Volume 2, No. 1, Jan-Feb 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Efficient Detection Technique for Optimization of Network Traffic

Rajesh Kumar Ahirwar*
Department of C.S.E.
S.A.T.I., Vidisha
M.P., India
rajeshkumar_916@rediffmail.com

Rakesh Kumar Vishwakarma and Sachin Sohra
Department of C.S.E.
SVITS, Indore, M.P., India
rakesh26_kumar@yahoo.co.in
sachinsohra@gmail.com

Abstract: Network operators often need to deal with events that compromise their networks. One approach to find these events is to monitor the aggregate traffic in one or several network links and then look for significant deviations from some statistical model of normal behavior. This problem, known as traffic anomaly diagnosis, involves two steps: anomaly detection and root cause analysis. Anomaly detection methods have to define first what constitutes normal traffic behavior. Given the large variability in Internet traffic behavior, current techniques learn their parametric models from traces that are assumed to contain no anomalies. Besides the computational overhead of periodically re-training the model, real traces are never guaranteed to be anomaly-free; anomalies in the training data can contaminate the detector's definition of normal traffic behavior. Another problem with current anomaly detectors is that, by aggregating traffic before detection, they lose information about which specific flows cause the anomaly. Root cause analysis is the process of recovering this information, by going back to the original traffic traces looking for events that could explain the alarm. Currently, there are few automated techniques that can help with root cause analysis; operators often rely on ad-hoc manual procedures, which are both time-consuming and error-prone. In a large network with hundreds of links, the number of events that can trigger alarms may easily overload the Network Operations Center, making anomaly detection tools useless. In this thesis we design an anomaly diagnosis system (i.e., detection and root cause analysis) that exposes a broad range of anomalies and automatically explains their causes. We design an anomaly detection method that uses a non-parametric model of normal traffic behavior, and thus is simple to compute and immune to data contamination. It also makes it easier to identify the flows responsible for an anomaly. Second, we propose a technique that automates the root cause analysis step by identifying the anomalous traffic and classifying it according to the type of root cause vent. Our results can correctly diagnose anomalies caused by a variety of events and also expose a different class of traffic anomalies when compared to previously proposed detection methods.

Keywords- Wireless Network, Multipath routing, Anomaly Detection.

I. INTRODUCTION

Network operators typically monitor traffic in order to find and correct problems that could affect Service-Level Agreements (SLAs) with their customers, or compromise their traffic engineering, network dimensioning, and security policies. These problems include faulty or misconfigured routers, as well as unexpected traffic such as flash crowds and attacks. A number of techniques have been proposed to automate this analysis. In one class of approaches, known as anomaly diagnosis, the operator looks for unusual changes in traffic behavior and subsequently tracks down the causes of these changes.

A. Anomaly Detection

The first step of anomaly diagnosis is called anomaly detection and its goal is to monitor traffic and flag an alarm whenever some sort of abnormal change happens. Several techniques have been proposed to address the basic problem .In these techniques, link or network traffic is aggregated into one more time series, and e.g., packet counts in fixed-sized time bins. Next, the time series is compared to a pre-selected model of normal traffic behavior and an anomaly is flagged whenever the observed traffic deviates from the model. Given the large variability in Internet traffic behavior (e.g., different transport layers, applications, and end users) all practical models of normal traffic are statistical models [1].

B. Root Cause Analysis

After an anomaly detector flags an alarm, network operators need to investigate the traffic bin where the anomaly was flagged to extract as much information as possible about its cause, including:

→ The amount of traffic (i.e., packets, bytes) involved in the anomaly;

→Features of the anomalous traffic, e.g., IP address, TCP/UDP ports;

→ The type of root cause event (e.g., link failure, attack, flash crowd).

Typically, this information is obtained manually by inspecting traffic traces, router logs, and other sources of data.

This investigation is known as root cause analysis and it is important because its output determines all decisions on whether and how the root cause event must be dealt with [2] and [3].

