UNIVERSITY OF CALIFORNIA, SAN DIEGO

Internet traffic characterization

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Computer Science & Engineering (CSE)

by

Kimberly C. Claffy

Committee in charge:

Professor George C. Polyzos, Chairperson
Professor Sidney Karin
Professor Bruce N. Lehman
Professor Joseph Pasquale
Professor Ramesh R. Rao

1994

The dissertation of Kimberly C. Claffy is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

1994

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am fortunate to have been advised by George Polyzos while at UCSD. He taught me a great deal about how to do research on untidy problems. His theoretical perspective also prevented me from falling headlong into the SNMP abyss.

I thank Sid Karin for accepting my stubborn persistence in wanting to study real networks, and yet keeping all my objectives thesis-worthy. He introduced me to the resources and collaboration of the San Diego Supercomputer Center, without which this research would not have been possible. In particular, he suggested I consult Hans-Werner Braun, with whom George and I wrote a proposal to the National Science Foundation to fund my work (NSF grant NCR-9119473). Hans-Werner provided endless guidance and support, and did his best to keep my research grounded in reality. He also made sure relevant results made their way back to NSF, so Steven Wolff could see them light up his very own SGI (and use them to improve the infrastructure). Fred Baker gave me the gift of a real router vendor perspective. Jordan Becker and many others at ANS, IBM, and Merit further ensured that my results were of interest to the Real World. Harvey Fraser and IBM provided a wonderful RS6000 workstation, with exceptional service support, as part of a joint study agreement that also facilitated this research. The SGI Indigos were also not shabby at sniffing packet headers.

I am also grateful to the members of my doctoral committee, Professors George Polyzos, Joe Pasquale, Ramesh Rao, Bruce Lehman, and Sidney Karin, for their valuable suggestions and feedback. I would also like to thank Vern Paxson and Roger Bohn for their comments, and Bruce Lehman for his periodic reminders that I really had nothing to be depressed about, what else was I going to do with my life anyway. (I'll let him know if I decide.) Many other researchers have also commented on my work during conferences, workshops and visits. I am thankful to all of them.

Sally Floyd provided a role model although she may not know it. John May kept me in line many times through countless runs in and out of wussdom and abjectly low self-confidence (mostly in). I could never have done this alone. (Who'd want to.)

Finally, to my parents, who gave me my wings, and continue to be the wind beneath them. *Te amo.*

VITA

1989	B.S., Stanford University
1991	M.S., Computer Science and Engineering University of California, San Diego
1989–1994	Research Assistant, Department of CSE University of California, San Diego
1991–1994	Student Fellow, Applied Network Research San Diego Supercomputer Center, San Diego
1994	Doctor of Philosophy, Computer Science and Engineering University of California, San Diego

PUBLICATIONS

R. Bohn, H.-W. Braun K. Claffy and S. Wolff, "Mitigating the coming Internet crunch: multiple service levels via Precedence," in Special Issue on Quality of Service of *Journal of High Speed Networks*, also SDSC Applied Network Research TR GA-A21530, forthcoming 1994.

K. Claffy, H.-W. Braun and G. C. Polyzos, "Tracking long-term growth of the NSFNET backbone", in *Communications of the ACM* August 1994.

K. Claffy, G. C. Polyzos and H.-W. Braun, "Application of Sampling Methodologies to Network Traffic Characterization", in *Proc. SIGCOMM '93*, pp. 194-203, September 1993, SDSC Applied Network Research TR GA-A21239.

H.-W. Braun, K. Claffy, and G. Polyzos, "A Framework for Flow-based Accounting on the Internet", in *Proc. IEEE SICON/ISIE '93*, pp. 847-851, September 1993, SDSC Applied Network Research TR GA-A21358.

K. Claffy, H.-W. Braun and G. C. Polyzos, "Long-term Traffic Aspects of the NSFNET", in *Proc. INET'93*; pp. CBA:1-10, SDSC Applied Network Research TR GA-A21238, August 1993.

K. Claffy and H.-W. Braun, "Network Analysis in Support of Internet Policy Requirements", in *Proc. INET'93*, pp. FAC:1-11, August 1993, SDSC Applied Network Research TR GA-A21253.

K. Claffy, H.-W. Braun and G. C. Polyzos, "Measurement Considerations for Assessing Unidirectional Latencies", *Internetworking: Research and Experience* 1993 4(3), pp. 121-132, SDSC Applied Network Research TR GA-A21018, UCSD TR CS92-253.

K. Claffy, H.-W. Braun and G. C. Polyzos, "Traffic Characteristics of the T1 NSFNET Backbone", in *Proc. of IEEE INFOCOM'93*, pp. 885-892, April 1993, SDSC Applied Network Research TR GA-A21019, UCSD TR CS92-252.

K. Claffy and H.-W. Braun, "Network analysis issues for a public Internet", in *Proc. of Public Access to the Internet*, workshop at the John F. Kennedy School of Government, Harvard University; SDSC Applied Network Research TR GA-A21350, May 1993.

H.-W. Braun, B. Chinoy, K. Claffy and G. C. Polyzos, "Analysis and modeling of wide area networks: annual report", SDSC Applied Network Research TR, GA-A21648, February 1994.

H.-W. Braun, B. Chinoy, K. Claffy and G. C. Polyzos, "Analysis and modeling of wide area networks: annual status report", SDSC Applied Network Research TR GA-A21224, February 1993.

H.-W. Braun, K. Claffy, B. Chinoy, and G. C. Polyzos, "Analysis and modeling tools for high-speed networks: project status report", SDSC ANR TR GA-A21224, UCSD TR CS92-237, April 1992.

T. Asaba, K. Claffy, O. Nakamura and J. Murai, "An Analysis of International Academic Research Network Traffic between Japan and other Nations", in *Proc. INET '92*, pp. 431-330, Kobe, Japan, June 1992.

K. Claffy and G. C. Polyzos, "Location Transparent Connection Management: a Survey of Protocol Issues", in *Proc. 11th Annual IEEE International Phoenix Conference on Computers and Communications*, April 1992.

R. Aiken (NSF), P. Ford (LANL), and H.-W. Braun, K. Claffy (editor), "NSF implementation plan for interagency interim NREN", *Journal of High Speed Networks*, 2(1), pp. 1-25, 1993; SDSC Applied Network Research TR GA-21174.

F. Teraoka and K. Claffy, "Virtual Internet Protocol: implementation and evaluation", (in Japanese), Sony Computer Science Laboratory Technical Report, September 1991

FIELDS OF STUDY

Major Field: Computer Science and Engineering
Internet traffic characterization
Advisor: Professor George C. Polyzos

Minor Field: Network policy
Internet policy issues
Hans-Werner Braun

Internet economic issues
Hans-Werner Braun and Professor Roger Bohn

ABSTRACT OF THE DISSERTATION

Internet traffic characterization

by

Kimberly C. Claffy
Doctor of Philosophy in Computer Science and Engineering (CSE)
University of California, San Diego, 1994
Professor George C. Polyzos, Chair

Traffic statistics normally collected during day-to-day operation of wide-area datagram networks are frequently insufficient for researchers to use in studying the workloads and performance of these realistic environments. As wide-area networks become more ubiquitous and service expectations rise, current methods for collecting data will become even less suitable. We examine ways to improve techniques for statistics collection so that the resulting data will enable researchers, and indeed service providers themselves, to develop more accurate Internet traffic models.

We first provide a taxonomy of traffic characterization tasks. We then use operationally collected statistics to characterize traffic of the T1 and T3 NSFNET backbones. Because current infrastructural statistics collection is oriented toward either short term operational requirements or periodic simplistic traffic reports to funding agencies, this data is often not conducive to assessing network workload or performance; we evaluate to what extent they are useful for tasks in the taxonomy, and propose improvements in current statistics collection architectures, with particular application to the NSFNET backbone. We include an investigation of the effects of sampling to characterize traffic and evaluate performance in a high-speed wide-area network environment.

In the second part of the thesis we focus on items in the outlined taxonomy that are not conducive to investigation using operationally collected statistics. These items mostly involve short-term aspects of Internet flows, which operationally collected statistics fail to expose. We develop a general methodology for use in assessing Internet flow profiles and their impact on an aggregate Internet workload. Our methodology for profiling flows differs from many previous studies that have concentrated on end-point definitions of flows defined by TCP connections using the TCP SYN and FIN control mechanism. We focus on the IP layer and define flows based on traffic satisfying various temporal and spatial locality conditions, as observed at internal points of the network. We first define the parameter space and then concentrate on metrics characterizing both individual flows and the aggregate flow. Metrics of individual flows include: volume in packets and bytes per flow, and flow duration. Metrics of the aggregate flow, or workload characteristics from the network perspective, include: counts of the number of active, new, and timed out flows per time interval; flow interarrival and arrival processes; and flow locality metrics. Applying the methodology to our measurements yields significant observations of the Internet infrastructure, which have implications for performance requirements of routers at Internet hotspots, general and specialized flow-based routing algorithms, future usage-based accounting requirements, and traffic prioritization.

Finally, we discuss trends that will affect how Internet service providers collect statistics in the future. Improvements in operational statistics collection, such as support for flow assessment, will help networking activities along various time horizons, from defining service quality patterns to long-term capacity planning. We offer a unique combination of operational and research perspectives, allowing us to reduce the gaps among (1) what network service providers need; (2) what statistics service providers can provide; and (3) what network analysis requires.

Chapter 1

Introduction

Never will a man penetrate deeper into error than when he is continuing on a road that has led him to great success.

– *Friederich von Hayek,* Counterrevolution of Science

A model is an artifice for helping you convince yourself that you understand more about a system than you do.

1.1 The problem

Existing literature in wide area network traffic characterization, both in the analytical and performance measurement domains, indicate that wide area networking technology has advanced at a far faster rate than has the analytical and theoretical understanding of network behavior. The slower and more containable realms of years ago were amenable to characterization with closed-form mathematical expressions, which allowed reasonably accurate prediction of performance metrics such as queue lengths and network delays.

Traditional mathematical modeling techniques, in particular queueing theory, have met with little success in today's networking environments. However, the need for network analysis has not diminished; on the contrary, realistic models and methodologies for understanding network behavior play an even more essential role in facilitating future evolution into gigabit/sec speeds and beyond. It seems inevitable, however, that simulation and empirical techniques to describe traffic behavior will play a larger role than traditional mathematical techniques have played in the past.

For multiple reasons, the characterization of aggregate Internet traffic, including on backbone networks, has so far not received a wealth of attention. A primary reason is the forced prioritization of efforts in operational environments, where service providers must concentrate principally on day-to-day requirements, leaving little time and resources to devote to systematic data acquisition, not to mention longer term analysis and traffic modeling. The work that does exist points to a danger in current network research: a gap between the unambiguous results of confined experiments which target isolated environments, and the largely unknown characteristics of the extensive Internet infrastructure that is heading toward global ubiquity. The target of my research, empirical investigation of high speed backbone networks which aggregate substantial amounts of traffic, is to shorten this gap, and contribute to a greater understanding of real computer networks of pervasive scale.

An obvious obstacle to the alignment of analytic studies with current practitioners is the fact that the former typically answer questions which the latter typically do not ask. By leveraging the ability of theorists to pose well-formed questions about network behavior, we can facilitate the top priority of network service providers: the effective implementation, management, and evolution of large scale infrastructure. In turn, by bringing the questions of pragmatic interest from the service providers to the network researchers, we can ensure the direct applicability of theoretical investigations to real world networks.

For example, service providers typically cannot afford the luxury of interest in end-to-end performance. They must attend to metrics of aggregate traffic volume as it is switched through the network nodes and its interfaces, both in terms of total packets, as well as individual packet payload. Also of interest is where the traffic is sourced and sent, and what applications are gaining or waning in popularity. Analytic studies have traditionally ignored such questions, directing themselves, for example, toward queueing theoretic questions of steady state queue length and expected mean delay. In turn, often these analytic studies, which are usually of small, artificially confined environments, cannot confirm their results with empirical data, because operational infrastructures many times do not support collection of the kind of data that would allow confirmation. The result has been a rift between theorists designing networks on paper based on formally defined performance expectations, and those building them based on practical experience. A larger systems approach may allow the realistic applicability of well-intended laboratory efforts.

One example of a successful Internet workload study offers insight into the potential of this systems approach for Internet evolution. In 1993, Advanced Network Services (ANS), the current service provider of the NSFNET backbone, was preparing for deployment of a new switching on the T3 network, specifically a replacement of the T3 interface card with a higher performance one. Card interoperability constrained their network transition; not only do the two cards have quite different performance characteristics, they also did not interoperate over serial links but rather only via a special bus. A smooth migration from the old to the new cards required finding internal link metrics that would prevent overloading of the hybrid links (i.e., machines with both cards installed) and at the same time minimize round-trip delay where possible. ANS engineers used historical SNMP and NNstat data of backbone transit traffic, which provided backbone link utilization values and the source and destination of the contributing traffic. They used this information to simulate performance of the existing and hybrid routers using different link metrics, and then used the most successful simulated metric configurations to guide the 6 week upgrade[1]. Similar analysis has served ANS during other upgrades of the T3 network.

1.2 Overview of thesis

This chapter presents the thesis problem, and how the thesis contributes to the field of research on wide area datagram networks, in particular, of the Internet. In the next chapter we present a taxonomy of Internet workload characteristics. We survey related work in several areas of the taxonomy that are relevant to the research in this dissertation.

Chapter 3 investigates existing operational statistics on the T1 and T3 NSFNET backbones, including the architecture for statistics collection and the extent of their usefulness for workload characterization. Chapter 4 analyzes the effect of a specific factor that constrains the usefulness of operationally collected statistics on wide-area network infrastructures, namely traffic sampling.

In the second part of the thesis we focus on items in the outlined taxonomy that are difficult or impossible to investigate using operationally collected statistics. These items mostly regard short-term aspects of Internet flows, which operational statistics collection cannot address, due to both its collection and aggregation granularity and the often forced use of sampling rather than comprehensive data collection. In Chapter 5 we develop a general methodology for assessing Internet flow profiles and their impact on an aggregate Internet workload. The methodology we describe in Chapter 5 structures the metrics we present in Chapters 6 and 7, where we apply it to direct packet measurements from a range of locations in the Internet fabric. These metrics fall into two categories: metrics of individual flows (Chapter 6) and metrics of the aggregate traffic flow (Chapter 7). Metrics of individual flows include: volume in packets and bytes per flow, and flow duration. Metrics of the aggregate flow, or workload characteristics from the network perspective, include: counts of the number of active, new, and timed out flows per time interval; the flow interarrival process; and flow locality metrics.

In Chapter 8 we discuss four trends that will affect how Internet service providers collect statistics in the future. These trends will also force service providers to give more attention to data collection than they have in the past. We provide evidence that service providers should in particular consider flow assessment as an ongoing component of their statistics collection architecture. In Chapter 9 we summarize our results and mention open areas for continued research in this area.

1.3 Contribution of our work

This thesis centers around the provision of empirical results on traffic characteristics of the Internet. The core contributions of the research encompass five areas:

1. provision and refinement of empirical results on workload characteristics of the NSFNET backbone;
2. analysis of the effects of sampling to characterize traffic in a high-speed wide-area network environment;
3. development of methodology for characterizing Internet flows from a network layer perspective
4. analysis of metrics derived from above methodology, divided into individual and aggregate flow metrics
5. evaluation and improvement of the data acquisition methodology of wide-area networks, with particular application to the NSFNET backbone.

The first two components differ from previous studies in their focus on existing operational wide area network infrastructures. We know of no studies on the utility of operational statistics collected on wide area backbone infrastructures for workload characterization tasks. The third and fourth components are unique contributions in the generality of the methodology they provide for characterizing IP network flows; its range of applicability exceeds any of which we are aware in the literature. Among the Internet research problems addressable by this approach to the definition and characterization of network flows are route caching, resource reservation at multiple service levels, usage based accounting, and the integration of IP traffic over an Asynchronous Transfer Mode (ATM) fabric.

Figure 1.1 illustrates how our work fits into existing work in the literature. Although many areas of Internet research on protocols and their performance are somewhat relevant, three core areas of research on datagram networks serve as inspiration for our work. First, local area network (LAN) traffic characterization studies, which are fairly comprehensive in their assessment of traffic, e.g., on an Ethernet [2] [3]. Second, wide area network (WAN) infrastructural traffic characterization studies, which focus on the use of operational statistics from wide area backbone infrastructures to investigate specific issues, e.g., routing [4], packet trains [5]. Finally, WAN traffic characterization studies which focus on a single or a few attachment points to transit networks to investigate shorter-term aspects of certain kinds of Internet traffic, e.g, TCP [6], TCP and UDP [7], *dns* [8].

The first half this dissertation combines the comprehensiveness of the first area, LAN characterization, with the infrastructural aspect of the second cloud, operational WAN traffic characterization. In the latter half we explore how studies in the third cloud, comprehensive WAN studies at a single point, can influence the kind of statistics collected operationally on wide area network infrastructures. The measurements we take in the process demonstrate several surprising aspects of Internet traffic which future designers of protocols or network architectures, in addition to designers of statistics collection architectures, should recognize. Tables 6.2 and 7.4 list the important findings of our measurements, including: (i) the brevity of a significant fraction of IP flows (ii) that the number of host-pair IP flows is not significantly larger than destination network or network pair flows, and (iii) that schemes for caching traffic information could benefit by making caching decisions taking into account higher layer information. These observations have implications for performance requirements of routers at Internet hotspots, general and specialized flow-based routing algorithms, future usage-based accounting requirements, and traffic prioritization.

The final component of this thesis synthesizes insights we have obtained from our investigations of operational as well as specific experimentally collected statistics to suggest why it will be more important to support flow assessment via operational statistics collection. Ongoing flow assessment can provide a valuable knowledge base for networking activities along various time horizons, from defining service quality patterns to long-term capacity planning. We offer a unique combination of operational and research perspectives, allowing us to reduce the gaps among (1) what network service providers need; (2) what statistics service providers can provide; and (3) what network analysis requires.

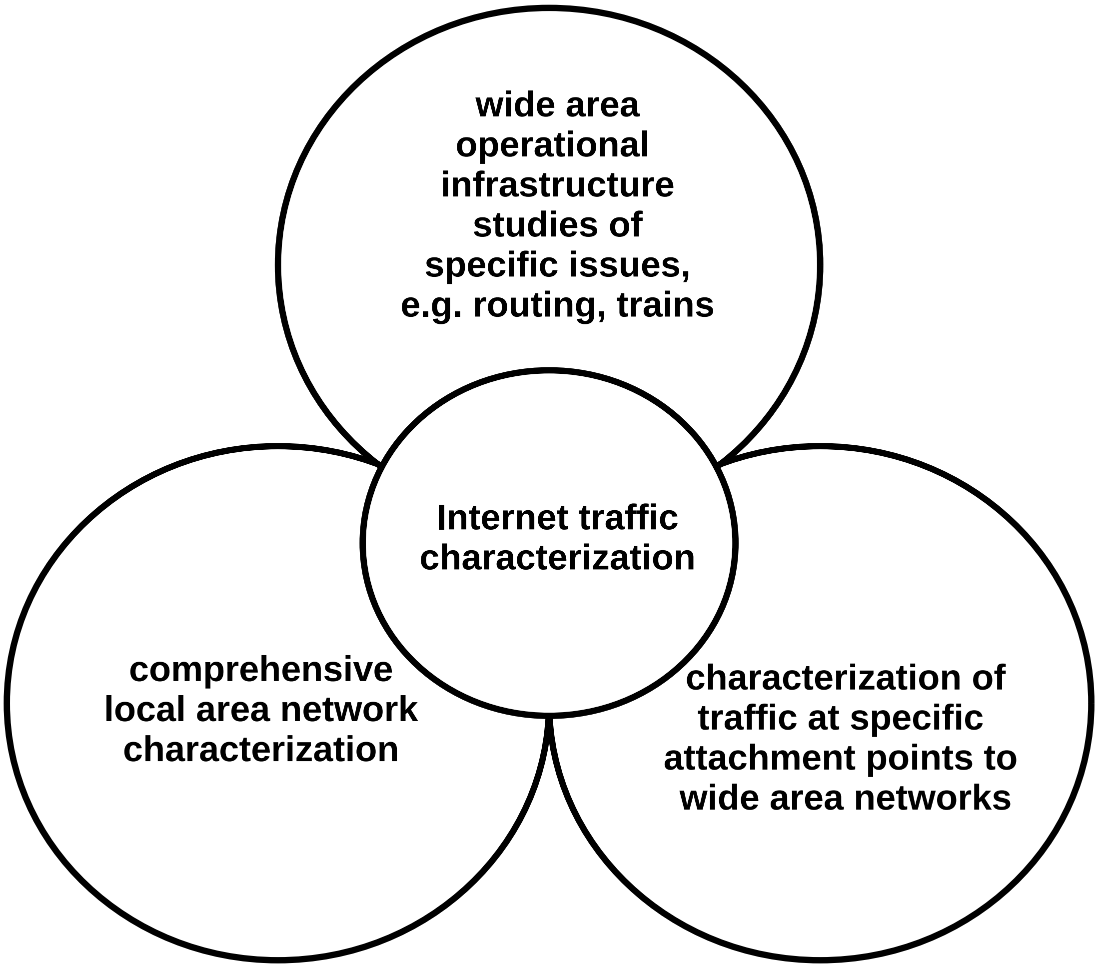

Figure 1.1: relation of our work to existing literature

Chapter 2

Taxonomy of traffic characteristics

I can calculate the motions of heavenly bodies, but not the madness of people.

– *Isaac Newton*

When the Lord created the world and people to live in it – an enterprise which, according to modern science, took a very long time – I could well imagine that He reasoned with Himself as follows: 'If I make everything predictable, there human beings, whom I have endowed with pretty good brains, will undoubtedly learn to predict everything, and they will thereupon have no motive to do anything at all, because they will recognize that the future is totally determined and cannot be influenced by any human action. On the other hand, if I make everything unpredictable, they will gradually discover that there is no rational basis for any decision whatsoever and, as in the first case, they will thereupon have no motive to do anything at all. Neither scheme would make sense. I must therefore create a mixture of the two. Let some things be predictable and let others be unpredictable. They will then, amongst many other things, have the very important task of finding out which is which.'

– *E.F. Schumacher,* Small Is Beautiful

The *multi-* prefix exhibits increasing popularity in Internet development: multi-platform, multi-protocol, multicast, multi-layer, multimedia. Less attention has gone into how to describe and modulate the effect of the increasing diversity of services, and their respective performance requirements, on aggregate Internet workload. Simple packet and byte counters are no longer sufficient to understand the magnitudes and trends of traffic flows, and their impact on performance and forecasting. The complexity of the system necessitates that we characterize workload as a function of multiple dimensions, in addition to understanding underlying network mechanisms. In this chapter we first discuss one aspect of statistics collection that constrains subsequent traffic characterization efforts: the aggregation granularity. We then offer a taxonomy of performance and traffic characterization tasks on datagram wide area networks. We discuss relevant research efforts in each area, positioning our work in the context of these categories.

2.1 Aggregation granularity

One aspect of collected statistics that constrains subsequent traffic characterization and modeling efforts is their aggregation granularity. Aggregation involves two stages: the granularity at which one collects information, and the granularity with which one presents information. For example, a packet count for a fifteen minute interval implies a selected collection granularity. In contrast, the "bucket size" of

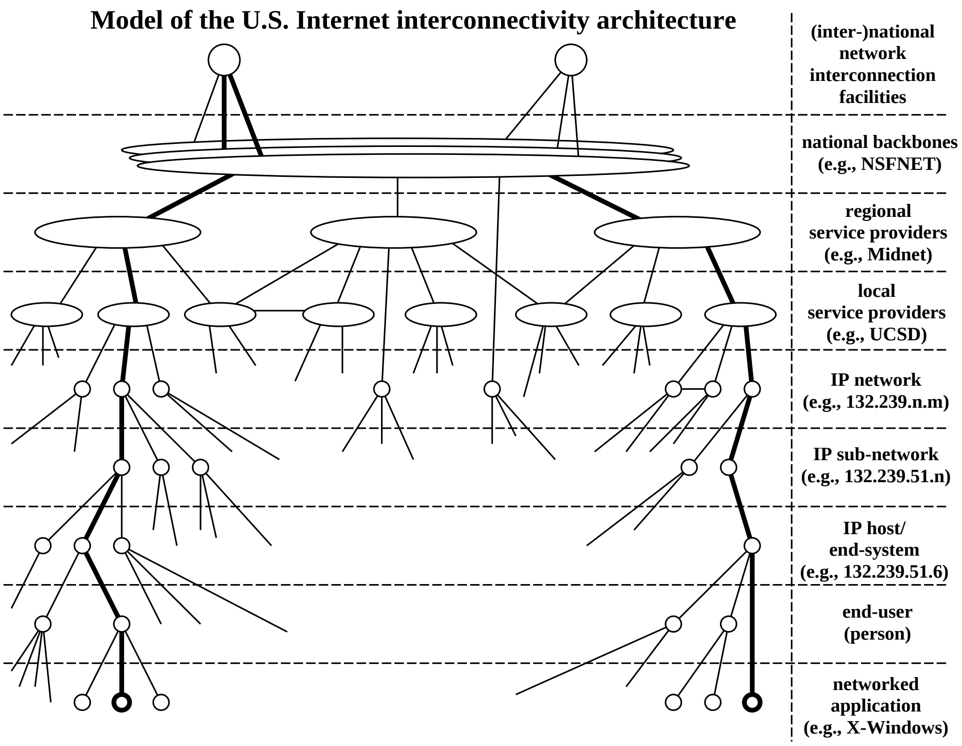

Figure 2.1: model of U.S. Internet interconnectivity architecture

an interarrival time histogram constrains the granularity of presentation. In both cases, selection of the appropriate granularity for aggregation depends on the question one addresses, and sometimes in turn defines the range of questions one can address.

In addition to these two stages of aggregation, we note two dimensions along which one aggregates data into a certain granularity: time and space. An example of time granularity is the current 15-minute aggregation interval for most statistics collection on the T3 backbone. Such a coarse granularity may be appropriate to answer questions about high-level distribution of network usage on a daily basis. Other questions, such as analyzing the dynamics of packet arrivals, or prediction of quality of service requirements for continuous media data flows, will require a much finer time granularity.

An example of granularity along the space dimension is the geographic focus of a particular measurement: one might want to explore a specific node or link to examine behavior such as favoritism or hotspots. Alternatively, when presenting internetwork traffic flows, one might want to develop a model of traffic flows according to policy requirements, such as flows among countries. Other granularities include traffic by: multibackbone environment (e.g., of different agencies), single backbone, backbone node, external interface of a backbone node, backbone client service provider, Administrative Domain[1], IP network number, host, end user, and application. These granularities do not have an inherent order, as a single user or application might straddle several hosts or even several network numbers. As with the time dimension, the appropriate granularity depends on the question of interest. Figure 2.1 presents some of the possible layers of interest to the Internet community.

[1] An Administrative Domain is a collection of end systems, intermediate systems, and subnetworks operated by a single organization or administrative authority [9]. We use the Autonomous System number to identify a particular Administrative Domain.

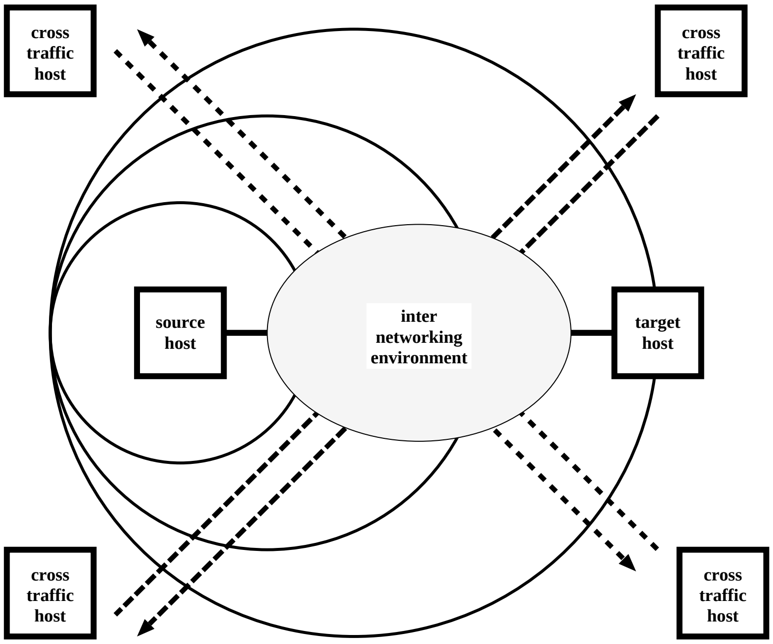

Figure 2.2: host-centric perspective of network analysis

2.2 Host versus network centric perspective

We divide traffic workload metrics into two categories: end-to-end based metrics, which assume a perspective centered around the end host, and aggregate workload metrics, which assume the perspective of a given transit network aggregating flows among many clients. Figures 2.2 and 2.3 present schematics of the host and network-centric perspective, respectively. In host-centric studies, a local host machine constitutes the center of the analysis environment; a second host remote from the local machine allows one to measure end-to-end performance under various conditions. These evaluations focus on the performance limits of the network without regard to its internal details, e.g., structure, implementation, other traffic. End systems may perceive the effects of some traffic aggregation in the network in such scenarios, but quantifying the effect of the aggregation is beyond the scope of most host-to-host performance evaluations.

In contrast to a host-centric perspective, a network-centric performance study places the network, often one that aggregates traffic from many sources, at the center of investigation. While the distinction between the two perspectives is irrelevant for small environments, such as local networks with few hosts, it becomes critical as one studies wider area networks which aggregate traffic from a large number of users and service categories. Host-centric studies, lacking data about aggregate traffic, often focus on end-to-end performance metrics, while network-centric studies can offer metrics regarding characteristics of the larger workload. Combining host and network-centric performance studies is also useful, for example to ascertain the effect of introducing traffic impulses from specific end systems on the aggregate workload of a wide area network.

2.3 Host centric perspective

For completeness we first mention performance metrics which assume an end host perspective, and then discuss network-centric metrics of traffic characterization. Because this dissertation focuses more on the latter category of metrics we will discuss these in more depth.

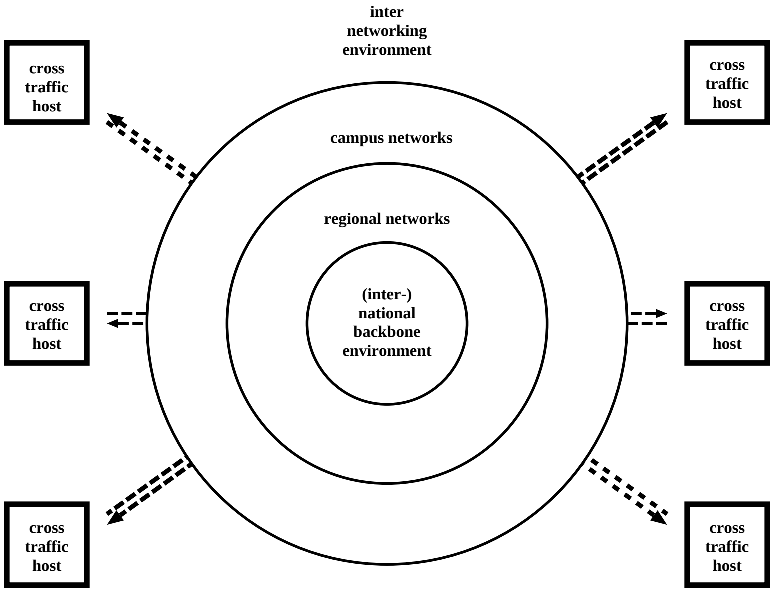

Figure 2.3: network-centric perspective of network analysis

2.3.1 Delay and jitter

Delay and jitter are typically end-to-end *performance* notions. Delay includes transfer delay, caused by the transmission from source to destination, and queueing and processing delay, caused by the intermediate switching nodes and end hosts. Many real-time multimedia applications may require predictable delay, notably inconsistent with the datagram best-effort architecture of the Internet. In addition, many continuous media applications will rely on synchronization between audio and video streams. Thus the variance in delay will also be an important Internet performance metric. Studies have provided evidence for the variance and asymmetry of delay on wide area Internet infrastructures [10] [11]. Floyd and Jacobson [12] [13] analyze traffic phase effects in packet-switched gateways and end systems, their potential damaging effects on Internet performance, and suggestions for possible ways to mitigate the systematic tendency of routing protocols to synchronize in large systems. Zhang *et al.* [14] studies the phase effects of congestion control algorithms or other aspects of TCP implementations [15].

2.3.2 Loss

Future multimedia applications may require low or predictable delay but not necessarily completely lossless services, while other applications require transmission guarantees but no strict delay bounds. Targeting a specified loss distribution is thus a likely future performance goal. In addition to the effect of loss on protocol performance and network dynamics [16] [17] [18] [10] studies have also investigated the potential impact of loss on charging policies [19] [20]. Related to loss metrics are those of reliability and availability of given links or nodes in the network.

2.3.3 Throughput

Throughput metrics include (from Jain [21]):

- nominal capacity or bandwidth: achievable throughput under ideal workload conditions
- throughput under actual workload conditions
- knee capacity, or point on the delay versus throughput curve where delay increases significantly
- efficiency: ratio of maximum achievable throughput (usable capacity) to nominal capacity

Studies of network throughput often focus on design or performance assessment of proposed or existing protocol mechanisms or algorithms [22].

2.4 Network centric perspective

We next discuss network-centric metrics.

2.4.1 Utilization

The first metric that typically comes to mind in describing a network, at least for a network operator, is utilization. Utilization metrics can reflect any measured granularity, and statistics of their distribution, including mean, variance, and percentile statistics, can reveal trends over both short and long time intervals. Related to utilization is the congestion of the network, or contention for either bandwidth or switching resources. Measurements of congestion include distributions of queue length or available buffers in nodes. Several studies of local area environments focus on short-term utilization characteristics [2] [23] [3]. Longer term utilization metrics would include traffic volume growth over several years on a backbone infrastructure [24]. An Internet service provider will tend to pay attention to utilization metrics as indicators of how close their network is to saturation so they can plan for upgrades.

2.4.2 Reachability

As networks increase their range of possible destinations, so does the size of routing tables, and thus the cost of maintaining them in switching nodes and the cost of searching them in forwarding datagrams. Metrics such as the size of these routing tables, or the number of IP network numbers to which an Internet component can route traffic, are indicators of network reachability.

2.4.3 Locality

Related to network reachability are metrics of traffic locality, which reflect the geographic non-uniformity of traffic distribution. Because we present metrics in both long and short term locality in sections 3.5.3 and 7.3, respectively, we provide more background and discussion of locality studies here.

Designers of computer systems have years ago incorporated the notion of memory reference locality in system design, largely through the use of virtual memory and memory caches [25]. In deriving metrics for network traffic locality, Jain [26] draws a comparison to memory reference locality, which is either *spatial* or *temporal.* Spatial locality refers to the likelihood of reference to memory locations near previously referenced locations. Temporal locality refers to the likelihood of future references to the same location. In network traffic, the *concentration* of references to a small fraction of addresses and the *persistence* of references to recently used addresses are analogous concepts [26]. Jain presents data using three measures of locality: the network traffic "income" distribution, which can reflect long or short-term locality; and two metrics that only apply to short-term locality assessment: the average working set size as a function of packet window size, and the stack depth probability distribution. The income distribution measures what percent of communicating network entities is responsible for what percent of traffic on the network. Changes in the *working set* of source or destination IP networks are indicators of source-based and destination-based favoristism. To measure the working set one plots the number of unique address references as a function of the number of total address references. The stack level probability distribution measures the likelihood of reference to a network address as a function of the previous reference to that address.

Gulati *et al.* [27] offers four locality metrics, two of which overlap with those of Jain: persistence; address reuse, which is similar to persistence with the requirement for consecutive reference loosened; concentration; and reference density. Reference density reflects the number of communicating entities responsible for a given percentile of the network traffic.

Many previous studies have established the existence of network traffic locality, in particular short-term traffic locality in specific network environments for selected granularities of network traffic flow. Jain originally established the packet train model to study traffic locality behavior on a local area network [28]. Other studies, though not focused on packet trains in particular, also find evidence for locality even in networks of wider geographic scope, e.g., regional networks and national backbones [29] [5] [30] [31] [32] [24] [7]. Others [33] [34] [35] have extended the packet train model to the transport and application layers, defining a train as a quadruple of source/destination address pairs in conjunction with port numbers.

Using a transport-layer definition of a flow rather than a packet train definition, Schmidt and Campbell [7] have explored locality of IP traffic flows for ATM design. Using both LAN and WAN traffic traces they find that for a 24-hour packet trace on the CSOnet (a campus network at UIUC), almost 40% of TCP packets are addressed to 4 hosts on the CSOnet. On a particular 24-hour trace of wide-area traffic, specifically on the Urbana-Champaign NSFNET node, they find that 15% of the traffic is sent to 5 hosts.

Just as program and data caching policies can exploit memory reference locality in a virtual memory system, router designers can exploit traffic locality with analogous schemes such as caching network address and specialized flow information in switching nodes. Feldmeier [36] and Jain [26] simulate caching algorithms on traffic traces taken from LAN gateways. Feldmeier [36] estimated the potential benefit of caching on the performance of gateway routing tables. Using measured traffic from gateways at MIT, he simulated a variety of fully associative caching replacement algorithms (LRU, FIFO, and random) to determine cache performance metrics such as hit ratio and interfault distance. His data indicated that the probability of reference to a destination address versus time of previous reference to that address monotonically decreases for up to 50 previous references, implying that an LRU cache management procedure is optimal for caches of 50 slots or less. His conservative conclusion was that caching could reduce gateway routing table lookup time by up to 65%. In addition to caching destination addresses, his simulations indicate benefits from caching source addresses as well.

Jain [26] also performed trace-driven cache simulations for traffic at DEC gateways. Simulating MIN (optimal), LRU, FIFO, and random replacement algorithms, he found significantly different locality behavior between interactive and non-interactive traffic. The interactive traffic did not follow the LRU stack model while the noninteractive traffic did. In particular, the periodic nature of certain protocols may make caches ineffective unless they are sufficiently large. Such environments may require larger or multiple caches, or new cache replacement/fetch algorithms.

Estrin and Mitzel [31] also explore locality in their investigation of lookup overhead in routers. They use data collected at the border routers of stub and transit networks to estimate the number of active conversations at a router, which reflect the storage requirements for the associated conversation state table. They find that maintaining fine grained traffic state may be possible at the network periphery, but deeper within the network coarser granularity may be necessary. They also use the traces to perform simulations of an LRU cache for different conversation granularities, and find that improvements in state lookup time are possible with a small cache, even without special hardware.

Gulati *et al.* [27] have explored LAN cache performance of source addresses, destination addresses, and both source and destination addresses. In their measurement study of LAN traffic they find that it is more important to cache destination rather than source addresses, especially for caches with more than 15 entries. One reason is that many source hosts send very few packets, and thus the cost of caching the source address is greater than the benefit. Another reason is that source addresses are poor predictors of destination address references in the future.

A simple next-hop routing cache is not the only possible exploitation of address locality in network infrastructure. Proposed protocols for interdomain routing [37] [38], soft-state or adaptive Internet routing strategies [39] [40], policy routing [41] [42] [43] or flow-based support for specific application requirements [44] [45] [46] [47] require detailed maintenance of flow state in intermediate switching nodes. Several new proposals involve queueing algorithms to control congestion [22] [48] [49] [50] [51] [52] [53], and thus assume integral caching mechanisms for the maintenance of state for each communicating host-pair flow. The potential efficacy of such proposals depends critically on network locality. Other studies have pointed to the need for a mechanism for managing such state information [54].

2.4.4 Burstiness

Related to the overall distributions of delay and jitter are metrics of burstiness of the overall traffic workload from the network perspective. Burstiness metrics fall into two categories: those that measure interarrival processes, e.g., time between packet arrivals, and those that measure arrival processes, e.g., number of packets per time interval. An additional parameter is what is arriving, e.g., packets, bytes, flows,

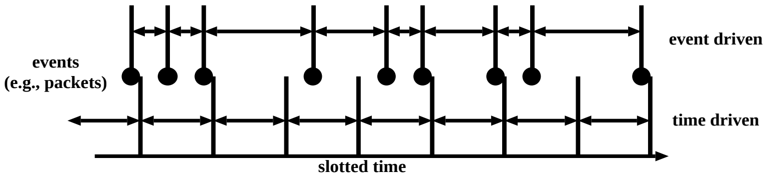

Figure 2.4: depiction of an event-driven process, which measures interarrival times between events, versus a time-driven process, which measures the number of arrivals during a given time period

transport-level sessions. Figure 2.4 depicts both metrics; the arrows in the top row reflect *packet interarrival times*, while the dots in the middle represent the *number of arrivals* in a given time period. One well-known relationship between the two descriptions is that a Poisson arrival process will generate exponentially distributed interarrival times, and vice-versa.

Traditionally, network researchers have used interarrival time distributions to indicate burstiness. Gusella [3] presents and interprets the distribution of packet lengths and packet interarrival times on a dedicated Ethernet for the three protocols that carry significant traffic: TCP, ND (Network Disk protocol, for paging traffic), and NFS (Network File System, for remote file access). On the network that he investigates the two latter protocols account for 68% of the packets and 94% of the bytes on the network. He also explored the burstiness of these protocols, including NFS bursts which can consume about 10% of Ethernet bandwidth sustained for several seconds. He concludes with a warning that the combination of bandwidth-intensive applications with high-performance virtual memory machines will require fast interactive communication protocols that function effectively under high network load.

More recent investigations have proposed other notions of burstiness, such as focusing on the arrival process, as a counting process, rather than the interarrival process. Willinger *et al.* [55] [23] have studied the packet arrival process, and the correlation of packet arrivals in local environments. In their study of several Ethernet environments [23] they present evidence that Ethernet traffic is *self-similar*, implying no natural burst length. Such traffic exhibits the same correlation structure at various aggregation granularities. They conclude that empirical data demands reexamination of currently considered formal models for packet traffic, e.g., pure Poisson or Poisson-related models such as Poisson-batch or Markov-Modulated Poisson processes [56], packet-train models [28], and fluid flow models [57]. In particular, their evidence indicates that Poisson modeling assumptions are false for environments that aggregate much traffic. Contrary to the Poisson assumption, the traffic profile of their measured environments becomes burstier rather than smoother as the number of active sources increases; the Poisson assumption appears to hold only during low traffic periods with mostly machine generated router-to-router traffic.

While these studies focus on LAN, specifically Ethernet, traffic, Paxson and Floyd [58] also comment on the potential self-similarity of wide-area traffic. They also note that packet interarrivals are not exponentially distributed [28] [3] [32] [13]. They evaluate several TCP arrival process in fifteen wide area packet traces to determine the error of modeling them as Poisson. We discuss their work further in section 7.2. Their traces indicate that TCP traffic is not as self-similar as when it is multiplexed with Mbone (UDP audio) and DECnet traffic, suggesting that self-similarity may result from how aggregated traffic sources affect one another.

2.4.5 Payload

Payload is the amount of information carried in a packet. What constitutes *information* will depend on the layer, e.g., the payload of an IP packet would be the contents of the packet following the IP header, while the payload of a *telnet* packet would exclude the TCP and IP headers. A loose definition of payload sometimes includes the entire packet including headers. Packet payload is one indicator of protocol efficiency, although a more accurate analysis of efficiency will also reflect end-to-end behavior, including acknowledgment, retransmission, and update strategies of different protocols.

The differences in payload per application are also visible at the aggregate level. Claffy *et al.* [24] [59] present evidence for significantly different distributions of traffic by packets and bytes by individual network numbers using the NSFNET. The disparity between the number of packets and the number of bytes sent by networks indicates a definite difference in workload profiles, where specific networks, likely with major data repositories, source mostly large packet sizes into the backbone.

2.4.6 Traffic cross-section

Many workload characteristics will exhibit significantly different values when applied to the overall traffic as opposed to specific applications. Applications and their underlying transport protocols likely differ in their profile, e.g., TCP differs from UDP, *ftp* differs from *telnet*, so in addition to the total makeup of the traffic, per-protocol profiles (discussed in section 2.4.7) are important network descriptors.

2.4.7 Individual flow metrics

Studies have shown that the time of day and type of network influence the cross-section of wide area traffic [6] [30] [58], making it important to measure not only the cross-section itself, but also how individual components of the cross-section contribute to the aggregate traffic workload. Examining the behavior of specific applications (such as interactive file transfers or remote windows) and their underlying lower layer transmission protocols (e.g., TCP, UDP, IP) is a part of the task of *flow profiling.* A loose definition of a flow is a sequence of associated packets, such as from a single application instance. We define flows more precisely in Chapter 5 A profile of a specific application involves many of the metrics we have previously discussed, e.g., the interarrival time of packets or flows of a certain application, the number of arrivals per time interval, packet size distributions, etc. Profiling also includes metrics of individual flows within that application, such as the number of packets per flow, the number of bytes per flow, and flow duration.

Several profiling studies of TCP traffic have investigated Internet traffic passing through attachment points to wide area transit networks. Caceres *et al.* [6] profile Internet conversations by characterizing individual end-to-end TCP communication transactions. They define a flow as a stream of packets traveling between the end points of an association, delimited by a twenty-minute silence. Their measurements explode several myths of wide-area network traffic, including: traditional "bulk transfer" applications such as *ftp* transfer surprisingly small amounts of traffic (less than 10 kilobytes per conversation); bulk traffic is strongly bidirectional, with bimodal distribution of packet sizes; interactive applications often generate ten times more data in one direction than the other; and interactive packet interarrival times closely match a uniform plus exponential distribution.

