Spectral Analysis of Bottleneck Traffic

Xinming He, Christos Papadopoulos, John Heidemann, Urbashi Mitra, Usman Riaz, Alefiya Hussain[§]

***Abstract*— Internet traffic contains a rich set of periodic patterns imposed by various processes. Examples include back-to-back packet transmissions on bottleneck links, periodic routing information exchange, transport layer effects such as TCP windowing behavior, and application layer effects such as misconfigured DNS clients. Analyzing such periodic patterns has wide applications, including a better understanding of network traffic dynamics, diagnosis of network anomalies, and detection of DDos attacks. However, current understanding of periodic behavior in aggregate traffic is quite limited. Many previous approaches often analyze traffic on a per-flow basis, and do not scale well to analyze high speed network traffic.**

This paper explores the application of spectral techniques to analyze network traffic. We propose an experimental methodology to guide the application, and as a case study, we use this methodology to analyze the spectral characteristics imposed by bottleneck links on aggregate traffic.

In our approach, we passively gather packet traces from the network and then apply spectral techniques to extract periodic patterns embedded in the trace, particularly the regularities imposed by various bottleneck links. Unlike techniques utilizing packet inter-arrival time, our approach does not require flow separation or grouping. The only information required is the packet arrival time. Our experiments show that bottleneck links impose distinct signatures on the underlying traffic, and these signatures can be detected by a downstream monitoring point. We introduce four non-parametric algorithms based on the Bayes Maximum-likelihood Classifier to detect bottleneck flows inside the aggregate, and evaluate their performance using real Internet traffic. As our future work, we plan to design parametric detection algorithms for better performance, and apply the methodology to study other periodic network phenomena.

***Index Terms*— Spectral Analysis, Bottleneck Traffic**

I. Introduction

There exist a variety of processes that govern the generation and shaping of Internet traffic. Many of them are periodic and reside at different communication layers. For example, fixed bandwidth at the link layer can cause packets to be transmitted back-to-back, creating periodic patterns in network traffic. At the network layer and transport layer, exchange of routing information and the TCP windowing mechanism can result in periodic packet transmission on the network. At the application layer, traffic automatically generated by machines, such as zombies in DDos (Distributed Denial-of-service) attacks or misconfigured DNS clients, can exhibit strong regularities. Such periodic processes imprint their own *periodic signatures* on network traffic. Periodicities are also visible at several timescales, ranging from microseconds (e.g., clocking out packets on gigabit links) to days and years (e.g., diurnal cycles and seasonal traffic variations).

Studying such periodicities can have wide applications, including a better understanding of network traffic dynamics, diagnosis of network anomalies, and detection of DDos attacks. For example, typical Dos attacks involve sending small packets at the maximum speed the machine and the network can support. Analyzing the traffic through a link saturated by such attack packets will reveal abnormally strong high frequency components compared with a link saturated by normal traffic composition. This strong high frequency signal may be utilized to distinguish a Dos attack from congestion due to high normal traffic load. Another example is to detect attacks attempting to overload a web server through repeated requests. Requests automatically originated by machines are typically more regular and this can be used to distinguish them from human-originated web requests.

Spectral techniques have been widely used in many other fields to detect hidden patterns and trends from noisy background. Examples include sonar detection of submarine signals from the ocean acoustic background, processing of weather data to model its patterns and forecast its future, analysis of stock market and other financial markets, etc. In the past few years, researchers have begun to apply spectral techniques to analyze network traffic for various purposes. Their work presents strong evidence that applying such techniques to the analysis of network traffic is a very promising approach to study denial-of-service attacks [1], [2], DNS traffic behavior [3], traffic anomalies [4], and even protocol behavior in encrypted traffic [5]. However, current understanding of periodic behavior in general aggregate traffic is limited. Many previous approaches often analyze traffic on a per-flow basis, and do not scale well to analyze high speed network traffic.

This paper explores the application of spectral techniques to analyze network traffic. We propose an experimental methodology to guide the application, and as a case study, we use this methodology to analyze the spectral characteristics imposed by bottleneck links on aggregate traffic. As a bottleneck link is saturated by packets, it transmits packets out back-to-back, which results in strong regularity in the packet stream. Such regularity can be used to detect bottleneck traffic from aggregate. Bottleneck traffic conveys important information about network status and is useful for network traffic engineering and planning. For example, if network operators can detect a link is constantly congested, they can increase the link capacity or divert part of the traffic through other links to ease the congestion.

Spectral techniques open a new approach to study bottleneck traffic. Such an approach would be advantageous compared to current techniques, such as those based on SNMP statistics

[§] X. He, C. Papadopoulos, J. Heidemann, A. Hussain are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA (email: xhe@usc.edu; {christos,johnh,hussain}@isi.edu).

U. Mitra and U. Riaz are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA (email: {ubli,uriaz}@usc.edu).

since not all network devices can provide SNMP statistics and problem may hide in the coarse SNMP information, and those using active probes since our approach is completely passive and incurs no additional network traffic. It is also complementary to techniques using packet inter-arrival time [6], since it can be carried out without flow separation and grouping. The only information required is the packet arrival time. This is valuable in the sense that it does not rely on packet content including packet header information.

The rest of the paper is organized as follows. We first propose our experimental methodology in section II. Then we visually demonstrate the spectral characteristics imposed by bottleneck links under various environments in section III. Four detection algorithms based on Bayes Maximum-likelihood Classifier are introduced in section IV, and evaluated using real Internet traffic in section V. Section VI reviews related work. Section VII concludes the paper.

II. Experimental Methodology

In recent years there have been a number of papers from the network research community that use signal processing techniques to analyze Internet traffic for various purposes. Examples include distinguishing single-source and multi-source DDoS attacks based on their spectra [1], detecting Dos attacks and other network anomalies by analyzing IP flow-level and SNMP information using wavelets [4], and detecting TCP flows based on its windowing behavior [7], [5]. Their work shows that spectral techniques can be a very powerful tool for network traffic analysis.

However, as warned by Partridge in [8], [9], there is a danger in blindly applying signal processing techniques to networking without careful analysis and knowledge of the ground truth. Without careful examination of the ground truth, we may reach wrong conclusions or interpret incorrectly the ”pretty pictures” obtained from spectral analysis.

To face the challenge, it is essential to come up with a clear methodology that defines the path from the raw data to the final conclusion. Figure 1 illustrates our experimental methodology originally suggested by Antonio Ortega. It contains three main components, data generation, data representation, and detection/estimation. The more detailed steps inside the three components are:

1) **Representative Data Sets:** This is the generation of representative data sets. It can be gathered from real-world traffic, generated from controlled lab experiments, or in more abstract level, synthesized from network simulations.
2) **Measurable Real-world Events** This is the selection of real-world events that can be directly measured. Examples are packet arrivals, packet losses, connection establishment, connection tear-down, etc. The corresponding raw measurement data can be packet arrival time, packet length, packet delay, packet loss rate, connection duration, etc. For the detection of bottleneck traffic, we select packet arrivals as the measurable event, and the raw measurement data are packet arrival times.
3) **Time Domain Representation:** This involves the conversion from raw traffic measurements to the signal represented by time series. This process is also called sampling. Depending on the sampling period length, we can have even sampling (with the same sampling period length), or uneven sampling (with varying sampling period length). In our work, we choose even sampling, as we use Fourier transformation in the following step.
4) **Spectral Domain Representation:** This transforms the time domain signal representation into the spectral domain signal representation. Examples include Fourier transformation, and wavelet analysis with different types of wavelets.
5) **System Modeling:** Based on knowledge gained from above steps, we model the underlying processes to capture the key features of the event and how they evolve. We do not intend to model the general Internet traffic, but only specific aspects of the traffic that affect detection.
6) **Non-parametric Detection:** Non-parametric detection methods identify and detect high-level events, such as existence of bottleneck flows, using heuristics. They do not require explicit modeling of the underlying processes.
7) **Parametric Detection:** Parametric Detection will utilize the system model of the underlying processes, and improve the detection probability of the event of interest.

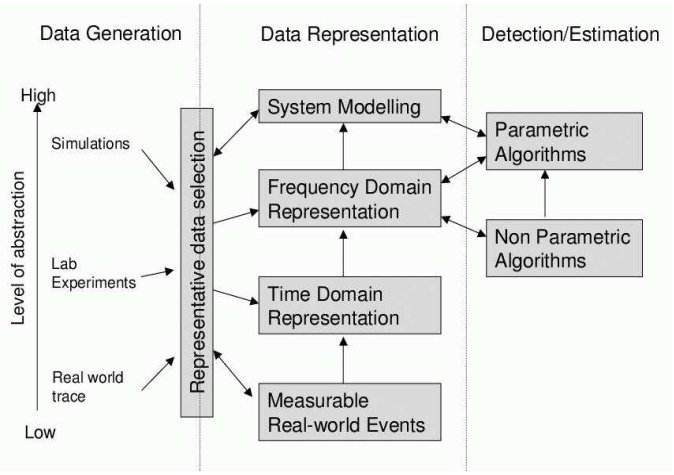


Fig. 1. Experimental Methodology (adapted from A. Ortega’s original proposal)

In our current work, we obtain representative data sets through both controlled lab experiments and real world experiments. Detailed description of these experiments is presented in section III and V. In the next subsection, we will describe step 2, step 3, and step 4 in detail for how to obtain the spectral representation from raw measurement data. Four non-parametric detection methods for detecting bottleneck traffic will be introduced in sectionIV. It is our future work to model the underlying processes and develop parametric detection algorithms.

A. Spectral Representation

In our methodology, there are three steps from raw measurement data to the spectral representation. They are selection of measurable real-world events, time domain representation, and frequency domain representation. In our current work, we adopt the techniques proposed by Hussain et al. in [1] for these three steps with slight modification. They are described in detail below.

First, we select packet arrivals as the measurable real-world event. We run tcpdump or other trace tools to capture timestamped packet traces from the network. The only information required from the packet trace is the packet arrival time. We divide the packet trace into l-second long slices before processing them in the following steps. The length of each slice is a configurable parameter and we will discuss shortly how to select it.

For the time domain representation, we sample each slice with a sampling rate p (we will discuss shortly how to select a proper p) to obtain a time series X, where $X(i)$ is the number of packets that arrive in the time period $[\frac{i}{p}, \frac{i+1}{p})$. The time is relative to the start of the slice, and i varies from 0 to $l \times p - 1$. This results in $N = l \times p$ number of samples for each slice. In addition, we subtract the mean arrival rate before proceeding with spectral transformation in the next step, since the mean value results in a large DC component in the spectrum that does not provide useful information for our purposes.

To get the frequency domain representation, we compute the power spectral density (PSD) by performing the discrete-time Fourier transform on the autocorrelation function (ACF) of the time series. The autocorrelation is a measure of how similar the steam is to itself shifted in time by offset k [10], [11]. When $k = 0$ we compare the packet stream to itself, and the autocorrelation is maximum and equals to the variance of the packet stream. When $k > 0$ we compare the packet stream with a version of itself shifted by lag k. The autocorrelation sequence $c(k)$ at lag k is

$$c(k) = 1/N \sum_{t=0}^{N-k} (X(t) - \bar{X})(X(t+k) - \bar{X}); \qquad (1)$$

where $\bar{X}$ is the mean of $X(t)$, N is the number of samples, and k varies from $-N$ to N.

The power spectral density $S(f)$ is obtained by applying discrete-time Fourier transform to the autocorrelation sequence of length M and using its magnitude.

$$S(f) = |\sum_{k=0}^{M} c(k) e^{-\imath 2\pi f k}| \qquad (2)$$

In addition, we calculate the cumulative spectrum P(f) as the power in the range 0 to f, and normalize P(f) by the total power to get the normalized cumulative spectrum (NCS) C(f).

$$P(f) = \sum_{i=0}^{f-1} \frac{S(i) + S(i+1)}{2} \qquad (3)$$

$$C(f) = \frac{P(f)}{P(f_{max})} \qquad (4)$$

Intuitively, spectrum $S(f)$ captures the power or strength of individual observable frequencies embedded in the time series, while the normalized cumulative spectrum $C(f)$ shows their relative strength.

There are two important parameters in the above steps. The first one is the length of each trace slice l. If the slice length l is too short, the spectrum will be sensitive to temporary or transient phenomena on the network. If it is too long, the arriving process is unlikely to be stationary. Since we target the spectral characteristics of bottleneck flows, we use a default value of 5 seconds for the slice length l.

The sampling rate p is another important parameter. Given a sampling rate p, the highest frequency that is observable is $\frac{p}{2}$ according to the Nyquist Theorem. If the sampling rate is too low, aliasing can occur. It it is too high, it will increase both storage and processing overhead unnecessarily. For a given link speed and packet size, one can compute the maximum required sampling rate by computing the minimum packet inter-arrival time and sampling at twice that frequency. A more thorough exploration of varying sampling rate is the subject of future work. In this paper, unless otherwise stated, we select a conservative sampling rate of 100kHz, which is sufficiently high to reduce the aliasing effect for typical packet streams over a 100Mbps Ethernet.

III. Visualizing Spectral Characteristics of Bottleneck Traffic

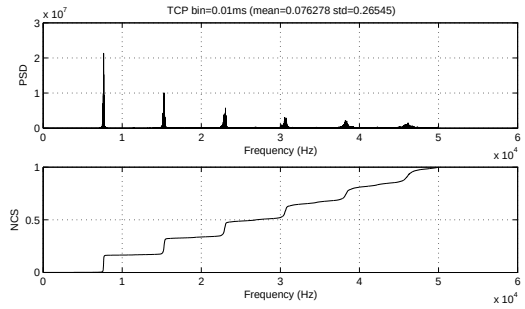
Before introducing algorithms that can detect bottleneck flows from aggregate traffic, we first show visually the spectral characteristics imposed by bottlenecks on aggregate traffic to qualitatively demonstrate the feasibility of detection. Our assumption is that if bottleneck flows can be visually detected from the spectrum, then a detection algorithm is possible.

In this section, we will present the spectrum of aggregate traffic under various scenarios to illustrate the signature imposed by bottlenecks and how it is affected by cross traffic. We start from simple scenarios where there is no cross traffic, and proceed to more complex environments with different types of cross traffic.

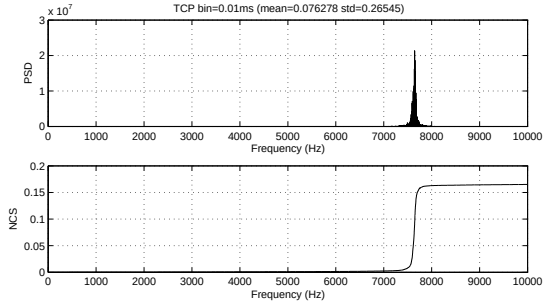
A. Signatures of Bottleneck Links

When a link is saturated, it sends packets out back-to-back. Assuming all packets are of the same length, we will see a single periodic pattern with the period equal to the packet transmission time (= link bandwidth / packet length). Even if packets are of different length, Internet traffic has certain packet length distribution as reported in [12] and [6]. The spectrum of the traffic through a saturated link will still show strong energy on frequencies regulated by the link bandwidth and the packet length distribution.

