# Instrumentation for exact packet timings in networks

Daniel A. Freedman*†, Tudor Marian*, Jennifer H. Lee‡, Ken Birman*, Hakim Weatherspoon*, Chris Xu‡

* Department of Computer Science, Cornell University, Ithaca, New York, USA
† Department of Physics, Cornell University, Ithaca, New York, USA
‡ Department of Applied and Engineering Physics, Cornell University, Ithaca, New York, USA

***Abstract*—We design and implement a novel class of highly precise network instrumentation, capable of the first-ever capture of exact packet timings of network traffic. Our instrumentation — combining real-time physics test equipment with off-line post-processing software — prevents interference with the system under test, provides reproducible measurements by eliminating non-deterministic error, and uses transparent and ubiquitous lab equipment and open-source software for ease of replication. We use our technique to perform *in-situ* observations of 10 Gigabit Ethernet packets in flight on optical fiber, showing improvements in timing precision of two to six *orders of magnitude* over existing methods of measurement, which generally employ software on commodity computer endpoints of network paths.**

## I. Introduction

The systems and networking disciplines in computer science have long depended upon quantitative measures of network performance, which play diverse but critical roles in system development and validation. Our work contributes to the science of instrumentation, with the larger goal of enabling better understanding of the behavior of high-speed networks and the applications that utilize them.

We design and implement novel high-precision instrumentation — BiFocals — for Internet timing measurements: it enables the generation of extremely precise traffic flows, as well as the *in-situ* capture and analysis of packets in flight. In contrast to existing methods, we do *not* interface with computer endpoints at all; rather, we directly tap optical fibers using typical physics test-equipment (oscilloscopes, frequency synthesizers, lasers, etc.). We generate and acquire, in real-time, waveforms of the optical power modulations on the fiber. We process these off-line to extract packets, thus avoiding the non-determinism and systemic noise that confound many conventional techniques. In doing so, we obtain six orders-of-magnitude improvement in timing precision over existing endpoint software techniques and two to three orders-of-magnitude relative to prior hardware-assisted solutions.

In this work, we focus on the design underlying BiFocals, while also comparing it to existing, common methods for timing packets at high speeds, with data rates up to 10 Gbps. The instrumentation lessons here are universal across different data-rate standards. Further, we outline novel applications, both in continued research and commercial settings, that arise from the guarantees provided by the design of our instrumentation.

## II. Motivation

In order to exactly measure timings in network packet flows, BiFocals departs substantially from existing techniques. We present a taxonomy of different approaches to measurement, of increasing precision, in order to motivate the resulting architectural decisions that inform our design of BiFocals.

As we shall see below, BiFocals' precision derives from its interaction with a much lower level of the network stack than existing methodologies. Thus, to understand this measurement taxonomy, we first must review the behavior of the Physical Layer — a portion of the network stack completely hidden from the end-host kernel and other software. For the ensuing discussion, we focus upon the Physical Layer of optical 10 Gigabit Ethernet (10GBase-R) [1].

### A. Physical Layer background

In a commodity end-host computer, the Ethernet controller of a typical 10GBase-R network adapter accepts Ethernet packets from higher layers of the network stack in the kernel and prepares them for transmission across the physical medium of the optical fiber span. However, the network adapter does not transmit individual Ethernet packets across the network, but instead embeds the data *bitstream* of discrete network packets within a continuously transmitted *symbolstream*. The crucial point here is that, while the higher-layer data bitstream involves discrete Ethernet packets, the lower-layer symbolstream is continuous. Every symbol is the same width in time ($\sim$100 picoseconds) and is transmitted at the precisely identical symbol rate ($\sim$10 GBaud), completely irrespective of the data rate of the actual network traffic.

We shall see that existing measurement techniques lack access to the continuous timebase of the symbolstream; they thus face difficulties in accurately determining the arrival times of such discrete network packets, resulting in errors in timing measurements, even though the network packets themselves are properly received and transferred to higher layers of the stack. The manner in which packets are time-stamped thus determines the precision of the resulting timing measurements.