Danzig and Jamin [32] use results of the work of Caceres *et al.* [6] to create a library of realistic empirically-based TCP workload characteristics for use in network simulation [32] [60] They create a workload library (*tcplib*) consisting of a *stub-dependent* component, since the breakdown of individual TCP applications differs by stub network, and a *stub-independent* component, since their data indicates that some characteristics of a given instantiation of a given TCP application are independent of the network on which it occurs.

Their empirically-based library does not model, and they identify as crucial future work, several metrics discussed in section 2.4.4, since they are site-dependent and thus "require detailed study of many Internet stub networks" [32]:

1. the modeling of conversation arrival processes (particularly for protocols such as *nntp* which exhibit distribution characteristics that are periodic but not necessarily synchronous across network components)

2. the interarrival time of *ftpcontrol* packets

3. the distribution of number of request response handshakes that occur during *smtp* and *nntp* conversations

4. distribution of conversation durations and idle periods per protocol

5. request and response side probability of sending data after a control message

6. transfer size distribution for connection and handshakes for periodic protocols (*x11*, *snmp*) which [re]establish a connection periodically to update, handshake, or transmit

Another limitation of the library is that it provides parameters for only five applications, all using TCP. Other applications, many likely with quite different characteristics, e.g., multimedia have begun to consume significant bandwidth. Nonetheless their library offers an important basis for comparison with data collected in other environments at other times.

Paxson [61] [62] [30] investigates wide-area TCP connections which originate from three levels of the Internet, spanning university campuses, corporate research sites, a regional site, and an international gateway environment. For one of the sites he uses several traces that span over a month, providing a picture of traffic behavior over time. He assesses the geographical distribution and growth of wide-area traffic from this particular site. Paxson then uses all the collected data sets to derive analytic models describing random variables associated with *telnet*, *rlogin*, *nntp*, *smtp*, and *ftp* connections. The random variables include: bytes sent in each direction of a given type of connection, and the ratio between the two; interarrival time distributions per protocol; and connection duration (for *telnet*). He then presents a methodology for comparing the effectiveness of the analytic models with empirical models such as *tcplib* [32]. He concludes that the analytic models provide good descriptions, generally modeling the various distributions as well as empirical models and sometimes better.

Other traffic studies have focused on local or wide area issues relating to a specific protocol. Danzig *et al.* [8] focuses on the profile of a single application in an analysis of wide-area *domain name system* traffic. They explore the performance of the *dns* protocol based on two 24-hour traces of *dns* traffic destined to a major root name server. They find that suboptimalities in implementations of distributed applications such as *dns* impose a serious impact on network load, and suggest improvements in the future deployment of name servers.

Traffic differs not only across different application protocols such as *telnet*, *smtp*, or *ftp*, but also among the underlying transport protocol, such as TCP or UDP. Some considerations may warrant characterizing traffic at the network layer, ignoring the characteristics of the higher layer application, for example to compare datagram oriented UDP traffic to that of TCP traffic. Acharya *et al.* [33] [34] provide data on different hierarchical components of traffic for a campus network at the University of Florida. Such measurements will become more important with the increasing popularity of alternative transport protocols supporting applications of much higher payload and duration than those of traditional TCP applications. Such uses of the network, such as with packet audio and video, are incompatible with currently deployed congestion control methodologies that assume cooperation from the virtual circuit oriented connection end points of TCP.

For example, Caceres *et al.* [6] investigate policies for caching virtual circuits in networks without permanent virtual circuits (PVC's). Because they did not have a conversation interarrival model as discussed above, they focused on intra-conversation activity, and did not address how long dynamic virtual circuits should remain open once the conversations using them have ceased. Still an open question is how to best tradeoff between the wasted resources of idle circuits and the delays incurred by set up and teardown of frequently used circuits. In Chapters 5, 6 and 7 we present a methodology that allows an investigation of how various parameters of a flow will affect such setup and teardown requirements.

2.4.8 Aggregate flow metrics

Since integrated networks will require some connection-oriented flow information maintained throughout the lifetime of some flows, several metrics of aggregate flow activity will be important for developers of network equipment: the number of active flows per second, the number of new flows requiring setup each second (related to the metrics of arrival processes described in section 2.4.4), the number of idle flows requiring deletion.

2.4.9 Environment-dependent characteristics

In addition to the hierarchical structure of protocols, an additional hierarchical component of traffic characterization is the architectural hierarchy of Internet connectivity. Workload profiles, and even individual protocol profiles, may differ across different network components which fit roughly into the hierarchical architecture depicted in figure 2.1. For example, one would expect that distributed file and window systems are responsible for a a larger proportion of network traffic in local environments than across wide area infrastructures. Within a given application, one might also expect different profiles from campus versus wide area traffic. Workload profiles of different network constituencies at the same level of the hierarchy will also differ, i.e., a supercomputer center workload profile will incorporate the impact of specific applications that the supercomputing centers introduce, differing considerably from those of a major university campus.

Acharya *et al.* [33] have hierarchically characterized traffic from a single campus network by analyzing traffic traces from three campus locations that capture intra-LAN, inter-LAN, and LAN-to-WAN traffic respectively. The authors compare the distribution of packets by application, transport, and network protocols at these three points.[2] Their data reveals that different layers exhibit workload profiles that are different enough to warrant attention when choosing input parameters or building traffic generators for simulation studies on network and protocol performance.

The Internet hierarchy is not limited to domestic U.S. infrastructure. Wakeman [63] and Asaba *et al.* [64] have published studies of traffic characterization across international infrastructure, specifically trans-Atlantic and trans-pacific traffic, respectively. These efforts are prerequisite, but only preliminary, to more in-depth study of geographic flow characterization.

Similarities and differences among the various traffic profiles will allow a better adaptation of network policies and algorithms to specific environments, in particular as the Internet environment grows at different rates in different places, with bandwidth gaps of up to six orders of magnitude, from 1200 bits per second for some slow speed dialup lines, to a gigabit per second and beyond. The range of reachability is also formidable; some backbones aggregate traffic for more than a million source/destination network number pairs, while other environments support very few hosts with little aggregation. It is unlikely that the same blueprint of networking equipment can address the disparities among such environments, particularly as environments of smaller scale will likely require very inexpensive switching equipment. Understanding hierarchical workload profiles will allow better judgments for network architecture, design and bandwidth management, and also be relevant to future network accounting and billing methodologies. The data sets we use in this dissertation span several levels of the loose Internet hierarchy, strategically selected to highlight the diverse requirements of different components of the infrastructure.

[2] Although we emphasize the hierarchy in the Internet architecture, the authors of this study actually call their study hierarchical due to their approach in classifying traffic according to the various protocol layers.

Chapter 3

Traffic Characterization of National Backbones

Quality is never an accident. It is always the result of high intention, genius efficiency, intellectual direction, and skillful execution.

– *Alvin Toffler*

Five senses; an incurably abstract intellect; a haphazardly selective memory; a set of preconceptions and assumptions so numerous that I can never examine more than a minority of them – never become even conscious of them all. How much of total reality can such an apparatus let through?

– *C. S. Lewis*

In this chapter we investigate existing operational statistics and the extent of their usefulness for Internet workload characterization. We first present related work in national backbone traffic characterization. We then focus on the NSFNET environment, beginning with a description of the previous T1 and current T3 NSFNET backbone architecture. We describe the mechanisms for data collection on the T3 backbone, and how they differ from those used on the T1 backbone. We then use operationally collected statistics for selected months, May 1992 for the T1 backbone, and December 1993 for the T3 backbone, to illustrate their usefulness for the workload characterization tasks outlined in Chapter 2.

3.1 Related Work

The first study on measured network behavior of an operational wide-area IP backbone, comparable to today's national research and education backbone infrastructure, was the study by Kleinrock and Naylor [65] (also in [66]), which presented measurements on the 1973 ARPAnet, almost 20 years ago. Since that study, the infrastructure has increased substantially in scope, bandwidth, and traffic aggregation, largely due to the advent of the NSFNET in the mid 1980's, which was able to extend the seeds of ARPAnet research into what is now a global TCP/IP infrastructure. Chinoy and Braun [67] describe in detail the underlying topology of the NSFNET backbone in its various stages.

More recent studies on isolated aspects of the NSFNET have investigated the existence of packet trains on the NSFNET backbone [5], and evaluated specific routing approaches for use on the backbone [4]. Heimlich [5] experimented with an extension of Jain's packet train model [28] to a wide-area environment, verifying the existence of packet trains on the 1989 NSFNET national backbone. While one can not expect the packet train phenomenon to be as striking in a traffic-aggregating national backbone network as it is on the local network that Jain studied [28], Heimlich found that it was still "quite strong given the great number of hosts communicating through the backbone." We discuss more recent studies that are closely related to our work in sections 3.4.1 and 3.5.1.

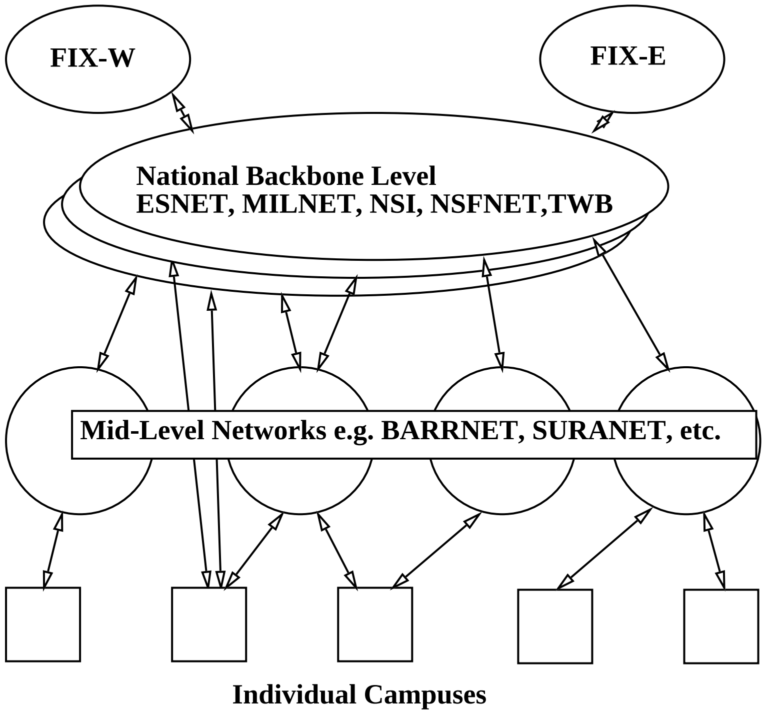

Figure 3.1: hierarchical model of NSFNET architecture

3.2 NSFNET backbone environment

The Internet is a global network infrastructure spanning many countries. The United States component of the Internet currently consists of a three-level hierarchy of national agency backbones, attached mid-level networks, and connected local sites. Figure 3.1 illustrates this hierarchical structure. ESnet, Milnet, NSI, NSFNET, and TWB correspond to national backbones of the Department of Energy (DOE), Department of Defense (DoD), National Air and Space Administration (NASA), National Science Foundation (NSF), and Advanced Projects Research Agency (ARPA), respectively. Appendix A provides more details of key components of the Internet environment.

NSFNET, the National Science Foundation Network, is a general purpose packet-switching network supporting access to scientific computing resources, data, and interpersonal electronic communications. Evolved from a 56kbps six-node network in the mid-1980s to today's 45Mbps network, the current NSFNET includes three different levels: the transcontinental backbone connecting the NSF-funded supercomputer centers and mid-level networks, the mid-level networks themselves, and the campus networks. The hierarchical structure includes a large fraction of the research and educational community, and even extends into a global arena via international connections. Figure 3.2 shows the logical topology of the backbone.

Since July 1988, Merit Network, Inc. has administered and managed the T1 NSFNET backbone, and in late 1990, in conjunction with partners IBM and MCI, began to deploy in parallel a replacement T3 network.[1] The T3 network provided a 28-fold increase in raw capacity over the T1 network (from 1.544 Mb/sec to 44.736 Mb/sec), and by November 1992 had completely replaced the T1 network.

In the interim, the status of the NSFNET shifted through organizational restructuring among original participants in the backbone project. In 1991, Advanced Network Services (ANS) began official operation and management of the national T3 backbone described above. Merit Network, Inc. still holds a cooperative agreement with NSF to provide NSFNET backbone services, although Merit no longer provides these services via a dedicated infrastructure. Merit now subcontracts these services to ANS, who provides them over ANSnet, their own backbone infrastructure. The "NSFNET backbone" now refers to a virtual

[1] The original cooperative agreement between Merit and NSF in 1987 allowed for this optional upgrade to T3 speeds, which a later follow-on proposal to the original agreement more clearly specified.

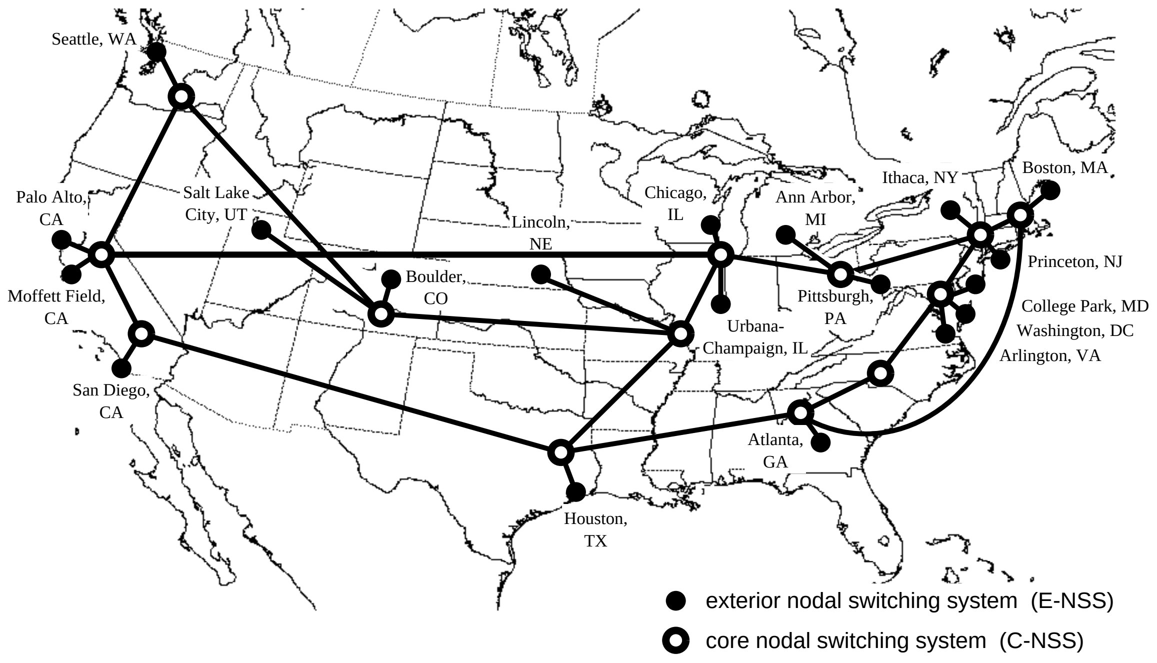

Figure 3.2: 1992 NSFNET T3 backbone service logical topology

backbone network, i.e., a set of services provided across the ANSnet physical backbone. Hereafter we refer to the "T3 NSFNET backbone" with the understanding that we are referring to a service provided to NSF, not a dedicated NSFNET infrastructure.

3.2.1 Architecture of the T3 Backbone

We review a few of the network parameters that affect traffic flow in the current T3 backbone. Chinoy and Smith [68] present details of the T3 network architecture, which evolved from the experience of managing the T1 network. Backbone nodes, the core packet switches in the T3 infrastructure, are designated as either Exterior Nodal Switching Subsystems (ENSSs) or Core Nodal Switching Subsystems (CNSSs). ENSSs are located on the regional network premises and CNSSs are co-located at carrier switching centers which are also known as "points-of-presence" (POPs) or "junction points." Co-location of the core packet switches within POPs provides several advantages. First, since these locations are major carrier circuit switching centers they are staffed around the clock, and have full backup power which is essential to the stability of the network. Second, this co-location allows the addition of new clients (e.g. ENSS nodes) to the network by connecting them to a CNSS without service disruption to other CNSS/ENSS clients. Colocation also allows network designers to more closely align the carrier-provided circuit-switched network topology with the packet-switched backbone topology, enabling a very robust core network that retains internal stability despite outages at external (ENSS) sites.

The serial line interfaces to each node on the T3 backbone are of "T3", or DS3 speed, 44.736 Mbits/second. There are T1 backup links as well; we will not focus on these backup links in our discussion of the architecture. The DS3 circuits are not subchanneled, so the full 45 Mbits/second, less framing and carrier management overhead, is available for user traffic. The physical and electrical interfacing to these lines is handled by a Data Service Unit (DSU). Nearly all of the DS3 circuits are terrestrial fiber-optic lines; other possible media are microwave and copper [68]. To access external client networks, the T3 backbone nodes currently use Ethernet and FDDI interfaces, with packet size upper limits of 1.5 and 4 kilobytes, respectively. Each packet is also encapsulated within an Ethernet or FDDI frame, which the LAN drivers at

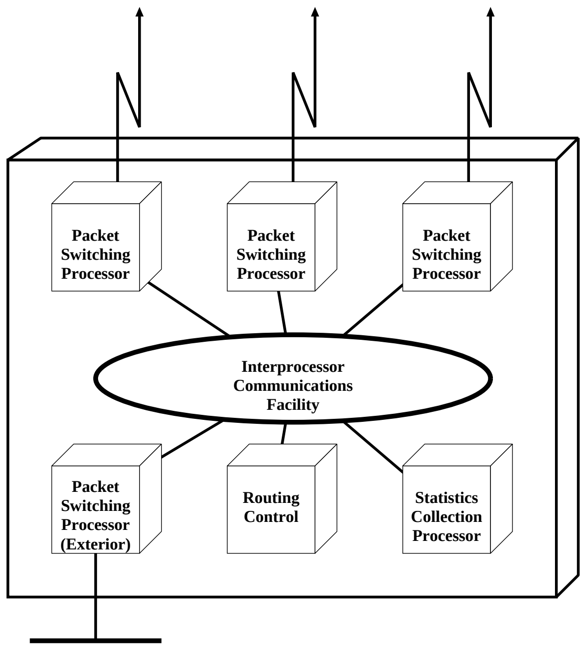

Figure 3.3: T1 NSFNET nodal switching system (NSS) architecture

the endpoints append and remove.

The T3 routing technology and architecture is functionally equivalent to that on the T1 network. Packets travel through the network individually and are passed from node to node aided by an adaptive, distributed routing procedure based on the standard IS-IS protocol [69]. Buffering on the output queues of the nodes contributes to the latency of the delivery of packets to the destination. On the T3 backbone, the number and size of buffers in each node depends on the interface type and operating system version.

Figures 3.3 and 3.4 illustrate the Nodal Switching Subsystem (NSS) architectures for the core backbone nodes on the T1 and T3 backbones, respectively. The T1 NSS architecture consisted of multiple, typically nine, IBM PC/RT processors connected by a common token ring. In contrast, the T3 backbone packet forwarding routers are based on the IBM RISC System/6000[2] architecture, with special modifications including high performance adapter cards and software. Initially, the interfaces to this uniprocessor architecture switched packets through to the outgoing interfaces via the main CPU. In the current implementation, the packet forwarding process is offloaded onto intelligent subsystems. Each external interface, including T3 serial lines, as well as connected Ethernet and FDDI LANs, lies on such a dedicated subsystem card. These cards have a built-in 32-bit Intel 960 microcontroller on board, and have local access to all information needed to switch a packet, including routing tables and relevant code. The cards can thus exchange packets among each other directly via the IBM Microchannel[3] bus, without the intervention of the main processor.

3.2.2 NSFNET statistics collection instrumentation

The principal sources of information about the T3 backbone come from routine collection of three classes of network statistics by NSFNET backbone packet-switching nodes: interface statistics; packet categorization; and internodal delays. Interface statistics derive from programs using the Simple Network Management Protocol (SNMP) [70]. Specialized software packages perform packet categorization: the T1 backbone utilized the NNStat [71] package for collection; the T3 backbone utilizes the ARTS (ANSnet Router Traffic Statistics) [72] package, which encompasses similar functionality.

[2] RISC System/6000 is a trademark of IBM Corporation.
[3] Microchannel is a trademark of IBM Corporation.

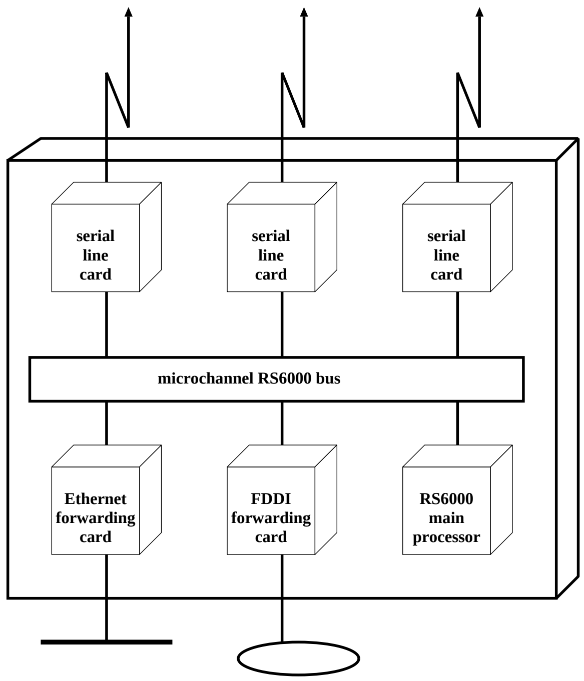

Figure 3.4: T3 NSFNET central nodal switching system (CNSS) architecture

Interface Performance

The mechanism for collecting interface performance statistics did not change from the T1 to the T3 backbone. Each backbone node, also called a Nodal Switching Subsystem (NSS) runs SNMP servers which respond to queries regarding standard SNMP Management Information Base (MIB) variables. Centralized collection of data via SNMP from each backbone interface on each NSS occurs once every 15 minutes. The counters are cleared in only two cases: when the machine is restarted; and when the 32 bit counters overrun. Cumulative counters, retrieved using the SNMP, include those for packets, bytes, and errors in and out of each interface. Other SNMP-based variables include descriptors of the current link topology and interface downtime to measure availability of NSFNET components. Table 3.1 compares the SNMP objects collected on the T1 and T3 backbones. Among other changes, the T3 backbone now supports counters of non-unicast packets.[4]

We note that thus far most metrics in standardized SNMP MIBs focus on immediate operational requirements of the network, and do not support objects that have not proven of vital interest, such as counters of packets or bytes into and out of a given interface by protocol, packets dropped due to queue overflow, and packet interarrival times. The NSFNET supports collection of some of these objects via custom-designed tools such as the one we discuss in the next section. Other objects are considered too expensive to justify the cost of their collection.

Packet categorization

Unlike the SNMP statistics, the data collection process for packet categorization was modified with the transition from the T1 to the T3 backbone. We briefly describe the process for both backbones.

As described above and depicted in figure 3.3, each T1 backbone node (NSS) was actually a set of interconnected IBM RT/PC processors, one of which was dedicated to statistics collection. To categorize IP packets entering the T1 backbone based on information contained in packet headers, this processor examined the header of every packet traversing the intra-NSS processor intercommunication facility, and used a modified version of the NNStat package [71] to build statistical objects based on the collected information.

[4] Object definitions found in McCloghrie and Rose [73] [74].

Table 3.1: SNMP objects collected per node on T1 and T3 backbones

object	description	**T1**	**T3**
ifOperStatus	operational status	Y	N/A
sysUpTime	system uptime	Y	Y
ifDescr	interface descriptors	Y	Y
ipAdEntIfIndex	IP address corresponding to interfaces	Y	Y
is-isIndex	remote address to interface index mapping	N/A	Y
ifInErrors	incoming errors occurring interface	Y	Y
ifOutErrors	outgoing errors occurring on interface	Y	Y
ifInOctets	bytes entering interface	Y	Y
ifOutOctets	bytes exiting interface	Y	Y
ifInUcastPkts	unicast packets entering interface	Y	Y
ifOutUcastPkts	unicast packets exiting interface	Y	Y
ifInNUcastPkts	non-unicast packets entering interface	N/A	Y
ifOutNUcastPkts	non-unicast packets exiting interface	N/A	Y

Because all packets traveled across the interconnection facility on their way through the node, the collection processor could passively collect data without affecting switching throughput. Nonetheless, the nodal transmission rate did eventually surpass the capability to keep up with the statistics collection in parallel, and this processor had to eventually revert to sampling [75].

The design of the T3 backbone required significant modification of this data collection mechanism. This modification actually occurred in two phases. In the first statistics collection design, all forwarded packets had to traverse the main RS/6000 processor itself, imposing a burden on the single packet forwarding engine and impeding comprehensive statistics collection. Figure 3.4 illustrates the current design of the backbone nodes, which offloads the forwarding capability to the cards as described in section 3.2.1. Because the packets do not necessarily traverse the main processor, accommodating the statistics collection required moving the software which selects IP packets for traffic characterization into the firmware of the subsystems themselves. Each subsystem forwards its selected packets, currently every fiftieth,[5] to the main CPU, where the collection software performs the traffic characterization based on these sampled packets. Multiple subsystems, including those connected to T3, Ethernet, and FDDI external interfaces, forward sampled packets to the RS/6000 processor in parallel. In Chapter 4 we discuss the ramifications of this sampling.

Because the main CPU card performs the categorization but the cards can switch packets without the CPU, the statistics aggregation mechanism does not affect switching throughput of the NSS. The sampling can, however, impose a burden on the subsystem-to-card bandwidth, and potentially interfere with other critical responsibilities of that bus, such as transferring routing information between the system and the card. Although the packet categorization mechanism at each node differs on the two backbones, the centralized collection of the data is the same. Every fifteen minutes, a central agent queries each of the backbone nodes, which report and then reset their object counters.

Table 3.2 illustrates the traffic characterization objects collected on the T1 and T3 backbones. Note that the T3 backbone only supports collection of the first three objects. The first item in the table, the matrix of network-number-to-network-number traffic counts, forms the basis for publicly available files characterizing traffic across the NSFNET backbone in terms of both individual network numbers and countries. Both backbones also support objects describing the distribution of packets by protocol (e.g., TCP, UDP, ICMP) and TCP/UDP port (application).

Internodal latency

On the T1 backbone, Merit used the `ping` utility to perform internodal latency assessments. `Ping` probes from one endpoint of the network to another using the ICMP Echo functionality [76] to record the

[5] The sampling microcode in the subsystem does not send the whole packet, but rather the first $min(packetsize, 128)$ bytes of the packet, starting from the beginning of the IP header [1].

Table 3.2: packet categorization objects collected per node on T1 and T3 backbones

Object	**T1**	**T3**
relative to exterior nodal interface		
source-destination matrix by network number (packets/bytes)	Y	Y
TCP/UDP port distribution, well-known subset (packets/bytes)	Y	Y
distribution of protocol over IP (e.g., TCP, UDP, ICMP) (packets/bytes)	Y	Y
Packet-length histogram at a 50-byte granularity	Y	N/A
packet volume going out of backbone node	Y	N/A
NSS-centric (entire node)		
per second histogram of packet arrival rates	Y	N/A
NSS (intra-NSFNET) transit traffic volume	Y	N/A

round-trip times (RTT) between the two endpoints. As of 1 February 1993, ANS collects delay data between nodes using the yet-another-`ping` (`yap`) utility, which runs on each backbone node and can measure delay to the microsecond level using the AIX system clock.

During the lifetime of the T1 backbone, and currently on the T3 backbone, the probe measurement occurs five times at the beginning of every fifteen minute interval between all pairs of backbone access points. On the T3 backbone, the architecture includes both external and internal access points (ENSSs and CNSSs) as described in section 3.2.1; ANS collects round-trip delay statistics between both sets of access points.

Halving this value yields an approximate one-way delay for the delay matrix among all the access points. On a relatively uncongested backbone with stable and symmetric routing, such a method of achieving one-way delays is justified.[6] A backbone node temporarily stores the delay data, transferring it routinely to a NOC data collector. From these statistics Merit and ANS publish reports of quartile statistics on the monthly internodal delay.

For the T3 backbone, ANS has recently investigated how to present the data to allow more insight into average delay behavior. They began by developing a new report format that includes six tables: four matrices of delay data between ENSSs and two matrices of delay data between CNSSs. The six tables present:

1. median delays between all pairs of ENSSs, and the change in the median from the previous month
2. a filtered view of the above: the median and difference appear only if there was some change from last month
3. median and interquartile difference (IQD) in delays between all pairs of ENSSs; the interquartile differences provide a measure of the spread of the distribution of the data
4. a filtered view of the above: the median and IQD appear only if the IQD is greater than 1 millisecond, highlighting backbone links with higher jitter
5. median and interquartile difference in delays between all pairs of CNSSs; this table duplicates the format of table 3 above for the CNSSs
6. a filtered view of table 5, duplicating the format of table 4 above for the CNSSs

3.3 Utility of collected statistics for analysis of the infrastructure

In the remainder of this chapter we illustrate the usefulness of operationally collected statistics for workload characterization tasks outlined in Chapter 2. Section 3.4 discusses how the statistics can support host-centric workload and performance metrics, including delay, loss, and throughput. Section 3.5 discusses

[6] Claffy *et al.* [11] provides evidence of the failure of round trip delays to adequately characterize unidirectional latencies across a wide-area network.

in more detail how the statistics can support network-centric metrics. Section 3.5.1 discusses a study in using operationally collected statistics to model long-term traffic volume. Section 3.5.2 presents metrics of reachability, i.e., the increasing geographic and administrative scope of the Internet. In particular we discuss the IP network address structure, and how the status of an IP address relates to the evolution of available Internet network numbering space. Section 3.5.3 discusses locality metrics, which measure the non uniform distribution of traffic among reachable sites. Section 3.5.6 discusses the growth in application/service diversity on the Internet as measured by TCP/UDP port numbers. Sections 3.5.4 through 3.5.9 cover other metrics that are more difficult to support with operational statistics: burstiness, payload, flow counts, per-protocol and per-environment characteristics, and other dynamic traffic flow aspects.

We focus on the limitations of current statistics collection methodologies, which were initially designed to support immediate engineering and planning needs, such as securing routing stability and tracking the rough cross-section of traffic. Suboptimalities in their architecture and implementation inhibit their effective use for some long-term forecasting and planning objectives, and completely confound attempts at characterization of short-term traffic behavior or performance assessment.

3.4 Host centric perspective

In general, operational statistics are somewhat limited in their ability to assess host-centric characteristics, most of which are performance metrics. Thus our discussions in this section are brief relative to those of network-centric metrics in the next section. One factor contributing to the imprecise state of Internet performance metrics is the lack of a common definition of Internet performance criteria and realistic measurements by which to judge them. For example, the length of queues in routers may provide some indication of congestion, but is in general too expensive for routers to continuously assess.

Currently collected statistics typically concentrate on performance bottlenecks of packet switches and switch interconnections. While network operators may discuss throughput rates and link utilizations, they do not ordinarily attend to details of traffic flow behavior. Lack of such data makes it difficult to parameterize service requirements of network clients such that the network operator could assess their ability to meet them. For example, there is no operational statistic to measure how responsive “the network” is to a transaction, much less associated margins of predictability for such an Internet response time metric.

3.4.1 Delay and jitter

Groschwitz [77] has recently investigated the utility of the collected node-to-node packet delay statistics on the NSFNET backbone described in section 3.2.2. Using the round-trip delay data collected on the T1 NSFNET backbone, she investigated the relationship between traffic volume and measured round-trip delay, and compared the trends in measured values to those predicted by queueing theory. The model predicted reasonably well the fixed (processing and propagation) delays but was less successful in predicting the effects of added traffic on the queueing delay. Although she did find evidence of a relationship between traffic volume and delay, the relationship appeared to be linear; there was no evidence for the curve predicted by queueing theory. One factor contributing to the disparity between theoretical and measured delay is the collection granularity, as defined in section 2.1. Because the delay data collected every 15 minutes are not nearly representative of delay throughout the 15-minute interval, quantifying the relationship between traffic volume and delay is difficult. Better correlation of performance with utilization levels on the backbone would require monitoring delay and utilization at an interval finer than fifteen-minutes. Most network service providers do not see a benefit to justify the resource consumption such frequent delay measurements would impose.

3.4.2 Loss

The primary loss statistics for the NSFNET backbone are the SNMP interface error counters described in section 3.2.2. These aggregate counters do not distinguish among the different causes of error, such as HDLC checksum errors, invalid packet length, and queue overflows resulting in discards. As with delay metrics, other metrics relevant to loss, such as nodal queue lengths, missing statistics per unit of time, or percent of time an interface is unavailable are difficult to gather because they require high resolution polling which results in undesirable network load. Worse yet, querying nodes for statistics such as queue length will deleteriously affect the value you are trying to measure. Because the cost of monitoring these variables exceeds their benefit to network operation, network operators in general will not make the effort.

In late 1993 ANS did implement and test a custom utility on the T3 NSSs to collect statistics on card to card transfers, including the number of packets dropped due to buffer exhaustion. The losses were consistently low enough that ANS did not continue to support the collection operationally.[7] However, in the face of growing traffic volume and performance expectations, ANS plans to support congestion assessments again on the NSS routers soon.

3.4.3 Throughput

The SNMP packet and byte counters described earlier provide measures of average throughput on the NSFNET backbone over fifteen minute intervals, but the NSFNET backbone supports no operational characterization of maximum achievable throughput, since any such assessment would deleteriously affect the overall performance of the network. Network operators may run throughput tests during installation to verify hardware or software specifications, but rarely perform ongoing assessments. If network clients want to assess throughput, they must resort to their own spot checks, e.g., results of FTP file transfers or related tools. Related to throughput are raw nodal *packet switching rates*, of both end hosts and intermediate nodes. As with throughput assessments, the NSFNET backbone operators test packet switching rates for special needs, such as during software or hardware upgrades, but not as a part of operational statistics collection.

3.5 Network centric perspective

3.5.1 Long-term utilization: traffic volume

The SNMP packet and byte counters that measure average throughput on a fifteen minute granularity are also measures of utilization at that granularity. ANS, the current backbone service provider, uses these statistics to assess utilization on a daily, weekly, and monthly basis.

Figure 3.5 uses monthly compounded statistics to show the quadratic growth in packet volume during the last five years of the NSFNET backbone operation. Groschwitz and Polyzos [78] have found that the operationally collected data do provide a good framework for models of long-term traffic volume on the backbone. The authors used ARIMA time-series modeling techniques [79] to characterize several years of NSFNET packet traffic growth, based on the operationally maintained daily packet totals[8] for NSFNET backbone nodes from August 1, 1988 to June 30, 1993. Their goal was to determine the strength of time-series models in making detailed long-range predictions about NSFNET backbone traffic at the granularity of the collected data. They found a reasonably close match between predicted and observed traffic levels, suggesting that modeling techniques would be adequate for long-range forecasts of fairly coarse granularity.

Although their ARIMA modeling study focused on the aggregate traffic volume, they also suggest that one could use ARIMA models to make predictions about traffic volume on individual NSFNET backbone nodes or links. A model of monthly traffic patterns on individual links would allow one to forecast when traffic will likely exceed link capacity. Alternatively, if traffic on a less busy link is growing more rapidly than on another link, models of each could predict when their usage levels would cross, which might influence routing decisions. However, preliminary model building suggests that it would be necessary to use a separate model for each node, unsurprising given the differences among traffic handled by different nodes. Other limitations in prediction derive from the time-series modeling techniques more than the collected data itself. Time-series models make predictions based only on previous data, and thus cannot take into consideration outside forces that may fundamentally change the pattern of the data. As multimedia applications are more widely deployed and used, the high traffic volume they generate will influence traffic patterns. Other new technologies, new government regulation, and changes in the national economy will also have significant effects that the ARIMA approach cannot predict.

[7] ANS reports that the highest congestion loss during the two month measurement period in 1993 was along the Chicago and Cleveland T3-B routes at about 0.1% loss over 1 hour periods [1].

[8] Merit Network, Inc. collected all data in its role of operation and management of the NSFNET backbone.

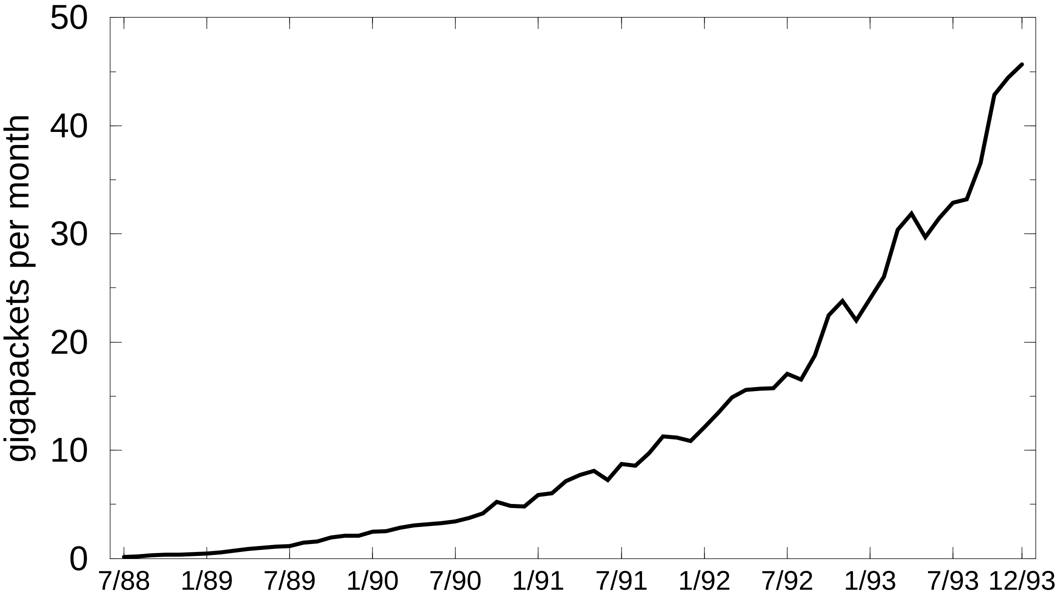

Figure 3.5: long-term growth of packet volume into the NSFNET
(Data source: Merit/NSFNET operations)

3.5.2 Reachability

Besides sheer traffic volume, another growth metric is the number of connected networks as reflected by traffic reaching the NSFNET backbone. Figure 3.6 shows the long-term growth of network numbers configured for communication via the NSFNET backbone [80]. These NSFNET numbers are the only destinations to which the NSFNET backbone will route packets; every network client must have an assigned IP network number to receive traffic. The figure shows dramatic growth over the last few years, including substantial increases in the international area.

Smith [81] [82] explored how to best represent NSFNET growth of both the total amount of traffic traversing the backbone and the number of connected and configured networks. He evaluated how much of the traffic increase was due to newly added networks versus an increase in usage by the older, more established networks. Three significant trends emerged:

1. the percentage of newly assigned IP addresses used during their first month is decreasing steadily.
2. a smaller percentage of configured networks are responsible for an ever-increasing portion of the new network traffic.
3. monthly changes in overall volume are mostly due to increases or decreases in activity from established networks, not to increases or decreases in new network activity.

This last observation is one facet of the complexity of accurately measuring reachability. A more basic danger in mapping the metric of assigned IP network numbers to the growth of the Internet is that not all IP addresses are created equal. To elucidate the problems in measuring reachability, we devote the next section to a discussion of the significance of the IP address structure to the service reachability of the NSFNET.

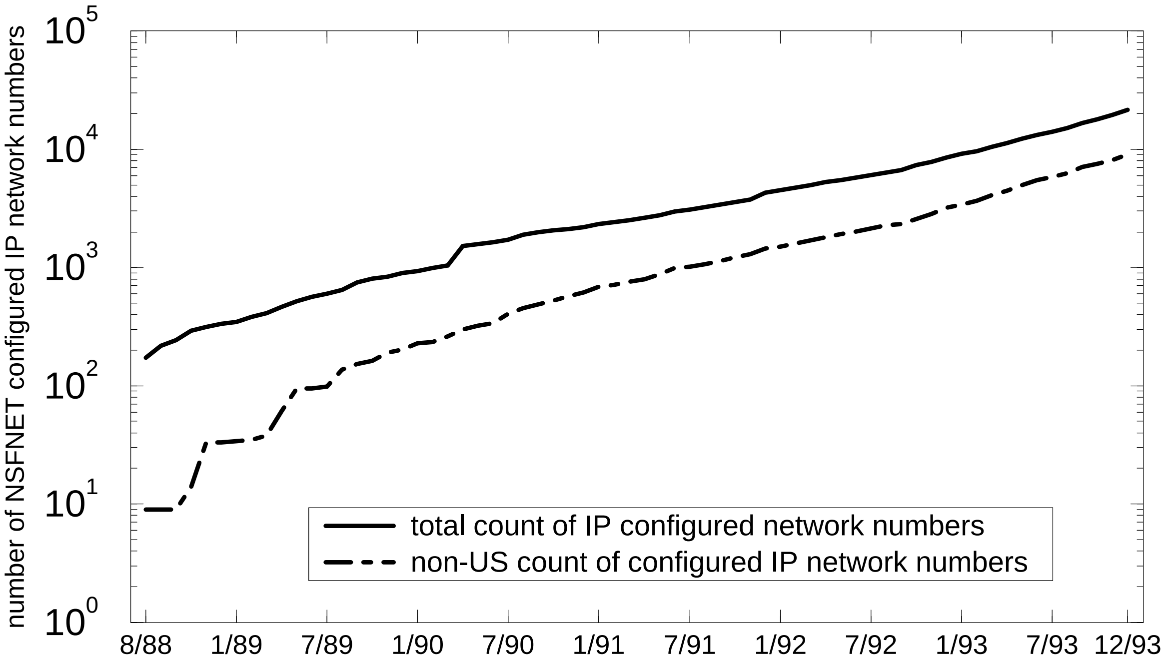

Figure 3.6: long-term growth of network numbers served by NSFNET
(Data source: Merit/NSFNET operations)

Background on IP address space

The IP address space architecture originated with RFC 791 [83], the initial Internet Protocol specification that defined a pool of available network numbers. Ignoring some special cases, such as multicast addresses, every network number on the Internet came from this pool of available network numbers. A large subset, although not every number in this pool, has been assigned to a requestor, typically on behalf of a company, university or other institutions, for active duty. The InterNIC[9] Registrar, on behalf of the Internet community, now formally registers these assigned network numbers in a database that also includes mappings to address information of the institution responsible for the network.

Each IP address is four bytes long, part of which identifies the IP network and the remainder of which identifies the host within that network. The IP protocol specifies three commonly used address classes which differ in the size of the host address component within the four byte IP address. Class A, B, and C networks have three, two, and one-byte host fields, allowing for a maximum of 2^{24}, 2^{16}, and 2^{8} individual addresses or hosts, respectively. The number of allocatable class A, B and C network numbers is 2^{7}, 2^{14}, and 2^{21}, respectively [84].

Over the years the Network Information Center (NIC) assigned IP network numbers to clients according to the number of hosts to be supported. Class B addresses were most attractive due to their size, which facilitated subnetting in an environment with many hosts. Eventually the InterNIC implemented restrictions on class B address assignment, using instead groups of class C addresses unless the client could justify the need for a class B space. This policy resulted in a deceleration of class B assignments, but substantially accelerated the assignment of class C addresses.

Since addresses from different classes absorb different amounts of the available 32-bit address space,

[9] Established by the National Science Foundation, the InterNIC (Internet Network Information Center) is a collaborative project of three organizations working to provide network information services. General Atomics provides Information Services; AT&T provides Directory and Database Services; and Network Solutions, Inc. provides Registration Services. Prior to April 1993 the Defense Data Network's Network Information Center (DDN NIC) performed this registry function.

figure 3.6 presented earlier, which depicts the growth in network number addresses regardless of address capacity, offers only part of the story. Figure 3.7 presents more of the picture, showing the growth in *total committed address space* served by the NSFNET. Currently, about 60% of the total available address space is assigned. In terms of total committed address space, not network numbers, the class A and class B growth has dwarfed the class C growth in the past. However the figure depicts an exponential growth in class C committed space that has accelerated with the new InterNIC policy, reflected by a change in slope of the bottom line of the graph in recent months (which would look much more dramatic on a linear y-scale). If current growth patterns continue according to this figure, the exponential growth of the class C committed address space will exceed that of the committed class A and class B spaces. Note that two class A address holders turned their addresses back in early 1993, causing a dip in the graph.

Figure 3.8 presents a schematic of the categories of IP network numbers as we will describe them. With the advent of RFC1366 [85] [86] in October 1992, the InterNIC began to assign addresses according to the geographic location of the requestor. The InterNIC also, in certain cases, delegates blocks of class C IP network numbers to other authorities for further assignment. For example, the InterNIC delegated a large portion of the class C space to Europe for further redistribution within their network community. From the InterNIC's point of view, these *delegated network numbers* are no longer available but not yet formally assigned until the Europeans notify them that they have really assigned those numbers to their final client IP networks.

