Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies

Matthew Roughan, Tim Griffin, Morley Mao, Albert Greenberg, Brian Freeman
AT&T Labs, 180 Park Av., Florham Park, NJ, 07932.
{roughan,griffin,zmao,albert}@research.att.com and bdfreeman@att.com

ABSTRACT

IP forwarding anomalies, triggered by equipment failures, implementation bugs, or configuration errors, can significantly disrupt and degrade network service. Robust and reliable detection of such anomalies is essential to rapid problem diagnosis, problem mitigation, and repair. We propose a simple, robust method that integrates routing and traffic data streams to reliably detect forwarding anomalies, and report on the evaluation of the method in a tier-1 ISP backbone. First, we transform each data stream separately, to produce informative alarm indicators. A forwarding anomaly is then signaled only if the indicators for both streams indicate anomalous behavior concurrently. The overall method is scalable, automated and self-training. We find this technique effectively identifies forwarding anomalies, while avoiding the high false alarms rate that would otherwise result if either stream were used unilaterally.

1. INTRODUCTION

Anomaly detection is useful in network management for a range of applications, from detecting security threats (e.g. denial of service attacks), to detecting vendor implementation bugs, network misconfigurations or faults. One wishes to detect times where the network is behaving abnormally, as action may then be required to correct a problem. Anomaly detection can be particularly useful in the context of reliability. Reliability is a critical objective in large IP networks, but many factors (for instance code bugs) are outside of an operator's ability to control. An alternative to preventing outages is to rapidly recover from these – for instance see the arguments presented in [1, 2]. In order to recover quickly, one must detect and localize a problem quickly.

However, while detection and alarming on real problems is important, it is equally important to keep the rate of false alarms low. A high false alarm rate results in genuine events being lost in the "snow" of false events. Statistical anomaly detection tests are run often (e.g., every five minutes), on large networks (with ten's of thousands of links), and so even a seemingly low false alarm rate may result in enough false alarms to overwhelm network operations staff. In the worst case, false alarms undermine anomaly detection, as operations staff tire of reacting to false alarms, and ignore or turn the system off entirely.

IP forwarding anomalies represent a large class of network anomalies, that relate to problems in forwarding packets to their destinations. More precisely, a *forwarding anomaly* is a period during which a significant number of packets fail to successfully exit the network at an appropriate point. Network component failures (line card, optical amplifier, or router outages, and fiber cuts) are not usually within the class of such anomalies. During such events, IP traffic is rerouted along alternate paths, resulting in at most a short transient anomaly while the routing protocols reconverge. Also, such failures are typically isolated, and easily detectable via other means, e.g. the Simple Network Management Protocol (SNMP). However, as we note below, component failures may trigger some larger network error, or occur simultaneously.

Forwarding anomalies can be the result of several problems:

- **Bugs:** Bugs in router software may cause forwarding problems that do not register via any hardware alerts, or may further be related to bugs in the instrumentation itself.

- **Misconfigurations:** The IP control plane — the distributed protocols that coordinate the building of forwarding tables throughout the network — is very complex. In such systems it is hard for an operator to understand the state of the system, and therefore the possible impacts of their actions [2]. The result is that routers may be configured in such a way that packets do not reach their destinations, even though the network may appear to be working normally. The distributed nature of the Internet means that such misconfigurations are not entirely under the control of a single operator. BGP allows network operators to misconfigure their systems in a way that may impact another network. For instance, injecting all of a providers internal routes into eBGP could result in large increase in the BGP table size across the whole Internet, potentially overloading router memory (at least on older routers), and causing router crashes across the Internet. Another example occurs if a customer that is multihomed to two large ISPs (A and B), mistakenly advertises BGP routes learned from A to B. In this case, if B does not have adequate filtering (which can be tricky to implement) it may end up routing its traffic to A across completely inadequate customer links rather than large peering links (in general customer links are preferred to peering links because they generate revenue). This type of anomaly is tricky because in principle the traffic can still make it to its destination — all the routes exist — but performance will be terrible. Note that in [3] the authors found that operator errors in the form of misconfiguration was the major cause of service affecting outages, and BGP in particular has been seen to be hard to administer, so we may expect occurrences such as this to be a major form of forwarding anomaly.

SIGMETRICS'04, NY, NY.

- **Cascading network failures:** In a cascading network failure, a simple failure (such as that of a line card) results in widespread disruption of the network. Although, the possibility of such a collapse is not anticipated in the Internet, such failures are difficult to predict, or control, and certainly have been observed in other network systems, for instance the power grid in North America in August 2003, and in the telephone network [4].

- **Latent errors:** It is possible to have latent errors: problems that are not significant until another error triggers them. A simple example might be a backup path that has been misconfigured, so that it does not work. Without careful testing such a problem might not come to light until after failure of the primary link. The failure of the primary link would be dealt with using normal procedures for detection and re-mediation, but without anyone realizing that the backup path was also failed, they might not give this task the priority it requires.

- **Exogenous factors:** It seems very unlikely that exogenous environmental effects can cause large scale failures. Networking equipment is generally held in tightly controlled environments, with redundant power supplies, and A/C. However, in rare cases, for instance the Sept 11 attack on the World Trade Center, the scale of external factors was large enough to effect a large part of the network. Although critical locations often have connections to multiple grids, and backup power supplies, very large scale power outages (such as the North-Eastern American blackout of August 2003) might also impact services in some networks if they do not have sufficient redundant power provisions for a long-term, large-scale outage. Obviously, detection of such impacts will be secondary to the event in question, but may nonethe-less be useful in order to quickly assess the scale of the impact.

- **Simultaneous failures:** Most networks plan for single component failures, for instance, by providing pairs of redundant links. Given independent failures (consideration must be given to shared risk link groups when making such assumptions [5]) simultaneous failures should be unlikely. However, occasionally failures occur simultaneously. For example in May 2000 Optus (a major provider in Australia) had simultaneous, independent failures of the two redundant links between a pair of major cities [6].

A consistent property in the above problems is that the standard methods for detecting network problems, for instance SNMP traps, syslog messages, etc., either do not detect such events, or are not likely to see the true extent of the problem. Understanding the extent of a problem quickly is important in order to prioritize the event appropriately.

Forwarding anomalies (by definition) have severe or network-wide impact, resulting in dropping large quantities of packets. For instance, on October 3rd 2002, a forwarding anomaly resulting from a router software bug caused a major tier-1 provider to loose a large volume of traffic [7], network-wide, over several hours. During this period, there were large drops in traffic on peering links (as measured by SNMP) and noticeable effects on Border Gateway Protocol (BGP) routing (as seen, for example, from customer and public viewpoints, such as Routeviews [8]). While this problem is near the extreme end of the spectrum there have been other, smaller scale forwarding problems in many ISPs (see, for example, email archives posted to the North American Network Operators' Group, NANOG, `http://www.nanog.org`). The problem in [7] motivated this work, in particular, the data sets used here — the event showed up particularly clearly in both SNMP measures of traffic, and BGP data, hence our use of these two data sources here.

We investigate an approach for reliable detection with a low false alarm rate, which integrates multiple data sources. Specifically, this paper presents an analysis of using SNMP usage data, in conjunction with BGP dynamic update data to detect and localize a class of network anomalies in a large tier-1 ISP environment. Each data source provides a different view of such anomalies. The SNMP data provides traffic volumes, while the BGP data concerns routing between autonomous systems. Both individual data sources have problems in data quality and in missing causality information that lead naturally to false alarms. On the other hand, if such problems are suitably uncorrelated in two data sources, then the false alarm rate can effectively be diminished by alarming only if both data sources indicate the anomaly concurrently. This is precisely the intuition behind our choice of data sources and our method.

First, we transform each data set individually to create useful anomaly metrics. Though SNMP usage data is relatively simple — the number of packets or bytes that traverse an interface between successive polling intervals — operational measurements for large networks can be relatively complex and noisy. We use two methods to extract the anomaly indicators from this data: a standard technique called Holt-Winters, and a second novel method based on a decomposition of the traffic, in a similar vein to Holt-Winters but exploiting a reasonable traffic model to improve results. BGP dynamic updates, on the other hand, provide a rich, high-dimensional data source, with considerable volatility. Here, to extract a useful anomaly indicator, we transform the raw data to simulate and track BGP tables at locations throughout the network. We then form the dynamic count of the number of routes in these tables satisfying a given predicate, and use a modified exponentially weighted moving average technique to signal anomalies. Last, we correlate the SNMP and BGP anomaly indicators in time to produce a combined indicator.

