A First Look at Modern Enterprise Traffic*

Ruoming Pang[†], Mark Allman[‡], Mike Bennett[¶], Jason Lee[¶], Vern Paxson[‡,¶], Brian Tierney[¶]
[†]*Princeton University,* [‡]*International Computer Science Institute,*
[¶]*Lawrence Berkeley National Laboratory (LBNL)*

Abstract

While wide-area Internet traffic has been heavily studied for many years, the characteristics of traffic *inside* Internet enterprises remain almost wholly unexplored. Nearly all of the studies of enterprise traffic available in the literature are well over a decade old and focus on individual LANs rather than whole sites. In this paper we present a broad overview of internal enterprise traffic recorded at a medium-sized site. The packet traces span more than 100 hours, over which activity from a total of several thousand internal hosts appears. This wealth of data—which we are publicly releasing in anonymized form—spans a wide range of dimensions. While we cannot form general conclusions using data from a single site, and clearly this sort of data merits additional in-depth study in a number of ways, in this work we endeavor to characterize a number of the most salient aspects of the traffic. Our goal is to provide a first sense of ways in which modern enterprise traffic is similar to wide-area Internet traffic, and ways in which it is quite different.

1 Introduction

When Cáceres captured the first published measurements of a site's wide-area Internet traffic in July, 1989 [4, 5], the entire Internet consisted of about 130,000 hosts [13]. Today, the largest enterprises can have more than that many hosts just by themselves.

It is striking, therefore, to realize that more than 15 years after studies of wide-area Internet traffic began to flourish, the nature of traffic *inside* Internet enterprises remains almost wholly unexplored. The characterizations of enterprise traffic available in the literature are either vintage LAN-oriented studies [11, 9], or, more recently, focused on specific questions such as inferring the roles played by different enterprise hosts [23] or communities of interest within a site [2]. The only broadly flavored look at traffic within modern enterprises of which we are aware is the study of OSPF routing behavior in [21]. Our aim is to complement that study with a look at the make-up of traffic as seen at the packet level within a contemporary enterprise network.

One likely reason why enterprise traffic has gone unstudied for so long is that it is technically difficult to measure. Unlike a site's Internet traffic, which we can generally record by monitoring a single access link, an enterprise of significant size lacks a single choke-point for its internal traffic. For the traffic we study in this paper, we primarily recorded it by monitoring (one at a time) the enterprise's two central routers; but our measurement apparatus could only capture two of the 20+ router ports at any one time, so we could not attain any sort of comprehensive snapshot of the enterprise's activity. Rather, we piece together a partial view of the activity by recording a succession of the enterprise's subnets in turn. This piecemeal tracing methodology affects some of our assessments. For instance, if we happen to trace a portion of the network that includes a large mail server, the fraction of mail traffic will be measured as larger than if we monitored a subnet without a mail server, or if we had an ideally comprehensive view of the enterprise's traffic. Throughout the paper we endeavor to identify such biases as they are observed. While our methodology is definitely imperfect, to collect traces from a site like ours in a comprehensive fashion would require a large infusion of additional tracing resources.

Our study is limited in another fundamental way, namely that all of our data comes from a single site, and across only a few months in time. It has long been established that the wide-area Internet traffic seen at different sites varies a great deal from one site to another [6, 16] and also over time [16, 17], such that studying a single site *cannot* be representative. Put another way, for wide-area Internet traffic, the very notion of "typical" traffic is not well-defined. We would expect the same to hold for enterprise traffic (though this basic fact actually remains to be demonstrated), and

*This paper appears in the Internet Measurement Conference, 2005.

therefore our single-site study can at best provide an *example* of what modern enterprise traffic looks like, rather than a general representation. For instance, while other studies have shown peer-to-peer file sharing applications to be in widespread use [20], we observe nearly none of it in our traces (which is likely a result of organizational policy).

Even given these significant limitations, however, there is much to explore in our packet traces, which span more than 100 hours and in total include activity from 8,000 internal addresses at the Lawrence Berkeley National Laboratory and 47,000 external addresses. Indeed, we found the very wide range of dimensions in which we might examine the data difficult to grapple with. Do we characterize individual applications? Transport protocol dynamics? Evidence for self-similarity? Connection locality? Variations over time? Pathological behavior? Application efficiency? Changes since previous studies? Internal versus external traffic? Etc.

Given the many questions to explore, we decided in this first look to pursue a broad overview of the characteristics of the traffic, rather than a specific question, with an aim towards informing future, more tightly scoped efforts. To this end, we settled upon the following high-level goals:

- To understand the makeup (working up the protocol stack from the network layer to the application layer) of traffic on a modern enterprise network.
- To gain a sense of the patterns of locality of enterprise traffic.
- To characterize application traffic in terms of how intranet traffic characteristics can differ from Internet traffic characteristics.
- To characterize applications that might be heavily used in an enterprise network but only rarely used outside the enterprise, and thus have been largely ignored by modeling studies to date.
- To gain an understanding of the load being imposed on modern enterprise networks.

Our general strategy in pursuing these goals is "understand the big things first." That is, for each of the dimensions listed above, we pick the most salient contributors to that dimension and delve into them enough to understand their next degree of structure, and then repeat the process, perhaps delving further if the given contributor remains dominant even when broken down into components, or perhaps turning to a different high-level contributor at this point. The process is necessarily somewhat opportunistic rather than systematic, as a systematic study of the data would consume far more effort to examine, and text to discuss, than is feasible at this point.

	D_0	D_1	D_2	D_3	D_4
Date	10/4/04	12/15/04	12/16/04	1/6/05	1/7/05
Duration	10 min	1 hr	1 hr	1 hr	1 hr
Per Tap	1	2	1	1	1-2
# Subnets	22	22	22	18	18
# Packets	17.8M	64.7M	28.1M	21.6M	27.7M
Snaplen	1500	68	68	1500	1500
Mon. Hosts	2,531	2,102	2,088	1,561	1,558
LBNL Hosts	4,767	5,761	5,210	5,234	5,698
Remote Hosts	4,342	10,478	7,138	16,404	23,267

Table 1: Dataset characteristics.

The general structure of the paper is as follows. We begin in § 2 with an overview of the packet traces we gathered for our study. Next, § 3 gives a broad breakdown of the main components of the traffic, while § 4 looks at the locality of traffic sources and destinations. In § 5 we examine characteristics of the applications that dominate the traffic. § 6 provides an assessment of the load carried by the monitored networks. § 7 offers final thoughts. We note that given the breadth of the topics covered in this paper, we have spread discussions of related work throughout the paper, rather than concentrating these in their own section.

2 Datasets

We obtained multiple packet traces from two internal network locations at the Lawrence Berkeley National Laboratory (LBNL) in the USA. The tracing machine, a 2.2 GHz PC running FreeBSD 4.10, had four NICs. Each captured a unidirectional traffic stream extracted, via network-controllable Shomiti taps, from one of the LBNL network's central routers. While the kernel did not report any packet-capture drops, our analysis found occasional instances where a TCP receiver acknowledged data not present in the trace, suggesting the reports are incomplete. It is difficult to quantify the significance of these anomalies.

We merged these streams based on timestamps synchronized across the NICs using a custom modification to the NIC driver. Therefore, with the four available NICs we could capture traffic for two LBNL subnets. A further limitation is that our vantage point enabled the monitoring of traffic to and from the subnet, but not traffic that remained within the subnet. We used an *expect* script to periodically change the monitored subnets, working through the 18–22 different subnets attached to each of the two routers.

