ConnectionWatch: Passive monitoring of round-trip times at end-hosts

Diana Zeaiter Joumblatt, Renata Teixeira

ConnectionWatch: Passive monitoring of round-trip times at end-hosts

Diana Joumblatt, and Renata Teixeira
Université Pierre et Marie Curie – CNRS, Laboratoire LIP6

1. INTRODUCTION

The current Internet offers a diverse set of applications for end-users (*e.g.*, online gaming, IPTV, VoIP, VoD). These applications require a timely delivery of packets to avoid the deterioration of user-perceived performance. Slow web-browsing, momentarily drops of voice in Internet telephony and lagging remote connections are some of the most common problems that affect the usability of Internet services. Ideally, if applications had access to measurements that quantify network performance, they could dynamically adapt to network conditions and improve user satisfaction. Most importantly, when application service quality falls below expectations, online measurements help users understand who to blame and what to do to bypass the problem.

We propose a troubleshooting tool that runs at end-hosts to automatically detect and diagnose performance degradations. The tool has three aspects: detection of performance degradation, identification of the location of the problem and characterization of the potential causes. In this work, we focus on the detection aspect, which requires continuous monitoring at end-hosts.

There are two types of monitoring: active and passive. Active probes like ping and traceroute suffer from four major drawbacks when they are used for the purpose of continuous monitoring. First, they incur a high probing overhead. Second, they are often subject to rate-limiting or complete blocking at various sites or end-hosts. Third, some firewalls and load balancers answer probe requests for the hosts they represent; and as a consequence, it is impossible to get accurate end-to-end measurements. Finally, active probes may not reflect the application's experience of network events. On the other hand, passive measurements are promising.

This work presents our efforts to build ConnectionWatch, a passive measurement tool that reports intolerable delays as perceived by users. In particular, we start by developing a basic tool that monitors TCP connections at end-users; and logs them for further processing and analysis. The experiments conducted with this preliminary tool will help in the design of ConnectionWatch. Two factors motivate our choice of TCP. Primarily, TCP accounts for 60% to 90% of the Internet traffic load [1]. In addition, tracking TCP connections offers the advantage of deriving path RTT estimates according to the applications' sending rate. Given that ConnectionWatch only observes traffic, it avoids measurement overload on the network induced by frequent active probes. This passive approach also has the advantage of measuring only active connections. In particular, it watches the paths that applications are effectively using in contrast to probing all the paths.

ACM CoNEXT '08 Student Workshop, Dec. 9, 2008, Madrid, Spain.

2. MEASUREMENT SETUP

ConnectionWatch should generate alarms to signal performance degradations. However, we need traces collected at end-hosts to experiment with different delay detection techniques. This section describes the building blocks of the basic tool that we use to collect insightful data for our research.

2.1 Basic tool

The tool performs the monitoring task by accessing TCP and IP header information as well as the packet timestamp recorded by the capture device. It is composed of three major modules:

1. The **packet sniffer** uses a libpcap-based library to intercept TCP packets as they arrive to the host's network interface. For evaluation purposes, we also configure this module to dump the entire packets to a trace file, which is uploaded to a central server.

2. The **flow analyzer** groups TCP packets into flows identified by the 4-tuple: source/destination IP address and source/destination port number. It maintains a list of active TCP connections for which it derives RTT estimates in real-time. Depending on the source's sending rate, the module computes many RTT samples during the lifetime of a flow based on TCP's algorithm for RTT estimation. In addition, we extend the tool to include a ping daemon that issues a ping probe to the destination of an active TCP connection for every data packet sent by the source. Our goal is to compare the RTT samples computed by our tool with those reported by ping.

3. Upon user request, the **TCP logger** module generates a set of log files that provide a summary of TCP connections during the monitoring period: start and end time, number of RTT samples and their values, IP addresses and port numbers of both ends of each connection.