In this investigation we have principally looked for rapid detection and diagnosis of larger scale events — that is, those that concern more than one router or link. Single router, or single link events can be reliably detected using other mechanisms (e.g. SNMP traps), and are not a significant problem because typical provider networks are designed to re-route traffic under such failures, and to have redundant capacity sufficient for this purpose. Problems involving more than one router, may result in loads in excess of the available redundant capacity. In this case, network operators need to rapidly identify the problem as a forwarding anomaly (as opposed to, for example, a large facilities outage) and localize the problem. We note that forwarding anomalies sometimes self-repair relatively quickly, so that no remedial action is possible or necessary. However, these anomalies can still cause transient disruptions and degradations of service quality for sensitive applications, such as Voice over IP or interactive gaming. Reliable detection is still important for tuning network protocols and processes to track and reduce the occurrences of the transients.

The novel results contained here are, firstly a new algorithm for finding anomalies in SNMP traffic data with better performance than Holt-Winters; secondly, a method for extracting anomalies from BGP data; and thirdly, and most importantly, we find that that using traffic and routing data together significantly reduces the false alarm rates for forwarding anomaly detection. The reason is that during a forwarding anomaly, traffic fails to reach its correct exit point from the network. A large part of the traffic on a major ISP exits the ISP at its peering points, and so a major forwarding anomaly will be noted by a change in traffic along this edge of the network. Similarly, the routing to peers is controlled via BGP, and so large scale forwarding anomalies will appear in this data source as well. With these two sources, we focus on detection of forwarding anomalies, however, the principle could be extended to other routing data (for instance, obtained from internal routing protocols such as OSPF or IS-IS), performance measurements from active probes, or more detailed traffic measurements such as Cisco Netflow data.

1.1 Related work

Previous work on network anomaly detection has primarily focused on security tasks (detecting DoS attacks, worms, or other intrusions) and has often been signature based. We seek to find anomalies which may never have occurred previously, and so do not have a known signature. In many cases providers use very simple techniques for anomaly detection, such as fixed thresholds, but such techniques are quite limited (as described below). There has however been some more sophisticated work in the detection and analysis of network anomalies. Instances are [9, 10, 11, 12, 13, 14]. Of these, the most directly relevant to this paper is [12] which tests the use of the Holt-Winters forecasting technique for for network anomaly detection. Also of note is [13] which proposes a wavelet based method with great potential, but is most noteworthy because it has the strongest set of data used for testing these algorithms. [13] contains one of the first large scale, quantitative tests of algorithms for network anomaly detection. It may be that such testing has occurred in other cases, but network faults are considered highly sensitive data, because of their potential impact on customers, and so such data can rarely be released. However, note that the data used in [13] was drawn from a university network, which might have quite different characteristics to those of the tier-1 ISP considered in this paper.

In addition to the papers describe above, there are quite a few works on correlating alarms or various sorts via various means, for instance see [15, 16, 9, 10]. Such work is relevant here in the sense that we are performing a type of alarm correlation, although the method we use here is very simple, despite its good performance. One might apply techniques such as those suggested in these papers to improve performance in the future.

2. BACKGROUND

2.1 The Network

An IP network is made up of IP routers and IP adjacencies between those routers, within a single Autonomous System (AS) or administrative domain. It is natural to think of the network as a set of nodes and links, associated with the routers and adjacencies, as illustrated in Figure 1. We refer to nodes and links that are wholly internal to the network as *backbone* nodes and links, and refer to the others as *edge* nodes and links.

In addition, it is helpful to further classify the edge for IP networks managed by Internet Service Providers (ISPs), such as the tier-1 ISP backbone investigated here. As shown in the figure, in general the network will connect to other autonomous systems and customers via edge links. We refer to the links to customers as *access* links, and to the links to other (non-customer) autonomous systems as *peering* links. A significant fraction of the traffic in an ISP network is *inter-domain* and is exchanged between peer networks over peering links.

One can also group the nodes/routers into geographic groups: Points of Presence (PoPs), which are generally a group of routers within one building. Only some subset of PoPs will contain peering links, and it is those that we concentrate on in this paper, though the methodology is applicable in other cases with minor modifications.

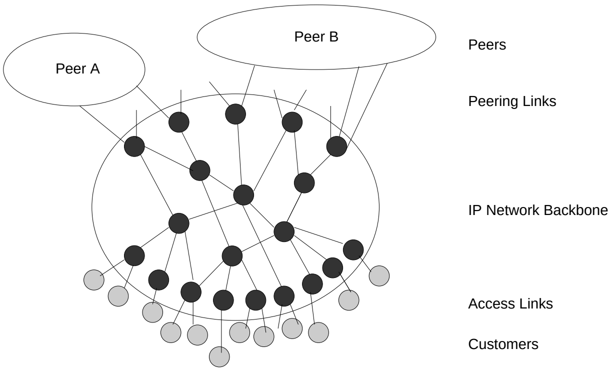


Figure 1: IP network components and terminology

2.2 SNMP

The methods we apply are critically dependent on the data on which they are built. In this paper, we analyze Simple Network Management Protocol (SNMP) traffic data extracted from an archive that includes more than 1 year's worth of data collected from a large tier-1 ISP's backbone network. SNMP is unique in that it is supported by essentially every device in an IP network. The SNMP data that is available on a device is defined in a abstract data structure known as a Management Information Base (MIB). An SNMP *poller* periodically requests the appropriate SNMP MIB data from a router (or other device). Since every router maintains a cyclic counter of the number of bytes transmitted and received on each of its interfaces, we can obtain basic traffic statistics for the entire network with little additional infrastructure.

Unfortunately, as a practical matter, SNMP data has many limitations – for instance missing data (it may be missing because SNMP uses UDP transport, or it may be lost while copying to our research archive), incorrect data (through poor router vendor implementations), and a coarse sampling interval (typically five minutes, though one could easily collect data at somewhat finer intervals, say one minute), due to the collection mechanism (polling from a small set of pollers) the sample times of the data are jittered both between successive measurements, and between measurements on different network components. A further limitation is that SNMP only provides aggregate link statistics – we cannot determine anything about the type of traffic using the link, nor its source or destination.

These limitations make algorithmic analysis such as time series analysis difficult on this data. As a simple example, consider computing the sum of traffic across the network edge. If only 1% of the data were missing (at random), then at any time, the sum (which includes many more than a hundred links), would on average be missing at least one data point. As the traffic between links varies, this would introduce apparent variations in the total traffic that are purely artifacts of the missing data.

In the network in question we have more than one year's SNMP data, gathered at five minute intervals. The data contains the MIB-II counters, which include the number of bytes in and out for each interface in the network, and this is the data we shall examine here. We first aggregate the data onto one hour intervals, using nearest neighbor interpolation of the samples. This transforms the problems of missing data and sample jitter into a small amount of noise in the measurements. We then aggregate this data to examine the total traffic along the peering edge at each Point-of-Presence (PoP) at which there are inter-peer connections.

2.3 BGP

The Border Gateway Protocol (BGP) is the only interdomain routing protocol on the Internet today. It has a very important role as it is responsible for exchanging the reachability information between the Autonomous Systems or ASes. BGP is a path vector protocol. Each route to a prefix contains the entire AS path which is a sequence or collection of ASes to reach the destination. BGP is policy-based, i.e., routing across Autonomous Systems does not use shortest path. Routing policies can override distance based metrics. There are no uniform routing policies across ASes, as they are determined by commercial relationships. BGP is incremental; thus, in steady state, no routing messages are exchanged between BGP speaking border routers across ASes. BGP updates dynamically announce and withdraw routes.

