A non-intrusive, wavelet-based approach to detecting network performance problems

Polly Huang, Anja Feldmann, Walter Willinger

Abstract-The **main objective of this paper is to explore how much information about the characteristics of end-to-end network paths can be inferred from relying solely on passive packet-level traces of existing traffic collected from a single tap point in the network. To this end, we show that a number of structural properties of aggregate TCP/IP packet traces reveal themselves and can be compared across different time periods and across parts of the traffic destined to different subnets by exploiting the built-in scale-localization ability of wavelets. In turn, these structural properties and the resulting comparisons suggest the feasibility of new approaches for inferring and detecting qualitative aspects of network performance in a fashion that is similar to relying on active measurements, but without disturbing or biasing the metrics of interest. To showcase the feasibility, we developed** WIND, **a prototype tool for wavelet-based INference for Detecting network performance problems and illustrate its capabilities to detect anomalies in underlying network path conditions with two examples of passively measured packet traces from two different networking environments. We address and experiment with ways of validating the output of** WIND **and end with a discussion of the potential of full-fledged wavelet-based analysis (i.e., the ability to localize a signal in scale and time) for future measurement studies.**

Keywords- **Wavelets, Scale-Localization, Energy Function, Passive Measurements, Network Performance.**

I. INTRODUCTION

Active or passive network measurements have been essential ingredients for inferring or assessing performance-related problems within individual IP clouds or along end-to-end paths in the Internet. By injecting new traffic ("stimulus") into the network, active measurement methodologies can often elicit certain responses to that stimulus from individual network components or from the network. In turn, these responses are used to infer performance metrics such as bulk throughput [29], packet delay [10], [18], [32], [35], packet loss [32], [35], [37], bottleneck link bandwidth [10], [14], [25], [32], [35], link bandwidths along the end-to-end path [17], [24], [26], hop counts [18], or other path characteristics [31], [32], [35], [23], [9], [30]. Active measurements cover only those parts of the network that are being probed, do not generally deal with application performance, and ISPs are typically restricted to use them within their network. In addition, active measurements have the potential to add significantly to the existing network load, thereby perturbing the network, biasing the ensuing observations, and complicating the interpretation of much of the resulting analysis. Drawing a natural analogy to physics, Paxson [33] refers to these drawbacks as "Heisenberg" effects. In contrast, by focusing exclusively on existing network traffic, passive measurement techniques typically have the advantage of completely eliminating these ill-understood Heisenberg effects. Unfortunately they suffer from the drawback that we are often at a complete loss as to how to infer important performance metrics or network path characteristics from the non-intrusively recorded traffic from any given link in the network.

In this paper, we explore the feasibility of alleviating this drawback of passive measurement methodologies – without compromising their appealing advantages – by focusing on and exploiting wavelet-based analysis techniques to infer and extract certain end-to-end network path characteristics from IP packet traces recorded off a network link. To this end, we rely on a built-in scale-localization ability of wavelets which is ideally suited for detecting "hidden" but highly regular/irregular traffic patterns in measured traffic traces. In turn, for a given connection or set of connections, such patterns can be used to accurately infer the time scales associated with the dominant round-trip times (RTT) experienced by the packets traversing the network and allow for a qualitative assessment of how network performance can change along network paths over time due to, e.g., sudden load changes, congestion build-ups, or possible route changes. To il-

P. Huang is with the Computer Engineering and Networks Laboratory, ETH Zürich, Zürich, Switzerland. E-mail: huang@tik.ee.ethz.ch.
A. Feldmann is with the Computer Science Department, Universität des Saarlandes, Saarbrücken, Germany. E-mail: anja@cs.uni-sb.de.
W. Willinger is with AT&T Labs-Research, Florham Park, NJ. E-mail: walter@research.att.corn.

lustrate these "detective" capabilities, we have developed WIND, a fast, close-to on-line prototype tool for Wavelet-based INference for Detecting network performance problems, that currently relies on passively recorded packet-level traces of existing traffic from a given link within the network. WIND filters the measured traffic according to the dominant applications running between representative customers and the most popular destinations, processes the resulting packet traces by applying wavelet techniques, and uses a set of heuristics for detecting potential network performance problems by correlating the wavelet-based trace analysis results across time and across network paths. We are not aware of any "obvious" alternatives for obtaining the same sort of information; e.g., NetFlow data is too coarse-grained, SNMP gives link statistics available only within an AS, and IP-accounting is inadequate. Using two data sets of measured packet-level traces, we report on WIND's abilities to infer certain performance metrics and to use them for pointing out potential network performance problems, under quite different networking environments. One of the traces was collected from a 100 Mbps FDDI ring of a commercial ISP, the second resulted from recording the traffic on a T3 link connecting an industrial research lab to the rest of the Internet. We also address the issue of validating the output of WIND by computing additional fine-grained statistics such as round-trip time distributions and comparing their characteristics to those inferred from the wavelet-based methods.

The rest of the paper is structured as follows. In Section II we review related work. Next in Section III, we describe our methodology of using the scale-localization ability of wavelets to extract, from a time series of traffic rate information, the round-trip times experienced by the individual packets; we illustrate our approach with a number of simple but informative examples. Section IV gives an overview of the WIND tool that runs on top of a passive monitor and implements this particular "fingerprinting" capability. We report in Section V on how WIND fares in practice when faced with measured packet traces from two very different networking environments and conclude in Section VI by commenting on how this work may open up new opportunities for exploiting passive as well as active network measurements.

II. Related work

Despite the existing vast literature on active and passive measurements, we are only aware of a few studies that are directly relevant for our purpose; i.e., share the same commitment to passive measurements of existing traffic and put similar emphasis on the need for novel analysis methods for extracting relevant performance-related information from these measurements. Valuable resources that provide general information about various measurement methodologies are [33], [8], [13], [12]. In terms of our focus in this paper on passive measurements of existing network traffic, the work by Seshan et al. [38] on SPAND (short for shared passive network performance discovery) is close in spirit to our present study and discusses the benefits of working with shared, passive measurements from a collection of hosts rather than with active measurements from a single host when trying to determine network path characteristics. In this context, another relevant study is by Balakrishnan et al. [6], whose empirical investigations revealed significant stability of network paths characteristics (i.e., available bandwidth) over time and across hosts that are close to each other; in other words, hosts which share portions of a network path tend to obtain similar amounts of throughput. Stability of the amount of available bandwidth over time has also been reported by Paxson [34], where it is noted that the opposite (i.e., wide variations) is true for other network path characteristics such as RTT and packet loss. As for passive network measurement studies in general, [11], [27], [36], [15], [40] are prime examples that illustrate the wealth of information that can be extracted from passively measuring traffic at a single tap point in the network and using commonly available time or frequency domain-based analysis techniques. More recently, [19] introduced a new traffic analysis method that exploits the full range of the time- and scale-localization abilities of wavelets and suggests new opportunities in traffic analysis: novel capabilities for detecting "interesting" features or patterns in time and scale in the underlying packet trace and correlating them to known networking mechanisms that reveal themselves in terms of specific network path properties. In fact, the present paper is motivated to a large degree by the empirical observation reported in [19] and expands on the wavelet-based traffic analysis techniques outlined in [19].

As far as active measurement studies are concerned, it is instructive to compare our approach with the multicast-based method for inferring network-internal characteristics pursued within the MINC project [41], [3]. While MINC relies on active measurements and is concerned with a quantitative statistical analysis of the resulting measurements for characterizing internal network behavior, our focus is on passive measurements and on a qualitative assessment of the measured data that relates statistically significant features directly to network-specific mechanisms. In the context of studies that use both active and passive measurements, some of the most significant work is due to Paxson [31], [34], [32], [35] (see also [8], [9] for WWW-specific work), who has previously noted a dire need for more adequate techniques for analyzing both active and passive network measurements in ways that can reveal hidden networking-related features and do so effectively [35, p. 277]. The present paper addresses his very concerns and identifies wavelet-based methods as prime candidates for replacing the traditional and generally inadequate time- or frequency domain-based analysis methodologies. It will be interesting to observe how in the years to come, these proposed wavelet-based techniques will impact the use of active and passive measurement strategies, the analysis and interpretation of the ensuing data, and the development of next-generation active and/or passive measurement tools.

III. DETECTING NETWORK PERFORMANCE: METHODOLOGY

The feasibility of analyzing passive measurements and detecting "fingerprints" of network path characteristics has been suggested in [19]. The analysis fully exploits the wavelets' abilities to uncover and identify highly irregular/regular traffic patterns that are well-localized in scale. By relating some of these patterns directly to known network-specific mechanisms, they enable us to infer certain characteristics of network paths, which in turn can be used to qualitatively assess network performance as experienced by the packets traversing these paths. The purpose of this section is to illustrate with a set of simple examples how a wavelet-based analysis of synthetically generated time series or of traffic rate processes obtained from a set of *ns-2* [5] simulations can uncover "hidden" structures in the data; e.g., the dominant RTT behavior associated with the packets that make up the measured traffic.

