Exact Temporal Characterization of 10 Gbps Optical Wide-Area Network

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 and apply this tool to perform the first exact packet-timing measurements of a wide-area network ever undertaken, capturing 10 Gigabit Ethernet packets in flight on optical fiber. Through principled design, we improve timing precision by two to six orders of magnitude over existing techniques. Our observations contest several common assumptions about behavior of wide-area networks and the relationship between their input and output traffic flows. Further, we identify and characterize emergent packet chains as a mechanism to explain previously observed anomalous packet loss on receiver endpoints of such networks.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local & Wide-Area Networks—*Ethernet, High-speed, Internet*; C.4 [**Performance of Systems**]: *Measurement techniques, Performance attributes*

General Terms

Experimentation, Measurement, Performance, Reliability

Keywords

Wide-Area Network, Optical Network, 10 Gbps, Ethernet

1. INTRODUCTION

In this work, we advance the state-of-the-art in Internet measurement by presenting the design, implementation, and application of a novel form of precise instrumentation — BiFocals — that allows for the exact characterization of network traffic in flight. In addition to introducing our methodology, we employ BiFocals to empirically characterize a particular wide-area network path: a 15 000 km static route across the 10 Gbps National LambdaRail optical backbone [22]. We focus our measurements upon inter-packet timings, a fundamental metric of traffic flows from which many secondary characteristics can be derived (jitter, link capacity, etc.) [23]. Further, inter-packet timings are independently important as a practical metric [5, 2, 6, 9, 16].

Our measurements of National LambdaRail (NLR) shed light on the puzzling phenomenon of anomalous wide-area network (WAN) packet loss that we recently observed [20]: even low to moderate data rates across a WAN can provoke endpoint receiver packet loss, although the same endpoint can service such data rates within a local-area network. As we show, when a flow traverses a WAN, the routers perturb inter-packet spacing to such an extent that, within a few hops, a flow that entered the network with large, fixed inter-packet spacing has degenerated into a series of packet chains (see Figure 1). We observe this phenomenon on an otherwise lightly loaded WAN, *irrespective of input data rate*.

Internet traffic has already been shown to be bursty [14]. However, the presumption within the field is that sufficient burstiness to cause packet loss does not arise in networks with ample excess bandwidth. Thus, the key surprise in our study is the emphatic finding that this is *not* true here: an input flow, with packets homogeneously distributed in time, becomes increasingly perturbed, so that the egress flow is transformed into a series of minimally spaced packet chains. Packets within the chain have miniscule gaps, while the chains themselves are separated by huge idle gaps. For example, an ideal inflow, with constant data rate of 1 Gbps, degenerates into an extremely bursty outflow with data rates surging to 10 Gbps for short periods of time. Not surprisingly, these can trigger packet loss even in core network routers, and such surges can easily overwhelm an endpoint.

We should pause to explain why this matters. First, 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 coordi-

Figure 1: Illustration of the perturbation of the time distribution of packets between homogeneous input to, and compressed output from, a wide-area network.

IMC'10, November 1–3, 2010, Melbourne, Australia.

nates, various quality-of-service schemes, etc.). Our work provides "ground truth" properties of the WAN and contests this notion, demonstrating that previous timing characterizations could be susceptible to distortions on end-host receivers, which dwarf both fine and coarse structure.

Second, many protocols assume, more or less, that if a flow enters the network satisfying some profile or property, and background traffic along the network path is low, then it will emerge downstream with bounded properties [17]. For example, there has been a great deal of work on protocols such as Diffserv [3], which pre-negotiate resources along some route, then use edge-marking to associate packets with service classes. Overloaded core network routers preferentially retain in-profile packets while dropping out-of-profile and unmarked packets. Our work makes it clear that even if a conditioned flow were in profile at the point of ingress, within a few hops, it may be far outside of the negotiated parameters. Further, some congestion-detection schemes look at inter-packet spacing as a signal of congestion [4,1,24]; our work makes it clear that micro-bursts can develop even in a mostly idle network. Moreover, the chains generated by transitting a long path can overwhelm endpoints with bursts of high-rate data that overrun buffers. Thus, a deeper appreciation of the dynamics of deployed high-speed networks will likely be needed to arrive at the best possible application architectures.

We note that our particular findings reflect a lightly loaded 10 Gbps network, with Quality-of-Service routing features disabled and all packets traversing the identical path. The situation in other settings may be different. Just the same, our observations here suggest that similar effects are likely present in those portions of the public Internet backbone with similar architecture.

In summary, this work contributes to the science of network measurement as follows:

Instrumentation: We design and implement novel high-precision instrumentation, BiFocals, to enable the generation of extremely precise traffic flows, as well as their capture and analysis. We do *not* use computer endpoints or network adapters for traffic capture at all; rather, we generate and acquire analog traces in real-time directly off optical fiber using typical physics-laboratory test-equipment (oscilloscopes, frequency synthesizers, lasers, etc.). We combine these with off-line post-processing, so as to completely avoid 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 end-host software and two to three orders-of-magnitude relative to prior hardware-assisted solutions.

Measurements: We apply BiFocals to exactly characterize the delay distribution of network traffic after transit across two paths: a single isolated router, and a deployed, but lightly used, 10 Gbps WAN path across eleven enterprise routers and 15 000 km of optical fiber. While exceptionally precise, the measurements presented here are computationally non-trivial, requiring over **two trillion** individual samples and over **5000** processor-hours of off-line computation.

Observations: We observe that as a flow traverses a long sequence of routers, packets cluster into chains, irrespective of data rate. This finding clarifies previously unexplained observations [20] of packet loss on endpoints of a WAN. Further, it 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. Additionally, we characterize the stability of such packet chains and their probability as a function of their length. Finally, we demonstrate that these observations would *not* be possible using common software techniques on commodity end-hosts.

Outlook: We provide support for the view that only high-fidelity experimental work can provide ground truth and answer some of the contentious questions about the behavior of networks.

2. MOTIVATION

In order to exactly measure timing in network packet flows, BiFocals departs substantially from existing techniques. This section presents a taxonomy of different approaches to measurement, of increasing precision, and motivates 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) [11], corresponding to our application in Section 3. Without loss of generality, this discussion could equally apply to other Ethernet standards, such as the 1000Base-X [11] also implemented by BiFocals.

2.1 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 continuous characteristic of this symbolstream, along with the application of complex line-coding protocols and scrambling functions (described in more detail in Appendix B), provide critical guarantees for the proper performance of the transmission line.[1]

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.

Figure 2 depicts a comparison between the Physical Layer symbolstream and two hypothetical data bitstreams, motivating the likely loss of timing precision in the bitstreams. The top panel shows the actual symbolstream as transmitted on fiber, with its continuous flow of symbols of equal width at equal rate. The remaining panels demonstrate the absence of a continuous timebase once the Ethernet packets have been extracted: the middle shows the implausible scenario where the fidelity of packet timings has been main-

[1]Namely, clock recovery, DC-balance, reduction of intersymbol interference, etc.

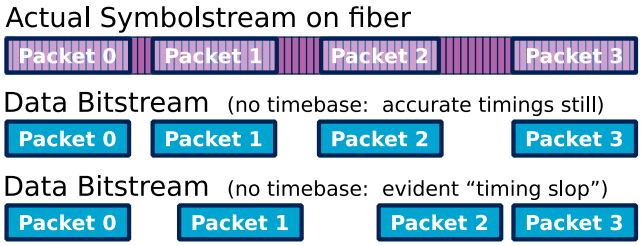


Figure 2: Comparison between an actual symbolstream on fiber (top panel) and two hypothetical examples of extracted data bitstreams, motivating both the potential (middle) and likely (bottom) loss of measurement precision due to the absence of a continuous timebase in the bitstreams.

tained, while the bottom demonstrates the most common case where timing information is perturbed. Namely, in the absence of the continuous timebase of the symbolstream, it is difficult to prevent "timing slop" of varying degree, with resulting errors in timing measurements, even though the network packets themselves are properly received and transferred to higher layers of the stack. This forms the crux of the issue of precise measurement of packet timings — the manner in which packets are time-stamped.

2.2 Sources of measurement error

With a renewed understanding of the Physical Layer of the network and the difference between symbolstreams and data bitstreams, we proceed to categorize the methods for time-stamping packets. The pertinent difference among these involves the "when" and "where" of time stamping as the packets transit the network and arrive at either the commodity end-host receiver, or our BiFocals tool, respectively. Here, 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 tools [7]. Embodying a balance among intrusiveness, cost, convenience, and accuracy, the canonical approach uses either an active or passive probe to observe traffic originating or terminating on the given end-host. 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 as in Figure 2.

