Automated Home Network Troubleshooting with Device Collaboration

Stéphane Wustner
UPMC / Technicolor
stephane.wustner@lip6.fr

Diana Joumblatt
UPMC
diana.joumblatt@lip6.fr

Renata Teixeira
UPMC
renata.teixeira@lip6.fr

Jaideep Chandrashekar
Technicolor
jaideep.chandrashekar@technicolor.com

Categories and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]: Reliability, availability, and serviceability

Keywords

Troubleshooting, Home Network

1. INTRODUCTION

With the increasing proliferation of networked devices inside the home, we see a corresponding increase in home networking complexity, and, this directly contributes to frequent network issues: poor video or skype quality due to contention, misconfigurations, wireless networking issues, and so on. At the same time, the tools available today – ping, traceroute, iperf, etc. – are antiquated and unfriendly. Thus, networking issues are a source of great frustration for end-users (most of who do not understand the underlying technologies and their operation). Even with the required technical acumen, it is quite challenging to conclusively pin down the root cause of a networking problem in the home. Who is to blame when the video player is stalling? the DNS server? the browser plugin? the wifi configuration? the access point? or the access network? Pinpointing the specific cause conclusively is challenging today because there is no single location where all the relevant information is available and can be analyzed; instead networking state is fragmented across many devices.

There has been a considerable body of work addressing the general area of network troubleshooting and these can be classified into two broad categories. First, approaches that reason about performance and correctness in terms of dependencies between various services (e.g., the DNS service has to operate correctly for a web page to be loaded) and identifying missing components [3, 7, 6]. For instance, Sherlock [3] builds the dependencies based on topological information and connection times. The second, we have a number of approaches that rely on crowdsourcing to build databases of performance snapshots, working configurations, or problem specific remediation techniques [2, 8, 1] and guiding users that face specific problem to the subset of collected knowledge that directly relates to the problem at hand.

In spite of all these previous efforts, networking issues in the home still plague end-users today. Let us consider a very simple example: an end-user watching video on their tablet but the stream periodically stalls. It is unlikely that the underlying problem is due to a misconfiguration. Moreover, the dependency structure in the services involved is quite simple (a dns request precedes a http request for a particular chunk of video). Thus, in this particular example, the approaches enumerated previously are not easily adoptable. At the same time, it is not difficult to reason about the particular example: there is a certain bandwidth envelope required for smooth video delivery, and it appears that not enough bandwidth is available. Possibly, this is due to another competing stream, poor wireless conditions, or a drop in the upstream bandwidth. Once the particular problem cause and the location are identified, various mitigation schemes can be applied.

The solution sketch just outlined seems trivial, but not easy to implement and deploy in today's home networks simply because the different bits of information required to reason about this problem is partitioned across a number of devices (the client, the gateway, competing clients, etc.). We believe that it is possible to realize an effective system to diagnose and troubleshoot home networking problems by enabling end-devices to share information with each other and building algorithms that can reason with this shared state in order to identify performance and configuration issues. In this thesis, we seek to answer the following questions: (i) What are the common networking issues experienced by end-users at home? and (ii) Can we build a system that can proactively diagnose the state of the network and perform the required troubleshooting without explicitly involving the end-user (as much as possible)?

2. APPROACH

Our approach will be to build a framework that allows devices inside the home to exchange network state information with each other so that any particular entity can construct a view of the network and reason about it. Before we do so, we need to get an understanding about the particular elements of network state that need to be recorded by each device,

CoNEXT Student'12, December 10, 2012, Nice, France.
Copyright 2012 ACM 978-1-4503-1779-5/12/12 ...$15.00.

and how to correlate these. In order to inform these choices, we direct our efforts at getting a high level understanding of problems that end-users experience in the home.

The work that will be done in this thesis consists of two broad tasks, and these are described below:

Understand common home-network problems: Our initial efforts have focused on analyzing traces collected by the Hostview project [5], which provides a rich dataset that contains network traces as well as user feedback about network perception. This particular dataset exposes the network as seen from a single end-host (where the traces are collected). We plan to collect other datasets that can reveal behavior from other networked devices in the home. From these datasets, we will develop methods to identify performance problems and misconfigurations and possible causes that will help us in the next component of the thesis.