Table 1 provides an overview of the collected packet traces. The "per tap" field indicates the number of traces taken on each monitored router port, and *Snaplen* gives the maximum number of bytes captured for each packet. For example, D_0 consists of full-packet traces from each of the 22 subnets monitored once for 10 minutes at a time, while D_1 consists of 1 hour header-only (68 bytes) traces from the 22 subnets, each monitored twice (i.e., two 1-hour traces per subnet).

	D_0	D_1	D_2	D_3	D_4
IP	99%	97%	96%	98%	96%
!IP	1%	3%	4%	2%	4%
ARP	10%	6%	5%	27%	16%
IPX	80%	77%	65%	57%	32%
Other	10%	17%	29%	16%	52%

Table 2: Fraction of packets observed using the given network layer protocol.

3 Broad Traffic Breakdown

We first take a broad look at the protocols present in our traces, examining the network, transport and application layers.

Table 2 shows the distribution of "network layer" protocols, i.e., those above the Ethernet link layer. IP dominates, constituting more than 95% of the packets in each dataset, with the two largest non-IP protocols being IPX and ARP; the distribution of non-IP traffic varies considerably across the datasets, reflecting their different subnet (and perhaps time-of-day) makeup.[1]

Before proceeding further, we need to deal with a somewhat complicated issue. The enterprise traces include *scanning traffic* from a number of sources. The most significant of these sources are legitimate, reflecting proactive vulnerability scanning conducted by the site. Including traffic from scanners in our analysis would skew the proportion of connections due to different protocols. And, in fact, scanners can engage services that otherwise remain idle, skewing not only the magnitude of the traffic ascribed to some protocol but also the number of protocols encountered.

	D_0	D_1	D_2	D_3	D_4
Bytes (GB)	13.12	31.88	13.20	8.98	11.75
TCP	66%	95%	90%	77%	82%
UDP	34%	5%	10%	23%	18%
ICMP	0%	0%	0%	0%	0%
Conns (M)	0.16	1.17	0.54	0.75	1.15
TCP	26%	19%	23%	10%	8%
UDP	68%	74%	70%	85%	87%
ICMP	6%	6%	8%	5%	5%

Table 3: Fraction of connections and bytes utilizing various transport protocols.

In addition to the known internal scanners, we identify additional scanning traffic using the following heuristic. We first identify sources contacting more than 50 distinct hosts. We then determine whether at least 45 of the distinct addresses probed were in ascending or descending order. The scanners we find with this heuristic are primarily external sources using ICMP probes, because most other external scans get blocked by scan filtering at the LBNL border. Prior to our subsequent analysis, we remove traffic from sources identified as scanners along with the 2 internal scanners. The fraction of connections removed from the traces ranges from 4–18% across the datasets. A more in-depth study of characteristics that the scanning traffic exposes is a fruitful area for future work.

We now turn to Table 3, which breaks down the traffic by transport protocol (i.e., above the IP layer) in terms of payload bytes and packets for the three most popular transports found in our traces. The "Bytes" and "Conns" rows give the total number of payload bytes and connections for each dataset in Gbytes and millions, respectively. The ICMP traffic remains fairly consistent across all datasets, in terms of fraction of both bytes and connections. The mix of TCP and UDP traffic varies a bit more. We note that the bulk of the bytes are sent using TCP, and the bulk of the connections use UDP, for reasons explored below. Finally, we observe a number of additional transport protocols in our datasets, each of which make up only a slim portion of the traffic, including IGMP, IPSEC/ESP, PIM, GRE, and IP protocol 224 (unidentified).

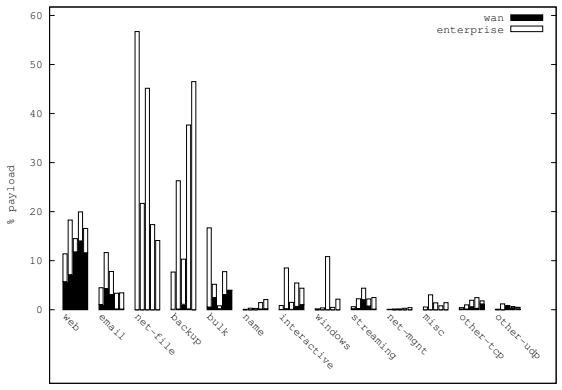
Category	Protocols
backup	Dantz, Veritas, "connected-backup"
bulk	FTP, HPSS
email	SMTP, IMAP4, IMAP/S, POP3, POP/S, LDAP
interactive	SSH, telnet, rlogin, X11
name	DNS, Netbios-NS, SrvLoc
net-file	NFS, NCP
net-mgnt	DHCP, ident, NTP, SNMP, NAV-ping, SAP, NetInfo-local
streaming	RTSP, IPVideo, RealStream
web	HTTP, HTTPS
windows	CIFS/SMB, DCE/RPC, Netbios-SSN, Netbios-DGM
misc	Steltor, MetaSys, LPD, IPP, Oracle-SQL, MS-SQL

Table 4: Application categories and their constituent protocols.

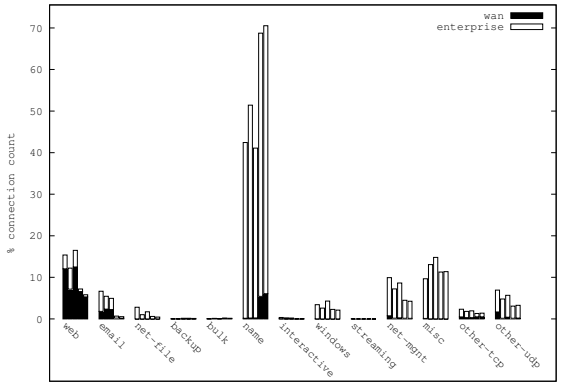
Next we break down the traffic by application category. We group TCP and UDP application protocols as shown in Table 4. The table groups the applications together based on their high-level purpose. We show only those distinguished by the amount of traffic they transmit, in terms of packets, bytes or connections (we omit *many* minor additional categories and protocols). In § 5 we examine the characteristics of a number of these application protocols.

Figure 1 shows the fraction of unicast payload bytes and connections from each application category (multicast traffic is discussed below). The five bars for each category correspond to our five datasets. The total height of the bar represents the percentage of traffic due to the given category. The solid part of the bar represents the fraction of the total in which one of the endpoints of the connection resides outside of LBNL, while the hollow portion of the bar represents the fraction of the total that remains within LBNL's network. (We delve into traffic origin and locality in more depth in § 4.) We also examined the traffic

[1] Hour-long traces we made of ≈ 100 individual hosts (not otherwise analyzed here) have a makeup of 35–67% *non*-IPv4 packets, dominated by *broadcast* IPX and ARP. This traffic is mainly confined to the host's subnet and hence not seen in our inter-subnet traces. However, the traces are too low in volume for meaningful generalization.



(a) Bytes



(b) Connections

Figure 1: Fraction of traffic using various application layer protocols.

breakdown in terms of packets, but since it is similar to the breakdown in terms of bytes, we do not include the plot due to space constraints. We note, however, that when measuring in terms of packets the percentage of interactive traffic is roughly a factor of two more than when assessing the traffic in terms of bytes, indicating that interactive traffic consists, not surprisingly, of small packets.

The plots show a *wider range of application usage* within the enterprise than over the WAN. In particular, we observed 3–4 times as many application categories on the internal network as we did traversing the border to the WAN. The wider range likely reflects the impact of administrative boundaries such as trust domains and firewall rules, and if so should prove to hold for enterprises in general. The figure also shows that the majority of traffic observed is local to the enterprise. This follows the familiar pattern of locality in computer and network systems which, for example, plays a part in memory, disk block, and web page caching.

