Characteristics of Fragmented IP Traffic on Internet Links

Colleen Shannon, David Moore, k claffy

Abstract— **Fragmented IP traffic is a unique component of the overall mix of traffic on the Internet that has not been well studied. Many assertions about the nature and extent of fragmented traffic are based in folklore, rather than measurement and analysis. In this paper, we examine the behavior of measured fragment traffic and compare those results with commonly cited beliefs.**

We analyze characteristics of fragmented traffic, and examine the causes of IP packet fragmentation. The effects of NFS, streaming media, networked video games, and tunneled traffic are quantified, as well as the prevalence of machines whose improper configurations were causing excessive amounts of fragmented traffic.

To understand the prevalence, causes, and effects of fragmented IP traffic, we have collected and analyzed seven multi-day traces taken from three sources. These sources include a university commodity access link, a highly aggregated commercial exchange point, and a local NAP.

Keywords— **fragmentation, fragment, CoralReef, TCP/IP**

I. Introduction

The Internet protocol (IP) was designed to facilitate communication between heterogenous networks, and thus implement a lowest-common-denominator, a protocol with which to facilitate communication among a plethora of machines with differing architectures, operating systems, and applications, connected by varying routes, paths, and protocols. Thus IP must address the problem of different networks supporting varying maximum sizes for transmitted packets. While it is trivial to move packets from a network with a smaller MTU (maximum transmission unit) to a network with a larger MTU, the reverse is problematic. To resolve this difficulty, IP packets were allowed to be fragmented as they traverse the network: the router breaks the datagram up into pieces, each of which receives an IP header that is a replica of the original IP header. Thus each fragment has the same identification, protocol, source IP, and destination IP as the original datagram. To distinguish fragments, each has its offset field set to the distance, measured in 8-byte units, between the beginning of the original datagram and the beginning of that particular fragment. So the first fragment has its offset set to 0, the second fragment has as its offset value the payload size of the first fragment, and so on. All of the fragments except the last have the 'more fragments' bit set, so that the end host can correctly reassemble the fragments into the original IP datagram. Each fragment is generally the size of the MTU of the subsequent link, minus the size of the header that is added to each fragment. Each fragment is then sent out into the network and is handled like all other IP packets as it is routed towards its destination. By providing an automatic network mechanism for handling disparate MTU sizes, IP allowed end hosts to exchange traffic with no knowledge about the path between them.

In their 1987 paper "Fragmentation Considered Harmful," Kent and Mogul [1] established that packet fragmentation is a suboptimal method of handling packets as they traverse a network. Some of their assertions no longer apply and researchers have shown that in certain specific controlled circumstances fragmentation can improve performance [2]. For example, modern routers have sufficient buffering capabilities to receive back-to-back packets, and current computers generally have sufficient buffer space to reassemble even very large packets. However, for other reasons presented in the Kent and Mogul paper, its conclusion remains valid for wide area transport: packet fragmentation can be detrimental to performance. First, an intermediate router must perform the fragmentation. This is a CPU-intensive operation that impedes the performance of the router. Then the additional packets increase the load on all routers and networks between the fragmenting router and the end host. Finally, once the fragments reach their destination, they must be reassembled by the end host. The loss of any fragment results in the destination dropping the entire packet. Thus, although much has changed in the intervening thirteen years, IP packet fragmentation is still "considered harmful".

In the interim between the Kent and Mogul paper and the present, many theories about the causes and effects of fragmented IP traffic have come to be taken as fact. First and foremost is the assertion that fragmented traffic doesn't exist. Others acknowledge the existence of fragmented traffic on LANs, but believe its scope to be limited such that it is not present on backbone links. Additional commonly held beliefs include that only UDP traffic is fragmented, that NFS is the source of all fragmented packet traffic, that fragmented IP traffic on the whole is decreasing, and that certain misconfigurations are causing an increase in fragmented traffic. Clearly these beliefs cannot all be true, since several are mutually exclusive. For example, fragmented traffic cannot be simultaneously nonexistent and composed of UDP packets. While one recent publication suggested that IP packet fragmentation is on the increase [3], the rest of this fragment folklore has no basis in current network measurements.

Yet IP packet fragmentation continues to play a small but vital role in facilitating communication between hosts on the Internet. The proliferation of protocols that send

All authors are with CAIDA, San Diego Supercomputer Center, University of California, San Diego. E-mail: {cshannon,dmoore,kc}@caida.org.

Support for this work is provided by DARPA NGI Contract N66001-98-2-8922 and NSF grant NCR-9711092.

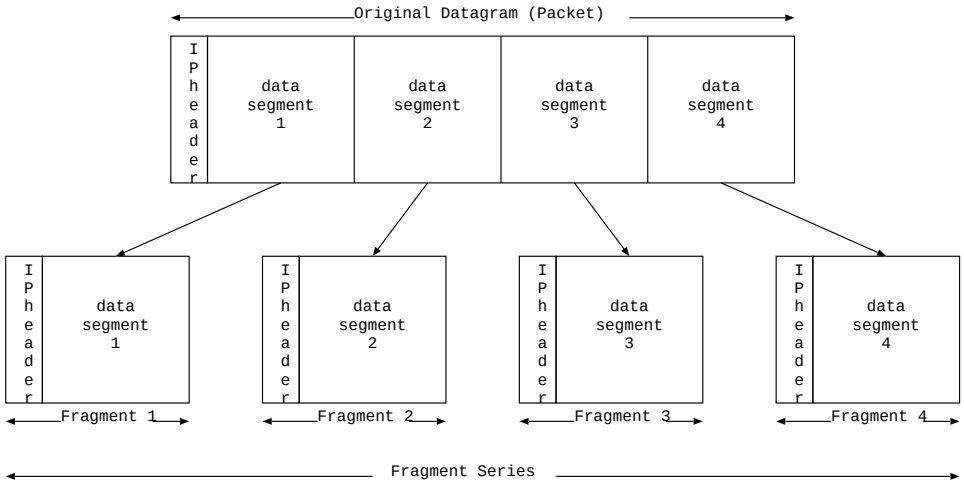

Fig. 1. Composition of a fragment series.

packets with different MTUs necessitates a system flexible enough to accommodate these variations. IP packet fragmentation increases the robustness and efficacy of IP as a universal protocol. In this paper, we examine the character and effects of fragmented IP traffic as actually occurring on highly aggregated Internet links.

