BGPMon and NetViews: Real-Time BGP Monitoring System

He Yan, Dan Massey
{yanhe,massey}@cs.colostate.edu
Computer Science Department
Colorado State University

Ernest McCracken, Lan Wang
{lanwang,emccrckn}@memphis.edu
Computer Science Department
University of Memphis

I. Introduction

Diagnosing interdomain routing problems can be extremely hard due to decentralized network architecture and extremely large routing tables. Monitoring interdomain routing through BGP is important for both operations and research; a number of public and private BGP monitors are deployed and widely used. These existing monitors typically collect data using a full implementation of a BGP router. In contrast, BGPmon [1] eliminates the unnecessary functions of route selection and data forwarding to focus solely on the monitoring function. BGPmon uses a publish/subscribe overlay network to provide real-time access to vast numbers of peers and clients. All routing events are consolidated into a single XML stream. XML allows us to add additional features such as labeling updates to allow easy identification of useful data by clients. Clients subscribe to BGPmon and receive the XML stream, performing tasks such as archiving, filtering, or real-time data analysis. BGPmon enables scalable real-time monitoring data distribution by allowing monitors to peer with each other and form an overlay network to provide new services and features without modifying the monitors.

NetViews is a system that works in conjunction with BGPMon. The NetViews system acts as a client to the BGPMon system receiving updates and forwarding the information to NetView clients. In order to maintain scalability to thousands of clients, NetViews makes use of multicasting through overlay networking provided by the HyperCast [2].

II. BGPmon

Open source routing software that is currently used as a collector typically implements a full routing protocol, including receiving routes, applying policies, setting forwarding states, and announcing routes to peers. These activities involve considerable complexity, but none of these actions are needed to be collector. A collector simply needs to receive and log routes. Our new collector design shown in Figure 1 focuses on a narrow set of data collection functions. By focusing on the collection functionality and eliminating unnecessary tasks, the new collector is able to scale up and support more peer routers while making the data available in real-time to a potentially vast number of clients.

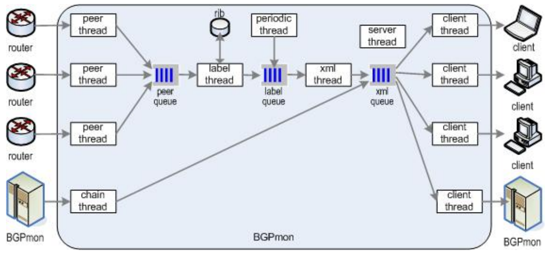


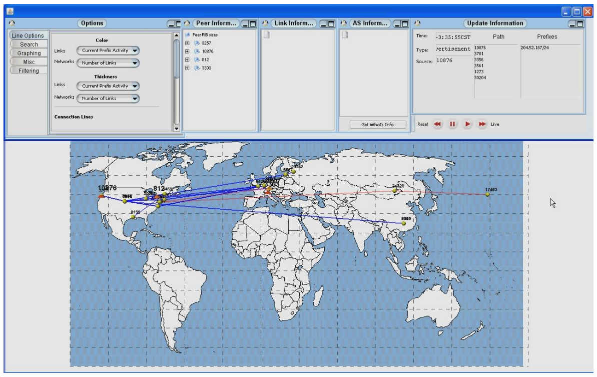
Fig. 1. BGPmon architecture.

To support hundreds of peer routers and clients, one would like to scale out across multiple systems by adding more collectors and distributing the services. At the same time, a client should see a single monitoring service and be unaware that the implementation of the service may be done through multiple collectors. Our approach is based on publish/subscribe overlay networks that consist of brokers, publishers, and subscribers. The brokers form the overlay network, allowing publishers to send event streams to the overlay network and allowing subscribers to receive event streams from the overlay network. Publishers and subscribers interact only with brokers, not with each other, allowing the overlay network to insulate publishers and subscribers from each other.

To improve fault tolerance, multiple brokers may monitor the same or different peers in an AS, yet appear to a client application as a single subscription. This allows critical applications to continue to receive event streams in the case of a failure of a peer, monitor, or broker. Our system implementation begins with BGPmon, a simple monitoring system now available that incorporates all three functions: publish, broker, and subscribe. The second stage is BGPbroker which separates these functions to support Internet scale and additional services.

III. NetViews

Netviews [3] will help network operators diagnose issues such as attacks and misconfigurations and monitor the connectivity of thier network. Netviews visualizes AS level topology based on the state of a peered router's routing table. Beyond just graphically representing the routing table Netviews can also display route state changes and statistical information in real time.