Build a collaborative, troubleshooting framework: The framework is enabled by two components. First, we will study and leverage existing technologies (e.g., publish-subscribe systems) to implement an API for networked devices to publish networking state information periodically. The particular networking state will include metrics that capture *configuration*, *reachability* to a set of landmark locations, and *performance characteristics* of ongoing network sessions. This information can be gathered by other entities in the network (including one that will live on the home DSL/Cable gateway). This presumes an operational network connection which may not always be the case. To deal with episodes of intermittent or no connectivity, we will use other available network interfaces (bluetooth, etc). Second, we will study existing approaches and algorithms for tracking performance and detecting anomalies on this shared knowledge plane; developing our own approaches when required.

3. PRELIMINARY RESULTS

Here, we briefly present some very early results toward identifying common home network issues. The analysis we describe is based on the data obtained from the Hostview project [5]. The data collected at the end-host, includes traffic traces, application context, as well as machine level metrics (cpu load, wireless signal statistics) and, importantly, feedback from end-users about their perception of the network performance (see [5] for details). With the traffic traces, we extracted a number of network metrics (rtt, jitter, throughput, retransmissions, etc.)

From the subset of data that was collected inside homes, we tried to correlate the different recorded metrics with episodes when users *complained about network performance* and a very high level summarization is presented in Fig. 1. Here, each vertical element represents a single instance when the end-user reported poor or unsatisfactory performance (the figure uses data from 85 samples) while each horizontal "strip" corresponds to a specific metric that was deemed anomalous. We used a simple anomaly detector that reported an anomaly if the metric exceeded the 95%-ile (< 5%-ile for SNR). Throughput (denoted as {05,95}, down{05,95} in the figure) is treated slightly differently: an anomaly occurs if the throughput (up and down) is too high (`up95` corresponds to when the value is greater than the 95%-ile) or if it is too low (`up05`, when the value is below the 5%-ile).

The presence of a dark element indicates an anomaly (for that metric). While no obvious pattern jumps out, we do see that the rtt metric is anomalous (i.e., the round trip times are relatively high and fall into the tail of the rtt distribution) for a very large number of samples. This leads us to believe that rtt is an important metric.

We further analyzed the rtt's reported in these incidents and uncovered an interesting observation. In about half of the incidents when users reported poor performance *and* the rtt was anomalous, a time series of rtt values, plotted for the window of time covering the user report (2 minutes before and after), exhibits a very pronounced sawtooth pattern that is characteristic of buffer bloat [4]. This gives us some intuition about how we can build logic to identify such patterns and build mechanisms to mitigate it. This seems to be a situation our troubleshooting system and technique might be able to address, thanks to communication between the gateway and the device, by having, for instance, the gateway telling the device its buffer is filling up, and additionally, by having the device telling the gateway it has high rtt with these flows, and it need to buffer them less.

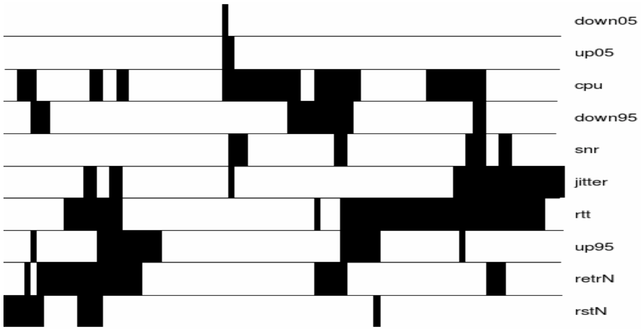

Figure 1: **Correlation bad performance/metrics.**

4. ACKNOWLEDGEMENT

This work was supported by the EC's 7th system Program (FP7/2007-2013) no. 258378 (FIGARO).

5. REFERENCES

[1] B. Aggarwal, R. Bhagwan, L. D. Carli, V. Padmanabhan, and K. Puttaswamy. Deja vu: fingerprinting network problems. In *CoNext*, 2011.

[2] B. Aggarwal, R. Bhagwan, T. Das, S. Eswaran, V. N. Padmanabhan, and G. M. Voelker. Netprints: diagnosing home network misconfigurations using shared knowledge. In *NSDI*, 2009.

[3] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *SIGCOMM '07*, 2007.

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

[5] D. Joumblatt, R. Teixeira, J. Chandrashekar, and N. Taft. Hostview: annotating end-host performance measurements with user feedback. *SIGMETRICS Perform. Eval. Rev.*, 2011.

[6] S. Kandula, R. Chandra, and D. Katabi. What's going on? learning communication rules in edge networks. In *SIGCOMM '08*, 2008.

[7] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl. Detailed diagnosis in enterprise networks. In *SIGCOMM*, 2009.

[8] N. Kushman, M. Brodsky, S. Branavan, D. Katabi, R. Barzilay, and M. Rinard. Wikido. In *HotNets*, 2009.