Network Troubleshooting: An In-band Approach

Murtaza Motiwala, Andy Bavier and Nick Feamster

1. Introduction

Users and operators of communication networks must be able to quickly diagnose (*i.e.*, detect and locate) network faults. The large body of sophisticated network diagnosis tools used are typically "out-of-band": they introduce a stream of probes that are separate from the data traffic experiencing a network fault. Out-of-band techniques only collect information about the network conditions at the time the tool is running, so they may fail to capture transient faults entirely. Also, the measurements may experience very different behavior than the data traffic that is actually subject to the network fault. To make matters worse, probing tools introduce additional traffic that may itself alter network conditions.

We propose a complementary technique, which we call *in-band network path diagnosis* (INPaD). The basic idea is simple: *include information associated with network faults in the data packets themselves*. Since INPaD approaches embed information on network conditions in the data stream itself, they can directly record the symptoms of even transient faults without the overhead of regular probing. We have developed Orchid which is a specific INPaD scheme and evaluated Orchid on VINI[1].

2. Orchid Design

Orchid inserts a new header in the data packet, which the routers use to communicate loss and other related information to the end user. We introduce a separate probe packet for recording the IP addresses of routers along the path. The probe packet is only used at the start of the flow and whenever a change in the path is detected. Routers maintain a counter corresponding to every active flow. The shim header contains three fields for detecting the router responsible for dropping packets: (1) *F*low - This 4-bit field indicates the sequence number for this packet in the flow. (2) *I*ncr - This 4-bit field indicates the number of packets lost by a router. (3) *R*outer ID - This 8-bit field is used to mark the hash of the router that lost packets. A router which has dropped a packet, will find a difference in the flow counter values in the next packet and the one stored at the router and would hence mark the other two fields in the packet indicating where and how many packets were dropped. The shim header can also be used to collect information about the router with the highest queueing delay, jitter along the path.

3. Implementation

VINI is a testbed for network experimentation that employs virtualization to run experiments simultaneously on the same physical infrastructure [1]. VINI provides both a realistic environment for performing network experiments and a high degree of control over the network conditions. We used PL-VINI, the VINI prototype running on PlanetLab [2], to evaluate Orchid. PL-VINI is an IP routing overlay consisting of both data and control planes. The PL-VINI architecture consists of a Click router [3], responsible for implementing the virtual topology via UDP tunnels, and a User-Mode Linux environment for running routing software. Our implementation uses the Quagga routing software in UML as the control plane.

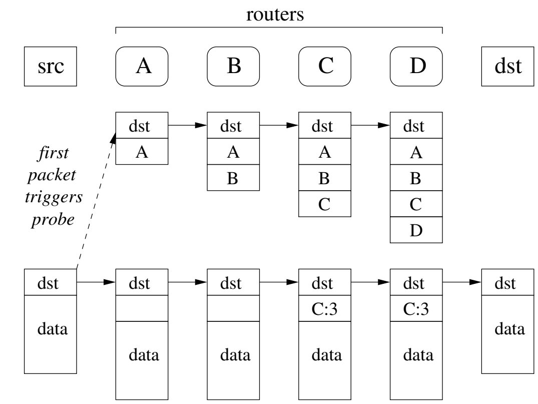

Figure 1: Orchid Overview. The first data packet between src and dst triggers Router A to send a probe packet. Router A adds the Orchid header to data packets, and other routers along the path mark them. Router D accumulates Orchid data and strips the Orchid header.

We created the following Click elements for our prototype and integrated them into PL-VINI.

Pkt Gen(num_packets, src_ip, dst_ip) Creates a UDP flow of $num\_packets$ 400-byte packets, spaced about 200ms apart. The element sends an Orchid probe packet, followed by the UDP data packets containing the custom header. It increments the flow counter value for each packet sent.

Modify Ing() Placed at each router's ingress link; stamps incoming packets with their arrival times. This element is used to calculate the queueing delay of the packet through the router.

BurstDrop(drop_prob, burst_size, burst_deviation) Drops a burst of packets, where the size of the burst has a Gaussian distribution with mean *burst_size* and deviation *burst_deviation*. The *drop_prob* determines the overall packet loss rate.

Modify Pkt(router_ip) Placed at the egress link of the router. It performst the task of comparing the flow counters and marking the packet if necessary.

Our evaluation shows that Orchid can diagnose the network fault (packet loss, reordering, etc) with high accuracy and gives some erroneous results only in the case when the network has multiple points of loss.

REFERENCES

[1] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford. In VINI Veritas: Realistic and Controlled Network Experimentation. In *Proc. ACM SIGCOMM*, pages 3–14, October 2006.

[2] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: an overlay testbed for broad-coverage services. *ACM Computer Communication Review*, 33(3):3–12, 2003.

[3] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, August 2000.