In this section, we first reveal such characteristics imposed by bottleneck links by conducting experiments in a simple topology where the sender and the receiver are directly connected through an Ethernet link and the trace machine runs tcpdump to capture packets from the Ethernet link. We vary the bottleneck link bandwidth and the traffic composition to get the spectra under different scenarios.



(a) Complete Spectrum



(b) Partial Spectrum

Fig. 2. Spectral signature of a 100Mbps link saturated with a TCP flow

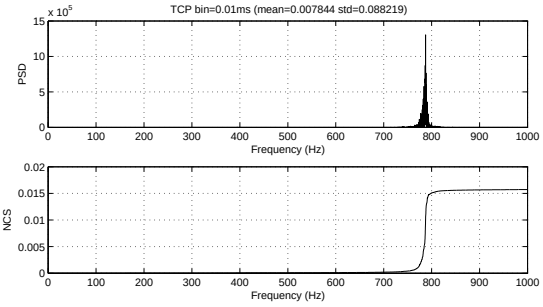


Fig. 3. Spectral signature of a 10Mbps link saturated with a TCP flow

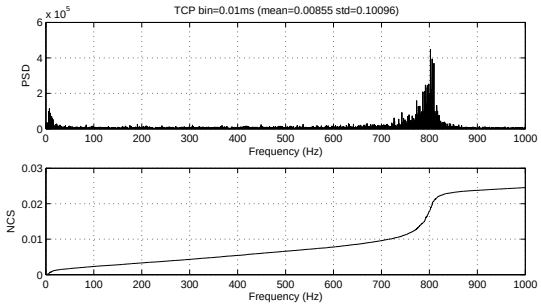


Fig. 4. Spectral signature of a 10Mbps link saturated with web traffic

For these experiments we use two traffic generation tools, namely Surge [13] and Iperf [14]. Surge is used to generate synthetic web traffic while Iperf generates controlled TCP and UDP streams, which, for example, can mimic file downloads (TCP mode) or Constant Bit Rate traffic (UDP mode). A typical experiment lasts for 30 seconds and contains six 5-second long slices. Although there is some variation in the power spectra for the six slices, the variation is small, and thus we only present the result from one representative slice here. We use both PSD and NCS since PSD captures the absolute power of individual frequencies while NCS shows their relative strength.

1) a 100Mbps Link Saturated with a Single TCP Flow: In the first experiment, we use a single Iperf TCP flow to saturate a 100Mbps Ethernet link. Since the sender and receiver are directly connected through the Ethernet link, the TCP window is large enough for the TCP flow to capture nearly the entire link bandwidth. The TCP throughput reaches almost 91.5Mbps.

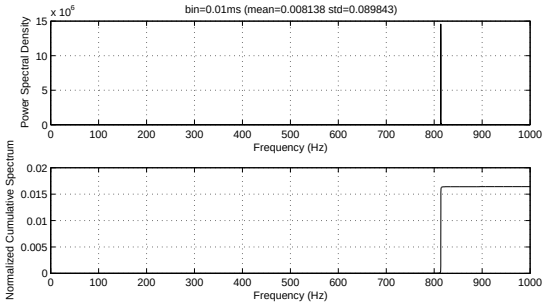
Figure 2(a) shows the full spectrum of the packet stream we observe on the Ethernet link. This specific example depicts a single TCP flow saturating a 100Mbps link with 1500-byte packets. The spectrum contains spikes at the fundamental frequency around 7630Hz and its multiples or harmonics. The harmonics exist because the signal is an impulse train in the time domain, resulting in a set of impulses in the frequency domain. The fundamental frequency is close to the maximum packet rate (8333 packets per second) on a 100Mbps link with 1500-byte packets. Since TCP adjusts its packet sending rate according to network conditions, it does not reach the 8333 packet per second rate, but as demonstrated in this example, the actual rate is quite close to the theoretical bound. Hence, a high energy concentration near the 8000Hz will be a strong indication that that the traffic contains a flow(s) through a 100Mbps bottleneck link.

To better demonstrate visually the effect of the bottleneck link on the fundamental frequency, we will zoom in to the interesting portion of the spectrum where the fundamental frequency of the bottleneck link lies, as illustrated in Figure 2(b), for all following results.

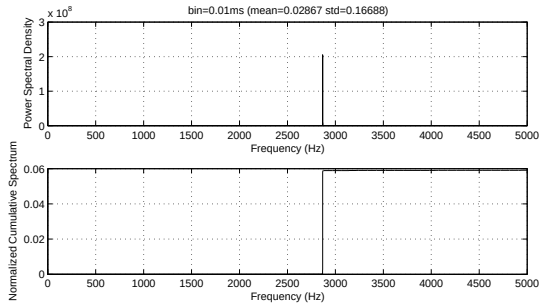
2) a 10Mbps Link Saturated with a Single TCP Flow: In this experiment, we change the link bandwidth to 10Mbps while keep the rest the same as in the previous experiment. The throughput of the Iperf TCP flow reaches almost 9.4Mbps. Figure 3 shows the corresponding spectrum of the packet stream. We see strong energy around 784Hz, which is close to the theoretical limit (833 Hz) imposed by a 10Mbps link with 1500-byte packets.

3) a 10Mbps Link Saturated with Web Flows: In this experiment, we replace the single TCP flow with web-like traffic generated by Surge [13]. We continue to use a 10Mbps Ethernet link, and configure Surge to emulate 640 "user equivalents" (UEs), achieving a throughput around 8.2Mbps.

Figure 4 shows the spectrum under this experiment. We see that there is still a spike around 800Hz, but it spreads out to a wider range than the single Iperf TCP flow case. Also, the highest amplitude is only 1/3 of the previous case. The reason is that Surge simulates multiple web flows which have different time duration depending on the download file size. It also simulates the on-and-off user browsing behavior. This results in lower link utilization and makes the bottleneck signature more blurry, but still there is strong energy concentration



(a) With 1500-byte Packets



(b) With 400-byte Packets

Fig. 5. Spectral signature of a 10Mbps link saturated with a UDP flow

around 800Hz because a number of 1500-byte packets are transmitted back-to-back in this case.

4) a 10Mbps Link Saturated with an Iperf UDP flow: To investigate how the spectrum changes with CBR traffic, we saturate a 10Mbps link with an Iperf UDP flow. We first configure Iperf to send out 1500-byte UDP packets with a sending rate greater than the link bandwidth. This yields a throughput of 9.6Mbps, slightly higher than the single Iperf TCP flow case.

Figure 5(a) depicts the spectrum with the Iperf UDP flow. We see that there is a single peak at 813.8Hz. Its amplitude is about ten times the highest amplitude in the single Iperf TCP flow case. This shows the UDP flow has a stronger regularity than the TCP flow. The reasons is because TCP will adjust its sending rate according to network conditions while the Iperf UDP sender does not take feedback from the network and it fully saturates the link.

In the second experiment, we configure Iperf to send out 400-byte UDP packets with a sending rate greater than the link bandwidth. This yields a throughput of 8.53Mbps. This is lower than the experiment with 1500-byte packets, because the Ethernet channel efficiency decreases with smaller packet size. The spectrum in Figure 5(b) shows a clear spike at 2867Hz, which is equal to the packet rate over the link. Its amplitude is also significantly higher than the experiment with 1500-byte packets.

From the above experiments we see that the spectrum of a saturated link can vary according to a number of factors. Among them, link bandwidth and packet length distribution are the two most important factors, and they will determine where the dominant frequency will appear. Beyond that, UDP or CBR streams appear more regular than TCP flows, resulting in higher amplitude in the bottleneck frequency. On the other hand, the complex interaction among multiple web flows may yield lower amplitude in the spectrum.

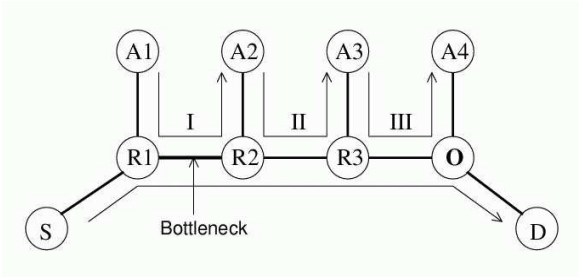


Fig. 6. Different types of cross traffic

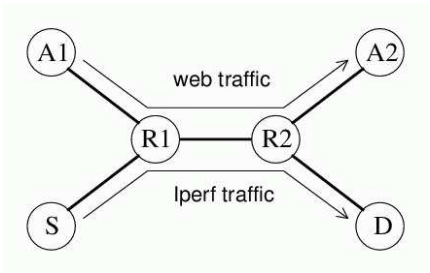


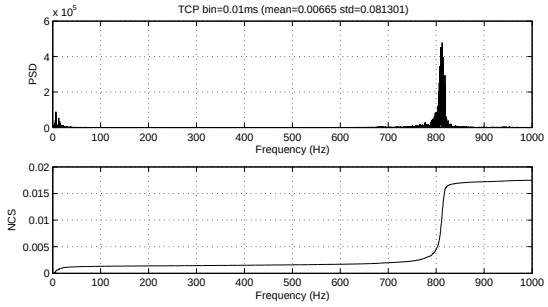
Fig. 7. Testbed topology

B. Effect of Cross Traffic

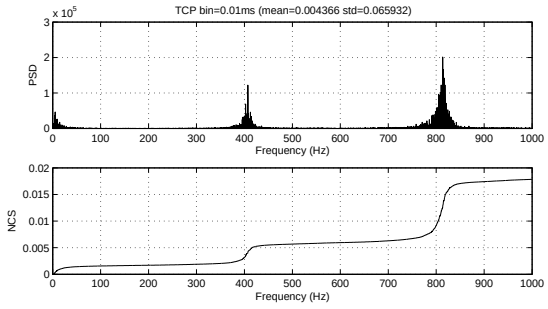
In previous examples, we present the spectral characteristics imposed by a bottleneck link when there is no cross traffic. In this subsection, we will investigate how cross traffic affects the spectral signature. We first classify cross traffic into three classes, and then carry out experiments to visually demonstrate the impact from each of them on the the bottleneck signature.

1) Classification of cross traffic: Figure 6 illustrates three classes of cross traffic that might affect our observations of the bottleneck traffic. In the figure, traffic travels from source S to destination D passing through a bottleneck between R1 and R2, and we monitor traffic on link R3-O at the observation point O. We are interested in observing the bottleneck signal generated at the R1–R2 link at the observation point.

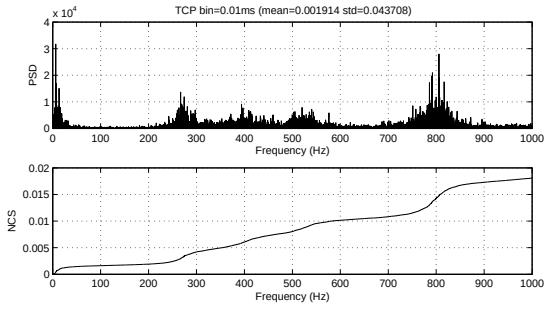
- *Type I, unobserved bottleneck traffic:* cross traffic that traverses the bottleneck link but does not reach our observation point. Such traffic carries part of the energy from the signature imposed by the bottleneck. Missing this traffic may attenuate the signal strength observed at our observation point.
- *Type II, unobserved non-bottleneck traffic:* Cross traffic that is introduced after the bottleneck link, but is not observed at the observation point. Such traffic can distort the signal of the bottleneck link, as it competes with bottleneck traffic in the shared path, introducing variation in packet arrival times and making the signal more noisy.
- *Type III, observed non-bottleneck traffic:* cross traffic that does not go through the bottleneck link but is observed at our observation point. It is also called background



(a) with light web traffi c (10 UEs)

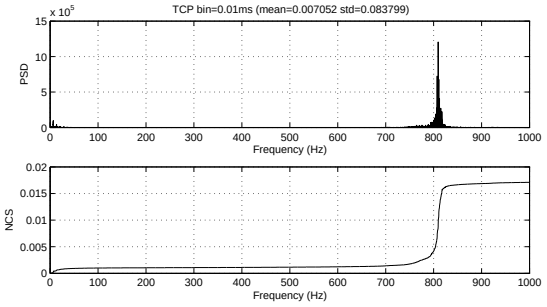


(b) with medium web traffi c (80 UEs)

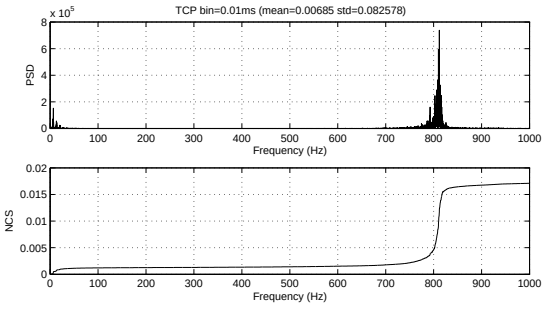


(c) with heavy web traffi c (640 UEs)

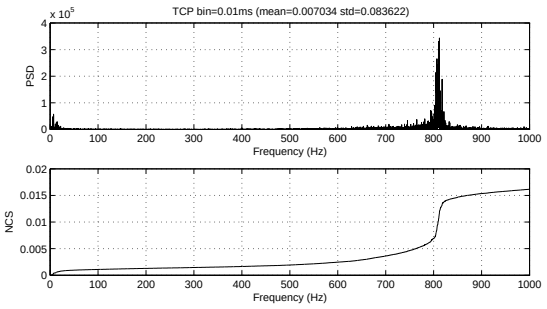
Fig. 8. Power spectra as Type I cross traffi c increases



(a) with light web traffi c (10 UEs)



(b) with light web traffi c (80 UEs)



(c) with heavy web traffi c (640 UEs)

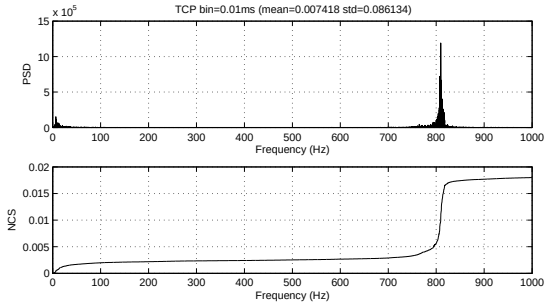
Fig. 9. Power spectra as Type II cross traffi c increases

traffic. Its impact comes from two aspects: (a) like type II cross traffic, it competes with bottleneck traffic in the shared path; (b) it directly influence the spectrum of observed aggregate traffic, as the aggregate contains both bottleneck traffic and background traffic.

