [**?**]. Moreover we used recommendations of various network application/content providers for high quality user experience [24, 29] and knowledge of the bitrates of encodings for video streaming applications. Figure 3 shows two defined utility functions for file transfer (elastic) and Netflix (stepwise) applications.

Using the utility functions, priorities and the additional constraints obtained from the user interface, the formulation of a typical optimization problem is as follows:

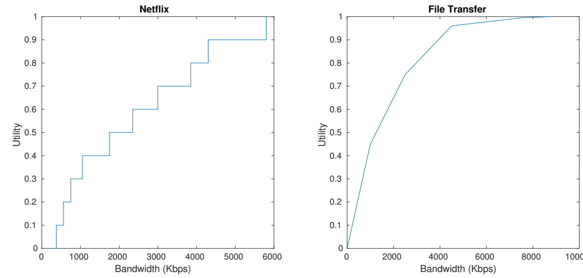

Figure 3: Defined Utility Functions for Skype and Netflix

$$\max c^T u$$

subject to:

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

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.

After an LP is solved, aggregated bandwidths are computed for the following traffic classes : Video, Web Browsing, File Transfer and Voice (by summing the bandwidths of corresponding applications). The aggregation makes sure that for each traffic class, a minimum amount of bandwidth is available even when there are no active applications in that class.

5.2.2 *Enforcing Bandwidth Allocations*

Due to limitations of OpenFlow protocol for changing QoS settings and bandwidth aggregation based on different classes of traffic, Contextual Controller enforces bandwidth allocations in the Contextual Router by exchanging messages with a custom developed application (called *qos_manager*) running inside the router. Two types of messages are defined: QOS messages and FLOW messages. QOS messages are used to change the bandwidth allocation of different traffic classes. QOS message includes the names and bandwidth allocations of the defined traffic classes. Upon receving a QOS message, *qos_manager* applies appropriate **tc** commands for changing bandwidth allocations.

In an earlier version of our prototype [5], QOS messages were the only type of messages that need to be exchanged since we were using a dual virtual switch architecture as in [26] and using OpenFlow to direct flows to different links corresponding to different traffic classes for prioritization. Unfortunately this architecture wastes any available excess bandwidth since each virtual link is rate limited and each flow has to go through one of them. In our current prototype, we

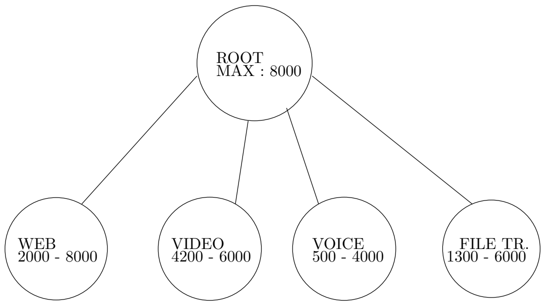

Figure 4: HTB classes and example bandwidth allocations

apply traffic shaping on a single interface using **tc** with Linux HTB(Hierarchical Token Bucket) scheduling discipline which enables us to define different classes of traffic and bandwidth sharing between different classes if any extra bandwidth is available. Traffic classes are defined using **tc** filters in the router and attached to the HTB scheduler. The Contextual Controller sends FLOW messages (which identifies a standard flow tuple and traffic class) to the router so that *qos_manager* can install appropriate **tc** filters for the flows which in turn will enable HTB scheduler to enforce bandwidth allocations. When a flow does not match any defined filter, it is classified as WEB traffic. To minimize the number and frequency of message exchanges, Contextual Controller keeps track of an internal flow table and only sends FLOW messages to the router the first time a flow is detected, and not after the same flow is detected later in the periodical context reports.