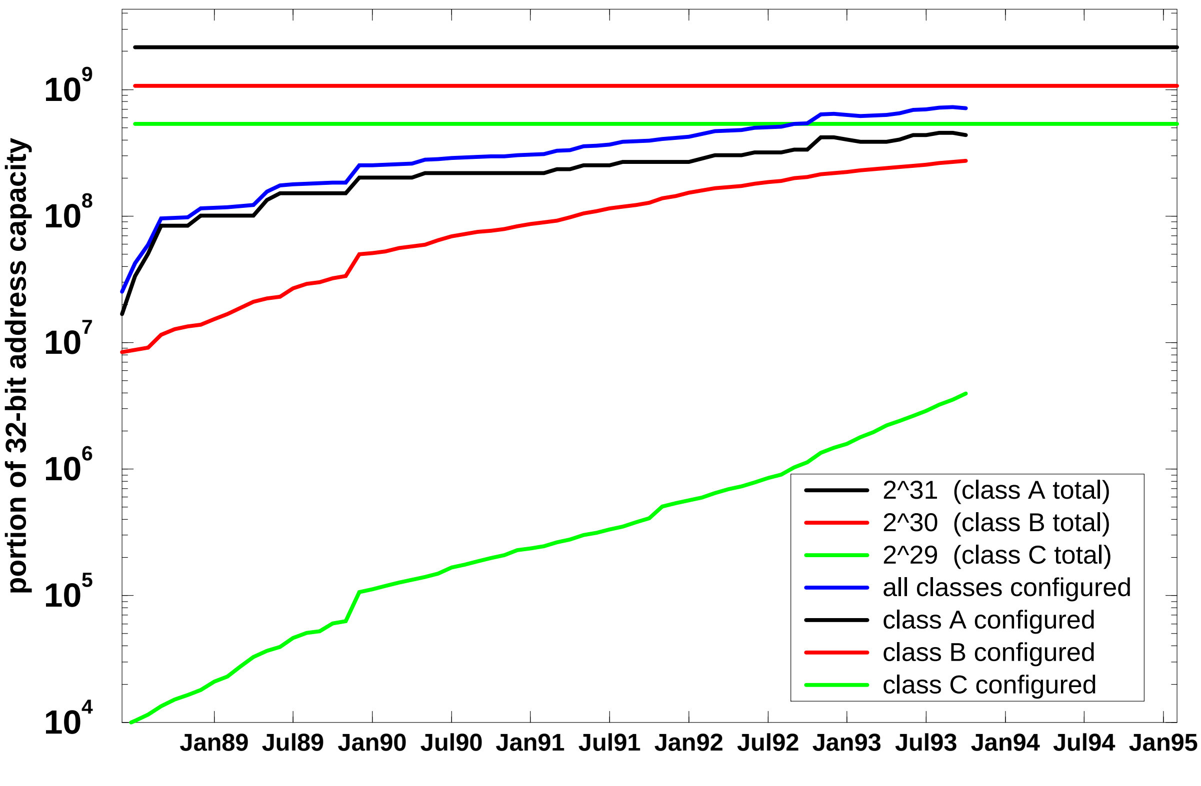

Figure 3.7: long-term growth in committed address space by assigned class A, B, and C IP network numbers served by the NSFNET
(Data source: Merit/NSFNET operations)

Assigned IP network numbers are necessary but not sufficient for communication across the NSFNET backbone, which is an important component of the global Internet. The NSFNET backbone uses a policy routing database as a truth filter to verify the validity of dynamic routing information that its backbone clients have explicitly specified. This database represents the set of *NSFNET-configured* network numbers that the NSFNET serves, a proper subset of the assigned network numbers. However, even though a network may be in this NSFNET database, the backbone still will not be able to send traffic to that network until it receives a dynamic announcement from that network via a router of a directly attached NSFNET client by means of an inter administrative domain routing protocol such as BGP or EGP. This announcement from an NSFNET client, either a mid-level or some other network connected to the backbone, reaches the NSS, which evaluates each incoming announcement, accepting those for configured nets that come from appropriate peers in an appropriate administrative domain, identified by its autonomous system number. This action turns an NSFNET-configured network into an *NSFNET-announced network*. The configuration database serves to sanitize dynamically announced routing information before the backbone actively utilizes it. This filtering is essential to the sanity of the larger infrastructure, and other networks often use similar mechanisms to accomplish the same task. Upon acceptance of the announcement, the NSS tags a path priority value, or metric, to the network number, to enable comparison to other announcements of the same network number.

After a network is assigned, configured, and announced, it can both send and receive traffic over the NSFNET backbone as an active network. A network remains active as long as connectivity exists to the destination and the appropriate service provider(s) announces the network directly to the backbone according to the appropriate procedure.

Unfortunately, Internet reality is not entirely faithful to this model. Theoretically, any network that sends traffic is active, even if it is not assigned, configured, and announced. To make these categories unambiguous, we call an illegitimately active network, i.e., a network missing any one or more of the three essential properties (assigned, configured, and announced), a *leaky network*.

One particularly insidious source of leaky networks are organizations that consider their local network environments wholly disconnected from the Internet, and with no plans for future connection, arbitrarily choose their own IP network numbers, independent of the InterNIC's registry, to satisfy their isolated TCP/IP protocol needs. Unfortunately, experience has shown that traffic from such networks often manages to find its way into the Internet, much to the surprise or ignorance of the local network administrators. When these network numbers become active, they join the set of *leaky unassigned numbers.* Leaky networks, particularly from unassigned networks, can pose difficulties for network operators in both routing and traffic management. An example of the complication to routing is a leaky network that is not configured for the NSFNET but sends routing announcements as though it were. Network operators must protect themselves against incorrect information that such networks inject into inter administrative domain routing protocol exchanges. An example of the complication to traffic management is an unassigned or unconfigured leaky network that sends traffic to a legitimately active network. Their packets may actually get delivered to the remote location, the if the remote location is assigned, configured, and announced. But the NSFNET will not deliver traffic for the return path to the original source. and if the legitimate destination network complains to the NSFNET about receiving traffic to which it cannot reply, or which it does not want in the first place, the network operator has little recourse to manage the problem, since the source host could easily have randomly chosen an IP network address for the endeavor. Needless to say, the Internet Assigned Numbers Authority (IANA) discourages the TCP/IP community from using self-selected network numbers, considering it uncivilized behavior in the increasingly, and often transparently, interconnected world.

Also problematic is the case of silent networks. Silent networks are those configured or announced but not active; i.e., they have not sent traffic across the backbone. The NSFNET project has analyzed the NSFNET Policy Routing Database (PRDB) and has developed methods to eliminate silent nets to prevent the needless burden on routing table size. But the number of silent networks has increased since announcement of the addressing guidelines outlined above. When service providers receive large blocks of class C addresses in anticipation of and aligned with Classless Inter Domain Routing (CIDR) [87] requirements, they immediately configure the addresses with the NSFNET before assigning them to customers, increasing the number of silent networks in the NSFNET backbone configuration database. Eventual CIDR deployment will rely on network masks to reduce such blocks to a single entry in the routing table, but until that time they pose an obstacle to efficient configuration of routing databases.

For the purpose of statistics collection and traffic analysis, there are a variety of ways to contend with these categories of network numbers. As an example, during the month of December 1992 the T3 backbone nodes recorded traffic from more than 14,000 networks, potentially spanning all categories in figure 3.8. Of these, about 9,700 were networks in the set of NIC-assigned network numbers. The number of

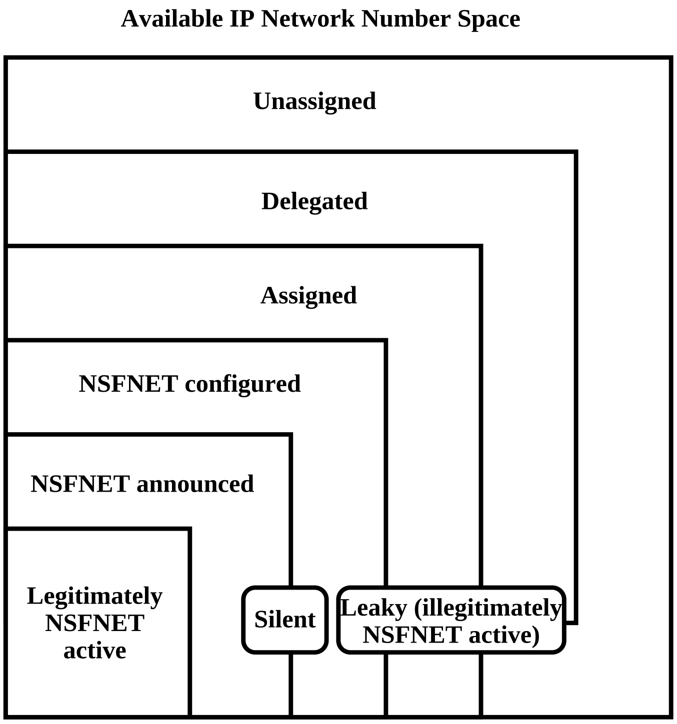

Figure 3.8: descriptive categories of IP network numbers

active networks that was also configured in the NSFNET/ANSnet topology database that month was 6,131. Leaky networks, assigned but not configured, who may send traffic into the NSFNET backbone, account for much of this disparity. Another contributing factor is out of date information in the Domain Name System about the addresses of root domain servers, which may be configured but no longer announced but to which many networks still try to send traffic.

More generally, the disparity arises from networks who send traffic to another network that the NSFNET does not recognize because the network is one of the following: configured but not announced, assigned but not configured, or not even assigned. The NSFNET may see such traffic, for example, when network service providers use a default route pointing to the NSFNET. However, because routing information for these destinations will not exist in the NSFNET forwarding tables, as soon as such packets reach the NSFNET the backbone node filters them out during the routing decision.

3.5.3 Locality

As mentioned in section 2.4.3, locality refers to the concentration of traffic among a small subset of possible network addresses. Taking advantage of locality can have great potential benefit in large scale infrastructures, which exhibit substantial gaps in traffic volume between the heavy and light flows. Although today's networks aggregate traffic among more flows than ever before, their connectivity matrix is clearly not uniform.

In this section we use operationally collected data from the T3 backbone during December 1992 to present metrics reflecting traffic locality. For the purposes of this analysis, we collated traffic data only to or from configured network numbers, and normalized our favoritism calculations to the amount of traffic sent among the 6,131 configured networks of the T3 backbone. As described in section 3.5.2, configured network numbers are the only ones for which NSFNET promises a service; traffic to other networks is essentially noise that imposes a service impact. There are also networks that are configured but apparently never announced to which many networks still try to send traffic.[10]

[10] Out of date information in the Domain Name System about the addresses of root domain servers is one source of such

Table 3.3: summary statistics on traffic locality on the T3 backbone in December 1992

Percent of (1378065) Network pairs responsible for traffic					
percent of traffic	25%	50%	75%	90%	97.5%
number of site pairs	740	4173	21132	74795	224015
percent of site pairs	0.054	0.303	1.53	5.43	16.26
Percent of (6131) Networks responsible for traffic					
percent of traffic	25%	50%	75%	90%	97.5%
number of total nets	7	42	150	399	1022
percent of total nets	0.114	0.685	2.446	6.508	16.669
Percent of (95) ADs responsible for traffic					
percent of traffic	25%	50%	75%	90%	97.5%
number of total AD's	6	13	24	34	45
percent of total AD's	6.32%	13.68%	25.26	35.79	47.37
Percent of (55) NSSs responsible for traffic					
percent of traffic	25%	50%	75%	90%	97.5%
number of total NSSs	3	7	12	17	22
percent of NSSs	5.45	12.73	21.82	30.91	40.00

Distribution of destination networks for three source networks					
Percent of (3906) Networks to which UCSD Sent Traffic					
percent of traffic	25%	50%	75%	90%	97.5%
number of networks	10	39	149	353	666
percent of networks	0.256	0.998	3.815	9.037	17.051
Percent of (1590) Networks to which SDSC Sent Traffic					
percent of traffic	25%	50%	75%	90%	97.5%
number of networks	1	10	60	182	350
percent of networks	0.063	0.629	3.774	11.447	22.013
Percent of (1451) Networks to which NSF Sent Traffic					
percent of traffic	25%	50%	75%	90%	97.5%
number of networks	16	45	116	254	530
percent of networks	1.103	3.101	7.994	17.505	36.527

Table 3.3 provides statistics on the concentration of T3 backbone traffic for December 1992 at the various granularities of aggregation: network pairs; networks; Administrative Domains (ADs), and nodal switching subsystems. Only 1.5% of the more than 1.3 million network pairs were responsible for approximately 75% of the traffic on the backbone. The traffic was somewhat more evenly distributed among the more coarsely aggregated groups, such as Administrative Domains.

Table 3.3 also provides measures of one-sided favoritism for three networks. Using selected networks at UCSD, SDSC, and NSF, the table shows the proportion of traffic from each network to its n favorite destinations for the month of December 1992. Kleinrock and Naylor referred to this tendency on the 1973 ARPAnet as the "favorite site" effect [65]. One can see a definite difference in workload profile among these three networks; the supercomputer center tends to exhibit a higher degree of favoritism than the other networks, and the campus communicates with a significantly larger number of sites. Note that favoritism at each source site involves a separate set of most popular destination sites, since each source need not have the same set of favorites. The number of users at each site (typically a supercomputer center such as SDSC will have considerably fewer than a university campus) may also affect this balance. NSFNET backbone traffic also exhibits a "favorite source" effect, i.e., a heavy concentration of traffic from a selected set of sources to a given destination.[11]

Given 6,131 configured networks, which can all communicate with each other, over 37.5 million net pairs could conceivably communicate. Of these, the NSFNET recorded traffic from only 3.67% (1.38 million) of them. Furthermore, 1% of those networks was responsible for 70% of the traffic for the month of December 1992. In other words, the top 15,000 net pairs collectively consumed approximately 70 times the bandwidth they would have consumed had the traffic matrix among communicating net pairs been uniform, and approximately 400,000 times the bandwidth they would have consumed had the traffic matrix among all configured net pairs been uniform.

These statistics, collected on a fifteen-minute basis and aggregated monthly, allow a reasonable assessment of locality over the medium to long term. The statistics are less useful for assessment of short-term locality, i.e., at time granularities under 15 minutes, such as packet train effects outlined in section 2.4.3. We will discuss short-term locality metrics in Section 7.3.

3.5.4 Burstiness

The 15-minute collection granularity of the NSFNET traffic volume statistics suffices for the long-term traffic volume assessment described in section 3.5.1, and allows tracking of evening, weekend, and holiday fluctuations. However, analyzing the dynamics of shorter-term traffic fluctuations, including prediction of bandwidth requirements for continuous media data flows, requires a much finer time granularity. As with many of the performance metrics discussed earlier, e.g., loss, throughput, delay, the degree of polling necessary to assess burstiness itself adds undue burden to the switching processors. Thus the NSFNET essentially does no characterization of fine-grained traffic burstiness.

Even if one could obtain finer grained burstiness measures, without detailed packet traces and state-based analysis it would be impossible to attribute given bursts to their sources, e.g., whether 3 our of 10,000 flows are causing 70% of the utilization, with the other 9997 consuming the rest. Such attribution requires more comprehensive collection of packets and subsequent analysis of the burstiness (e.g., packet interarrival time) characteristics within individual flows that contribute to the workload. Profiles of individual flows, whether based on applications, end hosts, or some other desired flow granularity, can also indicate trends in user demand that will influence router, bandwidth, and topology requirements.

3.5.5 Payload

The statistics collection software on the T1 NSFNET backbone supported the collection of packet size distributions of traffic passing through each node, but during the upgrade to the T3 backbone ANS engineers removed this functionality since they found little use for it to justify the resources it required. One can still obtain rough indicators of payload using the ratio of bytes to packets into and out of each node

behavior. Section 3.5.2 provides a detailed explanation of the relevance of network numbers and their configuration to traffic characterization.

[11] Further details on NSFNET traffic locality appear in Claffy *et al.* [24] [88].

aggregated every fifteen minutes. These data show daily cycles which are compatible with the hypothesis of bulk transfer applications, using larger packet sizes, intensifying during the off-peak hours, or correspondingly, that interactive traffic, generally characterized by smaller packet sizes, drops off during off-peak hours. In the longer term, such as monthly averages over the last few years, there is no particular trend in packet size.

3.5.6 Traffic cross-section

The increasing diversity of Internet applications presents another difficulty in characterizing long-term traffic trends on the NSFNET. However although the diversity makes tracking the traffic cross-section more difficult, it also makes it more important. As new applications often demand different, and higher, performance requirements, assessment of the type and scope of the range of applications on the infrastructure will be critical to network service planning. In this section we describe how the IP architecture limits the ability to attribute traffic by application.

Most applications on the NSFNET are built on top of the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). Both TCP and UDP packets use port numbers to identify the Internet application that each packet supports. Each TCP or UDP header has two 16-bit fields to identify the source and destination *ports* of the packet. Originally, the Internet Assigned Numbers Authority (IANA), at ISI (Information Sciences Institute, University of Southern California), on behalf of DARPA, administered a space of 1 to 255 as the group of port numbers assigned to specific applications. For example, *telnet* received port assignment 23 [89]. To open a *telnet* connection to a remote machine, the packet carries the destination IP address of that machine in its destination IP address field, and the value of 23 in the destination port field.[12]

During the early years of the TCP/IP-based Internet, particularly in the early eighties, Unix developers injected a bit of anarchy into the IANA system when they unilaterally began using numbers between 512 and 1024 to identify specific applications. For example, they used port 513 for rlogin. Eventually network users started to use numbers above 1024 to specify more services, extending the lack of community coordination. In July 1992 [89], the IANA extended the range of port assignments they manage to 0-1023. At this time, they also began to track a selected set of registered ports within the 1024-65535 range. IANA does not attempt to control the assignments of these ports; they only register port usage as a convenience to the community [89]. Figure 3.9 presents a schematic of the port number categories as we have discussed them.

These port numbers are the only mechanism through which the NSFNET can monitor statistics on the aggregated distribution of applications on the backbone. Specifically, Merit (and now ANS) collects port-based information in the ranges 0-1023, 2049 (for NFS) and 6000-6003 (for X-window traffic). Merit/ANS categorizes packets into these ports if either the source or destination port in a given packet matches one of these numbers. However, even within this range, not all ports have a known assignment. So packets using undefined ports go into the *unknown* port category [80].

Figure 3.10 uses this data to classify the proportion of packet traffic on the network by category since August 1989, based on categories in use by the NSFNET backbone service provider. These figures indicate an increasing diversity in the cross-section of NSFNET traffic, and a decreasing reliance on conventional protocols such as *telnet*, *ftp*, *smtp*, relative to the overall network traffic.[13] The categories in these figures correspond to:

- file exchange: *ftpdata* and *ftpcontrol* (TCP ports 20, 21)
- mail: *smtp*, *nntp*, *vmtp*, *uucp* (TCP ports 25, 119, 175, 540)
- interactive: *telnet*, *finger*, *rwho*, *rlogin* (TCP ports 23, 79, 513, UDP port 513)
- name lookup/*DNS*: (UDP port 53, TCP port 53)
- other TCP/UDP services: all TCP/UDP ports not included above (e.g. *irc*, *talk*, *x-windows*)

[12] In the case of *telnet*, the packet uses some arbitrarily assigned source port that has significance only to the originating host. These return address ports often have values greater than 1000.

[13] Note that Merit began to use sampling for this collection on the backbone in September 1991. In November 1991 traffic migration to the T3 backbone began; the majority of the links had migrated by May 1992 and in November 1992 the T1 backbone was dismantled. For June to October 1992 no data was available for either the T1 or T3 backbones.

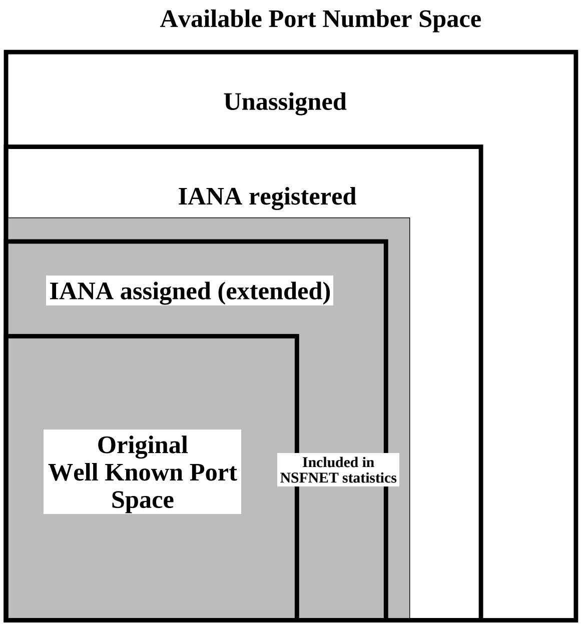

Figure 3.9: descriptive categories of TCP/UDP port numbers

- non-TCP/UDP services: Internet protocols other than TCP or UDP (e.g. ICMP, IGMP, EGP, HMP, etc.)

More detailed distribution of traffic by port on the NSFNET backbone is available via ftp from `nis.nsf.net`, and shows some details regarding the growing range of applications. Several Internet resource discovery services (e.g., *wais*, *www gopher*, *prospero*, *mosaic* [90]) have experienced tremendous growth in traffic volume since their deployment. Other applications have also gained a greater proportion of network bandwidth: e.g., *mud* (multi-user domain), a distributed electronic role playing environment; *x-windows*, for remote graphical displays across the network; and more recently, real-time applications such as packet video and audio. Many of these applications use multiple TCP/UDP port numbers, which often are not centrally coordinated and therefore unknown to anyone but the end sites using them. Such traffic is a subset of the *other services* category in figure 3.10, and add complexity to the task of Internet workload characterization. The number and traffic volume of non-categorized applications has grown much larger over the years, reflecting an increasingly multi-application environment, and a diminishing ability to assess the impact of individual new applications due to their use of non-standardized port numbers.

During 1993, ANS deployed software for the NSFNET service that allowed more flexibility with the port distribution assessments, though the inherent difficulty with the Internet model of application attribution remains. The recently established InterNIC may also allow greater flexibility in maintaining accurate databases of network number and traffic type statistics. Concerted attention to such activities will help foster an Internet environment where network planning and traffic forecasting can rely on more than the traditional traffic counters used in the past.

The issue of unknown applications is not by itself as disturbing as the dramatically changing nature of the newly introduced traffic. The recent deployment of more widespread packet video and audio applications bodes ominously for an infrastructure that is not able to preferentially deal with certain types of traffic. Bohn *et al.* [91] propose a scheme that begins to address the issue.

3.5.7 Flow profiling

Currently collected operational statistics in most environments do not yield significant insight into the structure of individual traffic flows. SNMP MIBs typically consist of simple interface counters or status variables; there is no currently defined publicly available MIB for inspecting ambient flow reservations or

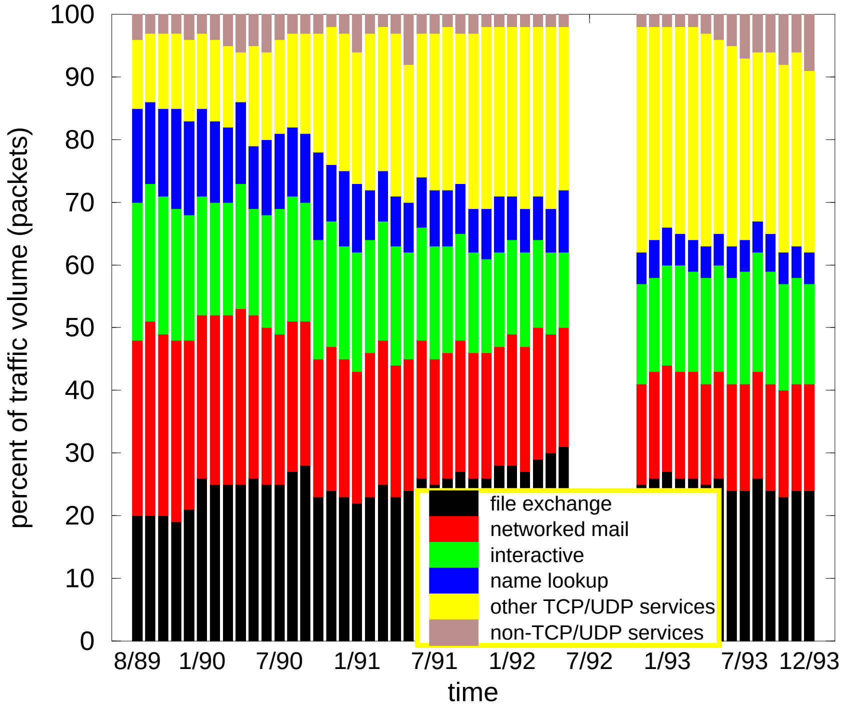

Figure 3.10: proportion of packets offered into NSFNET backbone by application categories
(Data source: Merit/NSFNET operations)

statistics. Other advanced statistics collection tools custom-designed for traffic characterization [71], such as that used on the NSFNET [72], still classify traffic based on information found each packet header in isolation. They maintain no state-based information regarding individual flows such as their packet volume or duration.

The lack of operational statistics to support flow assessment tasks led us to dedicate an investigation of flow assessment in the latter half of the thesis, with Chapter 6 focusing specifically on individual flow metrics.

3.5.8 Aggregate flow metrics

As with individual flow metrics, aggregate flow metrics require collection of data that no known wide-area networks currently support. Such metrics include the number of simultaneously active flows or the number of new flows per unit of time. We will cover aggregate flow metrics in Chapter 7. We base our flow assessments on comprehensively collected packet traces of relatively short duration. Operators of Internet components may find it advantageous to implement a flow assessment facility such as the one we have developed, for either continuous or periodic traffic monitoring on a statistics processor or a router. Network service providers could then maintain the resulting flow information on local routers, and even allow access to such information, or summaries thereof, via SNMP and a special flow MIB.

3.5.9 Environment characteristics

Operational statistics for wide-area infrastructures provide little information regarding how traffic characteristics differ at different locations in the Internet, for example at different layers of figure 2.1. One could filter out and investigate a subset of the NSFNET traffic statistics that reflect a specific component of the hierarchy to explore the impact of specific regional network, or administrative domains, or individual network numbers.

One could even use the methodology we describe in Chapter 5 to replicate flow assessments at strategically selected measurement locations for a more comprehensive image of traffic flow behavior throughout the Internet. Indeed, the few sites that we selected for testing our methodology reveal similarities and differences among components.

3.6 Conclusion

In this chapter we have used and evaluated the utility of operationally collected statistics in characterizing aspects of the workload of the T1 and T3 NSFNET backbones. Some of the current NSFNET statistics are useful for researchers, network planners, and engineers. Collected data allow tracking long-term growth in traffic volume, including attribution to domains and protocols; trend in average packet size on the network, both over long and medium term intervals; the most popular sources, destinations, and site pairs; source-centric favoritism; the international distribution of traffic; utilization statistics, both of the overall backbone as well as of specific links of interest; delay statistics; and, assessment of downtime. Our findings have extended the results of other recent investigations [77] [78] [81] [82] on the limitations of operationally collected statistics.

We also assess the difficulty of quantifying network ubiquity and service diversity for the current NSFNET environment, as measured by IP network numbers and port usage statistics, respectively. The collected statistics indicate superlinear growth of IP network numbers, and therefore Internet clients, over the last several years. The trend is clearly continuing on a global scale; international clientele as of the end of 1993 account for over 40% of the network numbers known to the U.S. NSFNET backbone. Increased interest in accounting and billing needs will render currently available data sets even more inadequate. Deployment of network number aggregation techniques (e.g., CIDR) [92], which hide the interior structure of a network cluster, will further aggravate the situation until network masks can mitigate its effect.

The statistics also reveal the tremendous growth in application/service diversity on the Internet as measured by TCP/UDP port numbers. The ever-increasing diversity in Internet application profiles, whose complexity will increase further with the newer continuous flow multimedia applications, will require reevaluation of design issues such as queueing management in routers. Even within the non-continuous flow paradigm, subcategories of traffic such as interactive, transaction, or bulk traffic, may exhibit performance requirements which require adaptive queue management. Networking research could clearly benefit from the assessment of queue length distributions, packet drop characteristics, and more detailed insight into interface error conditions, but all such collection comes at a cost, and a network operator must weigh these costs against the benefits that availability of such statistics will provide. We discuss the cost-benefit tradeoffs further in section 8.1.

In the next chapter we focus on an additional factor that constrains the usefulness of operationally collected statistics on wide-area network infrastructures. In the face of the limitations of the operationally collected statistics, we then undertake an in-depth measurement study on a more limited geographic scale. We focus specifically on characterizing traffic flows at various granularities, both in terms of usage of applications throughout the network, as well as geographic locality. Because it requires comprehensive and detailed statistics collection, characterizing flows is difficult to implement operationally. Nonetheless we advocate that network operators undertake periodic flow assessment at least over shorter intervals to obtain a more accurate image of the workload their infrastructure must support.

Chapter 4

Sampling Network Traffic

I cannot give you a formula for success. but I can give you a formula for failure: try to please everybody.

– *Herbert Swope*

It is more from carelessness about the truth than from intentional lying that there is so much falsehood in the world.

– *Samuel Johnson*

A statistician is someone who can draw a straight line from an unwarranted assumption to a foregone conclusion.

We do not underestimate the difficulty of modifying the statistics collection methodologies of operational wide-area networks. Parameterizing tool requirements for collection is challenging enough prior to the implementation of a large scale infrastructural network, and a serious undertaking if it must be retrofitted after the network has been operating for years. Developing tools is not the only obstacle; another problem in current environments is the huge amount of data which statistics collection generates. Recent dramatic increases in the speed of wide area backbones pose obstacles to complete statistics collection; managers of high-speed networks are under tremendous pressure to optimize resource usage to fulfill the data collection objective. Sampling offers a strategy to alleviate these pressures.

Implementing sampling techniques in an operational environment requires a concerted investigation into the effect of sampling on network analysis. This chapter presents a detailed study of how accurately various methods of sampling can answer questions related to wide area network traffic characteristics. The reversion of the NSFNET backbone project to sampling for gathering statistics on the backbone, and accompanying interest in its effect on traffic analysis, motivated our study of sampling network traffic data. This investigation includes methods and accuracy requirements for the characterization of aggregated traffic on a component network. Our objective is to investigate the effect of sampling on the ability to answer selected questions about network traffic characteristics.

We describe an experiment we conducted at an entrance point into the NSFNET backbone environment at which we were able to gather packet traces. These packet traces allowed us to explore the effect of different parameters of sampling, such as: (1) time-driven versus event-driven methods; (2) random versus deterministic selection patterns; (3) the granularity, or sampling fraction; (4) the interval, or length of time over which we sample. We use as characterization targets in this study the distribution of packet sizes and packet interarrival times.

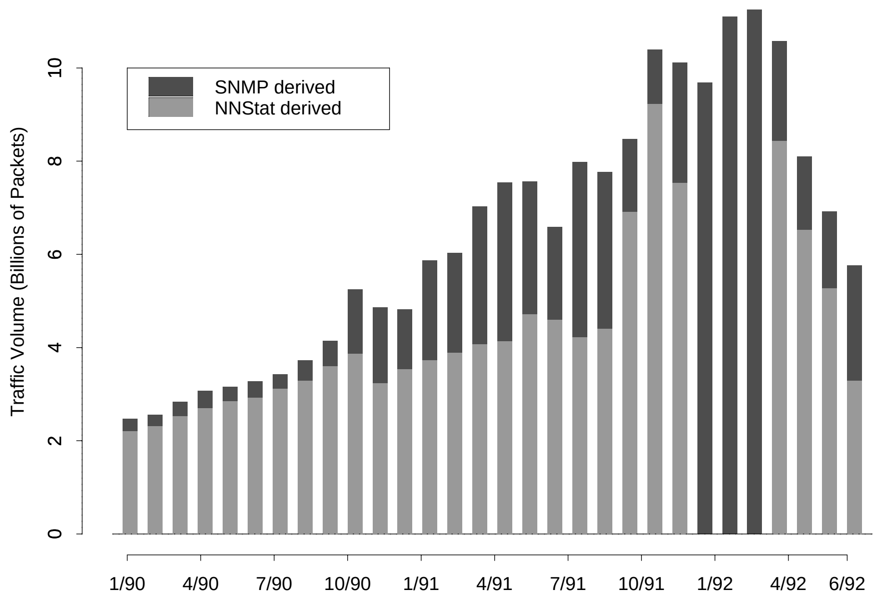

Figure 4.1: T1 backbone packet totals (billions of packets), as reported independently by SNMP and NNStat, indicate a discrepancy between the two collection processes.

4.1 NSFNET statistics collection

The statistics collection processes for the T3 NSFNET backbone, described in section 3.2.2, illustrates an example of a wide area environment faced with data collection demands that have forced the implementation of sampling.

We briefly review aspects of the statistics collection described in that section that are relevant to sampling. The principal sources of information for the T3 NSFNET backbone come from programs using the Simple Network Management Protocol (SNMP) [70] for simple interface statistics, and specialized software packages for more comprehensive traffic characterization based on traffic type and source/destination. For the T1 backbone, Merit used a modified version of the NNStat [71] package for traffic characterization. Advanced Network Services (ANS) now performs the network operations center (NOC) services for the T3 NSFNET backbone, and designed the ARTS (ANSnet Router Traffic Statistics) package [72], for traffic characterization. It is these latter two tools for packet categorization, NNStat on the T1 backbone and ARTS on the T3 backbone, that rely on sampling for traffic characterization.

Although the packet categorization mechanism at each node differs on the two backbones, the backbone-wide centralized collection of the data is the same. Every fifteen minutes, the central agent at the NOC running the collection software queries each of the backbone nodes, which report and then reset their object counters. The collection host is an IBM RS/6000 at the ANS NOC, which during mid-February 1993 was collecting around 25 MB of ARTS traffic characterization data on a typical workday.[1]

When the collection mechanism maintains sophisticated aggregate objects, even dedicated processors can begin to suffer degradation in the quality of collection under high load. For example, during the early years of the T1 backbone, the utilization was not high enough to strain the capacity of this dedicated processor. By mid-1991, however, the discrepancies between the SNMP based traffic counts and those derived by means of NNStat had grown to a significant fraction of the total traffic count, as shown in figure 4.1. It became clear that the processor collecting the NNStat data was unable to keep up with the total nodal traffic flow. Note that because the SNMP statistics are incremented in the mainstream of packet forwarding, they are more reliable. It is the traffic categorization information, specifically the net matrix, protocol, and port data, which is subject to losses during periods of high utilization.

[1] On the T3 backbone, the packet categorization collection mechanism uses a more efficient binary format than that used on the T1 backbone.

Table 4.1: summary statistics for distributions of per-second packet and byte volume, and average packet size

Distribution	Min	25%	Median	75%	Max	Mean	StdDev
Monday, 22 March 1993 (1.636 million packets during hour)							
Packet arrivals (packets/s)	156	364	412	473	966	424.2	85.1
Byte arrivals (kB/s)	26.591	71.1	90.9	117.6	330.6	98.6	38.6
Mean per-sec packet size (bytes)	82	190	222	259	398	226.2	50.5

In September 1991, responding to concerns over the integrity of the data, the operator of the T1 NSFNET backbone (Merit, at that time) deployed a sampling technique which captures only one out of fifty packet headers for traffic characterization purposes. The result was a significant reduction in the discrepancies. We do not discuss the remaining discrepancy here; some of the gap derives from leaky networks as described in section 3.5.2. Although the sampling imposes a cost of inaccuracies of the traffic signatures, there is no longer complete loss of statistical information during periods of high utilization.

Because each T1 backbone node facility had a separate processor dedicated to statistics collection, the collection mechanism never imposed a burden on the packet forwarding capacity of the node, although heavy network utilization may have rendered the statistics collecting processor unable to capture all the traffic. In contrast, staging the sampled packets from the T3 switching cards to the main processor is integral to the forwarding process, and therefore may potentially impact the switching capacity.

Although the motivation is different for the T1 and T3 architectures, both statistics collection mechanisms force the consideration of sampling. Future gigabit network environments will only intensify the problem. As their load outstrip the ability of even dedicated statistics processors to monitor the traffic, sampling will become essential to the integrity of sophisticated data objects that can reflect network usage and behavior.

4.2 Measurement methodology

We now describe the environment in which we collected the data for our study. The nature of our investigation demands detailed insight into traffic behavior, which requires evaluating each packet traversing the environment. Because NSFNET backbone core nodes typically cannot support the collection of traces capturing all packets over long periods of time, we collected packet traces at a single entrance interface into the backbone. Specifically, we collected a 24-hour trace of packets sent from the SDSC environment to the NSFNET San Diego ENSS via the FDDI interface.[2] Such an environment is more conducive to traffic capture than many other points, while still aggregating a substantial degree of traffic. We did not investigate the traffic back in the reverse direction, from the NSFNET back through this ENSS, nor did we investigate the traffic in or out of the ENSS Ethernet interface.

We use an SGI for the data collection; we describe the collection procedure in detail in section 5.5, where we use it for several other packet traces. Our 24 hour trace started at shortly after 22:00PST on 22 March 1993. Of the 24 hours we created a subset of about one hour, from 13:00 to 14:00 on 23 March 1993, consisting of over 1.6 million packets and 650 megabytes. We then simulated various sampling algorithms on this one-hour trace. Table 4.1 quantifies the statistics of the per-second packet, byte, and mean packet size distributions for the data set. For the purposes of this study we attend to the IP layer only, not distinguishing among TCP, UDP or other higher layer protocols. Also note that even though we collect traffic on an FDDI ring, which has a Maximum Transmission Unit (MTU) packet size of 4352 bytes [93], most of this traffic must traverse a medium with a lower MTU, such as an Ethernet, as it travels from its source to the ENSS. Thus the maximum packet size reflects the 1500 byte MTU of this commonly used medium.

[2]Preliminary experiments for this study used data from the FIX-West interexchange point in Moffet Field, CA. The results of the two data sets were quite similar, but the ENSS data set we use here is more relevant to the current NSFNET statistics collection situation.

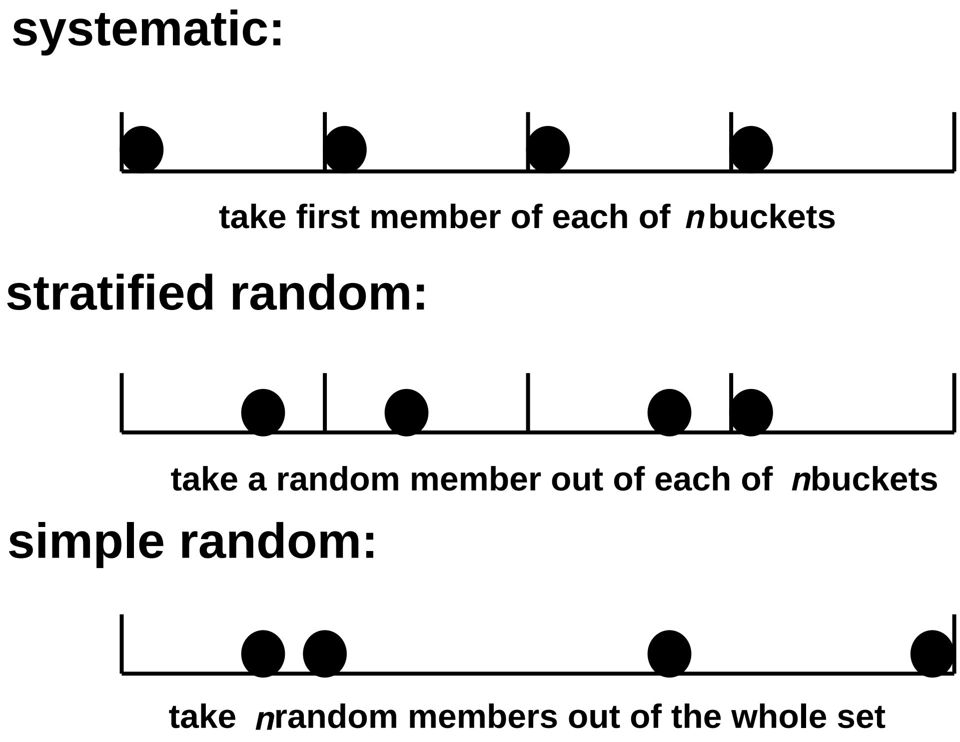

Figure 4.2: schematic of three sampling algorithms

4.3 Sampling mechanisms

Developing a sampling methodology requires an evaluation of the cost and benefit of sampling in the particular domain of study. Our goal is to evaluate the effects of certain sampling parameters on the integrity of the resulting samples. In general, a larger sample can more closely reflect the true parent population, but each instance of sampling imposes a cost, in terms of CPU time, buffer space, and sampling interval, or amount of calendar time one can devote to deriving a particular estimate. One must therefore weigh the sampling frequency against the accuracy requirements and complexity of a given object.

The one-hour packet trace we collected for our experiments represents only a brief interval, indeed itself a sample from the ongoing population of network traffic. For the purposes of our study we treat this packet trace as the true parent population, and the subpopulations drawn by our various sampling techniques as the samples. Standard statistical formulas generally rely on estimates of parameters of the parent population for the default case where the parent population is not known. Because we have access to the actual parameters of this parent population, we use them rather than estimates of them. Our goal is then to assess how close each sample is to its parent population for several key measurements.

Figure 4.2 illustrates an abstraction of the three main classes of sampling schemes we used in the study: systematic sampling; stratified random sampling; and simple random sampling. For each class, one can implement, or approximate, any particular method via either event-based or timer-based mechanisms. That is, one can use packet counts or timers to trigger the selection of a packet for inclusion in a sample. Implementing these methods at a variety of granularities allows a range of sampling fractions. Furthermore, one can vary the interval over which one samples: for a minute, 15 minutes, an hour, a day, etc. Since the processes are not time-homogeneous, it is not clear that spreading the same number of samples over longer intervals will generate the same result.

We briefly describe each method, first for packet-driven and then timer-driven implementations. The first class of methods, *systematic sampling* involves deterministically selecting every *kth* element (packet) of the data set. *Stratified random sampling* is similar to systematic sampling, except that rather than selecting the first packet from each bucket, a packet is selected randomly from each bucket. Although for both systematic and stratified random sampling the bucket sizes do not necessarily have to be constant, our experiments do use constant sizes. Finally, *simple random sampling* uniformly selects n packets from the total population at random.

Timer-driven sampling methods use a timer rather than a packet counter to trigger the selection of packets to include in the sample. Since our traffic analysis is based on packets as the atomic unit of data, timer-driven methods force a certain degree of approximation. As with granularities for packet-based sampling, one selects various time intervals to implement a desired range of sampling fractions. When the timer expires, we select the next packet to arrive. For example, in systematically sampling at the beginning of every n millisecond interval, there is no guarantee that a packet will arrive precisely at the beginning. We must take the packet immediately following the instant at which the interval begins, and then restart the interval from arrival timestamp of this sampled packet. The other two classes of timer-based sampling methods involve similar approximations.

Selecting the next packet to arrive after a randomly sampled time value will tend to miss bursty periods with many packets of relatively small interarrival times, and thus tends to skew the true interarrival distribution toward the larger values. Renewal theory provides a way to correct for this natural bias in timer-based sampling of a population counter-based distribution [94], which we will discuss in section 4.6.2.

We note an implementation detail with the last timer-based method, simple random timer sampling. For this sampling method, rather than having to select n sampling times uniformly distributed throughout the sampling interval, we take advantage of the fact that the distribution of Poisson arrivals in an interval $(0, t)$ is uniform [95]. We thus implement our simple random sampling using exponentially distributed intersample times with mean equal to the mean packet interarrival time multiplied by the reciprocal of the sampling factor. Our mean interarrival time was 2357.5 μseconds, so we distributed our simple random intersample times exponentially with means 2357.5×2, 2357.5×4, 2357.5×8, 2357.5×16, $\ldots$, 2357.5×32768.

We implemented all three of the classes of methods we describe, systematic, stratified, and simple random, for both packet and timer-based sampling, for a total of six basic methods. For each sampling method we selected, we were motivated by an interest in the effects of patterns in the data on the sampling results. We also wanted to determine whether the method currently employed for data collection on the T3 ANSnet backbone, systematically sampling every fiftieth packet, provided significantly different results from simple random sampling.

4.4 Methodological background

Cochran [96] and Krishnaiah and Rao [97] provide some comparative analyses of which sampling strategies offer lower variance under given conditions. These analyses use the variance of the estimate of the mean as a metric for the sampling method; the lower the expected variance of the estimate, the more *efficient* the sampling method. In our case we are more interested in assessment of the complete distribution. Nonetheless we offer some preliminary insights based on this evaluation method.

If the populations are randomly ordered, we expect all three methods (systematic, stratified, and random) to be equivalent. Systematic sampling spreads the samples more evenly over the population, which can potentially yield greater precision than stratified random sampling. In general, systematic sampling is more precise than simple random sampling if the variance within the systematic samples is larger than the population variance as a whole. If there is positive correlation between pairs of elements within the systematic sample, however, then stratified or simple random sampling will be more efficient. Periodic traffic behavior observed in Internet components [14] [12] [13] could present such positive correlation and lower the efficiency of systematic sampling.

For populations with a linear trend, stratified random sampling will be more efficient than systematic sampling. Intuitively, one can imagine how if the sample from the first bucket were too low, the sample from each subsequent bucket would also be too low. Stratified random sampling would alleviate this difficulty. Interestingly enough, simple random sampling is less efficient than either systematic or stratified random sampling in this situation [97].

4.4.1 Theoretical sample size for means

Cochran [96] provides a detailed explanation of the statistical determination of the appropriate random sample size for estimating a given parameter of a population, such as the mean or proportion. We provide an illustration of the appropriate sample sizes to estimate the mean for given confidence levels on the two metrics we selected as analysis targets. As an example we will specify an accuracy of $r = \mp 5\%$ and a confidence level of $100(1-\alpha)\% = 95\%$, which implies z-value of 1.96 in the following formula for the appropriate sample size n:

$$n = (\frac{100z\sigma}{r\mu})^2$$

where μ is the population mean and σ is the population standard deviation.

