Isolating Network Failures with Controlled Spoofing

Senior Thesis Proposal

Colin Scott

Introduction

Reliable network connectivity is essential for the success of content and cloud providers. For these Internet-based companies, even short-lived outages can result in lost revenue, damaged reputation, and wasted effort.

Nonetheless, connectivity problems are a common occurrence for many providers. To make matters worse, when outages occur network operators often lack the tools and resources needed to gain sufficient visibility outside of their own network, limiting their ability to isolate and diagnose problems in a timely manner. For example, in order to obtain a better view of an outage operators often resort to posting on forums asking for other operators to issue traceroutes on their behalf. Another problem is that relatively little is known about the causes or characteristics of network outages in general. Such information is necessary for designing networks which are failure resistant in the first place.

For my senior thesis I propose to combine new and existing measurement techniques to build a system to identify and isolate the location of network failures. Such a system must address several key issues. First, it must function during abnormal network conditions; part of the reason operators lack visibility is that traditional tools such as traceroute do not function properly or provide useful information during outages. Second, the system must quickly and reliably isolate failures; this is especially important since providers may lose thousands to millions of dollars per hour due to loss of connectivity. Finally, the system must not require instrumentation on infrastructure which is not owned by the network experiencing an outage.

A failure isolation system would enable further research in this area. For example, once we have an automated tool for isolating failures, we can investigate the effectiveness of BGP path poisoning and other techniques which allow operators to affect the routes used by problematic ASes without requiring human interaction. As another example, failure isolation would enable a study on root cause analysis, providing invaluable information about the characteristics of network failures in general.

Motivating Study

In order to quantify the need for failure isolation, we plan to perform an initial study of Internet connectivity. While several previous studies have been performed [Hubble, Planetseer], they typically only focus on long-lived outages (> 15 minutes). Moreover, these studies tend to be launched from academic networks, limiting their coverage of outage events experienced by commercial organizations.

To overcome both of these shortcomings, we will use distributed Amazon EC2 vantage points to track outage events among a set of strategically chosen targets. As a commercial network, EC2 provides us with a viewpoint on Internet connectivity which has yet to be

publicly investigated. In order to detect both short- and long-lived outages, we intend to keep the target list small (under 500 targets) so that we can issue measurements on a frequent basis.

Proposed Failure Isolation Technique

Failures and misconfigurations can either affect forward paths, reverse paths, or both. Consequently, distinguishing these three cases is the first step towards isolating the cause of an outage.

Issuing a traceroute from a machine affected by an outage does not help a network operator infer the direction of the failure. This is because probes need to traverse both the forward and reverse path for any given hop, such that a failure in either direction results in dropped probes. Therefore network operators need an alternate measurement technique for failure isolation.

As was demonstrated by Hubble, it is often possible to infer the direction of an outage by using spoofing; that is, we can have a source S (which we control) send a probe towards a destination D, spoofing as another vantage point S' such that the probe traverses only the forward path between S and D. If S' successfully receives the probe, we know that the failure must be on the reverse path.

While this technique is certainly useful, it does not help us identify the exact router or link causing an outage. We believe we can accomplish this goal by combining Hubble's technique with ttl-limiting, as well as leveraging the reverse traceroute system.

To measure the forward path between S and D, we propose the following technique: we will have S send ttl-limited probes towards D as in a normal traceroute, except that S will again spoof as S'. As described above, this ensures that the probes traverse only the forward path from S to D, except that now S' has gained information about which routers lie along on the forward path.

Forward Path Failures. If the problem is on the forward path, S' will never see a ping response from D. To check that the problem isn't also on the reverse path, we can either issue a reverse traceroute from D to S, or we can simply have S' spoof a probe as S towards D.

If the problem is only on the forward path, we examine the ICMP time exceeded messages received by S' in order to find the router adjacent to the failure. To ensure that we correctly interpret the order of the received probes, we encode the original ttl in the header of the packet, which is then reflected in the time exceeded message.

Bi-directional Failures. If we find that the failure affects both the forward and reverse path, examining the last ICMP time exceeded packet received by S' will again suffice to isolate the problem.

Reverse Path Failures. Finally, if the spoofed forward traceroute was successful we know that the problem must be on the reverse path. We can further isolate the failure by issuing reverse traceroutes from each hop on the forward path in turn back towards S. When we find the first hop that causes a failed reverse traceroute, we can consult historical reverse paths for that hop in order to obtain a reasonable guess of which link or router has failed.

Note that we cannot simply issue a reverse traceroute from D to S, as measurement probes destined for D will not reach S on their return path.

Multiple failures. The cases above have assumed that an outage is caused by a single router or link. In the unlikely event that multiple points of failure exist, our technique will tend to identify the failure closest to the source. In these cases we can simply iterate: after having fixed the failure closest to the source, subsequent attempts will identify the next closest failure, and so on.

Limitations

The technique outlined above is constrained by several limitations. First, it requires that the ISP of the network experiencing an outage does not institute egress spoof filtering. Second, it requires access to multiple vantage points. To overcome this limitation we plan to offer the use of PlanetLab receiver nodes as part of the reverse traceroute service. Third, it assumes that the destination D has a working path towards S'. To increase the likelihood that spoofed probes will be picked up, we can coordinate several vantage points to act as receivers, thereby increasing the diversity of return paths. Finally, since the technique depends on reverse traceroute, for full accuracy it requires that routers along the reverse path handle IP options properly. See the reverse traceroute paper for further discussion of this issue.

Failure isolation evaluation

In order to evaluate the effectiveness of the proposed approach, we intend to test our system on real cases of connectivity loss. To this end, we will again monitor a set of targets in the hope of identifying and reacting to network outages in real-time. More specifically, we will use a set of vantage points to send probes to a small target list on a regular basis. Whenever a VP observes multiple consecutive dropped probes to a particular target, we will launch the isolation technique described above. We will use the gathered to data to evaluate how often we can identify the exact location of a failure, how quickly measurements can be coordinated and executed, and ultimately whether the approach is sufficiently feasible for use by commercial network operators.