Kernel interrupt-handler stamping: Rather than having the user-space software application assign time-stamps to packets upon arrival, it is possible to modify the operating system kernel to internally time-stamp packets while servicing the network-adapter interrupts that announce the arrival of each packet. 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. Irrespective of its improvement over simple user-space software packet stamping, kernel interrupt-handler stamping still suffers from the non-determinism of the asynchronous interrupt-delivery mechanism[2] and manifests significant "timing slop" of the type seen in the bottom panel of Figure 2.

Network-adapter bitstream stamping: Both commercial solutions [8, 13] and academic projects [19] have been developed to 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 these designs 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. However, as suggested by Figure 2, once they do so, the accompanying continuous timebase is lost, and the discrete packets may be subject to buffering and other error-inducing logic of FPGA gateware. (Vendor-released error estimates are addressed in Section 4.) As such, these techniques 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, as in Figure 2, the difference between the discrete nature of the data bitstream, which causes "timing slop," 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.

2.3 Instrumentation architecture

Above, we articulate 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

[2]This is the classic "arbiter problem" of asynchronous events in a clocked digital system.

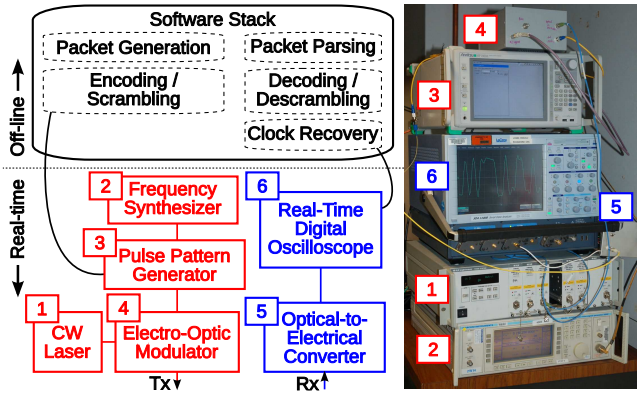


Figure 3: Diagram of BiFocals transmission and acquisition hardware and software (see detailed explication in the Appendices) connected across the network under test, with notations on the photograph of the hardware.

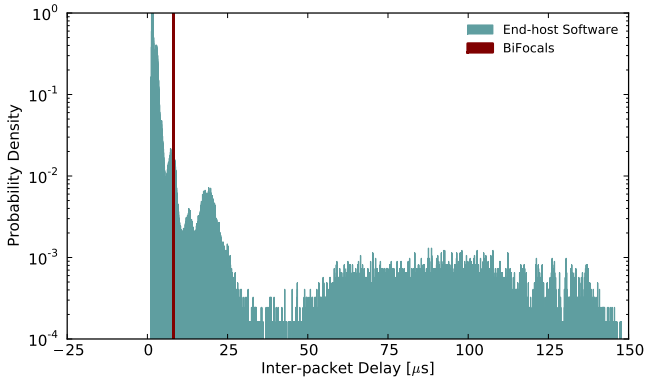


Figure 4: Timings for network traffic across a direct optical link between the sender and receiver: BiFocals presents an ideally homogeneous response, while kernel interrupt-handler stamping, a stringent type of end-host software, shows severe broadening and extensive distortion.

translates into practice, we briefly outline our instrumentation architecture here. In the Appendices, we detail the implementation and verification of BiFocals, expounding upon both the hardware foundation and software stack.

As depicted in Figure 3, 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[3] 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.[4]

2.4 Need for improved precision

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 observe in Section 3 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, which we recall 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 a customized `iperf` [12] application and a modified Linux 2.6.27.2 kernel to read the time-stamp counter register (`RDTSC`) upon handling the network packet interrupt.[5]

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 4 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 would create grave difficulties in differentiating the response due to the actual network path from that of the measurement tool. We further note that the broadening of the timings from the end-host software is sufficient even to overwhelm the coarse structure of our results, as presented in Section 3.

3. MEASUREMENTS

We apply our BiFocals instrumentation to study network transit effects for a variety of traffic flows on 10 Gbps Ethernet over fiber-optic links. This paper focuses on two scenarios: an isolated enterprise router that we use as a control case, and a high-performance, semi-private WAN path, spanning 15 000 km of the NLR [22] backbone, circumscribing the United States and traversing eleven routers. In both scenarios, we directly sample the symbolstream off the fiber and present exact characterization of the network path itself.

[3]Here, IEEE 802.3-2008 Clauses 49 (PCS) and 51 (PMA) for 10 Gbps optical Ethernet [11].
[4]In accordance with IEEE 802.3-2008 Clause 52 [11].
[5]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.

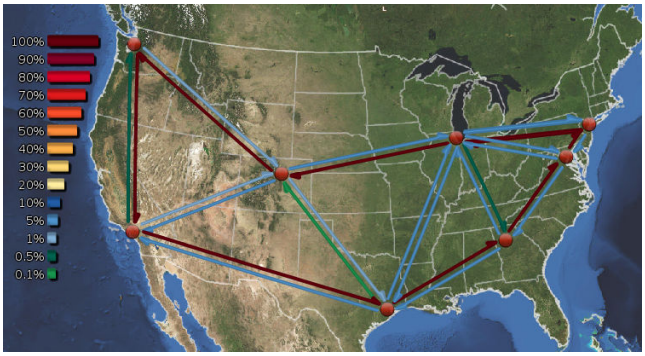


Figure 5: Map of the National LambdaRail (NLR), depicting high link utilization for BiFocals–generated traffic with a 9 Gbps data rate and 16 Mpps packet rate.

Packet size [Bytes]	Nominal data rate [Gbps]	Packet rate [kpps]	Inter-packet gap [bits]	Inter-packet delay [ns]
1500	1	82.0	109 784	12 199
1500	3	246.1	28 440	4 064
1500	9	740.5	1 304	1 350
46	1	1 755.6	5 128	570
46	3	5 208.3	1 352	192
46	9	15 625.0	72	64

Table 1: Ensembles of various packet sizes and data rates, with resulting packet rates and inter-packet gaps and delays, for network traffic homogeneous in time. Rows correspond to the six subfigures in Figures 6 and 7.

3.1 Experimental network setup

For the control router, we use a Cisco 6500, configured with IOS 12.2(33)SXI1 with tail-drop queueing, a CEF720 4-port 10-Gigabit Ethernet module, two 16-port Gigabit modules, and one Supervisor Engine 720. The 6500's centralized forwarding card forces all traffic to transit the router backplane, even though the two 10GbE ingress and egress interfaces share the same line card. While performing the control experiments, we isolated the router from any extraneous traffic.

Our main experimental path is a static route across the NLR PacketNet backbone, designed so that traffic originates and terminates at the same physical location at Cornell University. To reach the nearest NLR Point-of-Presence (POP), traffic is switched by a campus backbone switch (Cisco 6500) to an upstream campus router (Cisco 6500) in a New York City carrier-hotel and, there, routed onto the NLR backbone for subsequent transit across eight NLR routers spanning 15 000 kilometers. (Note that, in their primary failover role, neither of these campus Cisco 6500s handles commodity traffic, and they thus maintain light loads.) Figure 5 depicts the topology of our network path, as well as a real-time picture of one of our 9 Gbps traffic flows. All of these NLR optical links use Dense Wavelength Division Multiplexing technology and connect Cisco CRS-1 core routers, each with IOS XR 3.6.1[00] (tail-drop queueing), two 8-interface 10GbE line-cards, two modular service cards, and two 8-slot route processors. We note that Cisco recently upgraded the NLR infrastructure, so these routers are identical and contemporary, with passive optical components in ideal condition. We monitor the background traffic for all interfaces at each NLR POP and on both campus routers, using SNMP queries and RRDtool storage with 10-second resolution.

3.2 Measurement methodology

We characterize our traffic in terms of a number of quantities that we explicitly define here. The *packet size* refers to the size of the payload of our Ethernet packets.[6] We define the *data rate* as the data transmitted in the 64b/66b Physical Coding Sublayer (PCS) line code[7] over a given period of time, thus including the entire Ethernet packet as well as preamble and Start-of-Frame (SOF) delimiter. The packet rate is simply the number of Ethernet packets in a given period of time. Finally, in discussing the timings between packets, we define two separate quantities: *inter-packet delay* (IPD) is the time difference, or spacing, between identical bit positions in the Ethernet SOF delimiter for successive packets in the network flow, while the *inter-packet gap* (IPG) is the time between the end of one Ethernet packet and the beginning of the successive packet. More precisely, IPG is the bit-time measured from the last bit of the Ethernet Frame Check Sequence field of the first packet to the first bit of the preamble of the subsequent packet, thus including the idle (`/I/`), start (`/S/`), or terminate (`/T/`) control characters from the 64b/66b PCS line-code [11].