The paper is organized as follows. Section II defines terminology for understanding fragmented traffic. Sources of data and methodologies for analysis are presented in Section III. In Section IV we present our results on characterization of fragmented traffic. Finally, Section V summarizes our findings.

II. Terminology

This section introduces the terminology used in our discussion of IP packet fragmentation. Several of these terms are illustrated in Figure 1.

As described in RFC 1191 [4], the *Path MTU* is the smallest MTU of all of the links on a path from a source host to a destination host. In the context of this paper, values observed for a Path MTU reflect the smallest MTU of all links between the source and the passive monitor.

We use the term *original datagram* to mean an IP datagram whose size exceeds the MTU of the next link on the path to its destination, and consequently, it will be fragmented. By *packet fragment*, or simply *fragment*, we mean a packet containing a portion of the payload of an original datagram. Note that while, for the purposes of this paper, the terms packet and datagram are synonymous, we will use *original datagram* and *packet fragment* in the interest of clarity. A *fragment series*, or simply *series*, is the ordered list (as seen on the network) of fragments containing the data that composed the original datagram.

The *size* of the series will be used to refer to the total number of bytes in the series, while the *length* of the series will mean the number of fragments in the series.

The *first fragment* is the packet containing the original IP header and the first segment of the payload of the original datagram. The *last fragment* is the packet containing the last portion of the payload of the original datagram. Because packets can be reordered as they pass through a network, the first observed and last observed fragments do not necessarily contain (respectively) the first and last pieces of the payload of the original datagram, and are thus not necessarily the first or last fragment of the series.

The first fragment is often, but not always, equal in size to the largest fragment in each series. Thus the *largest fragment size* is greater than or equal to the size of the other fragments in the series. Similarly, the last fragment is not always the smallest fragment in a series. So the *smallest fragment size* is less than or equal to the other fragment sizes in a series.

Because the IP protocol permits networks to drop, duplicate or reorder packets, the individual fragment packets for a single original datagram may not arrive at the destination in transmission order. We define a series as *complete* when there are sufficient fragment packets for reconstruction of the original datagram (i.e. reordering or duplication may have occurred, but no dropping). Conversely an *incomplete* series does not have sufficient information to reconstruct the original datagram; some part of the payload never reached our monitor.

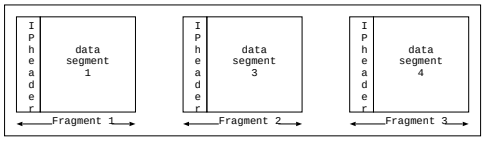

Fig. 2. Example incomplete series.

A series is *in-order* (Figure 3) if the fragments are observed arriving sequentially; we never monitor a fragment with an offset lower than its predecessors.

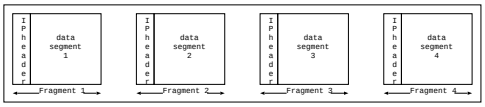

Fig. 3. Example in-order series.

Conversely, a series is considered in *reverse-order* (Figure 4 if its fragments are observed to have offsets that never increase. Thus a computer producing in-order series transmits data segment 1 through data segment N; and a computer producing reverse-order series transmits data segment N down through data segment 1. However, we cannot necessarily correlate the order in which we received the packet fragments and the order in which they were transmitted by the fragmenting router.

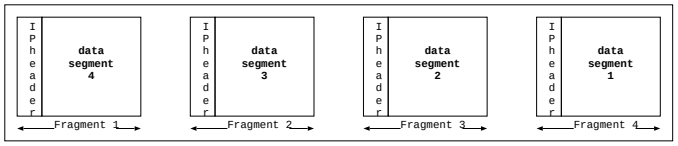

Fig. 4. Example reverse-order series.

A series contains a *duplicate* (Figure 5) if at least two of its fragments cover the exact same portion of the original payload.

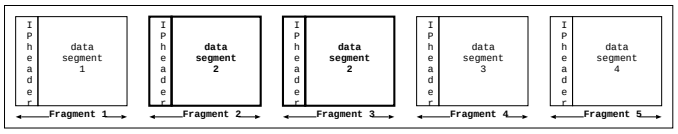

Fig. 5. Example duplicate series.

An *overlapping* series (Figure 6) has at least two fragment packets which contain overlapping portions of the original payload and the two fragments are not duplicates. Conversely, a *non-overlapping* series has no overlapping fragments. Note that the 'teardrop' denial of service attack sends large fragments which are overlapping except for a single byte, using up buffer resources in certain fragment reassembly implementations.

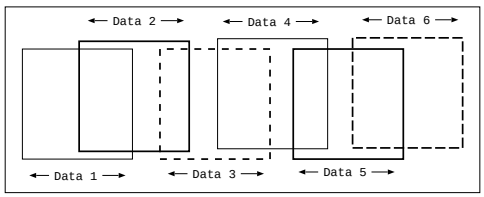

Fig. 6. Example overlapping series.

We define a *correct* series (Figure 7) as a series that is complete, with no overlapping or duplicated fragments. Any order of arrival of fragments is acceptable in a correct series.

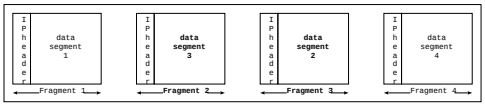

Fig. 7. Example correct series. Note that this series is *not in-order*.

III. Methodology

Measurement Sites

Traces for this study were collected from three different locations, summarized in Table I. One source of data for this study was a link at MAE-west. An Apptel Point card facilitated the collection of traffic exchanged by customers that peer at MAE-west. No intra-customer traffic is observed at this location. The second data source for this paper was traffic at SDNAP, a regional exchange point located in San Diego, California. We used libpcap to monitor this Gigabit Ethernet traffic. Using a FORE ATM OC3 card, we monitored the commodity access link that connects the University of California, San Diego campus (including such entities as the San Diego Supercomputer Center and the Scripps Institute of Oceanography) to CERFnet. Finally, traffic was collected from a link between Ames Internet Exchange (AIX) and MAE-west, using a WAND DAG [5] card.

The numbers of unique source hosts for each trace as shown in Table I were filtered to only count hosts which sent at least 3 packets over the lifetime of the trace. This filtering was applied to provide a more accurate count of the actual number of hosts transmitting across the link, since at least one trace at MAE-west contained periods where a random source denial of service attack was present.