### B. Sources of measurement error

In categorizing competing methods for time-stamping packets, the pertinent differences involve the "when" and "where" of time stamping as packets transit the network and arrive at either the commodity end-host receiver, or our BiFocals tool. We outline four approaches of increasing precision:

***User-space software packet stamping:*** Software applications, executing in user-space context and deployed on commodity operating systems and computer end-hosts, serve overwhelmingly as the most common network measurement

978-1-4244-7935-1/11/$26.00 ©2011 IEEE

tools [2]. Packets are assigned time-stamps as the user-space software processes them; such observations enable inference into traffic behavior on network paths. While software tools are essential and productive elements of network research, it has long been recognized that they risk distortion of the metrics they seek to measure. The core problem involves the unmeasurable layers between the software and the optical fiber: network adapter (with hardware queues, on-chip buffering, and interrupt decisions), computer architecture (chipset, memory, and I/O buses), device driver, operating system (interrupt delivery and handling), and even measurement software itself. Each of these layers adds its own dynamics, distorts measurements in ways not deterministically reproducible, and contributes strongly to the timing errors discussed in Section IV.

***Kernel interrupt-handler stamping:*** Rather than time-stamping packets in user-space upon arrival, the operating system kernel (with modification) can internally time-stamp packets while servicing the network-adapter interrupts that announce their arrival. Such a technique removes ambiguities involved with kernel scheduling of the measurement application, as well as contention across memory buses. This method is not often used in practice due to the complexity of kernel and application modification. However, as discussed below, we implement an example of this approach to serve as a more stringent control against which we can compare our BIFOCALS instrumentation. In Section IV, we show that it still causes severe measurement distortion.

***Network-adapter bitstream stamping:*** Both commercial solutions (DAG [3], Ixia [4]) and academic projects (NetFPGA [5]) address some of the sources of error above; the commercial varieties are primarily used by major router design firms and bear significant acquisition costs. These approaches involve specialized network adapters (generally, custom FPGAs) to enable packet time-stamping functionality in the network-card hardware. While they aim to stamp the packets as early in their processing as possible, they still must first extract individual packets from the underlying Physical Layer symbolstream. As the continuous timebase is lost in doing so, they remain unable to exactly characterize the timing of network packets.

***On-fiber symbolstream stamping:*** Our BIFOCALS instrumentation represents a substantial departure from the techniques enumerated above. Excluding the end-host completely and directly tapping the fiber transport, we record a contiguous portion of the entire Physical Layer symbolstream in real-time; only later, in off-line post-processing, do we extract the discrete Ethernet packets from this captured trace and assign time-stamps in relation to the symbolstream timebase. As our precision is significantly better than the width of a single symbol (∼100 ps), our time-stamps are exact. We recall the difference between the discrete nature of the data bitstream and the presence of a continuous timebase in the symbolstream, where every symbol is the same width and transmitted at an identical symbol rate, irrespective of the data rate of the actual network traffic. Therefore, the fidelity of our instrumentation is agnostic to the data rate of the network traffic, as we always generate and capture traffic at the full 10GbE symbol rate of 10.3125 GBaud of the underlying Physical Layer. Whether the actual captured symbolstream is embedded with no data traffic (only infinitely repeating "idle" codewords) or maximal traffic density, our instrumentation responds identically and provides the exact time measurement of each packet.

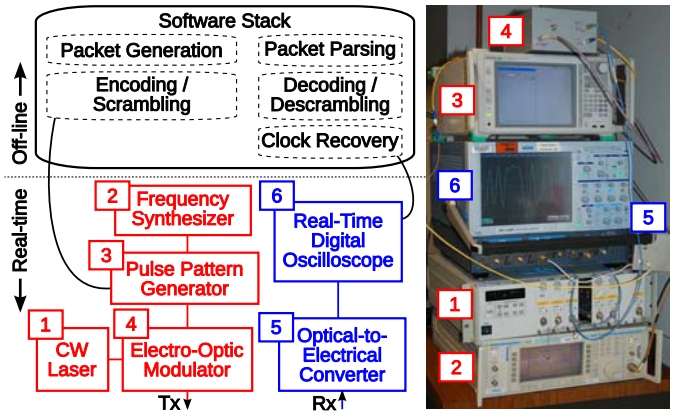



Fig. 1: Diagram of BIFOCALS transmission and acquisition hardware and software connected across the network under test, with notations on the photograph of the hardware.