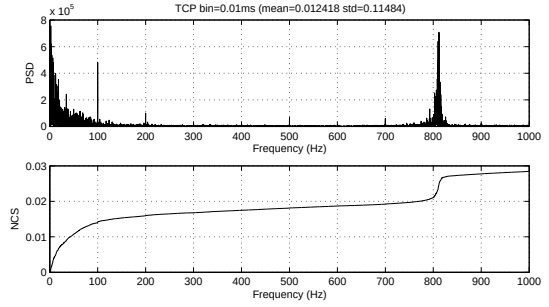
2) Impact of Type I Cross Traffic: To evaluate the impact of the three types of cross traffic identified above, we use a dumbbell topology depicted in Figure 7. We first investigate the impact of Type I cross traffic. In this experiment, we set the capacity of all links to 10Mbps. There are two types of traffic, a single Iperf TCP flow from node S to D and web traffic generated by Surge between nodes A1 and A2. The bottleneck link will be link R1-R2. We observe the traffic on link R2-D. In this experiment, the Iperf flow serves as the bottleneck traffic, while the web flows serve as Type I cross traffic. We vary the number of web users simulated by Surge to control the volume of Type I cross traffic competing with the Iperf TCP flow on link R1-R2.

Figure 8 shows the power spectra of the traffic observed at D when the number UEs in Surge vary from 10 to 640. The corresponding throughput at link R1-R2 stays around 8.3Mbps, while the throughput at link R2-D is decreases from 8.2Mbps to 5.3Mbps and 2.3Mbps as cross traffic increases. We can see that as the volume of Type I cross traffic increases, the energy around 800Hz becomes weaker, but still visible. In addition, we see a new spike around 400Hz in Figure 8(b), where it is common to see a Surge packet transmitted between two Iperf packets in link R1-R2. Figure 8(c) shows new spikes around 266Hz, 400Hz, and 532Hz (a multiple of 266Hz), where it is common to see one or two Surge packets transmitted between two Iperf packets. The presence of the new spikes indicates contention at the bottleneck link due to Type I cross traffic. This phenomena was also observed in study of packet inter-arrival times [6],

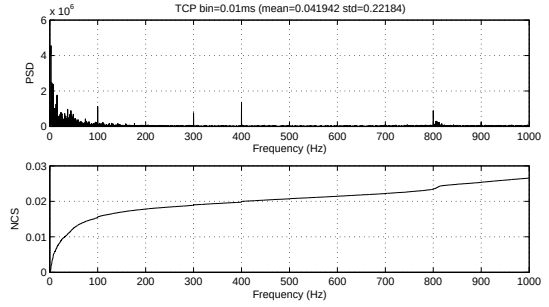
3) Impact of Type II Cross Traffic: To investigate the effect of Type II traffic, we set the capacity of link S-R1 to 10Mbps, and the capacity of all other links to 100Mbps. Other settings remain the same as in the previous experiment.



(a) with light web traffic (10 UEs)



(b) with medium web traffic (80 UEs)



(c) with heavy web traffic (640 UEs)

Fig. 10. Power spectrum as Type III cross traffic increases

In this experiment, link S-R1 becomes the bottleneck link, and the web flows from A1 to A2 serve as type II cross traffic.

Figure 9 shows the power spectra spectra of the traffic observed at D as the cross traffic volume increases. The corresponding throughput at link R1-R2 is 8.7Mbps, 13Mbps, and 38.9Mbps, respectively, and the throughput at link R2-D is steady around 8.3Mbps. We observe that as the volume of Type II cross traffic increases, the energy around 800Hz will spread out to a wider range, and the highest amplitude also decreases accordingly. But there is still strong energy concentration around 800Hz. The changes to the spectrum are due to the contention between the bottleneck traffic and the cross traffic at link R1-R2.

4) Impact of Type III Cross Traffic: Finally we consider the effect of Type III traffic. We use exactly the same setting in the second set of experiments, but move the observation point to link R1-R2. Figure 10 shows the spectra of the aggregate traffic through R1-R2 as load grows. We observe the following. First, the energy around 800Hz decreases, as more packets from the bottleneck flow experience queuing delay in link R1-R2 due to competition with the web traffic. Second, the energy over other frequencies increases as the observed aggregate traffic contains not only the bottleneck flow, but also the web traffic which has a strong low frequency component. Finally, the relative visibility of the bottleneck signal around 800Hz decreases (both in PSD and NCS) as the cross traffic load increases. When the cross traffic load increases to some point, the bottleneck signal becomes hard to detect visually from the spectrum of aggregate traffic.

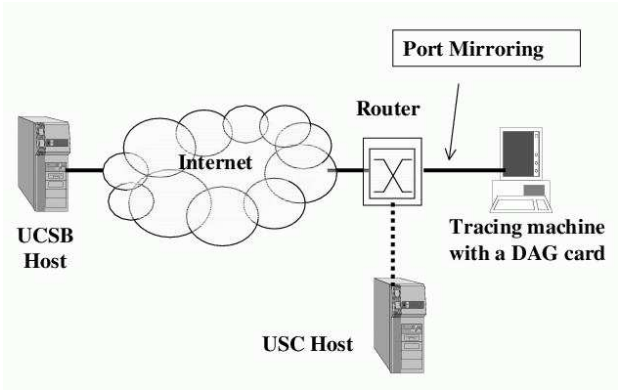


Fig. 11. Setup of the wide-area network experiment environment

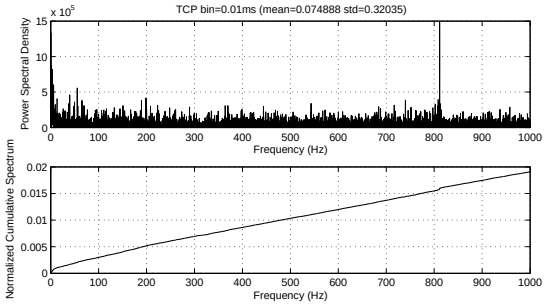
C. Wide-area network experiments

We next validate our testbed observations on the Internet by considering a wide-area, multi-hop topology with richer, live background traffic. The experiment environment is illustrated in Figure 11. We place the trace machine close to a router at the edge of USC. The latter forwards all incoming traffic through the USC Internet II link to the trace machine by port mirroring. The trace machine then records all packets using a Endace DAG Network Monitoring Card [15], which is capable of keeping up with 1Gbps link speed. Our bottleneck flow is from a PC connected to a 10Mbps LAN in University of Santa Barbara to a host at USC.

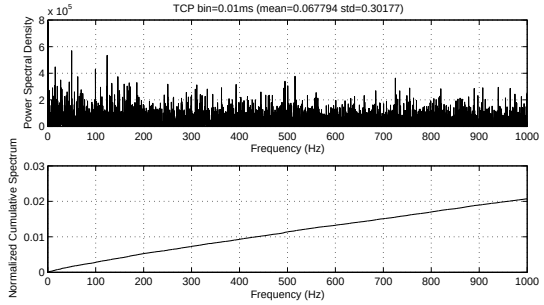
Figure 12(a) shows the spectrum of aggregate traffic observed at the USC trace machine with the Iperf TCP bottleneck flow inside. The throughput of the TCP flow is around 9.4Mbps, suggesting the bottleneck is the LAN at the source. The aggregate traffic volume is 24.2Mbps. We see a clear spike around 800Hz in PSD, suggesting that the bottleneck traffic is visible even *mixed in* aggregate traffic, although it is relatively small in NCS. For comparison Figure 12(b) shows the spectrum of aggregate traffic at a later time without the Iperf TCP bottleneck flow. The strongest energy around 800Hz is less than one fifth of the strength when the bottleneck flow is present.

IV. Detection of Bottleneck Signatures

In the previous section we have shown that bottleneck signatures can be visually observed. To automatically detect bottleneck signatures from the aggregate traffic spectrum, in this section we propose four non-parametric detection methods



(a) with an Iperf flow on a 10Mbps bottleneck link



(b) without the Iperf flow

Fig. 12. Power spectra of aggregate traffic at USC Internet II link

based on Bayes Maximum-likelihood classifier. In our current work, we focus on the accuracy of these detection methods, and it is future work to study and improve their performance in terms of running efficiency.

In Bayes Maximum-likelihood classifier, instances are classified into different groups. For example, packet traces without bottleneck flows are put into one group, and packet traces containing a 100Mbps bottleneck flow are put into another group. Without loss of generality, we name the group without bottleneck flows as group H_0, and the group with bottleneck flows as group H_w, where w is the bandwidth of the bottleneck.

In order to make the classification, Bayes Maximum-likelihood classifier first needs to select certain measurable property from the packet trace to distinguish different groups. This property can be the amplitude at certain frequency in the packet trace spectrum, or the highest amplitude at certain frequency band, etc. How to select this property is the main difference among the four non-parametric detection methods, and we will describe them in detail shortly. After selecting the property, Bayes Maximum-likelihood classifier needs a training phase to estimate the PDF (Probability Density Function) for the distribution of the values of this property for each group. Each group has its own PDF, and it is used for subsequent classification/detection.

Figure 13 illustrates the training process. For group H_0, we first gather packet traces that do not have bottleneck traffic and process them to get their spectra. We then gather the value of the selected property, e.g., the amplitude at a selected frequency, across all packet traces to form a distribution and estimate the PDF for group H_0. For group $H_{100Mbps}$, we

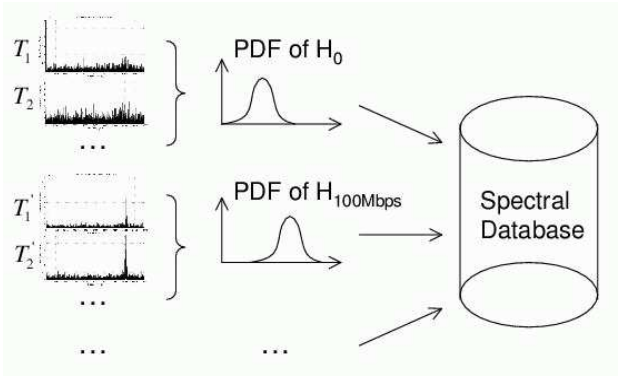


Fig. 13. Training of Bayes Maximum-likelihood classifier

intentionally introduce a flow to saturate a 100Mbps bottleneck link, gather packet traces again, and process them to estimate the PDF for group $H_{100Mbps}$. We repeat the same procedure for other groups, and put the PDFs of all groups into a spectral database which is used in subsequent classification/detection.

For classification/detection, we match the new packet trace against the database, and declare the trace contains traffic through a bottleneck link of bandwidth w if the trace is closer to group H_w than of group H_0. The match is carried out as follows. According to Bayes rule,

$$Pr(H|x) = Pr(x|H) * Pr(H)/Pr(x)$$

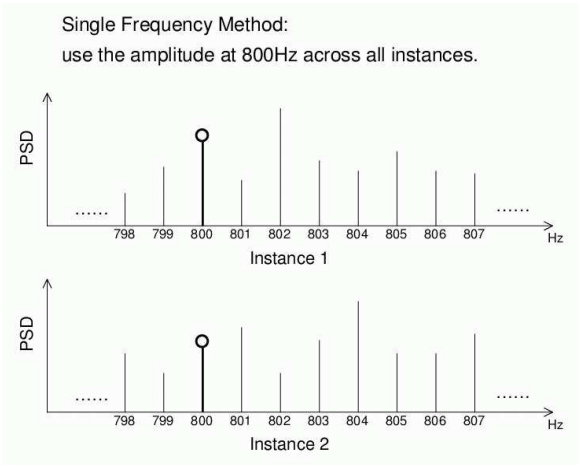
where $Pr(H|x)$ is the probability that a trace belongs to group H given the trace has value x on the selected property, $Pr(x|H)$ is the PDF of group H, $Pr(H)$ is the probability that a trace belongs to group H, and $Pr(x)$ is the probability that a trace has value x.

Bayes Maximum-likelihood Classifier classifies the packet trace with value x into group H_w if $Pr(x|H_w) * Pr(H_w) > Pr(x|H_0) * Pr(H_0)$, and vice versa. Due to the lack of information on $Pr(H_0)$ and $Pr(H_w)$, as the first step we directly compare $Pr(x|H_w)$ against $Pr(x|H_0)$ without considering $Pr(H_0)$ and $Pr(H_w)$. It is our future work to investigate $Pr(H_0)$ and $Pr(H_w)$ and include them in the decision process.

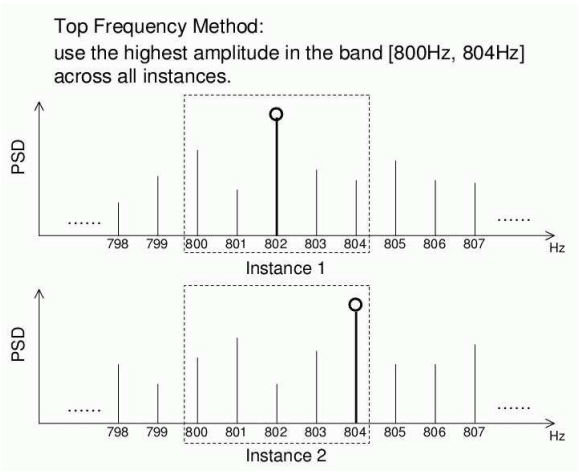
Based on how the property is selected, we have four different detection methods. Two of them, the Single Frequency Method and the Top Frequency Method, use a single variable for the detection. The other two use multiple variables, and they are the Top-M Frequencies Method and the All Frequencies Method. We describe them in detail below.

A. Method I: Single Frequency Detection

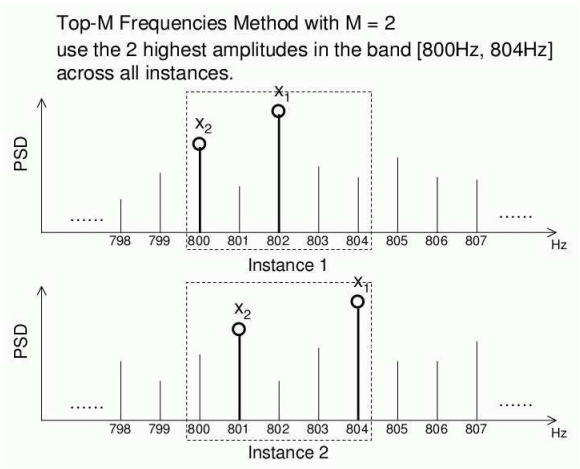
In this method, we first gather the distribution of the amplitude at a particular frequency F across all training instances for each group in the database. In the example illustrated in Figure 14(a), we gather the amplitude at 800Hz across all training instances in a group. After obtaining the distribution, we estimate the PDF (probability density function) for each group, and then follow the basic Bayes Maximum-likelihood Classifier to classify the new trace using its amplitude at frequency F. The intuition behind this method is that when



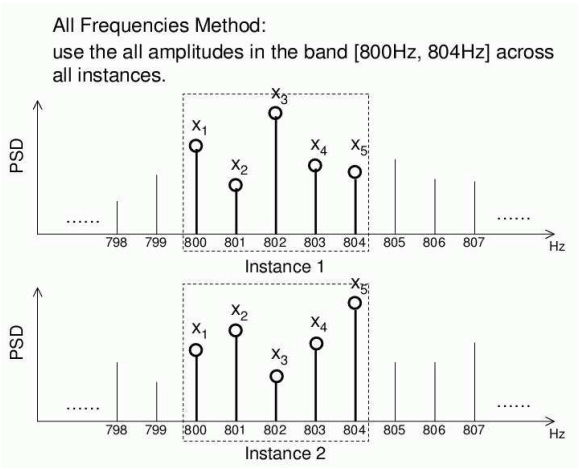
(a) Choosing x in the Single Frequency Method



(b) Choosing x in the Top Frequency Method



(c) Choosing x in the Top-M Frequencies Method, where M = 2



(d) Choosing x in the All Frequencies Method

Fig. 14. Illustration of Choosing Property x in Different Methods

bottleneck traffic is present, the aggregate traffic spectrum will typically have strong amplitude at some particular frequencies. Looking at any of these frequencies may yield clue to detect the bottleneck.