In addition, Figure 1 shows the reason for the finding above that most of the connections in the traces use UDP, while most of the bytes are sent across TCP connections. Many connections are for "name" traffic across all the datasets (45–65% of the connections). However, the byte count for "name" traffic constitutes no more than 1% of the aggregate traffic. The "net-mgnt", "misc" and "other-udp" categories show similar patterns. While most of the connections are short transaction-style transfers, most of the bytes that traverse the network are from a relatively few connections. Figure 1(a) shows that the "bulk", "network-file" and "backup" categories constitute a majority of the bytes observed across datasets. In some of the datasets, "windows", "streaming" and "interactive" traffic each contribute 5–10% of the bytes observed, as well. The first two make sense because they include bulk-transfer as a component of their traffic; and in fact interactive traffic does too, in the form of SSH, which can be used not only as an interactive login facility but also for copying files and tunneling other applications.

Most of the application categories shown in Figure 1 are *unbalanced* in that the traffic is dominated by either the connection count or the byte count. The "web" and "email" traffic categories are the exception; they show non-negligible contributions to both the byte and connection counts. We will characterize these applications in detail in § 5, but here we note that this indicates that most of the traffic in these categories consists of connections with modest—not tiny or huge—lengths.

In addition, the plot highlights the differences in traffic profile across time and area of the network monitored. For instance, the number of bytes transmitted for "backup" activities varies by a factor of roughly 5 from D_0 to D_4. This could be due to differences in the monitored locations, or different tracing times. Given our data collection techniques, we cannot distill trends from the data; however this is clearly a fruitful area for future work. We note that most of the application categories that significantly contribute to the traffic mix show a range of usage across the datasets. However, the percentage of connections in the "net-mgnt" and "misc" categories are fairly consistent across the datasets. This may be because a majority of the connections come from periodic probes and announcements, and thus have a quite stable volume.

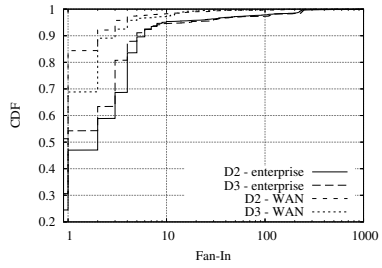
Finally, we note that multicast traffic constitutes a significant fraction of traffic in the "streaming", "name", and "net-mgnt" categories. We observe that 5–10% of all TCP/UDP payload bytes transmitted are in multicast streaming—i.e., more than the amount of traffic found in unicast streaming. Likewise, multicast traffic in "name" (SrvLoc) and "net-mgnt" (SAP) each constitutes 5–10% of all TCP/UDP connections. However, multicast traffic in the

remaining application categories was found to be negligible.

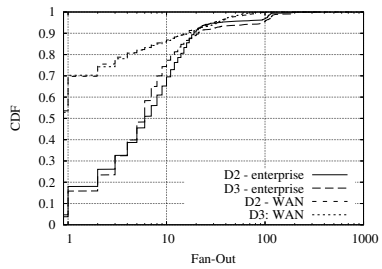
4 Origins and Locality

We next analyze the data to assess both the origins of traffic and the breadth of communications among the monitored hosts. First, we examine the origin of the flows in each dataset, finding that the traffic is clearly dominated by unicast flows whose source and destination are both within the enterprise (71–79% of flows across the five datasets). Another 2–3% of unicast flows originate within the enterprise but communicate with peers across the wide-area network, and 6–11% originate from hosts outside of the enterprise contacting peers within the enterprise. Finally, 5–10% of the flows use multicast sourced from within the enterprise and 4–7% use multicast sourced externally.

We next assess the number of hosts with which each monitored host communicates. For each monitored host H we compute two metrics: (i) *fan-in* is the number of hosts that originate conversations with H, while (ii) *fan-out* is the number of hosts to which H initiates conversations. We calculate these metrics in terms of both local traffic and wide-area traffic.



(a) Fan-in



(b) Fan-out

Figure 2: Locality in host communication.

Figure 2 shows the distribution of fan-in and fan-out for D_2 and D_3.[2] We observe that for both fan-in and fan-out, the hosts in our datasets generally have more peers within the enterprise than across the WAN, though with considerable variability. In particular, one-third to one-half of the hosts have only internal fan-in, and more than half with only internal fan-out — much more than the fraction of hosts with only external peers. This difference matches our intuition that local hosts will contact local servers (e.g., SMTP, IMAP, DNS, distributed file systems) more frequently than requesting services across the wide-area network, and is also consistent with our observation that a wider variety of applications are used only within the enterprise.

While most hosts have a modest fan-in and fan-out—over 90% of the hosts communicate with at most a couple dozen other hosts—some hosts communicate with scores to hundreds of hosts, primarily busy servers that communicate with large numbers of on- and off-site hosts (e.g., mail servers). Finally, the tail of the internal fan-out, starting around 100 peers/source, is largely due to the peer-to-peer communication pattern of SrvLoc traffic.

In keeping with the spirit of this paper, the data presented in this section provides a first look at origins and locality in the aggregate. Future work on assessing particular applications and examining locality within the enterprise is needed.

§ 5.1.1	Automated HTTP client activities constitute a significant fraction of internal HTTP traffic.
§ 5.1.2	IMAP traffic inside the enterprise has characteristics similar to wide-area email, except connections are longer-lived.
§ 5.1.3	Netbios/NS queries fail nearly 50% of the time, apparently due to popular names becoming stale.
§ 5.2.1	Windows traffic is intermingled over various ports, with Netbios/SSN (139/tcp) and SMB (445/tcp) used interchangeably for carrying CIFS traffic. DCE/RPC over "named pipes", rather than Windows File Sharing, emerges as the most active component in CIFS traffic. Among DCE/RPC services, printing and user authentication are the two most heavily used.
§ 5.2.2	Most NFS and NCP requests are reading, writing, or obtaining file attributes.
§ 5.2.3	Veritas and Dantz dominate our enterprise's backup applications. Veritas exhibits only client → server data transfers, but Dantz connections can be large in either direction.

Table 5: Example application traffic characteristics.

5 Application Characteristics

In this section we examine transport-layer and application-layer characteristics of individual application protocols. Table 5 provides a number of examples of the findings we

[2]Note, the figures in this paper are small due to space considerations. However, since we are focusing on high-level notions in this paper we ask the reader to focus on the general shape and large differences illustrated rather than the small changes and minor details (which are difficult to discern given the size of the plots).

make in this section.

We base our analysis on connection summaries generated by Bro [18]. As noted in § 2, D_1 and D_2 consist of traces that contain only the first 68 bytes of each packet. Therefore, we omit these two datasets from analyses that require in-depth examination of packet payloads to extract application-layer protocol messages.

Before turning to specific application protocols, however, we need to first discuss how we compute failure rates. At first blush, counting the number of failed connections/requests seems to tell the story. However, this method can be misleading if the client is automated and endlessly retries after being rejected by a peer, as happens in the case of NCP, for example. Therefore, we instead determine the number of *distinct operations* between *distinct host-pairs* when quantifying success and failure. Such operations can span both the transport layer (e.g., a TCP connection request) and the application layer (e.g., a specific name lookup in the case of DNS). Given the short duration of our traces, we generally find a specific operation between a given pair of hosts either nearly always succeeds, or nearly always fails.

5.1 Internal/External Applications

We first investigate applications categories with traffic in both the enterprise network and in the wide-area network: web, email and name service.