Monitoring BGP updates can reveal all the changes in the best route used by a given router and can in turn indicate routing problems in the network. Route monitors, supporting passive BGP peering sessions, can be set up to receive all the routing changes. Such route monitors archive all the update messages and regularly dump all the routes in the BGP table. The Oregon Routeviews [8] project and the RIPE NCC RIS project [17] have both set up such routing sessions with numerous ISPs all over the world to obtain the latest information of the best routes used by these networks.

Inside a service provider's backbone network, it is also very valuable to monitor the best routes used by various routers at different locations to study potential network problems. Such data along with the data from RIPE and Routeviews allow one to potentially differentiate the problems within one's network from those in neighboring networks. In our experiments, we make use of measurements from route monitoring sessions to route reflectors in all major PoPs of the backbone. The data include all the BGP updates, i.e., the routing changes, as well as daily table dumps of all the best routes used. The monitoring sessions advertise all the available routes. Given a BGP table which contains all the best routes used, one can apply the updates to obtain the best routes at any given time.

BGP updates comprise an extremely rich, high dimensional, and relatively volatile data source. The volatility arises from the fact that BGP sees routing changes over a large segment of the total Internet. The challenge is to define metrics that allow for rapid calculation and prediction of forwarding anomalies. In general, we have found metrics that count the number of routes that satisfy a given predicate to be very useful in detecting and explaining routing phenomena and various forwarding anomalies. In the particular case here, we shall focus on the distribution of routes via different egress points. A router typically has several choices of where to send traffic for most of the prefixes in the table. A sudden shift in the distribution of routes give good indication that the egress point that is less preferred may experience some network outages. In this paper, we will investigate metrics that track for each route reflector under observation, the number of routes that exit a given PoP. These metrics are easily calculated by combining BGP updates with configuration data that maps BGP next hops to BGP routers.

For each PoP we shall store a time series constructed from the number of best routes seen exiting the PoP by the local route reflector. We further aggregate this data into time bins so that we store the minimum and maximum number of routes seen during each bin. For the purpose of this study we use one hour bins, though this number is arbitrary, and can easily be reduced to bins as small as a minute with very little additional computational load.

This data can contain artifacts as a result of resetting the BGP session between the route reflector and the route monitor. These artifacts have been removed using data archived at the BGP monitor, but other artifacts may persist.

3. ANOMALY DETECTION

Anomaly detection can be thought of as an attempt to detect data that shows evidence of emerging from a different process or mechanism than typical data. There are far too many possible approaches to anomaly detection to list here. We shall take the approach of using several very simple, but reasonable detectors for individual data sets, and rely on the combination of data sets to provide good performance. One could obviously improve the quality of the detectors on individual data sets using more sophisticated algorithms, but this paper shows that such an approach is not needed, given the correct combination of data sets. Furthermore, the false alarm rate for any one data set will be intrinsically limited by that data set, and so one must use some combination of data in order to get useful performance.

An often used approach in detecting network anomalies in traffic data is to set arbitrary thresholds, based on experience, or engineering guidelines. These methods are greatly limited by the rapidly changing nature of the Internet, the inability to scale such methods to large networks where many different threshold might be appropriate, and the non-stationarity of Internet traffic (a spike in traffic would not be detected by such a threshold if it occurred during the early morning when traffic is very low).

Generalizing somewhat, while maintaining simplicity, another approach to anomaly detection is outlier detection — detecting observations that differ by a large amount from other observations. The method by which you measure the *normal* behavior of the observations is still of interest. There are a number of alternatives available, depending on the properties of the data in question. Two traditional examples are

- **EWMA:** The Exponentially Weighted Moving Average (EWMA) chart is a text-book method applied in quality processes (for instance see [18]). The method is broadly applicable to data with a stable, stationary mean, and independent observations. We will adapt this approach for use on the BGP data analyzed here. The modifications are required because BGP data does not quite satisfy the above requirements, but we can nevertheless create an effective anomaly detector with minimal modifications. Exponential smoothing is a method for computing a prediction of the value of some measured quantity, where the measurements include noise. Given the prediction, we can assess how far the next measured value is from its prediction, and thus decide if the measurement is an outlier. One assumes that the process can change (slowly) over time, and so more recent measurements are more relevant, but may be overridden by the body of preceding data. The EWMA prediction is given by

  $$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t, \tag{1}$$

  where α is a parameter that determines the weight given to the current measurement compared with previous measurements. One can similarly estimate the variance of the noise using

  $$\sigma_{t+1}^2 = \alpha e_t^2 + (1-\alpha)\sigma_t^2, \tag{2}$$

  where e_t is the one step prediction error $e_t = y_t - \hat{y}_t$. Given these estimates one can declare an observation an outlier if it lies more that δ standard deviations from the predicted value, i.e. if y_{t+1} lies outside the interval $(\hat{y}_{t+1} - \delta\sigma_{t+1}, \hat{y}_{t+1} + \delta\sigma_{t+1})$.

  An example in this vein is the TCP Retransmission Timeout computation [19], which uses exponential smoothing of round-trip times to estimate how long to wait for an ACK before deciding that a packet as been lost.

- **Holt-Winters:** The Holt-Winters method is designed for data which shows both periodic variations (both in mean, and variance), and long term trends, in addition to stochastic variations. The method, is described in more detail below where we apply it to the SNMP data. Holt-winters has been perhaps the most commonly tested algorithm in the context of Internet anomaly detection [12, 14, 13] because of the fact that while not optimal, it does not require extensive modeling assumptions and is therefore quite robust.

  The Holt-Winters (see [12, 20] for more details) method generalizes the above EWMA, by allowing the estimating seasonal, and trend components. One disadvantage is that it requires more parameter settings than EWMA, but there are typical ways for setting these (see also [12]).

In addition to Holt-Winters we apply a novel method (described below) using a model developed in the context of backbone Internet traffic. The advantage of the EWMA and Holt-Winters algorithms are that they are incremental, one-pass algorithms, with minimal storage and computation requirements. Thus they can be easily applied on-line as data is collected. The disadvantage of these methods is that they include the detected outliers in future measurements. That is, should we see a large spike in our traffic measurements, then this will be included in future predictions via equation (1), and will effect the estimated variance likewise. EWMA can be adapted to be less sensitive in this regard, which is the main modification we make to EWMA in our BGP anomaly detection. However, in the case of SNMP data we also test a model-based two-pass algorithm. This latter approach has many similarities (in its basic assumptions) to Holt-Winters, but proceeds by decomposition of the traffic into components, including a component describing anomalies, which may then be discarded before applying a second pass. This type of approach is also common in the time-series literature, with the exception that the model applied here is chosen specifically because of its better representation of traffic data. While this approach is not as appealing in the real-time context, note that the data which we are dealing with (SNMP for instance) are unlikely to be gathered more frequently than once per minute, and therefore our computational budget is not too tight. Further, the method described is not particularly onerous, having computational complexity of $O(N)$ for data sequence of length N.

In addition to the methods described here there are many alternatives. For instance, the wavelet techniques explored in [13] are particularly appealing. Other methods include ARIMA modeling, as used in [14], or Bayesian techniques, for instance see [9]. These methods might allow for improvements in false-alarm rates in themselves. However, the properties of the data itself suggest that there will always be false alarms for any detector of the required sensitivity.

3.1 SNMP

Internet traffic shows strong non-stationarity both in the mean, and the variance [12]. This non-stationarity has two major components – a long term trend, and a periodic, or seasonal component. What's more, as noted in [12] the periodic component can gradually evolve, for instance, as the number of evening daylight hours changes from summer to winter.

The approach described below is to model the traffic using a semi-standard time series model, and to then decompose the traffic directly into a non-stationary (but regular and predictable) component, a normal stochastic component, and a set of anomalies. This approach has the advantage of being simple, intuitive, and well-matched to existing traffic measurements (for backbone networks).

Figure 2 (a) shows the traffic along a backbone link over 4 weeks. The most noticeable trait of this traffic is a strong periodic component with a period of a day, and a slightly less obvious weekly cycle. On top of the periodic components there is some random variation, and when one observes Internet traffic over longer intervals, one sees long-term trends.

The typical time-series approach to modeling such series is to model the observed process as a combination of several components, in our case the following:

1. **Seasonal/Periodic Component:** S_t
2. **Long Term Trend:** T_t
3. **Normal Stochastic Component:** W_t
4. **Anomalies, modeled as impulse functions:** I_t.