For simplification purpose, we approximate the PDFs for group H_0 and H_w with log-normal distributions with the corresponding parameters (μ_0, σ_0) and (μ_w, σ_w) estimated from the training set, i.e.

$$Pr(x|H_0) = \frac{1}{\sigma_0\sqrt{2\pi}} e^{-(x-\mu_0)^2/2\sigma_0^2}$$

$$Pr(x|H_w) = \frac{1}{\sigma_w\sqrt{2\pi}} e^{-(x-\mu_w)^2/2\sigma_w^2}$$

where x is the log of the amplitude at frequency F, (μ_0, σ_0) and (μ_w, σ_w) are the mean value and standard deviation for group H_0 and H_w in the training set, respectively.

Although the log-normal distribution is not always the best fit for the actual distribution, we choose it because it is simple and often a good approximation according to the central limit theorem. Our experiment results also show it can approximate the actual distribution fairly well. For example, Figure 15 shows the actual distributions of the amplitude (after log) at 8132Hz frequency for group H_0 and group H_w, respectively, in one experimental set involving a 100Mbps TCP bottleneck flow. The dashed lines are the log-normal distributions with the corresponding mean values and variances derived from the experimental set for group H_0 and group H_w, respectively. We see the dashed lines are not far apart from the solid lines.

With the log-normal distribution assumption, we can also simplify the matching process by directly solving the equation of the two log-normal distributions to find the region for group H_w and the region for group H_0. The equation of the two log-normal distributions typically yields two roots, but only one of them carries significance and is selected as the cut-off threshold x_w. If a new trace has an amplitude greater than x_w at frequency F, it will be classified as in group H_w. By trying the Single Frequency Method with different frequencies, we can find frequencies that best capture the difference between group H_0 and H_w.

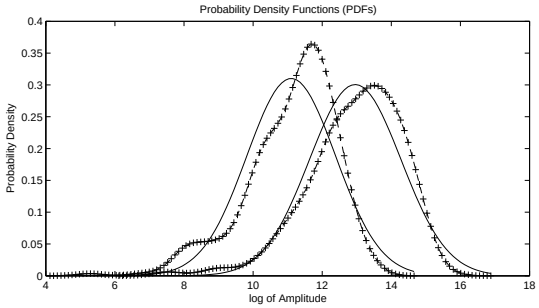


Fig. 15. PDFs for the distributions of the amplitude (after log) at 8132Hz for group H_0 (left solid line) and H_w (right solid line)

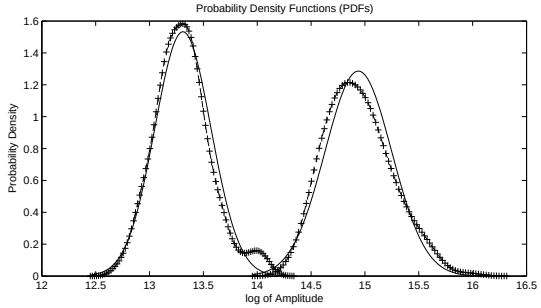


Fig. 16. PDFs for the distributions of the highest amplitude (after log) in [8100Hz, 8200Hz] for group H_0 (left solid line) and H_w (right solid line)

B. Method II: Top Frequency Detection

In the Top Frequency method, we first get the distribution of the highest amplitude in a particular frequency band across all instances for each group in the database. For example, in Figure 14(b), we gather the highest amplitude in the [800Hz, 804Hz] band across all instances in a group. After obtaining the distribution, we follow the same procedure as the Single Frequency method for the detection. The only difference here is that we use the highest amplitude in a frequency band for the detection, instead of the amplitude at one particular frequency.

The intuition behind this method is that when the bottleneck flow is present, it should typically have some strong amplitude in a particular frequency band related to the bottleneck bandwidth w and the packet size. Due to changes in the cross traffic and other time-variant factors, the strong amplitude may appear at different frequencies at different time, but it should stay in a relatively narrow band. So if we look the highest amplitude in this narrow band, we will get bigger difference between the cases with and without the bottleneck flow, compared with the Single Frequency Method. Hence it can result in a better distinction between group H_0 and H_w. The width of the band will depend on the impact of the background traffic on the spread of the bottleneck signature. If we select a very narrow band, it may not include the strong amplitude from the bottleneck for some instances. On the other hand, we should not use a very wide band as it may include strong amplitude caused by other reasons.

Figure 16 shows the distributions of the highest amplitude (after log) in the frequency band [8100Hz, 8200Hz] for group H_0 and H_w in the same experimental set as in Figure 15. We can see the two distributions can be closely approximated by log-normal distributions represented by the dashed lines. In addition, there is a wider gap between the two groups here, suggesting it easier to separate the two groups with the Top Frequency Method, compared with the Single Frequency Method.

C. Method III: Top-M Frequencies Detection

A generalization of the Top Frequency Method is to use M highest amplitudes instead of just the first highest amplitude in a frequency band for detection. It works in the following way. For each training instance, we select the M highest amplitudes in a particular frequency band to form a vector $[x_1, x_2, ..., x_M]$ from high to low. We then estimate the joint distribution of this vector for each group. Figure 14(c) illustrates the selection of the 2 highest amplitudes in the [800Hz, 804Hz] band across all instances for a group to obtain the joint distribution.

We approximate the joint distribution for both group H_0 and H_w using multi-variate log-normal distributions with the corresponding parameters (μ_0, C_0) and (μ_w, C_w), i.e.

$$Pr(x|H_0) = \frac{e^{-\frac{1}{2}(x-\mu_0)^T C_0 (x-\mu_0)}}{\sqrt{2\pi^m det(C_0)}}$$

$$Pr(x|H_w) = \frac{e^{-\frac{1}{2}(x-\mu_w)^T C_w (x-\mu_w)}}{\sqrt{2\pi^m det(C_w)}}$$

where x is a vector containing the log of the M highest amplitudes in the frequency band, (μ_0, C_0) and (μ_w, C_w) are the mean vector and the covariance matrix for group H_0 and H_w, respectively.

We estimate (μ_0, C_0) and (μ_w, C_w) based on the training set in the database, and then apply Bayes Maximum-likelihood classifier with the above PDFs to classify if an input trace with value x is in group H_w or not.

D. Method IV: All Frequencies Detection

In this method, we use all frequencies in a particular frequency band to form a multi-variate detector. For example, Figure 14(d) illustrates the selection of all amplitudes in the [800Hz, 804Hz] band across all instances for a group to obtain the joint distribution.

Similar with the Top-M Frequencies Method, we approximate the joint distribution for both H_0 and H_w using multi-variate log-normal distributions with the corresponding parameters (μ_0, C_0) and (μ_w, C_w). We use the training set to estimate the parameters (μ_1, C_1) and (μ_2, C_2). We then plug the values in the PDFs and apply Bayes Maximum-likelihood classifier to classify an input trace.

V. Experimental Evaluation

To evaluate the performance of the detection methods, we carried out experiments under different scenarios. These include:

- detecting bottlenecks of different bandwidth;
- detecting bottlenecks under different background traffic load;

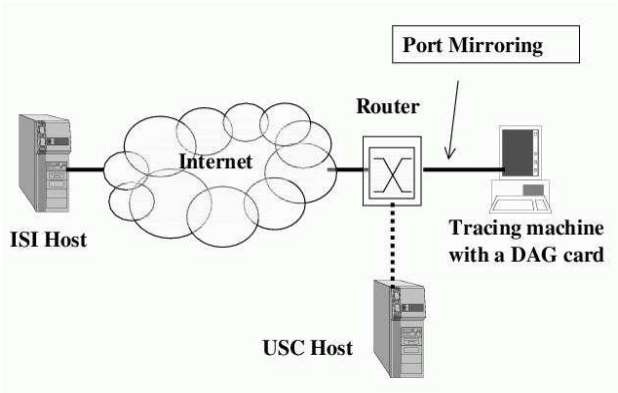
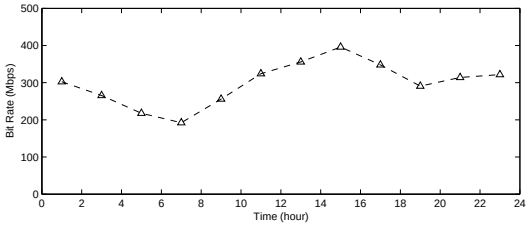
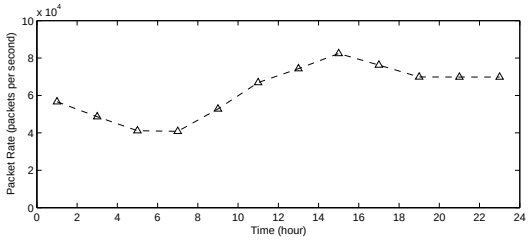


Fig. 17. Setup of the experiment environment



(a) Traffic Volume in Bit Rate (Mbps)

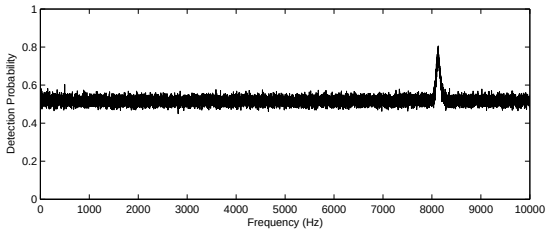


(b) Traffic Volume in Packet Rate

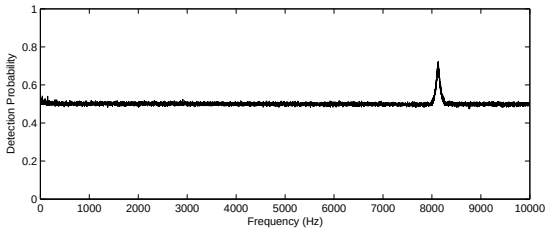
Fig. 18. Aggregate Traffic with a 100Mbps TCP Bottleneck Flow

- detecting bottlenecks saturated by different network protocols.

The basic experiment environment is illustrated in Figure 17. It is very similar to the environment in Figure 11, except that the Iperf flow is originated from ISI to USC. For each scenario, we gathered a pair of 5-minute long traces every two hours for 24 hours. In each pair, the first part was 5 minutes of background traffic alone, and in the second 5 minutes, we introduced a bottleneck flow of a particular type (e.g., an Iperf TCP flow through a known 100Mbps bottleneck, an Iperf TCP flow through a known 10Mbps bottleneck, and an Iperf UDP flow through a known 10Mbps bottleneck), and gathered the trace again. Each trace was then cut into 300 slices of 1 second long. These slices were then processed with a sampling rate of 200KHz to obtain the power spectral density. We use a shorter slice length here in order to get enough instances to have a meaningful distribution. We also use a higher sampling rate because the speed of the observed link is significantly higher.



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 19. Detection Probability of the Single Frequency Method with 100Mbps TCP bottleneck Traffic

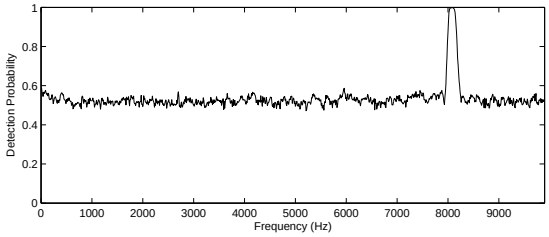
TABLE I
VARIATION WITH DIFFERENT TRAINING SETS FOR THE SINGLE FREQUENCY METHOD

Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	8131	0.743	0.722	12.783	11.282	11.946
3am	8127	0.783	0.721	12.975	11.196	12.144
5am	8133	0.820	0.700	12.958	10.829	11.890
7am	8132	0.807	0.717	13.084	11.038	12.082
9am	8124	0.778	0.714	12.875	11.129	12.025
11am	8134	0.707	0.721	12.461	11.489	12.205
13pm	8131	0.750	0.715	12.579	11.118	11.832
15pm	8127	0.702	0.720	12.463	11.370	12.045
17pm	8119	0.713	0.695	12.517	11.362	12.003
19pm	8131	0.737	0.734	12.796	11.381	12.127
21pm	8122	0.745	0.721	12.616	11.326	12.132
23pm	8135	0.737	0.709	12.606	11.328	12.042
mean	8129	0.752	0.716	12.726	11.237	12.039
std	5	0.038	0.010	0.214	0.183	0.110

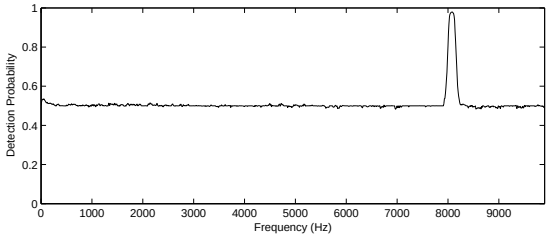
We train each detection method with a pair of traces, and then evaluate its performance by feeding the training set and the rest 11 pairs of traces to the detection method. The performance of the detection method is measured by the detection probability, which is the probability that the detection method gives the correct answer on whether a trace slice contains the particular bottleneck flow or not. We present the results under different experiment scenarios below.

A. Experiment I: Detecting 100Mbps TCP Bottleneck Traffic

In this experiment, the bottleneck flow is an Iperf TCP flow through a known 100Mbps bottleneck from ISI to USC. Figure 18 shows aggregate traffic volume in terms of bit rate and packet rate. In the graph, we use the average value for the 5-minute long trace with the Iperf flow to represent the traffic volume in the corresponding 2 hour interval. The figure shows that the traffic reaches the lowest (around 192Mbps or 40.8K packets per second) in the interval from 6am to 8am, and the highest (around 395Mbps or 83.4K packets per second) in the interval from 14pm to 16pm. The throughput of the Iperf TCP flow is about 90Mbps or 7.5K packets per second.