5.3 Contextual Router

The contextual routers are built atop Bismark routers [30]: essentially each contextual router is an OpenWrt router running OpenVSwitch [25]. Unfortunately, due to limitations of OpenFlow 1.0, we developed scripts to run on the routers to enable programmatic control over queues and rate-limiting. The configuration script sets up one or two HTB qdiscs and attaches it to incoming and/or outgoing network interfaces depending on the configuration. HTB classes corresponding to WEB, VIDEO, VOICE and FILE_TRANSFER traffic are created as well and attached to the HTB qdisc. Each class has a rate limit and ceil rate which determines the amount of bandwidth borrowing in case of spare capacity. An example configuration is shown in figure 4, which shows the root class which enables its children to borrow available excess bandwidth.

`qos_manager` application runs on Contextual Router and changes the bandwidth allocations, and defines `tc` filters based on the received QOS and FLOW messages received from Contextual Controller. `qos_manager` is developed in C (in 267 lines of code) and cross-compiled for the OpenWrt OS and Bismark router architecture.

6. EVALUATION

For a preliminary evaluation, we investigated whether ContextualRouter changes the allocation of bandwidth to different types of traffic in a meaningful way and tested QoE improvement in a simple test scenario. In this test, two users are connected to the Internet through a Bismark router controlled by our system. The client agents are configured to send reports in 1 second intervals and ContextualRouter is configured to process the reports and compute bandwidths at 2 second intervals. The available bandwidths in the home network is set to 8 Mbps for download and 4 Mbps for upload.

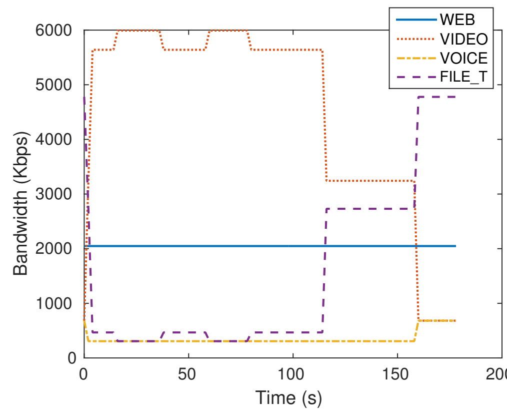

Figure 5: Change of bandwidth for different traffic classes

To see the benefits of our system under contention, we created a huge amount of traffic using `iperf` which is directed towards one test user, simulating a large file download. This user opens a HD quality YouTube video at the beginning of the test. The other user also starts viewing a YouTube video, and visits Wikipedia and Facebook homepages in the next minute. In reality all the tests and applications were started using shell scripts.

We recorded the QoE of video sessions and page load times of the visited websites when the home network is controlled by our system and also when there is no smart traffic prioritization. For our system, we assumed YouTube traffic is given HIGH priority by the end users and all other applications have MEDIUM priority. We used a Google-Chrome extension called YouSlow to record YouTube QoE data [23]. Page loads were orchestrated using Selenium and Firefox.

Figure 5 shows how ContextualRouter changed the total available bandwidth to different classes of traffic. Since two video sessions were started in the beginning, VIDEO bandwidth is immediately increased to 6 Mbps and the available bandwidth to the low priority file transfer decreased sharply. Even though two browsers were opened with 15 seconds in between, the bandwidth allocation to the WEB traffic remained constant. This is because we allocate a minimum %25 of the available bandwidth to the WEB traffic at all times (due to being the default class for any flows not matching any filters), and the high priority YouTube sessions prevented WEB class to obtain more bandwidth. Around 2 minute mark, the first video session ends and this enables File transfer application to grab more bandwidth. After the second video ends as well, bandwidth allocation to file transfer is increased further.

When the traffic was not prioritized, Facebook page took

	Num. Bufferring	Buff. Time	Res. Changes
No Prioritization			
	4	14	6
	5	23	7
Contextual Router			
	0	0	1
	0	0	1

Table 1: Video QoE. QoE of various video sessions

more than 45 seconds to load (at which point it timed out in our orchestrator) and Wikipedia home page was loaded in 29.83 seconds. Under control of Contextual Router, page load times were 3.83 seconds for Facebook and 4.90 seconds for Wikipedia, showing significant reduction in page load times.