5.1.1 Web

HTTP is one of the few protocols where we find more wide-area traffic than internal traffic in our datasets. Characterizing wide-area Web traffic has received much attention in the literature over the years, e.g., [14, 3]. In our first look at modern enterprise traffic, we find internal HTTP traffic to be distinct from WAN HTTP traffic in several ways: (i) we observe that automated clients—scanners, bots, and applications running on top of HTTP—have a large impact on overall HTTP traffic characteristics; (ii) we find a lower fan-out per client in enterprise web traffic than in WAN web traffic; (iii) we find a higher connection failure rate within the enterprise; and (iv) we find heavier use of HTTP's *conditional* `GET` in the internal network than in the WAN. Below we examine these findings along with several additional traffic characteristics.

Automated Clients: In internal Web transactions we find three activities not originating from traditional user-browsing: *scanners*, *Google bots*, and programs running on top of HTTP (e.g., Novell *iFolder* and Viacom *NetMeeting*). As Table 6 shows, these activities are highly significant, accounting for 34–58% of internal HTTP requests and 59–96% of the internal data bytes carried over HTTP.

	Request			Data		
	D0/ent	D3/ent	D4/ent	D0/ent	D3/ent	D4/ent
Total	7098	16423	15712	602MB	393MB	442MB
scan1	20%	45%	19%	0.1%	0.9%	1%
google1	23%	0.0%	1%	45%	0.0%	0.1%
google2	14%	8%	4%	51%	69%	48%
ifolder	1%	0.2%	10%	0.0%	0.0%	9%
All	58%	54%	34%	96%	70%	59%

Table 6: Fraction of internal HTTP traffic from automated clients.

Including these activities skews various HTTP characteristics. For instance, both Google bots and the scanner have a very high "fan-out"; the scanner provokes many more "404 File Not Found" HTTP replies than standard web browsing; *iFolder* clients use `POST` more frequently than regular clients; and *iFolder* replies often have a uniform size of 32,780 bytes. Therefore, while the presence of these activities is the biggest difference between internal and wide-area HTTP traffic, we exclude these from the remainder of the analysis in an attempt to understand additional differences.

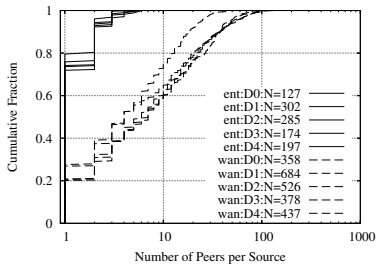


Figure 3: HTTP fan-out. The N in the key is the number of samples throughout the paper – in this case, the number of clients.

Fan-out: Figure 3 shows the distribution of fan-out from monitored clients to enterprise and WAN HTTP servers. Overall, monitored clients visit roughly an order of magnitude more external servers than internal servers. This seems to differ from the finding in § 4 that over all traffic clients tend to access more local peers than remote peers. However, we believe that the pattern shown by HTTP transactions is more likely to be the prevalent application-level pattern and that the results in § 4 are dominated by the fact that clients access a wider variety of applications. This serves to highlight the need for future work to drill down on the first, high-level analysis we present in this paper.

Connection Success Rate: Internal HTTP traffic shows success rates of 72–92% (by number of host-pairs), while the success rate of WAN HTTP traffic is 95–99%. The root cause of this difference remains a mystery. We note that the majority of unsuccessful internal connections are terminated with TCP RSTs by the servers, rather than going unanswered.

Conditional Requests: Across datasets and localities,

	Request		Data	
	enterprise	wan	enterprise	wan
text	18% – 30%	14% – 26%	7% – 28%	13% – 27%
image	67% – 76%	44% – 68%	10% – 34%	16% – 27%
application	3% – 7%	9% – 42%	57% – 73%	33% – 60%
Other	0.0% – 2%	0.3% – 1%	0.0% – 9%	11% – 13%

Table 7: HTTP reply by content type. "Other" mainly includes *audio*, *video*, and *multipart*.

	Bytes				
	D0/all	D1/all	D2/all	D3/all	D4/all
SMTP	152MB	1658MB	393MB	20MB	59MB
SIMAP	185MB	1855MB	612MB	236MB	258MB
IMAP4	216MB	2MB	0.7MB	0.2MB	0.8MB
Other	9MB	68MB	21MB	12MB	21MB

Table 8: Email Traffic Size

HTTP `GET` commands account for 95–99% of both the number of requests and the number of data bytes. The `POST` command claims most of the rest. One notable difference between internal and wide area HTTP traffic is the heavier use internally of conditional `GET` commands (i.e., a `GET` request that includes one of the `If-Modified-Since` headers, per [8]). Internally we find conditional `GET` commands representing 29–53% of web requests, while externally conditional `GET` commands account for 12–21% of the requests. The conditional requests often yield savings in terms of the number of data bytes downloaded in that conditional requests only account for 1–9% of the HTTP data bytes transfered internally and 1–7% of the data bytes transfered from external servers. We find this use of the conditional `GET` puzzling in that we would expect that attempting to save wide-area network resources (by caching and only updating content when needed) would be more important than saving local network resources. Finally, we find that over 90% of web requests are successful (meaning either the object requested is returned or that an HTTP 304 ("not modified") reply is returned in response to a conditional `GET`).

We next turn to several characteristics for which we do not find any *consistent* differences between internal and wide-area HTTP traffic.

Content Type: Table 7 provides an overview of object types for HTTP `GET` transactions that received a 200 or 206 HTTP response code (i.e., success). The *text*, *image*, and *application* content types are the three most popular, with *image* and *application* generally accounting for most of the requests and bytes, respectively. Within the *application* type, the popular subtypes include *javascript*, *octet stream*, *zip*, and *PDF*. The *other* content types are mainly *audio*, *video*, or *multipart* objects. We do not observe significant differences between internal and WAN traffic in terms of application types.

HTTP Responses: Figure 4 shows the distribution of HTTP response body sizes, excluding replies without a body. We see no significant difference between internal and WAN servers. The short vertical lines of the D_0/WAN curve reflect repeated downloading of javascripts from a particular website. We also find that about half the web sessions (i.e., downloading an entire web page) consist of one object (e.g., just an HTML page). On the other hand 10–20% of the web sessions in our dataset include 10 or more objects. We find no significant difference across datasets or server location (local or remote).

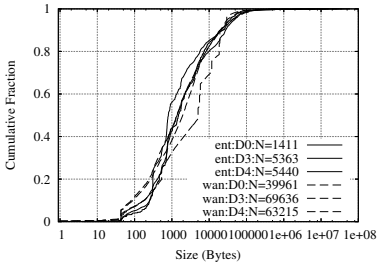


Figure 4: Size of HTTP reply, when present.

HTTP/SSL: Our data shows no significant difference in HTTPS traffic between internal and WAN servers. However, we note that in both cases there are numerous small connections between given host-pairs. For example, in D_4 we observe 795 short connections between a single pair of hosts during an hour of tracing. Examining a few at random shows that the hosts complete the SSL handshake successfully and exchange a pair of application messages, after which the client tears down the connection almost immediately. As the contents are encrypted, we cannot determine whether this reflects application level fail-and-retrial or some other phenomenon.

5.1.2 Email

Email is the second traffic category we find prevalent in both internally and over the wide-area network. As shown in Table 8, SMTP and IMAP dominate email traffic, constituting over 94% of the volume in bytes. The remainder comes from LDAP, POP3 and POP/SSL. The table shows a transition from IMAP to IMAP/S (IMAP over SSL) between D_0 and D_1, which reflects a policy change at LBNL restricting usage of unsecured IMAP.