Traffic Monitoring

Due to the large amount of traffic at some of the measurement sites, rather than taking complete header traces, a specialized tool, `crl_frag_capture`, collected the data for this study. `crl_frag_capture` examines only packet headers; we attempted no analysis of the payload portion of the traffic we monitored. The data collected is organized into hour long time intervals for post-processing. We collected four sets of data every hour:

frags.pcap — full header trace in libpcap [6] format of fragmented packets (either offset > 0 or 'more fragments' set).
src_ip.t2 — aggregated table of non-fragmented traffic with number of packets and bytes seen per source IP address.
proto_ports_folded.t2 — aggregated table of non-fragmented traffic with number of packets and bytes seen per 3-tuple of IP protocol, source port, and destination port. Since a significant amount of monitored traffic involves traffic to or from a well known port to an ephemeral port, additional aggregation was done for some commonly occurring ports. A list of 19 ports[2] was chosen from preliminary studies of non-fragmented traffic on these links. For each packet, if they match a source or destination of one of the chosen ports, the other port is set to 0, causing all traffic for the related services to fall into a single bucket. Additionally, all ports above 32767 were set to 32768, since the ports in this range are typically ephemeral and we found no well

[1]Unique IP source addresses which sent at least 3 packets over the trace lifetime.

[2]Specific ports in aggregation application order: 80, 53, 25, 443, 27015, 110, 113, 37, 20, 119, 5000, 6112, 6667, 6688, 6699, 6970, 8888, 9000, 27005.

Trace	Length		Characteristics		
Trace	Start Time (UTC)	Duration (hours)	Packets	Bytes	Src Hosts[1]
CERF-IN	Fri Mar 09 02:01	252.00	2797265857	1439570134374	2745493
CERF-OUT	Fri Mar 09 02:01	252.00	3394282672	1559169521158	37242
SDNAP	Fri Mar 09 01:36	259.58	1073320564	646676925673	328094
MAEWEST-1	Fri Mar 09 01:35	75.00	5307428883	2203613855199	1277423
MAEWEST-2	Tue Mar 13 02:12	132.00	8991448597	3963302269178	1691880
AIX-1	Fri Mar 09 01:38	58.00	8781881347	3281324259554	2684104
AIX-2	Mon Mar 12 04:35	49.00	8070585581	3743039881646	2478624

TABLE I
TRACES USED IN STUDY

known ports above 32767 with much traffic in the preliminary studies.

length.t2 — aggregated table of non-fragmented traffic with number of packets and bytes seen with given IP lengths.

Note that we did *not* collect full header traces for non-fragmented traffic, and that the partitioning of the data into separate tables for source IP address, protocol/ports, and packet length prevents recovering the original relationships between these fields.

`crl_frag_capture` relies on the CoralReef [7] software suite for header capture, interval handling and data aggregation.

Fragment Processing

To answer many of the unanswered questions about IP packet fragmentation, constituent fragments from an original datagram were assembled into a fragment series. Fragments were grouped into series using the identification, protocol, source IP address and destination IP address fields, since those fields uniquely define fragments of an original datagram. A timeout of 600 seconds was set on each series to provide sufficient time for all fragments to be seen, even if they were delayed by the network.

The payload of the original packet was not reconstructed, since to understand the properties of fragmented traffic, it was sufficient to look at the offset and size of each fragment.

Application Mapping

To better understand which applications or services produce the most fragmented traffic, we map the protocol, source port and destination port packet header fields to a named application using the well known ports methodology. An application is assigned by choosing the first matching rule from an ordered collection of protocol/port patterns. For this study, CAIDA's passive monitor report generator application list was used.[3] The list contained 81 entries, which includes common well known ports from the IANA port assignment list [8], as well as newer emerging multimedia and video game applications (such as RealAudio, Quake, Napster).

[3]The mapping code and list used in this study, as well as the current CAIDA list can be obtained from the authors or by emailing `coral-info@caida.org`.

IV. RESULTS

A. Overall trends of Fragmented Traffic

Table II shows the percentage of fragmented and non-fragmented traffic found in each trace. The total amount of fragmented traffic by bytes ranges from 0.09% (SDNAP) to 1.6% (MAE-west) of all traffic. By packets the amount of fragmented traffic ranges from 0.07% (SDNAP) to 0.7% (CERFnet). The percentage of unique source hosts[1] ranges from 0.04% to 0.2%. Note that some hosts which sent fragmented traffic also sent non-fragmented, so the host percentages may total to more than 100%.

The non-fragmented traffic measured by both the AIX and MAE-west monitors showed strong diurnal cycles. The traffic at SDNAP does not share the strongly cyclical nature of the other two locations, although it does show a daily decrease in traffic late at night (Pacific Standard Time). Figures 8 and 9 show time series plots of the non-fragmented traffic.

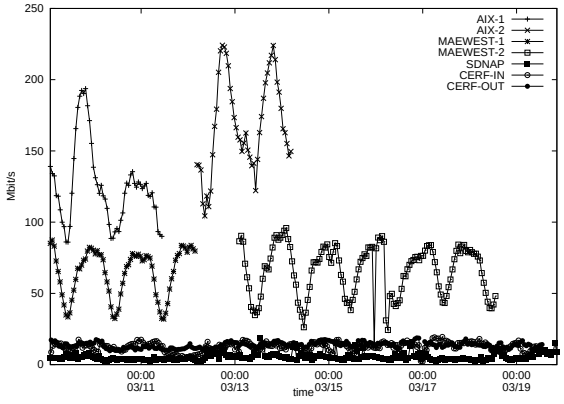

Fig. 8. Average hourly bandwidths for non-fragmented traffic.

The fragmented traffic also shows diurnal tendencies, but the data is noisier. Figures 10 and 11 show time series plots of the fragmented traffic.

Figures 12 and 13 show how the number of hosts sending non-fragmented and fragmented traffic varied over time. Note that these graphs not filtered for random source denial of service attacks.