## III. INSTRUMENT DESIGN

In Section II, we articulated the key design decision within BIFOCALS to allow us to recover the exact timing of network packets in flight: we time-stamp packets using their associated on-fiber symbolstream. To understand how this criterion translates into practice, we introduce and detail our instrumentation architecture here.

### A. Instrumentation architecture

As depicted in Figure 1, BIFOCALS can be viewed as a special network adapter decomposed into two independent layers — an off-line software stack for the generation and deconstruction of symbolstreams, and separate physics test equipment (oscilloscopes, pattern generators, lasers, etc.) to faithfully send and receive these symbolstreams on the optical fiber. Note that this clean decomposition also separates what we implement in software (the bits we send) from what we implement in hardware (how we send them), enabling us to separately validate the fidelity of our hardware, independent of the software implementation of the Physical Layer. Further, this ensures that we can reproducibly send identical traffic on successive iterations, unlike common methods (`tcpreplay`, `iperf`, etc.) that introduce non-determinism.

On the software level, information is represented in binary Ethernet-compliant symbolstreams, as sequences of ones and zeros (with each integer representing a distinct bit). On the hardware level, information is represented by light intensity: optical power modulated in time, off and on, to correspond to "0" and "1" bits, with unit length set by the symbol rate. This hardware implementation ensures that the binary symbolstreams are transmitted and acquired with perfect fidelity.

The IEEE 802.3ae standard employs an "NRZ" (Non-Return-to-Zero) data format, where the signal does not return to an intermediary analog position (the "zero" in NRZ) between pulses, but instead maintains the same analog level for repeating digital bits. The NRZ data format, while requiring less bandwidth for data transfer, requires a non-trivial clock recovery procedure. Fortunately, modern Ethernet standards have features designed to facilitate clock recovery and lock; for example, they ensure frequent bit transitions by using encoding tables or scrambling algorithms and alternating-bit sync headers.

### B. Hardware foundation

We reference Figure 1 above to depict both the transmission and acquisition hardware. All electrical and optical components used here are commercially available and commonly found in optical fiber communications labs. (Kaminow and Li [6] provide a comprehensive review of fiber components and systems.) The optical components for the transmitter consist of a continuous wave (CW) distributed feedback (DFB) laser centered at $\lambda = 1555.75$ nm (ILX) and an electro-optic modulator (EOM, JDS Uniphase). The constant intensity output of the CW laser is switched on and off by the EOM based upon a supplied electrical signal from a pulse pattern generator (PPG, Anritsu). The PPG is clocked by a precise frequency synthesizer (Marconi) tuned to 5.15625 GHz and frequency-doubled to 10.3125 GHz (corresponding to the specified 10.3125 GBaud symbol rate of Table 52–12 of IEEE 802.3-2008 [1]). The signal to the EOM is amplified to appropriate levels with RF amplifiers (Picosecond Pulse Labs). The PPG can be programmed with an arbitrary finite-length (here, 128 Mbit) bit sequence; it outputs an electrical waveform corresponding to these symbols continuously repeated. The resulting optical signal from the EOM has high light intensity representing "1" bits and no light intensity representing "0" bits. The optical signal from the EOM is output through a single-mode optical fiber, which completes the optical transmitter.

On the receiver side, the BIFOCALS acquisition hardware consists of a fast, broadband 12.3 Gbps optical-to-electrical (O/E) converter (Discovery Semiconductor) and a real-time digital oscilloscope (LeCroy) with fast sampling (40 GSa/sec), high detection bandwidth (11 GHz), and deep memory (100 MSa). The O/E converter, a broadband photodetector with a built-in high-gain current-to-voltage amplifier, transforms the incident optical waveform into an electrical output signal. We employ the real-time oscilloscope as an analog-to-digital converter (ADC), sampling the output from the O/E converter in excess of the Nyquist rate. Leveraging a precisely calibrated timebase, this real-time oscilloscope captures waveform traces that precisely reflect the symbolstream on the fiber. Waveform traces are processed off-line by our software stack.

