Chapter XV

Edge-to-Edge Network Monitoring to Detect Service Violations and DoS Attacks

Ahsan Habib
Siemens TTB Center, Berkeley, USA

ABSTRACT

This chapter develops a distributed monitoring scheme that uses edge-to-edge measurements to identify congested links and capture the misbehaving flows that violate service-level-agreements and inject excessive traffic that leads into denial of service (DoS) attacks. The challenge of this problem is to develop low overhead schemes that do not involve core routers in any measurement to achieve scalability. The main contribution of this work is overlay-based network monitoring schemes for efficient and scalable network monitoring. This monitoring scheme uses edge-to-edge measurements of delay, loss, and throughput to infer the internal characteristics of a network domain. The analytical and experimental results show that a network domain can be monitored with $O(n)$ probes, where $n$ is the number of edge routers. Upon detection of an attack, the routers regulate misbehaving flows to stop it. We introduce a new way to measure communication and computation overhead among monitoring schemes. This comparative study shows that core-assisted network monitoring has higher communication and computation overhead comparing to edge-to-edge network monitoring scheme.

INTRODUCTION

The aim of a Denial of Service (DoS) attack is to consume the resources of a victim or the resources on the way to communicate with a victim. By wasting the victim’s resources, an attacker disallows it from serving legitimate customers. A victim can be a host, server, router, or any computing entity connected to the network. In addition to DoS attacks, the quality of service (QoS) enabled networks are vulnerable to service level agreement (SLA) violations—namely, the QoS attacks. An attacker in this environment is a user who tries to get more resources (i.e., a better service class) than what s/he has signed (paid) for. Legitimate customers’ traffic may experience degraded QoS as a result of the illegally injected traffic. Taken to an extreme, that excess traffic may result in a denial of service attack. This creates a need for developing an effective defense mechanism that automates the detection and reaction to SLA violations.

Monitoring of a network domain can ensure the proper operation of a network by detecting possible service violations and attacks. Monitoring network activity is required to maintain confidence in the security and QoS of networks, from both the user (ensuring the service level paid for is indeed obtained)
and provider (ensuring no unusual activity or attacks take place) perspectives. Continuous monitoring of a network domain poses several challenges. First, routers of a network domain need to be polled periodically to collect statistics about monitoring parameters such as delay, loss, and throughput.

Second, this huge amount of data has to be mined to obtain useful monitoring information. Polling increases the overhead for high speed core routers, and restricts the monitoring process from scaling to a large number of flows. To achieve scalability, polling and measurements that involve core routers should be avoided.

To detect attacks and service violations, we propose a low overhead monitoring scheme that does not involve core routers for any kind of measurements. Our assumption is that if a network domain is properly provisioned and no user is misbehaving, the flows traversing through the domain should not experience high delay or high loss. An excessive traffic due to attacks changes the internal characteristics of a network domain. This change of internal characteristics is a key point to monitor a network domain. We employ agents on the edge routers of a network domain to efficiently measure SLA parameters such as packet delays, loss, and throughput. The delay is an edge-to-edge latency measurement; packet loss is the ratio of total number of packets dropped from a flow to the total packets of the same flow entered into the domain; and throughput is the total bandwidth consumed by a flow inside a domain. A flow can be a micro flow with five tuples (two addresses, two ports, and protocol) or an aggregate one that is combined with several micro flows. Delay and loss are important parameters to monitor a network domain because these parameters mostly reflect the QoS of user applications. High delay and loss can be used as an indication of service violations. Although, jitter (delay variation) is another important SLA parameter, it is flow-specific, and therefore is not suitable to use in network monitoring.

We develop an overlay-based monitoring scheme that forms an overlay network using all edge routers on top of the physical network. The probing identifies the congested links due to high losses are identified using edge-to-edge loss measurements. To identify the congested links, all edge routers probe their neighbors in clockwise and counter-clockwise direction. This method requires only \( O(n) \) probing, where \( n \) is the number of edge routers. Through extensive analysis, both analytical and experimental, we show that this identifies the congested links to a close approxima-

**BACKGROUND**

DoS attacks are a serious threat for the Internet. The San Diego Supercomputer Center reported 12,805 DoS attacks over a three-week period (Moore, Voelker, & Savage, 2001). The approaches for dealing with DoS attacks can be divided into two main categories: detection and prevention approaches. The prevention approaches (Ferguson & Senie, 2000; Jin, Wang, & Shin, 2003; Park & Lee, 2001) try to thwart attacks before they harm the system. Filtering is the main strategy used in the prevention approaches that identify and drop attack packets before entering a network. Filtering can drastically reduce the DoS attack if a substantial number of domains use it, which hard to deploy. If there are some unchecked points, it is possible to launch DoS attacks from those points. The detection approaches, on the other hand, capitalize on the fact that appropriately punishing wrong doers (attackers) will deter them from re-attacking again, and will scare others to do similar acts. The detection process has two phases: detecting the attack and identifying the attacker. To identify an attacker, several traceback methods (Savage, Wetherall, Karlin, & Anderson, 2001; Snoeren, Partridge, Sanchez, Strayer, Jones, &