Trace	Fragmented			Non-Fragmented		
	Pkts(%)	Bytes(%)	Hosts[1](%)	Pkts(%)	Bytes(%)	Hosts[1](%)
CERF-IN	0.675	1.556	0.042	99.325	98.444	99.989
CERF-OUT	0.742	1.283	0.177	99.258	98.717	100.000
SDNAP	0.069	0.090	0.023	99.931	99.910	99.998
MAEWEST-1	0.534	1.459	0.174	99.466	98.541	99.994
MAEWEST-2	0.578	1.573	0.183	99.422	98.427	99.996
AIX-1	0.269	0.835	0.172	99.731	99.165	99.973
AIX-2	0.250	0.590	0.162	99.750	99.410	99.974

TABLE II
PREVALENCE OF FRAGMENTED AND NON-FRAGMENTED IP TRAFFIC

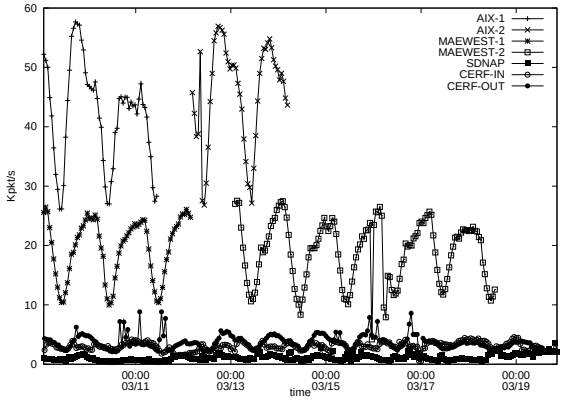

Fig. 9. Average hourly packet rates for non-fragmented traffic.

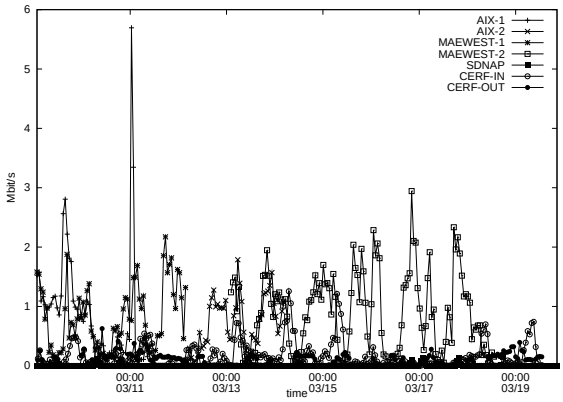

Fig. 10. Average hourly bandwidths for fragmented traffic.

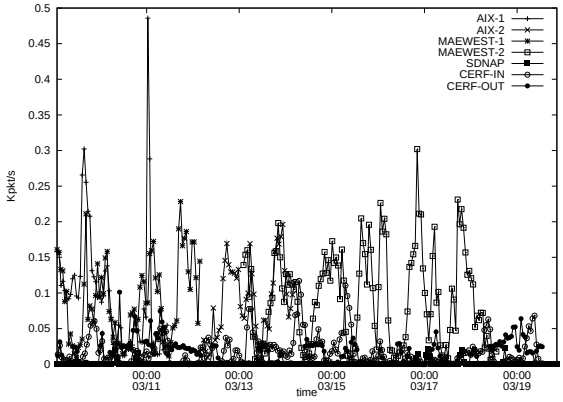

Fig. 11. Average hourly packet rates for fragmented traffic.

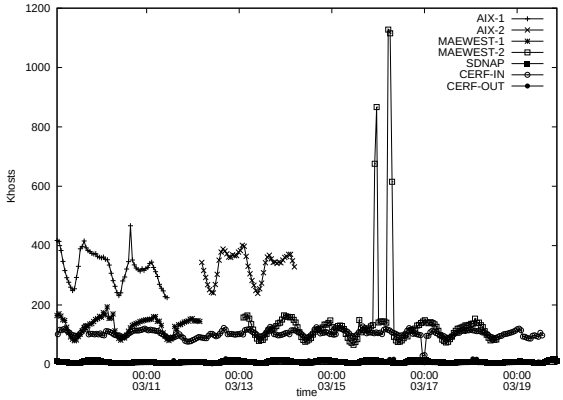

Fig. 12. Unique source hosts per hour for non-fragmented traffic.

B. Categorization of Fragmented Traffic

Fragment series can be categorized by the arrangement of their constituent fragment packets as received by our monitor. Table III shows the breakdown of all series based on the following attributes: correct, complete, in-order, reverse-order, overlapping, or duplicate, as defined in Section II. Of all series, 98.1% are complete, meaning they contain sufficient information to reconstruct the original datagram. Correct series (Figure 7) account for 89.6% of all series. Of complete series, 8.2% are in-order (Figure 3) and 81.4% are reverse-order (Figure 4); the remaining 10.4% are either overlapping (Figure 6) or duplicate (Figure 5) series; both are attributes which impede exact determination of ordering. Of all complete series, 1.1% are overlapping and 8.8% are duplicate.

The reverse-order series are not problematic; they are actually beneficial, since a host receiving a reverse-order series is able to allocate correct sized buffers immediately, using the fragment length and the offset fields, rather than growing or chaining buffers as subsequent fragments arrive. Note that Linux kernels send fragments in reverse order.

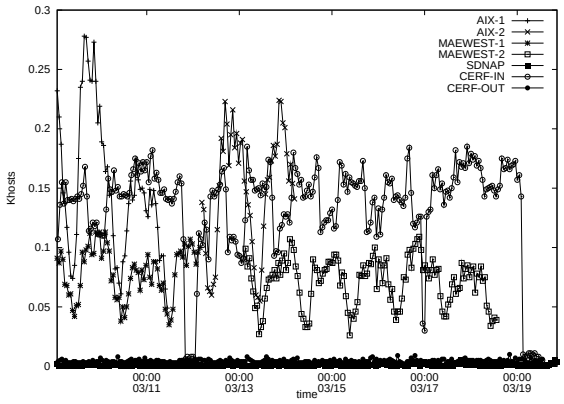

Fig. 13. Unique source hosts per hour for fragmented traffic.

C. Characteristics of Fragment Traffic

This section discusses some of the basic features of fragmented traffic. The data from the Ames Internet Exchange point is used because it is representative of fragmented traffic on all links studied. It also exemplifies anomalies typical of fragmented traffic. This trace contains 14,087 60828 byte fragment series and 39,090 series of 65888 bytes. Throughout the following sections, we will make note of the effects of these unusual occurrences. These fragment series have the following compositions:

The 60828 byte fragment series consist of 40 fragments of 1500 bytes follwed by 1 fragment of 828 bytes, with original datagram length of 60028 bytes. In this case 800 bytes of overhead (60828 - 60028) were caused by the 40 additional IP headers needed to transmit the series than would have been needed by a single packet.

The 65888 byte fragment series consist of 43 fragments of 1500 bytes follwed by 1 fragment of 1388 bytes, with original datagram length of 65028 bytes. In this case 860 bytes of overhead (65888 - 65028) were caused by the 43 additional IP headers needed to transmit the series than would have been needed by a single packet.