C. Short-Timescale Uncorrelated-Traffic Equilibrium

A Short-Timescale Uncorrelated-Traffic Equilibrium is fundamentally different from previous anomaly detectors as its model of normal traffic behavior does not need to be trained from historical data. Rather, we use a relatively simple, but surprisingly effective, statistical test for inferring strong correlations among flows on a single link. This test is based on a mathematical model of a type of equilibrium which we study in detail. It finds anomalies caused by strongly correlated flow changes, i.e., events where several flows simultaneously increase or decrease their volume,

© 2010, IJARCS All Rights Reserved

even when these flows do not share common 5-tuple features such as IP addresses, ports, and protocol number. It eliminates the need for training because it uses a static definition of normal behavior, based on a flow independence assumption which is reasonable in highly aggregated backbone links. The main drawback of our approach is that if, somehow in future, the nature of Internet traffic changes and all flows become strongly correlated, our method would not be able to adapt to such a change. Traditional anomaly detection methods based on parametric models would be able to adapt simply by learning the new definition of normal behavior. We show that many types of events (e.g., scanning and DDoS attacks, link outages, routing shifts) generate strongly correlated flows, and that our detector is capable of accurately finding these events. We also show that if an anomaly is aggregated into a few large flows, then it is not capable of detecting it. We use this observation to develop a heuristic that helps us identify the set of flows causing an anomaly. This heuristic works by aggregating flows according to features like IP addresses and ports, and checking if it still detects an anomaly after aggregation. When an aggregation level makes the anomaly disappear, we know the anomalous flows can be found by looking at only a few large flows, which is easily done through visual inspection.

A Short-Timescale Uncorrelated-Traffic Equilibrium has the following main characteristics:

(a) Low complexity - It can decide if a time bin is anomalous by looking only at that time bin and the previous one.
(b) Only one threshold - Our detector has a single threshold parameter that directly controls its false positive rate (under certain statistical assumptions).
(c) Anomalous flow identification - By isolating the anomalous flows, we can analyze their characteristics and understand the anomaly's root cause event.

II. BACKGROUND

A. Network Event Detection

Several methods have been developed to detect events that network operators care about a high-level taxonomy for the techniques. Event detection techniques can be broadly divided in two categories: anomaly based and signature-based detectors. Anomaly-based techniques involve statistical models of normal behavior, while techniques based on signatures focus on matching known patterns of unwanted behavior. We separate the anomaly-based techniques between those that analyze traffic data (i.e., packet or flow traces, SNMP traces with packet counts) from those that look at control data (e.g., routing messages, DNS queries). We further divide the techniques based on traffic data between those that analyze all traffic data regardless of the application, from those that analyze traffic from specific applications (e.g., e-mail, IRC).

Finally, we divide traffic based anomaly detectors among those that aggregate traffic into one or more time series (e.g., packet counts, entropy) and flag alarms per time bins, from methods which do not aggregate all the traffic, and instead flag alarms for individual hosts, flows or packets. Clearly, statistical techniques can be used to find unusual patterns in data from domains outside of networking, e.g., public health data and stock market fraud detection. A survey provides a comprehensive overview of anomaly detection methods in several areas. This thesis is concerned with problems related to anomaly detectors based on aggregate traffic data. Accordingly, our survey of event detection techniques goes deeper into this specific class of methods. Nevertheless, we also discuss the other techniques, identifying which parts of our problems are shared. We review techniques from each class of approaches discussed above, starting from those which are higher in the taxonomy of above figure and descending to the more specific ones [1] and [2].

B. Anomaly Detection on Control Data

Some methods look for anomalies in Domain Name System (DNS) queries and responses, including botnets, and fast flux. Previously observe query rates at a server, and look for time bins with an unusually high number of queries which could indicate bots connecting to a command-and-control machine. In fast flux, a hacker registers several IPs under the same domain name, each with a very small time-to-live. This type of technique is use to hide phishing web sites and command-and-control nodes in botnets. The perform queries for specific DNS names and records the number of IP addresses per name. A single domain with several different IPs can be an indication of fast flux. Several methods can detect routing-induced anomalies by analyzing BGP feeds. To use topology data as a baseline in order to identify abnormal BGP messages that could indicate a prefix hijacking attack. Use wavelet analysis to find spikes in the number of BGP advertisements. To combine BGP advertisements with traffic data in order to find routing events that have significant traffic impact. Most of these techniques have similar problems compared to traffic-based anomaly detection. First, when they find a time bin with an unusual amount of control data (whether it is DNS or BGP messages) they still need to identify which messages are abnormal (domain names in DNS, or prefixes in BGP). However, the amount of control data in a network is generally much smaller than the amount of traffic data, making manual root cause analysis less of a challenge than in the traffic anomaly case [4].