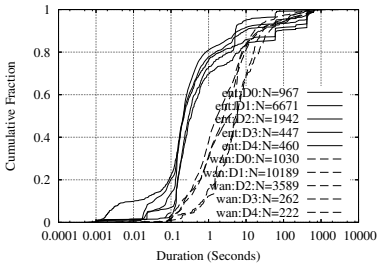
Datasets D_{0-2} include the subnets containing the main enterprise-wide SMTP and IMAP(/S) servers. This causes a difference in traffic volume between datasets D_{0-2} and D_{3-4}, and also other differences discussed below. Also, note that we conduct our analysis at the transport layer, since often the application payload is encrypted.

We note that the literature includes several studies of email traffic (e.g., [16, 10]), but none (that we are aware of) focusing on enterprise networks.

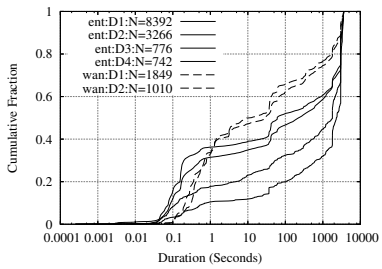
We first discuss areas where we find significant difference between enterprise and wide-area email traffic.

Connection Duration: As shown in Figure 5(a), the duration of internal and WAN SMTP connections generally differs by about an order of magnitude, with median durations around 0.2–0.4 sec and 1.5–6 sec, respectively. These results reflect the large difference in round-trip times (RTTs) experienced across the two types of network. SMTP sessions consist of both an exchange of control information and a unidirectional bulk transfer for the messages (and attachments) themselves. Both of these take time proportional to the RTT [15], explaining the longer SMTP durations.

In contrast, Figure 5(b) shows the distribution of IMAP/S connection durations across a number of our datasets. We leave off D_0 to focus on IMAP/S traffic, and D_{3-4} WAN traffic because these datasets do not include subnets with busy IMAP/S servers and hence have little wide-area IMAP/S traffic. The plot shows internal connections often last 1–2 orders of magnitude longer than wide-area connections. We do not yet have an explanation for the difference. The maximum connection duration is generally 50 minutes. While our traces are roughly 1 hour in length we find that IMAP/S clients generally poll every 10 minutes, generally providing only 5 observations within each trace. Determining the true length of IMAP/S sessions requires longer observations and is a subject for future work.



(a) SMTP

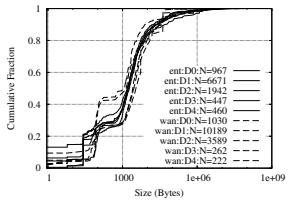


(b) IMAP/S

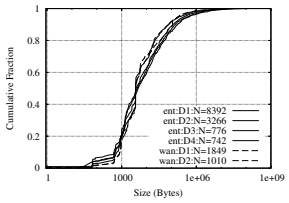
Figure 5: SMTP and IMAP/S connection durations.

We next focus on characteristics of email traffic that are similar across network type.

Connection Success Rate: Across our datasets we find that internal SMTP connections have success rates of 95–98%. SMTP connections traversing the wide-area network have success rates of 71–93% in D_{0-2} and 99-100% in D_{3-4}. Recall that D_{0-2} include heavily used SMTP servers and D_{3-4} do not, which likely explains the discrepancy. The success rate for IMAP/S connections is 99–100% across both locality and datasets.



(a) SMTP from client



(b) IMAP/S from server

Figure 6: SMTP and IMAP/S: flow size distribution

Flow Size: Internal and wide-area email traffic does not show significant differences in terms of connection sizes, as shown in Figure 6. As we would expect, the traffic volume of SMTP and IMAP/S is largely unidirectional (to SMTP servers and to IMAP/S clients), with traffic in the other direction largely being short control messages. Over 95% of the connections to SMTP servers and to IMAP/S clients remain below 1 MB, but both cases have significant upper tails.

5.1.3 Name Services

The last application category prevalent in both the internal and the wide-area traffic is domain name lookups. We observe a number of protocols providing name/directory services, including DNS, Netbios Name Service (Netbios/NS), Service Location Protocol (SrvLoc), SUN/RPC Portmapper, and DCE/RPC endpoint mapper. We also note that wide-area DNS has been studied by a number of researchers previously (e.g., [12]), however, our study of name lookups includes both enterprise traffic and non-DNS name services.

In this section we focus on DNS and Netbios/NS traffic, due to their predominant use. DNS appears in both wide-area and internal traffic. We find no large differences between the two types of DNS traffic except in response latency.

For both services a handful of servers account for most of the traffic, therefore the vantage point of the monitor can significantly affect the traffic we find in a trace. In particular, D_{0-2} do not contain subnets with a main DNS server,

and thus relatively few WAN DNS connections. Therefore, in the following discussion we only use D_{3-4} for WAN DNS traffic. Similarly, more than 95% of Netbios/NS requests go to one of the two main servers. D_{0-2} captures all traffic to/from one of these and D_{3-4} captures all traffic to both. Finally, we do not consider D_{1-2} in our analysis due to the lack of application payloads in those datasets (which renders our payload analysis inoperable).

Given those considerations, we now explore several characteristics of name service traffic.

Latency: We observe median latencies are roughly 0.4 msec and 20 msec for internal and external DNS queries, respectively. This expected result is directly attributable to the round-trip delay to on- and off-site DNS servers. Netbios/NS, on the other hand, is primarily used within the enterprise, with inbound requests blocked by the enterprise at the border.

Clients: A majority of DNS requests come from a few clients, led by two main SMTP servers that perform lookups in response to incoming mail. In contrast, we find Netbios/NS requests more evenly distributed among clients, with the top ten clients generating less than 40% of all requests across datasets.

Request Type: DNS request types are quite similar both across datasets and location of the peer (internal or remote). A majority of the requests (50–66%) are for `A` records, while 17–25% are for `AAAA` (IPv6) records, which seems surprisingly high, though we have confirmed a similar ratio in the wide-area traffic at another site. Digging deeper reveals that a number of hosts are configured to request both `A` and `AAAA` in parallel. In addition, we find 10–18% of the requests are for `PTR` records and 4–7% are for `MX` records.

Netbios/NS traffic is also quite similar across the datasets. 81–85% of requests consist of name queries, with the other prevalent action being to "refresh" a registered name (12–15% of the requests). We observe a number of additional transaction types in small numbers, including commands to register names, release names, and check status.

Netbios/NS Name Type: Netbios/NS includes a "type" indication in queries. We find that across our datasets 63–71% of the queries are for workstations and servers, while 22–32% of the requests are for domain/browser information.

Return Code: We find DNS has similar success (`NOERROR`) rates (77–86%) and failure (`NXDOMAIN`) rates (11–21%) across datasets and across internal and wide-area traffic. We observe failures with Netbios/NS 2–3 times more often: 36–50% of distinct Netbios/NS queries yield an `NXDOMAIN` reply. These failures are broadly distributed—they are not due to any single client, server, or query string. We speculate that the difference between the two protocols may be attributable to DNS representing an administratively controlled name space, while Netbios/NS uses a more distributed and loosely controlled mechanism for registering names, resulting in Netbios/NS names going "out-of-date" due to timeouts or revocations.

5.2 Enterprise-Only Applications

The previous subsection deals with application categories found in both internal and wide-area communication. In this section, we turn to analyzing the high-level and salient features of applications used only within the enterprise. Given the degree to which such protocols have not seen much exploration in the literature before, we aim for a broad rather than deep examination. A great deal remains for future work to develop the characterizations in more depth.