Color

Links: Current Prefix Activity

Networks: Number of Links

Thickness

Links: Current Prefix Activity

Networks: Number of Links

Connection Lines

Peer RIB sizes

3257

10876

812

3303

Time: -3:35:55CST

Type: vertisement

Source: 10876

Path	Prefixes
10876 3701 3356 3561 1273 30204	204.52.187/24

Fig. 2. NetViews Interface

Previous tools available like BGPlay [4] require batch updates and preprocessing to be performed server side. Thus they require the user to specify a time period. NetViews receives BGP updates in real-time from BGPMon and clients process information the user is interested in while forwarding all data to others on te overlay network.

NetView's visualization module, shown in figure 2, takes raw BGP information from BGPMon peers to create a spanning tree of routable AS's for each peer. These are used to infer the topology graph. AS's are represented as nodes and their geocoordinates are determined from address information obtained from WHOIS servers. Links between AS's are represented by edges in the graph. Different types of information can be visualized by altering both the size and color of nodes and edges.

With the NetView client a researcher or network operator can subscribe for different types of information. They can subscribe to only receive information for specific prefixes,all prefixes for a specific AS, or all prefuxes for all AS's. Based on these subscriptions clients will also obtain current routes from each peer's routing tables by requesting them from the Databroker. A user will then be able to see the connectivity of networks and prefixes in real-time. These live BGP sessions can be saved to the client's local computer and replayed at a later time.

IV. Demo Description

In our demo we plan on showing various abilities of BGPmon and NetViews. BGPmon is being integrated with the Oregon RouteViews project. In the demo, we will show how interested researchers can access RouteViews update data in real-time in XML. BGPmon currently receives data from 2 of the RouteViews collectors and provide access to approximately 25 peers in IPv4 or IPv6. Data feeds are available to any interested researcher and, along with the RouteViews team, we hope to have all RouteViews collectors and peers included at the demo. Our results also illustrate the new XML format and show how new applications such as the NetViews is now made possible by this change in BGP monitoring. The NetViews Databroker is the portal between BGPMon and the NetViews overlay network. The Databroker not only forwards BGP updates from BGPMon but also maintains an up to date routing table for each peer. We will show how users can easily configure NetViews to subscribe to prefixes of interest and monitor the connectivity of their own AS. We will also show how users can interact with the topology graph, alter the information visualized, and do searches.

V. Future Work

Based on discussions and user feedbacks, we are very encouraged by the deployment of BGPmon and NetViews thus far. These discussions and feedbacks also point out the future work of BGPmon and NetViews.

For BGPmon, the future work consists of 2 parts: a revised approach to send routing tables and a new queue mechanism for handling slow clients. First currently BGPmon sends both incremental updates and tables in the same XML stream. Due to the huge amount of data of routing table transfer, the important incremental updates get delayed in the internal queues of BGPmon. The possible solution is to introduce a second XML stream. The first XML stream contains only incremental updates. A new second XML stream will contain routing tables. In this way, the clients who want the real-time incremental updates should subscribe to the first XML stream and those who want routing tables should subscribe to the second stream. Secondly instead of disconnecting the slow clients as we are doing now, we will just skip messages for slow clients as most of them will come back right after being disconnected.

For NetViews, there are three directions of the future work. First, we plan to show the router level view by utilizing traceroute probes. This way a network operator can compare their network's BGP path with the IP forwarding path obtained from traceroutes to their network. Secondly, we plan to give clients the ability to communicate with each other using instant messaging and conferencing service within the NetViews client, so that they can easily coordinate their reactions to abnormal events. These two capabilities will distinguish our work from server-based monitoring tools such as Cyclops [5] and Routing Intelligence [6]. Finally, we will add more types of alerts to the users.

References

[1] H. Yan, R. Oliveira, K. Burnett, D. Matthews, L. Zhang, and D. Massey, "BGPmon: A real-time, scalable, extensible monitoring system," in *Proceedings of the Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH)*, Mar. 2009.

[2] J. Liebeherr and T. K. Beam, "HyperCast: A protocol for maintaining multicast group members in a logical hypercube topology," in *First International Workshop on Networked Group Communication (NGC '99)*, Jul. 1999.

[3] E. A. McCracken and L. Wang, "NetViews: Real-time visualization of Internet topology through peered vantage points," http://netlab.cs.memphis.edu/projects_netviews.html.

[4] L. Colitti, G. D. Battista, and F. M. M. Patrignani, "Visualizing interdomain routing with BGPlay," *Journal of Graph Algorithms and Applications*, 2005.

[5] Y.-J. Chi, R. Oliveira, and L. Zhang, "Cyclops: The AS-level connectivity observatory," *ACM SIGCOMM Computer Communication Review*, Oct. 2008.

[6] Renesys, "Routing Intelligence," http://www.renesys.com/products_services/routing_intelligence/.