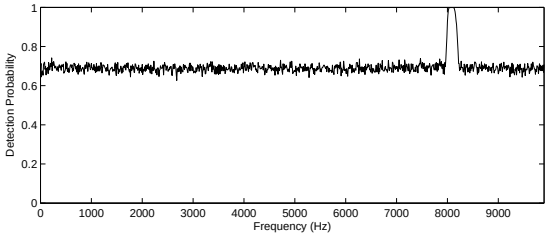


(a) detection probability on the training set

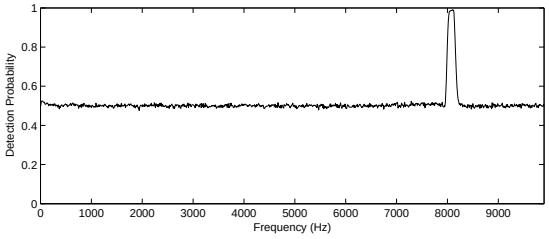


(b) average detection probability on all other sets

Fig. 20. Detection Probability of the Top Frequency Method with 100Mbps TCP bottleneck Traffic



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 21. Detection Probability of the Top-20 Frequencies Method with 100Mbps TCP bottleneck Traffic

TABLE II
VARIATION WITH DIFFERENT TRAINING SETS FOR THE TOP FREQUENCY METHOD

Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	8110	0.992	0.975	14.732	13.600	14.110
3am	8080	0.995	0.974	14.830	13.431	14.058
5am	8090	1.000	0.961	14.878	13.222	13.970
7am	8090	1.000	0.978	14.948	13.334	14.103
9am	8090	0.998	0.975	14.848	13.440	14.063
11am	8070	0.960	0.973	14.672	13.825	14.224
13pm	8070	0.987	0.953	14.537	13.497	13.950
15pm	8060	0.948	0.975	14.457	13.668	14.050
17pm	8080	0.962	0.980	14.669	13.743	14.178
19pm	8070	0.987	0.979	14.780	13.662	14.157
21pm	8100	0.993	0.978	14.703	13.561	14.104
23pm	8090	0.985	0.980	14.653	13.587	14.107
mean	8083	0.984	0.973	14.726	13.547	14.090
std	14	0.017	0.008	0.142	0.172	0.079

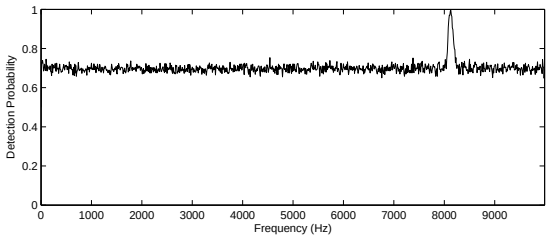
TABLE III
VARIATION WITH DIFFERENT TRAINING SETS AND M FOR THE TOP-M FREQUENCIES METHOD

Time	Top 1	Top 2	Top 5	Top 10	Top 20
1am	0.979	0.987	0.994	0.995	0.995
3am	0.975	0.982	0.993	0.996	0.997
5am	0.963	0.976	0.991	0.996	0.993
7am	0.978	0.987	0.993	0.988	0.987
9am	0.978	0.983	0.993	0.996	0.997
11am	0.973	0.985	0.988	0.988	0.990
13pm	0.953	0.960	0.974	0.983	0.987
15pm	0.978	0.986	0.990	0.992	0.992
17pm	0.979	0.988	0.993	0.995	0.996
19pm	0.979	0.987	0.993	0.996	0.995
21pm	0.979	0.986	0.993	0.995	0.994
23pm	0.980	0.987	0.993	0.995	0.996
mean	0.974	0.983	0.991	0.993	0.993
std	0.008	0.008	0.006	0.004	0.004

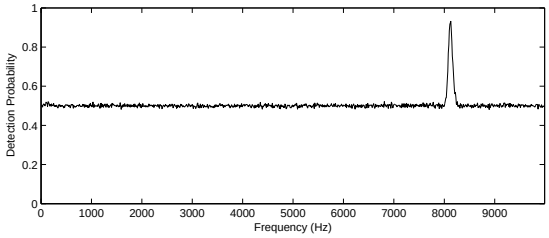
Figure 19 shows the detection probability of the Single Frequency method using the trace pair obtained around 7am as the training set. The top graph is the detection probability on the training set alone, and the bottom graph is the detection probability on all other 11 sets. In both graphs, the x axis is the frequency that is used to obtain the amplitude distribution. As we see from the graph, the detection probability has a sharp spike in the [8000Hz, 8250Hz] range. This frequency range is very close to the highest packet rate through a 100Mbps bottleneck with 1500byte packets. This clearly demonstrates that aggregate with a 100Mbps bottleneck flow (group H_w) differs significantly in statistics with aggregate without the 100Mbps bottleneck flow (group H_0) in their spectra around 8000Hz, and this difference persists over the time. Hence we can use the difference extracted from the training set to classify other instances. On the other hand, the detection probability on other sets is lower than the detection probability on the training set, suggesting that the difference can vary over the time.

Table I shows how things change over the time. The first field is the time when the training set was gathered. The second field is the frequency F in the [7500Hz, 8500Hz] range that yields the highest detection probability on the training set. The third and fourth fields are the corresponding detection probabilities on the training set and on all other sets, respectively. The fifth and sixth field are the mean values of the amplitude at F for group H_w and H_0, respectively. The seventh field is the corresponding cut-off threshold calculated by the Single Frequency Method. The result shows that the statistics of the training set can vary over the time, which results in variation on the cut-off threshold and the detection probability. But such variation is small. For example, the frequency that yields the highest detection probability on the training set stays in a close range around 8129Hz, and the detection probability on all other sets has a mean of 71.6% and a standard deviation of 1%.

Figure 20 shows the detection probability with the Top Frequency Method, again using the trace pair obtained around 7am as the training set. We use 100Hz wide frequency bands, and the x value represents the lower bound of the frequency band. To obtain a point (x, y) in the graph, we first get the distributions of the highest amplitude in the frequency

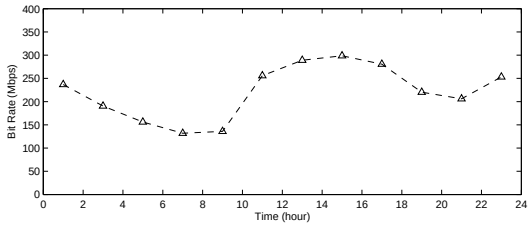


(a) detection probability on the training set

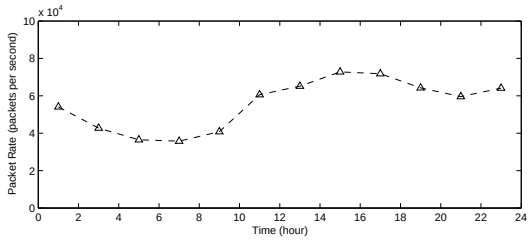


(b) average detection probability on all other sets

Fig. 22. Detection Probability of the All Frequencies Method with 100Mbps TCP bottleneck Traffic



(a) Traffic Volume in Bit Rate (Mbps)



(b) Traffic Volume in Packet Rate

Fig. 23. Aggregate Traffic with a 10Mbps TCP Bottleneck Flow

TABLE IV
VARIATION WITH DIFFERENT TRAINING SETS FOR THE ALL FREQUENCIES METHOD

Time	Freq	Pd Training	Pd All
1am	8130	0.965	0.924
3am	8120	0.987	0.935
5am	8120	0.993	0.928
7am	8130	0.995	0.932
9am	8110	0.982	0.917
11am	8120	0.942	0.923
13pm	8120	0.968	0.905
15pm	8120	0.928	0.930
17pm	8110	0.955	0.912
19pm	8130	0.958	0.927
21pm	8120	0.965	0.935
23pm	8120	0.973	0.929
mean	8121	0.968	0.925
std	7	0.020	0.009

band [x, x+100Hz] for both group H_0 and H_w using the training set, and then derive the the cut-off threshold based on the estimated (μ_0, σ_0) and (μ_w, σ_w). We then calculate the detection probability y by evaluating the detection method on the training set or on all other 11 sets.

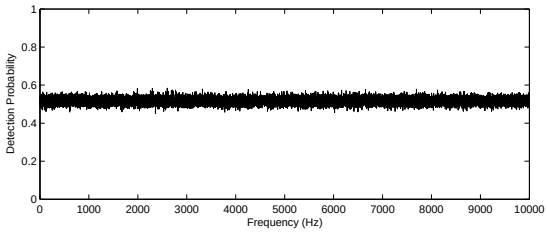
The result shows the detection probability can reach almost 100% if we use the frequency band in the [8000Hz, 8250Hz] range. This is significantly better than the Single Frequency method. It validates that the intuition that the highest amplitude in a frequency band can capture the difference between group H_0 and H_w better than the amplitude in a single frequency, and thus can yield better detection probability.

Table II summarizes the changes as we use different training sets. The meaning of each field is almost the same as in Table I, except that the second field refers to the lower bound of the 100Hz wide frequency band that yields the highest detection probability on the training set and the lower bound is in the [7500Hz, 8500Hz] range. Similar to the result for Single Frequency method in Table I, there are some changes for using different training sets with the Top Frequency Method, but the change is fairly small. For example, the detection probability on other sets has a mean of 97.3%, and a standard deviation of 0.8%.

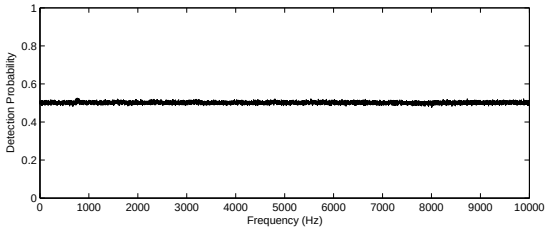
Figure 21 shows the detection probability with the Top 20 Frequencies Method. Again we use the 7am training set and 100Hz wide frequency bands. Like the result for the Top Frequency Method, the detection probability can reach almost 100% on both the training set and all other sets if we use frequency bands in the [8000Hz, 8250Hz] range. In addition, for other frequency bands, the detection probability on the training set is almost 20% higher than the detection probability on all other sets which stays flat near 50%. This suggests that the statistical difference extracted from the training set over these frequency bands is local to the training set, and does not persist over all other sets.

Table III shows the detection probability on other sets when we use top 1, 2, 5, 10, 20 frequencies in the frequency band [8070Hz, 8170Hz] and train with different training sets. The results indicate that increasing the number of top amplitudes can improve the detection probability, but the improvement is fairly small.

Figure 22 shows the result with the All Frequencies Method using the 7am training set and 10Hz wide frequency bands. We use a smaller band here because the All Frequencies Method uses all amplitudes in the frequency band. When the band is too wide, the covariance matrix of the joint distribution becomes singular and the All Frequencies Method can not classify the input trace. The result is very similar to the Top-20 frequency Method 21, except that the highest detection probability on all other sets can only reach 93%. Again we see the detection probability for the training set is almost about 20% higher than on other sets for frequency bands not in the [8000Hz, 8250Hz] range. This suggests that the statistical difference extracted from the training set over these frequency

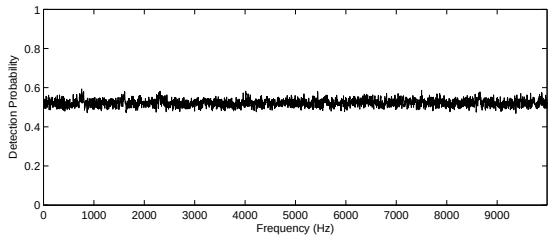


(a) detection probability on the training set

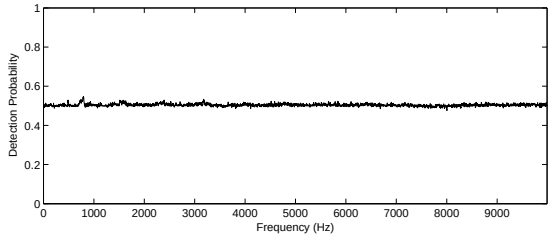


(b) average detection probability on all other sets

Fig. 24. Detection Probability of the Single Frequency Method with 10Mbps TCP bottleneck Traffi c



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 25. Detection Probability of the Top Frequency Method with 10Mbps TCP bottleneck Traffi c

TABLE V
VARIATION WITH DIFFERENT TRAINING SETS FOR THE SINGLE FREQUENCY METHOD

Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	793	0.573	0.526	11.212	11.011	11.602
3am	817	0.580	0.495	10.816	10.512	11.177
5am	805	0.582	0.511	10.736	10.449	10.774
7am	764	0.567	0.512	10.672	10.310	10.427
9am	771	0.567	0.520	10.810	10.670	11.389
11am	753	0.568	0.511	11.232	10.866	10.633
13pm	840	0.500	0.500	10.795	10.980	-5.085
15pm	764	0.567	0.514	11.014	10.789	11.303
17pm	809	0.558	0.509	11.013	10.983	12.047
19pm	761	0.563	0.510	11.042	10.779	11.213
21pm	848	0.500	0.500	10.595	10.797	3.729
23pm	813	0.593	0.500	11.439	12.087	23.568
mean	795	0.560	0.509	10.948	10.853	10.231
std	32	0.009	0.009	0.255	0.447	6.500

TABLE VI
VARIATION WITH DIFFERENT TRAINING SETS FOR THE TOP FREQUENCY METHOD

Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	793	0.612	0.530	12.860	12.669	12.826
3am	783	0.617	0.528	12.569	12.393	12.491
5am	783	0.623	0.527	12.406	12.163	12.258
7am	758	0.593	0.530	12.301	12.169	12.210
9am	750	0.592	0.529	12.549	12.405	12.486
11am	763	0.557	0.513	12.825	12.769	13.001
13pm	770	0.567	0.528	12.594	12.496	12.476
15pm	823	0.575	0.519	12.701	12.611	12.594
17pm	803	0.570	0.501	12.846	12.711	12.872
19pm	788	0.585	0.538	12.615	12.484	12.505
21pm	840	0.557	0.500	12.357	12.411	11.471
23pm	753	0.582	0.529	12.607	12.458	12.559
mean	784	0.586	0.523	12.603	12.478	12.479
std	28	0.023	0.012	0.186	0.192	0.394

bands does not persist across other trace sets.

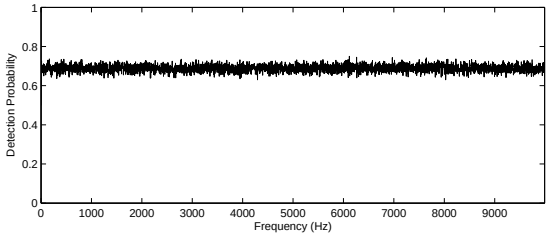
Table IV shows the variation on detection probability as we use different training sets. The meaning of each field is almost the same as in Table II, except that the frequency bands are 10Hz wide here. We see there is not much variation on the detection probabilities on both the training set and on all other sets as we use different training sets. Their mean values are 96.8% and 92.5% with standard deviation of 2% and 0.9%, respectively.