5.2.1 Windows Services

We first consider those services used by Windows hosts for a wide range of tasks, such as Windows file sharing, authentication, printing, and messaging. In particular, we examine Netbios Session Services (SSN), the Common Internet File System (SMB/CIFS), and DCE/RPC. We do not tackle the Netbios Datagram Service since it appears to be largely used *within* subnets (e.g., for "Network Neighborhoods"), and does not appear much in our datasets; and we cover Netbios/NS in § 5.1.3.

One of the main challenges in analyzing Windows traffic is that each communication scheme can be used in a variety of ways. For instance, TCP port numbers reveal little about the actual application: services can be accessed via multiple channels, and a single port can be shared by a variety of services. Hosts appear to interchangeably use CIFS via its well-known TCP port of 445 and via layering on top of Netbios/SSN (TCP port 139). Similarly, we note that DCE/RPC clients have two ways to find services: (i) using "named pipes" on top of CIFS (which may or may not be layered on top of Netbios/SSN) and (ii) on top of standard TCP and UDP connections without using CIFS, in which case clients consult the Endpoint Mapper to discover the port of a particular DCE/RPC service. Thus, in order to understand the Windows traffic we had to develop rich Bro protocol analyzers, and also merge activities from different transport layer channels. With this in place, we could then analyze various facets of the activities according to application functionality, as follows.

Connection Success Rate: As shown in Table 9, we observe a variety of connection success rates for different kinds of traffic: 82–92% for Netbios/SSN connections, 99–100% for Endpoint Mapper traffic, and a strikingly low 46–68% for CIFS traffic. For both Netbios/SSN and CIFS traffic we find the failures are not caused by a few erratic hosts, but rather are spread across hundreds of clients and dozens

	Host Pairs		
	Netbios/SSN	CIFS	Endpoint Mapper
Total	595 – 1464	373 – 732	119 – 497
Successful	82% – 92%	46% – 68%	99% – 100%
Rejected	0.2% – 0.8%	26% – 37%	0.0% – 0.0%
Unanswered	8% – 19%	5% – 19%	0.2% – 0.8%

Table 9: Windows traffic connection success rate (by number of host-pairs, for internal traffic only)

of servers. Further investigation reveals most of CIFS connection failures are caused by a number of clients connecting to servers on both the Netbios/SSN (139/tcp) and CIFS (445/tcp) port *in parallel*—since the two ports can be used interchangeably. The apparent intention is to use whichever port works while not incurring the cost of trying each in turn. We also find a number of the servers are configured to listen only on the Netbios/SSN port, so they reject connections to the CIFS port.

Netbios/SSN Success Rate: After a connection is established, a Netbios/SSN session goes through a handshake before carrying traffic. The success rate of the handshake (counting the number of distinct host-pairs) is 89–99% across our datasets. Again, the failures are not due to any single client or server, but are spread across a number of hosts. The reason for these failures merits future investigation.

CIFS Commands: Table 10 shows the prevalence of various types of commands used in CIFS channels across our datasets, in terms of both the number of commands and volume of data transferred. The first category, "SMB Basic", includes common commands used for session initialization and termination, and accounts for 24–52% of the messages across the datasets, but only 3–15% of the data bytes. The remaining categories indicate the tasks CIFS connections are used to perform. Interestingly, we find DCE/RPC pipes, rather than Windows File Sharing, make up the largest portion of messages (33–48%) and data bytes (32–77%) across datasets. Windows File Sharing constitutes 11–27% of messages and 8% to 43% of bytes. Finally, "LANMAN", a non-RPC named pipe for management tasks in "network neighborhood" systems, accounts for just 1–3% of the requests, but 3–15% of the bytes.

DCE/RPC Functions: Since DCE/RPC constitutes an important part of Windows traffic, we further analyze these calls over both CIFS pipes and stand-alone TCP/UDP connections. While we include all DCE/RPC activities traversing CIFS pipes, our analysis for DCE/RPC over stand-alone TCP/UDP connections may be incomplete for two reasons. First, we identify DCE/RPC activities on ephemeral ports by analyzing Endpoint Mapper traffic. Therefore, we will miss traffic if the mapping takes place before our trace collection begins, or if there is an alternate method to discover the server's ports (though we are not aware of any other such method). Second, our analysis tool currently cannot parse DCE/RPC messages sent over UDP. While this may cause our analysis to miss services that only use UDP, DCE/RPC traffic using UDP accounts for only a small fraction of all DCE/RPC traffic.

Table 11 shows the breakdown of DCE/RPC functions. Across all datasets, the *Spoolss* printing functions—and `WritePrinter` in particular—dominate the overall traffic in D_{3-4}, with 63–91% of the requests and 94–99% of the data bytes. In D_0, *Spoolss* traffic remains significant, but not as dominant as user authentication functions (*NetLogon* and *LsaRPC*), which account for 68% of the requests and 52% of the bytes. These figures illustrate the variations present within the enterprise, as well as highlighting the need for multiple vantage points when monitoring. (For instance, in D_0 we monitor a major authentication server, while D_{3-4} includes a major print server.)

5.2.2 Network File Services

NFS and NCP[3] comprise the two main network file system protocols seen within the enterprise and this traffic is nearly always confined to the enterprise.[4] We note that several trace-based studies of network file system characteristics have appeared in the filesystem literature (e.g., see [7] and enclosed references). We now investigate several aspects of network file system traffic.

Aggregate Sizes: Table 12 shows the number of NFS and NCP connections and the amount of data transferred for each dataset. In terms of connections, we find NFS more prevalent than NCP, except in D_0. In all datasets, we find NFS transfers more data bytes per connection than NCP. As in previous sections, we see the impact of the measurement location in that the relative amount of NCP traffic is much higher in D_{0-2} than in D_{3-4}. Finally, we find "heavy hitters" in NFS/NCP traffic: the three most active NFS host-pairs account for 89–94% of the data transfered, and for the top three NCP host-pairs, 35–62%.

Keep-Alives: We find that NCP appears to use TCP keep-alives to maintain long-lived connections and detect runaway clients. Particularly striking is that 40–80% of the NCP connections across our datasets consist *only* of periodic retransmissions of 1 data byte and therefore do not include any real activity.

UDP vs. TCP We had expected that NFS-over-TCP would heavily dominate modern use of NFS, but find this is not the case. Across the datasets, UDP comprises 66%/16%/31%/94%/7% of the payload bytes, an enormous range. Overall, 90% of the NFS host-pairs use UDP, while only 21% use TCP (some use both).

[3] NCP is the *Netware Control Protocol*, a veritable kitchen-sink protocol supporting hundreds of message types, but primarily used within the enterprise for file-sharing and print service.

[4] We found three NCP connections with remote hosts across all our datasets!

	Request			Data		
	D0/ent	D3/ent	D4/ent	D0/ent	D3/ent	D4/ent
Total	49120	45954	123607	18MB	32MB	198MB
SMB Basic	36%	52%	24%	15%	12%	3%
RPC Pipes	48%	33%	46%	32%	64%	77%
Windows File Sharing	13%	11%	27%	43%	8%	17%
LANMAN	1%	3%	1%	10%	15%	3%
Other	2%	0.6%	1.0%	0.2%	0.3%	0.8%

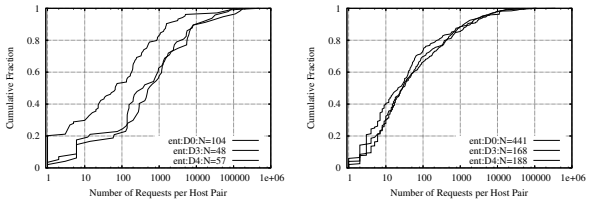
Table 10: CIFS command breakdown. "SMB basic" includes the common commands shared by all kinds of higher level applications: protocol negotiation, session setup/tear-down, tree connect/disconnect, and file/pipe open.