C. Anomaly Detection on Application-Specific Data

Some techniques analyze e-mail data to find spam and botnets. BotGraph is a botnet detection system which monitors web mail services (e.g., Hotmail and Gmail) and measures the correlation between logins from different users. They use random graph theory to build a model of normal behavior, and they look for sub graphs where an unusually high number of users share IP addresses. The E-mail Mining Toolkit (EMT) is systems that monitor e-mail messages. However, unlike BotGraph, these systems build graph models of normal user communication patterns, and identify spammers as nodes with abnormal connectivity in the graph. Binkley and Singh propose an anomaly detection method that analyzes TCP header fields and Internet Relay Chat (IRC) messages to find IRC channels where bots communicate with their botmaster. They look for scanning activity using the TCP headers and then look for IRC channels with a suspiciously high number of scanners. As in signature-based methods, these techniques detect very specific types of events (given the specific types of data they monitor) and in fine granularities (i.e., hosts, email users).

© 2010, IJARCS All Rights Reserved

These characteristics make root cause analysis trivial from the inspection of the output of these tools [5].

D. Anomaly Detection on Non-Aggregate Traffic Data

Some methods create statistical models for the behavior of individual hosts or even packets i.e., without aggregating all the traffic in a link. Threshold Random Walk (TRW) is a port scan detection mechanism that models the access pattern of benign and malicious source hosts as two independent random walks. Namely, as a source contacts different destinations, TRW computes the likelihood of its sequence of destinations under each random walk. TRW then uses a simple likelihood ratio test to decide if the source is a scanner. Time-based Access Pattern Sequential hypothesis test (TAPS) is another scan detector, which monitors the number of destination ports contacted per source host and also uses a sequential test to flag scanners. Some techniques detect TCP SYN flood attacks by monitoring the number of unfinished connections per destination host and applying the cumulative sum (CUSUM) sequential hypothesis test. BotSniffer groups sources by the destinations they access and analyzes the spatio-temporal correlation inside each group to find sets of malware infested hosts (i.e., bots) contacting the command-and control server of a botnet. Packet Header Anomaly Detection (PHAD) learns the typical ranges of packet header fields from the MAC, network, and transport layers, in order to flag packets with suspicious values in these fields. One advantage of these techniques is that once an alarm is raised, we automatically know which are the abnormal hosts and packets. On the other hand, each of these techniques is restricted to a specific type of event and maintaining per host state of the algorithms can be complex in highly aggregated links [3] and [5].

E. Anomaly Detection on Aggregate Traffic Data

As discussed in the previous, one of the main drawbacks of looking for an anomaly in aggregated traffic is that we lose information about the cause of an anomaly. For example, not all of the traffic flows in a link are anomalous. Thus, once an alarm is flagged, we need to disaggregate normal from anomalous traffic. Moreover, we still need to classify anomalies according to the events that flag them. All of the detectors described below have to deal with these problems. We identify the contributions from previous work by dividing them in three sub problems within traffic anomaly detection: (1) the strategies to aggregate traffic data into time series; (2) the definition of traffic metrics that can expose anomalies; and (3) the statistical techniques used to flag outliers in these aggregated metrics.

F. Aggregation Strategies

There are two main issues related to the way traffic data is aggregated in time series:

(a) The data reduction techniques used to reduce the overhead of traffic monitoring; and (2) the choice of converting the monitored traffic into one or multiple time series. Each of these issues can impact the performance of an anomaly detector. Data reduction techniques play an important role in anomaly detection for highly aggregated backbone links. In order to save on storage and processing overhead, flow measurements in core networks typically use packet sampling with rates between 0.1% and 1%.

G. Traffic Anomaly Root Cause Analysis

When an anomaly detector flags an alarm an operator must know what caused the alarm before reacting. We review three types of tools that can help with root cause analysis: (1) traffic analysis tools that help with visualization and finding large traffic clusters; (2) anomaly detectors that can also identify the traffic involved in an anomaly; (3) methods to classify anomalies by the type of root cause event.