B. Experiment II: Detecting 10Mbps TCP Bottleneck Traffic

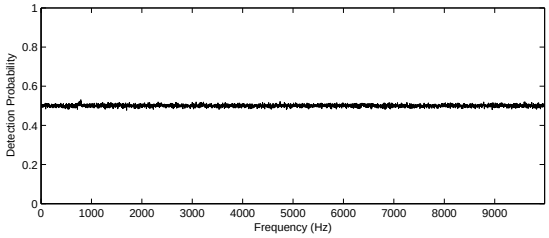
In this scenario, the bottleneck flow is an Iperf TCP flow through a known 10Mbps bottleneck link from ISI to USC. Figure 23 shows aggregate traffic volume in terms of bit rate and packet rate. Again we use the average value for the 5-minute long trace with the Iperf flow to represent the traffic volume in the corresponding 2 hour interval. The figure shows that the traffic reaches the lowest (around 131Mbps or 35.7K packets per second) in the interval from 6am to 8am, and the highest (around 298Mbps or 72.7K packet per second) in the interval from 14pm to 16pm. The throughput of the Iperf TCP flow is about 9Mbps or 750 packets per second.

Figure 24 shows the detection probability of the Single Frequency method using the 7am training set. Compared with the result for detecting 100Mbps TCP bottleneck traffic, the detection probability is very low for both the training set and other sets. The spike around 800Hz frequency associated with the 10Mbps TCP bottleneck traffic is barely noticeable. The reason is that the packet rate of the 10Mbps bottleneck flow is only about 1/10 of the packet rate for the 100Mbps bottleneck flow, and its presence does not bring much change to the overall aggregate spectrum. In other words, there is little statistical difference between group H_w which contains the 10Mbps TCP bottleneck flow and group H_0 which contains no such bottleneck flow.

Table V shows the variation over different training sets. The meaning of each field is almost the same as in Table I, except that the frequency F in the second field is selected from the [750Hz, 850Hz] range. The table shows there is only small difference on the average amplitude values between group H_w and H_0. For some training sets, like the 13pm training set, the statistics of group H_w and H_0 (e.g., mean value and standard

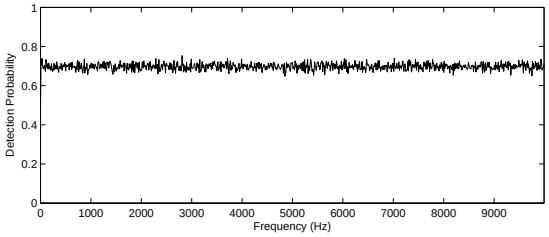


(a) detection probability on the training set

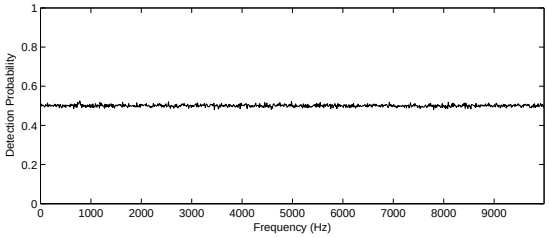


(b) average detection probability on all other sets

Fig. 26. Detection Probability of the Top-20 Frequencies Method with 10Mbps TCP bottleneck Traffi c



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 27. Detection Probability of the ALL Frequencies Method with 10Mbps TCP bottleneck Traffi c

TABLE VII
VARIATION WITH DIFFERENT TRAINING SETS AND M FOR THE TOP-M FREQUENCIES METHOD

Time	Top 1	Top 2	Top 5	Top 10	Top 20
1am	0.533	0.533	0.513	0.506	0.500
3am	0.528	0.532	0.536	0.531	0.520
5am	0.523	0.523	0.522	0.532	0.525
7am	0.534	0.536	0.516	0.520	0.514
9am	0.522	0.521	0.507	0.513	0.503
11am	0.541	0.531	0.532	0.518	0.512
13pm	0.534	0.530	0.515	0.525	0.515
15pm	0.525	0.530	0.523	0.537	0.519
17pm	0.533	0.529	0.531	0.519	0.517
19pm	0.531	0.536	0.518	0.511	0.509
21pm	0.528	0.514	0.493	0.497	0.504
23pm	0.533	0.531	0.536	0.522	0.512
mean	0.530	0.529	0.520	0.519	0.513
std	0.005	0.006	0.013	0.011	0.007

TABLE VIII
VARIATION WITH DIFFERENT TRAINING SETS FOR THE ALL FREQUENCIES METHOD

Time	Freq	Pd Training	Pd All
m	780	0.718	0.518
3am	770	0.722	0.507
5am	790	0.717	0.520
7am	770	0.722	0.515
9am	810	0.728	0.492
11am	820	0.725	0.491
13pm	800	0.718	0.501
15pm	750	0.715	0.499
17pm	750	0.725	0.496
19pm	840	0.720	0.507
21pm	790	0.723	0.500
23pm	790	0.733	0.508
mean	788	0.722	0.505
std	27	0.005	0.010

deviation) is so close that the Single Frequency Method fails completely by producing a negative cut-off threshold, basically classifying all instances as having 10Mbps TCP bottleneck traffic.

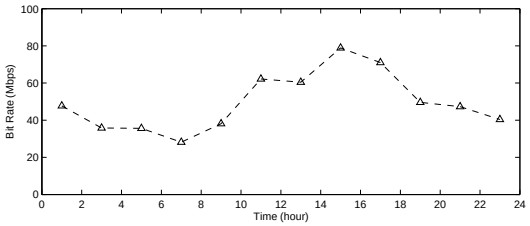
The result for the Top Frequency Method using the 7am training set is shown in Figure 25. The frequency bands here are 10Hz wide. A point (x, y) in the graph represents the detection probability using the distribution of the top frequency in the band [x, x+10Hz]. It shows only slight improvement compared with the Single Frequency Method. The highest detection probability on the training set can reach 59.2%, versus 56.7% for the Single Frequency Method.

Table VI shows the changes as we use different training sets. The meaning of each field is almost the same as in Table I, except that the second field refers to the lower bound of the frequency band that yields the highest detection probability on the training set and the lower bound is in the [750Hz, 850Hz] range. As we can see, there is only small difference on the average amplitude values between group H_w and H_0 across all different training sets. The detection probability stays fairly low no matter which training set we use.

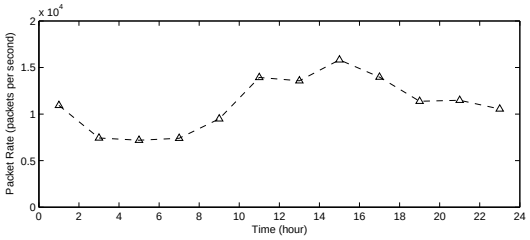
Figure 26 shows the detection probability with the Top 20 Frequencies Method using the 7am training set and 10Hz wide frequency bands. Similar to the result for detecting 100Mbps TCP Bottleneck traffic, the detection probability on the training set is almost 20% higher than the detection probability for all other sets, while the latter is almost the same as the Top Frequency Method. This again suggests that statistical difference extracted from the training set is local to the training set, and does not persist over the time.

Table VII shows the detection probability on all other sets when we use top 1, 2, 5, 10, 20 frequencies in the frequency band of [783Hz, 793Hz] and train with different training sets. The results indicate that increasing the number of top amplitudes does not improve much the detection probability, and in some case it even reduces the detection probability. But the change is bounded to a very close range (within 4%).

The result for the All Frequencies Method is shown in Figure 27. The training set was gathered at 7am, and the frequency bands here are 10Hz wide. The result is very similar to the result for the Top-20 Frequencies Method. Table VIII shows that there is only small variation on the detection

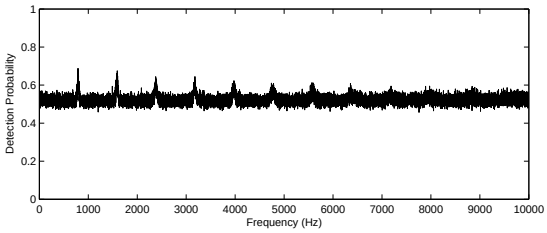


(a) Traffi c Volume in Bit Rate (Mbps)

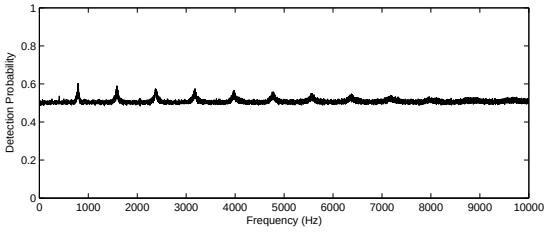


(b) Traffi c Volume in Packet Rate

Fig. 28. Aggregate Traffi c with a 10Mbps TCP Bottleneck Flow and Low Background Load



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 29. Detection Probability of the Single Frequency Method with 10Mbps TCP bottleneck Traffi c and low background load

probability as we use different training sets.

C. Experiment III: Detecting 10Mbps TCP Bottleneck Traffic with Low Background Load

In this scenario, we used an Iperf TCP flow to saturate the same 10Mbps bottleneck link, but the background traffic volume was significantly lower than the previous case, as we collected the trace at the start of a regular school semester. Figure 28 shows aggregate traffic volume in terms of bit rate and packet rate. The traffic reaches the lowest (around 28Mbps or 7.4K packets per second) in the interval from 6am to 8am, and the highest (around 78Mbps or 15.8K packets per second) in the interval from 14pm to 16pm. The throughput of the Iperf TCP flow is about 9Mbps or 750 packets per second.

Figure 29 shows the detection probability of the Single Frequency method with the 7am training set. We see the detection probabilities for both the training set and other sets are better than the previous experiment (Figure 24). The detection probability has spikes around 789Hz and its multiples (harmonics). This demonstrates that in this scenario aggregate with the 10Mbps TCP bottleneck flow shows noticeable statistical difference with aggregate without such flow in their spectra around 789Hz and its multiples.

Table IX shows the variation as we use different training sets. The meaning of all fields is the same as in Table V.We can see the average amplitude values for both group H_0 and H_w are generally much lower than their counterparts in the previous experiment (Table V), because the traffic volume is about 1/5 to 1/3 of the traffic load there. The gap between the average amplitude values for group H_0 and H_w is also wider than the previous scenario, making it easier to detect the 10Mbps bottleneck flow.

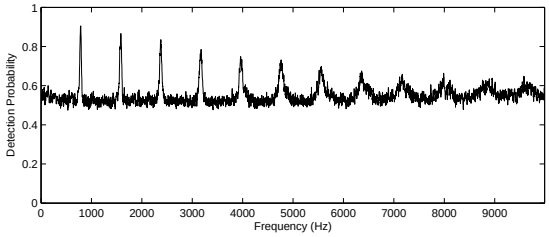
The result for the Top Frequency Method using the 7am training set is shown in Figure 30. The frequency bands here are 10Hz wide. The detection probability on the training set can reach almost 90%, while thedetection probability on other sets can reach 70%, both significantly higher than the Single Frequency Method. The 20% difference between detection probability on training set and other sets suggests that there is a significant mismatch between the statistics of the training set and other sets.

TABLE IX
VARIATION WITH DIFFERENT TRAINING SETS FOR THE SINGLE FREQUENCY METHOD

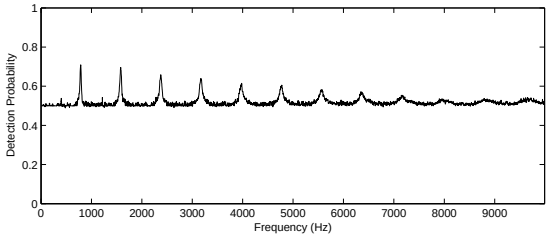
Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	785	0.668	0.598	9.740	8.903	9.429
3am	793	0.703	0.591	9.712	8.448	9.144
5am	794	0.705	0.589	9.450	8.452	9.164
7am	789	0.685	0.599	9.657	8.650	9.308
9am	789	0.677	0.601	9.675	8.767	9.339
11am	782	0.615	0.582	9.765	9.190	9.653
13pm	785	0.618	0.602	9.933	9.365	10.004
15pm	791	0.642	0.613	10.187	9.624	10.139
17pm	812	0.758	0.531	11.236	9.374	10.486
19pm	797	0.648	0.600	9.674	8.884	9.231
21pm	815	0.617	0.528	9.731	9.019	9.981
23pm	770	0.597	0.540	9.315	8.938	9.634
mean	792	0.661	0.581	9.840	8.968	9.626
std	12	0.047	0.030	0.490	0.369	0.437

Table X shows the changes as we use different training sets. Now we see the detection probability on other sets can differ by as much as 19%. This implies that in this scenario, the statistical difference between group H_0 and H_w can change significantly over the time, and it is important to use the proper training set. It is our future work to model the variation of the statistics over the time and load level, and design parametric detection methods that consider such variation for better detection probability.