All experiments consist of the BiFocals apparatus generating UDP traffic at nominal data rates of 1 Gbps, 3 Gbps, and 9 Gbps for packet sizes of 46 Bytes (the minimum allowed) and 1500 Bytes (default Maximum Transmission Unit). For each data rate, the packets are homogeneously distributed in time: separated by a fixed number of 64b/66b line code bits (for example, `/I/` control characters), to exhibit identical IPG and IPD at the ingress point. Table 1 depicts the parameter space of packet size and nominal data rate, with resulting packet rate, IPG, and IPD.

We note that, while enforcing a homogeneous distribution of packets in time, the specifics of the 64b/66b line code prevent the generation of packet streams at arbitrary data rates.[8] Therefore, inter-packet gaps can only be transmitted as a certain discrete number of control characters, most of which are idles (`/I/`). The significance of this constraint is apparent below.

To measure the timings for over a million network packets for each packet size and data rate in Table 1, we had to acquire over two trillion samples from the optical fiber and process them off-line using resources exceeding 5000 processor-hours.

3.3 Results for control router

Our first experiment transmits data across a single isolated router, disconnected from any outside network. We observe neither packet loss nor packet reordering with one exception: 5% loss occurs for our smallest packet size at the highest data rate (46-Byte packets sent at 9 Gbps, corre-

[6]Each payload has an extra 25 Bytes of header and footer.
[7]The 64b/66b PCS [11] defines the specifics of the symbol-stream of the 10GBase-R Physical Layer; see Appendix B.

[8]Ethernet packets must be aligned at the start or middle of the 64-bit PCS frame.

sponding to ∼16 Million packets per second). In this lossy scenario, it is interesting to recognize that input packets have an IPG of 72 bit-times: near the minimum allowed by the IEEE 10GBase-R standard, but above that value mandated for reception and below that for transmission.[9] Thus, the flow is legal. Nonetheless, it constitutes an edge case: any higher packet-rate traffic flow would violate the 10GbE specification.

Figure 6 depicts the distributions of inter-packet delay obtained in this experiment across the control router. Each subfigure represents one of the six ensembles, corresponding to various packet sizes and data rates enumerated above in Table 1. The large panel in each subfigure shows a probability density histogram of received packet delays, all with equivalent coordinate ranges and identical logarithmic vertical scales to allow for the visual comparison of the fine structure and broadening effects in the distribution. The upper-left inset of each subfigure shows the raw inter-packet delay of the received traffic as a function of time [packet #] for a small representative segment of the million-packet ensemble. The upper-right inset of each subfigure presents an enlarged graph of the histogram, centered around the peak of the distribution. For each ensemble, the inter-packet delay value of the injected homogeneous network traffic is also marked (red vertical dashed line). It is vital to understand what a "good" response looks like: the histogram should be nearly a vertical delta function, centered at the input data flow. The extent to which the output delay distribution deviates from this ideal serves as a measure of the router-induced dynamics.

We observe that:

1. Even this single isolated router broadens the inter-packet delay distribution (initial distribution has zero width, as input traffic arrives homogeneously in time). In effect, even though the packets arrive with perfect regularity, some manage to transit the router quickly, while others are delayed briefly.

2. At higher data rates, some packets emerge in closely packed chains with the spacing between packets reflecting the *minimum* legal number of IPG bits. We observed this effect for three of the measurement ensembles, only one of whose input stream itself included minimally spaced packets (namely, 46-Byte packets at 9 Gbps).

3. Packet loss is observed in the stream with the highest packet rate (46-Byte packets at 9 Gbps).

4. A fine-grain structure is evident, reflecting the architecture of the underlying 64b/66b PCS line code.

What should one take away from this experiment?

(1) Broadening of the delay distribution: We note that all ensembles (except the stream with highest packet rate) exhibit a significant broadening of the delay distribution with respect to that of the injected packet stream. Presumably, this is due to the store-and-forward nature of the router, as the router itself is a clocked device that undergoes some internal state transitions to determine its readiness to receive or send. Further, the half-width of most of these distributions — defined here as the range for −70 dB falloff from the distribution's peak — is approximately 200 ns. For these ensembles, the half-width represents a measure of the response delay to input packets evenly spaced in time. In contrast, the stream with the highest packet rate experiences negligible broadening of its distribution (see upper-right inset of Figure 6(f)); in this case, the input packets already arrive with minimum-allowed IPG. Thus, the distribution can only broaden in one direction (to higher IPD values), and any such broadening is associated with corresponding packet loss.

(2) Formation of packet chains: As previously mentioned, the inter-packet gap minimum is actually evident in the asymmetry of the delay distributions. Specifically, the IEEE 802.3ae-2002 standard [10] mandates a separation between packets to provide the receiver some latitude in processing the incoming packet stream. This property is expressed in terms of the inter-packet gap; as such, the corresponding minimum inter-packet delay is dependent on the packet size, yet independent of data rate. For example, given 1500-Byte packets, the minimum inter-packet delay (corresponding to an IPG of 96 bit-times) is actually 1230 ns, while, for 46-Byte packets, it is 66 ns.

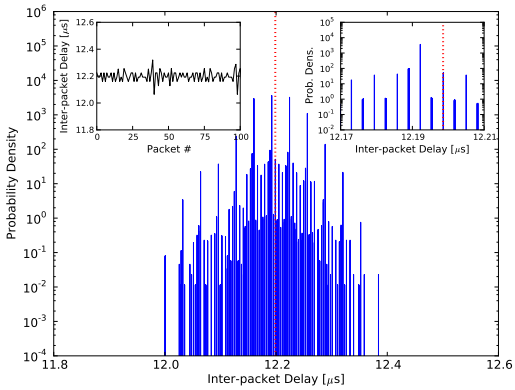
In fact, we observe precisely these minimum inter-packet delay values in most of our measurements. For ensembles with smaller input IPDs (1500-Byte packets at 9 Gbps and 46-Byte packets at 3 and 9 Gbps), it is readily observable:[10] we note extremely sharp drop-offs in the probability densities (left side of subfigures), with −50 to −80 dB suppression over 3–6 ns. We measure a minimum IPD of 1226 ns for our ensemble with 1500-Byte packets at 9 Gbps (Figure 6(e)), a minimum IPD of 64 ns for our ensemble with 46-Byte packets at 3 Gbps (Figure 6(d)), and a minimum IPD of 64 ns for the final ensemble with 46-Byte packets at 9 Gbps (Figure 6(f)). These closely agree with the above theoretical predictions for lower constraints due to minimal IPGs: the first observation is within 0.3%, and the latter two are within 3.5%.

(3) Packet loss across the router: We observe packet loss for 46-Byte packets at 9 Gbps (see Figure 6(f), especially the upper-right inset, for asymmetric broadening). We note that alternative mechanisms independently confirm the same level of packet loss for this ensemble: a careful examination of router statistics (accessed via SNMP and averaged over long periods) shows that this loss is almost completely confined to packets that are discarded in the outbound router interface before acquisition by our instrumentation. This corresponds to 4.7% packet loss. An additional 0.02% of packets exhibit errors at the inbound router interface after transmission from our instrument, though no packets without errors are discarded at this interface. Further, we note that, during measurement of this ensemble, our packet stream elicits continuous router-log warnings of excessive backplane utilization.

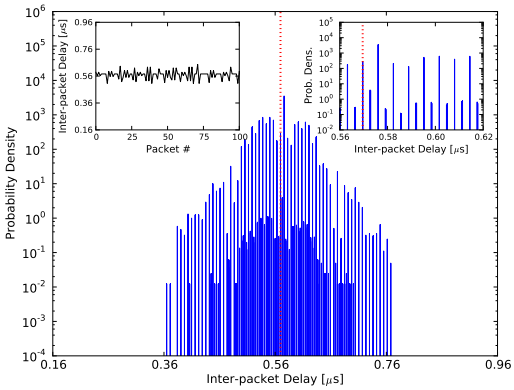
Ultimately, it is not particularly surprising that we are able to provoke router drops, even with a single flow running across a single enterprise router, when we consider that this measurement ensemble is constructed with packet rates approaching 16 million packets per second and IPGs of only 72 bits, very close to the ultimate allowed minimum of 40 bits. In particular, the 10GbE standards allows for packet

[9]See Note 4 of Section 4.4.2 of IEEE 802.3ae-2002 [10].