For our data set (of approximately 1.6 million packets), the packet size distribution had population mean $\mu = 232$ bytes and population standard deviation $\sigma = 236$. These values yield as the appropriate sample size: 1590. An accuracy of $r = 1\%$ would require 39,752 samples from the same data set. These formulas assume sampling from an infinite, though not necessarily normal, population, while we are actually using a population of about 1.6 million packets, of which 1,590 constitutes a sampling fraction of around 0.10%. Furthermore, the mean is not a particularly indicative description of the packet size distribution, which is bimodal around 40-byte and 552-byte packets.

For the interarrival time distribution of this data set the population mean is $\mu = 2358$ μsec and the standard deviation is $\sigma = 2734$. These values yield the appropriate sample sizes for 5% and 1% accuracy of 2066 and 51,644 packets, respectively.

4.4.2 Metrics of disparity between distributions

Since both of our characterization targets (and most others) come from distributions for which the mean is not such a helpful description, such estimates are of limited value to us. We seek a more sophisticated assessment of the various metrics, usually obtained through more comprehensive descriptions of the distribution.

Perhaps the best known metric is Pearson's χ^2 statistic, which compares the observed and expected counts within a set of bins which span the range of the data:

$$\chi^2 = \sum_{i=1}^{B} \frac{(O_i - E_i)^2}{E_i}$$

where B is the number of bins, O_i is the number of observations found in the ith bin of the sample, and E_i is the number of observations expected in the ith bin based on the parent population model. The sampling distribution of χ^2 is approximately the χ^2 distribution where the number of degrees of freedom equals the number of bins minus the number of independent parameters fitted minus one. This approximation improves as the number of counts in each cell increases, and is generally adequate if each cell has at least five expected counts. This statistic is the basis of the χ^2 test, which uses the χ^2 distribution to test hypotheses at specified significance levels about the goodness of fit between a model and a data set.

We performed χ^2 tests for our two target distributions on some of our samples varying several parameters. The results were remarkably compatible with statistical theory. For example, in our experiments for systematically sampling every fiftieth packet, only two or three out of the fifty possible replications produced χ^2 values that would convince a statistician to reject the hypothesis that they were produced by the original distribution at the 0.05 confidence level.

Unfortunately, the χ^2 statistic is sensitive to the size of the data set, making it difficult to compare samples of varying sizes. Therefore, it cannot quantify significant trends when varying the sampling fraction, one of our primary concerns. Goodman and Kruskal [98] note that although useful as a test for the significance of the association between two data sets, the χ^2 statistic, or any simple function of it (e.g., the significance level), cannot serve as a measure of *degree* of association between two sets. On the other hand, we did find significantly higher χ^2 values for the timer-based methods, which motivated us to drop them from the primary focus of our investigation as we describe in section 4.6.2. However, reasonable differentiation among the other methods was not possible with the traditional χ^2 goodness-of-fit testing methods.

Other sophisticated goodness-of-fit tests, such as the Kolmogorov-Smirnov [99] or Anderson-Darling A^2 [100] tests, have proven difficult to apply to wide-area network traffic data [61]. Another disparity metric, which we refer to as *cost*, measures the absolute distance, or l_1 norm, between the expected and observed bin counts:$\sum_{i=1}^{B} |O_i - E_i|$. Consider the following example use of the cost metric. Imagine a network service provider who uses traffic-based charging trying to convince his customers that sampling does not adversely affect their charges. He can offer to reimburse his customers for the difference between their real, if accessible, and observed, i.e., estimated via sampling, traffic. The provider would also like to avoid losing revenue through samples that underestimate the transmitted traffic. If X_i is the number of packets which the network provider attributes to his client based on his sampling, and Y_i is the number of packets which the client actually sent, then there are two possibilities:

- $X_i > Y_i$, in which case client i may express dissatisfaction at being overcharged, or
- $Y_i > X_i$, in which case the service provider loses earned revenue to client i

Note that the actual difference in the number of packets is important here, rather than metrics that compare the general shapes of distributions. Therefore, the provider should use a feasible sampling mechanism that minimizes the l_1 norm. By feasible we assume the comparison of sampling techniques with comparable cost. A service provider might also want a *relative cost* measure, for example the product of l_1 with the sampling fraction, to account for the resource savings of sampling less often.

All these metrics are still subject to the influence of the sample size. Fleiss [101] offers another alternative metric to measure the degree of similarity between two distributions which is free of the influence of the sample size: the ϕ *(*phi*) coefficient*. This metric is derived from the χ^2 metric as follows: $\phi = \sqrt{\frac{\chi^2}{n}}$, where $n = \sum_{i=1}^{B}(E_i + O_i)$. Unlike the χ^2 statistic, which uses the associated χ^2 distribution for hypothesis testing, we are aware of no such corresponding distribution for the ϕ metric.

Paxson [62] considers another χ^2-inspired metric which remains invariant with increasing sample sizes: $X^2 = \sum_{i=1}^{B} \frac{(O_i - E_i)^2}{(E_i)^2}$. If two distributions are different then for large data sets $\frac{O_i}{E_i}$ will approach some fixed factor ρ_i. If all bins have equal width this gives:

$$X^2 = \sum_{i=1}^{B}(\rho_i - 1)^2$$

which allows one to compute the "average normalized deviation" across all bins: $k = \sqrt{\frac{X^2}{B}}$.

In the next section we illustrate the application of several of these metrics with an example from our data set, and then select one metric to apply to our data from the domain of interest: a high speed wide area network.

4.5 Empirical evaluation

For the following example we use a single, approximately half-hour (2048 second) interval of packet trace data and sample at exponentially coarser granularities. Figure 4.3 plots as a function of sampling granularity, i.e., the inverse of the sampling fraction, the various metrics we described above which indicate the degree of disparity between the sample and the population: the χ^2 metric; the χ^2 significance level (for ease of comparison we plot (1 - the significance level) in the figure); the *cost* and *relative cost (rcost)* metrics; the X^2 metric; and the ϕ metric. Each metric in the figure attempts to measure the goodness of fit of a model to a data set, where in this case our subsamples are the model of the original (in the real world, unknown) data set. According to this figure the *cost*, X^2, and ϕ metrics all exhibit similar behavior. Because the ϕ metric is well established in the statistical literature for comparison across bins with different expected counts, we chose this metric for use in our investigation.

We will present results in terms of the range of ϕ-values for a given analysis target, and how these ϕ-values change as we vary one dimension of the parameter space holding the other dimensions constant. A ϕ-value of 0 is consistent with a sample which perfectly reflects the parent population. In general, larger ϕ-values will correspond to poorer samples, i.e., those that diverge more widely from the sampled population. When a network operator selects a sampling method, with an associated sampling fraction and interval, he buys a certain range of ϕ-values which will characterize his samples. Although we do not offer a precise threshold below which all ϕ-values are acceptable, we do offer suggestions for how the ϕ-value scale can guide a sampling methodology.

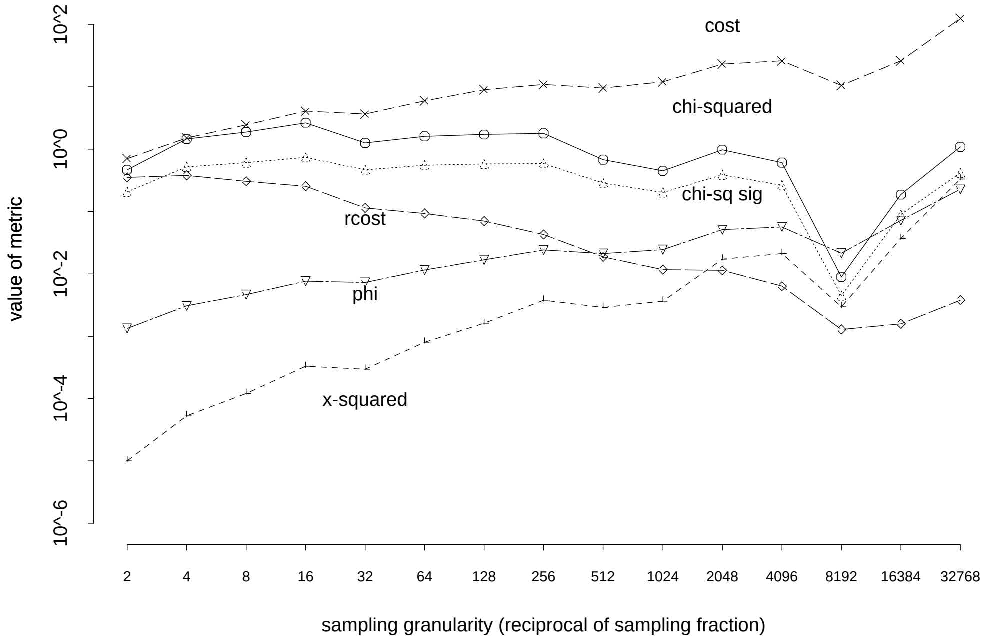

Figure 4.3: various metrics of disparity for samples as a function of exponentially increasing sampling granularities

A complication in our experiment is the fact that some of the samples share members with other samples, and thus there is correlation among the samples. This correlation inhibits statistically precise statements about the superiority of one sampling method over another. On the other hand this approach does allow us to easily order sampling methods based on their performance.

4.6 Application of methodology

Now that we have presented our methodology for scoring the samples for each target, we concentrate on the effects of the various sampling parameters in isolation. Our experiment consists of a large number of samples exploring the domain based on:

1. class of sampling method (systematic, stratified random, simple random)
2. time-driven versus event-driven methods
3. granularity, or sampling fraction
4. the interval, or length of time over which we sample

The first two dimensions cover the range of sampling methods which we employ. The latter two dimensions allow further subdivisions to the parameter space. We ran five replications for each method to avoid misleading outlying samples.

We apply our evaluation methodology to the analysis of two distributions: packet size and interarrival times. We show samples which reflect the true population to varying degrees. We then provide graphs which show the effect of varying a single parameter on the range of scores. The objective is to provide a framework for evaluating what count as good or bad ϕ-value scores, and to demonstrate our analysis for the selected targets.

In our sampling simulations we use an exponentially increasing time window relative to the beginning of the hour-long trace. To modulate both the time windows as well as the sampling interval, we also ran samples at exponentially decreasing sampling fractions, starting at every other packet, and decreasing the fraction down to one in 32,768 packets. We then binned the interarrival time and packet length distributions for use in our χ^2 based statistic calculation, as we describe below.

Table 4.2: summary statistics for distribution of packet sizes and interarrival times

Min.	**5%**	**25%**	**Median**	**75%**	**95%**	**Max.**	**Mean**	**Std.Dev.**
Total Population = 1.63 million packets								
packet size								
28	40	40	76	552	552	1500	232	236
packet interarrival times								
< 400	< 400	400	1600	3200	7600	49600	2358	2734

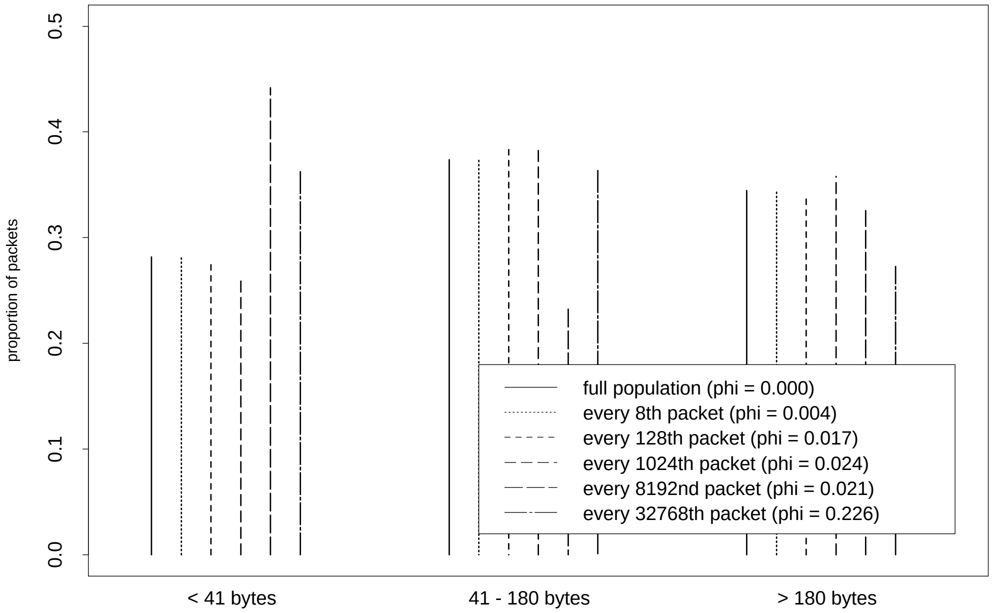

Figure 4.4: distribution of packet sizes for five samples at different granularities (1024 second interval, systematic sampling)

4.6.1 Bin selection

Calculation of the χ^2-based metric that scores our individual samples requires the selection of bins, or ranges, in which to group the data sets. In this section we present the ranges that we used for our two targets, and histograms which illustrate the distributions over these ranges. Table 4.2 provides summary statistics for the full population for both the packet size and interarrival time distributions.

Packet size distribution

To compare the packet size distributions, i.e., the number of bytes per packet, we compared the proportion of packets within the following three ranges (in bytes): less than 41; between 41 and 180; and greater than 180. We chose these bins based on our knowledge of the typical packet size distribution of network traffic. We experimented with bin sizes which accounted for a fairly large number of packets, and also which characterize certain protocols: ACKs, character echos, transaction-oriented, bulk transfer. Figure 4.4 compares the distribution of packet sizes into these bins at five sampling granularities.

Interarrival time distribution

For the packet interarrival time distribution, we used the following bins (in μsec): less than 800 μs; between 800 and 1199 μs; between 1200 and 2399 μs; between 2400 and 3599 μs; and greater than 3600 μs. We chose these bins to achieve a relatively even distribution of data among them. Figure 4.5 shows a

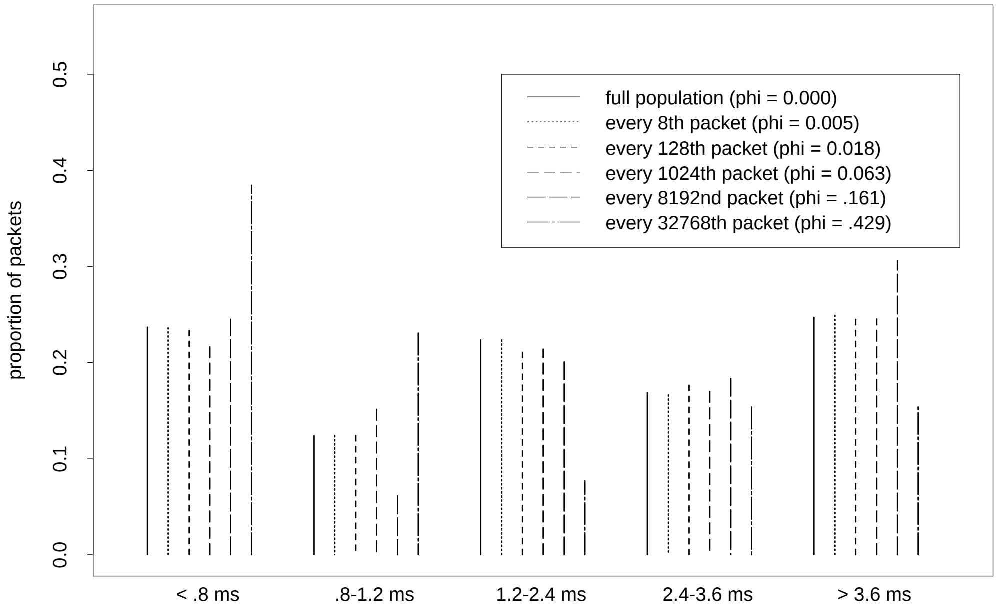

Figure 4.5: distribution of packet interarrival times for five systematic samples at different granularities (1024 second interval)

histogram of several samples of packet interarrival times dividing them into these ranges. The increasing ϕ-value scores shown in the legend reflect the divergence in the sample accuracy as the sampling fraction decreases. We discuss these scores in detail in the next section. Table 4.2 summarizes the parameters of the full hour packet population, subject to the 400 microsecond clock granularity described in section 4.2.

4.6.2 Sampling fraction and method

Using these bins to base our scoring, we investigated the variation of individual sampling parameters. To examine the effect of the sampling fraction, we first focused on one method, systematic sampling, and ran several replications of this method at a range of sampling fractions. To achieve a wider range of replications for systematic samples, we varied the point within the data set at which to begin the sampling procedure. The boxplots[3] in figure 4.6 show the range of ϕ-value scores for each systematic sample for the packet size distribution assessment. The x-axis corresponds to the sampling granularity, or the reciprocal of the sampling fraction. The first box plot on the left corresponds to every fourth packet, and most of the scores are near perfect zeros. The figure shows two clear effects of decreasing the sampling fraction, and holds with other methods as well: increasing values, which indicate poorer snapshots of the parent population; and increasing variance within the set of samples for each method. Figure 4.7 shows the means of the boxplots in figure 4.6.

To illustrate the effect of the sampling method, we used the six methods in our experiment to assess the packet size distribution. Like the boxplots in figure 4.6, figure 4.8 indicates the effect of increasing the sampling fraction on the ability of the sample to estimate the true population. The timer-based methods are notably worse at assessing the packet size distribution, due to renewal theoretic considerations mentioned above, i.e., sampled packets are those that follow larger interarrival times. The timer-based methods are particularly poor at smaller sampling granularities. Essentially, when we use only a few multiples of the mean packet interarrival time as the intersampling time, we will find many intervals in which there was no packet at all to sample, which leaves us with fewer total packets in the sample. Furthermore, the packets we do capture in a sample will be less representative of the population, since we miss many of the bursts that compensate for the empty intervals. Because we can quantify no clear relationship between packet size and interarrival time, it is difficult to correct for this discrepancy in our analysis. Renewal theory does provide

[3] In a boxplot, the dotted lines (or "whiskers") from the bottom to the top of the box, extend to the extreme values of data or 1.5 times the interquartile difference from the center, whichever is less.

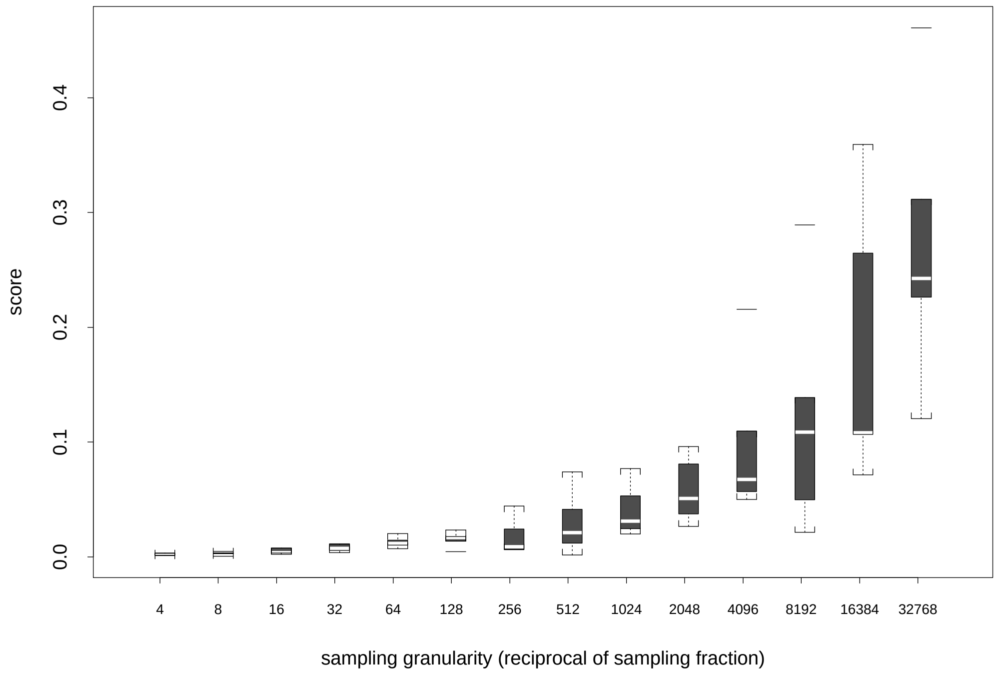

Figure 4.6: ranges of systematic sampling ϕ-value scores for packet size distribution as a function of sampling granularity for 1024 second interval

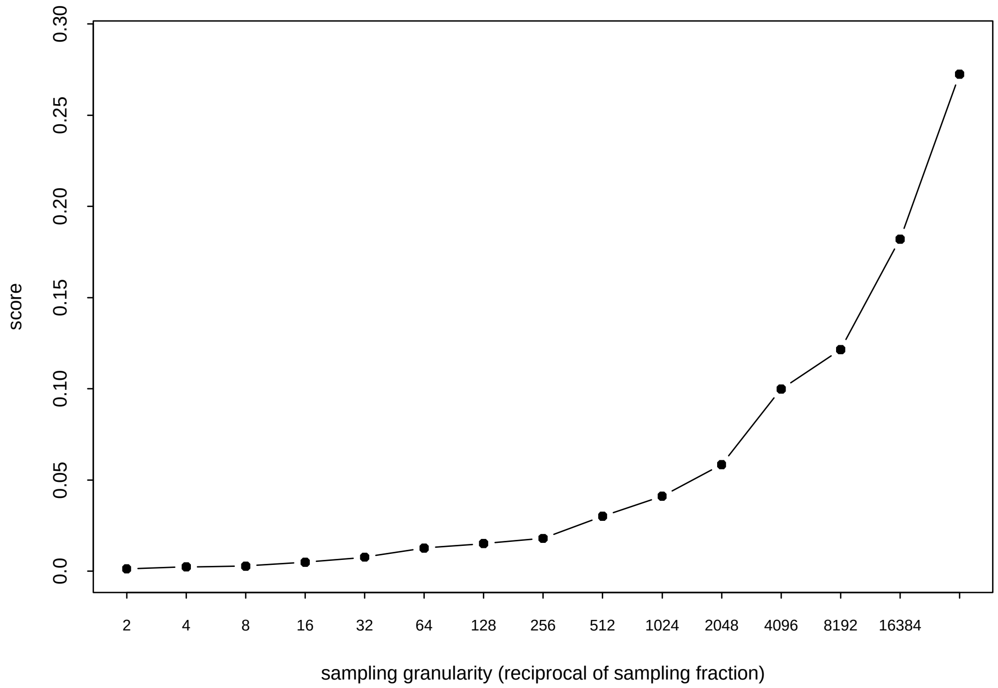

Figure 4.7: means of systematic sampling ϕ-value scores for packet size distribution as a function of sampling granularity for 1024 second interval

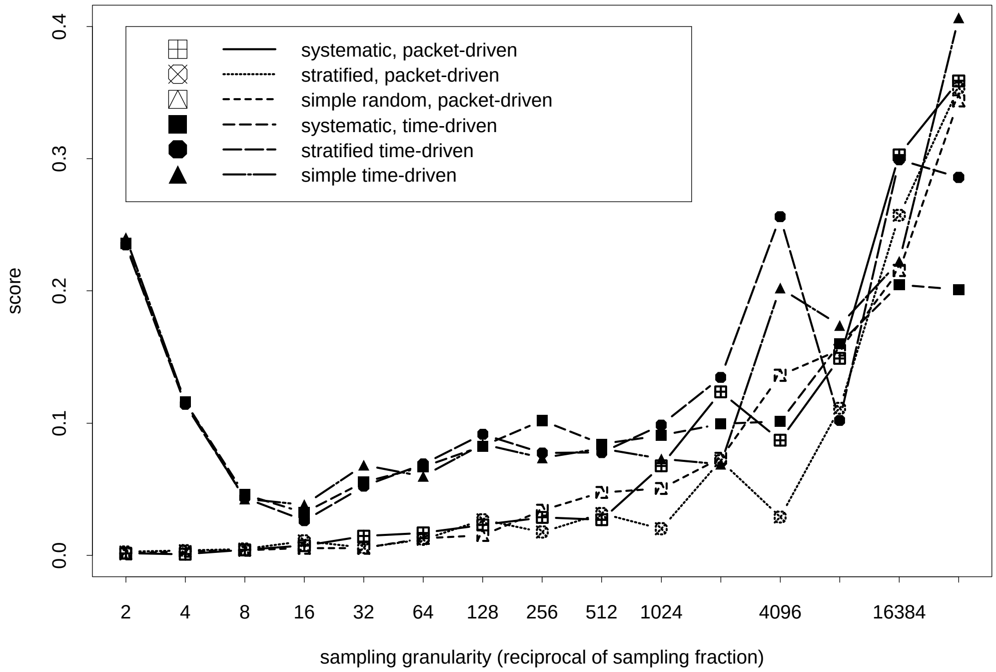

Figure 4.8: mean sample ϕ-value scores as a function of sampling granularity for packet size distribution

a way to correct for the discrepancy in timer-based samples of the interarrival time distribution, which we illustrate next.

Figure 4.9 illustrates the same metric for the packet interarrival time distribution. As we have discussed, timer-based sampling methods are at a disadvantage in assessing interarrival times, since one tends to miss bursty periods with many packets of relatively small interarrival times, skewing the sampled interarrival distribution toward the larger values. Renewal theory provides a way to correct for this natural bias in timer-based sampling of a population or counter-based distribution [94],

$$f_{adj\_obs}(x) = \frac{\bar{x} f_{obs}(x)}{x} \tag{4.1}$$

where $f_{obs}(x)$ is the observed distribution, $\bar{x}$ is the mean of x, and $f_{adj\_obs}(x)$ is the adjusted density of the observed interval. In our analysis we are using a set of bins that span the range of data rather than a continuous distribution, so we must use the corresponding renewal-theoretic relationship for a counting process:

$$P_{adj\_obs}(i) = \frac{\bar{n} P_{obs}(i)}{n} \tag{4.2}$$

where $P_{obs}(i)$ is the observed count in bin i, n is the midpoint of bucket i, $\bar{n}$ is the population mean, and and $P_{adj\_obs}(i)$ is the adjusted count in bin i.

Figure 4.9 incorporates this correction for the timer-based samples, which does improve their ϕ-value scores, but they still exhibit poorer performance than the packet-based sampling methods. In addition, this adjustment method requires constant assessment of the mean of the true population, as well as approximation using the midpoints of the buckets of the distribution, making it suboptimal for operational use. Timer-based sampling also requires clock hardware that may not be on the forwarding node or subsystem. For these reasons, and because figures 4.8 and 4.9 show that timer-based sampling methods are uniformly worse in assessing these two targets anyway, we do not devote any further attention to timer-based methods. Furthermore, because there is little difference even among the packet based methods, we focus the remainder of our discussion on only one of them, systematic random sampling.

4.6.3 Length of interval

We have investigated how the sampling fraction affects the size of the sample; another way to increase the sample size, and thus allow greater accuracy in any desired estimate of the parent population, is

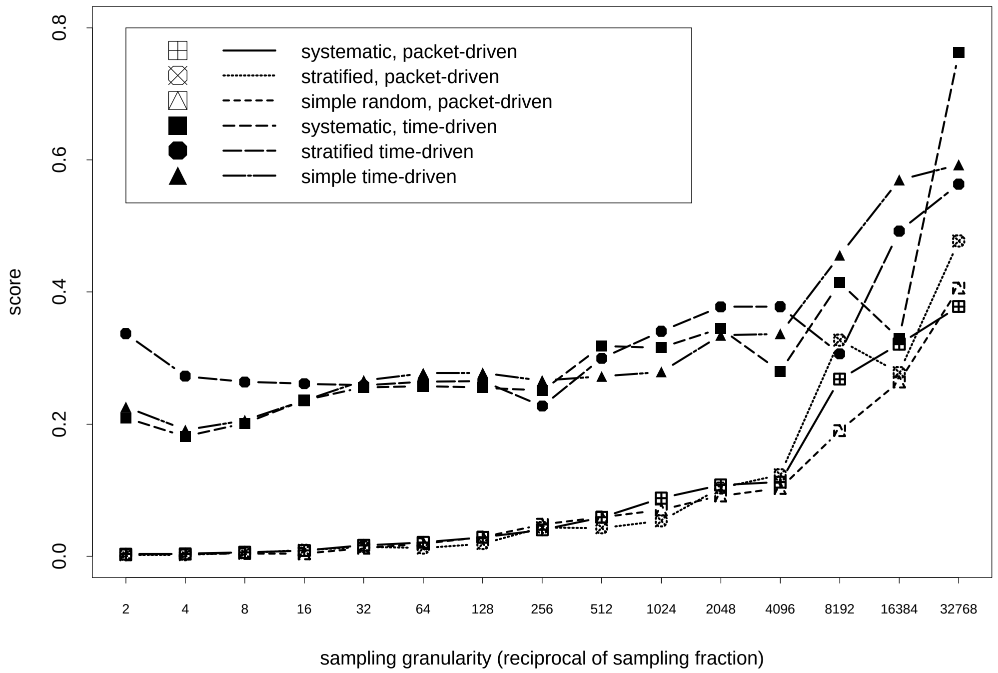

Figure 4.9: mean sample ϕ-value scores as a function of sampling granularity for packet interarrival time distribution

the duration of the sampling interval. However, network traffic is typically non-stationary, and so the effect of spreading the sampled packets over a longer interval is not clear. Previous studies indicate that packet arrivals are not independently distributed through time [28] [5] [6] [63] [64] [24], and so it is likely that 100 packets from a 10 second interval will not provide the same estimate as 100 packets spaced over a 1 minute interval, even when differing utilization may render the number of packets in the two parent populations equal.

In order to experiment with the effect of the interval on assessing a particular distribution over that sampling interval, we chose one particular sampling method, systematic sampling, and varied the interval during different runs of that method. Figures 4.10 and 4.11 show the resulting ϕ-value scores for the packet size and interarrival time distributions for our data set. Although the left side of these figures reflect smaller time intervals and are noisier, one can see general trends at later intervals. For all sampling fractions the sampling scores improve with elapsed time, as one might expect.

4.7 Conclusions

We have presented a framework for the empirical evaluation of sampling techniques for network traffic characterization. We have then applied our methodology to two target metrics: distribution of packet sizes, and distribution of packet interarrival times.

Our experimental data consisted of a packet trace obtained from an entrance interface into the NSFNET national backbone. Because the characteristics of our populations of network data do not fit into any categories analyzed in the literature, we offer in this paper empirical data on sampling simulations run on an isolated packet trace while controlling various experimental parameters.

We have applied the traditional χ^2 test to evaluate the goodness of fit of the sampled distribution to the original complete distribution. One important result is that the current technique of systematic sampling used for statistics collection on the NSFNET backbone provides samples that are compatible with the original distribution of packet sizes and interarrival times at the 0.05 significance level.

Because the χ^2 technique is sensitive to the sample size, which result from varying the sampling fraction, it is inappropriate for comparison of samples of different sizes. We have focused our evaluation instead on the ϕ metric which measures similar deviation but is not sensitive to the size of the sample. The ϕ metric characterizes the degree of association between the sample distributions and the population, but

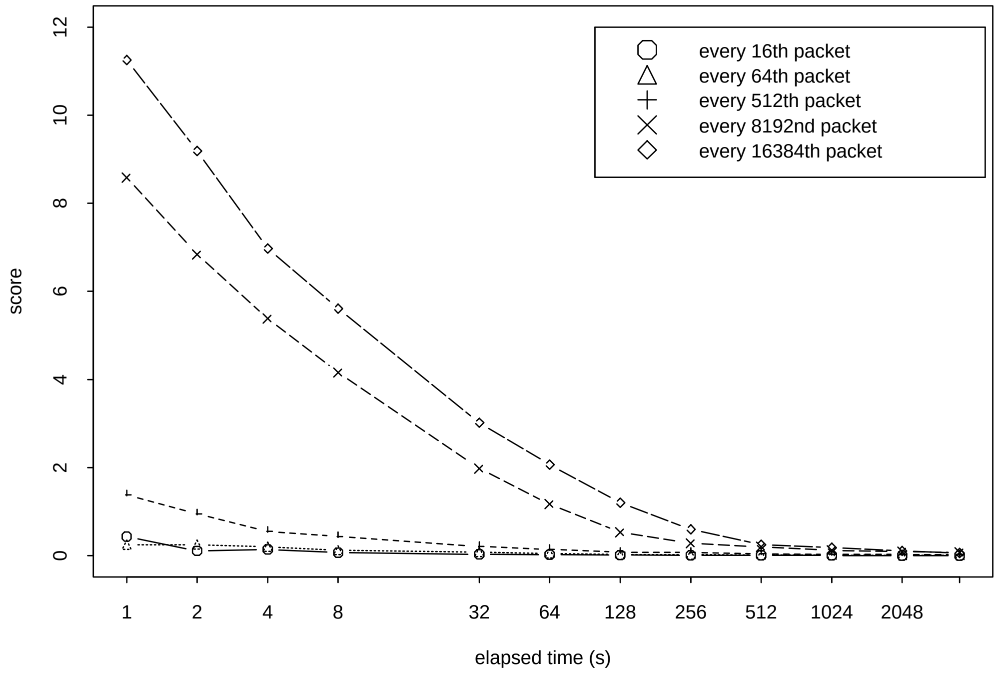

Figure 4.10: mean systematic sample ϕ-value scores for packet size distribution as a function of elapsed time (in seconds)

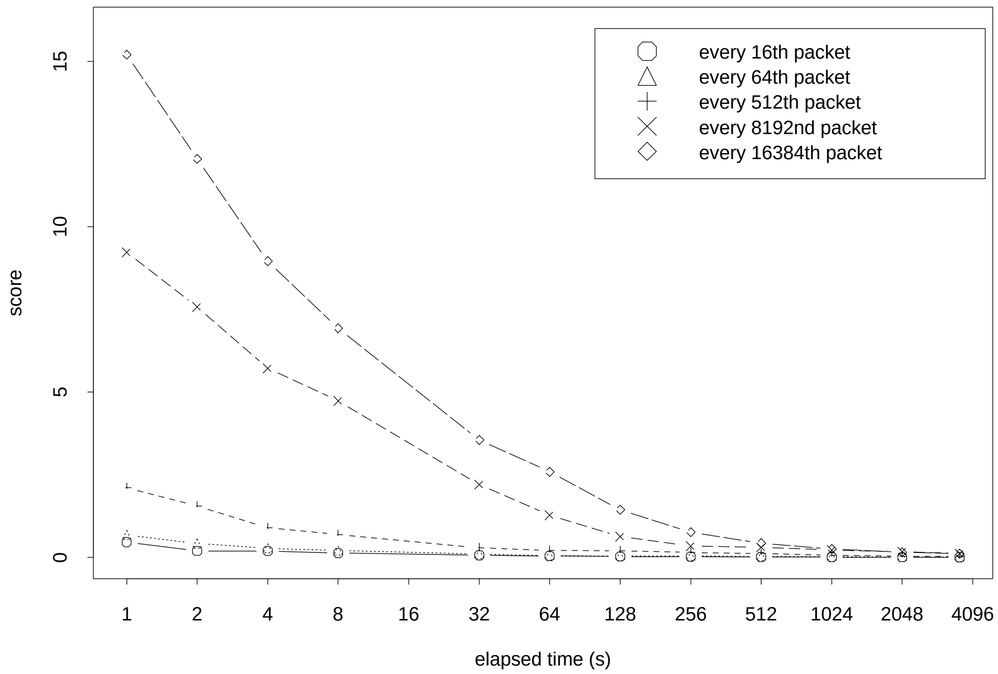

Figure 4.11: mean systematic sample ϕ-value scores for packet interarrival time distribution as a function of elapsed time (in seconds)

it does not provide absolute characterizations of sampling performance, and is not conducive to rigorous hypothesis testing. However, it is a useful tool to demonstrate that a technique is generally superior to another across sampling fractions and sampling intervals.

Based on this metric, we have considered systematic, stratified random, and random sampling by packet or time and various sampling fractions and sampling intervals. Our results revealed that the time-triggered techniques did not perform as well as the packet-triggered ones. Furthermore, the performance differences within each class, i.e., packet-based or time-based techniques, are small. Our methodology can be extended and applied to characterizations of network traffic that are based on proportions, e.g., TCP/UDP port distribution. More difficult would be to characterize the goodness of fit of the sampled source-destination traffic matrix, mainly because of its large size and because many traffic pairs generate small amounts of traffic during typical sampling intervals.

Chapter 5

Internet traffic flow profiling

...a better building block than the datagram for the next generation of architecture. The general characteristic of this building block is that it would identify a sequence of packets traveling from the source to the destination, without assuming any particular type of service... I have used the word 'flow' to characterize this building block. It would be necessary for the gateways to have flow state in order to remember the nature of the flows which are passing through them, but the state information would not be critical in maintaining the desired type of service associated with the flow.

– *David Clark,* "The design philosophy of the DARPA Internet protocols", SIGCOMM '88

But glory doesn't mean 'a nice knockdown argument', Alice objected;
'When I use a word,' Humpty Dumpty said, in rather a scornful tone,
'it means just what I choose it to mean – neither more nor less.'
'The question is', said Alice, 'whether you can *make words mean so many different things.'*
'The question is,' said Humpty Dumpty, 'which is to be master – that's all.'

– *Lewis Carroll,* Through the Looking Glass

In this chapter we present a methodology for profiling Internet traffic flows. Our methodology allows us to address some of the network centric characterization tasks in our taxonomy that the operational statistics covered in Chapter 3 do not address well. In the two chapters following this one we apply the methodology we describe to several strategically selected environments within the Internet hierarchy. In Chapter 6 we present metrics of individual flows, including the cross section of flows on the network, and their payload and duration. In Chapter 7 we present metrics of the aggregate flow, including flow counts, flow interarrival times, and flow locality indicators. The methodology we offer can form a complementary component to existing operational statistics collection, adding significant insight into traffic characteristics that current methods do not allow.

5.1 Introduction

During the early years of the Internet, one could characterize the traffic as clearly defined interconnections of hosts to a common network, the ARPAnet. Deployment of the NSFNET in the mid-1980's brought a more hierarchical architecture, as described in section 2.1, intended to support a national backbone network interconnecting mid-level networks of regional scope, which themselves support connectivity to campuses. Far more successful than anyone imagined, the NSFNET spawned connectivity well beyond the auspices of its original charter, acting as a catalyst to interoperability of networks in the research, educational, commercial, and government arenas across the globe. The resulting international fabric can transport millions of traffic flows simultaneously, aggregating traffic among many end systems, users, and applications. However, in the process the strict hierarchical architecture has given way to the flexibility of what has become a complex web of global interconnection. Although certain loose hierarchical components still exist within the global mesh, such as the U.S. NSFNET depicted in figure 5.2, the general case is quite arbitrary. Within the weave, individual Internet flows exhibit a variety of structure, and their transience and diversity frustrate attempts to model, or even define, them. Yet characterizing their nature will be critical to accommodating their increasing number and diversity.

Our methodology for modeling flows differs from previous studies that have concentrated on end-point definitions of TCP flows, such as by the SYN and FIN control mechanism of the TCP protocol.[1] The strength of a TCP SYN/FIN based approach is that determining the beginning and end of a connection based flow is relatively easy. However several other factors, described in section 5.3.4, motivate an alternative approach. Most notably, not all flows use transport layer protocols that support SYN and FIN functionality. In order to maintain generality across all traffic, we do not consider higher layer connections imposed by end systems, but rather define flows based on traffic satisfying various temporal and spatial locality conditions, as observed at internal points of the network. That is, we ground the definition of an IP traffic flow only in the appearance of packets within a given time interval to, from, or between entities, as perceived at a given network measurement point. This approach to the definition and characterization of network flows can address some central Internet problems, including route caching, resource reservation at multiple service levels, usage based accounting, and the transport of IP traffic over an ATM fabric.

The methodology we describe in this chapter structures the metrics we present in the next two chapters, where we apply the proposed methodology to measurements from a range of locations in the Internet fabric. In section 5.2 we give background on previous flow models, and in section 5.3 we discuss several aspects of flow structure before we formally define a flow in section 5.4. Section 5.5 describes the environments in which we apply our methodology, and section 5.6 describes the trace-driven simulation procedure that allows us to characterize flow workload and state information from a specified transit point within the network. Our metrics fall into two categories: metrics of individual flows, and metrics of the aggregate traffic flow. In Chapter 6 we focus on metrics of individual flows, including flow volume, in packets and bytes, and flow duration. In Chapter 7 we present aggregate flow metrics, as seen from the network perspective, which include counts of the number of new and active flows per time interval, flow interarrival times, and indicators of flow locality. Applying the methodology to our measurements yields significant observations of the Internet infrastructure, with implications for performance requirements of routers at Internet hotspots, general and specialized flow-based resource reservation algorithms, future usage-based accounting requirements, and traffic prioritization.

5.2 Previous flow models

One of the original models of a flow was inspired by the objective of characterizing network traffic *locality*. Locality describes the phenomenon of a non-uniform distribution of traffic to, from, or among a select few sites [65]. Network locality presents an obstacle to characterizing packet arrival processes as either Poisson or compound Poisson. As an alternative, Jain offered the *packet train* model of packet arrivals to describe traffic on a token ring local area network at MIT [28]. He defined a *packet train* as a burst of packets arriving from the same source and heading to the same destination. If the spacing between two packets exceeds some inter-train gap, they are said to belong to separate trains. Jain describes the packet train model as a general model of which the Poisson and compound Poisson models are special cases. In his model, the intertrain time is a user parameter, dependent on the frequency with which applications use the network. The intercar interval for a train is a system parameter and depends on network hardware and software. In Poisson arrival models, the intertrain and intercar interval parameters are merged to give a single parameter: mean interrival time. Compound Poisson models use an exponential inter-train interval distribution and a zero inter-car interval.

[1] The SYN packet serves to synchronize packet sequence numbers during the opening of a TCP connection. The FIN packet serves to clear the connection.

The packet train model reflects the fact that much of network communication involves many packets spaced closely in time between the same two endpoints. Request/response applications will accordingly yield bidirectional packet trains.

Another suggested definition of a flow derives from a different motivation: the need for service functionality inherently incongruous with the datagram architecture of the Internet. Clark [102] proposes:

> ...a better building block than the datagram for the next generation of architecture. The general characteristic of this building block is that it would identify a sequence of packets traveling from the source to the destination, without assuming any particular type of service... I have used the word "flow" to characterize this building block. It would be necessary for the gateways to have flow state in order to remember the nature of the flows which are passing through them, but the state information would not be critical in maintaining the desired type of service associated with the flow.

Clark refers to this concept of *soft state* as a potential method of achieving the "goals of survivability and flexibility, while at the same time doing a better job of dealing with the issue of resource management and accountability" [102].

The two motivating factors, characterizing traffic locality and supporting special service capabilities, are not unrelated. The ability of Internet equipment to maintain soft state will be directly related to the number and intensity of network flows, the nature of service they require, and their distribution in geographic space, or locality characteristics. Understanding the effect of the packet train phenomena on router behavior, and in turn how router behavior may intensify packet train phenomena [14] [15], will be essential to optimizing router efficiency. Although many have applied the packet train model of flows to the transport or application layers [35] [33] [34] or focused only on TCP traffic flows [6] [31], we offer a comprehensive methodology of timeout-based flow characterization at the IP layer for use in datagram environments. We incorporate the effect of the range of several flow parameters, such as the flow granularity and timeout, in a variety of environments.

5.3 Flow aspects

In this section we describe four aspects of a flow and how they structure flow measurement and subsequent analysis.

5.3.1 Directionality

First, one can define a flow as unidirectional or bidirectional. Connection-oriented TCP traffic generally exhibits bidirectionality: each flow from A to B also generates a flow from B to A, at the very least for acknowledgement packets. New multimedia applications will likely bring more unidirectional flows, e.g., non-interactive audio or video streams, perhaps using UDP and not requiring acknowledgements from the receiving end of the real-time data. Multicasting facilities, which require that routers only replicate data without feeding back status information from possibly many remote locations, further drives the popularity of such flows.

In this study we define flows as unidirectional, i.e., traffic between A and B would show up as two separate flows: traffic from A to B, and traffic from B to A. While the effect of bidirectionality of flows is generally important to investigate, unidirectional flows are more relevant to the issues that motivate us, among them routing optimization, accounting, and multiplexing IP on top of an ATM substrate [103] [104] [105].

5.3.2 One versus two endpoints

The second aspect of a flow is related to the first. We distinguish between single and double endpoint flows, that is, flows aggregated at the source or the destination of the traffic, versus flows defined by both the source plus the destination. An example is the difference between all traffic to the same destination network number, versus all traffic from the same network number and also to the same network number, that is, with a common network number pair.

Note that the directionality aspect mentioned above constrains this aspect, in that it only makes sense to talk about bidirectionality of two-endpoint flows. Single endpoint flows, which must specify either a source or a destination, are inherently unidirectional.

In this study we explore both single and two endpoint flows. The single sided definitions specify the source or destination host or network number, while the two-sided definitions use IP network number pairs, host pairs, or process identifiers, consisting of source and destination hosts, as well as source and destination application identifiers, such as UDP/TCP port numbers.