Figure 31 shows the detection probability with the Top 20 Frequencies Method using the 7am training set and 10Hz wide frequency bands. Compared with the result for Top Frequency Method, the most noticeable difference here is that using top 20 Frequencies yields higher detection probability on the

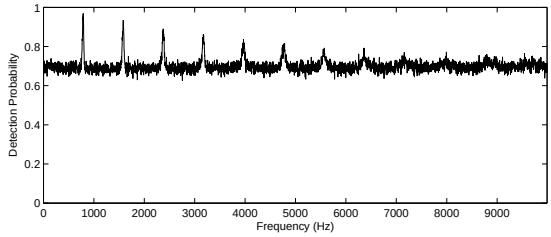


(a) detection probability on the training set

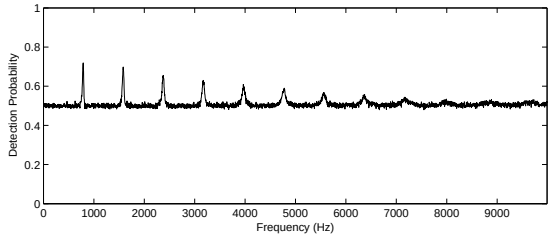


(b) average detection probability on all other sets

Fig. 30. Detection Probability of the Top Frequency Method with 10Mbps TCP bottleneck Traffic and low background load



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 31. Detection Probability of the Top-20 Frequencies Method with 10Mbps TCP bottleneck Traffic and low background load

TABLE X
VARIATION WITH DIFFERENT TRAINING SETS FOR THE TOP FREQUENCY METHOD

Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	788	0.842	0.737	11.481	10.688	11.084
3am	788	0.912	0.695	11.283	10.212	10.758
5am	788	0.890	0.692	11.208	10.227	10.735
7am	788	0.905	0.709	11.323	10.345	10.839
9am	783	0.862	0.727	11.402	10.579	10.993
11am	785	0.785	0.711	11.645	11.065	11.352
13pm	788	0.752	0.707	11.623	11.110	11.368
15pm	785	0.778	0.653	11.868	11.255	11.560
17pm	803	0.807	0.565	12.212	11.038	11.597
19pm	788	0.742	0.747	11.286	10.739	11.058
21pm	805	0.678	0.596	11.337	10.813	11.235
23pm	750	0.695	0.559	11.202	10.754	10.978
mean	787	0.804	0.675	11.489	10.736	11.130
std	13	0.079	0.066	0.303	0.348	0.294

TABLE XI
VARIATION WITH DIFFERENT TRAINING SETS AND M FOR THE TOP-M FREQUENCIES METHOD

Time	Top 1	Top 2	Top 5	Top 10	Top 20
1am	0.734	0.748	0.758	0.756	0.734
3am	0.692	0.696	0.696	0.691	0.685
5am	0.692	0.695	0.697	0.695	0.692
7am	0.708	0.718	0.719	0.719	0.719
9am	0.730	0.744	0.748	0.747	0.737
11am	0.715	0.720	0.725	0.716	0.696
13pm	0.705	0.716	0.724	0.711	0.694
15pm	0.651	0.649	0.662	0.655	0.648
17pm	0.702	0.709	0.715	0.715	0.693
19pm	0.745	0.754	0.764	0.751	0.727
21pm	0.752	0.755	0.746	0.725	0.678
23pm	0.745	0.748	0.742	0.724	0.675
mean	0.714	0.721	0.725	0.717	0.698
std	0.029	0.031	0.030	0.028	0.027

training set across all frequency bands other than those around 789Hz and its multiples. This again suggests that statistical difference extracted from the training set over these frequency bands is local to the training set, and does not persist over the time.

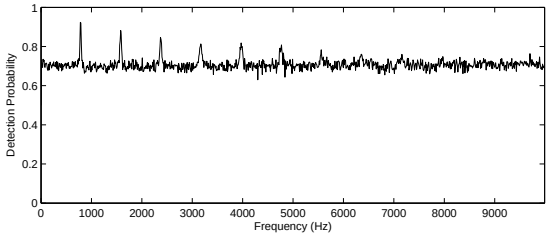
Table XI shows the detection probability on all other sets when we use Top 1, 2, 5, 10, 20 Frequencies in the frequency band of [788Hz, 798Hz] and train with different training sets. The result shows that increasing the number of top frequencies does not improve much the detection probability, and in some case it even reduces the detection probability. It suggests that the top frequency has captured the most important difference between group H_0 and H_w, and utilizing more frequencies does not yield significant gains.

The result for the All Frequencies Method using the 7am training set is shown in Figure 32. The frequency bands width here are still 10Hz wide. It is very similar to the result with the Top-20 Frequencies Method, although the highest detection probabilities on both the training set and other sets are slightly lower.

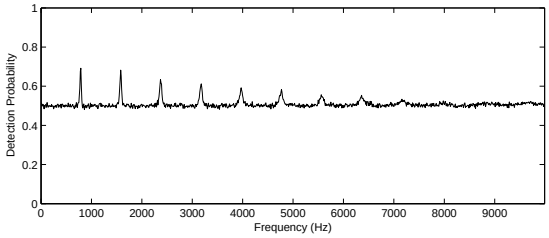
Table XII shows the changes on detection probability as we use different training sets. Like the result for the Top-Frequency Method, we now see the detection probability varies significantly across time, implying that parametric detection methods may be needed to consider the variation of the statistics over the time and load level.

D. Experiment IV: Detecting 10Mbps UDP Bottleneck Traffic with Low Background Load

In this scenario, we used an Iperf UDP flow to saturate a known 10Mbps bottleneck link. The UDP packet length is set to 1500 bytes. The traffic load level is comparable to the previous scenario. Figure 33 shows aggregate traffic volume in terms of bit rate. The traffic reaches the lowest (around 24.7Mbps or 7.2K packets per second) in the interval from 6am to 8am, and the highest (around 72.3Mbps or 15.9K packets per second) in the interval from 12pm to 14pm. The throughput of the Iperf UDP flow is about 9.6Mbps or 800 packets per second.

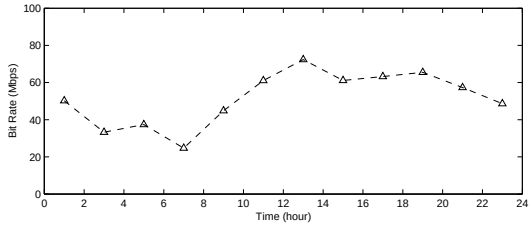


(a) detection probability on the training set

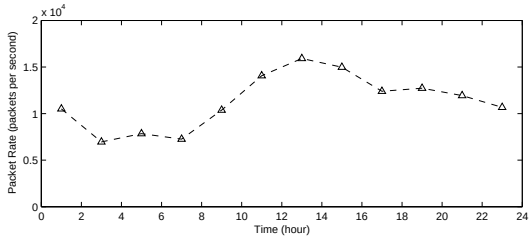


(b) average detection probability on all other sets

Fig. 32. Detection Probability of the All Frequencies Method with 10Mbps TCP bottleneck Traffic and low background load



(a) Traffic Volume in Bit Rate (Mbps)



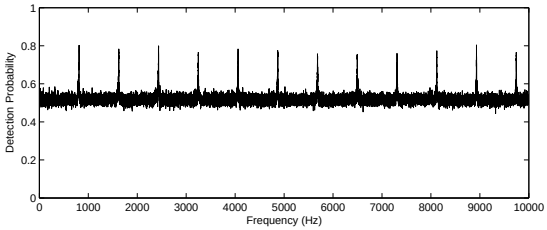
(b) Traffic Volume in Packet Rate

Fig. 33. Aggregate Traffic with a 10Mbps UDP Bottleneck Flow and Low Background Load

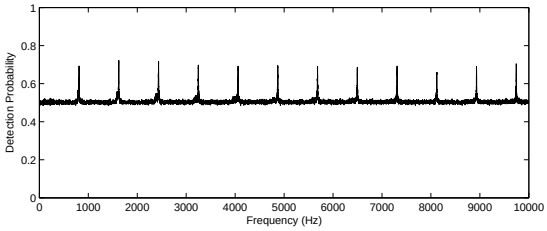
TABLE XII
VARIATION WITH DIFFERENT TRAINING SETS FOR THE ALL FREQUENCIES METHOD

Time	Freq	Pd Training	Pd All
1am	790	0.875	0.721
3am	780	0.932	0.646
5am	790	0.913	0.663
7am	780	0.923	0.677
9am	790	0.887	0.707
11am	790	0.813	0.677
13pm	790	0.827	0.688
15pm	780	0.840	0.621
17pm	810	0.827	0.535
19pm	790	0.797	0.686
21pm	810	0.770	0.546
23pm	750	0.780	0.537
mean	788	0.849	0.642
std	15	0.056	0.067

Figure 34 shows the detection probability of the Single Frequency method using the 7am training set. We see the detection probabilities for both the training set and other sets are better than the result for detecting 10Mbps TCP flow with comparable traffic load (Figure 29). The detection probability has spikes around 811Hz and its multiples (harmonics). 811Hz is slightly higher than 789Hz in the previous scenario, and closer to the highest packet rate through a 10Mbps bottleneck with 1500byte packets. The main reason behind the better detection probability is that the Iperf UDP flow fully saturates the bottleneck with a constant packet rate, while a TCP flow will adjust the packet transmission based on acknowledgments. As a result, the packet transmission of the UDP flow appears more regular than the TCP flow, resulting in stronger energy at a slightly higher fundamental frequency. The stinger amplitude will widen the difference between group H_0 and H_w, yielding better detection probability. Also we see much higher detection probability on the harmonics than the TCP case, since the UDP flow is more regular and its spectrum has stronger energy in harmonics.



(a) detection probability on the training set

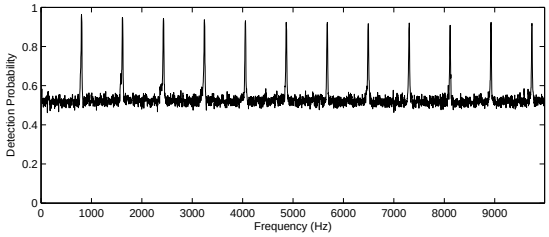


(b) average detection probability on all other sets

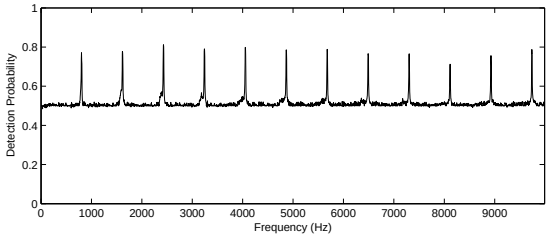
Fig. 34. Detection Probability of the Single Frequency Method with 10Mbps UDP bottleneck Traffic and low background load

TABLE XIII
VARIATION WITH DIFFERENT TRAINING SETS FOR THE SINGLE FREQUENCY METHOD

Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	810	0.725	0.689	10.581	9.062	9.769
3am	812	0.843	0.686	10.724	8.407	9.552
5am	811	0.848	0.694	10.802	8.592	9.687
7am	811	0.802	0.692	10.530	8.426	9.505
9am	811	0.810	0.724	11.001	8.979	9.958
11am	811	0.683	0.743	10.606	9.602	10.372
13pm	811	0.715	0.732	10.771	9.459	10.100
15pm	812	0.762	0.734	11.161	9.364	10.225
17pm	811	0.797	0.725	10.945	8.965	9.956
19pm	812	0.740	0.721	10.691	9.199	10.014
21pm	797	0.698	0.582	10.295	9.169	9.721
23pm	805	0.670	0.625	9.997	9.133	9.800
mean	810	0.758	0.696	10.675	9.030	9.888
std	4	0.062	0.048	0.314	0.385	0.264

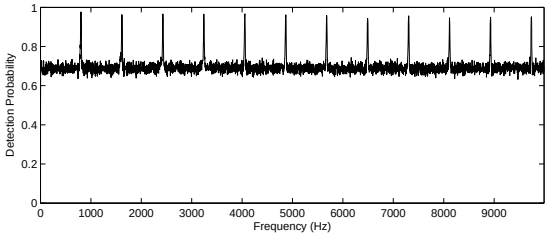


(a) detection probability on the training set

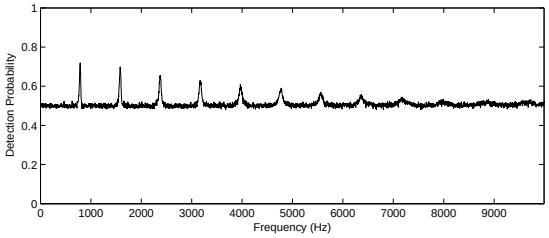


(b) average detection probability on all other sets

Fig. 35. Detection Probability of the Top Frequency Method with 10Mbps UDP bottleneck Traffi c and low background load



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 36. Detection Probability of the Top-20 Frequencies Method with 10Mbps UDP bottleneck Traffi c and low background load

TABLE XIV
VARIATION WITH DIFFERENT TRAINING SETS FOR THE TOP FREQUENCY METHOD

Time	Freq	Pd Training	Pd All	Mean H_w	Mean H_0	Threshold
1am	800	0.920	0.814	11.808	10.711	11.249
3am	805	0.980	0.796	12.013	10.174	11.150
5am	805	0.995	0.791	12.106	10.150	11.127
7am	803	0.963	0.771	11.904	10.255	10.987
9am	808	0.962	0.826	12.289	10.801	11.516
11am	805	0.828	0.833	12.088	11.290	11.697
13pm	803	0.880	0.824	12.217	11.256	11.731
15pm	805	0.918	0.807	12.318	11.197	11.772
17pm	805	0.962	0.836	12.317	10.828	11.557
19pm	805	0.880	0.846	12.091	10.983	11.533
21pm	790	0.888	0.625	11.715	10.851	11.284
23pm	800	0.872	0.821	11.676	10.783	11.227
mean	803	0.921	0.799	12.045	10.773	11.402
std	5	0.052	0.059	0.227	0.399	0.264

TABLE XV
VARIATION WITH DIFFERENT TRAINING SETS AND M FOR THE TOP-M FREQUENCIES METHOD

1am	0.823	0.826	0.841	0.841	0.837
3am	0.797	0.802	0.820	0.825	0.823
5am	0.794	0.802	0.814	0.818	0.814
7am	0.756	0.751	0.756	0.756	0.752
9am	0.839	0.850	0.857	0.858	0.856
11am	0.835	0.847	0.862	0.857	0.826
13pm	0.802	0.821	0.833	0.826	0.812
15pm	0.808	0.820	0.818	0.815	0.808
17pm	0.837	0.850	0.864	0.862	0.859
19pm	0.847	0.856	0.865	0.862	0.850
21pm	0.833	0.834	0.836	0.827	0.802
23pm	0.854	0.860	0.858	0.860	0.840
mean	0.819	0.827	0.835	0.834	0.823
std	0.028	0.031	0.031	0.031	0.029

Table XIII shows the variation as we use different training sets. Like the result for detecting TCP bottleneck flow in IX, we see significant variation over the time, suggesting smarter detection methods are needed for better performance. But in general, no matter which training set we use, the detection probability is much better than the detection of TCP bottleneck flow under similar conditions.

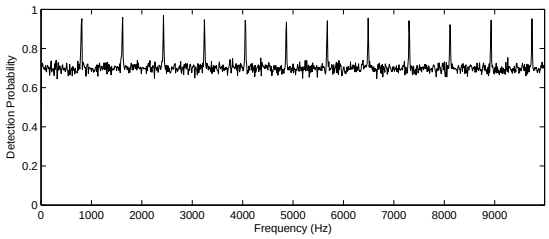
The result for the Top Frequency Method using the 7am training set is shown in Figure 35. The frequency bands here are 10Hz wide. Again we see better detection probability than the Single Frequency Method. It is also better than the TCP case (Figure 30), The detection probability on the training set can reach 96%, while the detection probability on other sets can reach 77% for the [803Hz, 813Hz] band. Again the 19% difference between detection probabilities on training set and other sets suggests that there is a significant mismatch between the statistics on the training set and other sets.

Table XIV shows the changes as we use different training sets. Like the result for detecting TCP bottleneck flow under similar environment X, we see significant variation of the detection probability on other sets, arguing for smarter detection algorithms that consider the variation of the statistics over the time and load level.