This set of components lies at the base of the Holt-Winters technique as well. In that method (as is typical in time series) the components are simply added together (one can also build multiplicative models which become simple additive models in the log domain). However, for backbone Internet traffic there is a more appropriate method of combining these components which takes into account the way traffic multiplexes together. The basic assumption is that traffic increases, not because a fixed number of sources increase their volume, but rather because the number of sources increases. One can think of exceptions to this, but in the backbone so many sources are aggregated that it is a reasonable model in the majority of cases [21]. The combination method that satisfies multiplexing requirements is given by

$$x_t = m_t + \sqrt{a m_t} W_t + I_t,$$

where x_t is the traffic rate at time t, a is the *peakedness* parameter, and m_t is the regular, predictable, but variable mean, which is given by

$$m_t = S_t T_t.$$

A simple method to decompose the first three components, given the model above, is given in [22]. The starting point is a Moving Average (MA), which is simply a convolution of the time series with a low-pass filter. We use only centered rectangular windows in this work. Thus the MA of width $2n+1$ applied to time series x_t is

$$\hat{T}_t = \frac{1}{2n+1} \sum_{i=-n}^{n} x_{t+i}. \tag{3}$$

If the filter has length greater than the period P of the seasonal component (1 week) then the MA acts to remove the periodic, and stochastic variations. Therefore $n = P/2$ yields an estimate of the trend, i.e., we estimate the trend using a MA with width ∼1 week. Once estimated we can form a detrended data set by $y_t = x_t/\hat{T}_t$.

The standard method in time series analysis used to estimate the seasonal component exploits the periodicity by using a Seasonal Moving Average (SMA) where we take a MA over a series of data points separated by the period, so as to estimate the periodic component at those time points [20]. In our analysis, we perform our average over the whole data set, i.e.

$$\hat{S}_t = \frac{1}{N_t} \sum_{i=0}^{N_t - 1} y_{t+iT}, \tag{4}$$

where $t \in [0, T)$, and N_t is the largest integer such that $t + (N_t - 1)T \leq N$, where N is the length of the data. We refer to this as the Seasonal Average (SA). We may estimate the mean by $\hat{m}_t = \hat{S}_t \hat{T}_t$.

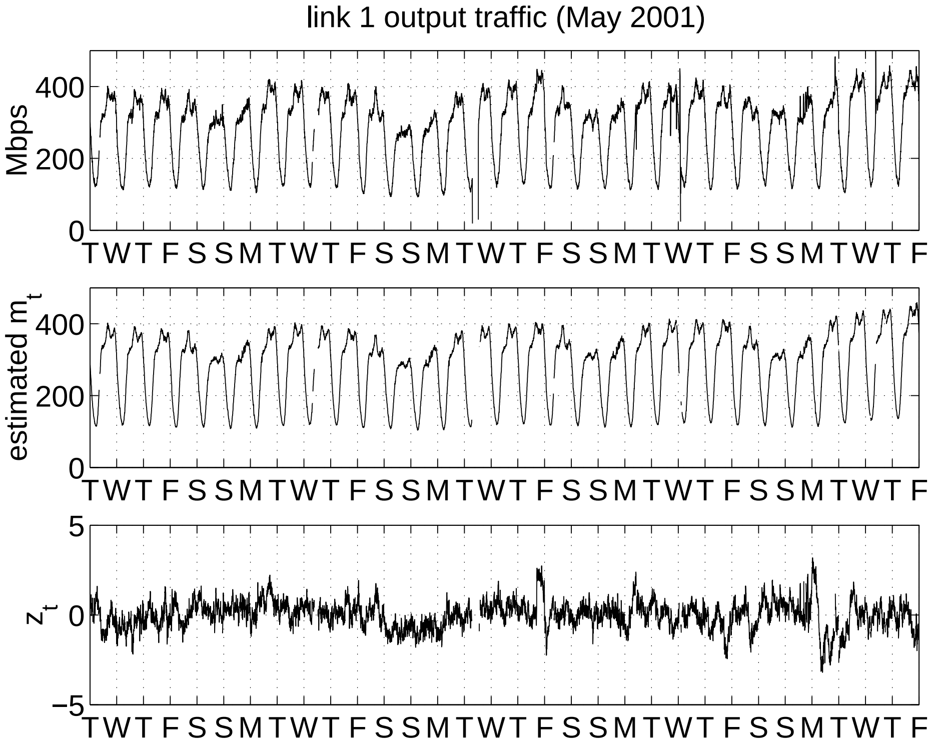


Figure 2: An example showing SNMP traffic data, decomposed into the regular component m_t and the stochastic component.

Once we understand the periodic and trend components of the traffic, the next thing to capture is the random variation around the mean. We have estimated $\hat{m}_t$ and so can estimate the stochastic components, by

$$\hat{z}_t = (x_t - \hat{m}_t)/\sqrt{\hat{m}_t}.$$

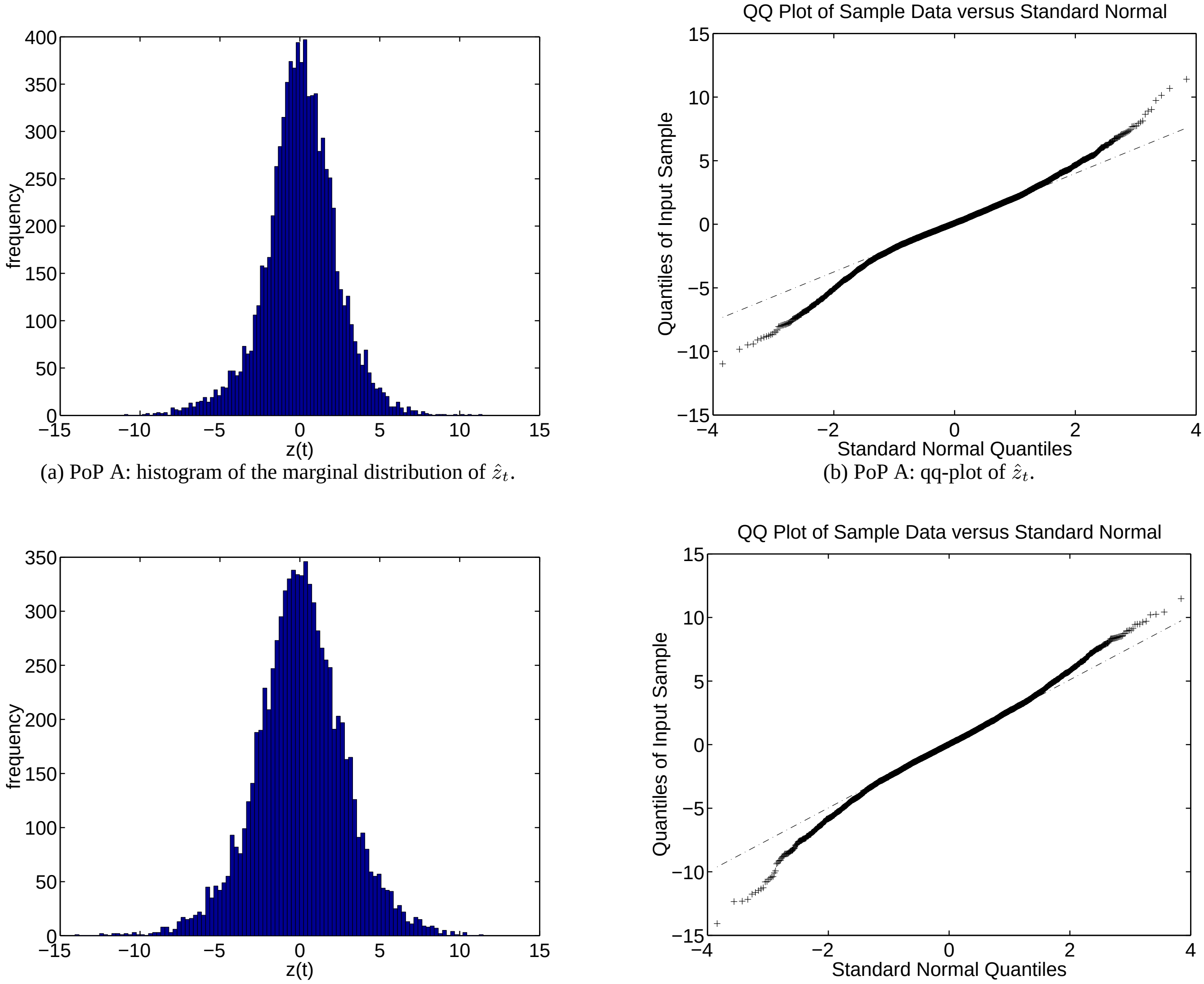
However, note that this includes both the normal stochastic component (W_t) and anomalies (I_t). We must therefore separate these. In this paper we use simple outlier detection, but one could apply a number of anomaly detection techniques to this sequence, because it is now approximately stationary. Even where we do simple outlier detection, based on the variance of $\hat{z}_t$, an advantage of this approach, as compared to Holt-Winters, is that one now has a long series of data points on which to estimate a variance, whereas Holt-Winters must estimate individual variances at each point in the period (for instance at each time-of-day), from much shorter sequences of data. Given five minute SNMP data, and a daily period, this represents a 288 fold increase in the number of data points that can be used at each time point for estimating the variance. Thus we can form a more accurate estimate of the variance.

