Locating Throughput Bottlenecks in Home Networks

Srikanth Sundaresan, Nick Feamster, Renata Teixeira

Locating Throughput Bottlenecks in Home Networks

Srikanth Sundaresan[†], Nick Feamster[†], Renata Teixeira[‡]
[†]Georgia Tech [‡]INRIA

Abstract

We present a demonstration of *WTF (Where's The Fault?)*, a system that localizes performance problems in home and access networks. We implement WTF as custom firmware that runs in an off-the-shelf home router. WTF uses timing and buffering information from passively monitored traffic at home routers to detect both access link and wireless network bottlenecks.

Categories and Subject Descriptors: C.2.3 [Computer-Communication Networks] *Network Operations*: Network Management; C.2.3 [Computer-Communication Networks] *Network Operations*: Network Monitoring

General Terms: Measurement, Performance

Keywords: bottleneck location; home networks; performance diagnosis; troubleshooting

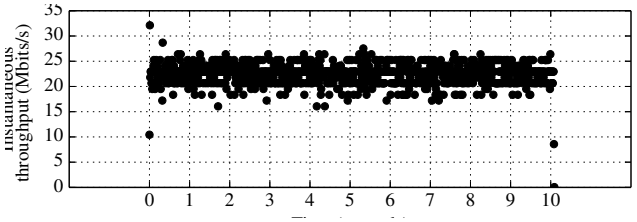
1 Introduction

We develop an algorithm and tool that determines whether network performance bottlenecks lie inside or outside the home network. *WTF (Where's The Fault?)* runs on commodity home routers and detects access link and wireless bottlenecks in a home network. WTF uses timing and buffering information from passively monitored traffic to design two maximum likelihood detectors: one detects access link bottlenecks and the other detects wireless network bottlenecks. Together, these detectors allow us to infer properties of the network and the most likely location of performance problems. These detectors can be easily measured from resource-constrained home routers. Although WTF does not determine *why* a particular bottleneck or problem exists, it takes an important first step in helping users and ISPs determine *where* the problem exists, at least to the granularity of whether the problem is inside or outside the home.

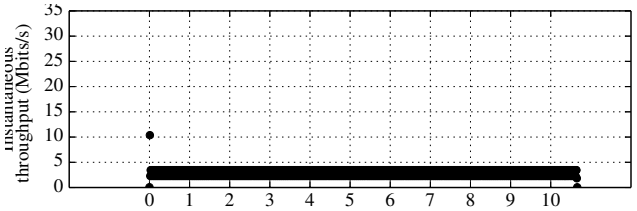
To deploy WTF in as many homes as possible, we implemented it as custom firmware that runs on a commodity home router. Although this approach allows us to collect measurements on a low-cost device that users are familiar with (and hence, more than willing and able to install), it introduces a unique set of challenges because the device is resource constrained. This environment makes it difficult to apply existing bottleneck detection and wireless analysis tools [1, 2], since they typically require additional affordances (*e.g.*, multiple wireless vantage points, significant trace collection). WTF works within these constraints and only collects lightweight passive measurements and conducts lightweight data processing on the home router.

SIGCOMM'14, August 17–22, 2014, Chicago, IL, USA.

http://dx.doi.org/10.1145/2619239.2631440



(a) Access link is not the bottleneck. *Instantaneous throughput at the WAN interface varies at short time scales due to high variance in packet inter-arrival times.*



(b) Access link is the bottleneck. *Instantaneous throughput at the WAN interface is steady, due relatively uniform packet interarrival times caused by upstream shaping.*

Figure 1: *Behavior of packet inter-arrival times.*

2 Detection Algorithm

We exploit a fundamental property of bottleneck links: *packets buffer at the head of the bottleneck queue*. This property manifests itself in two ways from the point of view of the access point depending on the location of the bottleneck.

Access link bottleneck.*We use the intuition that bottleneck links smooth packet arrival rates.* Because a bottleneck link services packets at a rate slower than they arrive, queues build up at the link, and the link paces packets at an even rate. Packets upstream of the bottleneck will arrive according to the natural variation induced by TCP congestion control, but packets are more evenly spaced downstream of the bottleneck link. We assume that the most likely bottleneck upstream of the home network is the access link, so *all* flows are buffered, which allows us to use the overall packet distribution for detection.

We expect to see high variance in packet interarrival times before the bottleneck link due to congestion control, but significantly lower variance after the bottleneck link itself because the buffer smoothes packet arrivals. Figure 1 shows this effect: It shows the instantaneous TCP throughput at a granularity of 10 ms, as measured from the gateway. In Figure 1a, the access link throughput is 100 Mbits/s; the wireless link is the bottleneck because the maximum TCP throughput it can support is about 21 Mbits/s. In Figure 1b, we shape the access link to 3 Mbits/s, significantly lower than the wireless capacity. In this case, throughput is less variable. Indeed, the coefficient of variation for packet interarrival times, c_v, when the