A. Wavelet-based scaling analysis of aggregate traffic

A.1 The energy function

Consider a time series $X_{0,k}$, $k = 0, 1, 2, \ldots$, at the finest level of resolution 2^{-n}, and interpret X_0 as a traffic rate process (i.e., the number of packets per 10 ms, for example). We coarsen X_0 by averaging (with a slightly unusual normalization factor) over non-overlapping blocks of size two

$$X_{1,k} = \frac{1}{\sqrt{2}}\left(X_{0,2k} + X_{0,2k+1}\right) \qquad (1)$$

and obtain a new time series X_1, a coarser resolution picture of the original series X_0. Taking differences rather than averages in Eqn. (1) results in quantities known as ***details***

$$d_{1,k} = \frac{1}{\sqrt{2}}\left(X_{0,2k} - X_{0,2k+1}\right). \qquad (2)$$

Note that we can reconstruct the original time series X_0 from the coarser representation X_1 by simply adding in the details d_1; i.e., $X_0 = 2^{-1/2}(X_1 + d_1)$. We can iterate this process (i.e., write X_1 as the sum of yet a coarser version X_2 of X_0 and the details d_2, etc.) for as many scales as are present in the original time series and can write $X_0 = 2^{-n/2}X_n + 2^{-n/2}d_n + \ldots + 2^{-1/2}d_1$. We refer to the collection of details $d_{j,k}$ as ***the discrete (Haar) wavelet coefficients***, ***they*** make up what is commonly referred to as the ***discrete (Haar) wavelet transform,*** and they may be calculated iteratively using Eqns. (1-2)[1].

To investigate the scaling properties of a given time series, we concentrate in this paper on the scale-localization property of wavelets; that is, we use the discrete (Haar) wavelet transform of the time series to examine the behavior of a given statistics of the wavelet coefficients at each resolution level or scale, as a function of scale. In particular, we focus here on a statistics known as the ***energy function*** E_j and defined by

$$E_j = \frac{1}{N_j}\sum_k |d_{j,k}|^2, \quad j = 1, 2, \cdots, n,$$

where N_j is the number of coefficients at scale j. We interpret E_j as the average energy contained at scale j of the trace and examine how this quantity changes as we move from finer to coarser scales. To this end, we plot $\log(E_j)$ as a function of scale j, from finest (i.e., $j = 1$) to coarsest scales, and use the resulting ***energy function plot*** to determine qualitative aspects of the scaling behavior of the underlying time series [19]. In all of the energy function plots in this paper, the scale j is on the bottom axis and the corresponding actual time (in seconds) is plotted on the top axis for reference.

A.2 The energy function of some synthetically generated time series

The following set of toy examples is intended to illustrate how a qualitative wavelet-based energy function analysis is capable of revealing certain structural properties exhibited by an underlying time series.

Periodic time series: Consider the periodic time series X of length 1024 that consists of the pattern 0, 0, 1, 1, 1, 1, 0, 0, repeated 128 times Manual inspection of Eqn. (2) shows that all wavelet coefficients $d_{j,k}$ ***are*** zero except those for scales $j = 2$ and $j = 3$, reflecting the presence of periodic patterns at scales $j = 1$, and $j = 4$ and beyond, and of irregular patterns at scales $j = 2$ and $j = 3$. This andsimilar examples suggest that highly regular or periodic structures in time series reveal themselves

[1]We restrict this exposition to the use of Haar wavelets. The scaling analysis results presented below are either based on Haar wavelets or more general wavelets (e.g., compactly supported Daubechies wavelets [16]).

in terms of small wavelet coefficients, and the energy function appears to capture such behavior adequately. We subsequently refer to such low values of the energy function as "dips." Note that using traditional Fourier analysis methods would give similar insight into the structure of X as does the energy function method considered here.

White noise time series: We consider two time series Y 1 and Y2. Y 1 is constructed by letting the interarrival times of individual events be i.i.d. exponentially distributed with mean 10 and counting the number of events that occur in each time unit. When calculating the corresponding energy function, Figure 1 (top plot, solid line) shows that the values are roughly constant across all scales, a trademark of white noise processes [1].

We construct Y2 just like $Y1$, but incorporate a simple form of local periodicity structure as follows. Let the interarrival times of individual events be i.i.d. exponentially distributed with mean 15, and count the number of events that occur in each time unit. Next pick a quarter of the events at random and assume that each of the randomly picked event triggers an additional event exactly 8 time units later; repeat this procedure with another set of a quarter of randomly picked event, but this time, the newly triggered event happen exactly 18 time units later. The resulting energy function is depicted as a dotted line in the top plot of Figure 1. Comparing the two different energy function plots, we notice that the local periodicity structure imposed onto Y2 results in wavelet coefficients at scale 4 that are smaller in magnitude than those corresponding to Y 1 at that same scale, reflecting the behavior observed earlier for time series X. This decrease in the absolute magnitude of the wavelet coefficients of Y2 around scale 4 manifests itself more clearly when examining the corresponding energy function which shows a dip at scale 4 that extends to some coarser scales due to the presence of the second local periodicity of 18 time units.

Note that our main reason for experimenting with time series of the type Y2 is the presence of a more or less periodic component in measured Internet traffic that is created by the TCP protocol. TCP is using ack clocking to space the sending of packets across some window of time; that is, if the time it takes between sending a packet and receiving the corresponding ack is the RTT, then a sender that transmits a packets at time t_i will send another packet at time $t_i + RTT$. Of course not all acks trigger the sending of a new packet, nor do acks arrive precisely in lock step, separated by RTT. Indeed, the behavior of TCP is, among other aspects, controlled by its congestion control mechanisms (slow start and congestion avoidance) and the receiver window [39]. What Y2 tries to capture is that the superposition of many TCP connections creates more

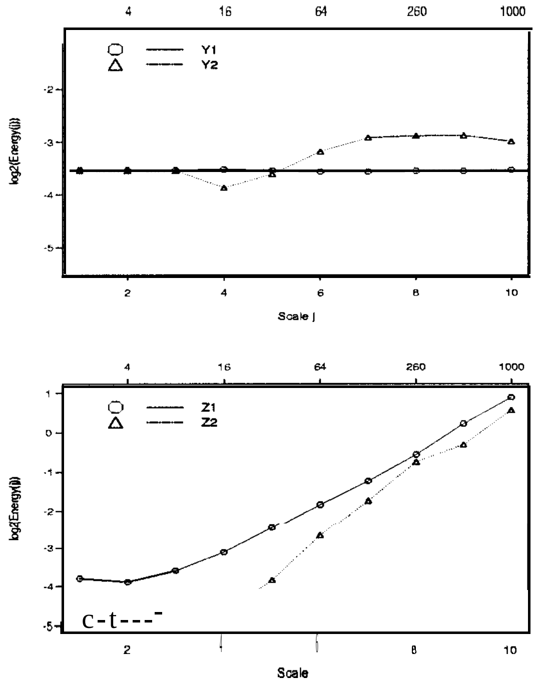


Fig. 1. Energy function plots of time series Y 1 and Y 2 (top), and $Z1$ and 22 (bottom).

or less pronounced local periodicities based on the dependency on RTT, and that the resulting regular traffic patterns can be inferred from the resulting energy function plot. Note also that conventional frequency-domain analysis is unlikely to succeed in detecting and identifying the sort of local periodicities considered in example Y 2.

Self-similar time series: Moving beyond the white noise examples $Y1$ and Y2, we construct next two self-similar time series $Z1$ and 22. The constructions of $Z1$ and 22 are identical to those of $Y1$ and Y2, respectively, with the exception that the interarrival times are now i.i.d. Pareto-distributed with $\alpha = 1.2$ and a mean of roughly 10 [15]. The results are shown in the bottom plot in Figure 1. As expected (e.g., see [1]), the wavelet coefficients of $Z1$ are scale-invariant in the sense that the resulting energy function plot shows a linear relationship between $\log E_j$ and j, with a slope between 0.5 and 1.0. As far as 22 is concerned, the explicitly imposed local periodicities show up in very much the same way as they did for the time series Y2. As before, conventional frequency-based methods are not likely to provide a similar assessment of the presence of local periodicities in time series 22.

A.3 The energy function of some NS simulation traces

To move one step closer to applying the above wavelet-based analysis to actual traffic traces, we examine in this subsection the scaling properties of traffic traces collected in a network simulation environment. The simulation engine used throughout this study is ns-2 [5], and in the following, we are especially interested in trying to understand the influence of round-trip times and network congestion on the wavelet decomposition of the resulting traces and on the properties of the corresponding energy function plots.

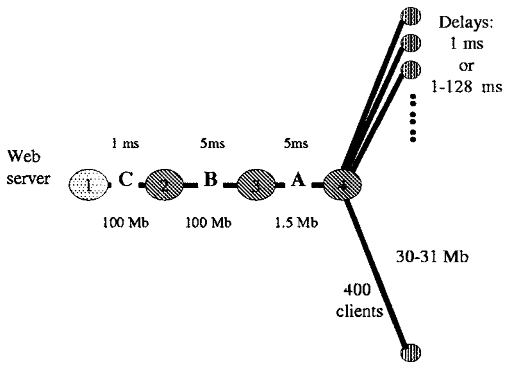


Fig. 2. Network configuration for ns-2 simulations.

