

## Application-Aware Bottleneck Detection Tool

Dr. Renata Teixeira

Inria Research Center, Paris - France

### Motivation

Despite the steady increase in home broadband speeds, home users can still experience low Quality of Experience (QoE) for their daily applications. Without the knowledge of where the problems might be located, users often contact their ISP complaining about the quality of their Internet access, but this is only one of the potential bottlenecks of Internet application performance. As such, the ability to monitor and analyze how QoE impairments correlate to the root cause generating them becomes key. The complexity of Internet applications makes it hard to locate the source of application performance bottlenecks. No single entity, be it the user, network provider, or service provider, has complete visibility on the end-to-end path. Therefore, identification of application performance bottlenecks requires inference from end-to-end probing. Further, different applications react differently to the same network conditions, hence, network performance monitoring must consider application-specific metrics. Unfortunately, existing network performance monitoring tools (e.g., ping, speedtest, traceroute) are application agnostic.

The novelty of our proposal is to develop end-to-end probing methods for per-application bottleneck detection. Although existing tools (e.g., SamKnows whiteboxes) do perform tests of application quality, these active tests fail to capture application quality impairments users experience. In addition, developing tests for the vast variety of possible Internet applications is prohibitive. The passive observation of users' traffic is key to infer real user experience and detect quality degradation. Application traffic, however, only captures end-to-end behavior, and hence is not sufficient to pinpoint bottlenecks. Active probing is therefore necessary to infer the performance of individual network segments and locate the origin of bottlenecks.

### Challenges of Application Bottleneck Detection

Two main active probing methods exist: (i) issuing TTL-limited probes (à la traceroute) to breakdown the end-to-end path and measure properties of the IP links composing the path, or (ii) applying network tomography to correlate end-to-end measurements collected from multiple vantage points. While simple in design, there exists multiple challenges that hinder the applicability of active probing for application quality bottleneck detection. Active measurements are subject to measurement artifacts and noise. Take a simple example of probing a user device when the NIC is on power-save mode. Such probes can experience delays of hundreds of milliseconds, as opposed to a few milliseconds when the NIC is fully active. Another challenge is that active probing can yield significant network overhead since it generates additional traffic into the network. Finally, probes may follow a different path than the application flow due to the existence of middleboxes in the path (e.g., load balancers, firewalls) that make forwarding decisions based on flow characteristics. A reliable and practical application bottleneck detection tool should be: 1) **accurate** with minimal false positives, 2) **lightweight** on the network, 3) **application-aware**, correlating network performance with application QoE degradation and traversing the same path as application flows, and 4) **timely**.

### Project Goal: Per-Application lightweight Bottleneck Detection

The goal of our research is to build an open-source software tool that performs lightweight

on-demand active probing for application performance bottleneck detection. Our research is part of a collaboration with Princeton University in the context of the NetMicroscope project. NetMicroscope aims to perform passive network and application performance monitoring. Our tool can work independently where a system administrator triggers monitoring of a given user application on-demand; or fit as a module in NetMicroscope, which can trigger probes once it detects application performance degradation.

We design our tool to run in a box acting as a home gateway. The home gateway is the ideal vantage point as it provides visibility of home users' application traffic and allows to separate between local and wide-area or service-related impairments. Our tool will locate bottlenecks at the level of network segments: i.e., home, access, transient networks, interconnects, and content provider. The main building blocks of our software are a WAN monitor that probes selected interfaces along the path from the home gateway to the requested service, a wireless monitor that probes the devices inside the home network, and a bottleneck detector that locates application performance bottleneck.