We show the measured eye diagram of the BIFOCALS transmitter in Figure 2. We measure the eye diagram by connecting the optical output of our transmitter to a wideband sampling oscilloscope (Agilent) triggered with the frequency synthesizer, which overlays the sequence of samples in time, synchronized at a fixed point in the symbol frame. The measured eye diagram for the BIFOCALS transmitter has lines that are thin and well-defined, indicating low amplitude and timing noise. Its central eye-opening is large and free of measured points, thus ensuring unambiguous "1" and "0" symbols in the signal. We also confirm via the measured eye diagram that our transmitter is in compliance with the time and amplitude standards for 10GBase-R.

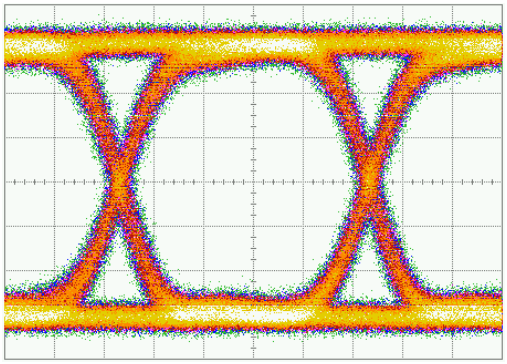


Fig. 2: Eye diagram of the optical signal transmitted by BIFOCALS hardware: large, open eye with negligible noise or jitter and conformance with 10GBase-R specifications for optical transmission power, rise time, eye mask, etc. Horizontal scale is 20 ps/div and vertical is 80 $\mu$W/div.

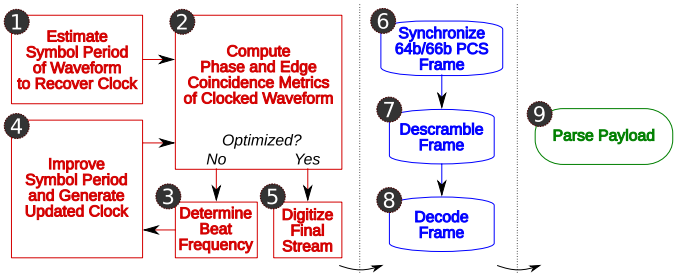
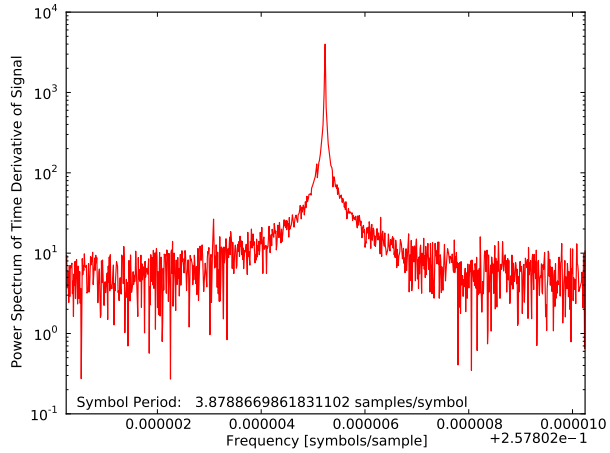



Fig. 3: Flow-chart depicting three stages (and underlying nine modules) of post-processing software stack: Clock recovery and digitization; packet recovery; and payload recovery.