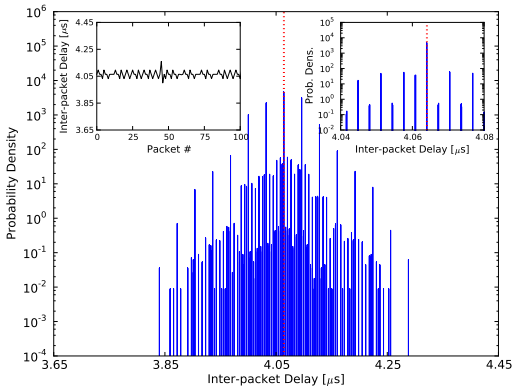
[10]For larger input IPDs (1500-Byte packets at 1 and 3 Gbps and 46-Byte packets at 1 Gbps), small sample sizes mask the phenomenon.



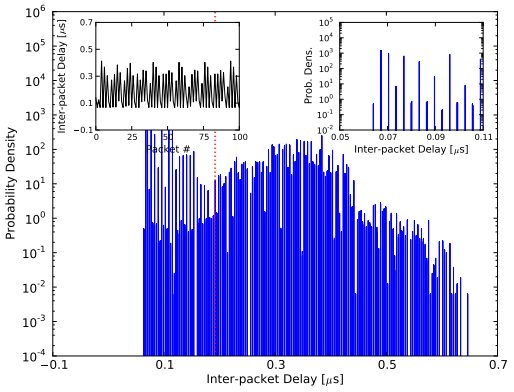
(a) 1 Gbps Data Rate (1500-Byte Packets)



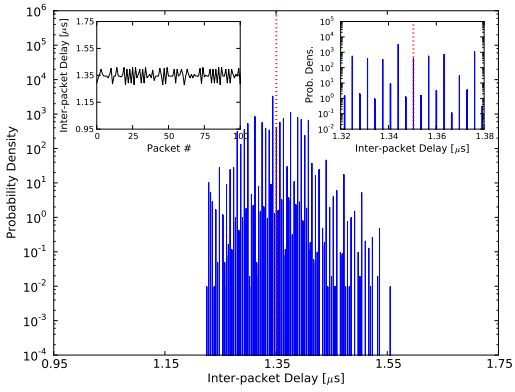
(b) 1 Gbps Data Rate (46-Byte Packets)



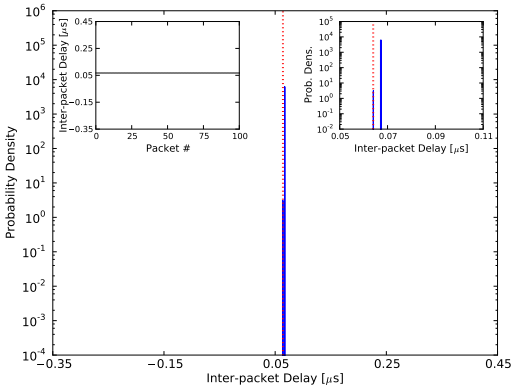
(c) 3 Gbps Data Rate (1500-Byte Packets)



(d) 3 Gbps Data Rate (46-Byte Packets)



(e) 9 Gbps Data Rate (1500-Byte Packets)



(f) 9 Gbps Data Rate (46-Byte Packets)

Figure 6: Comparison of packet delay across an isolated router (Cisco 6500) serving as the experimental control, with the input network traffic to the router perfectly homogeneous in time and the resulting delay distribution a response to transit across the router: subfigures show the probability density histograms of inter-packet delays [μs] for six ensembles (corresponding to the data rates and packet sizes enumerated in Table 1), with the delay for the input traffic marked with a dotted red line. Histogram coordinate axes are equivalent in their range (offset to center the distribution) to allow visual comparison of the broadening, and the ordinate axes are identical with a logarithmic scale to expose the fine-grained structure. Upper-left inset shows the raw delay [μs] as a function of time [packet #] for representative flows; while the upper-right inset is an enlarged view of the primary graph about its peak.

drops given that the minimum IPG of 40 bits for reception is lower than the minimum IPG of 96 bits for transmission; as a result, a router can drop packets due to this impedance mismatch between permissible IPGs.

Router drops often occur because a relentless, maximally dense stream of packets presents itself: if the router store-and-forward logic delays any packet for even the slightest amount of time, some packet will need to be dropped to compensate. This observation becomes particularly significant later for wide-area traffic; when we examine multihop data, as noted in the Introduction, we encounter a noticeable tendency for packets to form chains with minimal spacing, irrespective of the homogeneity of input packets in time. But now we recognize that packet chains can trigger router loss. It follows that long routes carrying high-speed data may be prone to loss, even in an otherwise lightly loaded network, if the receivers are not able to accept packets at the maximum-allowed rate.

(4) Fine-grained n-ary structure: Our final observation concerns the intriguing n-ary (secondary, tertiary, etc.) structure present in the packet delay histograms of all ensembles. This structure is most readily visible in the two ensembles comprised of 1500-Byte packets at 1 and 3 Gbps data rates. Per Table 1, these ensembles have the lowest packet rates and the largest inter-packet gaps. As seen in Figures 6(a) and 6(c), they manifest thirteen and fifteen local sub-peaks, respectively, superimposed atop a background distribution with typical monotonic fall-off from the central peak. These sub-peaks are substantial (100× the density of locally surrounding delay values) and uniformly distributed by the same delay offset of 32 ns.

This n-ary structure is closely related to the underlying 64b/66b Physical Coding Sublayer, as the timing separation between these peaks is almost precisely an integer multiple of 64b/66b frames; specifically, we measure 4.9992 frames. Furthermore, additional n-ary structure is seen, relating to single 64b/66b frames, as well as higher-order structure corresponding to half-frames (the former 100× more probable than the latter). We recognize that this structure results from the underlying PCS line code, which always aligns Ethernet frames with either the start or middle of a 64b/66b frame, thus explaining the fundamental half-frame period.

3.4 Results across Internet path

We next examine the same traffic flows transitting eleven routers and 15 000 km over NLR. The results appear in Figure 7, with the same six inputs of homogeneously spaced packet streams from Table 1.

Our observations here can be summarized:

1. *Irrespective of input data rate*, the delay distribution peaks at a value corresponding to a *minimum* IPD allowed by the IEEE standard, providing evidence for our contention that packets emerge from WANs in chains. So, after a sequence of 11 hops, even a 1 Gbps input flow evolves into a series of 10 Gbps bursts.

2. We observe packet loss, for the two highest packet-rate ensembles (5 and 16 Mpps: 46-Byte packets at 3 and 9 Gbps), of 1.9% and 32.4%, respectively.

3. We identify multiple secondary lobes in the delay distribution. These reflect the formation of packet chains, with lobe separation dependent upon data rate.

4. We again observe n-ary structure imposed by 64b/66b line coding.

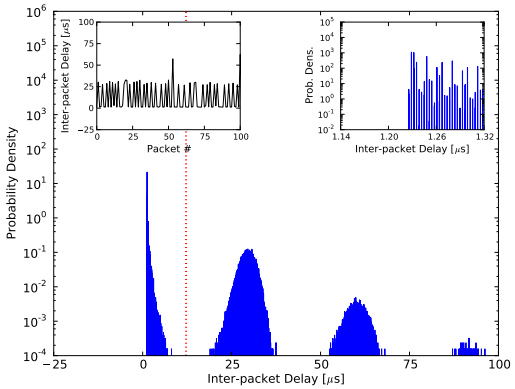
(1) Formation of packet chains for all inputs: The first, and most striking, observation in Figure 7 is that the location of the primary peak does *not* depend upon the input data rate. In fact, it closely corresponds to an inter-packet delay reflecting the minimum-allowed inter-packet gap. (As above, this IPD value is actually a function of the sum of IPG and packet size and hence not identical for ensembles of different packet size). More packets emerge from WANs with the minimum-allowed inter-packet delay, than with any other inter-packet delay. Actually, as seen more clearly in the upper-right inset of each subfigure, the distribution peak is actually offset from the lowest recorded value by a single half-frame of 64b/66b line code. We conjecture that this is related to the subtle distinction in minimum IPGs by standard (between 40 bit-times for receive and 96 bit-times for transmit).