Table 1 shows for the two scenarios, the number of times each video was buffered, total buffering time (in seconds) and the number of resolution changes. We see a dramatic difference between the two cases; Contextual Router allowed video sessions to obtain more bandwidth after detecting these flows, which led to one resolution change, and the sessions ended without experiencing any buffering. One video switched from small to medium, and the other switched from tiny to large. When there is no traffic prioritization, we see significant buffering and a large number of resolution changes. One video switched between tiny and small resolutions back and forth, for the other the changes were between medium, large, small and tiny resolutions (YouSlow extension, which uses YouTube player API is not able to report bitrates). For the video sessions too, we see that Contextual Router is capable of improving QoE.

Our tests included a small number of applications, hence the resulting optimization problems were also small (and simple), which helps obtaining bandwidth allocations in a few seconds. The scalability of the system with more constraints and more applications needs examination.

7. DISCUSSION AND ON-GOING WORK

The possibilities for improving the proposed design and prototype implementation are abundant. Combining our design with an NFV solution will help prioritize applications on a more diverse set of devices.

Our controller can be either inside the home network or in the cloud. Ideally the controller should reside in the ISP to control the access link, and our design works for this case with minimal changes. The home network can manage the upstream link.

A Thorough Evaluation for Optimization With many knobs for control, our prototype requires a thorough evaluation. Choice of the coefficient values based on the client reports, testing other types of utility functions and finding good ones is another direction for future work. It is easy to imagine with different characteristics of their Internet connections, different users might need different utility functions for the same applications. Moreover, more specific utility functions for different types of websites can also be defined. The controller might make use of a marketplace for utility functions chosen according to the Internet connection of the home network.

Handling Uplink Another future direction is handling the upload bandwidths. It is in fact necessary to define different utility functions for upload and download for the same application as applications vary in their characteristics. For example video streaming just needs to send back acknowledgements so it does not require much bandwidth, however one user uploading pictures to Facebook or Instagram can benefit from bandwidth guarantees in the uplink. Skype also needs a certain bandwidth for sending the sound and captured video frames.

8. CONCLUSION

In this work, we argue for more principled approaches to improving the QoE of applications within home networks. We explore the use of user defined utility function to guide resource allocations and develop a formulation that uses these utility functions and various QoE models in determining optimal allocation of resources. We develop an initial prototype of Contextual Networks and demonstrate that approach efficiency improves the QoE of users without 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 exploring more intuitive approaches to capturing user utilizing functions and analyzing our prototype under diverse settlings.

9. REFERENCES

[1] Extended statusbar. https://addons.mozilla.org/en-us/firefox/addon/extended-statusbar/.

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

[3] Youslow chrome extension. http://dyswis.cs.columbia.edu/youslow/.

[4] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang. Developing a predictive model of quality of experience for internet video. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 339–350, New York, NY, USA, 2013. ACM.

[5] I. N. Bozkurt, Y. Zhou, and T. Benson. Dynamic prioritization of traffic in home networks using software-defined networking. In *CoNext Student Workshop*, 2015.

[6] S. Brandt, G. Nutt, T. Berk, and M. Humphrey. Soft real-time application execution with dynamic quality of service assurance. In *IWQoS 98) 1998 Sixth International Workshop*, pages 18–20, 1998.

[7] S. Brandt, G. Nutt, T. Berk, and J. Mankovich. A dynamic quality of service middleware agent for mediating application resource usage. In *Real-Time Systems Symposium, 1998. Proceedings., The 19th IEEE*, pages 307–317, Dec 1998.

[8] K. L. Calvert, W. K. Edwards, N. Feamster, R. E. Grinter, Y. Deng, and X. Zhou. Instrumenting home networks. *SIGCOMM Comput. Commun. Rev.*, 41(1):84–89, Jan. 2011.

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

[10] C. Dixon, R. Mahajan, S. Agarwal, A. Brush, B. Lee, S. Saroiu, and P. Bahl. An operating system for the home. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 337–352, San Jose, CA, 2012. USENIX.

[11] N. Feamster. Outsourcing home network security. In *Proceedings of the 2010 ACM SIGCOMM Workshop on Home Networks*, HomeNets '10, pages 37–42, New York, NY, USA, 2010. ACM.