H. Traffic Analysis Tools

Although root cause analysis approaches still require manual investigation and visualization tools, there are traffic analysis techniques that can help an operator to discover what is going on in the traffic when an anomaly happens. Several methods can identify high volume traffic clusters (i.e., heavy-hitters), Auto Focus being the most well-known example. The disadvantages of relying purely on tools like Auto Focus for root cause analysis are: (1) some anomalies (e.g., scans) involve low traffic volumes and (2) none of these tools is explicitly trying to correlate their output with the alarms triggered by anomaly detectors, and thus still require manual intervention from operators.

I. Anomalous Flow Identification

*M*ost approaches to identify the flows responsible for an anomaly rely on sketch-based detectors. A k-array sketch is a k X h matrix associated with k independent hash functions that map a flow into one out of h buckets. In each time bin, a sketch based detector first hashes a flow using each of the k hash functions. Each flow is added k times to the matrix, in the cells corresponding to the buckets obtained from each hash function. By taking the matrices from different time bins, one ends up with kh time series, one for each cell location. Then, statistical techniques are applied to these time series to detect which cells contain anomalous behavior. During a time bin, if one or more cells flag an alarm, the anomalous flows are usually found by taking the intersection of the sets of flows hashed into these cells. The first to propose sketches to identify the flows involved in an anomaly. They used a naive detection rule, judging a sketch cell as anomalous if its relative change since the previous time bin exceeds a fixed percentual threshold. The subspace method to exploit the spatial correlation between different cells in a sketch. For using association rule mining to help identify dominant characteristics (e.g., IPs, ports) of anomalous flows after they have been identified by the sketch-based approach described above. While these works take concrete steps towards automated root cause analysis, their application is restricted to anomalies that can be found on sketches. Previous work has shown that the way we aggregate data play an important role in anomaly detection, and it is not clear at this point if sketch-based aggregation is capable of finding all types of anomalies. Unsupervised Root Cause Analysis on the other hand, does not rely on the way that traffic is aggregated.

J. Root Cause Classification

Since different types of events may require different types of mitigation strategies, it is important to classify anomalies after they are detected. Although it is hard to characterize in advance the types of anomalies that occur in

© 2010, IJARCS All Rights Reserved

a link or network (since new types of anomalies may appear over time), some typical classes include:

(a) DoS attack - an attempt to overload a host or a link by flooding it with a very large amount of traffic originating from one or several hosts.
(b) Port scan - packets sent to several ports in a target host to probe for services listening on these ports.
(c) Network scan - packets sent to several hosts within a network to find available services or vulnerabilities on these hosts.
(d) Flash crowd - the situation when a host or a network suddenly receives a large amount of traffic.
(e) Alpha flow - a data transfer that is substantially larger than typical data transfers; also known as a heavy hitter flow.
(f) Link outage - a failure, either in physical or software components, which cause a link to stop forwarding traffic.
(g) Routing change - a change in routing tables that causes traffic to shift to or away from an observed link [1] and [4].

III. PROPOSED TECHNIQUE

We propose a traffic anomaly detection technique which addresses limitations from previous detectors. Namely, previous detectors involve complex training phases and expose mostly events with large traffic volumes. In addition, most anomaly detection techniques in the literature do not provide enough information about the types of anomalies they detect.

A traffic flow is a set of packets that share the same values for a given set of traffic features (e.g., source and destination IP addresses, source and destination ports, and protocol number). To study the evolution of a flow, time is usually divided into fixed sized intervals called bins. The volume of a flow f during bin i, denoted by $x_{f,i}$, is the number of packets or bytes in the flow during the corresponding bin. In the model, flows crossing a link of interest are generated by a discrete time marked point process, where the mark process determines both the flow's duration and its volume per time bin.

While our model allows any distribution in the arrival and mark processes (e.g., arrivals can be Poisson or not, flow sizes can be heavy tailed or not), we make the following two assumptions:

A. ***(A1)Flow independence –*** There are two well-known ways through which flow independence can be violated. First, some flows can be grouped into sessions; e.g., after a client downloads a web page from a server, it may open connections to other servers to download objects contained in the page. Second, flows can be correlated during congestion episodes, since they share the same queues in routers. This can happen if a link is saturated, since some flows need to reduce their throughput so that other flows can increase theirs. Previous works have shown that, despite these two common reasons for correlation, the dependencies across flows observed in traces from real links are normally very weak. One of the reasons for this is that most backbone links are underutilized, as they are over-provisioned by design.