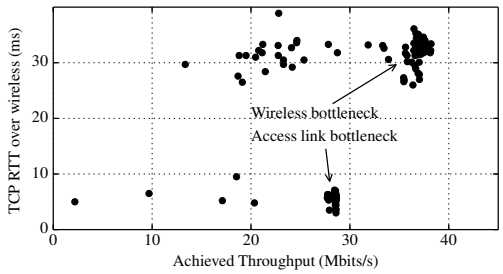


Figure 2: *TCP RTT between client and gateway. RTT is significantly higher when the wireless link is the bottleneck; this is caused by buffering.*

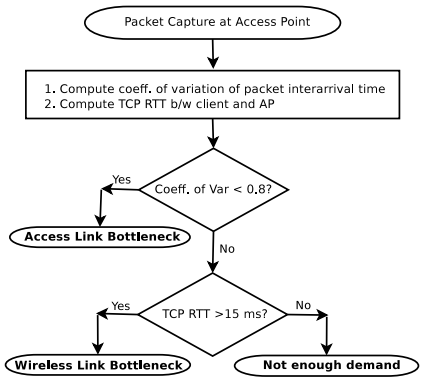


Figure 3: *Combining the two bottleneck detectors to create a single combined detection algorithm for access link bottlenecks (event B) and wireless bottlenecks (event W).*

access link is the bottleneck for this example is 0.05; in contrast, when it is *not* the bottleneck, c_v is 0.88.

Wireless bottleneck. *We use the intuition that TCP round-trip time from the wireless router to the client is high if the wireless link is the bottleneck.* Queues build up at the head of a bottleneck link. Because we cannot view the wireless buffer directly without instrumenting the driver, we look at the impact of buffering on TCP flows. We run `tcptrace` on the traces we collect on the router to obtain the RTT of TCP flows, τ, between the router and the clients in the local network. If the wireless link is not the bottleneck, the RTT is expected to be low, as the packet will dispatched without delay. Even though the wireless link is not work-conserving, the delays caused by medium access control are low compared to buffering delays.

Figure 2 illustrates this effect with an example. We run two tests in a setting where the wireless link capacity is about 40 Mbits/s (obtained by repeated measurements). In the first case, the access link is throttled to 30 Mbits/s, so it is always the bottleneck. In the second case, the access link is throttled to 70 Mbits/s so that the wireless link becomes the bottleneck. We see that there is a significant disparity in the TCP RTT in these two cases; when the wireless link is the bottleneck, the RTT is about 25–30 ms, while when the access link is the bottleneck, the RTT is about 5 ms. This effect does not depend on the achieved throughput; it depends solely on the occurrence of buffering.

Putting it together. We combine the access link and the wireless link bottleneck detectors using a simple algorithm. Figure 3 shows the algorithm. Both the detectors are simple threshold based; therefore there are four scenarios. When either the access link threshold or the wireless threshold is breached, we deem the corresponding link to be the bottleneck. When neither thresholds are breached, we

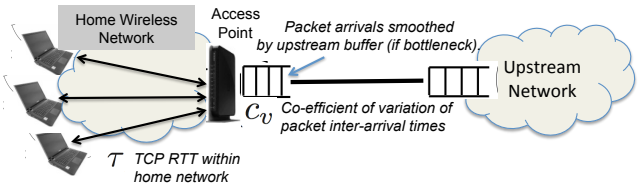


Figure 4: *WTF runs on the gateway between the home network and the access link, thus offering a unique vantage point for observing pathologies on either side.*

deem the bottleneck to be elsewhere, or the application demand to be insufficient. In our experiments, we model this by introducing latency or loss in the path so that TCP throughput is less than the access link and the wireless throughput—neither are the bottlenecks. The case when both the thresholds are breached is likely to happen only when the access link and the wireless link throughputs are closely matched, as there can only be one throughput bottleneck in the end-to-end path. We test all the above cases rigorously with controlled experiments. We obtained robust thresholds for both c_v and τ. We saw that testing for c_v with a threshold of 0.8 and τ with a threshold of 15 ms produced more than a 95% true positive detection rate and less than a 5% false negative rate for all the cases (the algorithm produces similar results over a broad range of thresholds; we use the above values).

3 Demonstration Details

In our demo, we will show WTF working on a commodity home router. We will use an experiment setup as shown in Figure 4. We will use traffic controllers to vary the bandwidth, latency, and loss in the end-to-end path with wireless clients, and show how WTF accurately detects cases where there are throughput bottlenecks — both access link and wireless bottlenecks — and also cases where there are no throughput bottlenecks. The tool front-end will be a simple web page hosted on the router that is constantly refreshed with the current state of the access link and the wireless clients. We will also provide detailed information about the wireless network performance that each client gets (bitrates, retransmission rates, *etc.*); this will allow us to better understand client performance. While we can conduct the demo without an upstream Internet connection, the demo will be enhanced with an Internet connection, especially as it will be easier to get more people to associate with our router wireless, and therefore emulate realistic scenarios.

Acknowledgments

We thank Dina Papagiannaki and Yan Grunenberger for early discussions that led to development of Where's the Fault. This project is supported by NSF Award CNS-1059350, a Google Focused Research Award, and the European Community's Seventh Framework Programme (FP7/2007-2013) no. 258378 (FIGARO).

References

[1] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the mac-level behavior of wireless networks in the wild. In *In Proc. ACM SIGCOMM*, pages 75–86, 2006.

[2] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the characteristics and origins of internet flow rates. In *Proc. ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002.