The outlier detection applied here involves simply measuring the standard deviation σ of the marginal distribution of z_t, and choosing outliers to be those events further than $N\sigma$ away from the mean of z_t. This has a simple interpretation in the event that the marginal distribution of the stochastic process W_t has a simple known form, for instance, if W_t is Gaussian, then $\pm 1.96\sigma$ are the 95% confidence intervals of the distribution, and so one would expect that data from W_t would fall outside these bounds 5% of the time. Choosing N is therefore easy in the Gaussian case, as we can choose it so as to fix our false alarm rate at some value. In reality the marginal distribution of W_t is unknown, but the data seems to support a Gaussian model to some extent. Figure 3 shows two representative histograms of the marginal distributions of z_t for the traffic of two PoPs traffic. The figure also shows corresponding qq-plots of the data — if the data were Gaussian, all of the points would lie on the diagonal line. We can see that over a range (of roughly two standard deviations) that the data does fit this line, but that the tails of the distribution are somewhat heavier than Gaussian. We shall, in this paper, use the assumption of Gaussianity to choose the threshold level N, but note that the Gaussianity assumption is not needed, but only applied as a guide to obtain a specific false alarm rate.

Now that we have applied our outlier detection we may segregate the time interval $t \in [0, T]$ into parts which represent anomalies, and otherwise. In the case that there is an anomaly we know that the model above does not hold (by definition), and so it is possible that the measurements x_t from those times bias our estimates of S_t, T_t, and W_t. To explain more simply, a single anomalous measurement, may take a value many times larger than the typical measurement (at that time of day), and might therefore, unduly increase the expected value of the traffic at that time of day.

The standard approach to circumvent this problem is to rerun the decomposition, but this time exclude measurements from time intervals at which the algorithm had detected anomalies. Note that all of the above decomposition was designed to work even with missing values, and so it can proceed with no difficulty, unless a great deal of data does not fit the model. One may consider rerunning the decomposition in this fashion a number of times, but we do so only once here.

The final result is a decomposition into the four components described above. The decomposition gives the first two components values for all time intervals, the third component for those time in-



(a) PoP A: histogram of the marginal distribution of $\hat{z}_t$.

(b) PoP A: qq-plot of $\hat{z}_t$.

(c) PoP B: histogram of the marginal distribution of $\hat{z}_t$.

(d) PoP B: qq-plot of $\hat{z}_t$.

Figure 3: On the Gaussianity of the marginal distribution of $\hat{z}_t$ for two example datasets.

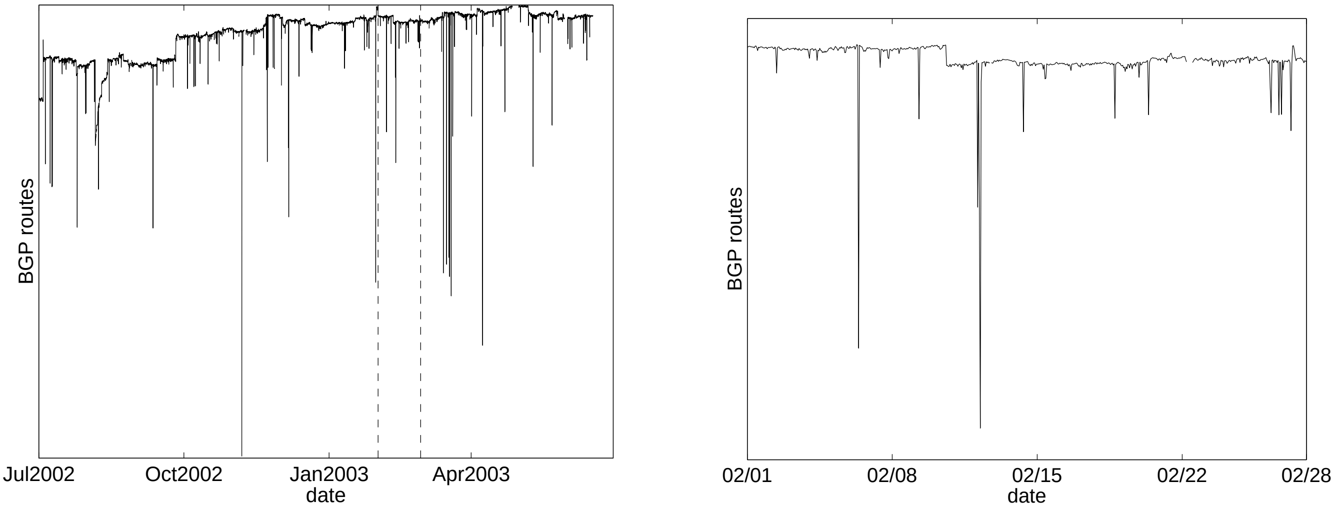
tervals at which anomalies do not occur, and the times at which anomalous measurements are observed. Figure 2 show an example of the decomposition of a set of traffic (over four weeks), into its regular component m_t (the combination of the trend and seasonal component), and the estimated stochastic component (denoted in the graph by z_t). Note that in this data set, no significant anomalies are observed, so $\hat{z}_t$ is an estimate of W_t. We will refer to this method for anomaly detection as the *decomposition technique*.

3.2 BGP

As with the SNMP data, the BGP data is non-stationary. It exhibits trend over long time intervals (typically the number of routes in the Internet has grown over time), and also dramatic changes where changes in routing policy (for instance introduction of prefix aggregation) have resulted in shifts of a large number of routes. A good example of the latter can be seen in the relatively recent absorption of Genuity's traffic into Level 3. All of the Genuity routes shifted to Level 3 over a relatively short interval of time (less than a day around May 7th 2003). BGP data is, as noted above, relatively volatile. However our metric – the number of routes matching a given predicate – shows good stationary properties over moderate time scales (days), and little periodic time-of-day, or day-of-week variation. Hence EWMA seems applicable. In the algorithms here we use as our observations the minimum number of routes in any time interval, which provides a faster method of alerting to a drop in the number of routes.

Figure 4 shows an example of the minimum number of best routes from a single PoP (for each one hour time bin), as seen by that PoP's route reflector, both over the whole year from July 2002 to June 2003, and just for the month of February. One can see a broad trend, and at least one large, sudden transition in the graph, as well as many short, but dramatic drops. However, note that over moderate periods of time (days to weeks), the number of routes appears to be well modeled by a stationary process (with low variance) with much larger anomalous points.

The relative simplicity of the observed data suggests EWMA as a good method to apply here, however, this does not have good behavior in the presence of a sudden level shift — it would signal a fault for an extended period of time, much longer than might be justified by the shift itself. Further, in this paper we assume that



(a) Over one year.

(b) Zoom into February (shown in plot (a) by dashed vertical lines.