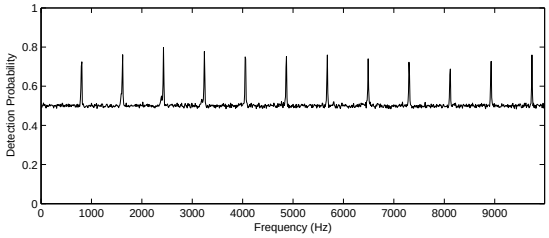
Figure 36 shows the detection probability with the Top 20 Frequencies Method using the 7am training set and 10Hz wide frequency band. Compared with the result for the Top Frequency Method, the only noticeable difference here is that using top 20 Frequencies yields higher detection probability on the training set on frequency bands other than those around 811Hz and its multiples. This again suggests that statistical difference extracted from the training set over these frequency bands is local to the training set, and does not persist over the time.

Table XV shows the detection probability on other sets when we use top 1, 2, 5, 10, 20 frequencies in the frequency band of [805Hz, 815Hz] and train with different training sets. The results indicate that increasing the number of top frequencies does not improve much the detection probability, and in some case it even reduces the detection probability.

The result for the All Frequencies Method using the 7am training set is shown in Figure 37. Again, it is very similar



(a) detection probability on the training set



(b) average detection probability on all other sets

Fig. 37. Detection Probability of the ALL Frequencies Method with 10Mbps UDP bottleneck Traffi c and low background load

TABLE XVI
VARIATION WITH DIFFERENT TRAINING SETS FOR THE ALL FREQUENCIES METHOD

Time	Freq	Pd Training	Pd All
1am	800	0.917	0.772
3am	810	0.962	0.759
5am	810	0.978	0.745
7am	810	0.952	0.724
9am	810	0.983	0.808
11am	810	0.828	0.792
13pm	810	0.902	0.797
15pm	810	0.930	0.788
17pm	810	0.958	0.809
19pm	810	0.892	0.794
21pm	790	0.887	0.602
23pm	800	0.890	0.761
mean	807	0.923	0.763
std	7	0.046	0.057

to the result of the Top-20 Frequencies Method, although the highest detection probabilities on both the training set and other sets are a bit lower. Table XVI shows the changes on detection probability as we use different training sets. Like the table for the Top-Frequency Method, we see the detection probability varies significantly across time, implying that parametric detection methods may be needed to consider the variation of the statistics over the time and load level.

E. Discussion

From the above results, we can see a clear relation between the detection probability and the ratio of the bottleneck flow volume and the background traffic volume (or signal-to-noise ratio). The first experiment scenario has the highest ratio ranging from 1/3.5 to 1/1.1 in terms of bit rate or 1/10 to 1/4.5 in terms of packet rate, and it has the highest detection probability, close to 100% with the Top Frequency Method. The second scenario has the lowest ratio ranging from 1/32 to 1/13 in terms of bit rate or 1/96 to 1/46 in terms of packet rate, and it has the lowest detection probability, below 55% even for the Top Frequency Method. The third scenario has a ratio about 1/8 to 1/2 in terms of bit rate or 1/20 to 1/9 in terms of packet rate, and its detection probability is in the middle, reaching 70% for the Top Frequency Method. This observation agrees with our intuition that the higher the ratio, the easier the detection.

Although training with different sets plays a limited role in the extreme scenarios where the signal-to-noise ratio is either very high or very low (the first and second experiment scenarios), it has a significant impact on the detection probability when the signal-to-noise ratio is in the middle (the third and fourth experiment scenarios). For example, the detection probability of the Top Frequency Method in the third scenario can vary by as much as 19% when we use different training sets. This implies that the statistical difference between group H_0 and H_w can vary significantly across time, and we need to model such variation further to design parametric detection methods that consider such variation.

Regarding the Protocol impact, we see that it is easier to detect the Iperf UDP bottleneck flow than the Iperf TCP bottleneck flow. The reason is that the Iperf UDP flow fully saturates the bottleneck with a constant packet rate, while a TCP flow will adjust the packet transmission rate according to network conditions. As a result, the Iperf UDP stream appears more regular than the TCP stream under similar conditions, yielding bigger difference when compared with background traffic.

Operated under the same conditions, the Single Frequency Method yields the lowest detection probability, while the Top Frequency Method performs significantly better as it considers the shift of the bottleneck signature in the spectrum across time. Both the Top-M Frequencies Method and All Frequencies Method improve the detection probability on the training set by considering more frequency information, but they do not produce significant gains on the detection probability on other sets, compared with the Top Frequency Method. This suggests that the top frequency captures most of the statistical difference that persists over the time. It is our future work to carry further investigation why these two multi-variate methods do not provide better performance and refine them for improvements.

VI. RELATED WORK

In recent years, a number of researchers have used spectral techniques to analyze network traffic. Hussain et al. apply spectral techniques to packet arrival time series to distinguish single-source and multi-source DDoS attacks [1], and more recently have extended this approach to attack re-identification [2]. But they mostly examine attack traffic in isolation of background traffic. Barford et al. use wavelets to analyze IP flow-level and SNMP information to detect Dos attacks and other network anomalies [4]. Cheng et al. also apply spectral analysis to separate normal TCP traffic which exhibit strong periodicities around its round-trip time from DOS attack traffic [7]. Magnaghi et al. propose a wavelet-based framework to proactively detect network misconfigurations, which utilizes the TCP retransmission timeout events during the opening phase of the TCP connection [16]. Partridge et al. apply the Lomb periodogram technique to retrieve periodicities

in wireless communication, including CBR traffic and the periodicity around the FTP round-trip time [5]. In addition, Partridge warns of the dangers of blind application of signal processing techniques in networking without careful analysis and knowledge of ground truth [8], [9].

Kim et al. apply wavelet denoising to improve the detection of shared congestion among different flows [17]. They use the cross-correlation between the one-way-delay experienced by packets in different flows for the detection, and then improve the performance by reducing the impact of random queuing behavior while preserving the behavior of highly congested link through wavelet denoising, since a highly congested link has a larger low frequency component in the spectrum of the one-way-delay sequence. The technique requires inserting active probe packets into the network to measure the one-way-delay. Another closely related work in detecting bottleneck traffic is in [6]. In this paper, Katabi et al. use packet inter-arrival times to infer the path characteristics such as bottleneck capacity, and bottleneck sharing among flows as the entropy of packet inter-arrival times is much lower for flows sharing the same bottleneck. A number of techniques have been used for estimating link capacity and available bandwidth [18], [19], [20], [21], [22], [23]. Besides scope, there are several key differences between these and ours. First, they need to isolate specific flows, whereas our techniques work on large aggregate traffic. Second, they need access to both ends of the path, whereas we assume a single observation point. Finally, most such techniques rely on active measurements, whereas ours are passive.

Network tomography typically uses a limited number of active or passive measurements (typically at network edges) to infer network performance parameters and topology. Examples include network and link parameter estimation [24], [25], topology inference [26], [27], [28] and traffic matrices [29], [30]. The input signal is typically packet delays, round-trip time, loss, etc., and tomography tries to infer network characteristics using correlation techniques such as maximum likelihood estimation and Bayesian inference. Unlike our techniques, in network tomography multiple observation points may be required and flows need to be separated from aggregate traffic.

As compared to other research, our approach is *passive* (i.e., does not require probing packets), can operate on *aggregate traffic* (so that component traffic flows do not have to be extracted), transform the data to a suitable *spectral domain representation* (rather than operating with time-domain information), and make use of *more rigorous statistical methods* (rather than relying on more qualitatively visual evidence).

VII. Conclusions

As the Internet has evolved to become an inseparable part of millions of people's daily life, it is important for us to better understand and diagnose it in a broad scale. Spectral techniques have been shown to a powerful tool to extract hidden patterns in many fields, and they are becoming widely used by the network research community to analyze Internet traffic to infer useful information ranging from network anomalies to protocol behavior.

In our work, we presented an experimental methodology for the application of spectral techniques to network problems, and used it to analyze the regularities imposed by bottleneck links. In addition to visual demonstrating the signature imposed by various bottleneck links and how it evolves as the bottleneck flow traverses the network, we proposed four detection algorithms based Bayes Maximum-likelihood Classifier to automatically detect the bottleneck signature embedded in aggregate traffic, and evaluated their performance using real-world Internet traces. Our results show that we can have fairly high detection probabilities in many cases.

In summary, we believe that spectral techniques provide a very promising approach to study the Internet traffic dynamics and the knowledge gained would make a vital contribution for the continuing success of the Internet. As future work we plan to strengthen our techniques by addressing the following aspects.

- Refine the detection algorithms by looking deep into their assumptions, such as the log-normal distribution assumption for the Single Frequency Method and the Top Frequency Method, and the multi-variate log-normal distribution assumption for the Top-M Frequencies Method and the All Frequencies Method.
- Investigate why the Top-M Frequencies Detection and the All Frequencies Method do not yield better performance than the Top Frequency Method. We see that they can improve significantly the detection probability on the training set, but not on the other sets. Understanding the reason behind this may help us find other heuristics for improving the detection probability.
- Study other multi-variate detection methods, like the one utilizing harmonics, as harmonics may convey additional information besides the fundamental frequency.
- Model the underlying processes that govern the generation of bottleneck traffic and how it is shaped by competing traffic, and use the model to design parametric detection algorithms that take traffic load and other time-varying factors into consideration. We see the performance of the non-parametric detection algorithms depends on the heavily on the signal-to-noise ratio, and in some cases it also depends on which training set we use. We hope to improve the detection probability by modeling the underlying processes and designing parametric detection algorithms that consider the variation across traffic load and time.
- Apply the detection methods in more diversified environment, including at different trace points, with different bottleneck locations, different types of traffic composition (e.g., single flow versus multiple flows), and different types of cross traffic, so that we can gain a more thorough understanding of how they perform and validate the findings from system modeling.
- Study and improve the performance of our detection methods in terms of efficiency, so that the detection can be done in real-time with high speed network traffic.
- Extend the techniques into a framework that can be applied to study other periodic traffic phenomena, such

as TCP windowing behavior and network anomalies. This will expand the applicability of our methodology and help gain insight of other network phenomena.

REFERENCES

[1] A. Hussain, J. Heidemann, and C. Papadopoulos, "A Framework for Classifying Denial of Service Attacks," in *Proceedings of the ACM SIGCOMM'2001*, Karlsruhe, Germany, August 2003.

[2] ——, "Identification of Repeated Attacks Using Network Traffic Forensics," under submission.

[3] A. Broido, E. Nemeth, and kc Claffy, "Spectroscopy of DNS Update Traffic," in *Proceedings of the ACM SIGMETRICS*, San Diego, CA, June 2003.

[4] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseilles, France, November 2002.

[5] C. Partridge, D. Cousins, A. Jackson, R. Krishnan, T. Saxena, and W. T. Strayer, "Using Signal Proceesing to Analyze Wireless data Traffic," in *Proceedings of ACM workshop on Wireless Security*, Atlanta, GA, Sept. 2002, pp. 67–76.

[6] D. Katabi and C. Blake, "Inferring Congestion Sharing and Path Characteristics from Packet Interarrival Times," MIT, Tech. Rep. LCS Technical Report, 2001.

[7] C.-M. Cheng, H. Kung, and K.-S. Tan, "Use of spectral analysis in defense against DoS attacks," in *Proceedings of the IEEE GLOBECOM*, Taipei, China, 2002.

[8] C. Partridge, "Frequency analysis of protocols," talk given at U. Minnesota.

[9] ——, "Internet signal processing: Next steps," talk given at the Workshop on Internet Signal Processing.

[10] G. Box, G. Jenkins, and G. Reinsel, *Time series analysis: forecasting and control*. Prentice-Hall, 1994.

[11] R. Bracewell, *The Fourier Transform and Its Applications*. McGraw-Hill, 1986.

[12] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an internet backbone," CAIDA, Tech. Rep., April 1998.

[13] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation," in *Proceedings of the ACM SIGMETRICS'98*, Madison, Wisconsin, USA, June 1998.

[14] "Iperf," http://dast.nlanr.net/Projects/Iperf/.

[15] "Endace DAG Network Monitoring Cards," http://www.endace.com/networkMCards.htm.

[16] A. Magnaghi, T. Hamada, and T. Katsuyama, "A Wavelet-Based Framework for Proactive Detection of Network Misconfigurations," in *Proceedings of ACM workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality*, Portland, Oregon, USA, August 2004.

[17] M. S. Kim, T. Kim, Y. Shin, S. S. Lam, and E. J. Powers, "A Wavelet-Based Approach to Detect Shared Congestion," in *Proceedings of the ACM SIGCOMM'2004*, Portland, Oregon, USA, August 2004.

[18] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*. ACM Press, 2003, pp. 39–44.

[19] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2002, pp. 295–308.

[20] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *INFOCOM '01: Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies.*, 2001, pp. 905–914.

[21] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," in *Proceedings of the 2000 IEEE Globecom Global Internet Symposium*, 2000.

[22] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth techniques," *IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling, 3rd Quarter, 2003*, 2003.

[23] A. B. Downey, "Using pathchar to estimate internet link characteristics," in *Proceedings of the ACM SIGCOMM Conference*. Cambridge, MA, USA: ACM, Aug. 1999, pp. 241–250. [Online]. Available: http://www.acm.org/sigcomm/sigcomm99/papers/session7-1.html

[24] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F. L. Presti, S. B. Moon, and V. Paxson, "The use of end-to-end multicast measurements for characterizing internal network behavior," *IEEE Communications Magazine*, vol. 38, no. 5, pp. 152–159, May 2000.

[25] N. Duffield, "Simple network performance tomography," in *ACM Internet Measurement Conference*. Miami Beach, FL, USA: ACM, 2003, pp. 210–215. [Online]. Available: http://doi.acm.org/10.1145/948205.948232

[26] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," *IEEE Transactions on Information Theory*, vol. 48, pp. 26–45, 2002. [Online]. Available: http://www.research.att.com/ duffield/pubs/DHLT01-TOIT.pdf

[27] M. Coates, R. Castro, and R. Nowak, "Maximum likelihood network topology identification from edge-based unicast measurements," in *Proceedings of the ACM SIGMETRICS*. Marina del Rey, CA, USA: ACM, 2002, pp. 11–20. [Online]. Available: http://doi.acm.org/10.1145/511334.511337

[28] M. Coates, M. Rabbat, and R. Nowak, "Merging logical toplogies using end-to-end measurements," in *Proceedings of the ACM Internet Measurement Conference*. Miami Beach, FL, USA: ACM, Oct. 2003, pp. 192–203.

[29] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *Proceedings of the ACM SIGCOMM Conference*. Pittsburgh, Pennsylvania, USA: ACM, Oct. 2002, pp. 161–174. [Online]. Available: http://doi.acm.org/10.1145/633025.633041

[30] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proceedings of the ACM SIGCOMM Conference*. Karlsruhe, Germany: ACM, Oct. 2003, pp. 301–312. [Online]. Available: http://doi.acm.org/10.1145/863955.863990