Now, as with the control measurements, we measure the inter-packet delay of the peak for both packet sizes: 1500-Byte packets show peak delays of 1226 ns, while 46-Byte packets have peak delays of 66 ns. Both of these values correspond almost precisely to the inter-packet delay between packets of this size at their maximum data rate (approaching 10 Gbps). In fact, if we assume that these packets are separated by the minimum-allowed inter-packet gaps (96 bit-times), we can find theoretical expectations for the delays: 1230 ns for 1500-Byte packets and 66 ns for 46-Byte packets. Our observed delays are within 0.3% and 0.5%, respectively, of such theoretical values.

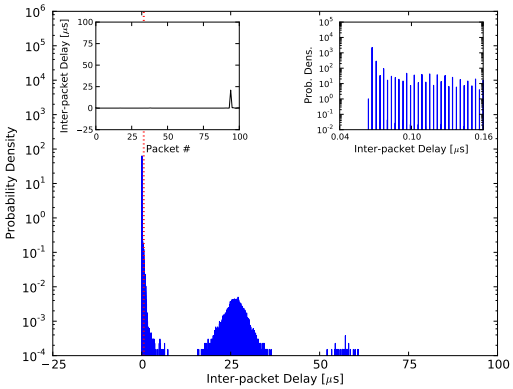
This demonstrates that, irrespective of the input data rate, network flows are compressed as they transit a series of routers on a WAN. This compression reduces the spacing between sequential packets, causing chains to form, while introducing larger gaps between these chains. In the experiments of Section 3.3 on the control router, we noted a lower bound on packet spacing that created asymmetric delay distributions, and we now see this greatly amplified by the WAN. Indeed, one might speculate that this effect results from received packets being queued and later batch-forwarded along the path as quickly as possible (at line rate). Regardless of the cause, the engineering implication is that downstream routers and receiver endpoints must be capable of lossless receipt of bursts of packets arriving at the maximum possible data rate — 10 Gbps, here. Moreover, this effect occurs even when the original sender transmits at a much lower average data rate. Failing to adequately provision any component will thus trigger loss.

This finding now clarifies a phenomenon we measured earlier [20], though could not explain: commodity servers were receiving and then dropping network packets that had been sent, at low data rates, across a 10 Gbps WAN. One can now see why the network path itself, as a collection of store-and-forward routing elements, will skew even a perfectly homogeneous packet stream towards a distribution with a dominant peak around the maximum data rate supported by the 10GbE standard. As these chains of minimally spaced packets increase in length, packet loss is inevitable, unless all network elements are able to handle continuous line-speed receipt of packets.

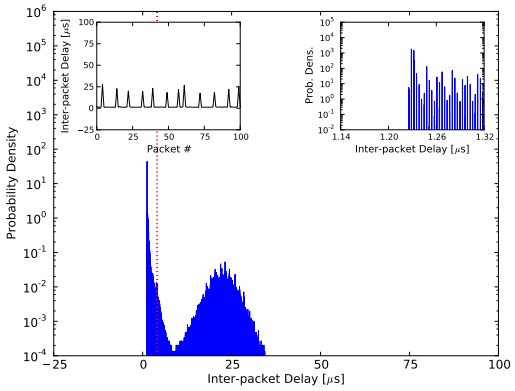
(2) Packet loss on the WAN: We indeed observe packet loss on this NLR path for ensembles with the highest input packet rates (46-Byte packets at 3 Gbps and 9 Gbps): 1.9%



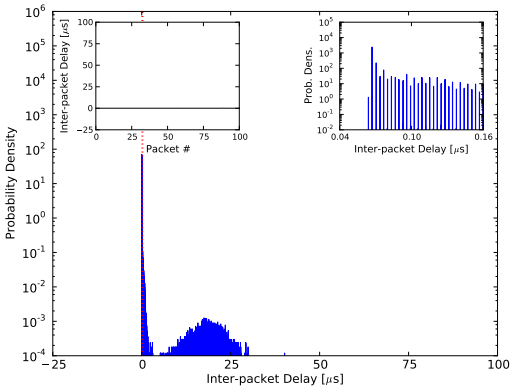
(a) 1 Gbps Data Rate (1500-Byte Packets)



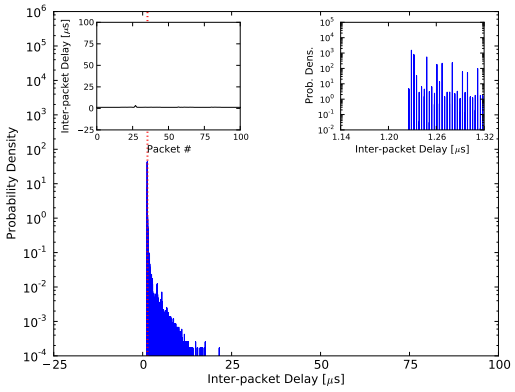
(b) 1 Gbps Data Rate (46-Byte Packets)



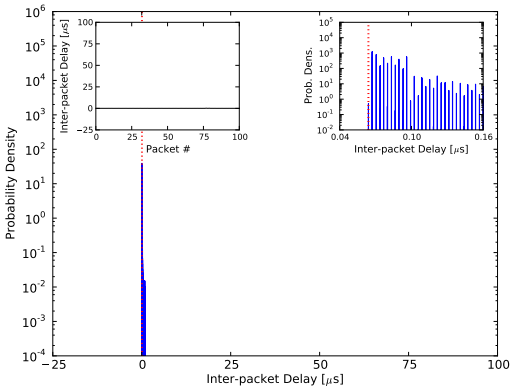
(c) 3 Gbps Data Rate (1500-Byte Packets)



(d) 3 Gbps Data Rate (46-Byte Packets)



(e) 9 Gbps Data Rate (1500-Byte Packets)



(f) 9 Gbps Data Rate (46-Byte Packets)

Figure 7: Comparison of packet delay across a lightly loaded 10 Gbps Internet path spanning 15 000 km of the National LambdaRail optical backbone and eleven routing elements (Cisco CRS-1 and 6500 routers), with the input traffic to the path perfectly homogeneous in time and the resulting delay distribution a response to transit across this wide-area network: subfigures and insets as specified earlier in Figure 6.

and 32.4%, respectively.[11] As noted above for the control router, it appears that this loss occurs as incoming traffic exceeds the backplane capacity of the routers and outbound buffers overflow, dropping packets before they can continue on the network path. We conjecture that the rate of loss might be related to the number of routing elements along the WAN path.

(3) Secondary lobes in delay distribution: Our third observation concerns the delay of the secondary lobes in Figure 7. For individual histograms with multiple lobes, the peaks are equidistant (separated by 30 μs in Figure 7(a), for example). For each packet size, we observe a negative linear correlation between data rate and peak separation (estimating the lobe location for ensembles without a distinct secondary lobe).

(4) Fine-grained* n*-ary structure: The final observation here mirrors the fourth point discussed for the control router. Once again, we see secondary (and tertiary) fine-grained structure atop the primary probability density distribution. While more difficult to discern for the ensembles with 46-Byte packets, we can readily measure it for the three ensembles with 1500-Byte packets. As above, we note a series of interwoven sub-peaks, with probability densities 100× above their surrounding background values; these sub-peaks are separated by delay values of 32 ns, with five tertiary peaks embedded between each. As in Section 3.3, this reflects the PCS substrate and its framing protocol.

3.5 Analysis of representative ensemble

While we report our measurements above for both control and Internet paths, we now further evaluate and analyze those data. Figures 6 and 7 present probability density histograms of inter-packet delays, showing the *statistical* behavior of network packet streams in our ensembles, but concealing the time correlations between neighboring packets. Here, we discuss these correlations and associate given delays with particular packets within an ensemble trace. Further, we connect our analysis to an investigation into the background traffic on the NLR backbone comprising our Internet path. Due to space constraints, the subsequent analysis examines only one such ensemble in detail: the 1500-Byte packet stream transmitted at a 1 Gbps data rate, described in Table 1 and presented in Figure 7(a), which is representative of traffic flows in this environment.

We find that:

1. Our results show *self-similar* behavior — measurements at differing time scales exhibit the same statistical properties — a recognized and critical property of network traffic [18].

2. Packet chains manifest similar characteristics irrespective of their particular definition; namely, chains of increasing length occur with exponentially less frequency.