Fragments per Fragment Series (Figure 14):

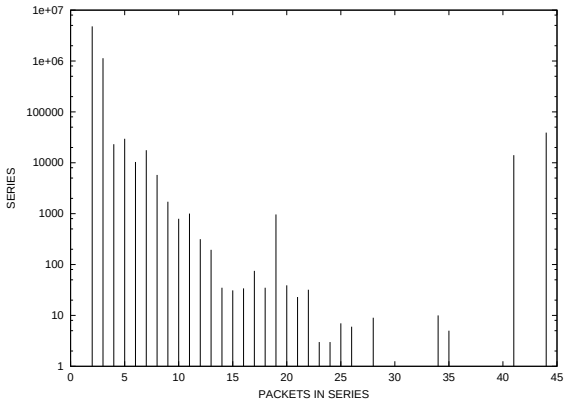

Fig. 14. Number of fragment packets for correct series for trace AIX-2.

Fragment series most commonly have lengths of two fragments. A high number of two-fragment series is expected, since it accounts for original datagrams that range from just exceeding the MTU of the next link to forty bytes less than double the MTU of the next link:

$$MTU < datagram \leq (2 * MTU) - 40$$

This spike in two-fragment series in Figure 14 is generally followed by decreasing numbers of packets with increasing length of the series. We often observe a pairing of even and odd lengths that results in a step-like deterioration in the frequency of occurrence of long fragment series. This behavior can be seen in the pairs (4,5), (6,7), (10,11), (14,15), (21,22), (23,24), and (25,26).

We observed an unusually large number of forty-one and forty-four fragment series at AIX because of the unusual frequency of packets of lengths 60028 and 65028 bytes, respectively. These packets were all broken up into 1500-byte fragments with one oddly sized leftover fragment.

Bytes per Fragment Series (Figure 15):

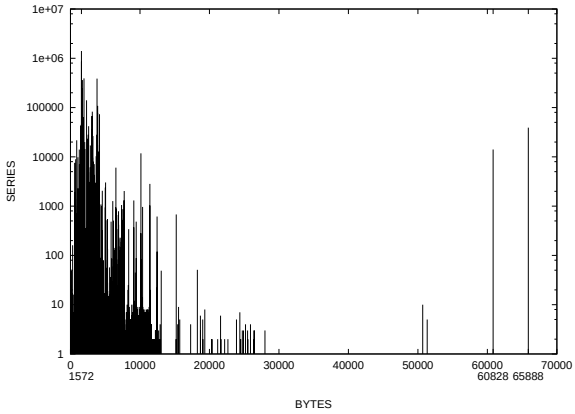

Fig. 15. Number of bytes transmitted for correct series for trace AIX-2. Note this includes the bytes in all of the IP headers for each fragment.

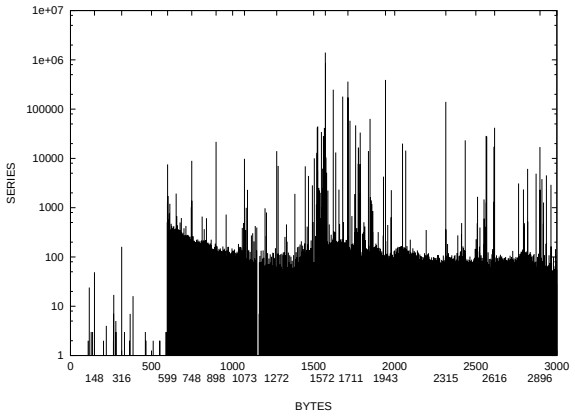

Fig. 16. Number of bytes transmitted for correct series for trace AIX-2. Note this includes the bytes in all of the IP headers for each fragment.

The size of the payload carried in each fragment series is highly variable. Similar to distributions of packet size in general, they have a random component. However, we

typically see a band around the range 1520-1636 of bytes per fragment series. Many of these sizes result from tunneled traffic. These original datagrams start out at 150 bytes – the MTU of Ethernet, and then have between 1 and 4 additional IP (or other) headers prepended. This banding effect, and the prevalence of original datagram sizes around 1500 bytes can be seen in Figure 16, an enlargement of the 0-3000 byte range of Figure 15. The most frequently occurring series size across all of the traces was 1572 bytes. We generally see a certain level of background "noise" that stretches across the graph. In this case, series up to around 10,000 bytes total occurred with a frequency of about one hundred series per size.

This graph shows evidence of fragmentation caused by MTU misconfiguration. We monitored a total of 92 series less than 256 bytes. Indeed, the smallest series, at 92 bytes, had only 52 bytes of payload. The overhead for this series, 40 bytes, is nearly twice the size of the payload! An additional 252 series are considered 'poorly configured' because they have series lengths less than 576 bytes. While in a few instances, for example, routers handling predominantly Voice over IP traffic, a low MTU is an optimal configuration, MTUs lower than 576 bytes are generally evidence of mistaken or misguided configuration. Some end hosts that use modem connections with SLIP set low MTUs for their dial-up link.

Largest Fragment Size Distribution (Figure 17):

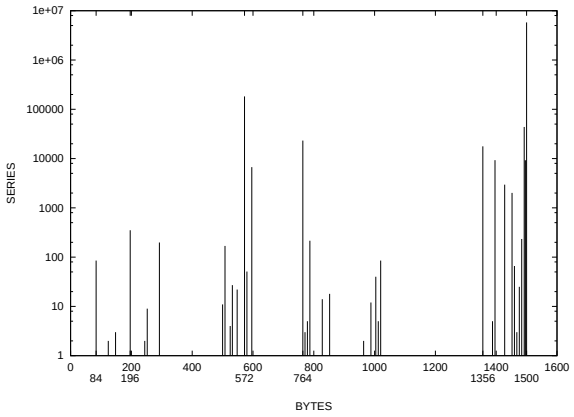

Fig. 17. Largest fragment size for correct series for trace AIX-2.

The size of the largest fragment found in a fragment series is indicative of the MTU of the link that necessitated fragmentation. Generally the first fragment in a fragment series is equal to this maximum size. However, this is not universally true. We identified in the AIX and MAE-west data a total of 237,263 two-fragment series in which the smallest fragment was sent first, with the largest trailing. While only 8.1% of the total correct fragment series were transmitted in reverse order, we cannot make the assumption that the first fragment of each series is also the largest.