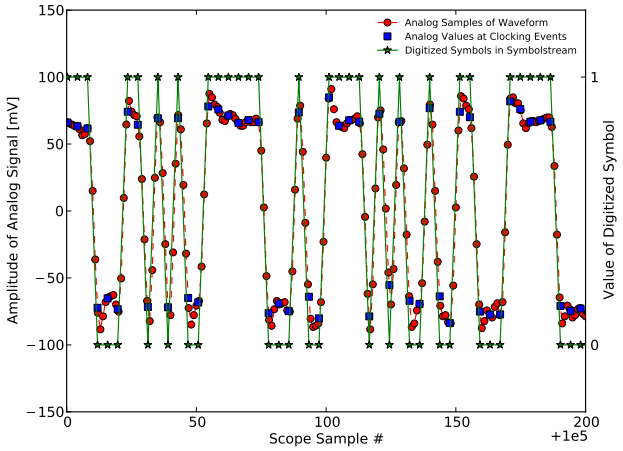
### C. Software stack

Figure 3 depicts the software stack for the BIFOCALS receiver as three primary stages, with nine underlying software modules. The three stages correspond to (1) *Clock recovery and digitization*: Converts analog waveform into digitized symbolstream; (2) *Decoding and descrambling*: Converts continuous symbolstream into discrete Ethernet packets; and (3) *Packet parsing*: Parses packets and analyzes payloads. This software stack interfaces with its underlying hardware foundation as depicted in Figure 1 above.

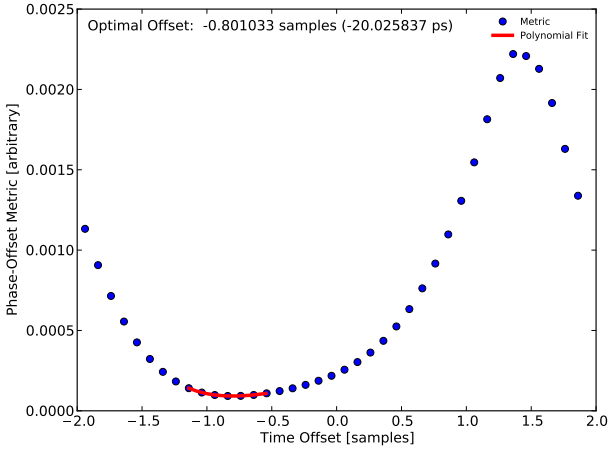
The operation of key modules for clock recovery is shown in Figure 4. This stage converts the captured waveform, oversampled by the oscilloscope hardware, into a valid Physical Layer symbolstream. The most demanding process here involves the extraction of the symbol rate with sufficient precision to enable accurate recreation of the Physical Layer symbolstream. The



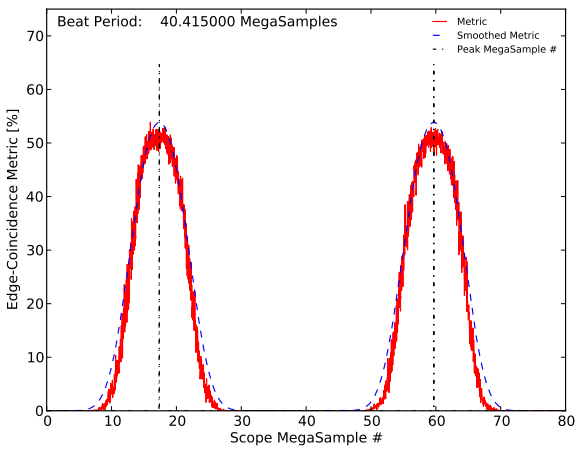

(a) Spectral line of symbol transitions




(b) Waveform of physical layer symbolstream




(c) Metric to align phase of clocking events




(d) Metric to refine symbol period (through beat frequency)

Fig. 4: *Clock recovery and digitization*: (a) Estimation of the symbol period of the waveform, through Fourier transform of its time derivative; (b) Representative waveform trace, showing raw analog samples from the oscilloscope, samples at clocked intervals, and (eventual) resulting digitized values of symbols; (c) Metric to optimize the phase of clocking events, relative to transitions in the waveform, by minimizing overlap between clocking events and rising or falling edges in the waveform; (d) Metric to iteratively refine the symbol period, and thus better align clocking events with symbol plateaus, by examining the beat phenomena generated when the symbol period is not sufficiently precise to accurately clock the entire waveform trace.

symbol rate dictates the symbol period of our waveform, which then defines the separation in time, in numbers of samples from the oscilloscope, between consecutive symbols in the resulting symbolstream. The goal of clock recovery is to obtain an accurate value for this period, as it represents the time between the clocking events we use to digitize the waveform.