3. The statistical distribution of inter-packet delays is relatively insensitive to background traffic.

We first must ensure that the statistical behavior seen in Sections 3.3 and 3.4 is not merely an anomaly, caused,

[11]Though NLR is a production network used by scientists nationwide, its routers might not be optimized relative to this particular sort of stress-test.

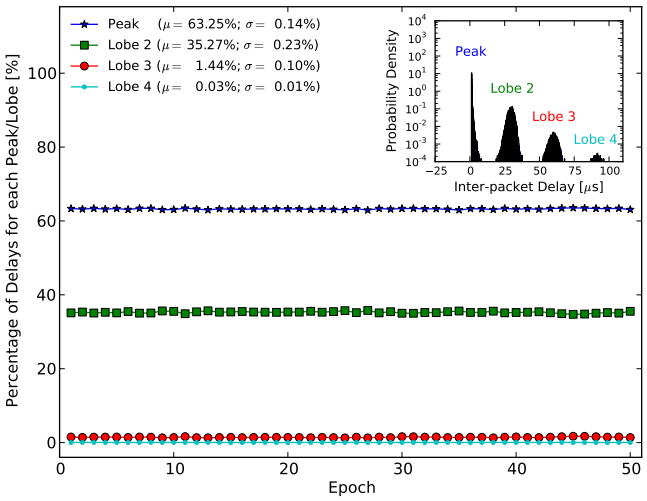


Figure 8: Self-similarity of inter-packet delays for network traffic: as a function of time-epoch, the percentage of delays associated with each peak or lobe of our 1 Gbps, 1500-Byte traffic (from Figure 7(a), reproduced as inset here, with labels); legend provides the mean and standard deviation.

for example, by the aggregation of distinct regimes of behavior in time (here, four separate regimes of different delays). To show the self-similarity of our observations, we start by proposing a metric of interest for our ensemble trace and by dividing this trace into some number of contiguous time-epochs, each containing the same number of packets. We then verify that this metric remains constant across all time-epochs. Figure 8 shows such a process: we employ fifty time-epochs (about 20 000 packets each) and compute, as a metric, the percentage of delays associated with the peak and each of the three lobes in Figure 7(a), reproduced here as inset. We immediately confirm that our metric holds constant across all epochs. Additionally, we report the mean and standard deviation of these values across epochs and note the relative proportion of delays among peak and lobe elements: $63.25 \pm 0.14\%$ of delays are in the peak, $35.25 \pm 0.23\%$ in the second lobe, $1.44 \pm 0.10\%$ in the third, and only $0.03 \pm 0.01\%$ correspond to the smallest fourth lobe. Though not illustrated here, we repeat this process for epochs of six alternate sizes, with each epoch containing between 2000 and 50 000 packets; in all cases, we observe identical mean percentage of delays for the peak and all three lobes, as well as similar constancy in time. This strongly affirms the self-similarity of the measured delays over the entire ensemble and provides assurance that our conclusions are not an artifact of the time or resolution of our measurements.

We now investigate the connection between the histograms of Section 3.4 and recognizable packet chains. We first define such a chain of packets by selecting a minimum inter-packet delay, below which packets are classified as a single packet chain. We set a delay threshold (1.25 μs) close to the minimum theoretical inter-packet delay (1.23 μs for this ensemble). Figure 9 shows the packet chains that emerge after transit across the NLR optical backbone; the figure presents the probability density for the occurrence of packet chains of prescribed lengths. For example, the histogram reveals that chains with ten packets occur approximately once per

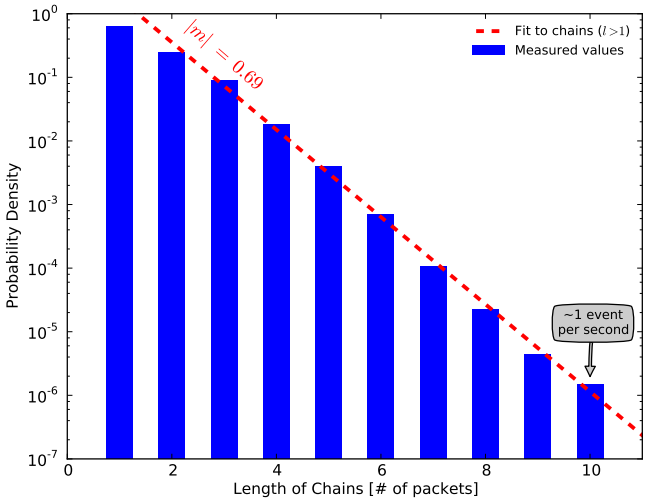


Figure 9: Transformation of a temporally homogeneous network packet stream into a series of packet chains, of varying lengths, after transit across the NLR optical backbone: probability density of the resulting packet chains as a function of chain length, showing exponential suppression of chains of increasing length (same ensemble as Figure 8).

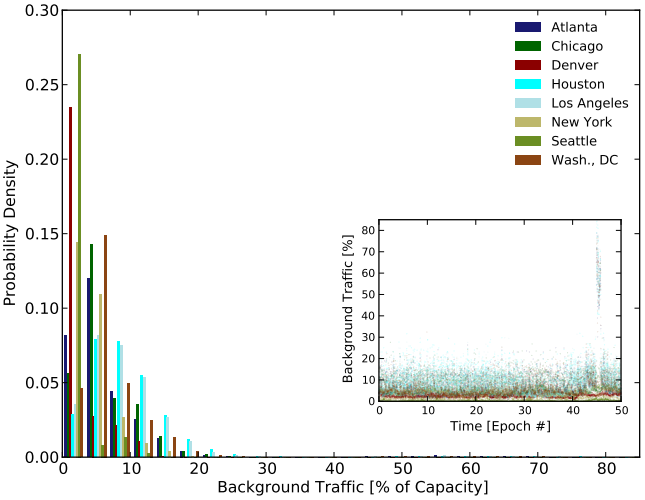


Figure 10: Background traffic across NLR routers during measurement: probability density of link utilization of background traffic transiting outbound NLR backbone interfaces, for each of eight NLR routers along our WAN path, simultaneous to our injected traffic at given time of measurement (same ensemble as Figure 8); inset depicts time-series, rather than statistical description, of traffic.

second. We observe the exponential suppression of packet chains of increasing length, $P(l) \propto 10^{-|m|l}$, and extract[12] the exponential coefficient, $|m| = 0.69$. This fit allows us to extrapolate the probability of chains longer than those we capture: 15-packet chains occur every forty-five minutes, while 20-packet chains are seen only once per ninety days, conveying the relative rarity of longer chains for this scenario. Finally, we analyze the sensitivity of packet chain formation to our particular choice of delay threshold. We increase the threshold by almost an order of magnitude, to a value of 12.20 μs, equivalent to the inter-packet delay of the input packet stream. Though not shown, we observe behavior almost identical to that of Figure 9, with an exponent only ∼15% lower ($|m| = 0.59$), thus confirming that chains are robust and quite insensitive to how they are defined. Moreover, this also reinforces the magnitude of the separations between the chains, compared to that between packets within a chain. With such robust chain formation, further predictions become possible: one can combine this data with separate knowledge of the effect of packet chains on end-hosts attached to the WAN,[13] in order to develop expectations of packet loss and service reliability of the end-to-end path, including these attached endpoints.

Finally, we examine the influence of background traffic along NLR backbone links on our statistical observations of packet chains. First, we define *background traffic*: for each NLR POP, it is the difference between the total data rate of outbound traffic on that POP's interface along our Internet path, and the data rate of the inbound traffic we inject from Cornell into NLR's New York POP. For each of the NLR POPs, Figure 10 shows the probability density of the background traffic, in percentage utilization of the full 10 Gbps link capacity; the inset depicts the time-series of the traffic. The mean background traffic is quite low, in fact, registering only ∼6% of the link capacity. The sole exception is a brief surge in background traffic to ∼60% of link capacity (6 Gbps). Representing an order-of-magnitude more traffic and lasting 90% of a time-epoch (Epoch 46 in Figure 8), this traffic surge shows no effect on the statistical distribution of inter-packet delays, nor on their resulting packet chains. Such apparent independence, between the statistical properties of Section 3 and the background traffic, suggests that the explanation for our observations might rely on other factors. Still, our analysis does not yet enable us to precisely identify the cause of packet chains, and their emergence remains an open, and fascinating, research question.