5.3.3 Granularity

The third aspect of a flow is the granularity, or the extent of the communicating entities. Potential granularities include traffic by application, end user, host, IP network number, Administrative Domain, backbone client service provider, external interface of a backbone node, backbone node, single backbone at large, or multibackbone environment (e.g., of different agencies). These granularities do not necessarily have an inherent order, as a single user or application might straddle several hosts or even several network numbers. One example flow granularity of interest derives from the fact that IP routers make forwarding decisions based on routing tables which contain next-hop information for a given destination network, a task implicitly grounded in one-sided destination network layer flows at the granularity of IP network number. Eventually, as policy routing issues render the source as well as the destination of a packet relevant to routing decisions, the issue of two-sided flow assessment will also become important. Furthermore, as new routing mechanisms utilize alternative hierarchical definitions related to IP network numbers (e.g., CIDR masks [87]), the desired granularity may evolve somewhat.

Network service providers may want to define coarser-grained flows as network number pairs for which they will create virtual circuits crossing their network, e.g., ATM cloud, each of which will bundle many finer-grained IP flows. Conversely, a more detailed granularity would be necessary for providing special service to a single instance of an application, e.g., a videoconference. Network equipment would have to keep track of state information about such flows to meet their service expectations, but also to account for their network usage. Although neither state maintenance strategies nor usage-based flow accounting currently exist in most Internet environments, we expect that a stable infrastructure supporting the former will inevitably require the latter, although the finest granularity necessary for state maintenance, and how to implement it, is not yet clear.

These examples illustrate the importance of flexibility of a flow model. We emphasize the need to ground a flow specification in the *requirements of the network*, and allow at any point in the network for multiple simultaneous requirements to assess flows, e.g., destination network address for routing, particularly for caching considerations, process pair for accounting, source address for accounting and policy routing, destination address or host or network address pair for bundling flows across ATM virtual circuits, address plus precedence information for flows at multiple priority levels. For our investigation we selected the granularities of network and host, and to highlight certain issues we use address/port quadruples.

5.3.4 Protocol layer

Finally, there is the functional, or protocol, layer of the network flow. For example, one could define flows at the application layer via the end hosts. Alternatively one could use the transport connection, e.g., via SYN and FIN packets of the TCP protocol which support explicit connection setup and teardown. In order to maintain generality across all traffic, we do not consider higher layer connections imposed by end systems, but rather base the definition on packet transmission activity between specified endpoints at the network layer. Such a flow definition will not have a one-to-one mapping to active TCP connections; a single flow could potentially include multiple active TCP connections, or an idle TCP connection may be contained in multiple flows over time. TCP traffic may furthermore be interleaved with UDP traffic, or a flow may consist entirely of non-TCP traffic.

Four factors motivate our decision to restrict ourselves to the IP layer. First, because the Internet is a connectionless datagram environment, dependence on connection-oriented information will often interfere with operational stability. If routes change during a flow, new routers will carry datagrams that never saw the transport layer SYN/SYN-ACK packets, and routers that did see earlier datagrams in a flow will never see the FIN/FIN-ACKs. State information that is dependent on this data will be vestigial in some cases and unavailable in others. Fragmentation will also pose a problem, since all but the first fragment lack the TCP/UDP port information, making it impossible to track fragmented packets to a higher layer flow. An additional constraint is that many wide area environments of today would have to rely on sampling for long-term flow assessment, in which case SYN/FIN requirements would lead to "losing" flows, including high volume ones.

Second, routing and accounting in a distributed datagram network must occur outside of transport level connections, and management of them must thus be independent of explicit transport connections.

Third, not all flows use transport layer protocols that support SYN and FIN functionality, and we note a trend toward alternative, lightweight transport mechanisms which lack explicit connection setup and teardown mechanisms[2], implying the need to rely only on IP layer information to define a network flow. A recent proposal for flow labels [106] highlights the need for end systems to identify and recognize "a sequence of packets sent from a particular source to a particular (unicast or multicast) destination that require special handling by the intervening routers." A flow label field would obviate the need for the strict connect-data-disconnect phases (SYN/FIN functionality of TCP), since each packet, or every nth packet, or periodic control packets (e.g., RSVP) can establish flow state. Flow labels can also reflect a coarser grain, e.g., aggregating multiple transport connections, or a finer grain, e.g., to support different qualities of service for different packets within a single transport connection, such as with a videoconference where loss of audio is less desirable than of video. Since the flow label assumes no router intelligence regarding the endpoint addresses and TCP/UDP port quadruples, it allows improved switching time.

Finally, new technologies for link level routers, e.g., ATM, will not have access to transport layer information; any Internet related transmission decisions will have to rely only on IP level information. In particular, until end-to-end ATM is a reality, IP gateways attached to ATM style networks will have to multiplex possibly many IP flows onto the ATM substrate. Mapping higher level (IP) flows to underlying link level virtual circuits [103] [104] [105] [107] will require effective setup, maintenance, timeout strategies, and accounting schemes. One requirement will be graceful handling of idle connections, regardless of transport layer behavior. Service providers may charge for these idle connections, such as in an ATM network, and terminating and restarting them as needed would be more efficient [108]. Transport layer flow assessment will not be adequate to determine whether to set up a new virtual circuit as a new flow arrives, or place the traffic into cells on an existing ATM virtual circuit between two end points of the ATM substrate.

In fact our measurements indicate that the highest potential burden for an ATM router will likely come from flows not amenable to assessment at the transport layer. Network layer flow assessment will thus be essential to graceful adaptation between the IP and ATM layers.

5.4 Flow definition

The four aspects of a flow model – directionality, one-sided vs. two-sided, granularity, and functional layer – structure the selection of a *flow specification*. For the first aspect, we chose to examine only unidirectional flows. For the second aspect, we examine both single and two-sided flows. For the third aspect, we study the flow granularities of network number, host, and in some cases address/port quadruples. Finally, we ground flows within the network layer. Because there is no information delineating the beginning or end of a flow at the network layer, we ground our model of a flow in the actual traffic activity from one or both of their transmission endpoints as perceived at a given network measurement point. A flow is *active* as long as packets meeting the flow criteria, at the granularity considered, are separated in time by less than a specified timeout value, as figure 5.1 illustrates. The lower half of the figure depicts multiple independent flows, of which thousands may exist simultaneously at wide area transit points.

Our definition of *timeout* is similar to that used in other definitions of packet trains [6] [28] [31] [34] [60], although most studies adopt an ad hoc timeout value rather than investigating the effect of varying it across a range as we do. Jain originally selected for his study of local network traffic a timeout of 500

[2] Such lightweight protocols are particularly appropriate for continuous media streams such as audio or video.

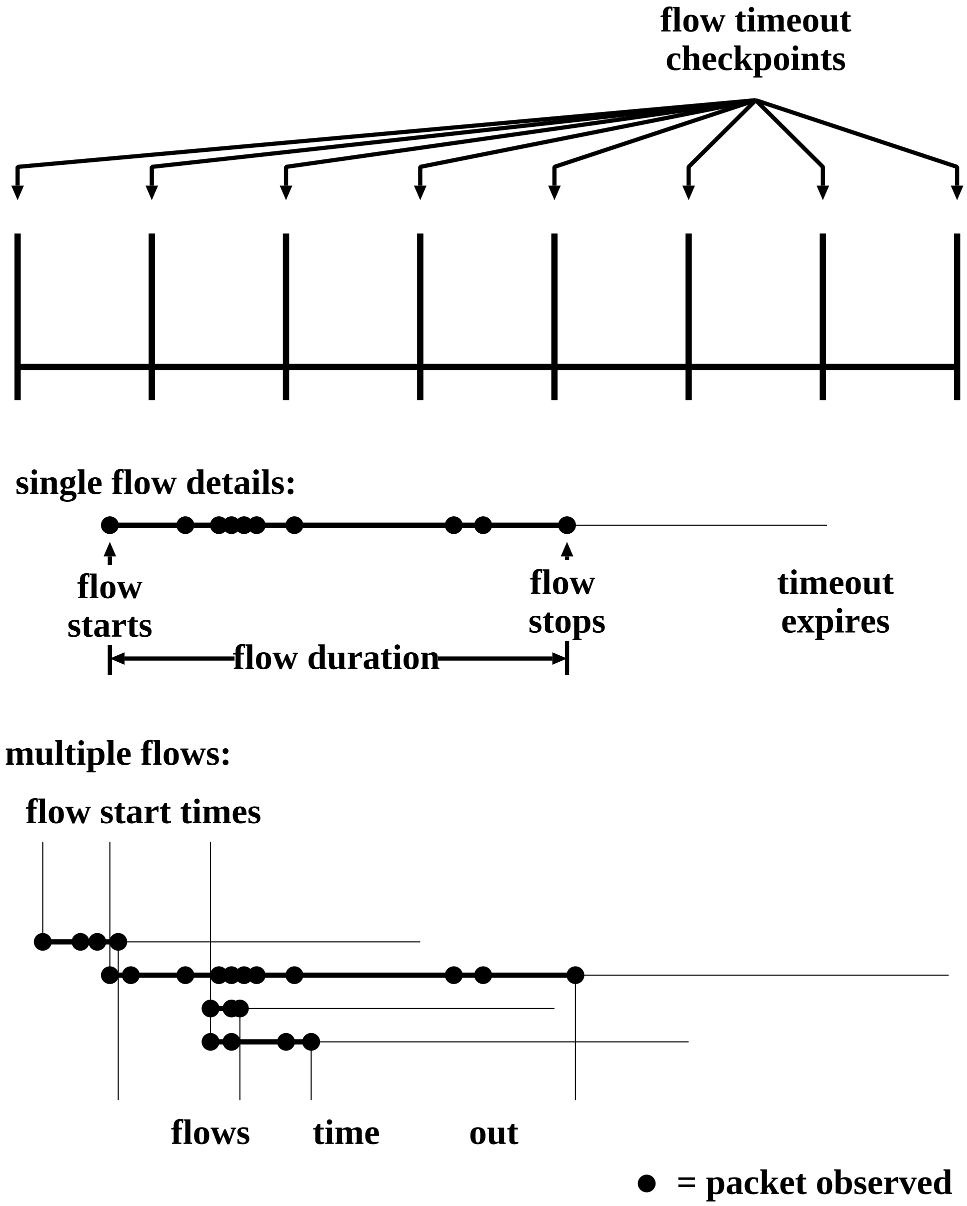

Figure 5.1: defining a flow based on timeout during idle periods

milliseconds.[3] For their studies of wide-area traffic at the transport layer, Caceres *et al.* [6] used a 20-minute timeout, motivated by the FTP idle timeout value of 15 minutes, and after comparison to a 5-minute timeout yielded minimal differences. Estrin and Mitzel [31] also compared timeouts of 5 and 15 minutes and found little difference in conversation duration at the two values, but chose to use a timeout of 5 minutes. Acharya and Bhalla [34] used a 15-minute timeout.

In contrast, we seek to systematically explore a larger range of this parameter, from 2 seconds to 2048 seconds by powers of two, in order to determine its interaction with the other components of the flow specification and the accompanying tradeoff in router requirements between flow setup and flow maintenance. We next describe how we apply our methodology to selected locations within the Internet environment.

5.5 Data collection

Figure 5.2 provides an abstract illustration of a subset of U.S. Internet interconnectivity as described in section 2.1; the actual implementation forms a much more complex framework. The lower half of figure 5.2 depicts (and Table 5.1 lists) the five sites at which we collected traffic data, covering a range of Internet traffic aggregation points. We collected packet traces from two T3 NSFNET backbone sites, both at supercomputing centers, for traffic going from those nodes into the backbone. In particular we selected the FDDI interfaces into external nodal switching subsystems (ENSSs): of the T3 NSFNET backbone: San Diego (ENSS 135) and Urbana-Champaign (ENSS 129). We did not simultaneously investigate the Ethernet interface, even though both interfaces carried operational traffic.

These two data trace locations allow us to conceptualize improvements in operational statistics collection for the NSFNET backbone. However, to test our methodology across a broader range of environments, we analyzed packet traces from three more locations: an internal FDDI LAN of one of these national supercomputing facilities, the San Diego Supercomputer Center (SDSC); an FDDI campus backbone of the University of California, San Diego (UCSD), and a smaller departmental subnet, the SDSC visualization laboratory Ethernet, which connects several workstations served by common file servers. For each site we used a dedicated SGI Indigo R4000 workstation to capture two one-hour traces: one during workday hours and the other during the night. For the first two traces, the backbone environments, we only collected data going in one direction, utilizing the FDDI MAC level address of the ENSS to filter the traffic into the backbone node. The unidirectional collection allows us to assess the impact of the inflow into the NSFNET backbone, and to mirror the results of the NNStat/ARTS traffic characterization described in Chapter 3, which collects statistics on inbound NSFNET traffic at each backbone node. For the remaining three data collection locations we did not apply filters but rather collected all IP traffic on the LANs.

The analysis in this chapter, similar to that in Chapter 4, requires a data collection mechanism that can capture timestamped header information from packets as they traverse a LAN. Since a processor implementing such a mechanisms must transfer information about each of the packets from the interface driver through the operating system into the collection application, the performance impact of the collecting process, and competing processes on the collection machine, is a concern. Even for these relatively brief intervals, the high volume of data which traffic collection generates, even when limited to packet headers and no user data, requires that we efficiently represent only essential header information for each packet.

We use a dedicated SGI Indigo R4000 processor as the statistics collection agent at each measurement point. We chose this processor for its strength in sustaining data collection at FDDI rates. We implemented software to gather the necessary information from each packet utilizing the *snoop* facility of the SGI IRIX operating system, which we describe further in the next section. For each packet the collection program writes to disk in binary format the following six 32-bit machine words:

0	timestamp (seconds portion)		
1	timestamp (microseconds portion)		
2	source IP address		
3	destination IP address		
4	packet length (bytes)	protocol	TCP flags
5	source port	destination port	

[3] Jain referred to this parameter as the *Maximum Allowable Intercar Gap*, or MAIG.

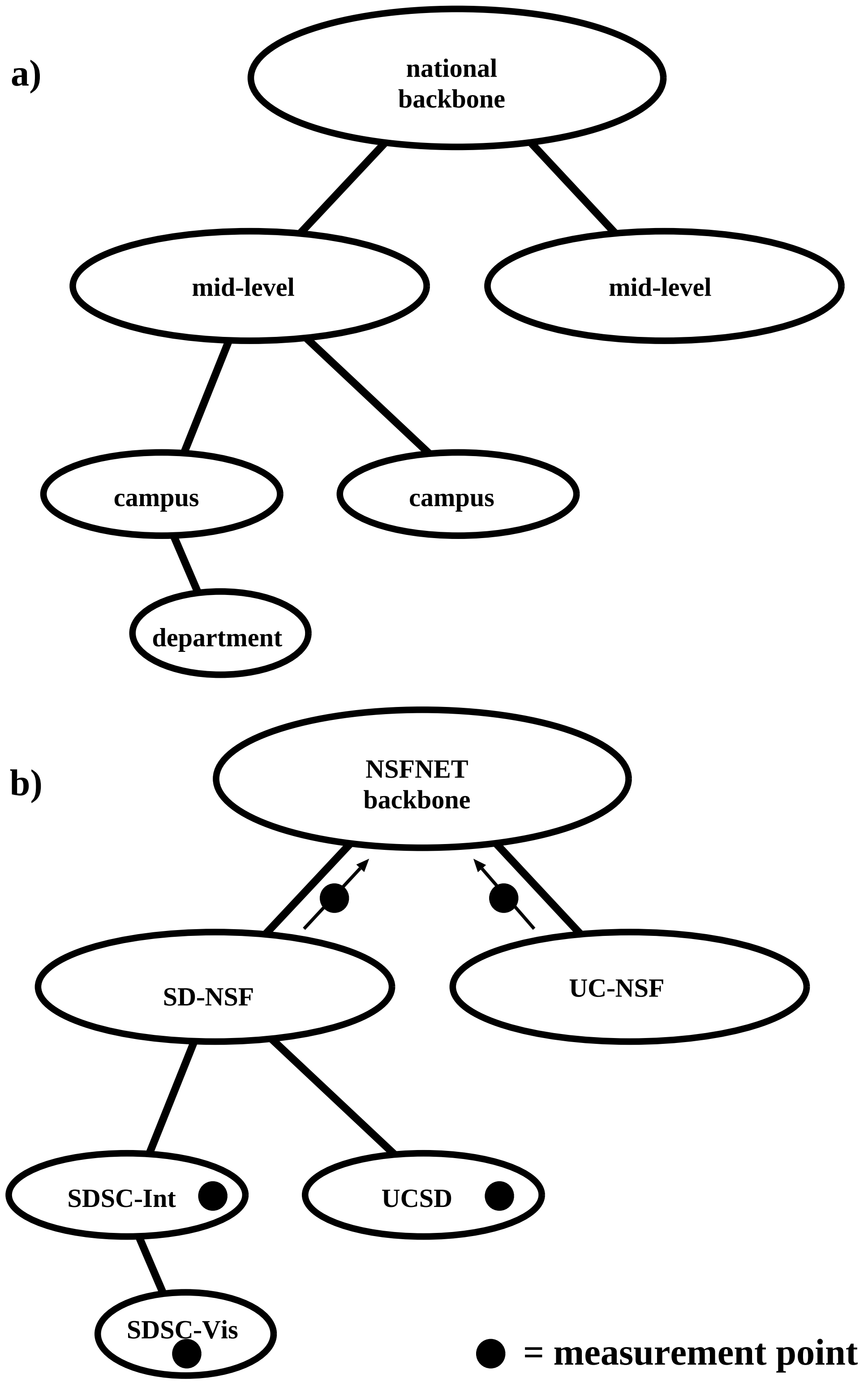

Figure 5.2: (a) abstract hierarchical model of U.S. Internet interconnectivity; (b) Internet locations we selected for characterization (SD-NSF: San Diego NSFNET node, traffic going into the backbone; UC-NSF: Urbana-Champaign NSFNET node, traffic going into the backbone; UCSD: UC, San Diego campus backbone; SDSC-int: San Diego Supercomputer Center, internal FDDI LAN; SDSC-viz: San Diego Supercomputer Center, visualization laboratory (small subnet of SDSC)

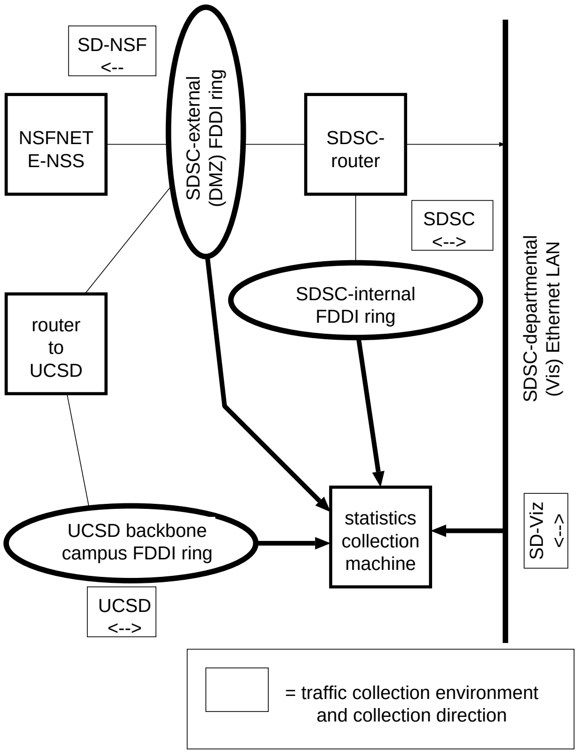

Figure 5.3: SDSC interconnection topology

Table 5.1: collection sites for flow profiling investigation

code	site	layer	date	start (MB)	start (MB)
SD-NSF	San Diego FDDI into NSFNET	backbone	23 mar 93	02:00 (15.3)	14:00 (30.8)
UC-NSF	Urbana-Ch. FDDI to NSFNET	backbone	29/25 mar 93	02:00 (31.4)	14:11 (97.6)
SDSC	SDSC FDDI	supercomp. ctr.	29/28 june 93	03:30 (16.5)	14:00 (49.8)
UCSD	UCSD academic FDDI	university	11 mar 93	02:00 (21.1)	14:00 (80.5)
SD-viz	SDSC visualization lab	dept. subnet	8/7 mar 93	02:00 (2.1)	14:00 (26.6)

For ICMP packets, there is no port information, but the ICMP type field is at same packet offset as the TCP/UDP source port number, so we capture this information instead. In order to trace non-IP packets, we record a value of *-1* in the *packet length* field, and use the *destination port field* to store the MAC level type. The only other field defined for non-IP packets is the timestamp.

Figure 5.3 shows a map of the SDSC topology, indicating how the collection probe machine attaches to the 100Mbps FDDI LAN ring which connects the center to the NSFNET ENSS. The location of the collection machine on these LANs allowed us to collect both internal LAN traffic at the supercomputer center as well as wide-area traffic outbound toward the NSFNET. The topology of the UC-NSF collection site is logically equivalent; we do not include the map here.

Table 5.1 lists the sites at which we collected data, including the starting times and size of each one-hour trace. For the graphs and tables in this paper we will refer to the data sets with the following five acronyms: SD-NSF, UC-NSF, SDSC, UCSD, and SD-viz. An AM or PM suffix will identify whether the data set was a nighttime (approximately 02:00-03:00 AM) or a workday hour (approximately 14:00-15:00 PM) of collection. Appendix 5.A provides some basic population parameters for these traces, including per-second packet and byte volumes and protocol cross-section.

5.6 Simulation details

We now describe how we used the collected traces to drive a simulation of soft-flow-state maintenance. To dynamically assess flows, we simulate a state-maintenance machine with an entry for each active flow, as defined above. The simulation proceeds as follows. Each time we see a new flow we create a new timestamped entry. We retain an entry as long as traffic exists for its associated flow. A flow "garbage collection" procedure executed at every second in the trace deletes all flows which no longer qualify as active according to the timeout value, and for those deleted flows reports the flow volume in packets and bytes, and the flow duration. The flow byte volume includes IP and transport protocol headers. In addition, each second we report the total number of active, new, and deleted flows for that second. Although we only time out flows synchronously at discrete observation points, every second, as figure 5.1 depicts, we do record the actual duration based on difference between the timestamp of the last packet seen in the flow and that of the first packet that incurred creation of the flow.

We devote the next two chapters to presentation of metrics obtained from applying our methodology to five strategically selected points at various levels of the Internet hierarchy. We divide these metrics into two categories: individual flow metrics (considered in Chapter 6), and aggregate flow metrics (considered in Chapter 7). Individual flow metrics include the number of bytes or packets per flow, flow duration, and their relationship. Our aggregate flow metrics fall into three subcategories: counts of the number of active and new flows per time interval; flow interarrival times; and flow locality metrics.

Table 5.2: population parameters for one-hour data sets measured by one-second intervals (AM = 02:00-03:00am; PM = 14:00-15:00pm)

	total(kp)	min	mean	95%	max	SD	total (MB)	min	mean	95%	max	SD
data set		packets/sec						kbytes/sec				
SD-NSF AM	637	44	178	272	1416	90	134	4	37	72	209	20347
SD-NSF PM	1285	79	358	475	924	70	246	16	68	123	225	28578
SDSC AM	316	30	88	176	708	46	67	3	17	46	307	17384
SDSC PM	1584	55	441	724	1141	160	414	6	115	241	974	82863
UCSD AM	840	102	234	361	1083	78	103	7	29	61	229	21792
UCSD PM	3094	424	861	1212	1765	185	516	44	143	298	540	72647
UC-NSF AM	1274	24	355	661	1032	165	436	2	122	320	653	93837
UC-NSF PM	4019	659	1118	1549	2355	228	1151	92	320	548	913	121645
SD-Viz AM	81	0	29	155	397	55	51	0	18	108	423	51744
SD-Viz PM	1101	1	314	695	1177	213	433	0	124	449	886	148234

Table 5.3: proportion of packets per protocol for each data set (AM = 02:00-03:00am; PM = 14:00-15:00pm)

protocol	SD-NSF		SDSC		UCSD		UC-NSF		SD-viz	
	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM
icmp	1.8	1.1	0.7	0.2	2.1	0.8	1.6	0.7	3.0	0.3
tcp	86.4	91.2	15.3	29.8	76.2	84.0	86.4	90.7	2.0	6.6
egp	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
udp	11.4	7.7	37.7	62.4	21.3	15.2	12.0	8.6	94.9	93.2
otherprot	0.3	0.1	46.2	7.6	0.4	0.1	0.0	0.0	0.1	0.0
all	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
applications on top of tcp or udp										
telnet	19.6	17.9	0.6	3.2	17.7	28.9	10.1	19.3	0.0	0.7
x11	0.4	1.4	11.5	2.8	6.6	5.4	0.3	1.1	0.0	0.8
rlogin	1.7	2.5	0.1	2.3	14.1	13.6	0.5	1.5	0.0	2.4
nntp	9.8	10.0	0.5	0.2	5.1	12.4	4.1	3.9	0.1	0.0
dns	2.2	2.6	0.5	0.2	3.1	2.6	7.3	2.5	0.7	0.4
smtp	3.5	6.8	0.0	0.1	2.9	2.6	4.4	4.1	0.2	0.2
ftpdata	32.3	25.2	1.6	8.5	5.8	6.4	37.9	29.7	0.2	0.7
nfs	0.0	0.0	1.5	52.1	3.3	7.2	0.1	0.1	54.2	77.2
andrew	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	0.0
otherap	28.4	32.3	36.9	22.9	38.9	19.9	30.3	37.2	41.5	17.4

Appendix 5.A Parameters of collected data sets

Table 5.2 provides basic population parameters of our data sets, including per-second packet and byte volumes. Tables 5.3 and 5.4 list the proportion of packet and bytes attributed to several network, transport, and application layer protocols for the ten data sets. Table 5.5 shows the average packet size for each protocol for the ten data sets. TCP/UDP port numbers sometimes map to a well-known TCP or UDP protocol, but often do not, precluding effective analysis of the cross-section of traffic. Both TCP and UDP utilize the Internet Protocol (IP) to transport the underlying datagram. Other protocols which utilize IP include: ICMP, IGMP, EGP, IGP, IPIP and several others outlined in the *Assigned Numbers* standard [89]. ICMP, the Internet Control Message Protocol, is used to communicate control information regarding the IP layer. IGMP (Internet Group Multicast Protocol) is used for multicasting; EGP (Exterior Gateway Protocol) and IGP (any private interior gateway protocol) are routing protocols. IPIP, IP-within-IP Encapsulation Protocol is used to encapsulate IP such as to tunnel across the multicast backbone Mbone [109]. IPIP is also used for some IPng protocol experimentation (i.e., PIP) [110].

The SD-NSF and UC-NSF data sets, both from interface points into the T3 backbone, are composed

Table 5.4: proportion of bytes per protocol for each data set (AM = 02:00-03:00am; PM = 14:00-15:00pm)

protocol	SD-NSF		SDSC		UCSD		UC-NSF		SD-viz	
	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM
icmp	0.5	0.4	0.3	0.0	1.2	0.5	0.3	0.2	0.2	0.0
tcp	94.2	95.2	9.4	42.8	73.1	79.3	84.4	97.1	0.3	4.7
egp	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
udp	5.0	4.3	39.5	50.5	24.4	20.1	15.3	2.7	99.5	95.2
otherprot	0.3	0.1	50.8	6.6	1.3	0.1	0.0	0.0	0.0	0.0
all	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
applications on top of tcp or udp										
telnet	4.9	6.1	0.4	1.4	8.2	12.8	2.8	5.0	0.0	0.1
x11	0.4	1.6	3.5	1.6	10.6	3.2	0.1	0.8	0.0	0.3
rlogin	0.4	0.9	0.0	1.2	9.5	5.8	0.1	0.4	0.0	0.4
nntp	9.3	11.1	0.9	0.4	10.3	21.9	2.3	2.9	0.0	0.1
dns	0.7	1.1	0.2	0.1	1.8	1.4	1.4	0.6	0.1	0.1
smtp	2.3	5.6	0.1	0.0	2.4	2.3	1.8	2.4	0.1	0.1
ftpdata	67.1	50.0	3.9	12.1	16.7	14.0	58.2	51.0	0.4	0.8
nfs	0.0	0.0	1.4	34.2	4.7	13.9	0.0	0.0	33.8	43.5
andrew	0.0	0.0	0.0	0.0	0.0	0.0	13.4	0.0	0.0	0.0
otherap	14.1	23.1	38.5	42.4	33.5	24.1	19.6	36.7	65.2	54.5

Table 5.5: average packet size per protocol for each data set (AM = 02:00-03:00am; PM = 14:00-15:00pm)

protocol	SD-NSF		SDSC		UCSD		UC-NSF		SD-viz	
	AM	PM	AM	PM	AM	PM	AM	PM	AM	PM
icmp	61	74	82	71	69	101	75	89	52	53
tcp	227	200	129	377	118	155	334	305	82	275
egp	49	46	NA	NA	NA	NA	60	57	NA	NA
udp	92	109	220	212	141	217	436	91	665	391
otherprot	171	168	231	228	349	282	126	97	28	28
all	209	192	210	262	123	164	342	285	635	383
applications on top of tcp or udp										
telnet	52	65	149	112	56	73	95	73	40	68
x11	168	217	64	149	196	98	121	216	40	147
rlogin	49	69	41	133	83	69	53	67	NA	64
nntp	199	213	424	453	248	289	190	213	344	683
dns	70	78	104	89	70	86	64	67	85	92
smtp	140	159	319	69	101	144	137	165	531	240
ftpdata	434	380	523	373	353	357	525	490	1372	495
nfs	140	143	195	172	177	316	143	140	395	216
andrew	68	66	NA	NA	65	65	1402	158	NA	NA
otherap	104	137	219	487	106	198	221	282	998	1200

of mostly TCP with a small amount of UDP traffic, in about a 7:1 ratio during the evening and 10:1 during the day. In this particular trace set, the UCSD campus backbone data set has a slightly higher proportion of UDP traffic, closer to a 3:1 ratio at night, 6:1 during the day. Large proportion of traffic in the *other applications* category is attributable to unregistered port numbers, many of which map to game playing servers on unregistered ports, e.g., *mud*.

In all traces the *other protocols* category is negligible, except for the SDSC internal network whose traffic consists of almost 50% *other protocols* during the evening trace. A significant portion of this traffic comes from one machine on this network that is a core multicast backbone (Mbone) router.

The SDSC visualization laboratory data trace is almost entirely UDP traffic, mostly in the *nfs* and *other application* categories. There is also substantial *nfs* traffic during the daytime on the SDSC internal network trace, and little Mbone traffic.

Appendix 5.B Performance integrity of the packet collection tool

We collected the data sets utilized in Chapter 5, 6, and 7 during March and June of 1993, using SGI Indigo R4000 based machines running the IRIX 4.0.5F version of their operating system. The Indigo supports FDDI and Ethernet interfaces and offers a proprietary facility, *snoop*, to allow an interface to promiscuously capture packets traversing these attached media. In this section we describe two components of the collection integrity, the packet loss of the capturing tool and the timer resolution of the operating system.

5.B.1 Packet loss

Due to resource contention, parts of the hardware and/or software components of the SGI facility we use for trace collection may not be able to absorb additional load above a certain packet arrival rate. Traffic bursts above a threshold, or competing processes in the machine, will result in loss bursts in one of three places, each represented by an associated *snoop* variable:

- the input interface, visible in the *if_ierrors* interface overflow counter
- at or below the interface queue, or driver, visible in the *ss_ifdrops* counter
- in the socket buffer, visible in the *ss_sbdrops* counter

Diagnostics of our snooping program indicated very low packet loss; during the busiest hour of collection for the UC-NSF data set which had by far the highest volume, the collection tool reported losing 0.125% of the approximately 4 million packets during the hour. However because these performance figures account for only the *ss_ifdrops* and *ss_sbdrops* information, we subsequently undertook tests to verify *snoop* performance further.

The first test was to push *snoop* to its performance limits using two R400 workstations on a dedicated an FDDI ring, one to send packets as fast as possible and the other to receive as many as it with *snoop*. We had the sending machine transmit sequences of packets of various sizes (40, 160, 520, 1470, and 4300 bytes), addressed to an non-existent machine on the ring, to emulate an environment where the collection machine is capturing all packets on the medium regardless of their destination. We monitored the number of packets and bytes sent per second, and the number captured via the snoop interface on the receiving machine. We then modulated the rate of packet transmission from the sending to receiving machine until the measurements diverged, i.e., until the receiving machine clearly reports reception of fewer packets than the transmitting machine sent. The only runs which resulted in any losses were with packet sizes of 4300 bytes, of which we have none in our collected data sets.

However, a machine is able to achieve a much higher per-second packet transmission rate using smaller packets. While the receiver was able to receive all those packets as well, in an operational environment with multiple fast transmitters of small packets the receiver may likely not be able to keep up. Therefore a real environment may be able to provide enough aggregate sending capacity to overpower our collection machine. We thus needed to characterize a more realistic environment with multiple sending machines in order to further secure our confidence in the performance of the snooping machine.

This objective inspired our second test. We constructed a program to report the following variables each second:

- timestamp (Unix format)
- *if_ierrors*
- *ss_ifdrops*
- *ss_sbdrops*
- IP input packet count
- non-IP packet count

- IP bits (number of IP bytes multiplied by 8)
- ratio of lost packets to total packets
- average IP packet size

We ran this program for 2285 busy seconds in November 1993 on the SDSC-DMZ FDDI ring (the SDSC traffic inflow into the ENSS FDDI interface), with no filtering of the traffic. The program received 4,476,938 packets during that time period, at a mean rate of 1959 packets per second and a maximum of 3470 packets per second. During that time period, the *if_ierrors* counter reported dropping 162 packets at the input interface, the *ss_drops* counter reported dropping none, and the *ss_sbdrops* counter reported dropping 63 packets. Therefore we lost 162+63 = 225 packets during the observation period, which is a loss rate of 0.005% of the almost 4.5 million packets. Because these packet rates were considerably higher than the ones in our study (the mean and peak per second packet rates for our busiest data set UC-NSF PM were 1100 and 2350, respectively), this test gave us an adequate degree of confidence in the integrity of the collection mechanism for our data sets.

5.B.2 Timer granularity

The IRIX 4.0.5F operating system (which SGI has now upgraded to version 5.2, with scheduled release during April 1994) on the SGI R4000s generates 100 clock interrupts per second in order to update its time of day clock in 10 ms increments. This resolution suffices for most situations but is suboptimal for packet collection because the timestamps for all packets are then modulo 10 ms. To obtain finer clock resolution the SGI offers a feature called *ftimer* (fast timer). Enabling the *ftimer* feature changes the number of clock interrupts, and thus the timer resolution, to that defined by the $FASTHZ$ kernel variable. By default this value is 1000 interrupts per second, for a 1 ms clock resolution, but one can recompile the kernel with different values of $FASTHZ$. Of course more frequent clock interrupts increase the burden on the CPU, so for performance reasons SGI limits the $FASTHZ$ variable on the R4000s to a maximum 2500, which allows for 400 μsec spacing between timer interrupts.

Although our preference was 2500 interrupts a second for a 400 μsec clock granularity, we did not have kernel reconfiguration access to the machines at SDSC-int and UC-NSF, and so these data sets use only the default *ftimer* resolution of 1 μsec, or 1000 system clock interrupts per second. The other data sets, at SD-NSF, UCSD and SD-viz, have a 400 μsec resolution.

In both cases the operating system imposed the following limitation. If the kernel had to provide a timestamp, e.g., in response to the *gettimeofday* unix system call, for multiple events within the same 400 or 1000 μsec interval (whichever was the minimum clock resolution), it incremented the timer by one μsec for each event. Thus if a 1000 μsec kernel received packets at 1250, 1300, 1400, 1900, and 1999 μsec, and no other processes in the system did *gettimeofday* calls during that interval, the kernel would timestamp the packets for the *snoop* application at 1000, 1001, 1002, 1003, and 1004 μsec.

This behavior contributes to distortion in the differences between timestamps of events separated by less than a few multiples of the timer resolution. We illustrate with examples of how interarrival times are bucketed into milliseconds. Two events that cross a system clock interval, e.g., one at 999 μsec and the other at 1001 μs, would appear to have an inter-event interval of 1 ms rather rather than 0 ms (which is admittedly itself different from the 2 μs, but that inaccuracy is from a different source). The phenomenon can even stretch across multiple clock intervals: events at 999 μs and 2001 μsec would appear to have an inter-event interval in the 2 ms bucket rather than the 1 ms bucket. Across several multiples of the clock resolution, the magnitude of the error decreases relative to the event interarrival spacing. Given these limitations that the operating systems impose on our observation platforms, time spacing measurements such as the flow interarrival times in section 7.2 are subject to noticeable distortions at very low millisecond values, and we exclude these values during our model fitting procedure in section 7.2.1.

Chapter 6

Individual flow metrics

It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.

– *Sir Arthur Conan Doyle*

What is the difference between method and device? A method is a device which you use twice.

– *quoted by George Polya*

In this chapter we present metrics derived from the methodology we described in Chapter 5. We focus on metrics of individual flows, those that one can attribute to a single flow, such as the duration, packet count, or byte volume. In Chapter 7 We present aggregate flow metrics, those inherently descriptive of a population of flows rather than a single flow.

We consider various flow granularities as outlined in the previous chapter, such as destination network, host-pair, or host and port quadruple. Our measurements demonstrate (i) the brevity of a significant fraction of IP flows (ii) that the number of host-pair IP flows is not significantly larger than destination network flows, and (iii) that schemes for caching traffic information could benefit by making caching decisions taking into account higher layer information.

In exploring each metric we vary one of several parameters:

- flow timeout (from 2 to 2048 seconds; a 2048-second timeout is essentially infinite for this data);
- flow specification (destination network (*dn*), destination host (*dh*), source host (*sh*), network pair (*np*), host pair (*hp*));
- the environment (five sites, measured for two separate hours each, as described in section 5.5);
- the transport or other protocol (TCP, UDP, EGP, ICMP) and application (*dns*, *nntp*, *ftpdata*, *telnet*, *smtp*, and *www*).

6.1 Flow timeout

Figure 6.1 shows for a range of timeouts the cumulative distributions of flow byte volume, flow packet volume, and flow duration. Recall that the byte values include the IP and transport or other header, e.g., TCP, UDP, or ICMP; the minimum value is 28 bytes, for example for ICMP echo request and reply messages. All three graphs in the figure aggregate all host pair flows regardless of application type, use the host pair flow specification, and the UC-NSF PM data set. The data indicate that regardless of timeout value, 80% of all flows are less than 37 packets and four kilobytes in volume. For timeout values of 64

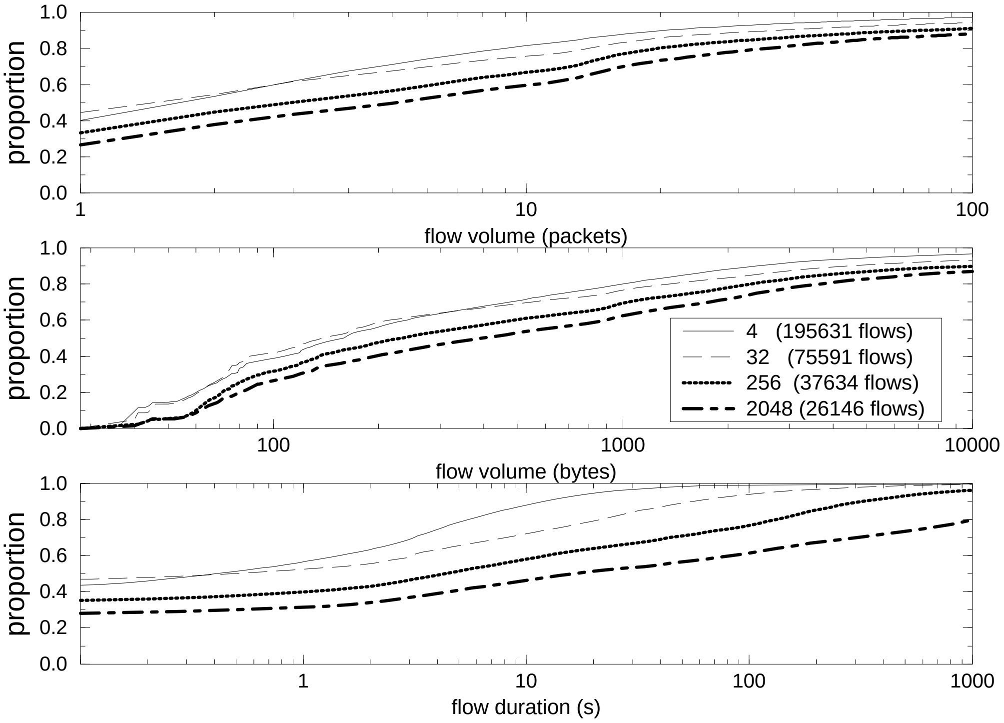

Figure 6.1: cumulative distributions of host pair flow packet volumes, byte volumes, and flow durations for a range of timeout values: 4, 32, 256, and 2048 seconds (UC-NSF PM)

seconds or less, 90% of the flows are less than 50 packets, 5.5 kilobytes and 100 seconds. For a 2048 timeout, essentially infinite relative to the duration of the data collection[1], *27% of the flows consist of a single packet of less than one hundred bytes.*

We list in the legend of figure 6.1 the number of flows in the hour-long data set given each timeout value. A short timeout value will split longer flows into several short ones, so naturally smaller timeouts will yield a much larger number of flows, and a greater proportion of flows of smaller duration. The top line in each graph corresponds to a 4-second timeout value. For readability we omit the lines for the other timeout values from figure 6.1, but to highlight a finer timeout granularity figure 6.2 reproduces the upper graph in figure 6.1, the distribution of flow packet volumes, but includes data for all ten timeout values we simulated.

In this figure the ten curves divide into three groups. As the timeout goes from 4 to 32 seconds, the slope of the curve decreases such that the curve crosses over the curve corresponding to the previous timeout value. From timeouts of 32 and above, the curves no longer cross but rather maintain a fairly constant slope, and just shift lower, indicating a larger proportion of flows at higher packet volumes. The change in slope for the first few timeout values is consistent with the low timeouts constructing an unnatural distribution of flow packet volumes by forcing a large number of naturally higher volume flows into several smaller ones. The result is a huge number of flows, so although the number of single or two-packet flows is very high, the proportion of them is lower since there are so many other short flows than there are at higher timeouts. A router using low timeouts, i.e., under 16 seconds for this data set, would essentially be thrashing, constantly replace flows it has only just timed out a few seconds ago. Note that the total number of unique flows in this data set is 25457, which a 2048 timeout closely approaches since it rarely closes a flow that was not finished anyway.

There is a considerable gap between the 128 second timeout and 256 timeout line. We interpret this to mean that for timeouts larger than 128 the router is essentially keeping open every flow until it is finished, or longer, a phenomenon which the legend confirms by showing the total number of flows for each timeout. As the timeout increases beyond 128 seconds, the distribution curve drops slightly to account for

[1] Running the simulation with an infinite timeout led to results insignificantly different from the 2048 timeout.

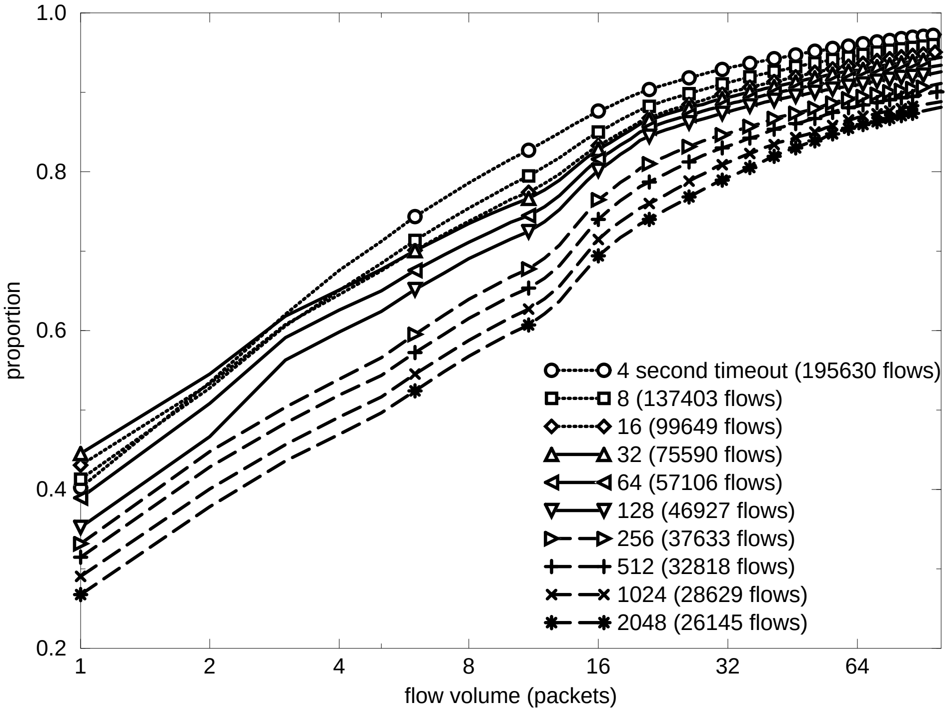

Figure 6.2: cumulative distribution of host pair flow packet volumes for a range of timeout values: 4, 8, 16, 32, 64, 128, 256, 512, 1024, and 2048 seconds (UC-NSF PM)

the merging of what used to be several moderately long flows into one longer one. However, the marginal benefit of merging these few flows is not nearly as large as it is for merging tens of thousands of flows such as in the difference between the 4 and 32 second timeouts, and the cost is considerably higher in terms of memory resources since higher timeouts causes many flows to stay around longer than necessary and thus consume undue resources in network equipment, e.g., routers. This data thus leads us to believe that for environments with traffic similar to this one, flow timeout values of between 16 and 128 seconds are appropriate.

