

## Can Information from End Systems Improve Routing?

Nick Feamster

College of Computing, Georgia Tech

At the same time as Internet users are increasingly demanding new types of services, high availability, good performance, and protection from attacks, network operators are facing pressure to provide these additional features and functions, while at the same time minimizing the time to respond and correct problems when they arise. These increasing demands on network operators make an already bad situation worse: network operators cannot quickly detect, diagnose, and mitigate network performance and security problems.

Although network operators have various tools to help them determine whether their configurations are clearly incorrect, they do not have tools and systems that provide them any recourse when the network does not behave as expected. A customer call to the network operations center is often an operator's first indication that the network is not behaving as expected. This situation is not for lack of effort by network researchers and operators alike to come up with various techniques for fault detection and troubleshooting. Although many of these solutions provide adequate workarounds, we argue that one of the key shortcomings of existing solutions is that routers and network devices (and, as a result, network operators) do not have the necessary measurements of end-to-end performance to quickly detect when a performance or security problem arises. On the flip side, while end-hosts have fine-grained measurements about network performance, they lack (1) a network-wide view that allows them to determine the significance of an observed problem; (2) a means for communicating these observations in a *bandwidth-efficient* manner back to network devices and network operators; and (3) the ability to exercise any *control* over what measurements are taken in the network.

We present examples where the disparity of information collected by routers and end users represents a missed opportunity for better detection, diagnosis, and mitigation of network faults. We also describe examples where sharing routing data across routers can help detect actionable network events that conventional mechanisms, such as those that rely on processing a stream of routing data would fail to detect. We then suggest that distributed network sensors could collect fine-grained information about network performance to allow routers to adapt in real-time to changing network conditions.

### Disparate Data Streams Frustrate Fault Detection

Because they reside in the middle of the network (*i.e.*, at vantage points traversed by many different network paths), routers are well-situated to collect vast amounts of both traffic and routing data. Additionally, these devices have the ability to take action to correct or mitigate any problems that arise, and the operators of these networks also have information about network configuration that is available to them that can help diagnose any problems that are detected.

Unfortunately, routers also have limited resources for data collection, no mechanisms for processing any of this data inline, no hints about what traffic crossing the network might be interesting for solving a particular problem, and no mechanisms for correlating data streams as observed by network devices at other positions in the network. Consider the following examples:

- *End-to-end performance.* The poor performance of many routing protocols (*e.g.*, BGP [8]) at detecting and reacting to failures of nodes and devices in the network result in situations where small groups of end-hosts can collaborate to notice failures and divert traffic around those failures [1]. Although end hosts can certainly work *around* network failures, information about the nature of these failures and performance degradations (*e.g.*, which hosts observed the failures along which paths) can provide useful information to network operators both for performing post-mortem analysis and for detecting and correcting these problems.
- *Network instability and failures.* In the absence of fine-grain active measurements, network events—link or router failures, routing instability, etc.—are typically an operator's first opportunity to detect a network problem (perhaps before an end user might even notice it). Unfortunately, as in the case of end-to-end performance monitoring, the sheer quantity of routing data is too massive and too noisy for an operator to process in real time. Some of our ongoing work demonstrates that, by combining data *across* multiple routing streams, operators can tease relevant groups of routing messages from noise and detect network events that they would otherwise miss [4].

In each of these cases, better facilities for correlating both data plane and control plane information could improve detection, diagnosis, and mitigation and might even also allow routers to automatically take corrective action, such as routing around these failures. Our challenge, then, is to both develop fault detection algorithms and build such a system. One of the most important questions that designing such a system entails are those that deal with placement of function, in particular, determining what functions should be placed in routers and network devices vs. in end hosts vs. off the network path entirely.

### Collection, Processing, and Correlation of Disparate Network Data Streams

Others have also recognized the potential benefits of using host measurements to improve network monitoring [6, 7]. We build on this previous work by not only using the data from end hosts as data for troubleshooting, but also by allowing network devices and routers and centralized measurement “coordinators” to take action based on the hints and alarms provided by end hosts. Specifically, our system has three components:

- 1. Sensors for fine-grained data collection at the network edge.** End hosts can monitor network traffic at a finer granularity than monitors that are placed in the middle of the network, which see a broader cross-section of traffic, but at much higher volumes. These *sensors* (*i.e.*, deployed either in cooperation with other networks or with untrusted end hosts) could exchange information with network devices in the core to allow them to focus on specific subsets of traffic, to perform remediation (*e.g.*, installing filters), or both.
- 2. “Hints” and alarms.** End hosts collect end-to-end performance metrics to various Internet destinations, act as network honeypots to learn about infected hosts in other regions of the Internet, or serve as spam sinkholes to gather information about spamming hosts in other parts of the Internet, etc. Similarly, the routers themselves could log routing messages and pass some or all of these messages to a coordinator that detects actionable network events. Network devices can then information about the existence and location of some network event to gather more data about the event.
- 3. Actuators in routers (and other network devices).** Network devices can take action based on the the hints and alarms delivered from sensors. Possible actions that a router might take include installing filters to drop certain traffic, instructing the sensors to take more measurements, adjusting in-network passive monitoring (*e.g.*, traffic sampling frequencies), automatically adjusting route selection or the configuration, and so forth.