B. ***(A2) Stationarity*** **-** The distributions of the flow arrival process and the mark process do not change over time. Stationarity is heavily dependent on the timescale in which we observe flows, i.e., the size of time bins. It is well known that traffic exhibits strong non-stationary behaviors over long timescales, including daily and weekly cycles, and long-term trends. However, several works have shown that, at short timescales (i.e., less than an hour), traffic can be well modeled by stationary processes. If a traffic anomaly detection technique is a valid model for normal traffic, then we can design an anomaly detector whose null hypothesis is the set of consequences.

Namely, we test with high confidence whether the volume changes of flows are i.i.d. samples of a zero-mean distribution. For this, we simply compute the confidence interval for the average volume changes across flows, and check if that confidence interval includes zero. If this condition does not hold for a given time bin, we mark that time bin as anomalous (we consider only traffic on non-saturated links, and using short-timescale bins).

We introduce Unsupervised Root Cause Analysis, a tool that automates traffic anomaly root cause analysis. Unsupervised Root Cause Analysis consists of two steps:

(a) Identifying the flows responsible for an anomaly,
(b) Classifying the anomaly according to the type of root cause event.

The inputs that the inputs that Unsupervised Root Cause Analysis requires from a traffic anomaly detection technique and the specific datasets used in our evaluation requires from traffic anomaly detection technique and the specific datasets used in our evaluation.

First we specify the interface between Unsupervised Root Cause Analysis and the anomaly detector, while also defining notation that we use in this chapter. We then describe flow features that are used by our algorithms, and how we extract these features from traffic traces. Finally, we describe the anomaly datasets used in our experimental evaluation; we re-use one of the traces from the previous chapter and we introduce five new ones.

Unsupervised Root Cause Analysis uses flow features to identify and classify the anomalous traffic. Specifically, we consider the following features: (1) source and destination IP addresses; (2) source and destination ports; (3) input and output router interfaces; (4) previous-hop and next-hop AS numbers; and (5) source and destination AS numbers. Note that, while features (1) and (2) describe a flow's end hosts, features (3), (4), and (5) describe the network path taken by the flow.

After identifying the anomalous flows, we have to infer the event that caused it. Our approach is to look for patterns in the set of anomalous flows. We first propose a graph-based representation of flows, which allow us to inspect and label new types of anomalies when they are first found. Then we show how we can compute certain metrics from the set of anomalous flows and use these metrics in a hierarchical clustering algorithm to automatically recognize similar anomalies.

IV. RESULTS

We compare our identification algorithm's output to the flows in our ground truth. For each anomaly, we measure

© 2010, IJARCS All Rights Reserved

(1) the fraction of ground truth flows (or packets) that are missed by Unsupervised Root Cause Analysis, and (2) the fraction of traffic found by Unsupervised Root Cause Analysis which is not in the ground truth. We call these metrics missed traffic and extra traffic respectively. Figure shows CDFs of missed and extra traffic (in flows and packets) across all anomalies in trace A. Unsupervised Root Cause Analysis does not miss any flow or packet for nearly 36% of the anomalies. In addition, it misses less than 11% of the flows in the ground truth for 97% of the anomalies. Note also that Unsupervised Root Cause Analysis does not introduce extra flows to the anomalies. Among the anomalies where at least one flow is missing, 94% are caused by short measurement gaps.

We evaluate our classification algorithm as follows. Given an anomaly flagged at time t, we consider that all previous anomalies have been correctly classified with their ground truth labels. And compute the coordinates of each anomaly using the flows found in the identification step. After query the classification algorithm and compare its output to the corresponding ground truth label.