6.2 Flow specification

Figure 6.3 shows for multiple flow specifications, i.e., source host, destination host, destination network, network pair, and host pair, the cumulative distributions of flow byte volume, flow packet volume, and flow duration. We use the UC-NSF PM data set with a 64 second flow timeout and aggregate all flows of a given granularity regardless of application type. The graphs reflect how multiple host pair flows aggregate into flows of coarser granularity, consistent with environments that simultaneously support flows from many active host pairs which share a common source or destination IP network number, e.g., backbone entrance points. This phenomenon yields flow duration and volume distributions which are skewed lower for host pair than for destination networks. Indeed, figure 6.3 indicates that the fiftieth percentile for host pair byte flow volume is less than 200 bytes, while for destination network numbers it is more than a kilobyte. The fact that this data set includes traffic only in one direction contributes to the disparity, since the ratio of the number of sources to destinations is considerably lower than with the bidirectional collection of the other data sets. Figure 6.4, plots the same distributions for the bidirectionally collected UCSD data set. A campus network backbone that aggregates traffic to a lesser degree, this environment exhibits much less disparity among the distributions of flows at different granularities.

The legends of figures 6.3 and 6.4 show another interesting aspect which is also true for our other data sets: the number of host pair flows is only two or three times the number of network pair or destination

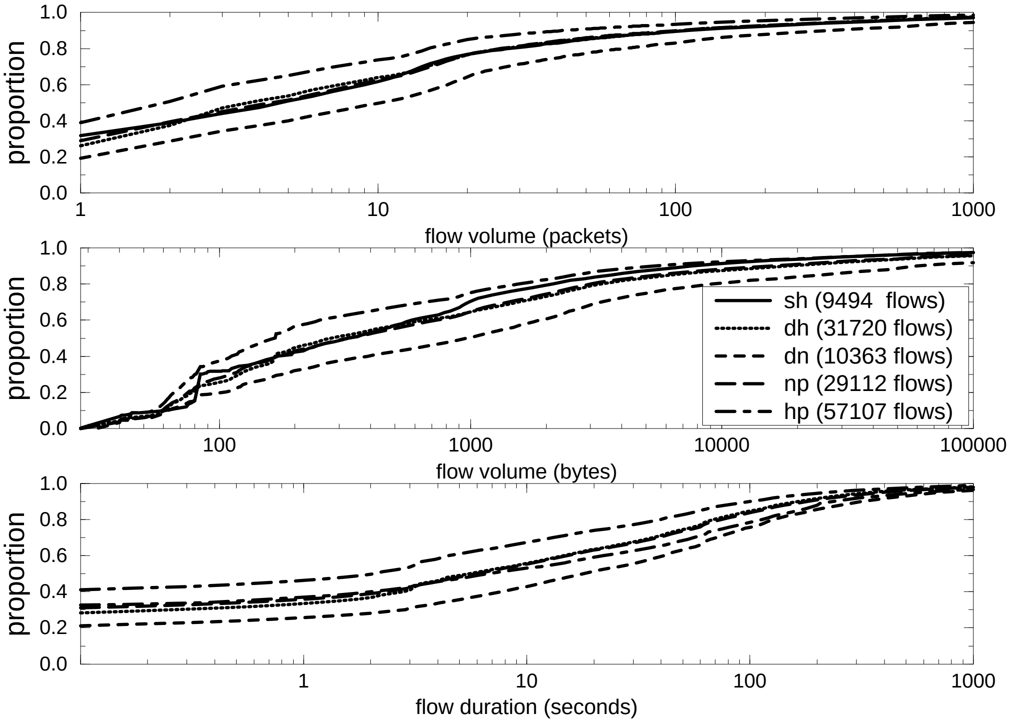

Figure 6.3: cumulative distributions of flow packet volumes, byte volumes, and flow durations for five flow specifications: source host (sh); destination host (dh); destination network (dn); network pair (np); host pair (hp) (UC-NSF PM, 64 second flow timeout)

network flows. In other words, even wide area traffic exhibits considerable locality: the number of host pair flows certainly does not scale on the order of the square of the number of network number pair flows, as a uniform matrix of traffic volume among connected sites might imply. In reality, as we also illustrated in section 3.5.3, the matrix of traffic among sites is sparsely filled. We discuss flow count metrics further in Chapter 7, but note here that these measurements indicate that maintaining host pair state in the Internet may not impose prohibitive load on the routers, auspicious for recently proposed soft-state based Internet routing and congestion control schemes [48] [52] [49] [51] [50] [53].

6.3 Environment

Figure 6.5 shows for five environments the cumulative distributions of flow byte volume, flow packet volume, and flow duration. All three graphs in the figure aggregate all host pair flows regardless of application type, use the host pair flow specification, and a 64 second flow timeout.

These three graphs further confirm the surprising features of figures 6.1, 6.3 and 6.4. *For the backbone environments, approximately 40% of the host pair flows consist of a single packet, and less then 100 bytes.* For these wide area environments generally between 50% and 60% of flows are less than 200 bytes; between 70% and 80% consist of less than ten packets. Half of the host pair flows are of less than one second duration, with the 90th percentile above 100 seconds. This distribution may imply that a flow cache designer should implement a two-phase timeout, e.g., flows that pass a one-second threshold would receive space in a longer term cache. We discuss caching issues further in Chapter 7.

The LAN environments tend to have a greater proportion of higher volume flows, consistent with typical long-term local usage patterns of workstations and terminals, as well as distributed file systems and print servers. We do see significant peaks for flows of 5 packets and 224 bytes for the SD-viz PM data set in the top and middle graphs, respectively. To gain further insight into this phenomenon we isolated the 554 host pair flows that consisted of 5 packets in the SD-viz PM data set. The 37 host pairs responsible for these

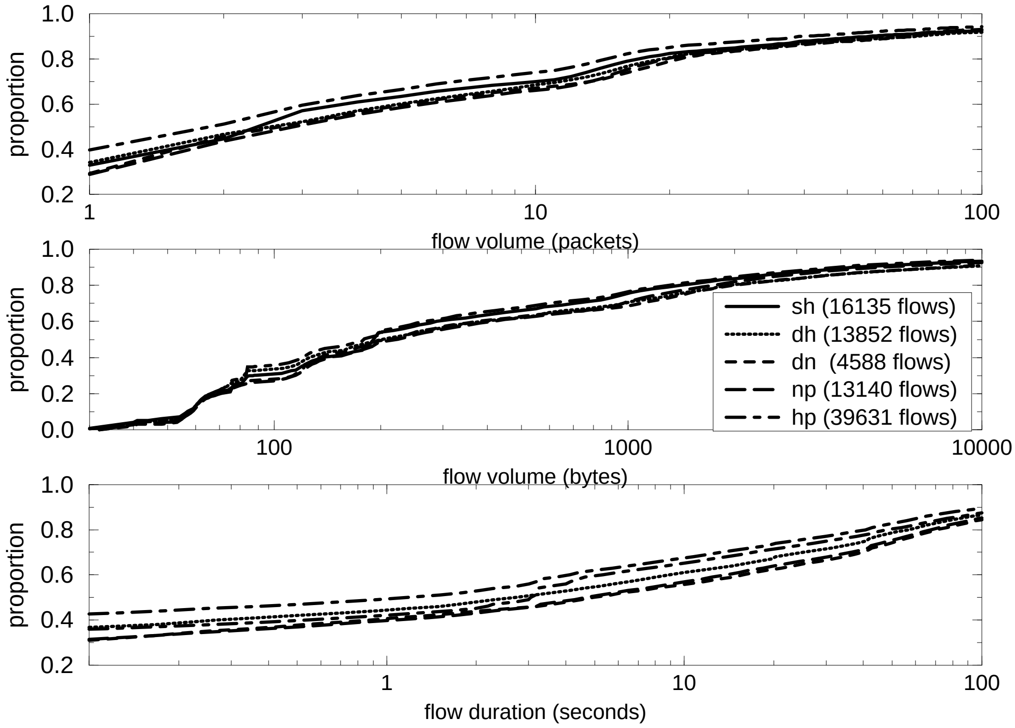

Figure 6.4: cumulative distributions of flow packet volumes, byte volumes, and flow durations for five flow specifications (UCSD PM, 64 second flow timeout)

flows were executing a *timeslave* system program that periodically exchanges a 32-byte UDP timestamp packet and four 48-byte ICMP time request/reply packets in order to synchronize time between machines. Aggregating traffic into host pair flows as we have done for this graph combines the UDP and ICMP time requests into a single host pair flow.

All of the graphs imply that there are very few flows that last very long. In almost every environment and using a 64 second timeout, approximately 80% of all host pair flows consist of less than 100 packets, less than 10 kilobytes, and less than 30 seconds in duration.

6.4 Higher layer protocol

The metrics we have shown thus far do not differentiate among flow type, which turns out to have a significant effect on the expected size and duration of a flow. Figure 6.6 illustrates how the transport protocol will affect our selected metrics. The three graphs display the same metrics as those in the previous sections: the cumulative distributions of flow byte volume, flow packet volume, and flow duration. The legend indicates the percent of the total packets, bytes, and flows each category contributes. The top graph shows that *50% of TCP flows consist of less than 10 packets; over 60% of UDP flows are single packet flows, and 52% of ICMP flows consist of a single packet.*[2] The middle graph shows that a large majority of these single packet ICMP flows carry 84 and 168 bytes, with smaller peaks at 56 and 112 bytes. By examining specifically these ICMP flows in the UC-NSF data set, we determined that the 84-byte flows were largely single packet ICMP echo request or echo reply messages, and the 168-byte flows were largely three-packet ICMP port unreachable flows. The smaller peaks at 56 and 112 byte-flows were also mostly due to single packet host and port unreachable messages. Unreachable and echo messages typically comprise the large majority of ICMP flows. There were only 19 EGP flows, two of which lasted the entire hour and carried over 200 packets each; the remaining 17 of which carried under ten packets each.

[2] We collapse our data by port quadruples, and thus fragmentation, if significant, will distort this figure somewhat. However we saw negligible fragmentation in the UC-NSF environments during our trace intervals.

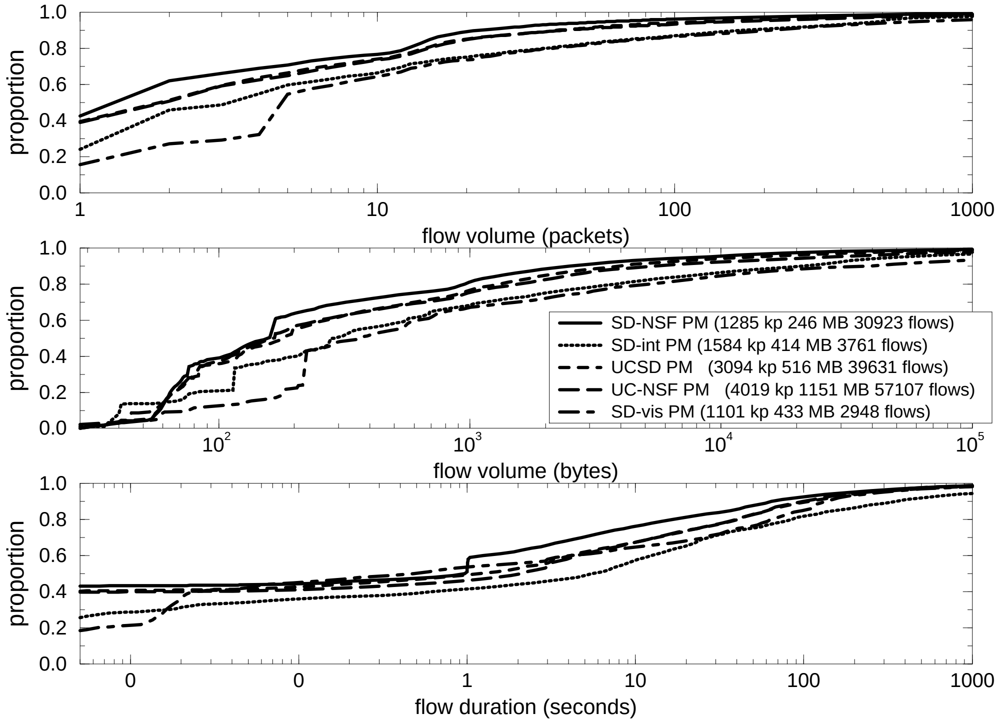

Figure 6.5: cumulative distributions of host pair flow packet volumes, byte volumes, and flow durations for five environments (64 second flow timeout)

The application layer also has a significant effect on the characteristics of an individual flow. We first illustrate in Figure 6.7 the pattern of packet arrivals for instances of four common applications (*smtp*, *telnet*, *ftpdata*, *nntp*) from the SD-NSF PM data set. These applications correspond to electronic mail, interactive remote login, file transfer, and network news distribution. Each horizontal line reflects a single host/port pair flow and each dot in the line represents a packet in that flow. The *ftpdata* flows tend to be brief and intense, but some sustain over long periods. *Smtp* and *nntp* flows are also short and sometimes exhibit several bursts with intervening periods of lower activity. *Telnet* flows typically are of much longer duration and lower intensity. Plotting flows for *dns*, *gopher*, and *ntp*, which are querying transaction applications for nameserver, information resource, and time information, respectively, yielded short periodic flows with much lower frequency. Many of these flows are of few or single packets, e.g., *dns* nameserver queries or *ntp*, which are then followed by long idle periods before a subsequent flow to the same destination. Other transaction flows, such as for *www gopher* or *dns* zone transfers, can be considerably larger, and may never recur again, e.g., if future *www* requests go to a different server. Video flows, such as those carried across the Mbone via the IPIP protocol of which there were none in our data sets, would show up as straight black lines in such a plot.

Figure 6.8 shows the volume and duration distributions for these and two other representative applications on the Internet, specifically *udp/dns*, *nntp*, *ftpdata*, *telnet*, *smtp*, and *www*. We aggregate all other flow types into an "other" category. The legend indicates the percent of the total packets, bytes, and flows each category contributes. The data indicate that a *large majority of the single packet UDP flows are from the* dns *protocol*, unsurprising given the nature of the *dns* protocol. With a 64-second timeout, 65% of *dns* flows consist of a single packet.

The longest average flow durations in our samples seem to characterize the *www* protocol for the UC NSF PM data set. Contributing to the appearance of *www* traffic at the UC-NSF measurement point is the fact the the National Center for Supercomputing Applications (NCSA), located on the UIUC campus, has led the development and prototyping of the *xmosaic* tool and servers. During our measurements in March 1993, only the UC-NSF measurement location exhibited significant *www* traffic. Since that time the absolute volume as well as proportion of *www* traffic at the UC-NSF backbone inflow point has grown

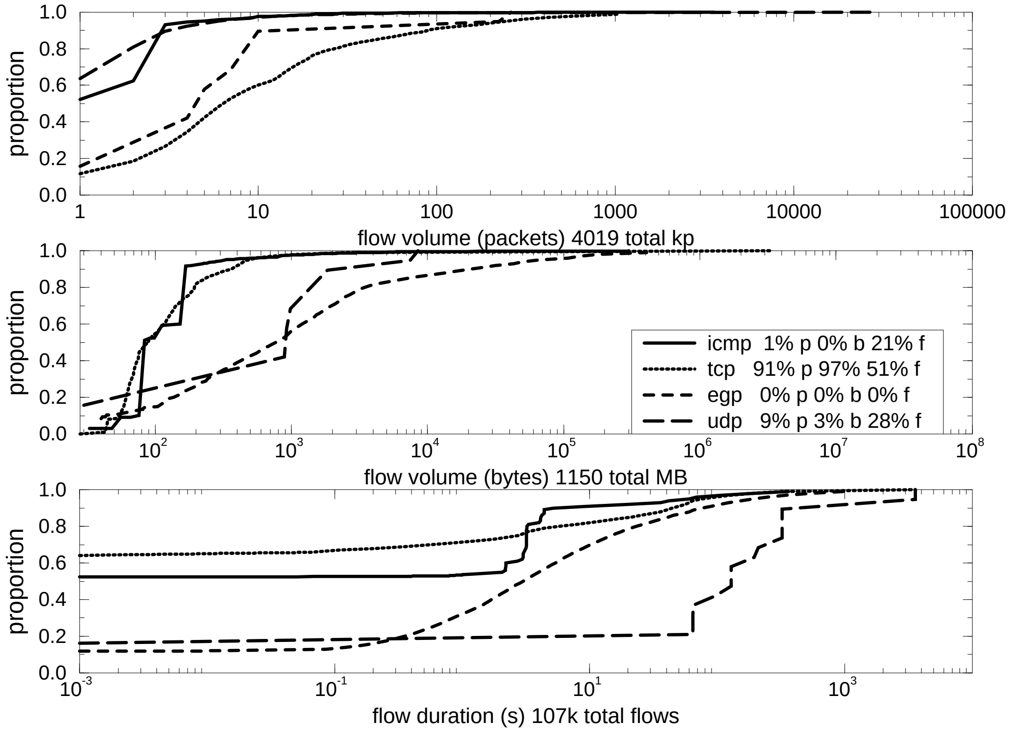

Figure 6.6: cumulative distributions of flow packet volumes, byte volumes, and flow durations for four transport protocols (UC NSF PM, 64 second flow timeout)

dramatically, causing concern to NCSA/UIUC system administrators regarding the operational impact of traffic from advanced Information Resource Discovery Services (IRDS).

The distribution of *ftpdata* flow durations is not significantly different from that of previous studies at the transport layer. Using timeouts of 64 seconds or larger, approximately 65% of the *ftpdata* flows are less than 10 kilobytes.[3]

6.5 Two-dimensional perspectives

One can also present two-dimensional perspectives of individual flow metrics. For example one could portray the packet-duration or byte-duration space of flows, while varying one of the parameters we have explored: flow timeout, environment, flow specification, higher layer protocol or application. The top half of figure 6.9 provides an example, outlining the packet-duration space of the same Internet applications shown in figure 6.8. Each diamond delineates the population of flows of that application; the vertices are at the 5th percentile, median, and 95th percentile for both the packet and the duration axis. As with previous figures, the legend indicates the percent of the total packets, bytes, and flows each category contributes. The graph of byte-duration space appears similar; we do not include it here.

The graph highlights visible differences among the applications, especially given the log scaled axes. For example, as an interactive protocol, *telnet* exhibits a large spread in flow duration, with the longest duration flows lasting much longer than those of bulk data protocols such as *ftpdata* or transaction style protocols such as *www* and *smtp*. Bulk transfer and transaction protocols also exhibit more predictable

[3] Caceres *et al.* [6] using a 20-minute flow timeout, found that 75-90% of bulk transfer conversations consist of less than 10 kilobytes of data. Our results, although slightly different, are also consistent with their measurements that indicate that over 90% of interactive conversations consist of less than a thousand packets. We note the difficulty of getting a picture of *telnet* flows given unidirectional traffic; Caceres *et al.* [6] and Paxson [61] show that interactive applications can generate as much as ten times more traffic in one direction than the other. Caceres *et al.* find in their measurements that bulk transfer flows are often bidirectional as well, but Paxson's measurements do not show a strong degree of bidirectionality.

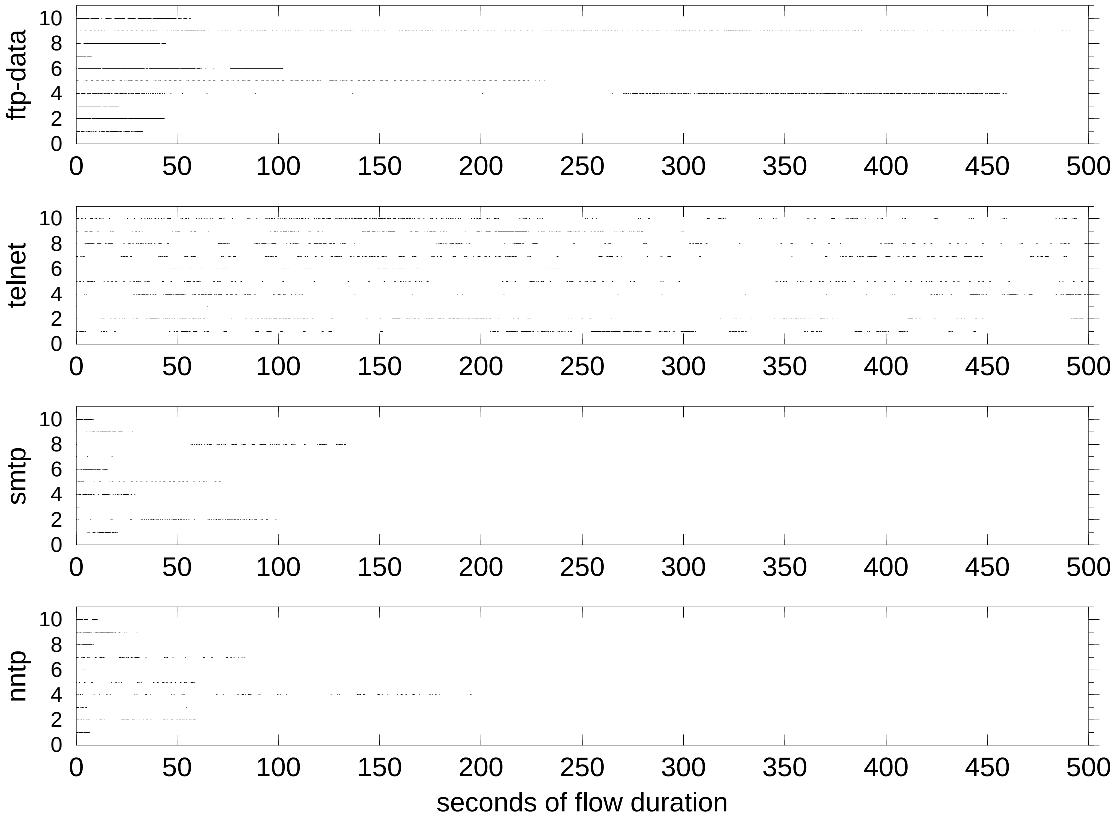

Figure 6.7: depiction of intra-flow packet arrivals of ten example flows for four common applications (SD-NSF PM)

duration, consistent with their typical single burst usage. An extreme case is the packet volume distribution of the *udp/dns* flows, many of which consist of a single packet, with the 95th percentile at six packets.

The top half of figure 6.9 elaborates on the implications of figure 6.8 regarding the influence of packet type on the utility of maintaining flow state. Table 6.1 provides further insight, ranking the ten protocols with the highest number of flows in the UC-NSF PM data set using a 64 second timeout. The table also lists the number, proportion, and rank of flows, packets, and bytes for each protocol. *Although* dns *constituted only 2.4% of the total packets, it constituted 22.8% of the total number of flows*, given a 64-second flow timeout.[4] Designers of routing or accounting flow state tables or caches may improve performance by selectively choosing not to store information for packets that are highly likely to represent short flows, e.g., *dns*, *gopher*, *ntp*. Such flows, whose packet-duration profiles are largely in the lower left of the top graph in figure 6.9, will contribute to cache thrashing because they consume valuable memory that will likely not require future reference. According to table 6.1, for the UC-NSF PM data set *a router could save over a third of its cache memory by refusing to cache information regarding dns, gopher, and ntp traffic.* Clearly ignorance of higher layer information regarding the nature of the traffic imposes a high opportunity cost.

This data also has implications for multiplexing IP flows onto a wide-area ATM substrate using virtual circuits (VCs). Caceres [107] suggests that the optimal multiplexing policy for bulk transfer conversations whose packet transmission rates are limited by transport protocol windows sizes and available bandwidth, e.g., *ftp*, *smtp*, *nntp*, would be one virtual circuit per conversation. He further finds that interactive conversations, e.g., *telnet*, should not share virtual circuits with bulk transfer conversations to avoid high delays, although many interactive conversations can share a virtual circuit among themselves without detrimental effect. Our data indicates that flows that do not fall into either of these categories, e.g., *dns*, *gopher*, *ntp*, and many unknown traffic flows, would likely require the suboptimal router behavior of assign-

[4] Using a 4-second timeout increased this proportion to 27.6%.

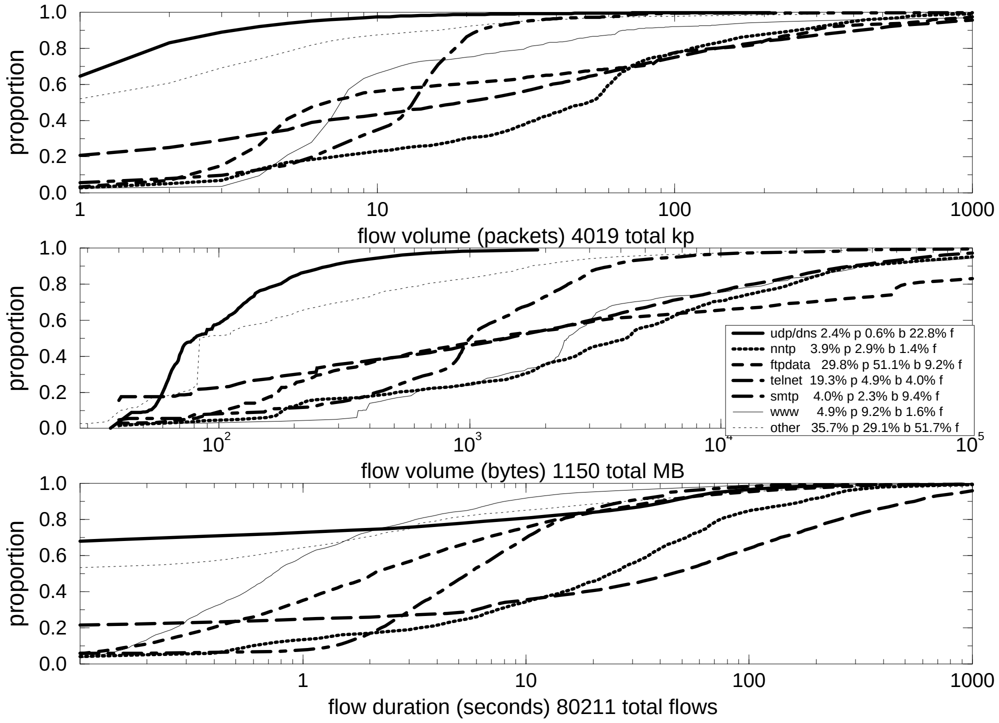

Figure 6.8: cumulative distributions of flow packet volumes, byte volumes, and flow durations for six port-based applications (UC NSF PM, 64 second flow timeout)

ing a virtual circuit for a flow that would not make any further use of it. One alternative policy would be to tear down such VCs immediately without waiting any timeout interval. Another potential technique for reducing the number of circuits needed for such flows would be to bundle them together with others to the same destination over the same virtual circuit. We discuss the maintenance of virtual circuits further in Chapter 7.

The application layer protocol also influences the interaction between other parameters and individual flow metrics. We discuss one example of such interaction using the flow timeout parameter. As figure 6.8 showed earlier, with a 64-second timeout, 65% of *dns* flows consist of a single packet. Allowing an infinite timeout does not change the situation significantly: 50% of flows consist of a single packet. The *ftpdata* profile also does not depend significantly on the timeout value. Using a 64-second timeout, the median number of packets in an *ftpdata* flow was 7, and with an unlimited timeout, the median was about the same, 8 packets.

However the timeout value does affect the packet volume and duration of other protocols. The lower half of figure 6.9 highlights two flow types for which the timeout affects the profile: *telnet* and *smtp*. We show for these two flow types the difference in flow packet volume and duration when using a 64-second versus an infinite timeout. For *telnet* flows using a 64-second timeout the median number of packets per flow was 20; with an unlimited timeout this median jumps to 78 packets, suggesting that telnet flows are often idle for more than 64 seconds. Similarly, for this data set the median and 95th percentile of *smtp* flows do not depend on the timeout, but the 5th percentile at the 64 timeout value is at a single packet, indicating a number of single packet *smtp* flows between the same two hosts separated by more than 64 seconds. These measurements indicate that for at least a few types of network traffic, the timeout value will affect flow assessment.

6.6 Conclusion

The proliferation of different Internet traffic types that exhibit fundamentally different workload

Table 6.1: proportion of flows, packets, and bytes attributed to major protocols (UC-NSF PM, 64 second timeout)

port	service	number			% total			rank		
	totals:	106848	4019.1	1150.9						
		flows	kp	MB	flows	pkts	bytes	flows	pkts	bytes
17:53:0	domain	24345	97.4	6.50	22.8	2.4	0.6	1	10	14
6:70:0	gopher	11507	146.7	57.71	10.8	3.7	5.0	2	6	5
6:25:0	smtp	10044	161.7	26.64	9.4	4.0	2.3	3	4	8
6:20:0	ftp-data	9807	1197.1	587.53	9.2	29.8	51.1	4	1	1
6:21:0	ftp-ctrl	5307	108.3	7.49	5.0	2.7	0.7	5	9	13
6:23:0	telnet	4222	774.9	56.88	4.0	19.3	4.9	6	2	6
6:79:0	finger	2253	10.6	.89	2.1	0.3	0.1	7	30	33
6:80:0	www	1662	196.9	105.55	1.6	4.9	9.2	8	3	2
17:123:0	ntp	1592	3.6	.27	1.5	0.1	0.0	9	50	63
6:119:0	nntp	1532	154.9	33.15	1.4	3.9	2.9	10	5	7
	subtotal	72271	2852.1	882.60	67.6	71.0	76.7			

Table 6.2: key results of individual flow profiling for five selected one-hour data sets

1. Host pair flow timeout values between 16 and 128 seconds seem to be an appropriate tradeoff between router processing and memory, i.e., incurring thrashing in a cache and requiring too many flow state entries that are often never again used after the first packet in the flow caused the creation of the entry in the router.

2. For timeout values of 64 seconds (or less), 90% of the flows are less than 50 packets, 5.5 kilobytes and 100 seconds. For virtually infinite timeouts, 27% of the flows consist of a single packet of less than one hundred bytes.

3. The large proportion of flows are very short in duration: using a 64-second timeout for the busiest backbone data set (UC-NSF PM), almost 60% of the flows are less than one second (40% consisting of a single packet).

4. Flow volume and duration is correlated to the higher level protocol. Given a 64 second timeout, only 10% of TCP flows consist of a single packet, but over 60% of UDP flows consist of a single packet, and 52% of ICMP flows consist of a single packet.

5. TCP/UDP ports also provide an indication of the expected duration and volume of a flow: e.g., the large majority of the single packet UDP flows are from the *dns* protocol (unsurprising given the nature of the *dns* protocol). Using a 64-second flow timeout, although *dns* constituted only 2.4% of the total packets, it constituted 27.6% of the total number of flows.

   The *www* protocol exhibits the longest flow durations for the UC-NSF PM data set (the only data set that had *www* traffic at the time of our collection). Since that time the absolute volume as well as proportion of *www* traffic at the UC-NSF backbone inflow point has grown dramatically, causing concern to NCSA/UIUC system administrators regarding the operational impact of traffic from advanced Information Resource Discovery Services (IRDS).

6. When examining the shorter flows as a function of protocol, it is evident that many of the short flows are *dns*, *gopher*, *ntp*, or *finger*. If a router could plan not to cache flows of a certain application type, it could save over a third of the memory it dedicates to maintaining flow entries.

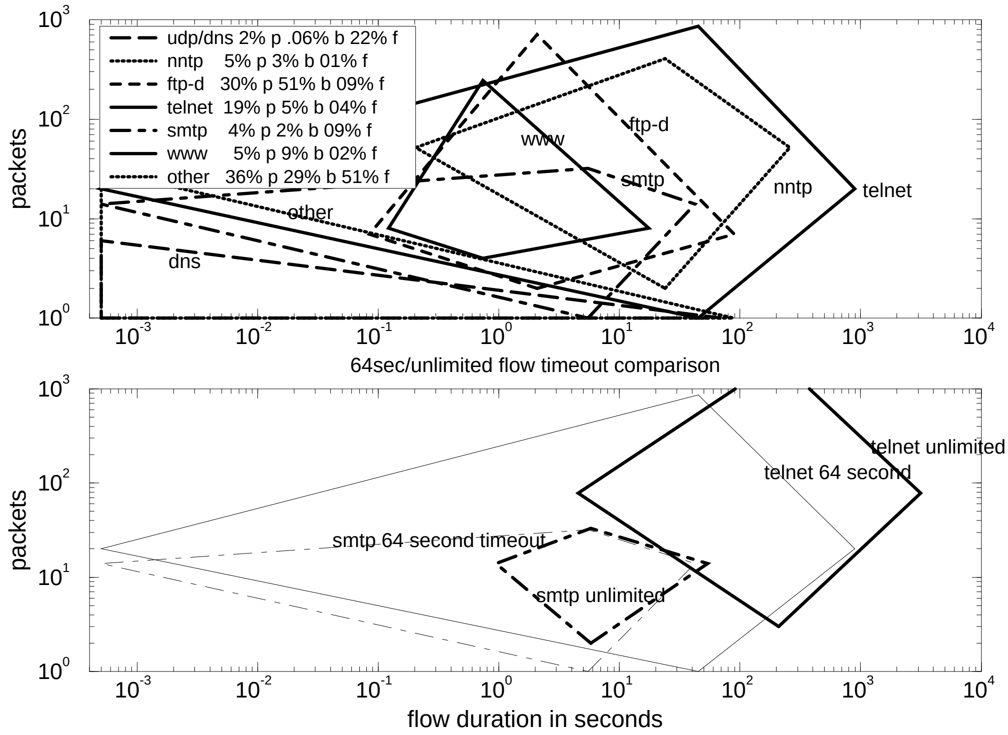

Figure 6.9: distributions in packet-duration space of host pair flows by application (UC-NSF, 64 second flow timeout) (a) top: seven common application categories (b) bottom: difference in packet-duration space between 64-second and unlimited timeout values for two applications: *telnet* and *smtp*

characteristics, including those requiring service guarantees and not using a transport protocol to delineate the beginning and end of a flow, makes it even more difficult to define an Internet flow, but also more critical. The Internet has survived, in fact succeeded beyond the wildest dreams of its initial designers and users, with a rather relaxed attitude toward systematic analysis of traffic flows. However the Internet will not be able to secure and maintain stability in the face of new traffic types and continued explosive growth without a more dedicated approach to Internet traffic analysis, the first step of which is accurate workload, or flow, characterization.

In this chapter we used the methodology described in Chapter 5 to derive metrics of individual Internet flows for several selected data sets. We presented metrics relevant to a variety of Internet issues, including route caching, accounting record maintenance, and other specialized routing algorithms. Table 6.2 summarizes key results of our study. However, describing individual Internet flows is not sufficient for understanding the aggregate traffic behavior at a systems level. We also need to find descriptors of the entire population of flows. We devote the next chapter to such metrics.

Chapter 7

Aggregate flow metrics

Our view...is that it is an essential characteristic of experimentation that it is carried out with limited resources, and an essential part of the subject of experimental design to ascertain how these should be best applied; or, in particular, to which causes of disturbance care should be given, and which ought to be deliberately ignored.

– *Sir Ronald A. Fisher*

The art of drawing conclusions from experiments and observations consists in evaluating probabilities and in estimating whether they are sufficiently great or numerous enough to constitute proofs. This kind of calculation is more complicated and more difficult than it is commonly thought to be...

– *Antoine Lavoisier*

As mentioned earlier, aggregate flow metrics are inherently attributes of a population of flows, rather than a single flow. We divide aggregate workload metrics into three classes: flow counts, flow arrivals, and locality, in sections 7.1, 7.2 and 7.3, respectively. The first class of metrics includes counts of new and active flows per unit of time. The second class of metrics characterizes the interarrival time distribution of flows. In contrast to Internet packet arrival processes, which do not seem to exhibit exponential interarrival times, the exponential model does seem to adequately characterize certain Internet flow arrivals. Finally, we present metrics of locality that reflect the non-uniformity of traffic among communicating sites, measured for example by the number of distinct addresses versus the number of total address references. As with the individual flow metrics presented in Chapter 6, each of these aggregate flow metrics have important performance implications for Internet equipment that must maintain flow state.

7.1 Flow arrivals

In this section we focus on the first class of aggregate flow metrics, specifically the number of new and active flows as arrival processes. These metrics are particularly relevant to memory and processor resource requirements for keeping state information based on flows. We described several possible flow definitions in section 5.3; we now discuss flow state requirements associated with those examples.

The simplest flow state maintenance mechanism is a routing cache, which holds entries for destination addresses with high imminent reference probability, often due to recent reference, in a separate, smaller and perhaps faster, memory. Even if one does not use a hardware cache which is typically considerably faster, keeping recently referenced records in a software cache can save time because the switching process does not have to consult the entire routing table.

Recent discussions of flow state have involved extensions to the Internet service model and routers that support it. These discussions assume a more general definition of a flow as a single stream of packets from a specified set of one or more sources to a specified set of one or more destinations that are subject to a single path selection constraint and queueing behavior in intermediate nodes. Maintaining the flow state typically will require holding an entry for each active flow which then controls forwarding behavior for packets belonging to that flow.[1]

Flow state will also be an important component of effectively using IP over the virtual circuits of an ATM network. Implementing a datagram service over a virtual circuit-oriented network service such as ATM requires setting up and tearing down virtual circuits, and accounting and charging for them, based on actual traffic activity.

For example, an ATM service provider could offer a fully connected mesh of permanent virtual circuits (PVCs) among all the entry and exit points of its ATM cloud, comparable to a current IP switching backbone. But ATM functionality allows more efficient service and bandwidth allocation by extending a simple PVC network through the use of switched virtual circuits (SVCs) for specific traffic flows, allowing multiple service qualities in the network. Supporting SVCs in real-time as needed requires establishing them upon appearance of traffic not fit for an existing PVC or SVC. For example when a packet from network A to network B arrives at A's interface to the ATM network, and an appropriate circuit between network A and network B does not already exist, the router at the inflow point would establish an SVC for this path, and then tear the circuit down if it becomes idle. This scenario imposes some VC setup overhead compared to the fully meshed PVC scenario, but saves resources if not all PVCs are necessary at all times.

In reality the general purpose infrastructure will have to take incremental steps from current IP switched networks toward an ATM cell-switched environment. For example, a wide-area backbone such as the new NSF vBNS may transition to an ATM environment using two fully connected meshes of PVCs, one for inter-supercomputer-center traffic, one for other traffic between centers and clients, with a different priority from that of the inter-center mesh. One could use the IP precedence field to signal multiple underlying service qualities, so that for example level three traffic could travel on a standard PVC, traffic below level three in precedence goes on a secondary PVC, and traffic above level three would incur dedicated SVC setup when resources are available. Parameters of flows that currently traverse wide area infrastructures are essential to engineering such configurations that can effectively replace existing dedicated point-to-point channels (e.g., T3 links). In this section we investigate how flow parameters will affect design decisions under such new technologies.

We use a one-second interval to measure flow arrival and departure rates, and then examine percentile statistics of the distribution of these counts over the course of the 3600 seconds in the hour data sets. We use the same systematic exploration of parameters that we used in Chapter 6:

- flow timeout (from 2 to 2048 seconds; a 2048-second timeout is essentially infinite for this data);
- flow specification (destination network (*dn*), destination host (*dh*), source host (*sh*), network pair (*np*), host pair (*hp*));
- the environment (five sites, measured for two separate hours each, as described in section 5.5);
- the transport or other protocol (TCP, UDP, EGP, ICMP) and application (*dns*, *nntp*, *ftpdata*, *telnet*, *smtp*, and *www*).

As a point of reference Table 7.1 lists, for each environment, flow metrics for a single set of parameters, specifically host pair flows and a 64-second timeout. The table lists for these parameters the total number of flows in the data set, the median and upper percentiles of the number of new and active host pair flows per second, and the same percentiles for the individual flow metrics from the last chapter, i.e., flow duration, packet volume and byte volume.

[1] Actually, the switch itself would likely not establish the table: for optimal performance it would simply forward cells using a table that another node is responsible for building. This latter node would know what flows it has allocated over a path and would work at path set-up time to provide a path with the required characteristics [111].

Table 7.1: percentiles of host pair flow metrics for ten one-hour data sets (64 second timeout)

Data set	total flows	object	**Median**	**95%**	**Max.**
SD-NSF am	12036	new flows (per sec)	3	8	47
		active flows (per sec)	464	716	778
		flow duration (secs)	0	328	3599
		flow volume (pkts)	2	83	83495
		flow volume (bytes)	120	7554	45970400
SD-NSF pm	30923	new flows	8	15	111
		active flows	982	1071	1128
		flow duration (secs)	0	182	3599
		flow volume (pkts)	2	62	47792
		flow volume (bytes)	160	9022	26345800
SDSC-int am	2206	new flows	0	3	36
		active flows	221	232	241
		flow duration (secs)	6	3469	3599
		flow volume (pkts)	2	292	144764
		flow volume (bytes)	148	27506	33547500
SDSC-int pm	3761	new flows	0	4	58
		active flows	289	310	328
		flow duration (secs)	5	1344	3599
		flow volume (pkts)	4	496	353352
		flow volume (bytes)	288	53813	85340400
SDSC-vis am	1129	new flows	0	2	10
		active flows	54	65	74
		flow duration (secs)	0	191	3599
		flow volume (pkts)	5	18	21073
		flow volume (bytes)	224	2411	19813000
SDSC-vis pm	2948	new flows	0	4	24
		active flows	116	183	211
		flow duration (secs)	0	277	3596
		flow volume (pkts)	5	660	231847
		flow volume (bytes)	368	211407	34977600
UCSD am	23348	new flows	6	16	81
		active flows	800	995	1128
		flow duration (secs)	0	163	3599
		flow volume (pkts)	2	43	44198
		flow volume (bytes)	132	4723	4426060
UCSD pm	39631	new flows	10	21	218
		active flows	1470	1672	1813
		flow duration (secs)	1	250	3599
		flow volume (pkts)	2	132	88085
		flow volume (bytes)	180	14071	35144000
UIUC am	30222	new flows	8	15	144
		active flows	995	1218	1356
		flow duration (secs)	1	239	3599
		flow volume (pkts)	2	91	39243
		flow volume (bytes)	129	15439	57306500
UIUC pm	57107	new flows	15	25	277
		active flows	1898	2169	2314
		flow duration (secs)	2	224	3599
		flow volume (pkts)	2	162	162394
		flow volume (bytes)	167	28012	89496000

7.1.1 Flow timeout

We first highlight the impact of the flow timeout parameter, fixing all other parameters, on the number of new and active flows for each environment. For the other parameters we select the destination network flow granularity, the UC-NSF PM data set, and consider all traffic (i.e., all protocols aggregated). We will also use the host pair flow granularity for comparison to destination network flows.

Maintaining state in a router, or virtual circuits in an ATM network, requires memory and computational resources for each flow. One objective of a router maintaining flow state is to optimize the tradeoff between maintaining state for many flows, which requires both memory for the information and search time for accessing the large state table for each packet switched, and maintaining state for few flows by means of a short flow timeout, which requires less memory but greater CPU resources and memory management effort to set up and tear down flows more often. If the timeout value is too low, flows may time out even though traffic between the two endpoints has not stopped, leading to potentially large delays and processing costs for reestablishing the flow. The analogy to virtual memory caching is *thrashing*, which will occur if flow demands are larger than available router resources and require constant closing and reopening of flows. In this section we explore how the timeout value influences the probability of thrashing.

Saran and Keshav [108] investigate one timeout strategy: timeout occurs if a packet for a flow has not appeared in the time between the last packet of the flow and the one immediately preceding it, i.e., within the last intra-flow packet interarrival time. Mankin and Ramakrishnan [112] suggest another possible strategy where the timeout value dynamically changes based on the number of currently active flows. They propose three administratively controlled variables: a minimum time; a maximum time and an adaptation factor in seconds per available flow. A flow times out once it has been idle for a time period equal to the minimum plus the adaptation factor times the number of available circuits, limited by the maximum time. The authors suggest that administrative adjustment of these variables can provide considerable flexibility in meeting the needs of a specific gateway, but do not offer any analysis, simulation, or empirical data to test the efficacy of such a scheme.

In figure 7.1 (and later in figures 7.4, 7.5, and 7.6) the upper graph provides a measure of the *flow turnover* rate, while the bottom graph indicates the number of entries required in a state table to hold entries for all active flows. The top graph in figure 7.1 plots for a single environment (UC-NSF) the median and 95th percentile of the number of new destination network and host pair flows. The medians for the new and timed out flows coincide, due to the forced flow law [113]. An implementation detail led to our use of the 95th percentile rather than the maximum of the distribution of new flows per second. In the first few seconds the number of flows is still establishing a steady state and so the hundreds of new flows for those seconds skew the distribution of the number of new flows per second. Similarly with the number of timed out flows; a few flows lasted beyond the hour interval of the data set and so were all timed out at the end of the hour, skewing the distribution of flow durations.

The bottom graph in figure 7.1 plots the median and maximum number of active destination network flows. We can use the maximum in this case since the ramp up phase does not affect this distribution. As expected, the larger the timeout, the greater the number of active flows per second, but the smaller the turnover rate as measured by the number of new and timed out flows. More specifically, the median number of active destination network flows per second using a 64-second timeout on this data set is 660, and the median number of new flows (requiring set up) is 3. Dropping to a 4 second timeout would have required maintaining a median number of only 300 active destination network flows per second but setting up a median number of 20 flows per second. The behavior of the host pair flow counts are similar although characterized by higher means, as one would expect. Using a 4-second timeout yielded 54 new flows on average and 506 active host pair flows per second. Using a 64-second flow timeout yielded approximately 2,000 for the median number of active flows per second, but trades off the greater number of active flows with a reduced setup requirement: only a median of 15 new flows requiring setup per second, and a maximum of 35 new flows per second. The contrast between the values of these metrics for host pair versus destination network flows highlights the difference in flow setup for the two flow granularities. Both cases show clearly the tradeoff between memory requirements of storing more flows at higher timeouts versus processing requirements of frequent flow setup and teardown at lower timeouts. The other flow specifications exhibit similar behavior with different parameters.

