BGPmon: Towards a More Robust BGP Monitoring System

M. Cazares[1], J. Bartlett[2], C. Olschanowsky[2], D. Massey[2]
[1]Texas State University, San Marcos, Texas 78666
[2]Colorado State University, Fort Collins, Colorado 80523

INTRODUCTION

Society relies on the Internet for communication, business, and entertainment. However, there are issues with reachability within the Internet, e.g., when one Internet subset suddenly cannot reach another subset. Such issues can arise from malicious attacks or misconfigurations. Detecting these problems is the first step to combating large-scale unreachable Internet space.

A solution to detection is BGPmon - an Internet routing monitoring system that enables researchers and operators to monitor routing issues on the global scale and in real-time. Previous work is expanded by increasing BGPmon robustness when handling corrupt input.

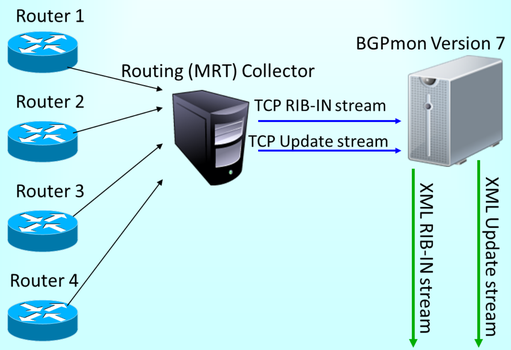
WHAT IS BGPmon?

BGPmon is a **B**order **G**ateway **P**rotocol **Mon**itoring System. The Border Gateway Protocol (BGP) dictates how all Internet traffic is routed around the world.

BGPmon provides a first-line defense against worldwide Internet outages such as the April 8, 2010 China Telecom prefix hijack incident.

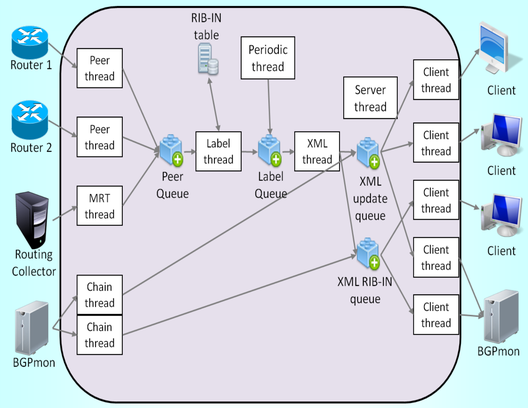
HOW DOES BGPmon WORK?

BGPmon Input and Output



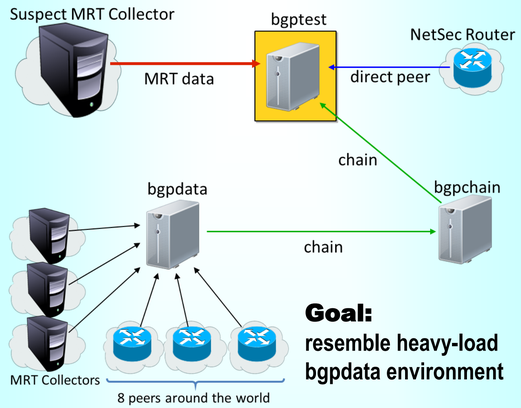
HOW IS BGPmon BUILT?

BGPmon Architecture



TESTING BGPmon

Testbed Setup



Goal:
resemble heavy-load bgpdata environment

SYSTEM-LEVEL GOALS

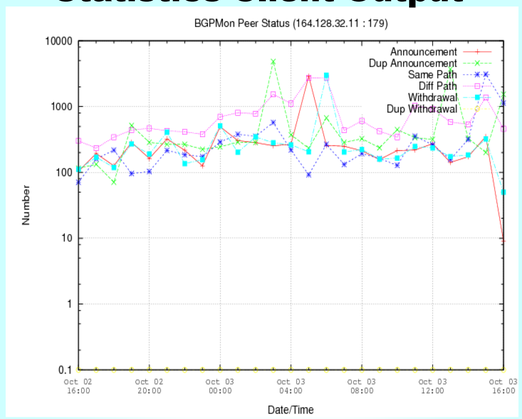
Scalability – connect to a high number of peers; increases volume of monitored data.

Delivery of Data – data must be delivered in real-time; increases usefulness of monitored data.

Robustness – errors in incoming data, such as incomplete messages, must be handled; increases system reliability while processing monitored data.

RESULTS AND EVALUATION

Statistics Client Output



Web client output aids in analysis of BGPmon testing.

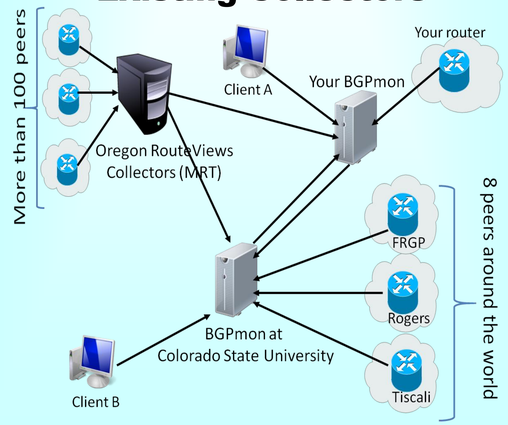
Outcome of Testing

Using 5-minute MRT input file led to identification of corrupted MRT messages in this file.

Implementing incoming corrupt message handling increased average BGPmon up-time from a few minutes to more than several weeks.

EXPANDING BGPmon COVERAGE

Merging Collector with Existing Collectors



FUTURE WORK

Data Storage – design, implement, and deploy BGPmon Archive Client that permanently stores BGPmon XML output.

Performance Analysis – deploy more sophisticated statistics web client that focuses on measuring BGPmon performance.

Code Maintenance – port code to object-oriented style to decrease time spent on system testing.

GET BGPmon

Download:
http://bgpmon.netsec.colostate.edu/index.php/download

Documentation:
http://bgpmon.netsec.colostate.edu/index.php/documentation

Web Client Live Data:
http://bgpmon.netsec.colostate.edu/index.php/live-data

REFERENCES

BGPmon Documentation. (2010). *BGPmon: Using Real-Time Data in Research and Operations.* Retrieved from http://bgpmon.netsec.colostate.edu/download/doc/BGPmon-deployment.pdf

Matthews, D., Parrish, N., Yan H., & Massey, D. (2008). BGPmon: A real-time, scalable, extensible monitoring system. *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC).*

Matthews, D., Yan H., & Massey, D. (2008). BGPmon Documentation. *BGPmon Administrator's Reference Manual*. Retrieved from http://bgpmon.netsec.colostate.edu/download/doc/arm.pdf

Yan H., Strizhov M., Burnett K., Matthews D., & Massey, D. (2010). BGPmon Documentation. *BGPmon Version 7 Implementation and Technical Specification.* Retrieved from http://bgpmon.netsec.colostate.edu/download/doc/techreport.pdf

ACKNOWLEDGMENTS

This work was made possible by the Distributed Research Experiences for Undergraduates (DREU) Program:
http://cra-w.org/distributed-research-experiences-for-undergraduates-dreu