All of the simulation experiments reported in this paper involve the simple network topology depicted in Figure 2. Our toy network consists of a single server (node 1), a set of high-speed clients (nodes $5 - 405$), and a number of links. The clients are connected to the network via $30 - 31$ Mbps links, the server has a 100 Mbps connection to the network, and two additional links (A, B) comprise the rest of the network. Link A is used to limit the capacity of the network to a value of 1.5 Mbps and represents, in fact, the bottleneck link in our simulation setup. The link B bandwidth is set at 100 Mbps, and we use this link to observe the dynamics of the traffic before it traverses the bottleneck link A. The number of buffer spaces available for buffering packets [2] in node 3 for link A is limited to 50 while all other nodes are allocated plenty of buffer space. All simulated routers use "drop-tail" buffer management. The delays of links A and B are 5 ms, while the delay of link C is 1 ms. To study the influence of round-trip time (RTT) variability we choose two different network environments: one in which the delay of the links to the clients is a constant equal to 1 ms, and another in which the delay of the links to the clients is uniformly distributed between 1 and 128 ms. We rely on the *Web* workload model considered in [19] [3] that is very similar to SURGE [7].

No congestion, single vs. variable RTT: To ensure minimal losses and a single RTT, the first simulated scenario involves just 50 Web clients, and the delays along all links to the clients are assumed to be 1 ms. In this situation, we expect to observe pronounced local periodicities on the order of 24 ms (complete RTT), 29 ms (complete RTT + transmission delay for 1000 bytes across a 1.5 Mbit link), 12 ms (the time it takes from observing a data packet from the server to a client on link B to seeing the corresponding ack from the client to the server on link B), and 2 ms (the time it takes from observing an ack from a client to the server on link B to seeing the next data packet from the server to the client on link B). Checking the resulting energy function plot in Figure 3 (solid line), we observe a dip that stretches from about 16 ms to about 128 ms; that is, the range of time scales corresponding to local periodicities of 12 and 29 and to a lesser degree to multiples of 29. The dip is strongest around 32 ms, the next-largest time scale corresponding to the RTT. As expected, when increasing the delay of the links to the clients to, e.g., 128 ms, the main dip from scale 32 ms to a corresponding larger value, which in this case is 512 ms (not shown here).

[2]*ns-2* allocates buffer space in terms of number of packets and not number of bytes.

[3]We consider the following probability distribution functions for the session attributes; number of pages per session: constant 300, inter-page time: Pareto with mean 50 and shape 2, number of objects per page: Pareto with mean 4 and shape 1.2, inter-object time: Pareto with mean 0.5 and shape 1.5, and object size: Pareto with mean 12 (in KB) and shape 1.2.

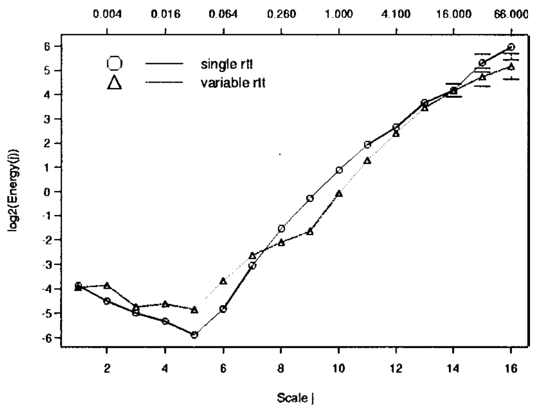


Fig. 3. Energy function plots for traces from simulations with negligible congestion; comparing single vs. variable RTT.

To introduce some amount of variability in the RTT, we next change the 1 ms delays along the links to the clients. More precisely, we allow the link delays to be sampled from a uniform distribution between 1 and 128 ms. As a result, the RTTs will increase from the dominant 24 ms value and take on values in the range of 24 to about 276 ms. The effect of this change from a single RTT value to a more heterogeneous RTT behavior can be seen in Figure 3 (dotted line), where the dip now extends into some of the larger scales (on the order of 512 ms).

Congestion, single vs. variable RTT: Returning to the single RTT environment from before, we rerun the simulation with 300 Web clients instead of the 50 considered before, with the intention to run the system into congestion, thereby causing losses, which in turn forces TCP to

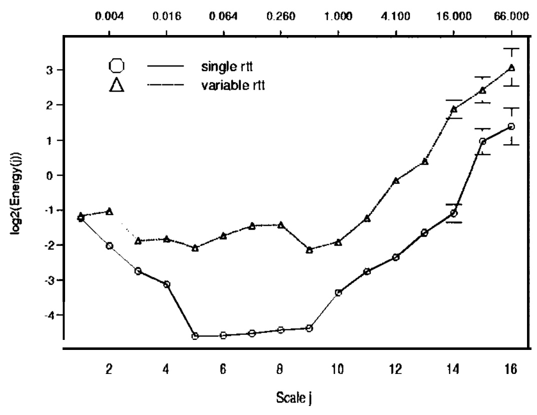


Fig. 4. Energy function plots for traces from simulations with significant level of congestion; comparing single vs. variable RTT.

retransmit. Thus, congestion can be expected to introduce yet another regular pattern, with a period that reflects the length of the timeout. The resulting energy function plot is shown in Figure 4 (solid line) and demonstrates convincingly that the dip has shifted to larger time scales (on the order of 512 ms to about 2 sec) and that because of congestion, the once dominant RTT value of 24 ms has become much more variable due to queuing delays in the buffers and TCP retransmission mechanism.

Rerunning the variable RTT environments for the 300 Web clients, the resulting energy function plot is depicted as a dotted line in Figure 4. As expected, congestion on top of variable RTTs ensures an even wider dip in the energy function, with additional mechanisms contributing to this widening (e.g., congested bottleneck link).

B. Distributional analysis of packet-specific properties

In the case of the *ns-2* simulation traces, we have used the knowledge of the simulation setup to infer the dominant round-trip time characteristics in the network. However, the available packet-level data allow us to actually compute detailed round-trip time (RTT) and retransmission timeout (RTO) statistics using the following heuristic. Each packet is either an acknowledgment packet (ack) or a data packet. If a packet is a data packet with sequence number s, we check if it is the first packet with this sequence number; if it is not, the packet is a retransmitted packet, and we can compute a sample of the RTO by taking the difference between the timestamp of the current packet and the time we encountered the most recent packet with that same sequence number. If the packet is an ack packet with sequence number s, we check if it is the first ack packet; if it is not the first ack packet, we ignore it, otherwise we compute the difference between the timestamp of the ack packet and the timestamp of the corresponding data packet. This is a sample of the RTT.[4]

From these computed sample RTTs and RTOs, we compute the (empirical) ***log-density functions*** (i.e., conventional density function on logarithmic x-axis) and the (empirical) ***complementary cumulative distribution function (CCDF),*** where the latter can be obtained by integrating the former from x to ∞, $x \geq 0$. Figure 5 shows the resulting log-density functions (top) and CCDFs (bottom) for the measured RTT values; similar plots can be obtained for the RTO values but are not shown here. Note that the overall behavior of these functions is as expected[5]: the RTT (and RTO) values gradually shift to the right (i.e., increase in magnitude) as we move from no congestion/single RTT to no congestion/variable RTT to congestion/single RTT and finally to congestion/variable RTT. In particular, we note that while increased congestion does not dramatically impact the RTT CCDFs (with the exception that multiple RTT values appear due to increased queueing), it has a significant impact on the RTO CCDFs. In fact, while we hardly experience any timeouts in the lightly loaded scenario with 50 Web clients (if there are any, the RTO is negligible), under congestion, the retransmission timeouts can become significant and take on relatively large values.

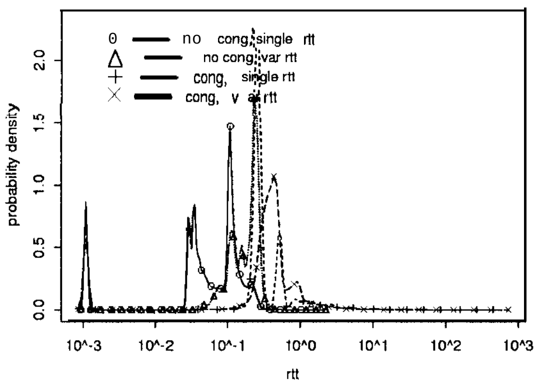
C. From energy plots to RTT behavior and vice versa

For the set of ***m-2*** simulation experiments performed in Section III-A.3, we can now address the question whether or not wavelet-based traffic analysis techniques – in the case at hand the energy function plots – can be used reliably to detect and/or infer "fingerprints" of network path properties when relying solely on passive network measurements. While the feasibility of such an approach to detecting network path conditions was originally advocated and illustrated in [19], its practical relevance has remained uncertain, if not doubtful.

More specifically, by combining the results from our energy function experiments with the observations concerning the behaviors of RTT (and RTO), we would like to know if changes in the energy function plots are good indicators of changes in network path conditions (and vice versa). For a qualitative answer to this question, we con-

[4]Note that in Section 3 below, we use a slightly more elaborate way to compute the RTT and RTO samples. Referring to the time interval between a data-a&/data-data pair as an RTT/RTO sample is a very rough classification of periodic elements in TCP transmission. Refinements such as Kam's algorithm are a subject of further investigation.