The first module, shown in Figure 4(a), provides an initial estimation of the symbol period of the sampled analog waveform, by computing the Fourier transform of its time derivative. Using this information, we can transform the analog waveform into a digital symbolstream: Figure 4(b) depicts the analog samples of the waveform (red circles) as captured by the hardware oscilloscope, the related analog waveform values interpolated at the times of each clocking event (blue squares), and the resulting symbol values after these clocked values are digitized (green stars).

The subsequent two subfigures depict metrics that are necessary to refine the precision of our initial estimate of the symbol period. (Space constraints here preclude complete enumeration of the formulation of each, though we outline their use). The phase-offset metric in Figure 4(c) helps align the phase of clocking events with that of the symbols, in terms of a phase offset of some fraction of the symbol period (here, in units of analog waveform samples). Its minimization ensures that successive clocking events are temporally aligned with the level plateaus of their accompanying symbols, preventing

otherwise inevitable ambiguity and resultant error in the symbol value. Finally, Figure 4(d) illustrates the edge-coincidence metric used to refine the symbol period, an estimate of which was originally formulated above in Figure 4(a). We examine conditions during which clocking events coincide with the transition edges between level plateaus, even after successful determination of the phase offset, and generate high values in this metric, which lead to erroneous digitized output. Such periodic phenomena reflect the accumulation of very small errors in our estimation of the symbol period — in other words, drift or walkoff across the entire length of the sampled analog waveform. The edge coincidence metric demonstrates obvious beat-frequency–like phenomena, representing insufficient precision in our determination of the symbol period from the spectral analysis in the first module. By calculating the beat frequency and using it to refine the symbol frequency, and thus the symbol period, we can ensure that our clocking events occur only at valid, unambiguous times for each symbol value. We witness the success of this iterative process in the unambiguous digitization of analog samples shown in Figure 4(b).

With the successful completion of the clock recovery and digitization module, we turn to the final two modules of packet and payload recovery in Figure 3. These modules internalize the intelligence and semantics of the Physical Layer and all other network layers: first, the descrambling, and then the decoding, of the 64b/66b Physical Coding Sublayer (PCS) required for transmission on the physical fiber media, as specified by IEEE 802.3-2008 [1]. This module then provides the raw, discrete Ethernet packets, from which higher layers (IP, UDP, and various application payloads) of the network stack can be extracted through straightforward parsing.

In comparison to the computation and complexity of clock recovery in the BIFOCALS receiver, the transmission stack is much more straightforward. It simply inverts the order and operation of the latter two modules; hardware components handle the clock generation (namely, the frequency synthesizer seeding the PPG, as described above).

## IV. COMPARISON

It is worthwhile to question the extent of the need for the improved precision that BIFOCALS provides. Indeed, as we mention in the Introduction above, the packet chains that we ultimately discuss in Section V-A show regimes of tiny timing delays interspersed by gaps of huge delays. This leads one to wonder: Could not such qualitative behavior be captured by existing techniques that use software on endpoints, without the difficulty of such specialized instrumentation as ours?

To probe this question quantitatively and further motivate our instrumentation, we conduct reference experiments comparing BIFOCALS to the above method of kernel interrupt-handler stamping, from Section II-B. This competing method is a more rigorous and less error-prone evolution of the typical end-host software methodology. While space constraints preclude a full description of this comparison setup, we note in passing our use of high-end multicore servers as end-hosts, running

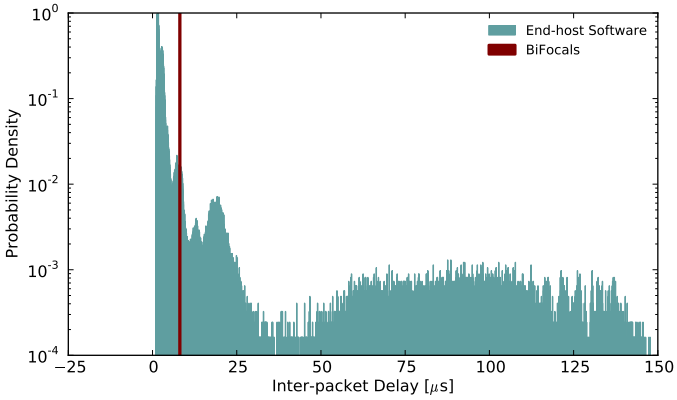



Fig. 5: Timings for network traffic across a direct optical link between the sender and receiver: BIFOCALS presents an ideally homogeneous response, while kernel interrupt-handler stamping, a stringent type of end-host software, shows severe broadening and extensive distortion.