Figure 4: The number BGP routes from a major PoP.

the models used must self-train without the benefit of clean training data. To be more precise, the parameters of the model used for outlier detection must be estimated from the data, without the benefit of knowing a-priori which data points are outliers. However, when a fault occurs this results in a large transient outlier, which would disturb future predictions. Hence, we must also incorporate a method to exclude such outliers from use in training the model.

The method used here is to estimate the mean of the normal number of BGP routes using the EWMA. However, if the value falls outside our the threshold around the predicted value, indicating a failure, we shall not use it to update the EWMA. If we were to discount such outliers entirely, we would loose track of the process at any large step. We would mark the first point of the step as an outlier, and so discount it, and therefore the estimated mean would remain unchanged, causing us to mark the next and all future points as outliers. To avoid this pitfall, we allow a maximum gap in the data – if outliers persist for longer than this gap then we reset the value of the EWMA.

To be precise about the exact algorithm used we provide MATLAB code implementing the simple algorithm below.

```
gamma = 0.05;
thresh = 0.2;
max_gap = 30;
gap = 0;

exp_av(1) = data(1);
anomaly(1) = 0;
for i=2:length(data)
  if (data(i) > exp_av(i-1)*(1-thresh) & ...
      data(i) < exp_av(i-1)*(1+thresh))
     anomaly(i) = 0;
     exp_av(i) =  gamma*data(i) + ...
                (1-gamma)*exp_av(i-1);
     gap = 0;
  else
     anomaly(i) = 1;
     exp_av(i) =  exp_av(i-1);
     gap = gap + 1;
     if (gap >= max_gap)
       gap = 0;
       exp_av(i) = data(i);
     end
  end
end
```

Note that `exp_av(i)` represents the EWMA prediction at time interval i (the order of computations is slightly different from equation (1) above to make the test before the update algorithm, so we can exclude outliers from the updates), the threshold used is 20%, and the anomalies are signified by setting `anomaly(i)` to one at the relevant time interval. The maximum gap allowed was chosen by examining the length of a large number of anomalous events, and determining a value large enough to encompass all such events. Note that to adapt this algorithm to shorter time interval data (from the one hour data currently considered) one need only multiply the `gamma` and `max_gap` parameters by the appropriate factors. We will denote this algorithm *EWMA-2* below.

3.3 Correlating results

To correlate the results from the anomaly detectors formed from the SNMP and BGP data streams, we shall use arguably the simplest approach possible: for any one PoP if both BGP `AND` SNMP anomaly indicators report anomalies, we shall generate an alarm. This simple correlation allows us to focus on the hypothesis that the power of the approach derives principally from combining indicators whose false alarms will be uncorrelated, as the false alarms relate to independent glitches in data collection or natural variations of the individual data streams. The vast literature on pattern recognition provides more sophisticated correlation approaches, which remain open for improving the results – for instance the authors of [9] propose the use of Bayesian inference networks to combine multiple features. However, note that the following tests demonstrate that the very simple approach used here produces results of practical quality, though they might be improved.

4. RESULTS

To evaluate the anomaly detection method, we use three sets of data:

- A long set of SNMP/BGP data covering eight large PoPs in the tier-1 ISP backbone IP network spanning an interval of more than one year.

- A second shorter set of data from February to May 2003, inclusive, consisting of detailed *fault tickets* from the network. Fault, or trouble tickets are the representation of a network problem in the database system used by operators to track a problem from detection to re-mediation. These tickets contain data on the start, and end time of a fault, and its root cause. However, this data is primarily text, entered by operators as they discover, and analyse a problem. Hence the times are often not precise, and the records must be processed laboriously by hand to obtain the relevant data. For instance, see [3] for more discussion of this type of data source.

- A list of the events that were considered by the IP operations group to be of the type that we wished to detect (using the methods above). This list is very short, as such events are rare, but they are of such scale that it is still important to detect them, even if they occur less than once a year.

The results given here are broken into two sections. First we shall consider two illustrative examples to see how the algorithm works. We shall then undertake a more systematic examination of the algorithm's performance.

4.1 Examples

We start by considering two illustrative examples of the algorithm in practice. The example shown in Figure 5 covers the failure of a major network peer. During this failure, the peer dropped traffic along its peering links in a number of locations. In the Figure, we focus on a PoP where nearly half of the traffic (and corresponding routes) arose from that peer, and so the failure stands out clearly. However, the failure was also detected in a number of other locations.

While this example would not lead to a particular remedial action on the part of ISP, some operational actions may be called for to mitigate the impact of the outage. Moreover, the example is highly illustrative for two reasons. Firstly, it shows the close match between the estimated model m_t and the real traffic for SNMP (except during the anomaly), and the simple and effective nature of the BGP time series used for anomaly detection. Secondly, it clearly shows the large deviations from the model during the outage.

A third feature to note in the Figure are the presence of two false alarms in the BGP data – dips that do not correspond to noticeable traffic changes.

The second example, in Figure 6, shows an example of what we see when a router with many peering links has an outage (in this case a planned outage). We can see early in the morning, local time (the date ticks occur at 00:00 GMT), a dip in both BGP, and SNMP. The plot should reinforce those points above, with the addition that we see a false alarm (near 00:00 5/15)) in the SNMP due to missing data. Note that such a large drop in traffic *should* be detected by any anomaly detection algorithm one might build. However, it is an artifact in the SNMP data. The presence of such artifacts is a key reason why correlation between data sources is needed. The data sources are generated by completely different processes, and so artifacts are unlikely to occur in two data sets at the same time.

4.2 Statistical analysis

In this section we shall conduct a more systematic examination of the data. The first task is to determine the sensitivity, or detection probability – the probability that an anomaly is detected. For this we use the short list of known events, and the SNMP/BGP data set over the enclosing time interval. The algorithms above (Holt-Winters or the Decomposition technique applied to SNMP data, and EWMA-2 applied to the BGP data) performed perfectly on this list, and further, indicated the location of the problem (where relevant). Of course, with a small set of event data it is hard to determine precisely the detection probability, but it is reasonable to assume that high detection probabilities are easily achievable because the events we wish to detect are relatively large scale, with a relatively large impact on measurements.

The second, and more difficult task is to determine the selectivity, or false alarm rate. To assess the false alarm rate we take each detected event from the period of February to May 2003, and examine the detailed fault tickets to determine it's root cause. To give one an indication of how often major events occur, none from our list occur in this time interval. The results of this investigation are shown in Table 1 — note we cannot report the actual number of events because these numbers are considered sensitive data, however the numbers are quite small.

root cause	percentage of events	Holt-Winters events
edge node/link outage	67%	55%
simultaneous outages	11%	18%
unknown cause	22%	18%
error	0%	9%

Table 1: Root cause analysis of the detected events. The table shows the proportion of different root causes for the decomposition and Holt-Winters approaches.

The majority of events detected by either technique turn out to be caused by simple router or link outages on the peering edge of the network, such as shown in Figure 6 (both methods detect exactly the same set of such events). For instance, if a router with many peering linkages is taken out of service, then the routes that would normally exit the network at this point will be rerouted to alternative points, resulting in changes in the SNMP traffic, and BGP routes. The majority of such events are part of planned maintenance (for instance for software updates), and occur during the maintenance window early in the morning when not much traffic will be effected, if there is any impact at all. By correlation with operational work flow information, such false alarms may be filtered out. In addition, the vast majority of such events occur in isolation, and are easily detected through standard mechanisms such as SNMP traps/polls of the routers, and so, they are not actually of great interest here. Note also that we do not detect all such events, because in many cases the actual change in traffic volumes, and/or BGP routes may not be that significant (for instance the failure of a peering link to a small peer).

In the process of analyzing the above events we found a small percentage which were the result of simultaneous link/node outages. Such simultaneous outages might be more serious than single outages because the network is designed to deal with single component failures. Some scenarios involving multiple points failing simultaneously might result in excess loads being placed in parts of the network as traffic is rerouted. The cause of the simultaneous events is unknown, but could be an unplanned outage occurring concurrently with a planned outage. Such outages are worth detecting, so that operations staff can defer planned operations to a time when the network has no unplanned outages. Note that once again, both Holt-Winters and the decomposition techniques came up with the same list of simultaneous outages.