The same misconfiguration that was apparent in the graph of the bytes per fragment series is visible here: it is unlikely that a packet would need to be fragmented to a size less than 576 bytes as it makes its way towards an exchange point. However, there are no observable artifacts of the 60028 or 65028 original datagram phenomena in this graph. All of those anomalies result in 1500 byte largest fragments, and since 1500 is by far the most common largest fragment size, the anomalies have no effect on the largest fragment size distribution.

Fragment Size (bytes)	Occurrence(%) # Series
1500	85.314
1484	11.112
572	1.186
1492	1.086
1496	0.455
1356	0.183
1396	0.120
124	0.115
764	0.097
1452	0.068

TABLE IV
TOP TEN LARGEST FRAGMENTS FROM CORRECT SERIES - ACROSS ALL TRACES

Many of the largest fragments occur at sizes easily predicted from the MTUs of common link types. Table IV shows the largest fragment size per series seen across all combined traces. For example, 1500 bytes is by far the most common largest fragment size; it is the maximum packet size for Ethernet networks. Ethernet networks using LLC/SNAP, in accordance with RFC 1042 [9] produce 1492 byte IP packets. DEC Gigaswitch traffic results in packets of length 1484 bytes. 572 bytes is a widely used PPP MTU and also results from usage of the default 576 byte transmission size. The largest size packet that a host is required to accept is 576 bytes by RFC 791 [10] and RFC 879 [11], therefore when Path MTU discovery fails or is not implemented, packets are sent at a size less than or equal to 576 bytes. Note for IPv6, the minimum MTU of any link must be 1280 bytes [12].

The default packet packet size of 576 bytes results in fragments of 572 bytes, because the length of the payload of each fragment packet except the last must be divisible by eight. This size requirement is based on the design of the IP packet header which specifies that the offset field holds the position of each fragment within the original datagram in eight-byte units.[10] The size of the entire fragment is the sum of the length of the IP header and the payload. Since IP options rarely occur, the IP header is 20 bytes. Therefore, the entire packet size for non-last fragments is $20 + N * 8$ for some N. The largest valid fragment packet size less than or equal to the default transmission size of 576 bytes is 572. Such a packet would consist of 20 bytes of IP header and 69 eight-byte units of fragment payload.

Many largest fragment sizes demonstrate configuration errors. This evidences the utility of Path MTU Discovery, since there is no "safe" transmission size at which a host can send packets to prevent fragmentation without an un-

acceptably high increase in per-packet overhead. Yet since there is no guarantee that two packets sent back-to-back from a source will take the same path to their destination, the same result validates the existence of the packet fragmentation mechanism of IP, since once a packet diverges from the path traveled in the MTU discovery phase, the packet could encounter links with a wide range of MTUs.

Smallest Fragment Size Distribution (Figure 18):

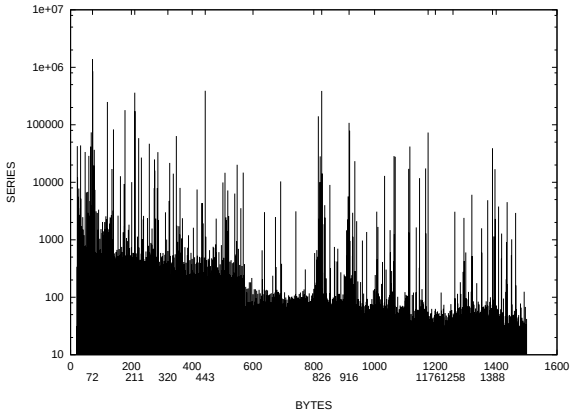

Fig. 18. Smallest fragment size for correct series for trace AIX-2.

The comparatively small variation in the MTUs of common links, in the face of the wide variation of original datagram sizes, produces a background frequency of approximately one hundred series across a wide variety of smallest fragment sizes. This is the direct result of the inherent randomness of the sizes of the original datagrams. This background level decreases across the range of packet sizes because the frequency of occurrence of packet sizes decreases with a rate of $(sizeofpacket)^2/2$. Since the first and middle fragments of most series are the uniform size of the MTU of the subsequent link, the random size of the original fragment is seen only in the sizes of the smallest fragment. The most commonly occurring smallest fragment size is 72, which corresponds to the most common fragment series sizes of 1572 bytes. After a 1500 byte largest fragment has been composed, 72 bytes is the leftover value. Each spike in the graph corresponds to a frequently occurring combination of original datagram length and MTU of the fragmenting router.

Our 60028 and 65028 byte fragment series anomalies are not visible in this graph. The 60028 byte original datagrams result in smallest fragments that are 320 bytes in size. Likewise, 1258 bytes corresponds to the smallest fragments from the 65028 byte original datagrams. However, since there are only 41 and 45 examples of each unusual series, these occurrences are masked by the general background rate.

The Effects of Fragments Larger than 1500 Bytes

As we have seen in the previous graphs, the most frequently occurring original datagram sizes shape the characteristics of their resulting fragments. Fragment traffic at MAE-west is unusual in that a common largest fragment size for this link is 4348 bytes, rather than the usual sizes of less than 1500 bytes.

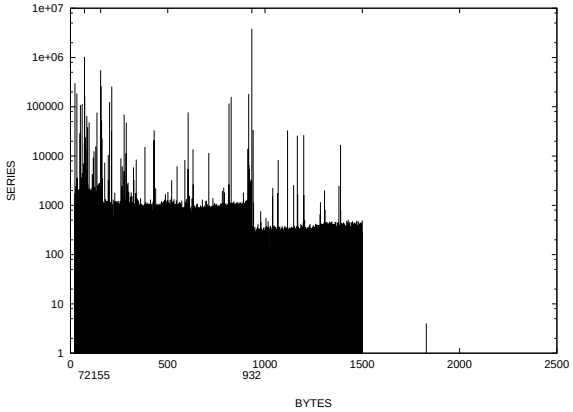

Fig. 19. Smallest fragment size for correct series for trace MAEWEST-1.

Smallest fragments larger than 1500 bytes accompany the largest fragments that are greater than the usual size, as shown in Figure 19. They occur less frequently, however, due to their "leftover" nature. Large packets resulting from a higher MTU do not necessarily have last fragments of increased size.

D. Fragmented Traffic Protocols and Applications

This section examines which services, protocols and applications contribute to fragmented traffic. Values presented are for all study traces combined.