	Request			Data		
	D0/ent	D3/ent	D4/ent	D0/ent	D3/ent	D4/ent
Total	14191	13620	56912	4MB	19MB	146MB
NetLogon	42%	5%	0.5%	45%	0.9%	0.1%
LsaRPC	26%	5%	0.6%	7%	0.3%	0.0%
Spoolss/WritePrinter	0.0%	29%	81%	0.0%	80%	96%
Spoolss/other	24%	34%	10%	42%	14%	3%
Other	8%	27%	8%	6%	4%	0.6%

Table 11: DCE/RPC function breakdown.

Request Success Rate: If an NCP connection attempt succeeds (88–98% of the time), about 95% of the subsequent requests also succeed, with the failures dominated by "File/Dir Info" requests. NFS requests succeed 84% to 95% of the time, with most of the unsuccessful requests being "lookup" requests for non-existing files or directories.

Requests per Host-Pair: Since NFS and NCP both use a message size of about 8 KB, multiple requests are needed for large data transfers. Figure 7 shows the number of requests per client-server pair. We see a large range, from a handful of requests to hundreds of thousands of requests between a host-pair. A related observation is that the interval between requests issued by a client is generally 10 msec or less.



(a) NFS (b) NCP

Figure 7: NFS/NCP: number of requests per client-server pair, for those with at least one request seen.

Breakdown by Request Type: Table 13 and 14 show that in both NFS and NCP, file read/write requests account for the vast majority of the data bytes transmitted, 88–99% and 92–98% respectively. In terms of the number of requests, obtaining file attributes joins read and write as a dominant function. NCP file searching also accounts for 7–16% of the requests (but only 1–4% of the bytes). Note that NCP provides services in addition to remote file access, e.g., directory service (NDS), but, as shown in the table, in our datasets NCP is predominantly used for file sharing.

	Connections	Bytes
VERITAS-BACKUP-CTRL	1271	0.1MB
VERITAS-BACKUP-DATA	352	6781MB
DANTZ	1013	10967MB
CONNECTED-BACKUP	105	214MB

Table 15: Backup Applications

Request/Reply Data Size Distribution: As shown in Figure 8(a,b), NFS requests and replies have clear dual-mode distributions, with one mode around 100 bytes and the other 8 KB. The latter corresponds to write requests and read replies, and the former to everything else. NCP requests exhibit a mode at 14 bytes, corresponding to read requests, and each vertical rise in the NCP reply size figure corresponds to particular types of commands: 2-byte replies for completion codes only (e.g. replying to "WriteFile" or reporting error), 10 bytes for "GetFileCurrentSize", and 260 bytes for (a fraction of) "ReadFile" requests.

5.2.3 Backup

Backup sessions are a rarity in our traces, with just a small number of hosts and connections responsible for a huge data volume. Clearly, this is an area where we need longer traces. That said, we offer brief characterizations here to convey a sense of its nature.

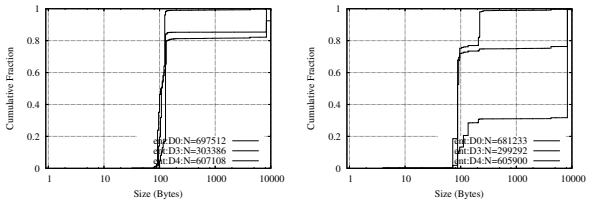
We find three types of backup traffic, per Table 15: two internal traffic giants, Dantz and Veritas, and a much smaller, "Connected" service that backs up data to an external site. Veritas backup uses separate control and

	Connections					Bytes				
	D0/all	D1/all	D2/all	D3/all	D4/all	D0/all	D1/all	D2/all	D3/all	D4/all
NFS	1067	5260	4144	3038	3347	6318MB	4094MB	3586MB	1030MB	1151MB
NCP	2590	4436	2892	628	802	777MB	2574MB	2353MB	352MB	233MB

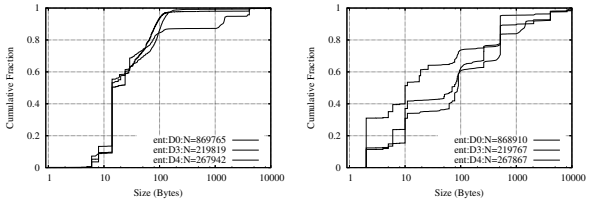
Table 12: NFS/NCP Size

	Request			Data		
	D0/ent	D3/ent	D4/ent	D0/ent	D3/ent	D4/ent
Total	697512	303386	607108	5843MB	676MB	1064MB
Read	70%	25%	1%	64%	92%	6%
Write	15%	1%	19%	35%	2%	83%
GetAttr	9%	53%	50%	0.2%	4%	5%
LookUp	4%	16%	23%	0.1%	2%	4%
Access	0.5%	4%	5%	0.0%	0.4%	0.6%
Other	2%	0.9%	2%	0.1%	0.2%	1%

Table 13: NFS requests breakdown.



(a) NFS request (b) NFS reply



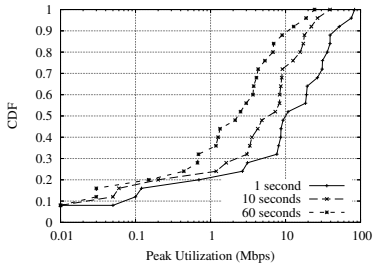
(c) NCP request (d) NCP reply

Figure 8: NFS/NCP: request/reply data size distribution (NFS/NCP message headers are not included)

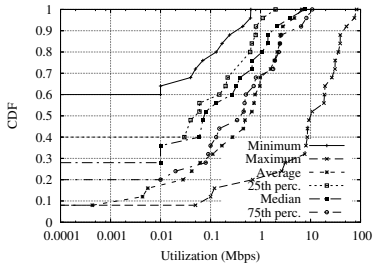
data connections, with the data connections in the traces all reflecting one-way, client-to-server traffic. Dantz, on the other hand, appears to transmit control data within the same connection, and its connections display a degree of bi-directionality. Furthermore, the server-to-client flow sizes can exceed 100 MB. This bi-directionality does not appear to reflect backup vs. restore, because it exists not only *between* connections, but also *within* individual connections—sometimes with tens of MB in both directions. Perhaps this reflects an exchange of fingerprints used for compression or incremental backups or an exchange of validation information after the backup is finished. Alternatively, this may indicate that the protocol itself may have a peer-to-peer structure rather than a strict server/client delineation. Clearly this requires further investigation with longer trace files.

6 Network Load

A final aspect of enterprise traffic in our preliminary investigation is to assess the load observed within the enterprise. One might naturally assume that campus networks are underutilized, and some researchers aim to develop mechanisms that leverage this assumption [19]. We assess this assumption using our data.



(a) Peak Utilization



(b) Utilization

Figure 9: Utilization distributions for D_4.

	Request			Data		
	D0/ent	D3/ent	D4/ent	D0/ent	D3/ent	D4/ent
Total	869765	219819	267942	712MB	345MB	222MB
Read	42%	44%	41%	82%	70%	82%
Write	1%	21%	2%	10%	28%	11%
FileDirInfo	27%	16%	26%	5%	0.9%	3%
File Open/Close	9%	2%	7%	0.9%	0.1%	0.5%
File Size	9%	7%	5%	0.2%	0.1%	0.1%
File Search	9%	7%	16%	1%	0.6%	4%
Directory Service	2%	0.7%	1%	0.7%	0.1%	0.4%
Other	3%	3%	2%	0.2%	0.1%	0.1%

Table 14: NCP requests breakdown.