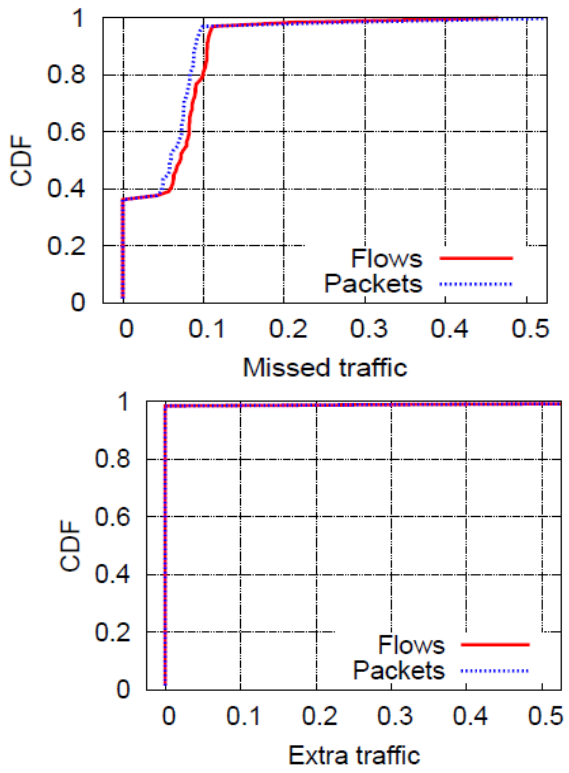

Figure.1 Identification performance for a traffic anomaly detection technique anomaly in trace.

V. CONCLUSION AND FUTURE WORK

First, we propose an anomaly detector called traffic anomaly detection technique, which provides information that facilitates root cause analysis. After characterizing the limits of what is detectable by traffic anomaly detection technique, we design a heuristic to isolate the set of strongly correlated flow from the remaining normal traffic. This heuristic relies on tracking traffic anomaly detection technique at different flow aggregation levels, and looking for aggregation levels where it is not violated, in order to identify dominant features of the anomalous flows, such as IP addresses and ports. While traffic anomaly detection technique facilitates root cause analysis, it is not a complete solution to the problem; the flow identification heuristic is semi-automated, and traffic anomaly detection technique itself does not solve the root cause classification problem. We then propose an automated root cause analysis technique, Unsupervised Root Cause Analysis, which can explain anomalies using feedback from traffic anomaly detection technique. Unsupervised Root Cause Analysis operates in two steps: (1) isolating the anomalous traffic by iteratively discarding flows that seem normal; and (2) classifying the anomaly based on its similarity to previously classified events.

The four directions for future research which derive from or extend the results in this thesis: (1) since traffic anomaly detection technique relies on a static model of normal traffic behavior and the possibilities for this model becoming invalid in the future; (2) Outline the steps needed to adapt more anomaly detectors to Unsupervised Root Cause Analysis; (3) Talk about the possibility of using other types of data (besides traffic) in Unsupervised Root Cause Analysis; (4) How an anomaly diagnosis systems may learn what types of anomalies an operator cares about.

VI. REFERENCES

[1] Nick Duffield, Francesco Lo Presti, Vern Paxson, and Don Towsley. Network loss tomography using striped unicast probes. IEEE/ACM Transactions on Networking, 14(4):697–710, 2006.

[2] Dusit Niyato and Ping Wang, “Optimization of the Mobile Router and TrafficSources in Vehicular Delay-Tolerant Network”, IEEE transactions on vehicular technology, 58(9) November 2009.

[3] Nicolas Hohn, Darryl Veitch, and Patrice Abry. Cluster processes, a natural language for network traffic. IEEE Transactions on Networking, pages 2229–2244, 2003.

[4] Italo Cunha, Fernando Silveira, Ricardo Oliveira, Renata Teixeira, and Christophe Diot. Uncovering artifacts of flow measurement tools. In Proceedings of PAM, 2009.

[5] Yanfang Deng, Hengqing Tong and Xiedong Zhang, “Dynamic Shortest Path in Stochastic Traffic Networks Based on Fluid Neural Network and Particle Swarm Optimization”, IEEE 2010 Sixth International Conference on Natural Computation.

[6] Paul Dagum and Michael Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. Artificial Intelligence, 60(1):141–153, 1993.

[7] Amogh Dhamdhere, Renata Teixeira, Constantine Dovrolis, and Christophe Diot. NetDiagnoser: troubleshooting network unreachabilities using end-toend probes and routing data. In Proceedings of CoNEXT, 2007.

[8] Nicolas Hohn, Darryl Veitch, and Patrice Abry. Inverting sampled traffic. IEEE Transactions on Signal Processing, 51(8):2229–2244, 2003.

© 2010, IJARCS All Rights Reserved