Network Troubleshooting on Data Plane Coattails

Murtaza Motiwala and Nick Feamster
College of Computing, Georgia Tech

1 Motivation

The Internet's increasing complexity has made it extremely difficult for network operators to efficiently detect and troubleshoot network problems, which can cause both performance degradations and downtime. Although tools exist to help network operators detect configuration faults before their configurations are deployed on a live network, operators currently have very poor resources for identifying the cause of a failure when the network does not behave as expected (e.g., due to node or link failure). *Debugging* routing problems on the Internet—which involves both detection and identification of the problem—is important not just for enabling network operators to run their networks efficiently but also for end users and enterprises to hold ISPs accountable to service-level agreements for specific availability and performance guarantees.

We hold that the difficulty of troubleshooting network faults on today's networks results from the fact that the probes required to gather information about network faults are separate from the data path.

Despite the growing need for both users and operators to be able to quickly isolate and mitigate network problems, this task is becoming increasingly difficult with the large scale deployment of NATs, ISPs blocking ICMP packets because of security/management reasons, etc. The predominant tools for troubleshooting today's networks—ping and traceroute—are not sufficient. Ping is blocked by ISPs at many places, it lacks the capability to detect network re-ordering of packets, and it is processed on the "slow path" of routers, it cannot detect packet reordering in the data path, and it cannot detect hosts running behind firewalls and NATs. Additionally, as many ISPs are deploying MPLS in their core, IP-level paths are becoming invisible to conventional tools like traceroute. Despite its conceptual appeal, traceroute does not trace the path of any single packet through the network; indeed, the output from a single traceroute may not reflect a path taken by *any* single packet in the data plane. Finally, these tools are hampered by network administrators, who often filter protocols like ICMP entirely.

The IP Measurement Protocol (IPMP) [1, 2] attempts to address some of the shortcomings of ping and traceroute and also to add additional debugging capabilities to probe packets. IPMP is based on an echo request and reply packet exchange for measuring packet delay and associated path metrics and is similar to the technique that ping uses with ICMP echo capabilities. On an IPMP echo request packet, each host along the path records its IP address and a time-stamp value. Even though IPMP tries to mitigate the DoS attack problems by making the recording of IP addresses and time-stamp optional, it does not eliminate the problem completely [2] as attacks can be disguised as legitimate measurement traffic Also, IPMP packets being separate from data packets they may not detect any transient network problems.

2 Towards diagnosis in the data plane

Although we believe that IPMP is a good starting point for thinking about how to embed more useful data in probing packets, we propose that, instead of sending separate probes for performance monitoring and troubleshooting, *the information needed to carry out troubleshooting enquiries should be carried in the data packets themselves.* With such an approach, the router simply forwards data packets as usual, with the exception of modifying a few additional fields (determining what to embed in the existing fields and how to encode the most information possible is one of the major challenges). This idea borrows from other similar protocol extensions used for similar purposes: for example, there are methods to use fields in data packets for informing the end systems about network conditions like the ECN flag in the TCP header, but these techniques are limited in their scope.

If done correctly (a big if!), embedding information in data packets could allow for *proactive* debugging: the information that is needed to debug a problem is available at the time when the performance degradation is occurring. When end hosts witness a transient failure or performance degradation in the data plane, today's natural approach is *reactive*: users and operators can send additional probe packets after a failure has occurred, but they cannot "rewind" and see the conditions of the network at the time the failure actually occurred.

Because the available fields in the data packets are limited, a data-plane debugging protocol may not be able to encode all of the information needed to perform the complete set of functions offered in IPMP. Instead of trying to solve many problems at once (i.e., determining packet reordering, loss, round-trip time, and bandwidth availability on the path) with a single—but separate—probing protocol, we argue instead that perhaps we should focus on some critical subset of these properties (e.g., reachability, packet loss, and reordering) and figure out how we might wedge these into data plane packets with minimal overhead.

3 Challenges and Research Agenda

Despite the appeal of embedding information for helping network troubleshooting in the data packets, we must solve many open questions, including the following:

- How does the sender retrieve information "stamped" on data packets on the forward path? We propose the following three techniques:
  - There is a likelihood of another connection being open from the receiver to the sender. The useful debugging information collected from the data packets can be piggy-backed on those data packets. Also, for reliable transport protocols, the ACK packets can be easily use to piggy-back the data.
  - The receiver can periodically update the sender with the debugging statistics gathered from the connection.
  - The receiver can collect the information from the data packets, merge and publish it to some common repository from which the sender can retrieve it.
- Does the scheme impose too much overhead on routers? If so, can routers selectively mark packets with the relevant data plane information while still proving enough information to troubleshoot routing problems?

Evaluation plan. The Virtual Network Infrastructure (VINI) [3] could serve as a platform for evaluating the effectiveness and overhead of the troubleshooting techniques proposed above. VINI provides both a realistic platform and high degree of control over the network conditions. For example, VINI provides researchers with the ability to arbitrarily fail links or induce other network faults in the network. We believe these characteristics will prove useful this line of research.

References

[1] **M. J. Luckie, A. J. McGregor and H. Braun**, Towards Improving Packet Probing Techniques, *ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[2] **A. J. McGregor**, "The IP Measurement Protocol." "http://watt.nlanr.net/AMP/IPMP/".

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