[5]While one might expect to see a single mode at the value of the RTT in the log-density function corresponding to the no congestion/single RTT simulation, note that even in this case queueing will occur at the bottleneck link; given a speed of the bottleneck link of 1.5 Mbit, it is not surprising that for TCP connections with large amounts of data to transfer, the RTT increases to roughly 100 ms.



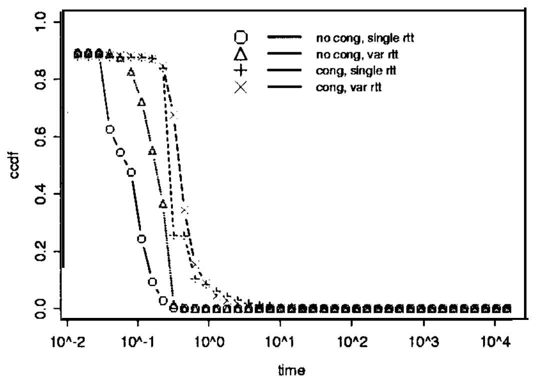


Fig. 5. Log-density functions (top) and CCDFs (bottom) for RTT values for the different simulations.

sult Figures 4 and 5 and observe a relative clean-cut, though by no means perfect, one-to-one correspondence between the shape of the energy functions resulting from the four simulation experiments (i.e., no congestion/single RTT, no congestion/variable RTT, congestion/single RTT, and congestion/variable RTT) and the location on the x-axis of the corresponding RTT (and RTO) CCDFs: the wider the dip, the more likely it is to find larger RTT (and RTO) values. Put differently, RTT (and RTO) values that are on the average smaller and less variable result in energy function plots for the underlying packet traces with more pronounced dips, reflecting a higher likelihood for giving rise to regular traffic patterns over a narrow range of time scales. Note that this qualitative assessment applies for the CCDFs as well as for the log-density plots, but for the latter, it is less obvious how to account for the different modes that are present in the different log-densities. Because the CCDFs are obtained by integrating the corresponding densities, the former tend to obscure the presence of some of the modes, are generally much smoother, and are therefore more amenable for making comparisons with other CCDFs. Given this encouraging result about being able to infer – at least qualitatively, and for that only in the context of a toy simulation model – network path conditions such as RTT behavior and congestion scenarios from energy function plots of the underlying traffic, the challenge remains whether the same sort of qualitative "fingerprinting" capabilities can be used and will work when applied in real-life networks. In the remainder of this paper, we illustrate how to overcome this challenge by developing a tool that mimics that hybrid quantitative/qualitative approach outlined in this section.

IV AN INFERENCE TOOL FOR DETECTING NETWORK PERFORMANCE PROBLEMS

To implement the methodology outlined in the previous section, we developed a prototype tool called WIND, short for Wavelet-based INference for Detecting network performance problems. We use the term network performance problems to refer to load changes, increased network congestion, or route changes that impact the round-trip times that, in turn, can significantly affect the performance of applications using the network. In its present version, WIND is non-intrusive; it only records existing traffic and does not generate new traffic. The main objective of WIND is to enable an on-line almost real time wavelet-based analysis of measured traffic destined for the busiest subnets. WIND operates from observation period to observation period (the default value for the duration of an observation period is 10 minutes); at the end of each period, it generates various statistics. Among these are energy function plots for detecting network problems and various counters to notice and react to changes in server popularity, user behavior, and/or protocol and application usage. For example, WIND can re-rank the subnets based on popularity and in this way change the subnets that it focuses the analysis on for the next observation period.

To support this functionality, WIND has an efficient core which streamlines packet header extraction, traffic volume accounting, and wavelet coefficient computation in real time. A supplementary component of WIND handles calculation of round-trip time (RTT) and retransmission timeout (RTO) associated with those packets that come from or go to the busiest subnets. Being computationally very intensive, this component is operated only off-line and the results are used for validation purpose. WIND comes with a rich set of traffic filtering options. For example, we can filter the measured traffic by dominant services and applications running between (a set of) representative customers and the most popular destination(s). The filtering options offer flexibility to "zoom in" on interesting observation periods, subnets, or applications.

A. Streamlined per-packet processing

The core processing performed by WIND consists of three phases: low-level packet capturing, traffic volume accounting, and wavelet coefficient computing. The processing time of the three phases combined is constant or close-to constant; that is, WIND'S core that promises wavelet-based inference for detecting network performance problems can be conducted in real time.

In the first phase, WIND collects packet-level traces from a tap point in the network, by adopting the `libpcap` [21] implementation. Each packet captured by `libpcap` is passed on to the packet header processing part of `t cpdump` [22] which is also adopted by WIND. The packet header information is subsequently streamlined into the second and third phases. Optionally, the packets captured by `1 i bpcap` can be duplicated, and written into files in binary format for off-line processing.

In the volume accounting phase, WIND classifies packets into subnets based on configurable netmasks of the form 255.255.0.0. For each of the source and destination IP addresses, WIND looks up a hash table based on the subnet address and updates the packet volume statistics. The kind of statistics collected are described in Section IV-D below. The computation cost is bounded by the fraction of the number of subnets to the number of bins in the hash table, which enables one to trade off memory for constant speed. Only the source and destination subnet addresses and packet timestamps are passed to the third phase.

In the wavelet analysis phase, WIND processes only packets traversing the top X busiest subnets (the default value for X is 20) where information about the top X busiest subnets is obtained through sorting traffic volumes in the previous observation period(s). Given that WIND needs the information about which subnets to consider before it can compute the wavelet coefficients, the computation of the wavelet coefficients is typically done based on the volume statistics of the previous time period or some predetermined set of subnets if WIND is operated on-line. If WIND is operated off-line it can consider any set of subnets. For each packet traversing one of the X busiest subnets, **WIND** checks the timestamp and generates the associated packet rate process entries (i.e., time series entries indicating number of packets per 10 msec). If the current packet arrives within the same 10 msec bin as the previous packet, a packet counter is incremented; otherwise, the new time series entry is pushed to the wavelet coefficient computation and the packet counter is reset. The implementation of the wavelet coefficient computation uses the recursive filter-bank-based pyramidal algorithm (e.g., see [43]) which has a lower computational cost than that of a fast Fourier transform and can be implemented on-line and executed in real time (our default analyzing wavelet is the d4 wavelet [16]). In our case, the average computation cost is bounded by the number of levels or time scales considered during the recursive wavelet coefficient computation and is pre-determined by the duration of observation period and the time unit used in generating the packet rate processes. At the end of each time period WIND computes the energy function for each subnet from the wavelet coefficients.

B. Additional feature

Another feature of WIND concerns the inference, calculation, and processing of round-trip times (RTT) and retransmission timeouts (RTO). In contrast to the core processing which aims at generating real-time network reports, results from this RTT/RTO feature are mainly used to validate potential performance problems identified by the wavelet-based inference methodology. RTT and RTO calculations are computationally intensive and are performed off-line. To perform the RTT/RTO estimation, WIND typically operates on the `t cpdump` trace generated by the real-time processing. For each packet that travels to or from one of the X busiest subnets, WIND extracts the exact source address port, destination address port, and sequence number. If the packet carries data, WIND looks up a hash table indexed by the unique source and destination address port pair and sequence number. If no existing entry is found (i.e., a new data packet), the timestamp of the data packet is recorded in a new entry into the hash table. If the packet is an ack and WIND finds a corresponding data packet in the hash table, the time difference is taken as an RTT sample. Upon obtaining an RTT sample, WIND updates the various RTT-related statistics for the corresponding source and destination subnets. If the packet carries data and a corresponding entry is found in the hash table (i.e., a retransmission), WIND takes the difference of the two timestamps as an RTO sample and updates the RTO-related statistics for the corresponding subnets.

Occasionally, there is more than one ack per data packet. This can happen either when the next data packet is lost and the subsequent data packets trigger the TCP receiver to send duplicate acks, or when the original ack packet gets stuck somewhere in the network for too long and causes the retransmission timer to expire (also known as the false retransmission problem.) The retransmitted data then triggers the same ack to be sent. To account for these situations, the corresponding RTT samples are obtained by subtracting the data packet's timestamp from the times the duplicate acks are observed. Similarly, there can be occasionally two or more retransmissions per data packet, due to actual data packet drops or false retransmission. The resulting RTO samples are all accounted for in the resulting traffic statistics. Upon obtaining an RTT sample, WIND updates the average RTT for the corresponding subnet and increments the empirical histograms of RTT's respectively. The process is similar for RTO samples. The histogram bins are sized proportionally to the time scales used in energy function analysis.

C. Filtering options

WIND offers a rich set of traffic filtering options. Some of the options are inherited from `tcpdump`. For instance, when applying the filters `net 128.3` and `port 80`, WIND's `tcpdump` part streamlines port 80 packets coming in to or out from network 128.3, which may constitute a set of representative customers or popular destinations. Other options are add-ons to enable flexible "zooming" functionalities. For example, traffic can be further classified into subnets with user-specified aggregation levels. As part of the scaling and RTT/RTO analysis, data can be collected for the busiest X subnets observed in the previous observation period or according to a pre-specified list of interesting subnets. In the former case, WIND's data collection can adapt to changes in service and/or server popularity. When interesting scaling phenomena are identified in an interesting subnet, one can "zoom" in for more details by adding a filter for the subnet and specify a finer-grained level of aggregation. WIND also analyzes traffic based on its direction. For data presented in this paper, we consider bi-directional traffic.

