Multi-Hop Packet Tracking for Experimental Facilities

Tacio Santos
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
tacio.santos@
fokus.fraunhofer.de

Christian Henke
Technical University Berlin
Str. des 17. Juni 135
10623 Berlin, Germany
c.henke@tu-berlin.de

Carsten Schmoll
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
carsten.schmoll@
fokus.fraunhofer.de

Tanja Zseby
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
tanja.zseby@
fokus.fraunhofer.de

ABSTRACT

The Internet has become a complex system with increasing numbers of end-systems, applications, protocols and types of networks. Although we have a good understanding of how data is transferred over the network we cannot observe what happens with our data after sending and before receiving it - how packets traverse through the network and with which QoS characteristics remains unknown. Towards this objective we have developed a multi-hop packet tracking system intended to be used in experimental facilities, such as PlanetLab, where we have made our first tests. This paper describes our packet tracking realization and the results from our prototype implementation.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Operations—*Network monitoring*

General Terms

Experimentation, Measurement

Keywords

multipoint measurement, hash-based packet selection, IPFIX

1. INTRODUCTION

Making observations is essential for experimental research. Measurements support scientist in experiment supervision and capturing of environment conditions. It also gains importance for network operation, where QoS demanding applications and adaptive algorithms require feedback about the quality experienced by packets on the path. To serve such needs, we developed a multi-hop packet tracking architecture that passively monitors the paths that packets take throughout the network and also records detailed hop-by-hop metrics like delay and loss. These measurements can be used for example for traffic engineering, for the validation of routing algorithms or traffic distribution protocols (multicast). Furthermore, packet tracking enables measurements of environment conditions like cross-traffic and its influence on the user or experimenter traffic. Tracking single packets through the network supports trace-back systems [5] by deriving the source of malicious traffic and revealing the location of the adversary. Resource limitations usually prevent us from tracking all packets in a network or flow. Therefore we use a hash-based packet selection technique that ensures a consistent selection throughout the network while maintaining statistically desired features of the sample.

SIGCOMM'10, August 30–September 3, 2010, New Delhi, India.
ACM 978-1-4503-0201-2/10/08.

2. ARCHITECTURE

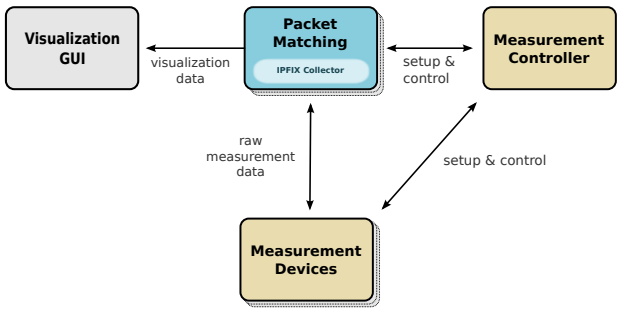

Figure 1: Packet Tracking Overview

Our packet tracking architecture consists of 1) multiple passive measurement probes deployed in the network, 2) a packet matcher with an IPFIX collector that correlates the probe's measurements, 3) a measurement controller to coordinate setup and control of measurement parameters and finally 4) a visualization tool to facilitate analysis of processed data. We use the IPFIX protocol [1] to transfer packet tracking data from the probes to the matcher. IPFIX was primarily developed to export flow information, but also allows the reporting of per-packet information. It uses a template-based approach that assists in the definition of new information elements and also defines a standardized file format for storing measurement data (RFC5644). The probes export at least a packet ID and either the TTL or an

arrival timestamp for each observed packet to the collector. Based on the packet ID the packet matcher can correlate the observations and determine the packet's direction by the TTL or timestamp. The exported timestamps can also be used for calculating one-way delay between the observation points, which further requires that the measurement nodes' clocks are synchronized.