Due to limited space, we discuss only D_4, although the other datasets provide essentially the same insights about utilization. Figure 9(a) shows the distribution of the *peak* bandwidth usage over 3 different timescales for each trace in the D_4 dataset. As expected, the plot shows the networks to be less than fully utilized at each timescale. The 1 second interval does show network saturation (100 Mbps) in some cases. However, as the measurement time interval increases the peak utilization drops, indicating that saturation is short-lived.

Figure 9(b) shows the distributions of several metrics calculated over 1 second intervals. The "maximum" line on this plot is the same as the "1 second" line on the previous plot. The second plot concretely shows that typical (over time) network usage is 1–2 orders of magnitude less than the peak utilization and 2–3 orders less than the capacity of the network (100 Mbps).

We can think of packet loss as a second dimension for assessing network load. We can form estimates of packet loss rates using TCP retransmission rates. These two might not fully agree, due to (i) TCP possibly retransmitting unnecessarily, and (ii) TCP adapting its rate in the presence of loss, while non-TCP traffic will not. But the former should be rare in LAN environments (little opportunity for retransmission timers to go off early), and the latter arguably at most limits our analysis to applying to the TCP traffic, which dominates the load (cf. Table 3).

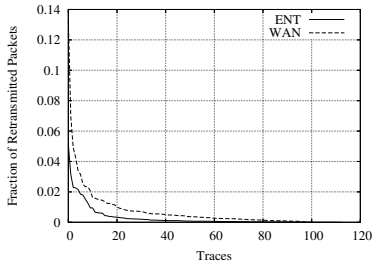


Figure 10: TCP Retransmission Rate Across Traces (for traces with at least 1000 packets in the category)

We found a number of spurious 1 byte retransmissions due to TCP keep-alives by NCP and SSH connections. We exclude these from further analysis because they do not indicate load imposed on network elements. Figure 10 shows the remaining retransmission rate for each trace in all our datasets, for both internal and remote traffic. In the vast majority of the traces, the retransmission rate remains less than 1% for both. In addition, the retransmission rate for internal traffic is less than that of traffic involving a remote peer, which matches our expectations since wide-area traffic traverses more shared, diverse, and constrained networks than does internal traffic. (While not shown in the Figure, we did not find any correlation between internal and wide-area retransmission rates.)

We do, however, find that the internal retransmission rate sometimes eclipses 2%—peaking at roughly 5% in one of the traces. Our further investigation of this last trace found the retransmissions dominated by a single Veritas backup connection, which transmitted 1.5 M packets and 2 GB of data from a client to a server over one hour. The retransmissions happen almost evenly over time, and over one-second intervals the rate peaks at 5 Mbps with a 95^{th} percentile around 1 Mbps. Thus, the losses appear due to either significant congestion in the enterprise network downstream from our measurement point, or a network element with flaky NIC (reported in [22] as not a rare event).

We can summarize these findings as: packet loss within an enterprise appears to occur significantly less than across the wide-area network, as expected; but exceeds 1% a non-negligible proportion of the time.

7 Summary

Enterprise networks have been all but ignored in the modern measurement literature. Our major contribution in this paper is to provide a broad, high-level view of numerous aspects of enterprise network traffic. Our investigation runs the gamut from re-examining topics previously studied for wide-area traffic (e.g., web traffic), to investigating new types of traffic not assessed in the literature to our knowledge (e.g., Windows protocols), to testing assumptions about enterprise traffic dynamics (e.g., that such

networks are mostly underutilized).

Clearly, our investigation is only an initial step in this space. An additional hope for our work is to inspire the community to undertake more in-depth studies of the raft of topics concerning enterprise traffic that we could only examine briefly (or not at all) in this paper. Towards this end, we are releasing anonymized versions of our traces to the community [1].

Acknowledgments

We thank Sally Floyd for interesting discussions that led to § 6, Craig Leres for help with several tracing issues and Martin Arlitt and the anonymous reviewers for useful comments on a draft of this paper. Also, we gratefully acknowledge funding support for this work from NSF grants 0335241, 0205519, and 0335214, and DHS grant HSHQPA4X03322.

References

[1] LBNL Enterprise Trace Repository, 2005. http://www.icir.org/enterprise-tracing/.

[2] W. Aiello, C. Kalmanek, P. McDaniel, S. Sen, O. Spatscheck, and K. van der Merwe. Analysis of Communities Of Interest in Data Networks. In *Proceedings of Passive and Active Measurement Workshop*, Mar. 2005.

[3] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In *ACM SIGMETRICS*, pages 151–160, July 1998.

[4] R. Cáceres. Measurements of wide area internet traffic. Technical report, 1989.

[5] R. Caceres, P. Danzig, S. Jamin, and D. Mitzel. Characteristics of Wide-Area TCP/IP Conversations. In *ACM SIGCOMM*, 1991.

[6] P. Danzig, S. Jamin, R. Cáceres, D. Mitzel, and D. Estrin. An Empirical Workload Model for Driving Wide-area TCP/IP Network Simulations. *Internetworking: Research and Experience*, 3(1):1–26, 1992.

[7] D. Ellard, J. Ledlie, P. Malkani, and M. Seltzer. Passive NFS Tracing of Email and Research Workloads. In *USENIX Conference on File and Storage Technologies*, 2003.

[8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1*, jun. RFC 2616.

[9] H. J. Fowler and W. E. Leland. Local area network traffic characteristics, with implications for broadband network congestion management. *IEEE Journal on Selected Areas in Communications*, SAC-9:1139–49, 1991.

[10] L. Gomes, C. Cazita, J. Almeida, V. Almeida, and W. M. Jr. Characterizing a SPAM Traffic. In *Internet Measurement Conference*, Oct. 2004.

[11] R. Gusella. A measurement study of diskless workstation traffic on an Ethernet. *IEEE Transactions on Communications*, 38(9):1557–1568, Sept. 1990.

[12] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. In *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

[13] M. Lottor. Internet Growth (1981-1991), Jan. 1992. RFC 1296.

[14] B. Mah. An Empirical Model of HTTP Network Traffic. In *Proceedings of INFOCOM 97*, Apr. 1997.

[15] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *ACM SIGCOMM*, Sept. 1998.

[16] V. Paxson. Empirically-Derived Analytic Models of Wide-Area TCP Connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, Aug. 1994.

[17] V. Paxson. Growth Trends in Wide-Area TCP Connections. *IEEE Network*, 8(4):8–17, July/August 1994.

[18] V. Paxson. Bro: A system for detecting network intruders in real time. *Computer Networks*, December 1999.

[19] P. Sarolahti, M. Allman, and S. Floyd. Evaluating Quick-Start for TCP. May 2005. Under submission.

[20] S. Sen and J. Wang. Analyzing Peer-to-Peer Traffic Across Large Networks. In *Internet Measurement Workshop*, pages 137–150, Nov. 2002.

[21] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb. A case study of OSPF behavior in a large enterprise network. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 217–230, New York, NY, USA, 2002. ACM Press.

[22] J. Stone and C. Partridge. When The CRC and TCP Checksum Disagree. In *ACM SIGCOMM*, Sept. 2000.

[23] G. Tan, M. Poletto, J. Guttag, and F. Kaashoek. Role Classification of Hosts within Enterprise Networks Based on Connection Patterns. In *Proceedings of USENIX Annual Technical Conference*, June 2003.