2.2 Initial experiment

We use this version of the tool in an initial deployment to explore the monitor's capabilities and limitations, fix potential bugs, and get insights into the type of data that passive measurements provide. Five graduate students installed the tool on their machines and ran it for a three-day period while they were using their Internet connection at home and at work in LIP6. We collect a 3.15 GB volume of payload traces.

3. INSIGHTS AND DISCUSSION

This section discusses some of the practical insights gained from the experiment.

Passive measurements are encouraging: The traces cover 3,584 paths, for which we get RTT samples from passive measurements. We see 2,242 unique destination IPs spread in 412 autonomous systems[1] and process 44,715 TCP connections. We observe that despite the small volume of data collected, we are able to passively measure paths to a large number of autonomous systems and many popular services like Google, Ebay and YouTube.

Ping might not work: We find that 16,5% of 1,072 destinations probed don't reply to pings. Note that pings are sent to the destination IP address of an active TCP connection. Indeed, the result confirms the well-know observations that active measurement tools are dropped by some networks or remote destinations and that end-users can only partially rely on them to debug performance problems. We believe that ConnectionWatch is particularly useful for monitoring paths, where the destinations block active probes.

The detection of large RTTs is challenging: Figure 1 plots the cumulative distribution function of the RTT samples. Overall, there is a large disparity in the distribution. We observe that 26% of RTTs are less than 10 ms and 0.03% have values higher than 1 second. The initial results show that RTTs vary across users and paths. They also highlight two important issues: (i) a path with 2 ms RTT that suddenly goes to 10 ms might be experiencing problems while the same change is insignificant for a path with 200 ms RTT, (ii) even if we develop a technique to detect when RTTs are abnormally high, it is challenging to determine whether ConnectionWatch should trigger an alarm for each of these anomalous RTTs. Ideally, alarms should match the user's perception of network performance, which may depend on the application and the user's tolerance. A general detection scheme for faulty flows requires prior knowledge about applications and also feedback from users.

[1]We use Team Cymru's IP-to-AS mapping [4].

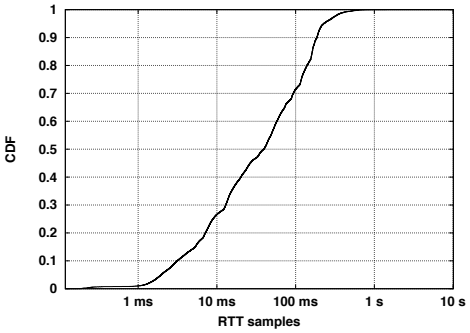

Figure 1: RTT distribution

There are important issues that need to be addressed before the large-scale deployment of the tool: Our work calls for large-scale measurements to study performance degradations, which are exceptions and not the rule. It also requires capturing a lot of personal information at the end-hosts: favorite applications, frequently visited web-sites and IPs that are regularly contacted by the host machine. Anonymization procedures should be implemented to mitigate privacy concerns.

4. FUTURE WORK

Our main objective is to study the impact of network disruptions on application performance as perceived by end-users. This learning phase is crucial for the design of a detection technique that automatically alerts users or applications to performance degradations. We intend to include an 'I am annoyed' button to ConnectionWatch, that users can click on when they are dissatisfied with the quality of their Internet application. This kind of data about user perception will hopefully shed the light on the interaction between the network layer and the application's perceived quality of service. Finally, ConnectionWatch will be extended to passively track other metrics such as packet loss and bandwidth usage.

Acknowledgements

We thank Nina Taft for her helpful feedback and suggestions.This work was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) no. 223850.

REFERENCES

[1] M. Fomenkov, K. Keys, D. Moore, and K. Claffy, "Longitudinal study of internet traffic in 1998-2003," in *ACM International Conference*, 2004.

[2] R. G. Cole and J. H. Rosenbluth, "Voice over ip performance monitoring," in *ACM SIGCOMM Computer Communication Review, v.31 n.2*, April 2001.

[3] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying skype user satisfaction," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, September 11-15, 2006, Pisa, Italy.

[4] "Team cymru." `http://www.team-cymru.org/Monitoring/`.