Figure 7.2 plots two measures of the imprudence of timing out flows too early. We first note that

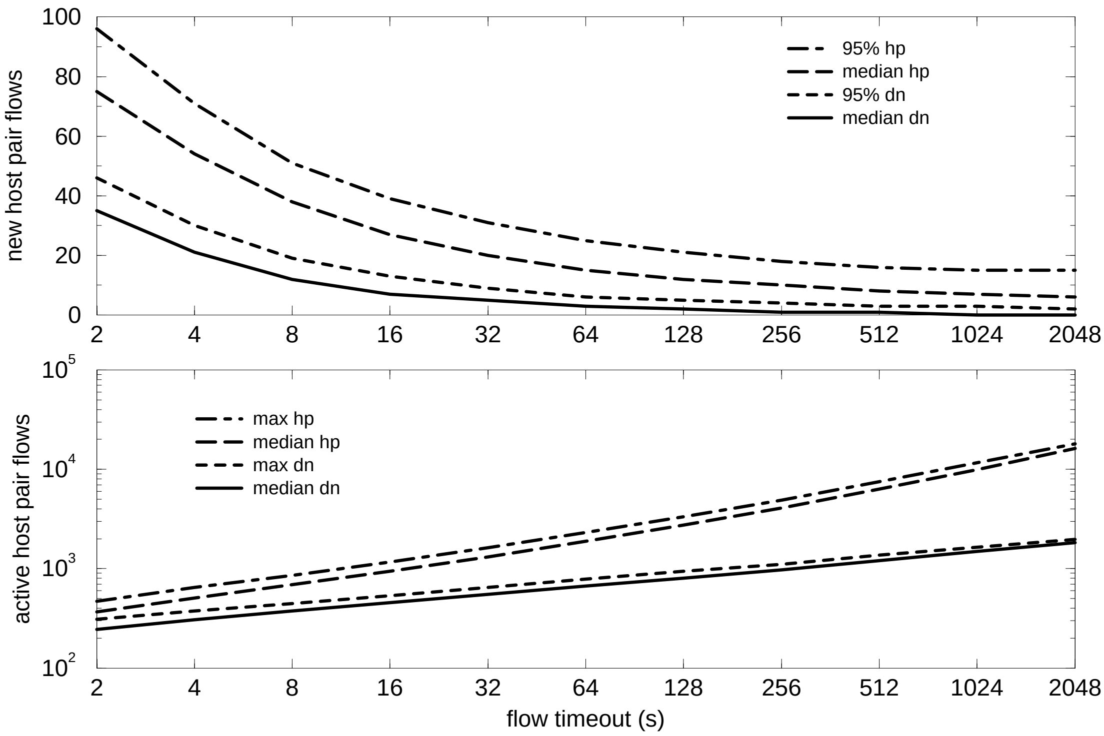

Figure 7.1: as a function of timeout value (a) top: median and 95th percentile of new destination network and host pair flows per second (b) bottom: median and maximum of number of active flows per second (UC-NSF PM)

many flows between the same host pairs occur more than once during the hour. For example, for the UC-NSF data set there were only 25,358 unique host pair flows; the different numbers in the legend of figure 6.1, and in the top line in the top graph of figure 7.2, reflect the fact that at shorter timeouts many flows will reappear multiple times in this tally.

Given this information we can better understand the implications of figure 7.2. The top graph shows the total number of host pair flows as a function of flow timeout, and the ones that were recreated within the same number of seconds as the flow timeout. The bottom graph shows the ratio of flows that were recreated within the flow timeout value to the 25,358 unique host pair flows. At a timeout value of two, each of the 25 thousand unique host pair flows is set up and torn down almost ten times *on average* during the hour; a timeout of 16 seconds brings this redundancy factor down to 2.9. Dividing by the total number of flows rather than the number of unique flows is also indicative: *at a 16 second timeout value, approximately 74% of all flows are ones that have been recreated in the last 16 seconds.* Using a 64-second timeout, half of the flows had been recreated within the last 64 seconds. Both graphs plot the x-axis on a log scale; the top graph also plots the y-axis on a log scale. This graph is consistent with figure 6.1 where lower timeouts seem to incur undue thrashing, reflected here by ratios of over 2 for flow timeouts under 32 seconds.

Figure 7.3 plots a related statistic: the mean time until flow recreation for given timeouts. This figure plots for four flow specifications (host pair, destination network, source host, source network) the mean and 95th percentile of the distribution of flow recreation times. There is a drop toward the right hand side of the graph (not shown) for larger timeout values, reflecting the fact that our data set lasted for only one hour; longer timeouts left us with less time in the rest of the hour for flows to exist. Those distributions are thus slightly skewed toward shorter flows. In general, destination network flows that are timed out are recreated up to twice as fast as timed out host pair flows, with the disparity decreasing somewhat as the timeout increases. We discuss differences among flow specifications more in the next section.

7.1.2 Flow specification

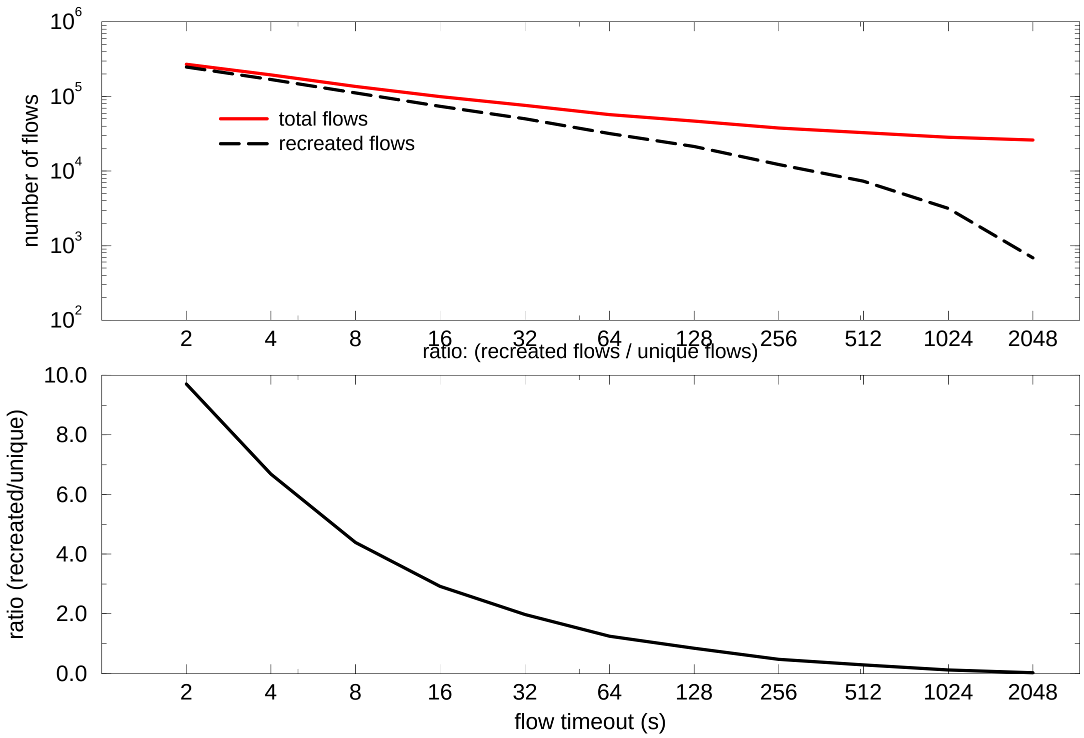

Figure 7.2: (a) top: total number of host pair flows as a function of flow timeout, and the ones recreated within the same number of seconds as the flow timeout; (b) bottom: ratio of flows that were recreated within the flow timeout value to the 25,358 unique host pair flows (UC-NSF PM)

Figure 7.4 focuses on the same environment, UC-NSF PM, but rather fixes the timeout value to 64 seconds and varies the flow specification among: destination networks; destination hosts; network pairs; host pairs. The top graph plots the median and 95th percentile of new flows per second, and the bottom graph plots the median and maximum number of active flows as a function of the flow specification. The most critical impression from these measurements is that the number of host pair flows is not much larger than the number of network pair or destination network flows, at most three times as many.

To better quantify how many more flows exist at the host versus the network granularity, Table 7.2 shows the maximum number of each of our four flow objects active during any one second, given a five minute, i.e., essentially infinite, flow timeout (so that we capture virtually all flows). The number of host pairs is between 1.6 and 2.8 times the number of network number pairs, and between 3.1 and 7.3 times the number of destination networks.[2] At backbone inflow points, the ratio of host pairs to destination hosts is typically lower than at campuses, whose ratio is in turn lower than the supercomputer center environments. These measurements are consistent with our intuition that the diversity of traffic is much greater in the wider area, as opposed to research LANs or campuses which typically favor a large number of popular hosts at a few destination networks.

A different ratio, that of host pairs to network pairs, is lower at the backbone inflow points, slightly higher in the supercomputer LAN and campus environments. Again we infer that campuses and research LANs typically have very many host pair conversations occurring to a select set of destinations, while the backbone environment exhibits less favoritism. We discuss locality metrics further in section 7.3; the main point we want to make here is that the number of host pair flows certainly does not scale on the order of the square of the number of network number pair flows, as a uniform matrix of traffic volume among connected sites implies. These measurements are auspicious for RSVP [50] and other soft-state based Internet routing schemes; i.e., maintaining host pair state in the Internet may not impose prohibitive load on the routers.

[2] This generalization excludes the SD-viz set since it is somewhat anomalous with so few flows.

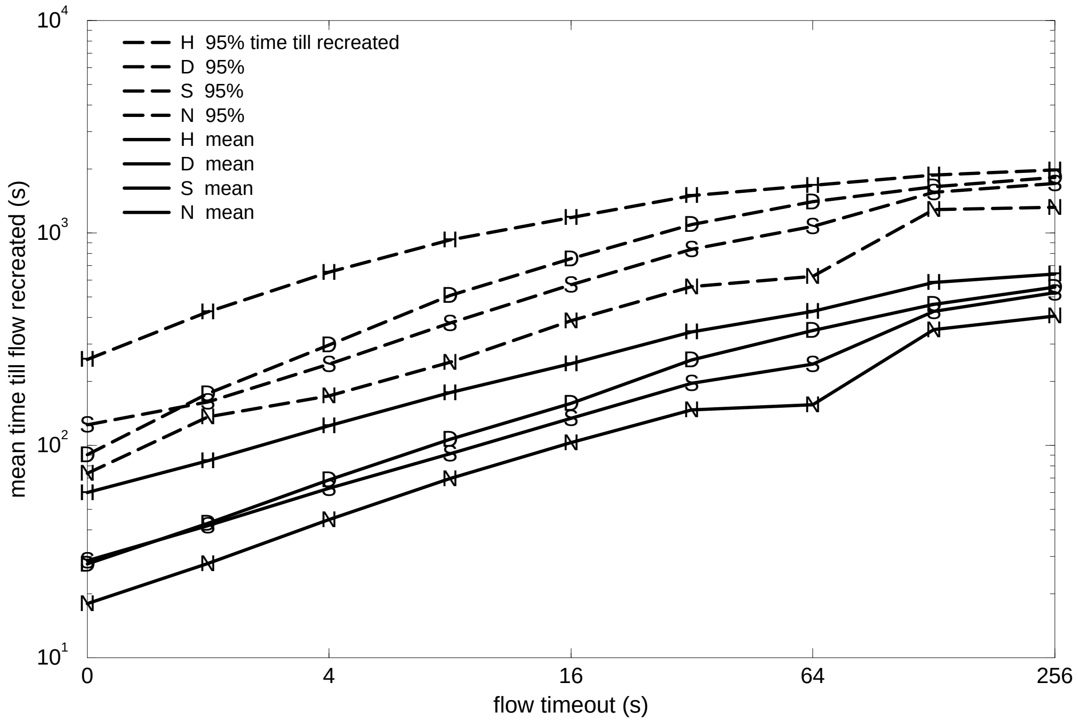

Figure 7.3: mean and 95th percentile of the distribution of times until flow recreation as a function of flow timeout for four flow specifications: host pair (H), destination network (D), source host (S), and source network (N) (UC-NSF PM)

7.1.3 Environment

In this section we focus on the differences among the environments we investigated, i.e., the different measurement locations (shown in figure 5.2). For illustration purposes we concentrated on the busier afternoon (PM) data sets, and limited the timeout to a fixed value of 64 seconds. The top graph in figure 7.5 plots the median and 95th percentiles of the distribution of the number of new destination network flows per second, the bottom graph plots the median and maximum number of active destination network flows per second. As in figures 7.1 and 7.4, the metric in the upper graph indicates *flow turnover* rate, while the one in the lower graph indicates the size of a flow state table.

Figure 7.6 plots the same metrics with the same ordering of environments, for the host address pair flow specification. We order the x-axis by the number of active destination network number flows, starting

Table 7.2: maximum number of active flows per second using a five minute flow timeout

data set	time	host pairs	dst hosts	net pairs	dst nets
SD-viz	AM	107	50	9	6
	PM	352	112	15	9
SDSC	AM	292	102	136	40
	PM	440	166	174	62
SD-NSF	AM	1662	982	995	5.3
	PM	2855	1834	1515	688
UCSD	AM	2410	986	888	364
	PM	3900	1495	1369	532
UC-NSF	AM	2867	1710	1651	809
	PM	5377	3066	2900	1162

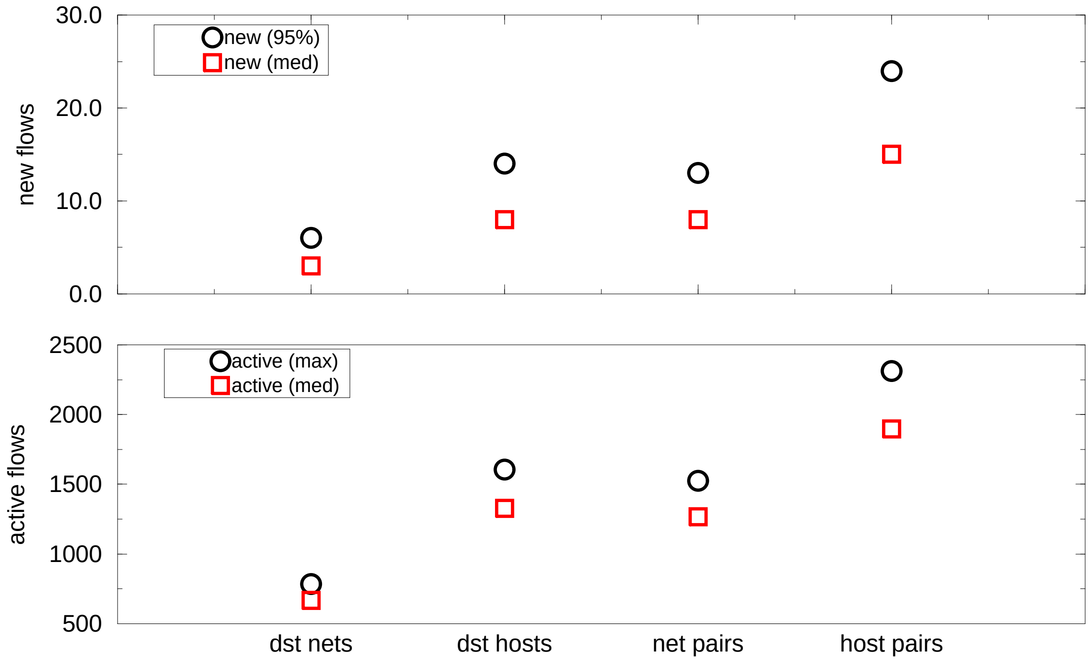

Figure 7.4: for four flow specifications (a) top: median and 95th percentile of new and flows per second (b) bottom: median and maximum of number of active flows per second (UC-NSF PM, 64 second timeout)

with the SD-viz Ethernet which had the fewest active destination network flows, and ending with the two NSFNET backbone inflow locations, SD-NSF and UC-NSF, which had the greatest number of active flows. The turnover rate of both destination network and host pair flows generally varies with the total number of active flows, but is about two orders of magnitude less for the parameters we use here. Approximately the same ordering and relative differences in turnover prevail for host pair flows, except for the UCSD campus backbone FDDI ring, which has a significantly higher ratio of host pairs to destination networks than the other environments. The median number of host pair flows is around 6.5 times the median number of destination network flows, versus a ratio of less than 3 for UC-NSF (see also table 7.2). Since the UCSD data set includes intra-campus traffic as well as traffic destined to off-campus locations, the greater number of hosts communicating for a given destination network is not surprising.

One important application of these flow assessment metrics is in configuration of an ATM environment, where they indicate the number of virtual circuits (VCs) an ATM switch would have to simultaneously maintain and establish, based on a per-second measurement granularity. Another related application is in network accounting, where they would correspond to the number of accounting records a statistics collection process would have to maintain or create. Note that even back in March 1993 for a busy entrance point into the NSFNET backbone, the median number of active host pair flows per second using a 64-second timeout was approximately 2000, with a median of 15 new or deleted flows per second and a maximum of 35 new flows per second. An ATM router would thus have to maintain 2000 virtual circuits on average, and be able to set up approximately 35 flows per second without impeding switching performance in this environment. Of course, the level of traffic has been rising steadily since that time, so these benchmarks will continue to increase.

Eventually a metric as simple as a flow count, which measures the degree of aggregation, may be too limited to measure the impact of flows on the overall workload in a specific environment. To explore the interaction between the number of flows and the total traffic volume, we depict a two-dimensional profile, using the median number of active flows per second for the hour in one dimension, and the median per-second traffic volume in packets in the other dimension. Figure 7.7 plots this two-dimensional metric for each environment, using host pair flows and a 64 second timeout.

As expected, the flow-traffic volume product is highest for the UC-NSF PM environment, which

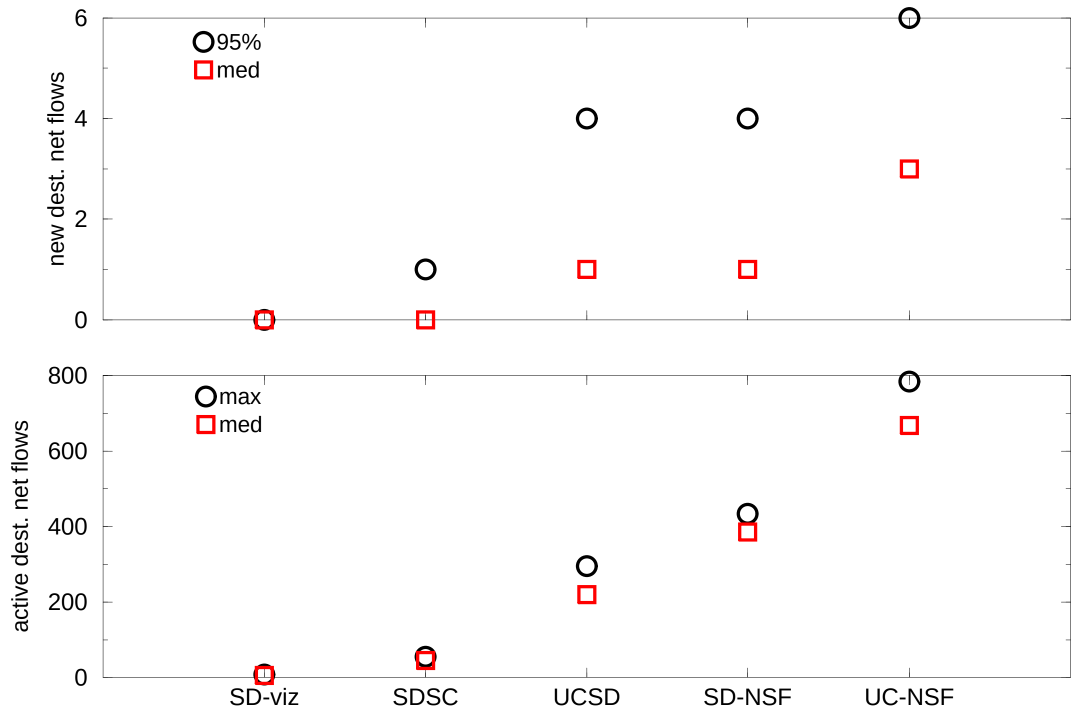

Figure 7.5: for five environments (a) top: median and 95th percentile of new and destination net flows per second (b) bottom: median and maximum of number of active flows per second (64 second timeout)

has both high flow aggregation and high utilization. UCSD PM is the second highest, aggregating many host pair flows at a fairly high per second packet rate. Dividing figure 7.7 into quadrants leaves these two in the upper right with everything else toward the lower left. This use of the flow-traffic volume metric can help assess multiple dimensions of the workload of a specific networking environment. If the value representing a specific environment on this graph moves up the diagonal of this graph, then the compounded characteristics are fairly similar, just more of the same. On the other hand, if the packet volume rate (x-value) stays the same, and the number of flows (y-value) significantly changes, then the traffic composition is changing. In particular if a service provider notices that on his network the packet volume rate on the x-axis increases while the number of flows on the y-axis decreases, he has a serious problem, as it indicates that fewer flows are imposing more traffic, and he loses the ability to aggregate among his customers, and thus potential revenue. The worst quadrant to be in for a datagram environment that relies on heavy aggregation, and the quadrant toward which multimedia flows will tend to shift an environment, would be the lower right.

Our measurements indicate that at least in the environments we studied, current IP traffic still consists more of short transaction behavior rather than longer term flows. The short packets and short flows together shed doubt on a strategy of optimizing for long flows that are in fact the minority case. However in the next section we discuss trends that may change this situation, and have ominous implications for operators of Internet components.

7.1.4 Higher layer protocol

We next explore how the flow type affects aggregate flow metrics. We discussed in section 6.4 how differences in flow profiles by protocol may lead a designer of an accounting scheme to choose not to charge for small flows at all, either by not maintaining records for certain flow types, or by having a special cache for (free) flows until they exceed a certain packet volume. Small flows will also have a dramatic impact on ATM or other link-level circuit multiplexing policies, if setup or teardown costs more than the entire flow. In this section we extend these results to aggregate flow metrics, using a two-dimensional perspective to show insight into the effect of the protocol on router workload.

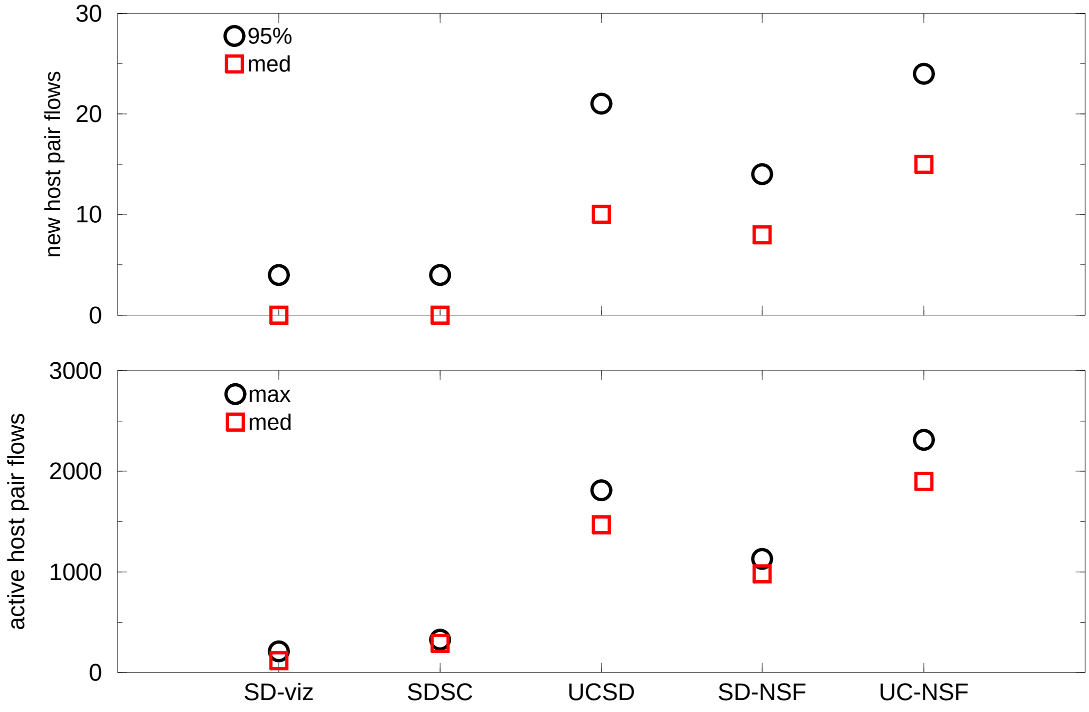

Figure 7.6: for five environments (a) top: median and 95th percentile of new and host pair flows per second (b) bottom: median and maximum of number of active flows per second (64 second timeout)

To illustrate our point in the next two graphs we use a newer data set from March 1994 at the UC-NSF site in order to get enough IPIP and *www* traffic to profile their behavior. Figure 7.8 uses a 64 second flow timeout in the UC-NSF environment and plots the number of flows of a given packet type versus the mean number of packets per that type of flow. The graph highlights the range from extremely low volume flows of *dns* to short duration transactions of *smtp* and *www*, to heavier flows carried by file transfers and interactive communications. Note that the otherwise transaction-like *nntp* protocol exhibits a relatively high average flow volume.

Figure 7.9 plots the same data but uses a log scale to include traffic carried via a more recently popular protocol, IPIP, used for carrying video and audio multicast flows on the Mbone.[3] The lower right quadrant of this graph emphasizes the impact of continuous-stream protocols, where relatively few flows will have packet and byte volumes orders of magnitude higher than conventional applications. Applications in the lower right quadrant of this figure will fundamentally change the nature of Internet environments, bringing them toward the lower right quadrant of figure 7.7. The sustained high flow volume of such traffic flows will impede the ability of a network to aggregate among the vast number of flows that most current Internet environments carry. Such flows also pose a problem for the usage-insensitive charging scheme of the current Internet. As such, continued flow assessment will be critical in the next few years, as the cross-section of traffic at many Internet components changes.

We illustrate how the number of packets per flow constrains a router that must create state for a population of flows. Assuming that it takes p instructions to forward a regular packet, r instructions to forward a packet for which special flow information exists, q instructions to install a flow, and a flow comprises n packets on average, then maintaining flow state makes sense when

[3] IPIP is a generic protocol to allow for tunneling IP, i.e., carrying IP packets within IP packets. This facility is useful for creating a virtual IP infrastructure on top of a routed IP network. An example application, in popular use today, is the creation of the Mbone [109], a multicast service over an infrastructure that does not itself support multicasting. The implementation creates tunnels of multicast paths across the unicast infrastructure, while multicasting or even replicating packets at the tunnel exit point to create the multicast effect. Such a facility is today most useful for applications where one sender sends to many receivers who do not have strict requirements for path reliability or sequencing.

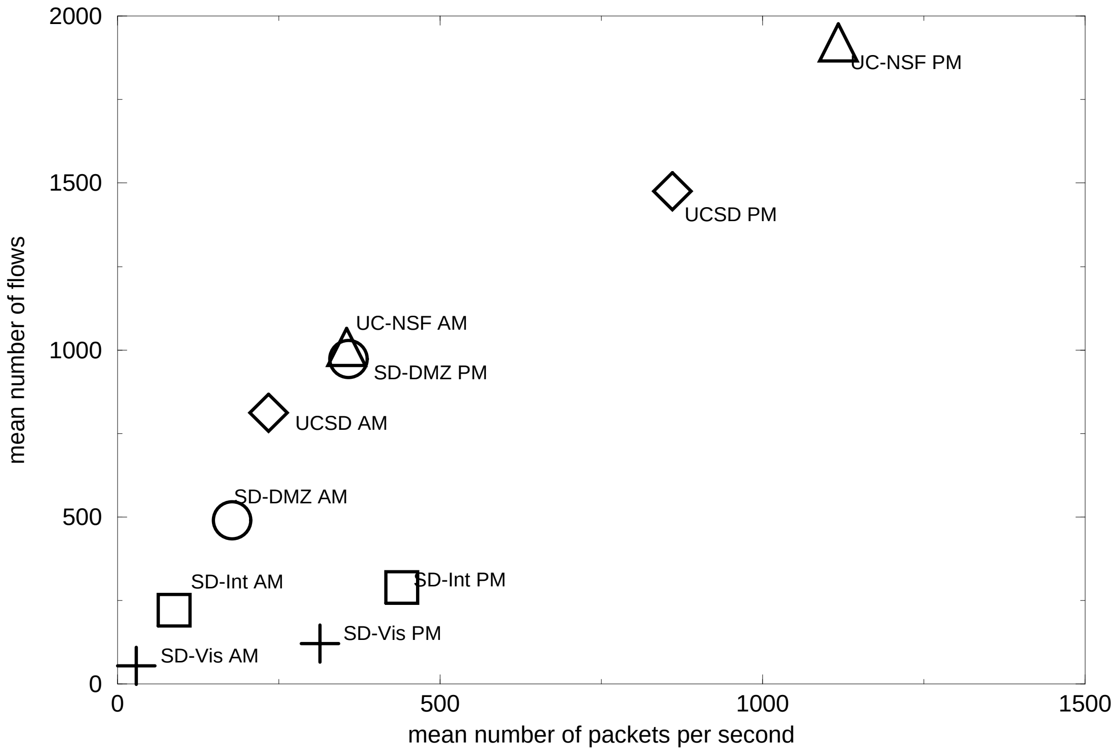

Figure 7.7: mean number of host pair flows per second versus mean per-second packet rates for each environment (64 second flow timeout)

$$p + q + (n-1) * r < n * p$$

or

$$q < (p - r) * (n - 1)$$

.

For example, if $p = 600$ instructions, $r = 300$ instructions, and $n = 10$, then flow installation must require less than $q = 2700$ instructions. A router designer could take advantage of empirical data such as in figure 7.9 to determine which flow establishments to avoid, or a preferred PVC/SVC configuration of an ATM network. For example, a PVC substrate could still offer standard service for background traffic, while other traffic flows would travel on dynamically established switched circuits based on actual flow requirements. A high priority video stream could travel on a separate switched virtual circuit (SVC), while an aggregation of low priority video streams and other services whose flow profiles are in the upper left of figure 7.9 may travel on a PVC network allocated for such background or non-premium traffic.

7.2 Flow interarrivals

We presented metrics of flow *arrivals* in the previous section. In this section we focus on the distribution of *interarrival* times between flows. Figure 2.4 depicts the difference between the arrival process as a counting process and the interarrival times. Several studies have found evidence that Internet packet arrivals do not seem to exhibit Poisson behavior [28] [3] [32] [23] [58]. In contrast, as we will see, the Poisson model does seem to adequately characterize Internet flow arrivals as defined by certain parameters.

Previous attempts to characterize Internet arrival processes have concentrated on traffic by component, e.g., *telnet*, *ftpdata*. As mentioned in section 2.4.7, Caceres *et al.* [6] provide evidence that characteristics of an instantiation of a specific TCP application do not depend on the environment, but that characteristics of the conversation arrival process itself do. They admit that they were "unable to form a realistic and network-independent model of conversation arrivals, since the arrival parameters depend on geographic site, day of week, time of day, and possibly other factors" [114]. Paxson [61] provides further evidence that traffic patterns vary greatly, both over time and more so from site-to-site, not only in traffic cross-section but also in connection characteristics.

Paxson and Floyd [58] use fifteen wide-area traces to investigate the extent to which TCP arrival processes (session and connection arrivals, *ftpdata* connection arrivals within *ftp* sessions, and *telnet* packet arrivals) are Poisson. They find that user-initiated TCP session arrivals, e.g., remote login and file transfer, reasonably reflect Poisson processes with fixed hourly rates, but other connection arrivals are less convincingly Poisson. Furthermore, they find that modeling *telnet* packet arrivals as exponential inaccurately reflects *telnet* burstiness, although the empirical interarrivals of *tcplib* preserves burstiness over many time scales. Finally they determine that *ftpdata* connection arrivals within *ftp* sessions come bunched into "connection bursts," the largest of which are so large that they completely dominate *ftpdata* traffic.

Breaking traffic up into components such as the above studies is helpful when the dominant application in the traffic cross-section at a given site will overshadow many characteristics of overall traffic measurements, and thus characterizing the dominant application is very close to characterizing the overall workload. Examining the distribution of conversation interarrival times by application at various sites is also relevant because applications may differ by site, e.g., the arrival characteristics of *nntp* traffic on a regional network may differ from those on a larger backbone, due to how the *nntp* protocol distributes news.

However we also feel it will be important to characterize the aggregate arrival process, regardless of transport protocol or application. This approach will be increasingly relevant as different types of Internet traffic proliferate, decreasing the proportion of traffic carried by traditional protocols. The model in [6] provides parameters for only five TCP applications (*ftpdata*, *rlogin*, *nntp*, *smtp*, *telnet*), which represent a decreasing proportion of Internet traffic (see figure 3.10). These five applications represent only about half of current NSFNET traffic in packets, about 63% in bytes, decreasing gradually over the last few years [80]. Popularity of the Mbone, which uses IPIP, and also of alternative non-IP protocols will contribute further to Internet traffic diversity. Between January 1993 and March 1994 the monthly volume of IPIP traffic on the NSFNET backbone increased by more than two orders of magnitude. By March 1994 IPIP accounted for more than 9% of overall NSFNET traffic, up from less than 0.2% in January 1993.

These factors motivate our attention in this section to the interarrival processes of all Internet flows, regardless of flow type. Consistent with previous chapters, we systematically investigate the effect on flow interarrivals of varying relevant parameters one at a time:

- flow timeout (2 to 2048 seconds)
- flow specification object
- environment (five different locations ranging in utilization and aggregation)
- protocol/application

7.2.1 Flow timeout

We first vary the timeout, holding other factors constant. Figure 7.10 shows the distribution of host pair flow interarrival times using a range of flow timeouts for the UC-NSF PM data set. The legend lists for each timeout value the resulting number of flows and the mean flow interarrival time. Since lower timeouts incur a larger number of flows via more frequent flow deletion and subsequent recreation, they yield smaller mean flow interarrival times. The number of flows during the hour for 4, 64, and 1024 second flow timeouts were 195.6×10^3, 57×10^3, and 28.6×10^3, respectively; the mean flow interarrival times were 18, 63, and 125 milliseconds, respectively.

In fitting these data to exponential distributions we exclude the first point of the data set, that for the smallest interarrival time, due to an implementation detail described in Appendix 5.B. The figure plots lines through the exponential fits of each adjusted data set. We use R^2, the square of the correlation coefficient between the fitted and actual values, as a metric of the goodness of fit. The three adjusted data sets have R^2 values of 0.998, 0.981, 0.940, respectively. Censoring the upper 5% tail of the distribution did not have a significant effect on the exponential fit.

Table 7.3: statistics for flow interarrival time distributions: number of flows in data set, R^2 measure of fit to exponential distributions excluding lowest interarrival time data point, mean flow interarrival time, and number of flows in data set

distribution	**# flows**	R^2	**mean (ms)**	total flows
UC-NSF PM, 4 sec, hp	195630	.998	18	195630
UC-NSF PM, 64 sec, hp	57106	.981	63	57106
UC-NSF PM, 1024 sec, hp	28629	.940	125	28629
UC-NSF PM, 4 sec, dn	77443	.990	46	77443
UC-NSF PM, 64 sec, dn	10362	.734	347	10362
UC-NSF PM, 1024 sec, dn	2736	.539	1314	2736
UCSD PM, 64 sec, pp	92707	.370	39	92707
UCSD PM, 64 sec, hp	39630	.547	91	39630

We also tried fitting these data sets to Erlang/Gamma distributions, but the resulting fitted parameters were always close to 1, between .95 and 1.2, indicating that the exponential function is sufficiently descriptive for these distributions. Table 7.3 presents summary statistics on the goodness of fit of the exponential distributions to these data sets. The table also includes statistics for exponential fits of the distribution of destination network interarrivals at the same range of timeouts, and of the UCSD port quadruples and host pair data sets using a 64-second timeout. These last two data sets were not as convincingly exponential according to the goodness of fit metric; we discuss their behavior below.

7.2.2 Flow specification object

Using a 64 second flow timeout, figures 7.11 and 7.12 show the cumulative probability distributions and probability density functions, respectively, of flow interarrival times for several different flow specifications: host/port quadruples, host pairs, destination hosts, network pairs, destination networks, source hosts, and source networks. We used the UCSD data set to highlight the effect of this parameter because the unidirectional collection of the UC-NSF data set (see section 5.5) yields somewhat anomalous behavior, with the number of source host flows and source network flows considerably lower than had we collected data going in both directions. The UCSD PM campus backbone data set, for which we collected data going in both directions, reflects much closer distributions between source and destination host flows.

The data indicate that the interarrival time distribution of network pair flows is very near to that of destination host or source host flows. Maintaining flow state, including establishing flow records upon arrival of new flows, would thus impose similar workload whether the flow is defined by network pair, source host or destination host. For each of these specifications, approximately 60% of the flows arrive within 200 milliseconds of the previous flow. Flow state for host pair or host/port quadruple flows would impose a higher benchmark, where approximately 90% of the flows arrive within 200 milliseconds of the last one; state for destination network flows would impose a lower benchmark, with only about 26% of the flows arriving within 200 milliseconds of the last one.

Figure 7.12 shows the non-cumulative version of figure 7.11, i.e., the probability density functions. Table 7.3 lists the measure of exponential fit of the port quadruple and host pair distributions for this UCSD data set. We were less successful in fitting these flow arrivals, regardless of specification, to exponential distributions than we were for the UC-NSF PM data sets above. due to the lesser degree of aggregation, or number of flows. Two factors reduce the number of flows, using a 64 second rather than a shorter timeout, and the fewer flows in the UCSD data sets than in the UC-NSF one for which we fitted exponential curves in section 7.2.1.

7.2.3 Environment

Figure 7.13 shows the empirical distributions of interarrival times between host pair flows using a 64-second flow timeout for the busy hours of all five environments. They are close in shape, although the distributions of the transit nodes, UC-NSF and SDSC-NSF, are shifted toward shorter flow interrival times, consistent with the larger number of flows for those data sets during the hour. As with the data in figure 7.12, there is not enough aggregation in these data sets to secure reasonable exponential fits.

7.2.4 Higher layer protocol

In sections 6.4 and6.5 we discussed the effect of the traffic type, or higher layer protocol carrying the IP traffic. We concluded there that flow packet volume and duration are correlated with the application or transport protocol of the flow, so that the traffic type could influence the decision to establish a flow state entry or an ATM switched virtual circuit (see figure 6.9 and table 6.1). Transaction traffic, such as *dns*, *gopher*, and *ntp*, and many unknown traffic flows, often consisting of a single packet, would ideally not incur circuit establishment, but rather travel along a background PVC mesh. Placing flows with a high volume-duration product, e.g., long term file exchanges, video, or audio sessions, on SVCs can lessen their burden on a standard PVC that aggregates many flows. In this section we explore the effect on the flow interarrival time distribution if a router implemented such a policy, i.e., separating the distributions of flow interarrival times into classes according to their status in a PVC/SVC configuration.

Our data indicates that most conventional applications consist of very few packets, implying that switched circuits may only be necessary for traffic of a different priority or specific quality of service (QoS) requirements or for flows with continuous sustained bandwidth demand that would unduly hinder other flows sharing the same PVC, e.g., video sessions.[4] Our traffic traces are not optimal for this experiment because SVCs are most conducive to flows that continuously demand high bandwidth, of which there are very few in our data sets.

Nonetheless we offer an example of the potential impact of separating traffic into rudimentary service classes, using the TCP/UDP port information as in sections 6.4 and6.5. Figure 7.14 shows the normalized flow interarrival time distributions for these seven categories of flows, using a 64 second flow timeout. The graph shows that, as a group, *nntp* flows tend to have the longest interarrival times, consistent with the period of network news distribution. Although *www* flows also seem to have longer flow interarrival times for this data set, we attribute this to the relatively low number of *www* flows in this data set; we expect their flow interarrival time distribution to look quite different today. The smallest interarrival times seem to characterize the *udp/dns* and *other* categories of flows, consistent with the large numbers of these types of flows we saw in section 6.4. The remaining categories, *ftpdata*, *telnet*, and *smtp*, have more typical distributions, in between those of the very frequent *dns* flows and the less frequent *nntp* flows on figure 7.14.

7.3 Flow locality

In Chapter 6 we discussed how the volume and durations of individual flows will influence the utility of maintaining state about them in a router. In this section we discuss another factor which affects utility, traffic locality. In this section we confirm and extend the results of traffic locality studies outlined in section 2.4.3, and how locality may offset the disadvantage of the tendency toward brevity of Internet flows.

Current interest in traffic locality derives primarily from its potential use to optimize forwarding mechanisms in network switching nodes. The rapid proliferation of IP network numbers, with a simultaneous requirement for fast switching performance, renders critical the optimization of routing table size, potentially through caching routing information for flows that exhibit a high probability of expected future reference. In addition to the full routing tables in the main processor on each T3 NSNFET backbone node, each 960 based controller subsystem also has a full routing table so that they can forward packets without intervention from the main processor. The subsystem must communicate with the main processor in order to obtain updated routing information.

Although the NSS cards do not implement a hardware-supported cache, they do have two separate memory stores: one for all routes in a highly compact format, and a smaller store of recently used routes in a less compact format better suited to fast search. The miss penalty for this type of cache is quite low because the forwarding card does not have to back to the main processor to service a cache miss. A hardware supported cache could potentially offer an even greater advantage.

[4] Whether a video session would receive a VC of lower or higher service quality than other flows is largely a non-technical issue and beyond the scope of this thesis.

Any caching mechanism requiring intervention by another processor incurs substantial performance penalties upon cache misses. If the NSSs were to implement a caching strategy, the traffic rate and composition at many Core NSSs and External NSSs (see section 3.2.1 for description of the NSFNET backbone architecture) would require a very low cache miss rate. The routing table size was in excess of 20,000 routes as of February 1994, although the CNSSs aggregate more flows than the ENSSs, since they need information for all potential destination networks. In contrast ENSSs need routing information only for destinations reached through that ENSS. Locality assessment on a per-ENSS basis would thus be important to cache configuration.

One metric in particular highlights the potential advantage of a cache: the depth of a stack of network addresses. The stack depth is a measure of how many references to other addresses occurred between successive references to a given address. Figure 7.15 shows the probability distribution of the stack depth for our five data sets during a work day hour. The x-axis shows the stack depth, and the y-axis shows the probability of reference to an address at a given depth in the stack. The data shows that for the hour we collected data in the SDSC visualization laboratory, there was a 90% chance that the destination network address of a given packet was the same as that of the immediately previous packet. For the SDSC internal network, there was an 80% chance that the address of a packet is the same as one of the last two packets. For the UCSD campus network, 19 references of address history is enough to account for 80% of the packets, or in other worse 80% of packets are to destination network addresses seen within the last 19 unique address references. The backbone inflow points of course require going deeper into the stack. However it is still noteworthy that given thousands of possible destination network addresses, 60% of the traffic across the San Diego and Illinois inflow points are to destinations that were referenced within the 30 and 90 previous address references, respectively.

Our data sets confirm the established phenomenon of favoritism discussed in section 2.4.3 and 3.5.3, including the contrast between local and wide area network environments. As expected, a departmental LAN such as SD-viz exhibits a tendency to communicate with relatively few destinations, with much of the traffic confined to the local environment; in fact more than 90% of the traffic goes to a single local destination. If our data sets are representative, supercomputing centers tend to have a moderate to low number of client networks; two destination networks account for 90% of the traffic which SDSC sent during the busy collection hour. University campuses have an even larger clientele; 20 destination networks accounted for 90% of the UCSD traffic. Major traffic aggregation points, such as the SD-NSF and UC-NSF NSFNET backbone access points, require significantly more destinations to account for 90% of outbound traffic: in these data traces the 90% mark required more than 100 networks for SD-NSF and about 200 for UC-NSF.

In section 2.4.3 we described a number of studies indicating that caching offers strong benefits for local area network gateways. Obviously environments that do not aggregate a lot of traffic between different locations would be the easiest in which to maintain flow state, but our measurements indicate that even environments of much wider scope are conducive to flow state maintenance mechanisms. Of course the cost of establishing flow state for single packet or very small flows, as we discussed in sections 6.5 and 7.1.4 could potentially cancel out the benefit gained from exploiting locality in a cache. But since certain situations will require flow state state regardless of the flow volume, even if only for routing information, we rather view the glass as half full: locality can in fact offset the cost that so many short flows impose on stateful routers. For example, many single or few-packet flows may be directed to a few locations, e.g., nameservers. Configuration of PVCs among those sites will benefit from empirical locality data.

7.4 Conclusion

In this chapter we have used the methodology described in Chapter 5 to derive metrics of aggregate Internet flows. Table 7.4 summarizes key results of our study of aggregate flow metrics. We have shown metrics of the quantity and turnover rate of flows, metrics which are relevant to route caching strategies in network switching equipment. We have also expressed metrics of flow interarrival times and locality, which have implications for queueing algorithms that require separate queues for each flow [52] [50] or for an accounting statistics collection agent that maintains host-to-host accounting records. One important result of our measurements is that the number of host-host pair flows appears far less than proportional to the square of the number of network number pair flows, as a uniform matrix of traffic volume among

Table 7.4: key results of aggregate flow profiling for ten selected one-hour data sets

1. For the data sets we studied, the number of host pair flows is only two to three times the number of network pair or destination network flows.