Solving specific problems will require different types of sensors, different types of “hints” to be passed from the sensors to the network devices, and different sets of actions that can be performed by the network devices themselves. Combatting spam might require the deployment of spam traps across a wide array of domains and passing information about spamming IP addresses (or address ranges) to devices in the network that perform some type of filtering in the network itself, rather than close to the network edge. On the other hand, debugging network performance problems requires the ability to measure the network from a diverse set of vantage points, so the hints to the network devices might indicate the existence of the severity of a failure, and the network devices themselves may respond by initiating more active measurements from the deployed monitoring infrastructure.

### Placement of Functionality: What Functions Must Routers Support?

Because the range of network operations tasks that can benefit from correlation of data streams appears broad, we believe that the network should provide monitoring, detection, and mitigation functions must be general. In particular, it is clear that simply logging routing data for subsequent post-processing will not suffice. The system must be intelligent enough to correlate routing streams to detect actionable network events, take input from end hosts and adjust monitoring accordingly, etc.

Simple traffic sampling techniques are inadequate, and routers should likely expose some API that allows operators (or even end hosts!) to control what measurements the router takes, how these measurements can be adapted, and who can control the adaptation. Even within the context of traffic sampling, an important question is how much flexibility routers must have to be able to capture the data that allows network operators to perform the network operations tasks listed above. For example, what new detection algorithms could be possible if routers could take complete packet captures upon request from an end host (and how would the system enforce access control to ensure that requests for packet capture were warranted)? Beyond traffic sampling, we must determine how much processing the routers themselves should do before

passing information to other nodes for correlation and processing: while some amount of pre-processing can reduce the bandwidth requirements for exchanging statistics and computational requirements on centralized processing nodes, such processing also spends precious router cycles that could be used either for performing measurements, forwarding packets, or providing other features.

### Other Research Challenges

In addition to the above challenges, such a system must also leverage algorithms for data mining in massive data sets. While many existing network anomaly detection algorithms exist [2, 5], these algorithms may need to be adapted in ways that permit distributed, real-time detection; although some ongoing work has begun to study this problem (*e.g.*, [3]), it remains to be seen how such solutions can be implemented in an actual system. Finally, when sensors are distributed across multiple competing networks, the system must solve many complex issues involving privacy, trust, and incentives.

### References

- [1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. 18th ACM Symposium on Operating Systems Principles (SOSP)*, pages 131–145, Banff, Canada, Oct. 2001.
- [2] N. Duffield. Simple Network Performance tomography. In *Proc. ACM SIGCOMM Internet Measurement Conference*, Miami, FL, Oct. 2003.
- [3] L. Huang, M. Garofalakis, J. Hellerstein, A. D. Joseph, and N. Taft. Toward Sophisticated Detection With Distributed Triggers. In *ACM SIGCOMM Workshop on Mining Network Data (MineNet-06)*, Pisa, Italy, Sept. 2006.
- [4] Y. Huang, N. Feamster, A. Lakhina, and J. Xu. Exposing routing problems with network-wide analysis. Technical report, Georgia Tech, May 2006. Number forthcoming. Available upon request.
- [5] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proc. ACM SIGMETRICS*, pages 61–72, New York, NY, June 2004.
- [6] R. Mortier, R. Isaacs, and P. Barham. Anemone: Using End-Systems as a Rich Network Management Platform. Technical Report MSR-TR-2005-62, Microsoft Research, June 2005. <http://research.microsoft.com/projects/anemone/papers/tr-anemone.pdf>.
- [7] V. Padmanabhan, S. Ramabhadran, and J. Padhye. Client-based Characterization and Analysis of End-to-End Internet Faults. Technical Report MSR-TR-2005-29, Microsoft Research, Redmond, WA, Mar. 2005. <http://research.microsoft.com/research/pubs/view.aspx?type=Technical\%20Report&id=879>.
- [8] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, Jan. 2006. RFC 4271.