## Deliverables

- **WAN Monitor.** WAN monitoring must be able to monitor the quality of the paths of application flows. It entails identifying all the flows that compose an application session; for each of these flows, breaking down the end-to-end path into network segments, and finally building probes with the same characteristics as the application flows. The resulting measurements coupled with passive inference of end-to-end performance will reveal properties of the different network segments and feed into our detection method to locate the application performance bottleneck. We will leverage an existing tool, Service Traceroute, developed by our group at Inria. Service Traceroute issues probes that copy the fields of application flow packets. This method ensures that the probes will follow the same Internet path as the application traffic. One challenge for our monitor is that many applications are composed of short-lived flows, and we need to ensure that we measure all network segments within the lifetime of the flow. Otherwise middleboxes along the path will often drop our probe packets. Moreover, identifying which application flows to trace is not straightforward due to the heterogeneity of applications and service architecture. For instance, modern Internet services, such as Youtube and Netflix, involve multiple flows to different servers; i.e., retrieve the manifest file from one node and download the application from a different node. Finally, we will rely on bdrmap<sup>1</sup>, an existing tool, to infer interconnects between the access network and the next-hop networks. Subsequently, we envision probing the two ends of an interdomain link to infer the properties of interconnect links. Similarly, probing the entry points and exit points of networks in the path will provide information about the performance of a specific network.
- **Home Monitor.** Monitoring in home networks is complex as most modern households rely on wireless, which can be volatile. Further, in home monitoring, the measurement end-points are home user devices, hence, active probing may interfere with user experience. For example, probing too often a device will drain its battery. On the other hand, passive measurements of wifi quality (e.g., RSSI, PHY rate) require monitoring directly on the AP, which may not be possible, or deployment of wifi sniffers, which comes at a higher cost. In the context of this proposal, we will evaluate the cost-benefit of active and passive wireless monitoring. Our goal is to understand how well these different monitoring approaches detect when wireless is

---

<sup>1</sup>Luckie, Matthew, et al. "bdrmap: inference of borders between IP networks." Proceedings of the 2016 Internet Measurement Conference. ACM, 2016.

the application performance bottleneck. This study will guide our design of a cost-effective and lightweight wireless monitoring system.

- **Bottleneck Detector.** A bottleneck detector must correlate between the measured properties of individual network segments and the passive inference of end-to-end performance to accurately locate application performance bottlenecks. In the context of this proposal, we will explore different data-driven approaches to build such a detector: from time-series analysis to more sophisticated machine learning approaches. We will feed these methods with training data collected from real home deployments and in-lab experiments where we can exercise different network conditions. We will validate the accuracy of our bottleneck detection tool in testbeds (e.g., Planetlab<sup>2</sup>, and Orbit<sup>3</sup>), as well as in real home deployments. In our work with Princeton, we currently have 60 deployments in homes across the US and Paris. We also plan to validate the accuracy of our inference via operator’s feedback. We hope Comcast can assist us in the validation step. We also hope that this grant could be an occasion to deploy such tools with Comcast home routers.

This system will be useful for users and network operators to diagnose application performance, and hopefully improve overall user experience. It will also help educate individual users on the bottlenecks affecting their Internet experience. Beyond that, a large-scale deployment of such a system will shed light on the interaction of network performance and application quality, and help us answer a number of open research questions: what are the main source of application performance bottleneck? how does these sources change between different classes of applications, access types?

## Budget Outline

The project will gather a group of existing researchers from the MiMove team at Inria Paris (two postdoctoral fellows: Sara Ayoubi and Francesco Bronzino and one Ph.D. student: Israel Marquez), under the supervision of Dr. Renata Teixeira, and in collaboration with Princeton University under the supervision of Prof. Nick Feamster. This proposal will help fund a one year salary for a post-doctoral fellow (Sara Ayoubi), and travels for researchers working on the project (in the context of the project) to different conferences, and/or visits to Princeton University and Comcast. In addition, a portion of the budget is allocated to equipment purchase to increase the number of boxes deployed in homes across Paris.

Budget Item	Cost
One year postdoctoral fellow salary	52,000 Euros (60,437 USD)
Travel (Conference, workshops, visits to Princeton and/or Comcast)	10,000 Euros (11,623 USD)
Equipment (Odroid, adaptors, routers)	5,000 Euros (5,812 USD)
Total	67,000 Euros (77,872 USD)

<sup>2</sup>Chun, Brent, et al. "Planetlab: an overlay testbed for broad-coverage services." ACM SIGCOMM Computer Communication Review 33.3 (2003): 3-12.

<sup>3</sup>Ott, Maximilian, et al. "Orbit testbed software architecture: Supporting experiments as a service." Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on. IEEE, 2005.