This analysis shows that the exact characterization of packet timings provides meaningful insights into the behavior of Internet paths. While reliable and reproducible instrumentation and measurements are critical foundations, further investigations are clearly warranted here: varying WAN path lengths, modifying router model selection and configuration, controlling explicit background-traffic scenarios, and more. BiFocals provides a framework to conduct these in an empirically rigorous manner.

4. RELATED WORKS

Quantitative measurements of network traffic, both in the wild on the Internet and in isolation in the laboratory, hold a long established role within the systems and networking communities [7].

Our results showcase the application of BiFocals to exactly capture the time characteristics of network packets in flight — a landmark in Internet timing measurement. BiFocals achieves this precision by introducing on-fiber symbolstream time stamping to the taxonomy of measurement

[12] We fit $l > 1$ as, by definition, $P(l = 1)$ corresponds to the probability density of non-chain packets.

[13] For example, knowledge of chains' interactions with network-adapter buffers or other hardware or software specifics.

methods described in Section 2.2. Here, we compare our technique with the hardware-assisted method of network-adapter bitstream time stamping, used in the academic NetFPGA project [19] and the commercial Ixia [13] and DAG [8] frameworks. However, unlike BiFocals, these measurement techniques are simply unable to deliver the exact timings of packets, irrespective of their other benefits or drawbacks.

NetFPGA provides Verilog gateware and software for accurate packet transmission and acquisition, using rate limiters, delay modules, and time stamping at the gate-array level. Packet processing is done off-line, relying upon `PCAP` to generate and save traffic traces. Support for Ethernet standards differs from ours: NetFPGA currently implements 1000Base-T (with parallel symbolstreams at 125 MBaud), while BiFocals delivers the 100-fold higher symbol rate of 10GBase-R (10.3125 GBaud serial symbolstream). Further, our architecture allows for flexible interchange of standards (for example, we also support 1000Base-X) and seamless progression to higher-performance Ethernet (40GbE, 100GbE, etc.), while requiring minimal rewriting of our software and replacement of only certain individual pieces of test equipment (faster optics and electronics). In contrast, to adopt additional standards with higher symbol rates, NetFPGA would arguably require significant gateware revision, as well as likely a complete re-engineering of the data-path.

Similar to NetFPGA, Ixia network service modules use an FPGA-based domain-specific implementation to accurately generate and capture packets with high fidelity. (For example, the Ixia 10GBase-R solution possesses a resolution of 20 ns, yet this precision is still at least 200-fold worse than ours.) Ixia excels in providing a complete turn-key solution for the commercial market with strong support for application-load and content-processing modules. However, Ixia is a closed, expensive (∼$1M) platform with capabilities that cannot be independently expanded.

Endace's DAG (Data Acquisition and Generation) Network Monitoring Cards offer functionality comparable to that of Ixia, with lower cost and slightly better advertised time resolution (7.5 ns), although they lack Ixia's diversity of application testing modules. While similarly proprietary, DAG offers existing options for link monitoring, capacity planning, and forensics analysis. All told, DAG appears useful to the network operator, but less appropriate for precise measurement or network research.

Unlike such domain-specific technologies, our BiFocals instrumentation relies solely on conventional physics test equipment — oscilloscopes, pattern generators, lasers, etc. — to achieve exact timing characterization. Though our test equipment is not inexpensive (∼$200k), it is general in purpose and widely available within academic and research communities. In fact, when not serving as part of the BiFocals work, our hardware components are used for other Cornell University research, and no additional physics test equipment needed to be purchased to build our instrumentation. We envision that a similar environment of re-use would hold at other academic and research institutions and thus enable economical reproduction of this work.

5. CONCLUSIONS

BiFocals responds to the recognized need for principled, precise, and reproducible measurements [21], especially in the domain of packet timings for high-speed networks. Our instrumentation achieves remarkable levels of temporal precision to enable the exact characterization of the timing of network packets in flight on fiber. For 10GbE measurements, we achieve up to six orders-of-magnitude improvement in timing precision over existing end-host software. This is attained by eschewing computer endpoints and network adapters and instead generating and acquiring the symbolstream directly off optical fiber with real-time physics test equipment and off-line software.

Using BiFocals, we accomplish what we believe to be the most precise timing measurements ever made for various packet flows in relatively simple scenarios: through a single isolated router, and also across a statically routed wide-area network, spanning eleven routing elements and 15 000 km of the National LambdaRail optical backbone. We explore a range of traffic patterns, with packet sizes from 46 to 1500 Bytes, data rates up to 9 Gigabits per second, and single-flow packet rates up to 16 million packets per second.

Our instrumentation reveals phenomena previously obscured by the relatively imprecise methods available to classic network performance studies. In particular, we show that — irrespective of the input data rate — routers introduce burstiness. Thus, downstream routers and receiver endpoints must be prepared to accept extended chains of packets with the minimum legal 10GbE packet spacing, or, equivalently, the highest possible instantaneous data rate. Commodity receiver endpoints will often drop packets from flows with such bursts of minimally spaced packets. In fact, our prior observation of this anomalous behavior served originally to motivate this study [20].

In analyzing our data set, we probe the self-similarity of our results and exclude anomalous explanations for the observed probability densities of inter-packet delays. Further, we validate the stability of our definition of packet chains and find that chains of increasing length occur with exponentially less frequency. Finally, we comment upon the relative insensitivity of our distribution of packet delays to the background traffic along our WAN path.

All data collected here are available to the scientific community at `http://bifocals.cs.cornell.edu/`. Similarly, we freely distribute the software component of BiFocals under a two-clause BSD license.

6. ACKNOWLEDGEMENTS

We thank Jennifer Rexford for her insightful suggestions, David Bindel for his informative discourse, Robert Broberg and his colleagues at Cisco for their technical knowledge, and our referees for their anonymous, though no less thoughtful, input. We further recognize the engineers who helped establish and maintain the network infrastructure on which we performed these experiments: Eric Cronise, Dan Eckstrom, and Ed Kiefer (Cornell Information Technologies); Greg Boles, Brent Sweeny, and Joe Lappa (National LambdaRail); Scott Yoest and Larry Parmelee (Cornell Computer Science Technical Staff). Funding for this work was provided by the Air Force Research Laboratory (AFRL) and the National Science Foundation (NSF).

7. REFERENCES

[1] Anderson, T., Collins, A., Krishnamurthy, A., and Zahorjan, J. PCP: Efficient Endpoint Congestion Control. In *NSDI* (2006).

[2] Baccelli, F., Machiraju, S., Veitch, D., and Bolot, J. On Optimal Probing for Delay and Loss Measurement. In *IMC* (2007).

[3] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. RFC 2475: An Architecture for Differentiated Services, 1998.

[4] Brakmo, L. S., and Peterson, L. L. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE J. Sel. Area. Comm. 13* (1995), 1465–1480.

[5] Carter, R. L., and Crovella, M. E. Measuring Bottleneck Link Speed in Packet-Switched Networks. *Perform. Evaluation 27–28* (1996), 297–318.

[6] Choi, B., Moon, S., Zhang, Z., Papagiannaki, K., and Diot, C. Analysis of Point-To-Point Packet Delay In an Operational Network. *Comput. Netw. 51* (2007), 3812–3827.

[7] Crovella, M., and Krishnamurthy, B. *Internet Measurement: Infrastructure, Traffic and Applications*. Wiley, 2006.

[8] Endace DAG Network Cards. `http://www.endace.com/dag-network-monitoring-cards.html`.

[9] Hohn, N., Papagiannaki, K., and Veitch, D. Capturing Router Congestion and Delay. *IEEE ACM T. Network. 17* (2009), 789–802.

[10] IEEE Standard 802.3ae-2002. `http://grouper.ieee.org/groups/802/3/ae/`.

[11] IEEE Standard 802.3-2008. `http://standards.ieee.org/getieee802/802.3.html`.

[12] Iperf. `http://iperf.sourceforge.net/`.

[13] Ixia Interfaces. `http://www.ixiacom.com/`.

[14] Jiang, H., and Dovrolis, C. Why is the Internet Traffic Bursty in Short Time Scales? In *SIGMETRICS* (2005).

[15] Kaminow, I. P., and Li, T., Eds. *Optical Fiber Telecommunications: IV A & IV B*. Academic, 2002.

[16] Kompella, R. R., Levchenko, K., Snoeren, A. C., and Varghese, G. Every Microsecond Counts: Tracking Fine-Grain Latencies with a Lossy Difference Aggregator. In *SIGCOMM* (2009).

