Reachability Monitoring and Verification in Enterprise Networks

Bo Zhang T. S. Eugene Ng Guohui Wang

Rice University, Houston, TX, USA

{bozhang,eugeneng,ghwang}@cs.rice.edu

ABSTRACT

Enforcing correct reachability is crucial for an enterprise network to achieve access control, privacy, security and so on. Many sophisticated mechanisms such as router ACLs and firewalls have been developed to enforce the desired reachability. In addition, many other factors such as network dynamics can also impact the network reachability. Thus it is challenging to configure the reachability correctly. Therefore, the ability to monitor and verify network reachability becomes valuable for many tasks such as verifying the original intent of the network administrator and trouble-shooting reachability problems. In this poster, we present efficient algorithms to monitor and verify the reachability of all-pairs nodes in real-time.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks:**]: Network Operations

General Terms

Algorithms, Management, Verification

Keywords

Reachability, Monitoring, Verification, Enterprise Networks, Management

1. INTRODUCTION

Enforcing correct reachability is critical for an enterprise network to achieve global security policy, access control, privacy protection and so on. As a result, many sophisticated mechanisms such as router access control lists (ACLs) and firewalls have been deployed to help manage network reachability. Network dynamics (e.g., routing changes, topology changes and configuration changes) also constantly affect network reachability. Thus it is challenging to tell whether two hosts can communicate without knowing all network packet filters, routing policies, topology and so on.

Therefore, the ability to analyze and verify the network reachability in real-time would be valuable for many network tasks such as reverse-engineering the reachability of an existing network and trouble-shooting reachability problems. Previous work addressed this goal in different ways. Significant progress in monitoring routing-related reachability problem at both a global level (e.g., [2, 6]) and an intra-domain level (e.g., [5]) have been made but they mainly focus on routing. Many "ping" and "traceroute" based tools have also been developed. However, to test the reachability of all-pairs nodes for all types of traffic, prohibitive amount of measurements must be injected into the network. In order to overcome the shortcomings of the direct measurement approach, researchers (e.g., [7, 1, 4]) have proposed to statically analyze the configuration files of routers and firewalls to calculate reachability. Since routing information is not used in the static analysis approach, all possible paths between the two nodes have to be considered. Consequently, the calculated reachability is just an upper bound of the instantaneous reachability and any changes to the network may require recalculating reachability of all-pairs nodes. What is worse, the complexity of the algorithm is exponential in the worst case. [1] reports that it takes 3 seconds to calculate the reachability for one pair of routers on a 20-router topology. Given that a large enterprise network may have thousands of routers [7], the static analysis approach will not scale. In summary, existing approaches are either impractical (due to the daunting overhead) or incomplete (only focusing on routing related problems).

Our goal is to monitor and verify the reachability of all-pairs nodes in real-time. Our contribution is the exploration of a new design point where routing information is leveraged to provide real-time monitoring of the instantaneous reachability. The idea is to run a monitoring agent on each gateway to collect the up-to-date packet filtering polices (e.g., ACLs) and forwarding states. All the collected information is then sent to a central coordinator who will calculate the all-pairs reachability. Although the basic idea is conceptually clear, two key challenges remain to be addressed: *(1) How can we leverage all collected information to efficiently calculate all-pairs reachability?* First we can classify nodes with the same reachability into the same zone and then we calculate reachability on a per-zone basis. Then depending on whether shortest path routing is used, different algorithms are developed to efficiently calculate the all-pairs reachability. *(2) How can we react to network or configuration changes and update the reachability quickly and efficiently?* The computation required to update the reachability should be incremental.

2. DESIGN

2.1 Background

Most of the reachability policy at the network level is enforced by access control lists (ACLs). An ACL is a list of ordered rules that collectively define a packet filtering policy. The "intersect" of two ACLs returns the set of packets that are permitted by both ACLs. Then in order to calculate the reachability of a path con-

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

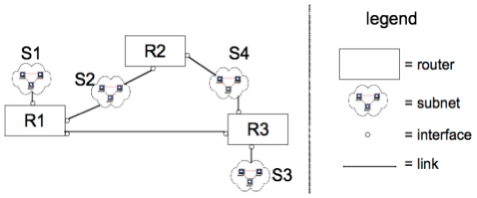

Figure 1: A simple enterprise network example

taining multiple ACLs, we just need to intersect all ACLs along the path.

A simple enterprise network can be viewed as a collection of links and gateways such as routers and firewalls. Gateways are connected via links terminating at interfaces. Each interface is attached to a subnet that is directly reachable from the interface. Each interface may be associated with two ACLs: one for filtering incoming packets and the other for filtering outgoing packets. Figure 1 shows a simple enterprise network with three gateway routers and four subnets. If a gateway is a router, then multiple routing processes will run on it to collectively calculate its forwarding states. A monitoring agent is run on each gateway to collect forwarding states and ACLs for the central coordinator.