D. Periodic network reports

WIND generates periodic reports on traffic volume, scaling behavior, and RTT/RTO estimation. Within each observation period and given some aggregation of IP addresses, WIND keeps track of the cumulative (i.e., since the beginning of the measurement period) number of IP packets and bytes, as well as of the number of packets and bytes within the current observation period; it does so on a per-packet-type (i.e., IP, TCP, UDP, and others) basis. At the end of the entire measurement period, WIND generates a file containing subnet addresses of the top interesting subnets according to the cumulative traffic volume, which can be re-used for further off-line processing [28] [44].

Within each observation period, and given the top interesting subnets, WIND keeps track of the relevant packet rate processes, i.e., the time series of packet arrival per 10 msec. These time series are then streamlined into the wavelet coefficient computation as described in Section IV-A. At the end of each period, WIND computes the per-scale energies in the traffic traces to the different subnet. These per-scale energies are used for generating the corresponding energy function plots, the basic tool for our wavelet-based approach to detecting network performance problems. The per-period reports contain mainly the energy function information, but can also contain the entire wavelet transforms and/or the time series data, useful for applying or experimenting with other possible wavelet-based analysis techniques. Lastly, these periodic reports can also provide RTT and RTO information in terms of averages and histograms on a per-subnet basis.

V. AN ILLUSTRATIVE EXAMPLE

In this section we explore the capabilities of WIND to detect network performance problems on measured data from two different networking environments. We use one of the data sets to illustrate the main features of WIND, including a number of heuristics for extracting "interesting" events from the periodic reports provided by our tool. Here, an "interesting" event typically refers to a time period where

- increased network load in some part of the network increased congestion significantly;
- a route change resulted in significantly increased round-trip times for some part of the traffic;
- server or network outages had a severe impact on the performance of the network application.

Note that the performance problem does not necessarily affect all traffic. Indeed, it is likely that it will only impact the traffic along some network paths to a few subnets. The second data set is used to showcase WIND's flexibility to work in different networking environments, to point out shortcomings of the current implementation of our tool, and to suggest possible improvements.

Our idea for identifying "interesting" events is to identify a typical or expected behavior (performance) of traffic rate processes. While it is hard to do this in general due to the huge variability in the Internet, we expect packets between the same subnets to traverse the same networks and therefore experience the same service from the network. Therefore it should be much easier to define the typical behavior of the packets going from, e.g., representatitive customer, to a subnet specified by an IP address and a network mask. While some packet will experience better service from the network than others one would expect that the differences are relatively small as long as there are no significant change in either the routes, the load, and/or the congestion levels along the network path that the packets take. Yet, these are exactly the kind of events that we want to detect. Therefore we propose to monitor the performance as characterized by the energy function and look for significant changes that happen from one time period to the next. (To add stability we propose to use an expected energy function rather than the one from the last time period.) In addition one can use the energy functions to observe the relative performance that traffic to a subnet might experience.

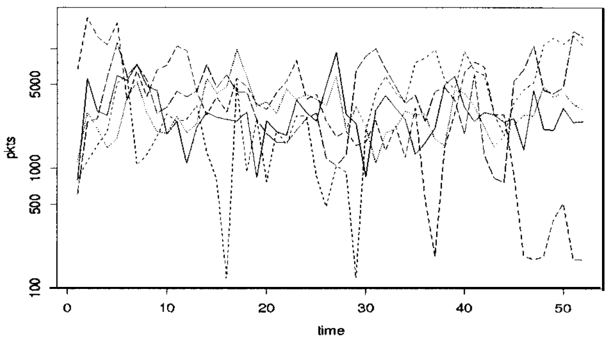
A. Description of available data sets

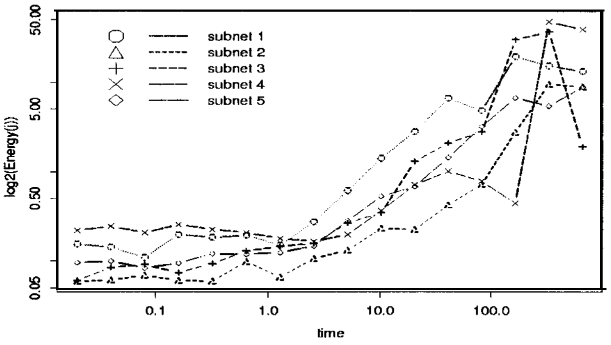
Throughout this paper we use the following high-quality data sets. The trace DIAL was gathered from an FDDI ring (with typical utilization levels of 5-10%) that connects about 420 modems to the rest of the Internet. Although we collect every packet seen on the FDDI ring, we restrict our analysis of DIAL to (bidirectional) ***modern user traffic only.*** Collection of the trace started on Wed. June 28, 2000 at 15:57 and ended Sun. July 2 at 00:40 and consists of a total of 360,000,000 packets or more than 14.5 GB of compressed data. We refer to the subset of the data from Fri. 14:10 to 23:30 as DIAL1. A second data set, non-ISP, was collected off an Ethernet connecting an industrial research laboratory to the Internet via a T3 connection. The trace LAB was collected on July 11, ***2000,*** between 18:08 and July 12, 2000 04:05 and consist of 18,000,000 packets or more than 0.7 GB of compressed data.

B. A step-by-step application of the WIND tool

To start, the top plot in Figure 6 shows the traffic rate processes (i.e., number of packets per 10 minutes) associated with five different subnets for the duration of ***8.5*** hour or 52 observation periods, each of length 10 minutes from DIAL1. For observation period 12, the middle plot shows the energy functions corresponding to the traffic rate processes over 10 minutes for the five different subnets. Being more interested in the shape of the energy functions (e.g., dips, breakpoints, etc.) than in their magnitude, we devise the following simple heuristic procedure (Heuristic 1A) for comparing different energy functions and relating them to a "benchmark" or reference energy function. We choose as our reference energy function the (arithmetic) average of the five subnet energy functions (taking a per-scale average) and normalize by requiring all energy functions to coincide with the reference energy function at the smallest scale (in subsequent plots, the reference energy function is always high-lighted in black). Note that this normalization simply means adding or subtracting an offset to each energy function so as to shift them up or down to take the same value at the smallest scale[6]. In the case of the five different subnet energy functions depicted in the middle plot of Figure 6, the effect of this normalization is

[6]This corresponds to multiplying the original time series by some factor.





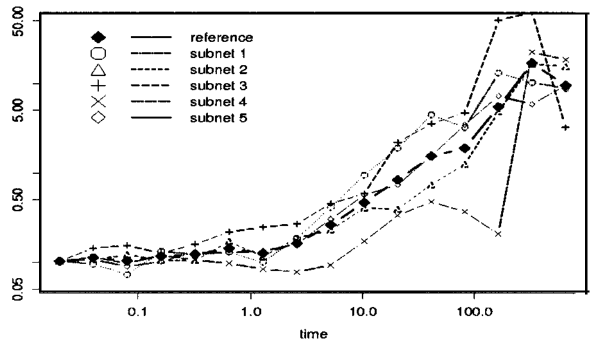
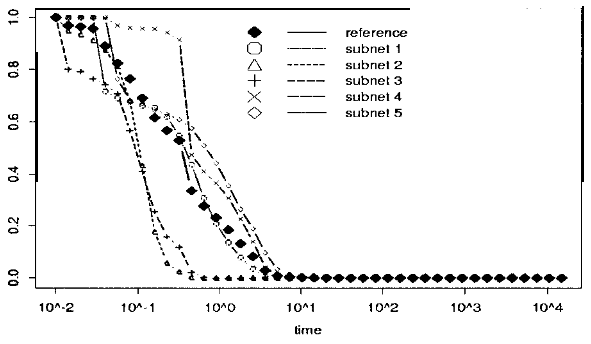


Fig. 6. A view across subnets: Traffic rate processes for five different subnets for the duration of 52 periods from trace DIAL1 (top); energy function plots resulting from the period 12 traffic rate processes for the five subnets (middle); normalized versions of the energy function plots (bottom).

shown in the bottom plot of that same figure and illustrates that the overall shape of the individual energy functions remains preserved.

Next, to check and validate some of the WIND results, we make use of WIND'S feature to calculate RTT and RTO information. To this end, Figure 7 shows the CCDFs of the RTT and RTO values calculated from the packet traces of the same five period 12 traffic rate processes used in the middle and bottom plots in Figure 6. Again, for comparing the different CCDFs and relating them to some "typical" CCDF, we rely on a heuristic (Heuristic 1B) and define the reference or benchmark CCDF for RTTs (RTOs) to be the one constructed from all RTT (RTO) samples from all five packet traces and add them as dotted black curves to the plots. We will illustrate momentarily how to use the various reference functions (i.e., energy function, RTT



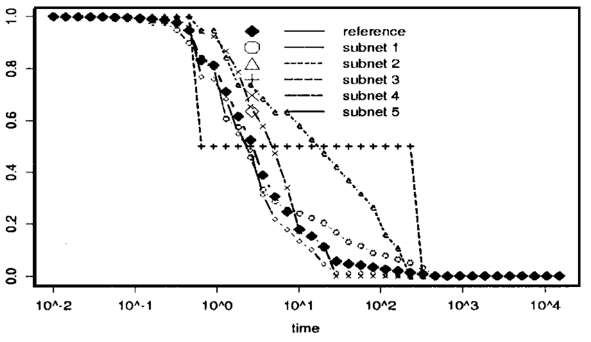


Fig. 7. RTT (top) and RTO (bottom) CCDFs associated with the period 12 traffic rate processes for the five different subnets from above.