Services Causing Fragmentation

Protocol		Occurrence(%)
Name	Number	# Series
UDP	17	74.981
IPENCAP	4	11.645
ESP (IPSEC)	50	4.881
ICMP	1	2.546
TCP	6	2.261
GRE	47	1.906
IPIP	94	1.084
AH (IPSEC)	51	0.664
IGMP	2	0.030
AX.25	93	0.001

TABLE V
TOP PROTOCOLS FROM CORRECT SERIES - ACROSS ALL TRACES

While rumors abound that NFS causes all of the fragmented traffic on LANs and backbone networks, in reality, tunneled traffic is the most prevalent. For example, on the link between the UCSD campus and CERFnet, the largest cause of fragmented traffic, by several orders of magnitude, was IPIP tunneled traffic. The fragmented traffic consists of IP packets sized at the MTU of their local network (generally 1500 bytes) which were then tunneled, causing the addition of an additional 20 byte IP header. The resulting

Category						Occurrence(%)
Correct	Complete	In-Order	Reverse	Overlap	Duplicate	# Series
YES	YES	YES	-	-	-	79.922
YES	YES	-	YES	-	-	8.051
-	YES	-	-	-	YES	7.493
YES	YES	-	-	-	-	1.620
-	YES	-	-	YES	YES	1.093
-	-	YES	YES	-	-	1.016
-	-	YES	-	-	-	0.595
-	-	-	YES	-	-	0.111
-	-	YES	YES	-	YES	0.044
-	-	-	-	-	-	0.030

TABLE III
TOP SERIES KINDS FROM ALL SERIES ACROSS ALL TRACES

Protocol		Fragmented		Non-Fragmented	
Name	Number	Pkts(%)	Bytes(%)	Pkts(%)	Bytes(%)
UDP	17	0.300	0.800	12.197	3.713
IPENCAP	4	0.061	0.108	0.123	0.039
ESP (IPSEC)	50	0.014	0.025	0.278	0.277
ICMP	1	0.044	0.139	1.860	0.447
TCP	6	0.008	0.018	84.815	94.213
GRE	47	0.005	0.009	0.176	0.130
IPIP	94	0.004	0.007	0.033	0.021
AH (IPSEC)	51	0.002	0.003	0.053	0.042
IGMP	2	0.001	0.002	0.000	0.000
AX.25	93	0.000	0.000	0.004	0.001

TABLE VI
PROTOCOL BREAKDOWN FOR FRAGMENTED AND NON-FRAGMENTED IP TRAFFIC. PERCENTAGES ARE OF TOTAL TRAFFIC.

Protocol		Fragmented		Non-Fragmented		Ratio (Frag/Non-Frag)	
Name	Number	Pkts(%)	Bytes(%)	Pkts(%)	Bytes(%)	Pkts	Bytes
UDP	17	68.256	72.033	12.251	3.754	5.571	19.187
IPENCAP	4	13.857	9.687	0.123	0.040	112.329	243.500
ESP (IPSEC)	50	3.234	2.273	0.280	0.280	11.567	8.107
ICMP	1	10.049	12.494	1.868	0.452	5.379	27.655
TCP	6	1.734	1.588	85.189	95.271	0.020	0.017
GRE	47	1.162	0.793	0.177	0.132	6.571	6.025
IPIP	94	0.972	0.669	0.033	0.021	29.577	31.355
AH (IPSEC)	51	0.399	0.285	0.053	0.043	7.560	6.710
IGMP	2	0.334	0.177	0.000	0.000	812.948	5294.136
AX.25	93	0.001	0.000	0.005	0.001	0.198	0.083

TABLE VII
DIFFERENCES IN PROTOCOL BREAKDOWN FOR FRAGMENTED AND NON-FRAGMENTED IP TRAFFIC. PERCENTAGES ARE RELATIVE TO EACH CATEGORY. RATIOS GREATER THAN ONE REFLECT A HIGHER PERCENTAGE OF FRAGMENTED TRAFFIC FOR THAT PROTOCOL THAN OF NON-FRAGMENTED TRAFFIC; SIMILARLY RATIOS LESS THAN ONE REFLECT A GREATER PROPORTION OF NON-FRAGMENTED COMPARED TO FRAGMENTED.

Protocol		Fragmented		Non-Fragmented	
Name	Number	Pkts(%)	Bytes(%)	Pkts(%)	Bytes(%)
UDP	17	2.403	17.725	97.597	82.275
IPENCAP	4	33.169	73.220	66.831	26.780
ESP (IPSEC)	50	4.862	8.343	95.138	91.657
ICMP	1	2.322	23.694	97.678	76.306
TCP	6	0.009	0.019	99.991	99.981
GRE	47	2.821	6.336	97.179	93.664
IPIP	94	11.558	26.039	88.442	73.961
AH (IPSEC)	51	3.232	7.006	96.768	92.994
IGMP	2	78.223	98.346	21.777	1.654
AX.25	93	0.087	0.093	99.913	99.907

TABLE VIII
DIFFERENCES IN PROTOCOL BREAKDOWN FOR FRAGMENTED AND NON-FRAGMENTED IP TRAFFIC. PERCENTAGES ARE RELATIVE TO EACH PROTOCOL.

Application	Occurrence(%)
	# Series
Unclassified UDP	73.865
ICMPECHOREQUEST	1.560
SMTP	1.217
ICMPECHOREPLY	0.940
FTP_DATA	0.730
L2TP	0.309
REALAUDIO_UDP	0.245
SQUID_ICP	0.125
CUSEEME	0.118
WWW	0.115
NAPSTER_DATA	0.113
NFS	0.111
Unclassified TCP	0.068
QUAKE	0.058
ICMP 13/0	0.045
MS_MEDIA	0.036
HALFLIFE	0.036
DISCARD	0.032
NETBIOS	0.018
DAYTIME	0.015
GNUTELLA	0.014

TABLE IX
TOP APPLICATIONS FROM CORRECT SERIES - ACROSS ALL TRACES

ICMP Application	Occurrence(%)
	# Series
ICMPECHOREQUEST	1.560
ICMPECHOREPLY	0.940
ICMP 13/0	0.045
ICMPNOPORT	0.001
ICMP 11/1	0.000
ICMPNOHOST	0.000
ICMP 3/4	0.000
ICMP 69/0	0.000
ICMPTTL	0.000

TABLE X
TOP ICMP APPLICATIONS FROM CORRECT SERIES - ACROSS ALL TRACES

1520 byte datagram exceeds the MTU of the subsequent link, and is fragmented into either a 1500 byte first fragment and a 40 byte second fragment or a 1484 byte first fragment and a 56 byte second fragment. This fragmentation is entirely preventable – a machine that is known to send traffic through an IPIP tunnel could set the MTU of the interface through which it sends traffic to be 1480 bytes, rather than 1500. This would reduce the network load resulting from the tunneled traffic by 98.7% – the machine would generate an extra packet for only every seventy-fifth packet sent, rather than requiring a second packet for every original datagram sent from the machine.