[12] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. 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*, FhMN '13, pages 15–20, New York, NY, USA, 2013. ACM.

[13] H. H. Gharakheili, J. Bass, L. Exton, and V. Sivaraman. 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*, pages 1–6. IEEE, 2014.

[14] S. Grover, M. S. Park, S. Sundaresan, S. Burnett, H. Kim, B. Ravi, and N. Feamster. Peeking behind the nat: An empirical study of home networks. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 377–390, New York, NY, USA, 2013. ACM.

[15] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari. Confused, timid, and unstable: Picking a video streaming rate is hard. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 225–238, New York, NY, USA, 2012. ACM.

[16] T.-Y. Huang, P. Huang, K.-T. Chen, and P.-J. Wang. Could skype be more satisfying? a qoe-centric study of the fec mechanism in an internet-scale voip system. *Network, IEEE*, 24(2):42–48, March 2010.

[17] J. Jiang, X. Liu, V. Sekar, I. Stoica, and H. Zhang. Eona: Experience-oriented network architecture. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, pages 11:1–11:7, New York, NY, USA, 2014. ACM.

[18] C. M. Julius Schulz-Zander, S. S. Bogdan Ciobotaru, and A. Feldmann. Opensdwn: Programmatic control over home and enterprise wifi. In *SOSR*, 2015.

[19] H. Kim, S. Sundaresan, M. Chetty, N. Feamster, and W. K. Edwards. Communicating with caps: Managing usage caps in home networks. *SIGCOMM Comput. Commun. Rev.*, 41(4):470–471, Aug. 2011.

[20] H. Kumar, H. H. Gharakheili, and V. Sivaraman. User control of quality of experience in home networks using sdn. In *Advanced Networks and Telecommuncations Systems (ANTS), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[21] G. Liu, S. Zhou, X. Zhou, and X. Huang. 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*, pages 203–203, Nov 2006.

[22] J. Martin and N. Feamster. User-driven dynamic traffic prioritization for home networks. In *Proceedings of the 2012 ACM SIGCOMM Workshop on Measurements Up the Stack*, W-MUST '12, pages 19–24, New York, NY, USA, 2012. ACM.

[23] H. Nam, K.-H. Kim, D. Calin, and H. Schulzrinne. Youslow: A performance analysis tool for adaptive bitrate video streaming. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 111–112, New York, NY, USA, 2014. ACM.

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

[25] B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, and S. Shenker. e.a.: Extending networking into the virtualization layer. In *In: 8th ACM Workshop on Hot Topics inNetworks (HotNets-VIII).New YorkCity,NY(October 2009*.

[26] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, and Y.-Q. Song. Flowqos: Qos for the rest of us. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 207–208, New York, NY, USA, 2014. ACM.

[27] S. Shenker. Fundamental design issues for the future internet. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 13:1176–1188, 1995.

[28] Y. Shirali-Shahreza Sajad; Ganjali. Software-defined firewall for home networks.

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

[30] S. Sundaresan, S. Burnett, N. Feamster, and W. De Donato. Bismark: A testbed for deploying measurements and applications in broadband access networks. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'14, pages 383–394, Berkeley, CA, USA, 2014. USENIX Association.

[31] S. Sundaresan, N. Feamster, and R. Teixeira. Locating throughput bottlenecks in home networks. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 351–352, New York, NY, USA, 2014. ACM.

[32] C. Taylor, Curtis; Shue. Home improvement: Transitioning enterprise network innovations to residential networks.

[33] Y. Yiakoumis, S. Katti, T.-Y. Huang, N. McKeown, K.-K. Yap, and R. Johari. Putting home users in charge of their network. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 1114–1119, New York, NY, USA, 2012. ACM.

[34] X. Yin, V. Sekar, and B. Sinopoli. Toward a principled framework to design dynamic adaptive streaming algorithms over http. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, pages 9:1–9:7, New York, NY, USA, 2014. ACM.