[17] Kurose, J. On Computing Per-session Performance Bounds in High-Speed Multi-hop Computer Networks. In *SIGMETRICS* (1992).

[18] Leland, W. E., Taqqu, M. S., Willinger, W., and Wilson, D. V. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE ACM T. Network. 2* (1994), 1–15.

[19] Lockwood, J. W., McKeown, N., Watson, G., Gibb, G., Hartke, P., Naous, J., Raghuraman, R., and Luo, J. NetFPGA – An Open Platform for Gigabit-rate Network Switching and Routing. In *MSE* (2007).

[20] Marian, T., Freedman, D. A., Birman, K., and Weatherspoon, H. Empirical Characterization of Uncongested Optical Lambda Networks and 10GbE Commodity Endpoints. In *DSN* (2010).

[21] Mytkowicz, T., Diwan, A., Hauswirth, M., and Sweeney, P. F. Producing Wrong Data Without Doing Anything Obviously Wrong! In *ASPLOS* (2009).

[22] National LambdaRail. `http://www.nlr.net/`.

[23] Prasad, R., Murray, M., Dovrolis, C., and Claffy, K. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network 17* (2003), 27–35.

[24] Wei, D. X., Jin, C., Low, S. H., and Hegde, S. FAST TCP: Motivation, Architecture, Algorithms, Performance. *IEEE ACM T. Network. 14* (2006), 1246–1259.

APPENDIX

A. HARDWARE FOUNDATION

We reference Figure 3 of Section 2.3 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 [15] 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 (here: ILX Lightwave 79800E centered at $\lambda = 1555.75$ nm) and an electro-optic modulator (EOM, here: JDS Uniphase OC-192 Modulator). The CW laser outputs a constant light intensity, which is switched on and off by the EOM based upon a supplied electrical signal. This electrical impulse to the EOM is provided by the combination of a precise frequency synthesizer and a pulse pattern generator (PPG). The frequency synthesizer (here: Marconi 2042 Low Noise Signal Generator) is tuned to 5.15625 GHz, which is doubled (with a Narda 4453 frequency doubler) to 10.3125 GHz[14] to seed the clock of the PPG (here: Anritsu MP1800A / MU181020A with Picosecond Pulse Labs Model 5865 broadband RF amplifier). 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, at a symbol rate of 10.3125 GBaud (as determined by the clock seed). The PPG output drives the EOM, resulting in an optical waveform with high light intensity representing "1" bits and no light intensity representing "0" bits. The amplitude and bias of the electrical signal can be adjusted to ensure maximal light intensity for the 1 bits and minimal for the 0 bits (here: this EOM can achieve 20 dB extinction of "on" to "off"). 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 (here: Discovery Semiconductor DSC-R402) and a real-time digital oscilloscope (here: LeCroy SDA 11000-XL) 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 subsequently transfered for later off-line deconstruction by our software stack.

[14] In accordance with the nominal symbol rate specified in Table 52–12 of IEEE 802.3-2008 [11].

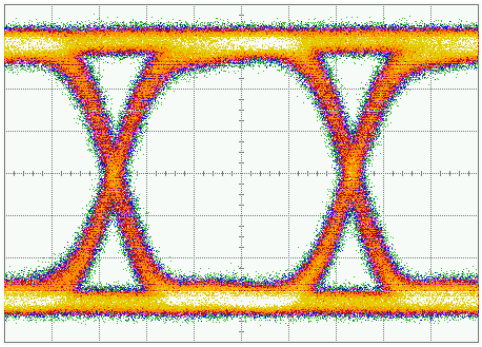


Figure 11: Eye diagram of the optical signal transmitted by BiFocals hardware: shows a large, open eye with negligible noise or jitter and conformance with 10GBase-R specifications for optical transmission power, rise time, eye mask, etc. Horizontal scale is 20 ps/div and vertical is 80 μW/div.

To validate the hardware instrumentation of BiFocals, we ascertain the quality of the transmitted optical signal (independent of the hardware receiver) by measuring its eye diagram, a standard measurement in digital communications. We connect the optical output of our transmitter directly to a wideband sampling oscilloscope (here: Agilent 86100A with 30 GHz 86109A optical module) and trigger it at the clock frequency of our transmitter, which overlays the sequence of samples in time, synchronized at a fixed point in the symbol frame, as shown in Figure 11. The degree to which the eye is "open" (in both time and amplitude) provides a direct metric of the quality of the signal. Vertical eye closure (indicative of high noise levels or weak signal) leads to ambiguity in the digitization of analog signals, while horizontal eye closure (from timing jitter or pulse walkoff) can result in symbol errors due to mis-sampling on a transition. 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.[15]

B. SOFTWARE STACK

With the description of the hardware layer of our BiFocals instrumentation complete, we now discuss the off-line software stack (shown also in Figure 3) that internalizes the intelligence and semantics of the Physical Layer and all other network layers. The transmission software stack involves two primary stages (analogous to PHY and MAC/IP/UDP layers): first, the creation of a sequence of discrete Ethernet packets; and second, the insertion of these packets into a continuous 64b/66b Physical Coding Sublayer (PCS) symbolstream suitable for transmission on the physical fiber media. Generation of a sequence of packets, in compliance with appropriate protocols, is a straightforward task of software engineering: our current implementation incorporates Ethernet, IP, UDP, and various application payloads.

The integration of these discrete packets into a standards-compliant 64b/66b PCS symbolstream, however, involves a number of subtleties. PCS wraps 64 bits of symbols into a 66-bit frame, resulting in a 3.125% overhead and requiring a 10.3125 GBaud symbol rate for a 10 Gbps data rate. The 64 bits are 8 octets of data or control information, while the two bits that delineate the frame ensure a signal transition every frame (as only "01" and "10" are allowed), easing clock recovery and frame synchronization. The content of these 64b/66b frames are mandated by the control code and data block formats allowed,[16] especially with respect to *Start* (`/S/`), *Terminate* (`/T/`), and *Idle* (`/I/`) control codes. The resulting sequence of the 64 payload bits from each frame is sequentially fed through a multiplicative self-synchronizing scrambler, defined by the polynomial $G(x) = 1 + x^{39} + x^{58}$ [11], to ensure that the resulting signal has desired DC-balance characteristics, irrespective of transmitted data. Self-synchronization ensures that the descrambler of the receiver does not need knowledge of any given initial state to implement $G(x)$.

Finally, when inserting discrete packets into the symbolstream, our encoder must minimize any boundary effects of the finite-length PPG memory depth by: (1) ensuring symbolstreams are integer numbers of 64b/66b frames (for PCS frame-sync); (2) positioning `/I/` codes at the start of the symbolstream (to mitigate the initial 58 bits necessary for self-synchronization of the descrambler at the beginning of the symbolstream); (3) maintaining identical numbers of `/I/` codes, as desired, across the periodic boundary of the symbolstream as it wraps around (for complete homogeneity of packets in time, required in this study); and (4) maximizing the length of the symbolstream, so as still to be able to fit it within the given finite PPG memory depth. To sum, our 64b/66b software encoder must comply with these requirements and implement all necessary functionality to generate a valid 64b/66b PCS symbolstream that can be understood by any deployed 10GBase-R implementation, such as commercial routers and switches.

In comparison with our software stack for symbolstream generation, our software for symbolstream deconstruction requires an additional clock recovery and waveform digitization step before conducting the inverse of the functions described above. (Note, on the transmission side, the hardware, rather than the software, handles the clock generation.) Accurate clock recovery is non-trivial and computationally expensive, but necessary for subsequent success at decoding symbolstreams and parsing packet streams. Our software clock recovery involves mathematical transformations (Fast Fourier Transforms, convolutions, etc.) of the acquired sampled waveform and a number of intermediary modules to iteratively refine our estimate for the symbol period associated with the actual symbol rate of the sender. These numerical calculations consume the bulk of the 5000+ processor-hours required for this data set.

The next two stages of our acquisition software provide 64b/66b PCS decoding and descrambling, as well as higher layer packet parsing, in exact analogue to the transmission scenario. Finally, we use the symbol period extracted during clock recovery, T_S, and convert the measured bit offsets, between successive packet Start-of-Frame (SOF) delimiters, into inter-packet delay times. Here, we measure $T_S = 96.9710 \pm 0.0001$ ps, with accuracy determined by BiFocals acquisition components.

[15] See Table 52–12 in IEEE 802.3-2008 [11].

[16] See Figure 49–7 of the IEEE 802.3-2008 standard [11].