Finally, we also found a small proportion of events that had no

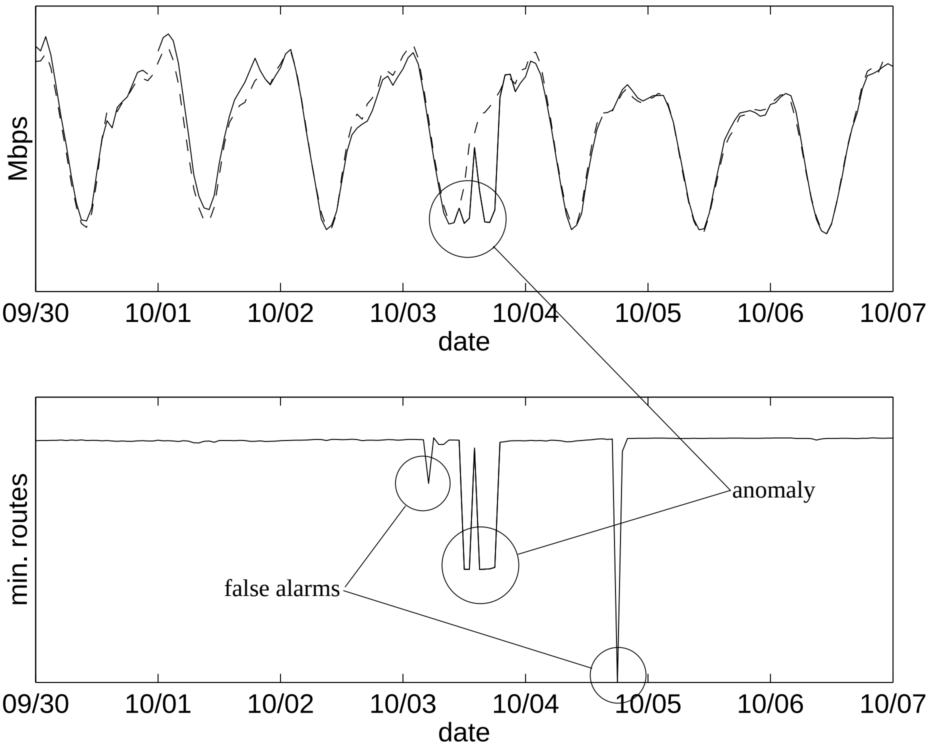


Figure 5: An example showing SNMP traffic data, and BGP routes. Plot (a) shows the SNMP data, while plot (b) shows the BGP data. In plot (a) there are two curves, the actual data (solid line) and the estimated m_t (dashed line).

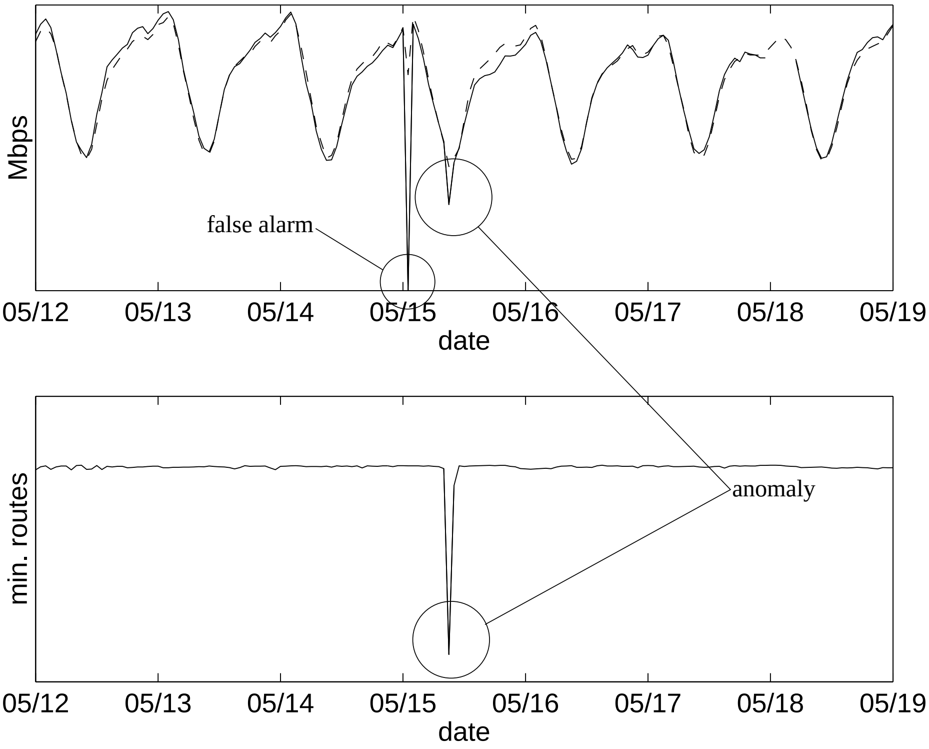


Figure 6: An second example showing SNMP traffic data, and BGP routes. Plot (a) shows the SNMP data, while plot (b) shows the BGP data. In plot (a) there are two curves, the actual data (solid line) and the estimated m_t (dashed line).

apparent cause in the trouble ticket data. The cause of these events remains a mystery, but they appear to be genuine results (they are not the result of missing, or obviously erroneous data). Such events could have two causes: problems in the network that are not detected via other means, or with more likelihood network problems on adjacent peers. In either case the events are interesting to network operations. In the former, it would allow better network management, in the latter case it gives us a window onto the behavior of the greater Internet. Holt-Winters detected only half of these events, and in addition detected a small number of events that seem to be artifacts of the detection mechanism, and are not readily apparent in the data at all.

Given the above (the fact that the majority of events correspond to causes that we don't need to detect here), the false alarm probability drawn from this table seems quite large (64-67%), but note that the frequency or rate of events is very low even so. A false alarm rate of at most a couple of events detected per month is quite acceptable.

Now compare these result with the false alarm rates for the individual data sources. Table 2 shows these results, based on the more than one years worth of data we have available. Because we don't have fault tickets for this entire period, we infer a false alarm from the fact that both data sources do not signal alarms. This is clearly only an upper bound on the false alarm rate, as some of these alarms may fall into the categories above (i.e., single edge node failures), but the vast majority of such events are false alarms (tested on a sizable sample of these events). We compute these false alarm rates based both on the proportion that happen per PoP per hour over all the eight PoPs considered here. We examine these false alarm rates for the two SNMP detection algorithms, and for the EWMA-2 algorithm on the BGP data. Note that the latter algorithm has the lowest false alarm rate. Holt-Winters has a slightly higher false-alarm rate than the decomposition technique.

data set	algorithm	false alarm rate	expected false alarms per day
SNMP	Decomposition	3.4%	78
SNMP	Holt-Winters	4.3%	99
BGP	EWMA-2	0.5%	12

Table 2: False alarm rates for individual data sources.

Note that the rates in Table 2 are different from the percentages in Table 1. In the previous table we reported the percentage of reported events of each type. In Table 2 we report the proportion of *tests* that return a false alarm. That is, around three in one hundred decomposition test return a false alarm. Assuming only 8 PoPs, and five minutes measurements (288 per day) one would expect between 12 and 99 false alarms per day (see Table 2, column 4). These are supposed to major events, and given high priority treatment, and the amount of work required to determine the cause of each alarm is large. Such a large number of false alarms is unreasonably high, and would result in the anomaly detection being switched off, or ignored. It would be even worse for a larger scale network, or tests performed at a finer level of detail. Relative to the number of tests performed, the percentages in Table 1 would be *very* small, producing fewer than an event per week. This demonstrates this paper's main point — we can get a very large false alarm reduction using the combination of data sources. The reduction is greater than a factor of one hundred times.

4.3 Discussion

The above results suggest that the decomposition technique is a little better than Holt-Winters, in this context. Both have perfect sensitivity, however, the false alarm rate for the Holt-Winters technique is slightly higher when we use the SNMP data alone. Furthermore, although, in Table 1, they appear to have the same selectivity, notice that in the Holt-Winters test some of the nominal false alarms are generated by errors, whereas all of those in the decomposition technique are generated by real data events, even though some are not interesting. Further note that the decomposition found more of the interesting, unknown events. Hence we conclude that the decomposition technique is superior to the Holt-Winters approach in this context, though the improvement is minor in comparison with the benefit of combining the two data sources.