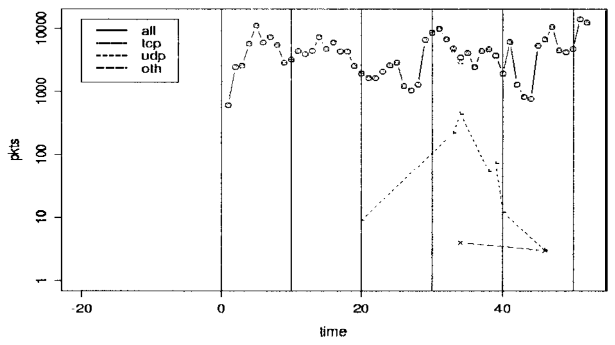
CCDF, and RTO CCDF) to interpret the output of WIND. Figures 6 and 7 illustrate that the performance indeed depends on the subnet considered. For example subnet 3 has much smaller RTTs and experiences almost no retransmissions, while subnet 2 has small RTTs but quite a few and long retransmissions. These correlate well with the results of the scaling analysis shown in the energy plots. More specifically, the scaling plot for subnet 2 shows a dip at the scales corresponding to 10 to 100 seconds, the range of the RTO values for this subnet. The scaling plot for subnet 3 shows a dip at 0.2 seconds matching the median RTT. In this sense, the experience gained from the NS-2 simulation experiments discussed in Section 2 also applies to real networking environments-at least for the one considered here-and our heuristic methods for calculating reference functions seem reasonable.

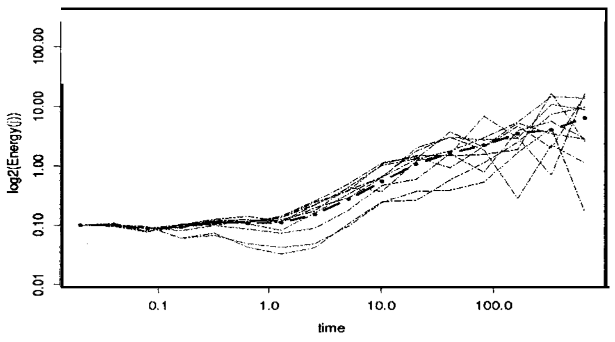
Instead of fixing an observation period and analyzing the traffic to and from different subnets during that period, we can instruct WIND to focus on a single subnet and analyze traffic to and from that subnet across different observation periods. To this end, the top plot in Figure 8 shows the traffic rate process over about 50 successive periods for a single subnet, separated according to the dominant protocols (note that in this case, TCP is responsible for almost all the traffic). For 10 observation periods numbered 22 – 31, the middle plot shows the resulting normalized energy function plots, including the reference energy function, which in this case is the average energy function over all 52 time periods. Our goal is to use this plot to identify "interesting" time periods; i.e., periods that are likely to experience performance problems.

To automate the process of identifying "interesting" periods, we rely on yet another heuristic (Heuristic 2A) for determining whether an energy function is "noticeably" different from the "typical" or reference energy function. In short, we decided on a heuristic procedure that (1) focuses on those energy functions that are below the reference function (for low energy function values suggest either low overall traffic volume or highly regular traffic flow-indications of possible network problems such as congestion, route change, or user behavior change); and (2) takes into account different scaling behavior in small, medium, and large time scale regions, reflecting respectively inter-packet, round-trip, and retransmission timeout time scales. To quantify the difference between a given energy function and the reference energy, we use a metric that measures some notion of "region-specific area below the reference function"[7]. If the difference in any of the three regions exceeds a certain level, the corresponding observation period will be automatically flagged as "interesting". The level is determined by considering the energy functions whose metrics exceed the 90% quantile. To see Heuristic 2A in action, the bottom plot in Figure 8 shows the energy functions corresponding to those periods that our implemented procedure flagged as "interesting" (i.e., periods 26, 30, and 31). Note that when compared to all the 10 energy function curves shown in the middle plot, Heuristic 2A appears to succeed in picking out the likely "trouble makers".

Once WIND has flagged an observation period "interesting", how can we check whether or not there is indeed a problem during the period in question along the paths to the subnet at hand? Furthermore, if there is a problem, can we explain what sort of performance problem occurs by correlating the energy functions with traffic volume, RTT or RTO information, or with other available data? Motivated by the NS-2 experiments in Section 2 which suggest the feasibility of correlating the energy function with the RTT and/or RTO CCDFs, we devise a new heuristic (Heuristic 2B) along similar lines to Heuristic 2A to flag an observation period as "interesting" based on a comparison of the RTT and RTO CCDFs against the reference RTT CCDF and reference RTO CCDF, respectively, where the latter are computed based the samples obtained from all 52 periods. Heuristic 2B uses the same metric as Heuristic 2A for measuring the differences between a given CCDF and the corresponding reference CCDF, and a CCDF is flagged

[7]We approximate the area by computing a weighted sum of the differences between the energy function and the reference energy function.





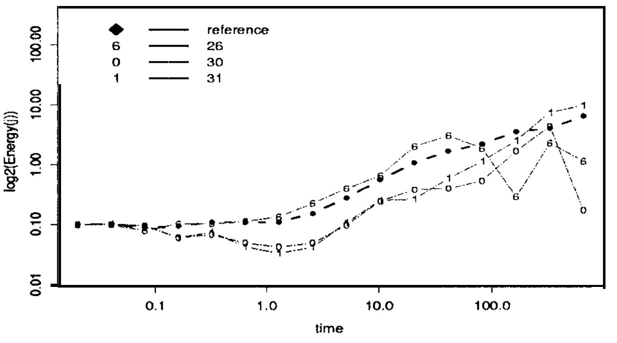
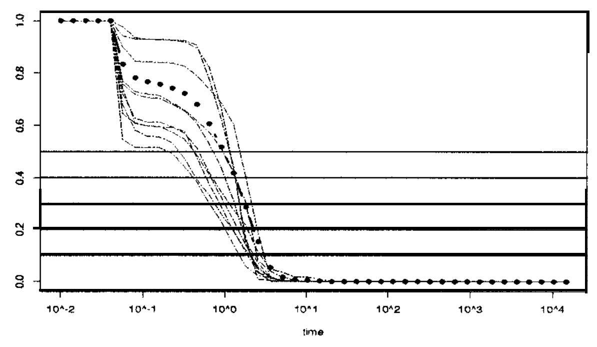


Fig. 8. A view across time: Traffic rate process for a single subnet over a number of different observation periods, separated by protocol (top); normalized energy functions and reference energy function corresponding to 10 successive periods numbered 22 – 31 (middle); normalized energy functions (and reference energy function) corresponding to the three periods that WIND flagged as "interesting" (bottom).

(and the corresponding period is deemed "interesting") as soon as the difference exceeds a given threshold. The results of applying Heuristic 2B are depicted in Figures 9 and 10. For example, the top plot in Figure 9 shows the RTT CCDFs resulting from the 10 observation periods 22 – 31 and also includes the reference CCDF. The bottom plot depicts those RTT CCDFs that Heuristic 2B identified as troublesome, resulting in the periods 29, 30, and 31 to be classified as "interesting" from a RTT CCDF perspective. Similarly, Figure 10 shows that periods 23, 26, and 31 are flagged as "interesting" as a result of the automated RTO CCDF analysis. This implies that in this case, WIND'S selection of periods 26, 30, and 31 as being "interesting" is a success. Time period 26 is "interesting" because of in-



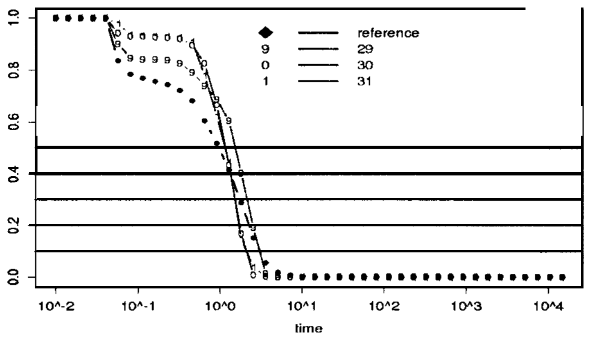


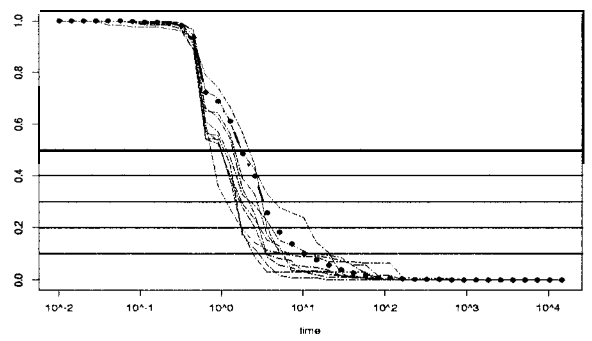
Fig. 9. A view across time (cont.): RTT CCDFs and reference RTT CCDF corresponding to 10 successive periods numbered 22 – 31 for a single subnet (top); RTT CCDFs (and reference CCDF) corresponding to the three periods that Heuristic 2B flagged as "interesting" as far as unusual RTT behavior is concerned (bottom).

creased RTOs while periods 30 and 31 are "interesting" because of increased RTTs. False positives with regard to the correlation with RTT and RTO behavior, are those periods that the wavelet heuristic identifies as "interesting" but that are not flagged by the RTT or RTO heuristic.