a customized `iperf` [7] application and a modified Linux 2.6.27.2 kernel to read the time-stamp counter register (`RDTSC`) upon handling the network packet interrupt. Further, we took care to maximize `RDTSC` precision by properly inserting explicit memory barriers to serialize instructions, binding `iperf` to the same processor core, and disabling any processor power-conservation features.

Using both BIFOCALS and this reference kernel interrupt-handler stamping, we directly connect transmitter and receiver via fiber-optic link and measure the inter-packet delay. Figure 5 overlays the probability density histogram of inter-packet delays for each method and clearly depicts qualitative and quantitative distinctions between these techniques: BIFOCALS presents a perfect delta function where all packets have the same inter-packet delay, while the comparison end-host software shows severe broadening and excessive structure, with errors up to 150 μs. Any attempt to characterize the timing response across actual network paths with such a distortive tool, such as in Section V below, would create grave difficulties in differentiating the response due to the actual network path from that of the measurement tool.

## V. APPLICATIONS

The BIFOCALS architecture enables novel types of network measurements that provide unprecedented temporal precision of individual Ethernet packets. Such precision presents opportunities for both continued research as well as commercial utilization.

### A. Research applications

A number of network research questions remain unanswered with respect to the timings of packets in flight. We apply BIFOCALS to focus upon *inter-packet* timings, a fundamental metric of traffic flows from which many secondary characteristics can be derived (jitter, link capacity, etc.) [8]; such timings are independently important as a practical metric [9]–[11].

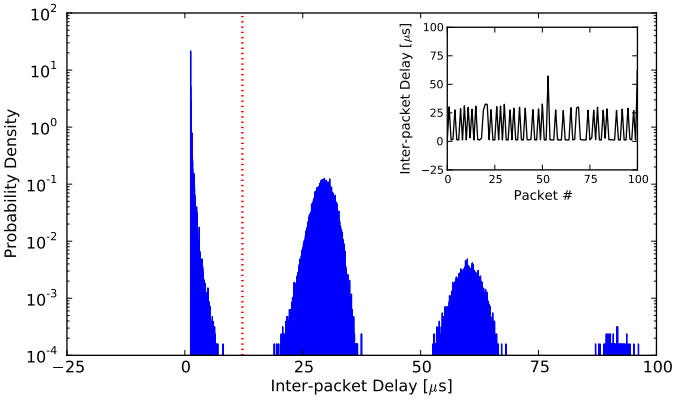



Fig. 6: Packet delay for traffic (1 Gbps data rate and 1500-byte packets) across wide-area network, with input (red dotted line) homogeneous in time and output delay distribution (blue histogram) showing stark packet chaining: Inset shows raw delay [µs] in time [packet #] for 100 arbitrary packets.

In Section IV, we characterized the representative error of other common measurement techniques, demonstrating not only the magnitude of error but also showing the introduction of specious spectral components. Indeed, such measurement errors can have significant ramifications. Accurate timings are critical, and a wide range of protocol and application research implicitly assumes that it makes sense to measure networks with software running on end-hosts (tomography, Internet coordinates, various quality-of-service schemes, etc.).