Tunneled traffic is not a local phenomenon. The combination of IPENCAP, IPIP, GRE, and UDP-L2TP accounts for 15% of all fragmented traffic – by far the largest single cause of fragmentation. In contrast, NFS accounts for only 0.1% of fragmented traffic.

The most frequently fragmented protocol is IGMP – some 78% of IGMP packets are fragments. However, since IGMP accounts for only 0.001% of all traffic, this fact is of purely academic import.

The most prevalent protocol of fragmented traffic is UDP, which accounts for 68.3% of fragmented traffic is UDP, followed by IPENCAP at 13.9%, ICMP at 10.0%, and ESP at 3.2%. Fragmented ICMP traffic consists primarily (98.1%) of echo requests and replies, although a small but significant number of timestamp requests were also monitored. Path MTU Discovery successfully limits the amount of TCP traffic that is fragmented; however, its effects are not quite as ubiquitous as some might claim. More than three million packets over the course of a week, 0.009% of the total TCP traffic, consisted of fragmented packets. Fragmented TCP traffic does indeed exist

TCP Application	Occurrence(%)
	# Series
SMTP	1.217
FTP_DATA	0.730
WWW	0.115
NAPSTER_DATA	0.113
Unclassified TCP	0.068
GNUTELLA	0.014
X11	0.002
BGP	0.000
SSH	0.000
KERBEROS	0.000

TABLE XI
TOP TCP APPLICATIONS FROM CORRECT SERIES - ACROSS ALL TRACES

UDP Application	Occurrence(%)
	# Series
Unclassified UDP	73.865
L2TP	0.309
REALAUDIO_UDP	0.245
SQUID_ICP	0.125
CUSEEME	0.118
NFS	0.111
QUAKE	0.058
MS_MEDIA	0.036
HALFLIFE	0.036
DISCARD	0.032

TABLE XII
TOP UDP APPLICATIONS FROM CORRECT SERIES - ACROSS ALL TRACES

on highly aggregated links.

TCP applications

Half of all fragmented TCP traffic, 53.9%, is composed of SMTP packets. FTP data and WWW follow, with 32.3% and 5.1%, respectively. Napster accounts for 5.0% of all fragmented TCP traffic, and Gnutella produces 0.6%, for a total of 5.6% of fragmented TCP traffic from these two peer-to-peer file-sharing applications.

UDP applications

L2TP accounted for 28.9% of the fragmented UDP traffic with identifiable applications. RealAudio followed close behind with 22.9%. Microsoft's Windows Media Player weighed in with 3.6%, for a total of 26.2% streaming media. The Squid caching protocol (SQUID_ICP) composes 11.7% of the identifiable UDP applications, barely edging out video-conferencing software CUSEEME at 11.0%. 10.4% of identifiable UDP application traffic composed NFS packets. Finally, Quake accounted for 5.4% of identifiable UDP traffic. Halflife followed with 3.6%, for a total of 8.8% of identifiable UDP application traffic from video games. Unfortunately, we were unable to classify the majority (73.9%) of UDP traffic. As many possible sources of this traffic, including multicast, have been ruled out, we conjecture that dynamic H.323 video-conferencing applications account for a significant portion of the unknown UDP applications.

V. CONCLUSION

Many assertions about the nature and extent of fragmented traffic are based in folklore, rather than measurement and analysis. Common folklore includes: fragmented traffic is decreasing or nonexistent, fragmented traffic exists only on LANs (due to NFS) not on backbone links, misconfiguration causes most fragmentation, only UDP traffic is fragmented,

Fragmented traffic does regularly occur at highly aggregated exchange points as well as on access links.

While the majority of fragmented traffic is UDP (68% by packets and 72% by bytes), ICMP, IPSEC, TCP and tunneled traffic are all present. Tunneled traffic forms a large portion of fragmented traffic (at least 16% of packets and 11% of bytes).

NFS only accounts for 0.1% of fragment series seen. Most UDP traffic was not classifiable, because of the use of ephemeral ports and dynamically exchanged ports. The classifiable UDP traffic was comprised primarily of tunneling, streaming media and game traffic.

VI. ACKNOWLEDGMENTS

We would like to thank Ryan Koga for writing the program used to collect data for this study, and Ken Keys for suggestions and help using CoralReef. This paper would not have been possible without the assistance of Steve Feldman and Bobby Cates for help with the MAE-west and AIX passive monitors. As always, the support of the CAIDA staff was invaluable.

REFERENCES

[1] C. A. Kent and J. C. Mogul, "Fragmentation considered harmful," *WRL Technical Report 87/3*, Dec. 1987.

[2] G. P. Chandranmenon and G. Varghese, "Reconsidering fragmentation and reassembly," in *PODC: 17th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 1998.

[3] Sean McCreary and k claffy, "Trends in wide area IP traffic patterns: A view from Ames Internet Exchange," in *ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, Sept. 2000.

[4] J. C. Mogul and S. E. Deering, "RFC 1191: Path MTU discovery," Nov. 1990.

[5] Waikato Applied Network Dynamics group, "The DAG project," `http://dag.cs.waikato.ac.nz/`.

[6] S. McCanne, C. Leres, and V. Jacobson, *libpcap*, Lawrence Berkeley Laboratory, Berkeley, CA, available via anonymous ftp to ftp.ee.lbl.gov.

[7] Ken Keys, David Moore, Ryan Koga, Edouard Lagache, Michael Tesch, and k claffy, "The architecture of CoralReef: an Internet traffic monitoring software suite," in *PAM2001 — A workshop on Passive and Active Measurements*. CAIDA, Apr. 2001, RIPE NCC, http://www.caida.org/tools/measurement/coralreef/.

[8] "IANA port assignments," `ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers`.

[9] J. Postel and J. K. Reynolds, "RFC 1042: Standard for the transmission of IP datagrams over IEEE 802 networks," Feb. 1988.

[10] J. Postel, "RFC 791: Internet Protocol," Sept. 1981.

[11] J. Postel, "RFC 879: The TCP Maximum Segment Size and Related Topics," Nov. 1983.
[12] S. Deering and R. Hinden, "RFC 2460: Internet Protocol, Version 6 (IPv6) specification," Dec. 1998.