Running WIND on other parts of the data, with either pre-determined or running lists of top-20 subnets, we found that after some fine-tuning of the heuristics used in the automated procedures for detecting and identifying periods of network performance problems, our tool successfully classified problematic periods in 80 – 90% of the cases. By successful, we mean here that either the corresponding RTT or RTO analysis or a simple investigation of traffic loads indicated unusual network performance during the periods that WIND flagged as "interesting."

C. WIND *in practice: Experiences and shortcomings*

After illustrating the network performance detection methodology of WIND on a data set collected from a particular networking environment, we next report on some of our experiences when running WIND on data sets collected from a very different environments with distinctly different characteristics as far as RTT behavior, application and traffic mixes, user behavior, etc. are concerned. In short, our experience has been that fine-tuning the parameters used in



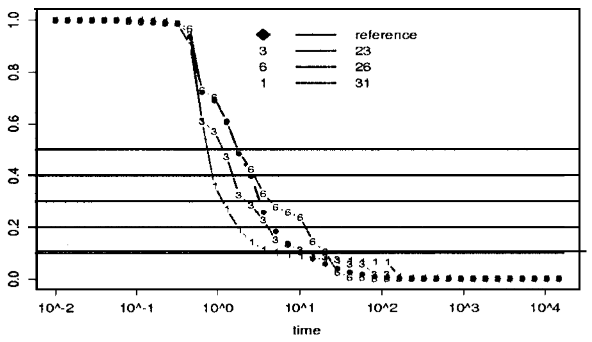
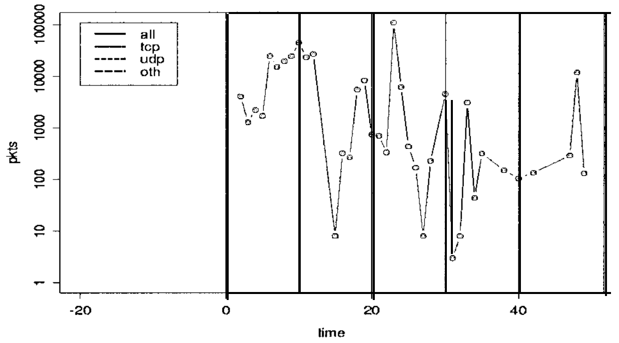
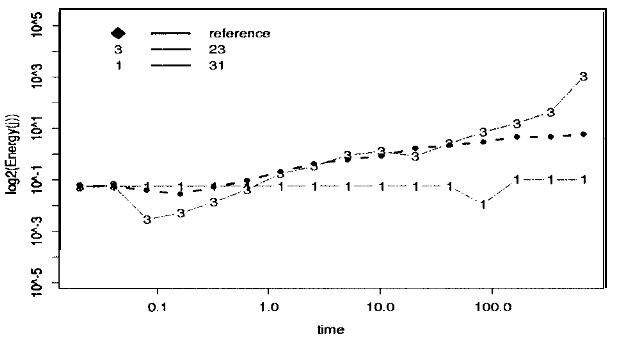


Fig. 10. A view across time (cont.): RTO CCDFs and reference RTO CCDF corresponding to 10 successive periods numbered 22 – 31 for a single subnet (top); RTO CCDFs (and reference CCDF) corresponding to the three periods that Heuristic 2B flagged as "interesting" as far as unusual RTO behavior is concerned (bottom).





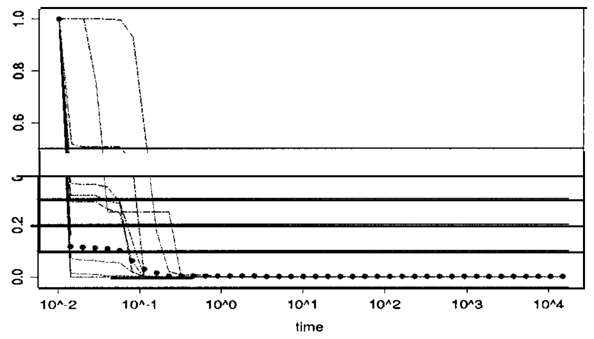
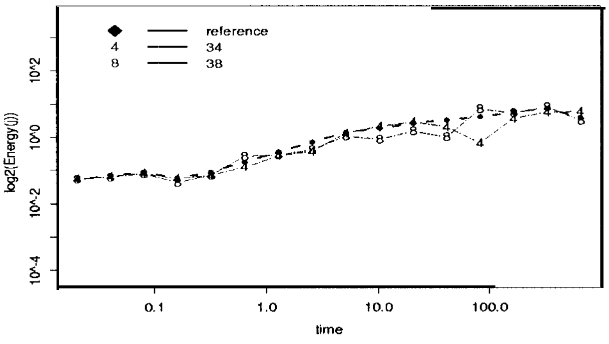


Fig. 11. A view across time (for our second data set): Traffic rate process to single subnet cross periods 1 – 50 (top); normalized energy function plots (and reference energy function) for period 23 and 31 flagged by WIND as "interesting" out of the 10 periods labeled 22 – 31 (middle); RTT CCDFs for the 10 periods 22 – 31 (bottom).

our heuristics, especially in Heuristics 2A and 2B, to capture "normal" operating conditions for the network under consideration is crucial for keeping the number of false-positive below 10% or so. To illustrate the effects that lead to false-positives, consider Figure 11 which was obtained running WIND on our second data set, focusing on a single subnet, and looking across 10 successive periods labeled 22 – 31. While the top plot in that figure shows the traffic rate process (all TCP) over the whole measurement period (i.e., 48 observation periods), the middle plot states that periods 23 and 31 have been identified as "interesting" by WIND. However, upon closer inspection, the fact that the energy function corresponding to period 31 is almost flat turns out to be an artifact of the very low traffic load during this period (see top plot). In this sense, some care is needed when interpreting such outliers in the space of possible energy functions, but finding remedies is sometimes as easy as checking traffic volume statistics.

The scaling plot labeled 23 also indicates a potential problem with our simple heuristics. In fact, while WIND flags period 23 as "interesting," indicating that the application suffers worse than average performance, a more careful investigation into this particular packet trace reveals that quite the opposite is true; that is, the packets of this traffic experienced better RTT performance than during some of the other observation periods (see the bottom plot of Figure 11, where the CCDF corresponding to the period 23 scaling plot is in the very lower left-hand comer, suggesting very good RTT performance. The reason why our Heuristic 2A yields this false-positive is that it does not account for dips in the energy functions that shift to smaller time scales. The RTT CCDFs in the bottom plot of Figure 11 also point out that without more refined heuristic procedures, WIND will not detect all possible shifts in round trip time.

In the examples we considered so far, the differences between the energy function plots have been in general significant. Figure 12 (top) shows selected energy function plots where the differences seem minimal. Nevertheless the corresponding RTT CCDFs give clear evidence that the



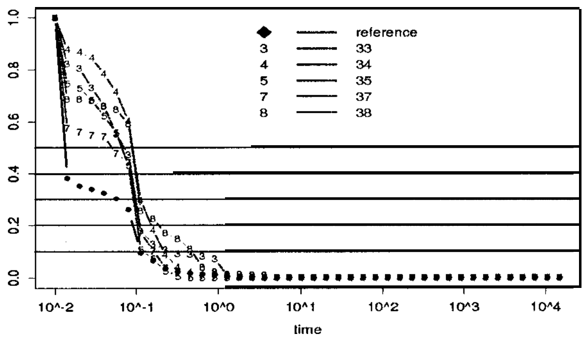


Fig. 12. Normalized energy function plots for selected time periods (top); RTT CCDFs corresponding to selected periods (bottom).

RTT behavior was significantly different during those time periods. The reason for this justified or unjustified "warning" is again in the simplicity of our heuristics; improving them by, for example, requiring the measured differences between a given energy function and the corresponding reference function to exceed a certain absolute threshold, will likely take care of the problem pointed out by this example. We have also experienced cases where the energy function plots indicate potential problems that none of the other metrics considered so far (e.g., RTT, RTO, volume, or other simple detection heuristics) could capture and/or confirm. We suspect that in some of these cases, the energy function plots pick out systematic/periodic components that are present over certain time scale(s) but remain hidden from those other metrics; however, we will pursue this in future work.