Recently [12], we applied BIFOCALS to shed light on the puzzling phenomenon of anomalous packet loss [13]; we investigated a representative wide-area network (WAN) path (15 000-km static route across eleven routers on the 10 Gbps National LambdaRail optical backbone) and observed unexpected perturbations in our traffic. In fact, we observed that WAN routers perturb packet timings so much that they compress the packets into a series of chains — packets enter the WAN homogeneous in time, with large inter-packet spacing, and exit as chains, with minimal internal packet spacing. Explicitly, for a series of 1500-Byte packets in a 1 Gbps stream, Figure 6 presents the probability density function for various inter-packet delays upon transit across this WAN. We observe this phenomenon on an otherwise lightly loaded WAN, *irrespective of input data rate*. This calls into question some basic premises of WAN paths, notably the common (although not universal) assumption that a well-conditioned packet flow will remain well-conditioned as it travels along a lightly loaded route. By comparing the scale of these effects against the errors of alternative measurement techniques (from Figure 5), we recognize that such observations would *not* be possible using common software techniques on commodity end-hosts.

### B. Commercial applications

While BIFOCALS is primarily driven by the goal of enabling reproducible network measurements through transparent instrumentation, specific commercial opportunities present themselves as well. In situations not previously possible, BIFOCALS can detect network-flow signatures, with important applications to identification of abnormalities, such as those likely present in targeted denial of service attacks. Lacking a real-time response, BIFOCALS would primarily provide audit capabilities to generate admissible evidence for criminal proceedings or civil tort suits as well as redress under business continuity insurance contracts. Similarly, verifiable audit trails are critical within the realm of electronic stock exchanges and crucial for compliance with various SEC regulations and proof of order fulfillment. These issues assume greater importance as electronic trading desks compete with one another for improvements in order fulfillment of fractions of a second, and traders formulate arbitrage strategies based on microsecond time differentials. Finally, BIFOCALS can also acquire precise network traffic metrics to formulate improved Service Level Agreements (SLAs) between corporate customers and Internet Service Providers (ISPs). A similar argument can be made in the context of peering arrangements between network operators, where BIFOCALS could contribute to mediations of technical issues and contractual agreements.

## VI. CONCLUSIONS

This work responds to the recognized need for greater precision and reproducibility in network measurements through the design and implementation of our BIFOCALS system of instrumentation. BIFOCALS enables reproducible *in-situ* network measurements, strict characterization of measurement error, transparency into the metrological tool chain, and accessibility of hardware and software. The software components of BIFOCALS are distributed at http://bifocals.cs.cornell.edu/ via a BSD license.

## REFERENCES

[1] IEEE Standard 802.3-2008, http://standards.ieee.org/getieee802/802.3.html.
[2] M. Crovella and B. Krishnamurthy, *Internet Measurement: Infrastructure, Traffic and Applications*. Wiley, 2006.
[3] Endace DAG Network Cards, http://www.endace.com/dag-network-monitoring-cards.html.
[4] Ixia Interfaces, http://www.ixiacom.com/.
[5] J. W. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo, "NetFPGA – An Open Platform for Gigabit-rate Network Switching and Routing," in *MSE*, 2007.
[6] I. P. Kaminow and T. Li, Eds., *Optical Fiber Telecommunications: IV A & IV B*. Academic, 2002.
[7] Iperf, http://iperf.sourceforge.net/.
[8] R. Prasad, M. Murray, C. Dovrolis, and K. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," *IEEE Network*, vol. 17, pp. 27–35, 2003.
[9] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot, "On Optimal Probing for Delay and Loss Measurement," in *IMC*, 2007.
[10] N. Hohn, K. Papagiannaki, and D. Veitch, "Capturing Router Congestion and Delay," *IEEE ACM T. Network.*, vol. 17, pp. 789–802, 2009.
[11] R. R. Kompella, K. Levchenko, A. C. Snoeren, and G. Varghese, "Every Microsecond Counts: Tracking Fine-Grain Latencies with a Lossy Difference Aggregator," in *SIGCOMM*, 2009.
[12] D. A. Freedman, T. Marian, J. H. Lee, K. Birman, H. Weatherspoon, and C. Xu, "Exact Temporal Characterization of 10 Gbps Optical Wide-Area Network," in *IMC*, 2010.
[13] T. Marian, D. A. Freedman, K. Birman, and H. Weatherspoon, "Empirical Characterization of Uncongested Optical Lambda Networks and 10GbE Commodity Endpoints," in *DSN*, 2010.