Holt-Winters could no doubt have its parameters more finely tuned (it has 5) to get better results, and one could do likewise with the decomposition method. However, we have relatively few samples of events to be detected. This means that we can only coarsely measure the selectivity of the algorithms, and hence such tuning is a dicy business. Further, such precise tuning would then raise the point of whether the parameter values should be the same everywhere, for all tests. It seems likely that the optimal parameter values would be different on different data sets (e.g. for different PoPs), and so we now have a complicated process to learn what parameters work, which must be repeated every time the network changes. Also, there will always be a trade-off between selectivity and sensitivity, but we must be somewhat conservative here for the same reasons. To summarize, although one might improve performance slightly, this would be at the cost of robustness of the algorithms.

Of course one can consider improving the algorithms used for detection of both SNMP and BGP events to reduce the false alarm rate individually. However, there are in general fundamental problems with this approach. There are artifacts in the data, as well as genuine anomalies in both data sets that are the result of causes that we are not interested in detecting. For instance, the SNMP data may naturally vary for a number of reasons we do not consider to be network problems, at least in the sense that there is nothing we can do about them using current IP infrastructure: for instance traffic may vary from its normal patterns on holidays. Similarly BGP routes are not all responsible for equal traffic volumes. A large number of routes may change, but if these routes do not carry much traffic, there will be little impact on the network. Furthermore, these routes might be changing due to changes in policy outside the network's administrative domain, and therefore once again there is little action needed from network operators, and little need to detect this event. An additional concern is problems in the data: SNMP data can contain erroneous or missing data, which could appear as untoward spikes, or dips in the data. The BGP data also contains artifacts – for instance, if a BGP session between peers is reset, the number of routes from this table drops temporarily to zero, even if they are quickly replaced, so that the event has only a small impact on traffic.

By combining multiple sources of data we gain a specific, and powerful way of detecting forwarding anomalies, which avoids any of the above pitfalls.

5. CONCLUSION

This paper has described an important class of network anomalies — forwarding anomalies — and specific methods for combining routing and traffic data to perform accurate forwarding anomaly detection. The choice of data sets used here was motivated by a specific example, but we found them to be a good choice — the method has a perfect detection rate, while dramatically reducing the false alarm rate. Moreover, the method is automated and self-training — essential characteristics for deployment in large operational networks.

Further, we found several events in the data, using this technique, that had not been previously detected through any other alarming mechanism, and are therefore worthy of further study in their own right. These events may be the result of undiagnosed internal network problems, but with more likelihood are the result of external network changes, outside the normal view of the single domain under investigation. Thus we may be able to gain a window into major external Internet changes.

The method is very simple, but this should be seen as an advantage, as simplicity makes the method more scalable, and more easily extendible to include additional data sets. For instance, we might use data gathered from

- OSPF (via a route monitor similar to that used to collect BGP data [23]),
- Data from other external route monitors such as those of RIPE or Routeviews [8],
- Data from active network performance probes,
- Router logs, and
- Flow level data, such as that from Netflow, perhaps via an algorithm such as that proposed in [14].

One could also improve the algorithms applied here to individual data sets, for instance by applying the wavelet techniques of [13], and there are also many different ways in which data from these various sources could be combined. We have used possibly the simplest method for data fusion available – the `AND` operation, but there is a great deal of literature in the areas of pattern recognition, detection and classification, that is relevant to this task, and many much more sophisticated methods for connecting these data sets — for instance in [9] they propose the use of Bayesian networks for combining features. We have not found such methods to be necessary to build a useful algorithm (we already have practically useful results), but these methods would most likely improve performance still further.

Finally, we have considered only one class of network anomalies here — forwarding anomalies, but it is likely this sort of technique could be extended to other sorts of anomalies, for instance of application in security.

6. REFERENCES

[1] A.Brown and D. A. Patterson, “To err is human,” in *Proceedings of the First Workshop on Evaluating and Architecting System dependability (EASY ’01)*, (Gteborg, Sweden), 2001.

[2] D. Patterson, A. Brown, P. Broadwell, G. Candea, M. Chen, J. Cutler, P. Enriquez, A. Fox, E. Kiciman, M. Merzbacher, D. Oppenheimer, N. Sastry, W. Tetzlaff, J. Traupman, and N. Treuhaft, “Recovery-oriented computing (roc): Motivation, definition, techniques, and case studies,” Tech. Rep. UCB//CSD-02-1175, UC Berkeley Computer Science, 2002.

[3] D. Oppenheimer, A. Ganapathi, and D. A. Patterson, “Why do Internet services fail, and what can be done about it?,” in *4th Usenix Symposium on Internet Technologies and Systems (USITS’03)*, 2003.

[4] D.J.Houck, K.S.Meier-Hellstern, F.Saheban, and R.A.Skoog, “Failure and congestion propagation through signalling control,” in *Proceedings of the 14th International Teletraffic Congress (ITC-14)* (J. Labetoulle and J. W.Roberts, eds.), vol. 1a, pp. 367–376, Elsevier, 1994.

[5] J. Strand, A. Chiu, and R. Tkach, “Issues for routing in the optical layer,” *IEEE Communications Magazine*, February 2001.

[6] Nanog mailing list `http://www.cctec.com/maillists/nanog/historical/0005/msg00073.html`, 5th May 2000.

[7] Nanog mailing list `http://www.cctec.com/maillists/nanog/historical/0210/msg00058.html`, 3rd October 2002.

[8] “University of Oregon Route Views Archive Project.” `www.routeviews.org`.

[9] C. Hood and C. Ji, “Proactive network fault detection,” *IEEE Trans. Reliability*, vol. 46, no. 3, pp. 333–341, 1997.

[10] M. Thottan and C. Ji, “Proactive anomaly detection using distributed intelligent agents,” *IEEE Network*, Sept/Oct 1998.

[11] A. Ward, P. Glynn, and K. Richardson, “Internet service performance failure detection,” *ACM SIGMETRICS Performance Evaluation Review archive*, vol. 26, pp. 38–43, December 1998.

[12] J. D. Brutag, “Aberrant behavior detection and control in time series for network monitoring,” in *Proceedings of the 14th Systems Administration Conference (LISA 2000)*, (New Orleans, LA, USA), USENIX, December 2000.

[13] P. Barford, J. Kline, D. Plonka, and A. Ron, “A signal analysis of network traffic anomalies,” in *ACM SIGCOMM Internet Measurement Workshop*, (Marseilles, France), November 2002.

[14] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, “Sketch-based change detection: Methods, evaluation, and applications,” in *ACM SIGCOMM Internet Measurement Conference*, (Miami, Florida, USA), October 2003.

[15] Katzela and Schwartz, “Schemes for fault identification in communication networks,” *IEEE/ACM Transactions on Networking*, vol. 3, 1995.

[16] M. Grossglauser, N. Koudas, Y. Park, and A. Variot, “Falcon: Fault management via alarm warehousing and mining,” in *NRDM 2001 workshop*, (Santa Barbara, CA), May 2001.

[17] R. NCC, “Routing Information Service Raw Data.” `http://abcoude.ripe.net/ris/rawdata/`.

[18] S. H. Steiner, “Grouped data exponentially weighted moving average control charts,” *Applied Statistics*, vol. 47, no. 2, 1998.

[19] G. Wright and W. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation*. Addison-Wesley, 1995.

[20] P. Brockwell and R. Davis, *Time Series: Theory and Methods*. Springer-Verlag, 1987.

[21] M. Roughan and J. Gottlieb, “Large-scale measurement and modeling of backbone Internet traffic,” in *ITCOM, Boston, MA, USA*, 2002.

[22] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, “Experience in measuring internet backbone traffic variability: Models, metrics, measurements and meaning,” in *to appear in the International Teletraffic Congress-18*, 2003.

[23] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb, “A case study of ospf behavior in a large enterprise network,” in *ACM SIGCOMM Internet Measurement Workshop 2002*, 2002.