2.2 Calculating All-Pairs Reachability

The central coordinator is responsible for calculating the all-pairs instantaneous reachability using the collected information. It will first classify nodes with the same reachability into the same zone. Initially each subnet is treated as a zone, and all IP addresses not belonging to the enterprise form a special Internet zone. Then the network forwarding states are examined to see whether the IP addresses in each zone always use the same forwarding entry. If they do not, the zone needs to be split into multiple zones accordingly. Similarly all the rules in ACLs are examined to see whether the IP addresses of the same zone always have the same filtering policy. If they do not, the zone needs to be split accordingly. Finally the IP address space is divided into N zones.

The naive way to calculate the all-pairs reachability is to directly calculate reachability by intersecting ACLs along all the N^2 paths. If we assume the average path length is $O(log(N))$, then the complexity of this naive algorithm is $O(N^2 \times log(N))$ with respect to the number of intersect operations. However, an enterprise network is usually a single domain network, where routing protocols (e.g., RIP, OSPF and IS-IS) are typically based on finding shortest paths with respect to configured link weights. One useful property of shortest path routing is that if the shortest path from node A to node C (denoted as SP(A,C)) goes through node B, then SP(A,B) and SP(B,C) must be the sub-paths of SP(A,C). Given this observation, we can calculate the all-pairs reachability more efficiently as follows: **Step 1**: For each pair of source and destination zones (SD pair), calculate the hop-by-hop routing path. **Step 2**: Sort all the N^2 SD pairs in ascent order according to the hop counts of the calculated paths. **Step 3**: Then start calculating the reachability of all SD pairs according to the sorted order. According to the property of shortest path routing, when we need to calculate reachability of a longer path, all sub-paths of the longer path must have already been calculated, so only one more intersect is needed to calculate the reachability for the longer path. Thus the total number of intersect operations required to calculate reachability for N^2 pairs is reduced to N^2, that is, one intersect operation for one pair on average. We can prove that N^2 is the minimum number of intersect operations for calculating all-pairs reachability.

However, shortest path is not always used for intra-domain routing. For example, Cisco's EIGRP presents a more complex intra-domain routing model than shortest path and some networks even use BGP for intra-domain routing [3]. In order to handle the non-shortest-path routing, the following algorithm is developed: **Step 1**: For each SD pair, we calculate its hop-by-hop routing path. All the SD pairs and their corresponding paths are then inserted into a table. **Step 2**: For each SD path, each of its sub-paths is checked to see whether the sub-path is already in the table. If it is not, then we insert the sub-path into the table. **Step 3**: Sort all the paths in the table in ascent order according to their hop counts. **Step 4**: Calculate reachability for all paths in the table according to the sorted order. If M paths are in the table finally, then M intersect operations are needed to calculate the all-pairs reachability.

2.3 Handling Network Dynamics

Networks are constantly changing, so the system must react to changes quickly and update the reachability efficiently. For example, *(1) If an ACL is updated:* basically only one link L that uses that ACL is affected. The central coordinator can easily find all paths that contain L, then it can start updating reachability from the affected path with the smallest hop count. If M paths contain link L, then M intersect operations are needed to finish the update. *(2) If routing is changed:* the central coordinator must first determine the new paths for all affected SD pairs (inserting sub-paths of the new paths if shortest path is not used for routing) and then it should start calculating reachability of all new paths from the one with smallest hop count. If M paths are affected, then M intersect operations are needed.

2.4 Open Issues

We are currently working on extending our model to incorporate other middleboxes such as deep packet inspection (DPI), application layer firewall, NATs, load-balancer and traffic shaper, which can also affect the network reachability.

3. REFERENCES

[1] Eric Gregory Wen Wen Wong; Validating Network Security Policies Via Static Analysis of Router ACL Configuration. Master Thesis of Navel Postgraduate School; Dec. 2006.

[2] Ethan Katz-Bassett, Harsha Madhyastha, John John, Arvind Krishnamurthy, David Wetherall, and Thomas Anderson. Studying Black Holes in the Internet with Hubble. In *USENIX NSDI*, 2008.

[3] David Maltz, Geoffrey Xie, Jibin Zhan, Hui Zhang, Gisli Hjalmtysson, and Albert Greenberg. Routing Design in Operational Networks: A Look from the Inside. In *ACM SIGCOMM*, 2004.

[4] Alain Mayer, Avishai Wool, and Elisha Ziskind. Fang: A Firewall Analysis Engine. In *IEEE Symposium on Security and Privacy*, 2000.

[5] A. Shaikh and A. Greenberg. OSPF Monitoring: Architecture, Design and Deployment Experience. In *USENIX NSDI*, 2004.

[6] J. Wu, M. Mao, J. Rexford, and J. Wang. Finding a needle in a haystack: Pinpointing significant BGP routing changes in an IP network. In *USENIX NSDI*, 2005.

[7] Geoffrey Xie, Jibin Zhan, David Maltz, Hui Zhang, Albert Greenberg, Gisli Hjalmtysson, and Jennifer Rexford. On Static Reachability Analysis of IP Networks. In *IEEE INFOCOM*, 2005.