VI. Conclusions

In this paper, we explored the feasibility of inferring information about network path properties from passive packet-level measurements collected from a single tap point to the Internet. To this end, we exploit a number of structural characteristics of measured TCP traffic that reveal themselves by relying on the scale-localization ability of wavelet-based analysis methods. We show that it is possible to detect different network performance problems in a manner that is completely non-intrusive (i.e., no artificial traffic is generated). In fact, relying exclusively on existing network traffic, we can obtain information about the present state of (parts of) the network. We concurrently collect and analyze packet-level measurements from active users and applications as they communicate across the Internet. Our methodology is adaptive in that it can follow popularity and activity shifts; it also appears scalable since it keeps statistics only for the most popular networks.

To illustrate our proposed methodology, we developed WIND, a prototype tool for Wavelet-based INference for Detecting network perfromance problems. We showcase WIND'S capabilities (and shortcomings) for detecting network performance problems by applying it to measured user traffic from two different networking environments. For each environment we correlate the results of the wavelet-based trace analysis across time and across different network path. In addition to performing the computationally inexpensive wavelet-based trace analysis, we also compute other, computationally more expensive networking-related quantities such as RTT and RTO statistics. We observe how the different statistics change over time and how they are correlated to the results of our wavelet-based analysis. Despite implementing relatively simplistic heuristics for correlating the various quantities, for the two networking environments considered, we found WIND to have a 80 – 90% success rate in correctly identifying network path-related problems; that is, detecting network performance problems that can be confirmed (or even explained) via use of some other data. Studying the degree to which there are network problems that RTT/RTO- and/or WIND-based analysis methods are unable to detect remains an interesting future research topic.

Although our approach in this paper focuses exclusively on the use of passive measurements for detecting network problems, there is nothing that prevents one from applying the same type of wavelet-based analysis to active measurements. In fact, in an effort to ultimately correlate network performance problems with network topology-specific aspects, we have already begun to extend WIND to also perform, collect, and analyze active measurements, in particular traceroute data. We fully expect that applying the broad spectrum of available wavelet techniques, especially a combined time- and scale-localization approach for investigating network-related measurements locally in time and scale, has the potential for creating unprecedented opportunities for exploiting the information contained in a combination of carefully-made active and passive measurements. In view of the potential of such a full-fledged wavelet approach towards analyzing network measurements, our present exclusive focus on the wavelets' scale-localization abilities seems narrow-minded, but the results obtained here already point out new possibilities/opportunities for designing and implementing measurement tools. Note that such opportunities have been largely obscured in the past by some of the shortcomings of traditional time- or frequency-domain based traffic analysis methodologies.

REFERENCES

[1] P. Abry and D. Veitch. Wavelet analysis of long-range dependent traffic. *IEEE Transactions on Information Theory 44,* 1998.

[2] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch. Wavelets for the analysis, estimation and synthesis of scaling data. In: *Self Similar Network Traffic Analysis and Performance Evaluation,* K. Park and W. Willinger (Eds.), pp. 39-88, Wiley, 2000.

[3] A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S. B. Moon, V. Paxson, and D. Towsley. The use of end-to-end multicast measurements for characterizing Internal network behavior. *IEEE Communications,* Vol. 38(5), pp. 152-159, May 2000.

[4] M. Allmann and V. Paxson. On estimating end-to-end network path properties. *Proc. of the SIGCOMM'99*, Cambridge, MA.

[5] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu and H. Yu. Advances in Network Simulation. IEEE Computer, May 2000.

[6] H. Balakrishnan, S. Seshan, M. Stemm, and R. H. Katz. Analyzing stability in wide-area network performance. *Proc. of the ACM/SIGMETRICS'97*, Seattle, WA, 1997.

[7] P. Barford and M. E. Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of ACM Sigmetrics'98,* pages 151-160, 1998.

[8] P. Barford and M. Crovella. Measuring Web performance in the wide area. *Computer Performance Review,* Vol. 27(2), 1999.

[9] P. Barford and M. Crovella. Critical path analysis of TCP connections. *Proc. of the ACM/SIGCOMM'00*, Stockholm, Sweden.

[10] J. C. Bolot. End-to-end packet delay and loss behavior in the Internet. *Proc. of the ACM/SIGCOMM'93*, San Francisco, CA.

[11] R. Caceres, P. Danzig, S. Jamin, and D. Mitzel. Characteristics of wide-area TCP/IP conversations. *Proc. of the ACM/SIGCOMM'91*, Zurich, Switzerland.

[12] R. Caceres, N. Duffield, A. Feldmann, J. Friedmann, A. Greenberg, R. Greer, T. Johnson, C. Kalmanek, B. Krishnamurthy, D. Lavelle, P. Mishra, K. K. Ramaktishnan, J. Rexford, F. True, and J. van der Merwe, Measurement and analysis of IP network usage and behavior, *IEEE Communications,* Vol. 38(5), 2000.

[13] CAIDA. Performance measurement tools taxonomy. http://www.caida.org/tools/taxonomy/performance.xml

[14] R. L. Carter and M. E. Crovella. Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation,* Vol. 27&28, pp. 297-318, 1996.

[15] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic - evidence and possible causes. In *Proc. of ACM/SIGMETRICS'96*, Philadelphia, PA, pp. 160-169, 1996.

[16] I. Daubechies. *Ten lectures on wavelets.* SIAM, 1992.

[17] A. B. Downey. Using pathchar to estimate Internet link characteristics. *Proc. of the ACM/SIGCOMM'99*, Cambridge, MA. .

[18] A. Fei, G. Pei, R. Liu, and L. Zhang. Measurements on delay and hop-count of the Internet. *Proc. IEEE Global Internet'98.*

[19] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. *Proc. of the ACM/SIGCOMM'99*, Cambridge, MA.

[20] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford. NetScope: Traffic engineering for IP networks, *IEEE Network,* Vol. 14(2), pp. 11-19, 2000.

[21] V. Jabcobson, C. Leres, and S. McCanne. libpcap. Available from ftp://ftp.ee.lbl.gov/libpcap.tar.Z, June 1989.

[22] V. Jabcobson, C. Leres, and S. McCanne. tcpdump. Available from ftp://ftp.ee.lbl.gov/tcpdump.tar.Z, June 1989.

[23] V. Jacobson. traceroute. Available from ftp://ftp.ee.lbl.gov/traceroute.tar.Z, 1989.

[24] V. Jacobson. pathchar. Available from ftp://ftp.ee.lbl.gov/pathchar/, 1997.

[25] K. Lai and M. Baker. Measuring bandwidth. *Proc. IEEE INFOCOM'99*, New York, NY, 1999.

[26] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. *Proc. of the ACM/SIGCOMM'00*, Stockholm, Sweden.

[27] W. Leland, M. S. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. In *Proc. of the ACM/SIGCOMM'93*, San Francisco, CA.

[28] G. R. Malan and F. Jahanian. An extensible probe architecture for network protocol performance measurement. *Proc. of the ACM/SIGCOMM'98*, Vancouver, Canada.

[29] M. Mathis and J. Mahdavi. Diagnosing Internet congestion with a transport layer performance tool. *Proc. INET'96,* Montreal, June 1996, http://www.psc.edu/ mathis/htmlpapers/inet96.treno.html.

[30] W. Matthews and L. Cottrell. The PingER project: Active Internet performance monitoring. *IEEE Communications Magazine,* Vol. 38(5), pp. 130-137, May 2000.

[31] V. Paxson. End-to-end routing behavior in the Internet. *Proc. of the ACM/SIGCOMM'96*, Stanford, CA.

[32] V. Paxson. *Measurement and analysis of end-to-end Internet dynamics.* Ph.D. dissertation, U. C. Berkeley, May 1996.

[33] V. Paxson. Towards a framework for defining Internet performance metrics. *Proc. INET'96,* Montreal, June 1996, ftp://ftp.ee.lbl.gov/papers/metrics-framework-INET96.ps.Z.

[34] V. Paxson. End-to-end Internet packet dynamics. *Proc. of the ACM/SIGCOMM'97*, Cannes, France.

[35] V. Paxson. End-to-end Internet packet dynamics. *IEEE/ACM Transactions on Networking,* Vol. 7(3), pp. 277-292, 1999.

[36] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *Proc. of the ACM/SIGCOMM'94*

[37] S. Savage. Sting: A TCP-based network measurement tool. *Proc. of the USENIX Symposium on Internet Technologies and Systems, 1999.*

[38] S. Seshan, M. Stemm, and R. H. Katz. SPAND: Shared passive network performance discovery. *Proc. of the USENIX Symposium on Internet Technologies and Systems,* 1997.

[39] W. R. Stevens. *TCP/IP Illustrated Volume* 1. Addison-Wesley, 1994.

[40] K. Thompson, G. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network,* Vol. 11, 1997.

[41] D. Towsley et al. MINC: Multicast-based Inference of Network-internal characteristics. http://gaia.cs.umass.edu/minc, 1998.

[42] Y. Zhang, V. Paxson, S. Shenker, and L. Breslau. The stationarity of Internet path properties: Routing, loss, and throughput. Preprint, 2000.

[43] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch. Wavelets for the analysis, estimation and synthesis of scaling data. In: *Self Similar Network Traffic Analysis and Performance Evaluation,* K. Park and W. Willinger (Eds.), pp. 39-88, Wiley, 2000.

[44] R. Braden and A. Deschon. NNStat: Internet Statistics Collection Package, Introduction and User's Guide. Technical Report RR-88-206, ISI, USC, 1988