While designing the packet tracking architecture we have concentrated on

1. Efficient export of measurement results. We use the IPFIX protocol and standardized Information Elements.
2. Choice of suitable packet ID generation functions. We evaluated different functions for observation correlation in [4] and decided to use the BOB hash function.
3. Reduction of measurement traffic. We use hash-based packet selection, a deterministic filter based on a hash over the packet content that synchronizes the selection of packets at different observation points. Our evaluations show that hash-based packet selection can emulate random sampling [3] when using the BOB hash function.
4. Synchronization of the sampling fractions in the network. In the case of bandwidth depletion in the network the sampling fraction of packets should be adjusted in order to reduce the measurement traffic. To support this process we export node information like bandwidth, CPU and RAM usage via IPFIX.
5. Visualization of measurement results. We use a Java visualization, which makes use of OpenStreetMap in order to visualize packet paths, their hop-to-hop characteristics and information about the nodes.

3. DEMONSTRATION

We made first tests of our packet tracking architecture in PlanetLab [6] and also plan to deploy the architecture in G-Lab [2], two experimental facilities with world-wide / German-wide distributed nodes but different policies for experimenters. In order to make use of the packet tracking architecture we create a routing overlay so that the hosts also work as intermediate routers and we can track packets over multiple hops. We visualize the packet path in a Java application using OpenStreetMap and a Java animation framework. An aggregate number of packets taking the same path will be visualized as a moving light dot. For the packets' paths we used cubic splines in order to differentiate between packets that travel the same links but have different ingress and egress nodes. We use different layers to visualize node properties (CPU, RAM, sampling fraction) and link characteristics (delay). Popups over the nodes and links show a graphical representation of the data in a time window. A captured video of an earlier demonstration has been recorded and made available at `http://bit.ly/packet-tracking` where you will also find further information about the packet tracking architecture.

4. SOFTWARE COMPONENTS

Our packet tracking architecture is built upon the following open source software components developed by Fraunhofer FOKUS. All software components are available at `http://bit.ly/packet-tracking`.

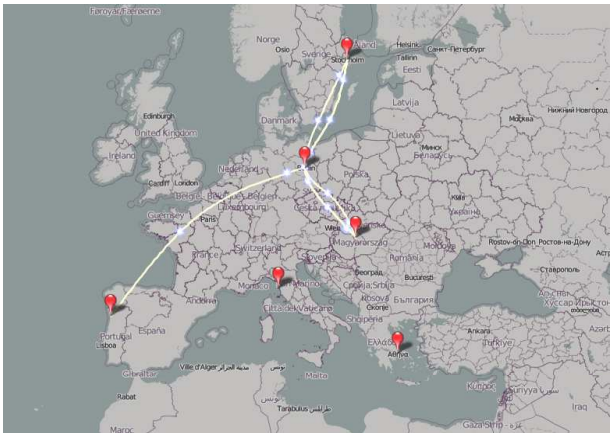

Figure 2: Screenshot of the Demonstrator

1. impd4e - a small open source measurement probe intended for embedded systems
2. OpenIMP - an open source measurement platform including probes and measurement controllers
3. libIPFIX - an open source C-library for exporting measurement results via the IPFIX standard
4. Packet Tracking Visualization - a Java visualization tool based on OpenStreetMap and several other open source projects.

5. ACKNOWLEDGMENT

This work has been partially funded by the European IST OneLab2 project under grant agreement 224263. We further would like to thank the developers of the following Open Source software: Freimap, OpenStreetMap, and Avaje Ebean ORM Persistence Layer.

6. REFERENCES

[1] B. Claise. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101 (Proposed Standard), January 2008.

[2] G-Lab. German Lab Experimental Facility. http://www.germanlab.de/.

[3] C. Henke, C. Schmoll, and T. Zseby. Empirical evaluation of hash functions for multipoint measurements. *SIGCOMM Comput. Commun. Rev.*, 38(3):39–50, 2008.

[4] Christian Henke, Carsten Schmoll, and Tanja Zseby. Empirical evaluation of hash functions for packetid generation in sampled multipoint measurements. In *PAM*, pages 197–206, 2009.

[5] Alex C. Snoeren. Hash-based ip traceback. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–14, New York, NY, USA, 2001. ACM.

[6] Tanja Zseby, Michael Kleis, and Christian Henke. Packet tracking in planetlab europe with a use case. In *Proceedings of TRIDENTCOM '10*, May 2010.