2. The number of flows a router must be able to maintain, create and delete in a state table depends on the timeout value and environment, but for a busy entrance point into the NSFNET backbone, the median number of active host pair flows per second using a 64-second timeout is approximately 2000, with a median of 15 new or deleted flows per second, and a maximum of 35 new flows per second. An ATM router would thus have to maintain 2000 virtual circuits on average, and be able to set up approximately 35 flows per second without impeding switching performance in this environment.

3. The tradeoff between timing out flows too early and leaving them to chew up memory resources is significant. For our busy backbone data set, at a timeout value of two, each of the 25 thousand unique host pair flows is set up and torn down almost ten times *on average* during the hour; a timeout of 16 seconds brings this redundancy factor down to 2.9.

4. Our measurements indicate that, at least in the environments we studied, current IP traffic still consists more of short transaction type traffic rather than longer term flows. The short packets and short flows together shed doubt on a strategy of optimizing for long flows that are in fact the minority case. However we note that many new applications may change this characteristic of Internet environments, as they introduce traffic flows with different behavior, particularly real-time continuous media flows, which tend to exhibit greater duration and flow volume. Such a shift in the aggregate flow profile will threaten the integrity of the current datagram Internet which relies on high aggregation among many bursty sources.

5. The interarrival time distribution of Internet flows as defined by certain parameters fits an exponential distribution well, with goodness-of-fit as measured by r^2 of up to 0.998 (using a 4-second timeout).

6. Flow interarrival time distributions are similar in shape among the environments we studied, although the distributions of the transit nodes, UC-NSF and SDSC-NSF, are shifted toward shorter interarrival times, consistent with the higher number of flows per unit time for those environments.

7. The data sets we examined exhibited significant flow locality, confirming results of several previous studies. For the SDSC internal FDDI LAN, there was an 80% chance that the address of a packet was the same as one of the last two packets. For the UCSD campus, 19 references of address history was enough to account for 80% of the packets. The backbone inflow points require more history to account for the same percent of traffic, but it is still noteworthy that given hundreds of thousands of possible destination network addresses, 60% of the traffic across the San Diego and Illinois inflow points was to destinations that were referenced within 30 and 90 previous address references, respectively. Obviously those environments that do not aggregate a lot of traffic between different locations would be the easiest in which to maintain a cache or some other state mechanism, but our measurements indicate that even environments of much wider scope are conducive to flow state maintenance mechanisms.

connected sites implies. This phenomenon is prominent even in wide area environments which aggregate among a large number of users, and bodes well for routing methodologies requiring end host pair state. Traffic locality can compensate somewhat for the brevity of such a large proportion of Internet flows that we found in Chapter 6. Understanding these interactions, i.e., how individual flows and the aggregate flow profile influence each other is essential to securing Internet stability, and requires ongoing flow assessment to track changes in Internet workload in a given environment. The methodology we described in Chapter 5 can form a complementary component to existing operational statistics collection, yielding insights into larger issues of Internet evolution, i.e., how environments of different aggregation can cope with resource contention by an ever-changing composition and volume of flows.

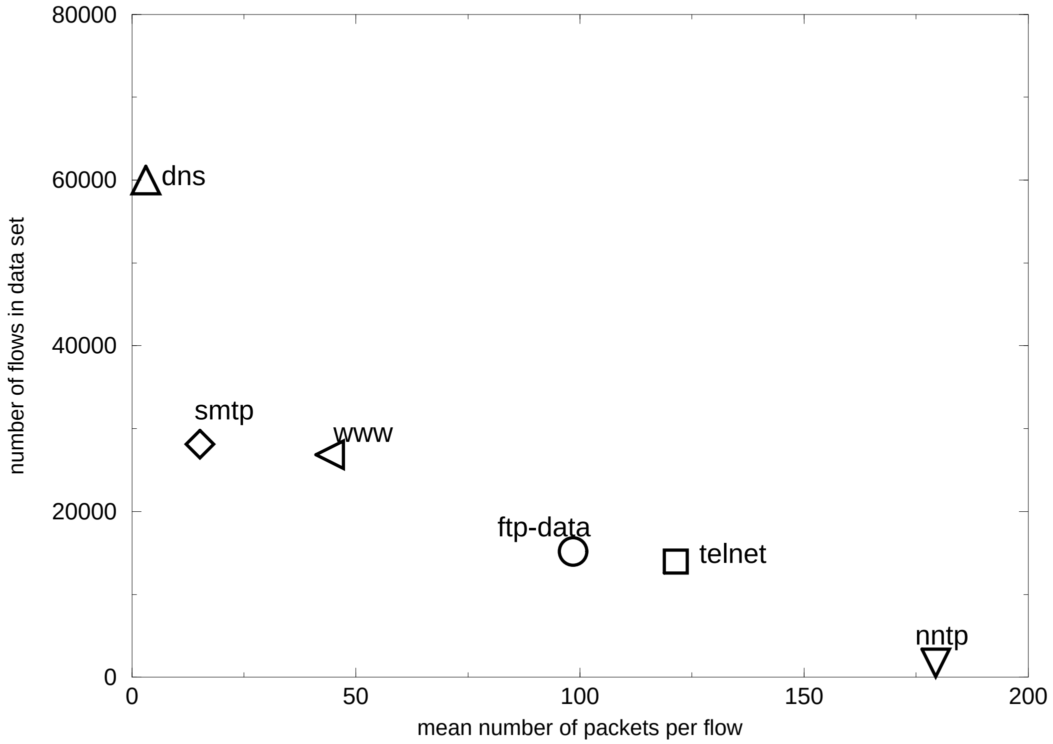

Figure 7.8: number of flows per protocol versus packet volume per flow (UC-NSF 1994, 64 second flow timeout)

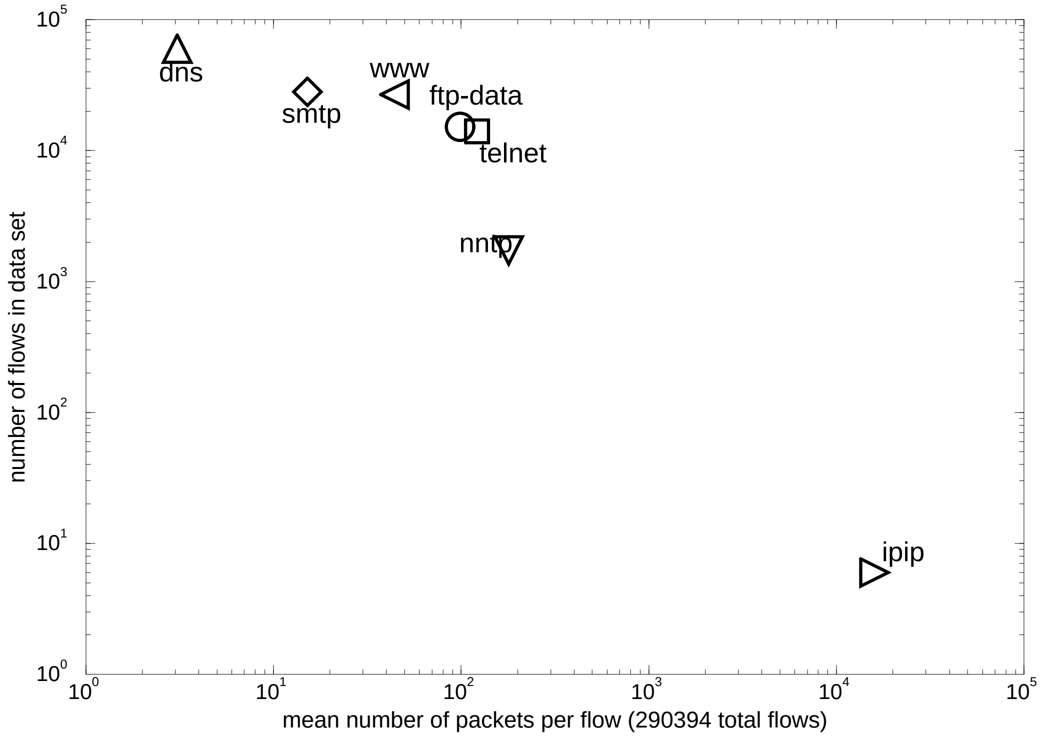

Figure 7.9: number of flows per protocol versus packet volume per flow including two newer protocols: IPIP (which includes Mbone) and *www* traffic (UC-NSF 1994, 64 second flow timeout)

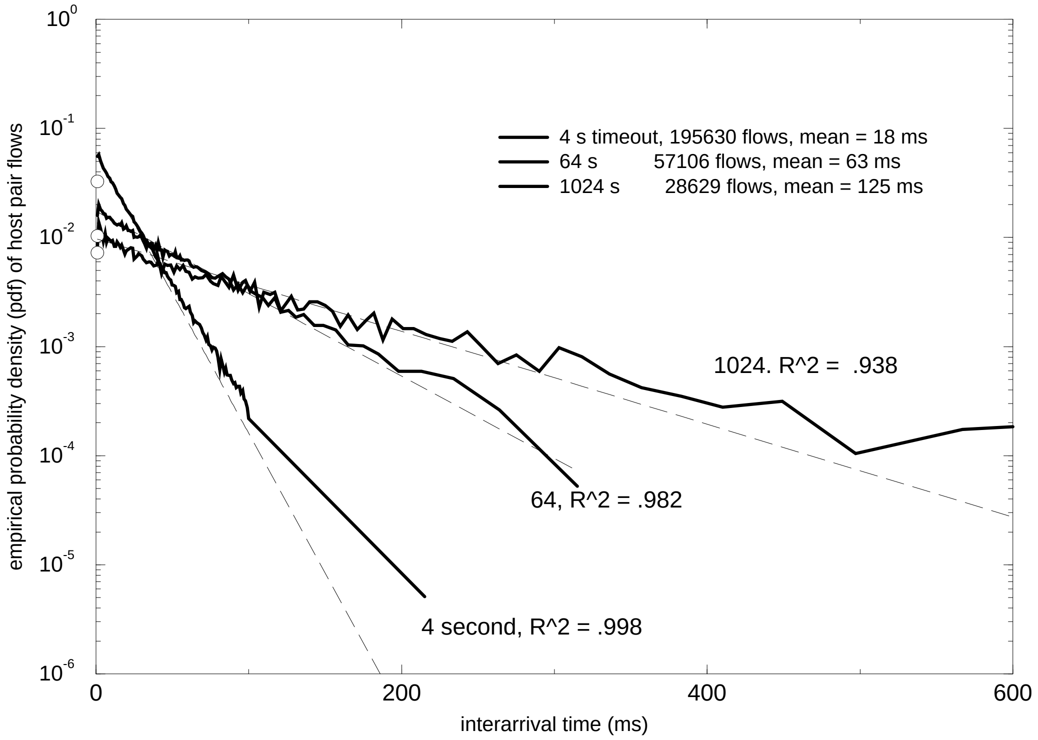

Figure 7.10: empirical probability density functions, exponential fits, and associated goodness of fit metrics for host pair flow interarrival times for three flow timeouts: 4, 64, 1024 seconds (UC-NSF PM, first data point corrected)

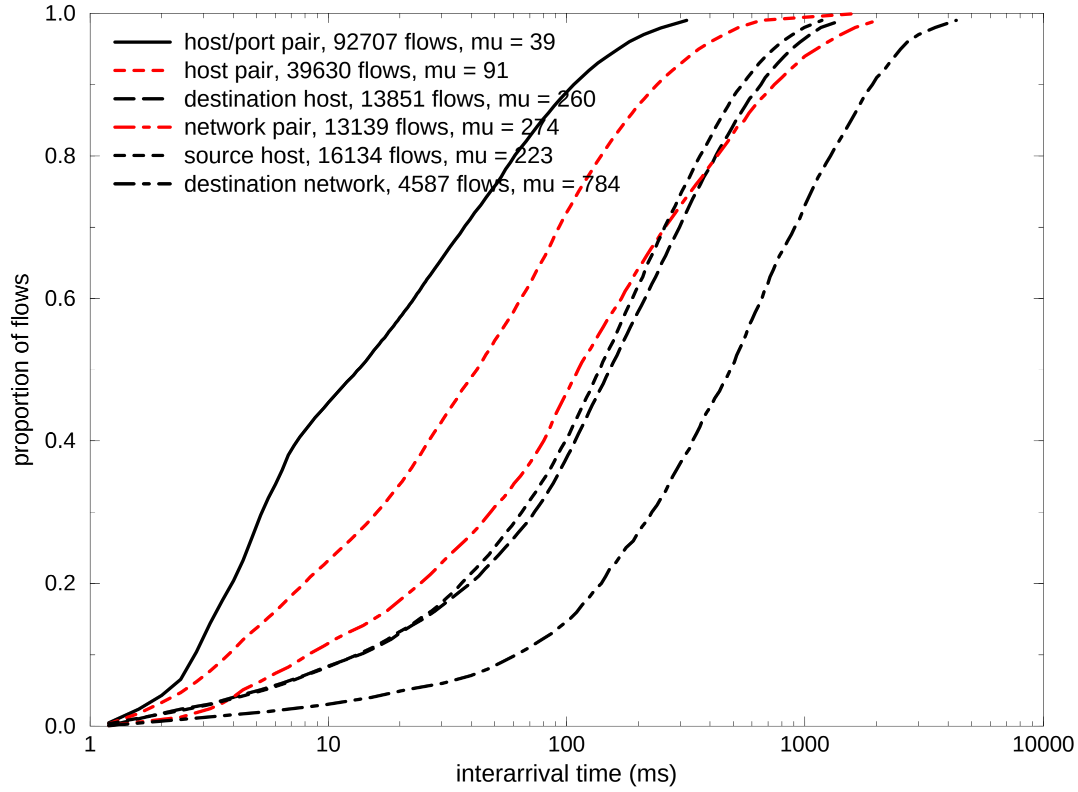

Figure 7.11: cumulative probability distribution (cdf) of flow interarrival times for six flow specifications (UCSD PM, 64 second flow timeout)

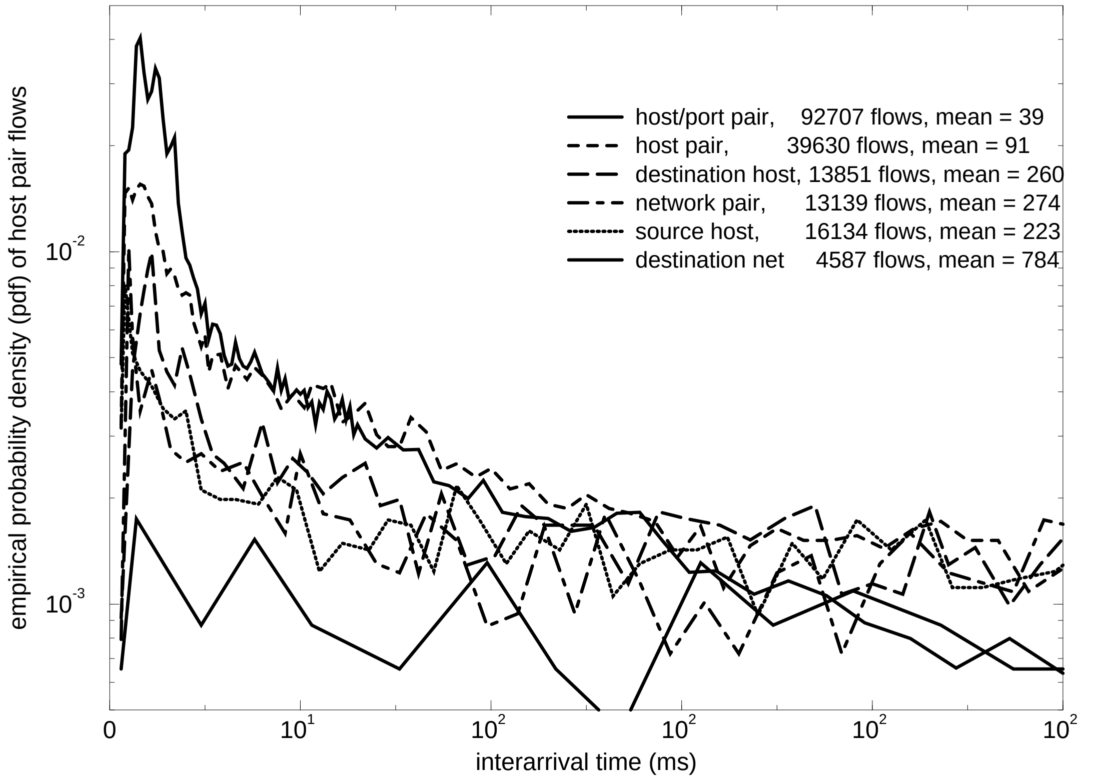

Figure 7.12: empirical probability density function (pdf) of flow interarrival times for six flow specifications (UCSD PM, 64 second flow timeout)

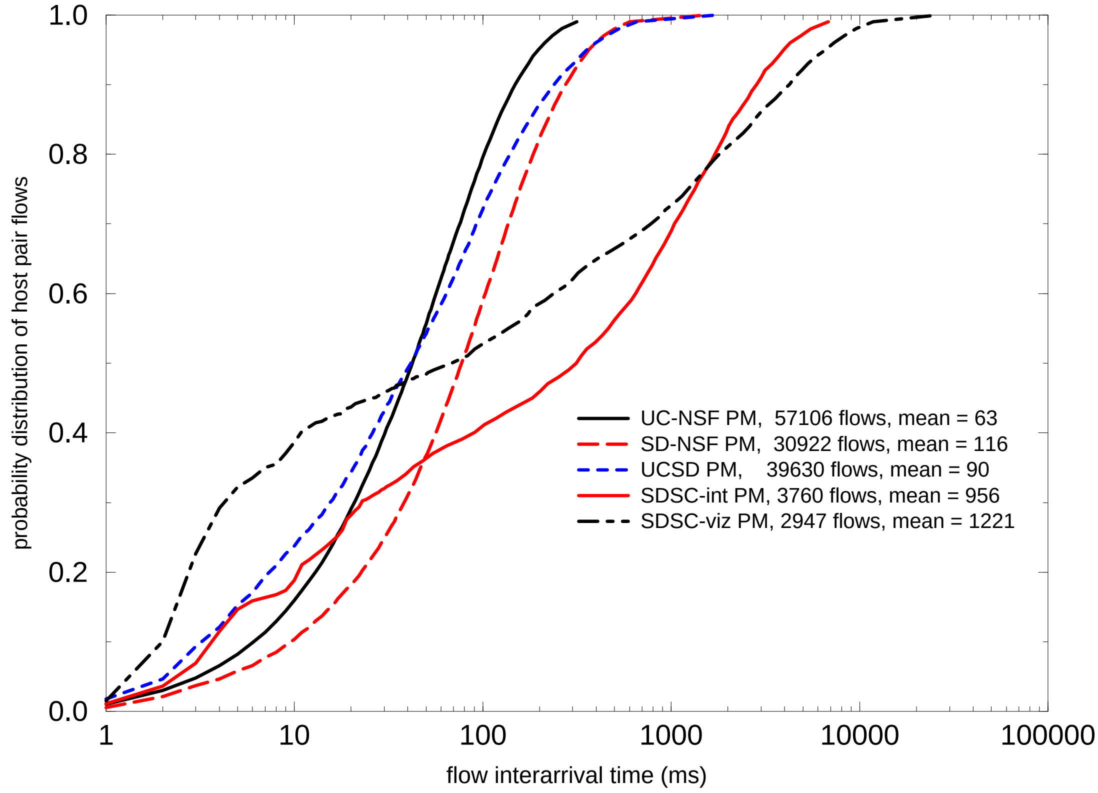

Figure 7.13: empirical distribution of host pair flow interarrival times for five environments (64 second flow timeout)

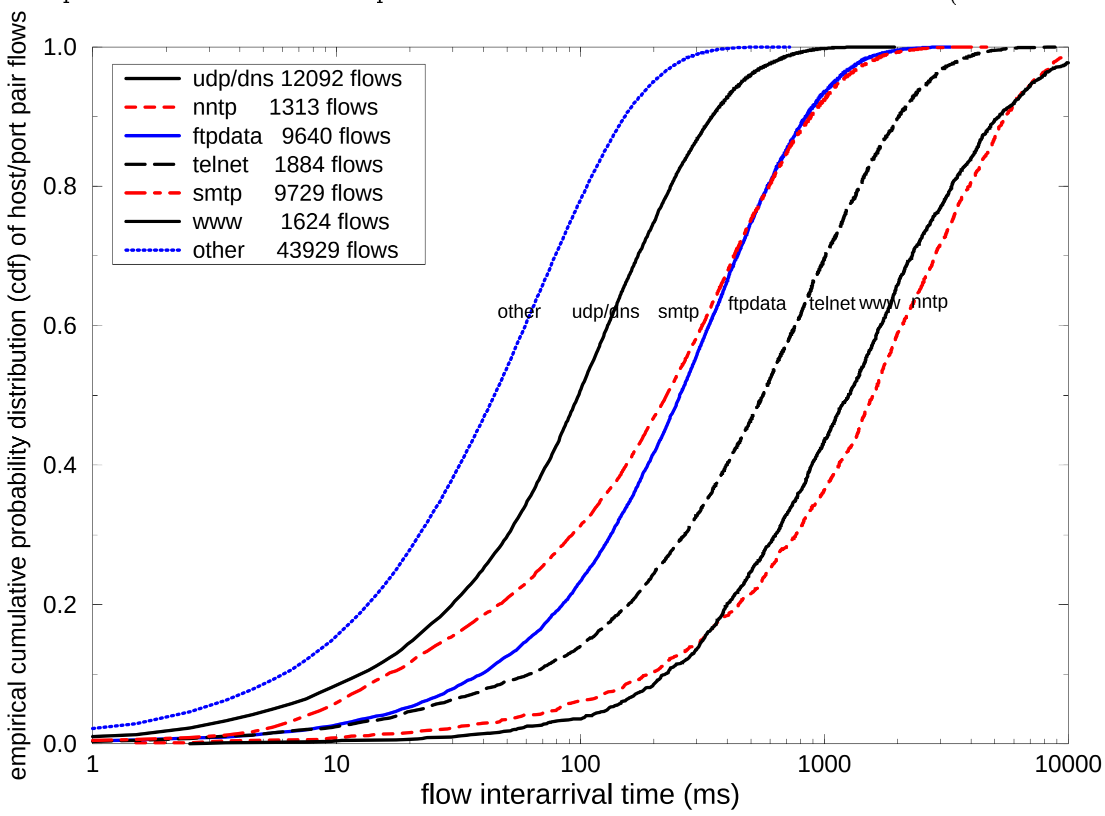

Figure 7.14: cumulative probability distribution (cdf) of flow interarrival times for seven TCP/UDP port classes (UC-NSF PM, 64 second flow timeout)

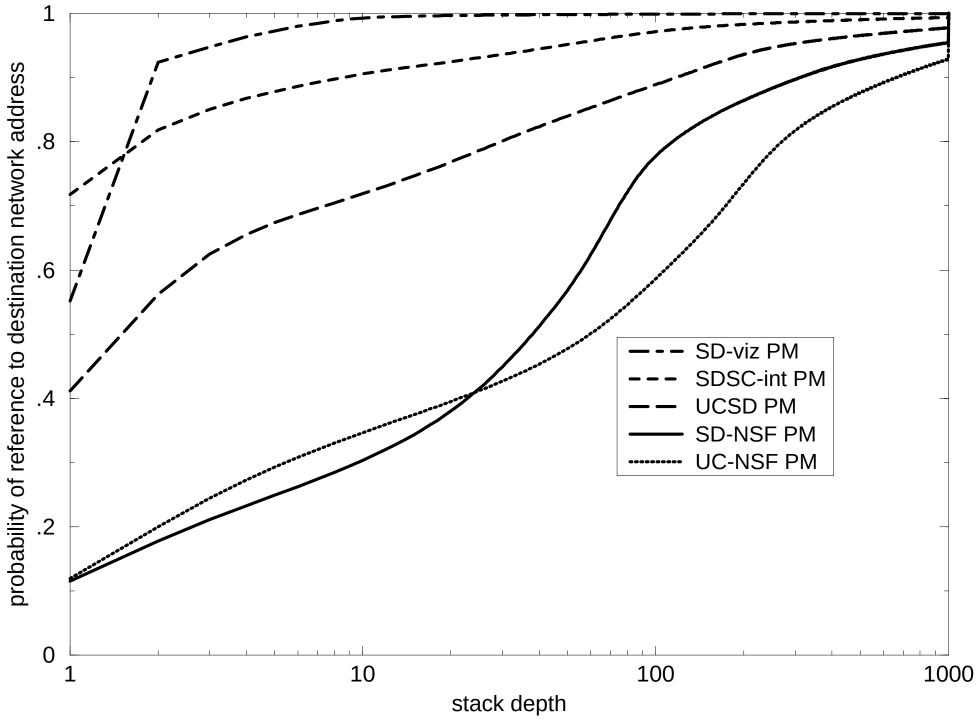

Figure 7.15: address reference stack depth probability distribution for five environments (busy hours)

Chapter 8

Future statistics collection

The measure of success is not whether you have a tough problem to deal with, but whether it's the same problem you had last year.

– *John Foster Dulles*

The new frontier of which I speak is not a set of promises – it is a set of challenges. It sums up not what I intend to offer the American people, but what I intend to ask of them.

– *John F. Kennedy*

In this chapter we discuss four trends that will affect how Internet service providers collect statistics. These trends will also force service providers to give more attention to data collection than they have in the past.

First, as the performance of the underlying link and switching equipment improves, its cost continues to rise. Designers of switching equipment will allocate every available cycle to increasing packet forwarding capacity, leaving few cycles left to devote to statistics collection. Similarly, service providers will allocate every available dollar to procuring these switches and the connecting bandwidth to serve their customers, leaving few dollars left to fund analysis of the traffic itself.

Second, the increasing reachability and interdependence of Internet components limits the value and practicality of isolated or specialized efforts to measure traffic, and yet brings many more potential dissatisfied customers if unanalyzed traffic profiles exceed the ability of service providers to cope with them.

Third, the demands of new, high-bandwidth applications threaten the current Internet datagram service model in which service providers aggregate traffic among a large number of clients running asynchronous, low-bandwidth applications such as rlogin or electronic mail. Service providers will have to collect statistics to distinguish and serve different classes of traffic.

Finally, the changing cost structure of the Internet industry will heighten the need for usage-based accounting, imposing a more complex set of demands on statistics collection agents.

8.1 Equipment and collection cost

As described in Chapter 4, network service providers face cost-benefit tradeoffs in deciding which traffic statistics are most important to collect given finite statistics collection and analysis capacity. We have covered many examples of metrics which are impossible to assess given the 15-minute granularity of the NSFNET backbone statistics collection, e.g., queue lengths, throughput capacity, delay distributions. When a network operator perceives little benefit from such statistics, and particularly when statistics collection must take at least some resources away from those otherwise allocated to packet forwarding, a network operator would be unwise to pursue their collection.

Many network operators follow the consensus of the Internet Engineering Task Force (IETF) Operational Statistics working group, who in 1992 recommended that Network Operations Centers (NOCs) poll metrics at fifteen minute intervals[115]. They reason that a five minute interval requires too much disk and CPU resources which it has not proven is justified. One alternative which they did not research was to poll as often as every five minutes but only store the high, low, and average values once per hour. Such adjustments could offer a finer-grained picture of these simple metrics, but will give no indication of the characteristics of individual, perhaps very high volume and duration, traffic flows.

Simply procuring faster or more switches will not solve the problem. Bandwidth and switching demand, as well as service expectations, always manages to eat up available computing resources, at least in peak periods. A trend toward fixed cell switching, in particular with the small ATM cell format, further limit the available switching time. Emerging real-time applications that consume significantly more resources than current ones will also add to the burden of network switching equipment. Several new proposals involve queueing algorithms to control congestion in the face of such new applications on the network [22] [48] [52] [49] [51] [50] [53], and will rely on integral caching mechanisms to maintain state for each communicating flow. Future Internet routers will likely have to cope with vastly different forwarding procedures in cases of different packet addresses or policy options.

Although new demands on equipment would only seem to reduce the priority of statistics collection even further, in fact the efficacy of proposed mechanisms will rely more critically on empirical traffic characteristics than have previous Internet protocols and mechanisms. In particular, the effectiveness with which a router can maintain flow state will depend on traffic type and locality. Design of cache memory configurations and optimization of routing tables in the packet forwarders will require up-to-date knowledge of how many different kinds of policy or service quality options are required per switch interface, and the resources each flow will consume. Collecting these statistics once, e.g., as a research study, will not suffice; ongoing assessment of traffic workload and its changes will be important components of the new service model.

8.2 Reachability

As Internet reachability increases, so does the potential influence of traffic from one user on any other user. The increasing reachability and interdependence of Internet components limit the applicability of isolated or specialized efforts to measure traffic.

Another variable that increases with Internet reachability is the gap in traffic distribution between the heaviest and the smallest users. The current implementation of the Internet assume a vast aggregation of traffic from many sources and wide distribution of traffic both in space (traffic source) and time (burstiness of traffic volume). Its datagram architecture typically supports no admission control in packet forwarders, and thus Internet components have little if any ability to control the volume and distribution of incoming traffic. Most entrance points into transit networks can not provide back pressure to other points of the network that inject more traffic than the network can handle. End systems, in particular, those using real-time video and audio applications that continuously block significant fractions of the available bandwidth, can thus unfairly monopolize available bandwidth for sustained periods and cause significant congestion in the larger network.

Service providers that aggregate traffic from an increasing number of network customers will need to know if a given level of utilization is a result of many flows, or of a few large flows that take over significant fractions of the available resources from other potential customers. The trend away from short flows and toward more continuous real-time multimedia flows, and a more general shift toward applications of higher flow volume and duration, will lower the threshold under which a service provider can aggregate traffic among customers. Network operators will get a far greater return on their investment in traffic characterization if they can extrapolate expected demand not only from aggregated link utilizations, but also actual application and user requirements. Their return will only increase as requirements for accounting increase, as we discuss further in section 8.4.

8.3 New real-time applications

The ever-increasing diversity in Internet application profiles, whose complexity will increase further with the newer continuous-flow multimedia applications, will also generate new requirements of statistics collection. The conventional method of supporting new applications – upgrading the performance of highly utilized network resources, whether bandwidth or switches – will become insufficient as the system continues to grow. Already such a method is financially impractical in an environment with an increasing number of relatively small service providers. Furthermore, software developers continue to build advanced network applications that can consume as much bandwidth as network engineers can provide. In particular, applications using packet audio, video, or rapidly changing graphics, require continuous delivery of large amounts of traffic in real-time and thus consume significant bandwidth for sustained periods. Clearly, usage of such applications will not scale in the current Internet architecture. Yet, this architecture will need to support such continuous point-to-point and point-to-multipoint connections simultaneously in the near future.

It is difficult to overestimate the dramatic impact that continuous media will have on the Internet fabric. No other phenomenon could more strongly drive the reassessment of network mechanisms such as queueing management in routers, and instrumentation of the network for admission control and multiple service levels, as well as accounting and billing. Even within the traditional flow paradigm, subcategories of traffic such as interactive, transaction, or bulk traffic, may exhibit performance requirements which are different enough to justify priority queueing.

The current paradigm of statistics collection, based on assumptions of a conventional traffic profile and only few new network numbers per week, assumes fairly even network utilization over time, so service providers have months to prepare for changes. This paradigm is breaking down with the current and imminent significant changes in application profiles. Service providers will have to deal with demand for multiple multicast video sessions, perhaps 10 today, 100 or more next year, and eventually thousands of point-to-point video and audio flows, scientific applications transmitting massive data flows on the order of terabytes, continuous exchange of weather data, constant consumption of 10-100 kbps by someone who accidentally left their video camera on[1], etc. Statistics collection will have to play a role in facilitating a transition to and subsequently accommodating (and likely modulating) the new demands of such traffic.

8.4 Cost structure

The current cost structure of the Internet also faces a transition which will affect the kind of statistics collection required. Currently the network costs are typically not based on volume and often hidden from end users. Most architecture and instrumentation for accounting in the Internet reflects this status as bulk-funded good for the academic community rather than the free market pay-per-service datagram environment toward which it is currently evolving. As a result the current Internet architecture is not conducive to pricing resource consumption among multiple entities and at multiple qualities of service. Quadratic traffic volume increases that still characterize many network clients reflect this current perception of the network as a research environment with usage-insensitive costs that are often transparent to the end-users. Application designers in such a network environment have no incentive to optimize their applications for network efficiency over performance. As a result, many applications exhibit grossly low per-packet data payload and other suboptimal traffic behavior.

As a wider variety of commercial and non-commercial providers offer Internet services, and as inequities among client usage of those services increase, the traditional flat-fee approach to service will become untenable.[2] Service providers must be able to adapt rapidly to the demand for greater capacity and sophistication. To support the infrastructure, providers will need a predictable revenue stream from fees that are based on the quantity of resources that users consume. Usage-sensitive accounting will become imperative. Perfectly accurate measurements of resource usage may be impractical, but providers will need

[1] A T1 bandwidth link could support only 15 to 150 such customers.

[2] Some commercial service providers have already introduced transport service products (ANS Litespeed, Alternet Low Volume) with lower access charges for infrequent or bursty use. ANS gateway customers can elect to pay according to a tiered structure if their average link utilization (SNMP 15 minute averages measured) falls within the 10%, 20%, 30% bands. Mills *et al.* [116] discuss other decentralized schemes that link statistics to billing by relying on interactions between adjacent Administrative Domains (ADs).

data that is adequate to present to users who wish to verify their bills. Service providers may offer a variety of billing schemes, as phone companies do today, such as flat fees, or flat up to some usage point, with usage-based billing beyond that point.

Although increasingly critical, these topics are beyond the scope of this thesis. Several studies have investigated pricing based on usage and/or priority [117] [118] [119] [120]. However, prerequisite to accounting and billing instrumentation is a more accurate model for the attribution of resource consumption, including distinguishing among different qualities of service and between sender-initiated (e.g., e-mail) and receiver-initiated (e.g., gopher, mosaic, www, ftp) traffic. Braun *et al.* [121] and Bohn *et al.* [91] discuss incremental steps that would move the infrastructure in this direction. They address the time window where applications exist on a network not designed for them, but before an appropriately architected network can augment the current infrastructure and cope with the new type of workload. They propose a scheme for voluntarily setting Internet traffic priorities by end-users and applications, using the existing 3-bit precedence field in the IP header.

The proposal has three elements. First, network routers would queue incoming packets by IP Precedence value instead of the customary single-threaded FIFO. Second, users and their applications would voluntarily use different and appropriate precedence values in their outgoing transmissions according to some defined criteria. Third, network service providers may monitor the precedence levels of traffic entering their network, and use some mechanism such as a quota system to discourage users from setting high precedence values on all their traffic. All three elements can be implemented gradually and selectively across the Internet infrastructure, providing a smooth transition path from the present system. Experience gained from an implementation will furthermore provide a valuable knowledge base from which to develop sound accounting and billing mechanisms and policies in the future.

It is not yet clear what role sampling, discussed in Chapter 4, will play in cost accounting. In this thesis we investigated sampling for packet length and interarrival time distributions, dividing the distributions into a small number of buckets in order to calculate deviations between sampled and full populations. We did not investigate more complex sampling objects that extend to many buckets. For example, the ANSnet/NSFNET backbone employs 1-out-of-50 sampling to derive an IP network number matrix for flows aggregated over the range of a month. As the number of participating IP networks grows beyond 14000, the matrix among them grows extremely sparse. A 1/50 sampling fraction, while appropriate for exposing general trends based on monthly aggregation, will be insufficient from which to draw results for short durations. Tracking flows for accounting purposes between network end systems (hosts) is quite uncomparable to a rough aggregation of net number based flows on a one month granularity. Individual host flows may only exist a few times during the month, and be of low traffic volume and duration, and the sampling scheme may in fact never catch one of their packets. Finding an acceptable level of sampling accuracy will also require an understanding of the nature of flows and how they may affect the efficacy of a sampling component of a billing methodology.

8.5 Summary and future work

In this chapter we have discussed four issues that will increase the importance of statistics collection in the future.

Chapter 9

Conclusion

There are no whole truths; all truths are half-truths. It is trying to treat them as whole truths that plays the devil.

– Alfred North Whitehead

Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.

– Sir Winston Spencer Churchill, 10 Nov 1942

The three main goals of this dissertation have been to use existing operational statistics to characterize Internet traffic on a wide area datagram infrastructure, to evaluate the utility of these statistics for workload characterization tasks, and to develop and test a methodology for statistics collection that can facilitate more accurate workload characterization than current statistics allow.

To fulfill the first goal, we began with a taxonomy of Internet workload characteristics, which form a base from which to measure this complex system of interconnected components. We then investigated existing operational statistics on core network backbones, specifically the T1 and T3 based NSFNET backbones. We used these statistics to assess several items in the taxonomy, such as long term growth in traffic volume, reachability, service diversity, and traffic locality.

To fulfill the second goal, we discussed how these statistics limit the ability to assess these items, and completely prevent the assessment of other items in the taxonomy. We also included a dedicated study of the impact of sampling on accurate workload characterization, based on the sampling methodology deployed on the NSFNET backbone. Our evaluation highlighted an important issue: statistics collection in currently deployed Internet components is typically driven by short term requirements, e.g, immediate operational status information or engineering data such as aggregated link utilization. Statistics collection often takes a back seat to more immediate network management; resulting inattentive data collection prevents collection at the level of detail, completeness and confidence needed for many workload characterization tasks. Resulting statistics allow some tracking of Internet growth, but limit the ability to forecast capacity requirements in a network with ever richer functionality, and also do not allow quantification of detailed traffic characteristics, which vary considerably in both time and space granularities.

A good example of the limitations of current statistics is in tracking the tremendous growth in Internet application/service diversity as measured by TCP/UDP port numbers. Although the statistics indicate a proliferation of utilized ports, there is no mechanism to determine what application, or even class of application, an arbitrary port represents. Yet assessing the service profiles of these new applications will be important to accommodating them on Internet components. In particular, as newer continuous flow multimedia applications contribute to the complexity of the aggregate demand, they will require reevaluation of design issues such as queueing management in routers in order to provide multiple service classes. Even within the non-continuous flow paradigm, subcategories of traffic such as interactive or bulk, may exhibit performance requirements which require adaptive queue management. Operators of Internet components could clearly benefit from a more accurate assessment of the impact of certain applications on the overall demand they must satisfy. Other metrics that would facilitate workload characterization include queue length distributions, packet drop characteristics, and more detailed insight into interface error conditions. However, all such collection comes at a cost, and a network operator must weigh these costs against the benefits that availability of such statistics will provide.

Limitations of the operationally collected statistics for in depth workload characterization and modeling tasks led us to focus on the third goal of this thesis, development and testing of a methodology to address tasks for which available operational statistics are not useful. Specifically, we investigated Internet traffic flow characterization, which covers several of the tasks discussed in Chapter 2. We developed a methodology for describing flows in terms of their impact on an aggregate Internet workload, and tested it using packet header traces from a variety of Internet locations. Our methodology defines a flow based on actual traffic activity from, to, or between entities, rather than using the explicit setup and teardown mechanisms of transport protocols such as TCP. Using this definition we explore the effect of several parameters, such as flow timeout, granularity, environment, and higher layer protocol, on several flow metrics. These metrics fall into two categories: metrics of individual flows in Chapter 6, and metrics of the aggregate traffic flow in Chapter 7. Metrics of individual flows include: volume in packets and bytes per flow, and flow duration. Applying the methodology to the measurements yielded significant observations of the Internet infrastructure, which have implications for performance requirements of routers at Internet hotspots, general and specialized flow-based routing algorithms, future usage-based accounting requirements, and traffic prioritization.

A particularly surprising observation is that a large proportion of flows are very short in duration: using a 64-second timeout for the busiest backbone data set (UC-NSF PM), almost 60% of the flows are less than one second (40% consisting of a single packet). Flow volume and duration is correlated to the transport or higher level protocol. Given a 64 second timeout, only 10% of TCP flows consist of a single packet, but over 60% of UDP flows consist of a single packet. TCP/UDP ports, which map to higher level applications, also provide an indication of the expected duration and volume of a flow: e.g., the large majority of the single packet UDP flows are from the *dns* protocol, unsurprising given the nature of the *dns* protocol. Using a 64-second flow timeout, although *dns* constituted only 2.4% of the total packets, it constituted 27.6% of the total number of flows. When examining the shorter flows as a function of protocol, it is evident that many of the short flows are *dns*, *gopher*, *ntp*, or *finger*. If a router were able to selectively refuse to save state or set up virtual circuits for flows of a certain application type, it could save over a third of the memory it dedicates to maintaining state entries.

Metrics of the aggregate Internet flow also have implications for the design of future network equipment and protocols. For example, we illustrated the tradeoff between timing out flows too early and leaving them to chew up memory resources. For our busy backbone data set, at a timeout value of two, each of the 25,000 unique host pair flows is set up and torn down almost ten times *on average* during the hour; a timeout of 16 seconds brings this redundancy factor down to 2.9. Our measurements indicate that, at least in the environments we studied, current IP traffic still consists more of short transaction type traffic rather than longer term flows. The short packets and short flows together shed doubt on a strategy of optimizing for long flows that are in fact the minority case. However we note that many new applications may change this characteristic of Internet environments, as they introduce traffic flows with different behavior, particularly real-time continuous media flows, which tend to exhibit greater duration and flow volume. Popularity of such applications will shift the aggregate flow profile in a direction that will threaten the integrity of the current datagram Internet which relies on high aggregation among many bursty sources.

Another important result of our measurements is that the number of host-host pair flows appears far less than proportional to the square of the number of network number pair flows, as a uniform matrix of traffic volume among connected sites implies. This phenomenon is prominent even in wide area environments which aggregate among a large number of users, and bodes well for routing methodologies requiring end host pair state. Traffic locality can compensate somewhat for the brevity of such a large proportion of Internet flows that we found in Chapter 6. Understanding these interactions, i.e., how individual flows and the aggregate flow profile influence each other, is essential to securing Internet stability, and requires ongoing flow assessment to track changes in Internet workload in a given environment.

We devoted the last component of this thesis to a discussion of several factors that will increase the importance of statistics collection, in particular flow assessment, in Internet components. These factors included the increasing complexity in equipment, connectivity, service expectations, and financial structure of the Internet. The combination of these factors will make critical not only improved statistics collection, but also technology for accounting and billing, accompanied by network mechanisms such as queue management and multiple service qualities in routers. These four factors doom the paradigms of many Internet service providers who base network engineering decisions on simple statistics data sets.

As a result of our investigations we have proposed mechanisms to support operational flow assessment, to gain insight into both individual traffic signatures as well heavy aggregations of end users. The proliferation of different Internet traffic types that exhibit fundamentally different workload characteristics, including those requiring service guarantees and not using a transport protocol to delineate the beginning and end of a flow, makes it even more difficult to define an Internet flow, but also more critical. The Internet has survived, in fact succeeded beyond the wildest dreams of its initial designers and users, with a rather relaxed attitude toward systematic analysis of traffic flows. This attitude may in fact have been a factor in its success; analysis of traffic often took a back seat to the priority of getting more traffic, i.e., sites, on the network.

However the Internet will not be able to secure and maintain stability in the face of new traffic types and continued explosive growth without a more dedicated approach to Internet traffic analysis, the first step of which is accurate workload, or flow, characterization. The flow assessment methodology we described in Chapter 5 can form a complementary component to existing operational statistics collection, yielding insights into larger issues of Internet evolution, i.e., how environments of different aggregation can cope with resource contention by an ever-changing composition and volume of flows. We have focused on methodologies, and representative environments, not an exhaustive exploration of all possible environments or questions. Indeed, Internet traffic cross-section and flow characteristics are a moving target, and we intend that our methodology serve as a tool for those who wish to track and keep pace with its trajectory. For example, as video and audio flows, and even single streams combining voice and audio, become more popular, Internet service providers will need to parametrize them to determine how many such customer streams they will be able to support, and how many more resources each new such customer would require. Multicast flows will also likely constitute a increasingly significant component of Internet traffic, and applying our methodology to multicast flows would be an important step toward coping with their impact on the infrastructure.

Because it requires comprehensive and detailed statistics collection, service providers may not be able to afford to continuously monitor flow characteristics on an operational basis. Nonetheless we advocate that network operators undertake flow assessment at least periodically to obtain a more accurate picture of the workload their infrastructure must support. The methodology we describe can serve as a valuable tool for such assessments. Our methodology will be increasingly applicable, even on a continuous basis, for tasks such as ATM circuit management, accounting, routing table management, and load balancing in future Internet components. Our methodology can improve current operational statistics collection architectures, allowing service providers to prepare for more demanding use of the infrastructure and allowing network analysts to develop more accurate Internet models. In short, we can reduce the gaps among (1) what network service providers need; (2) what statistics service providers can provide; and (3) what network analysis requires.

Appendix A

Key components of the Internet environment

You've got to think about 'big things' while you're doing small things, so that all the small things go in the right direction.

– *Alvin Toffler*

You've never been out of school! You don't know what it's like out there! I've worked in the the private sector. They expect results.

– *Dr. Raymond Stantz,* Ghostbusters

The Internet is a global network infrastructure spanning many countries. The United States component of the Internet grew from a three-level hierarchical model of national agency backbones, attached mid-level networks, and connected local sites. Figure 3.1 illustrates this hierarchical structure. ESnet, Milnet, NSI, NSFNET, and TWB correspond to national backbones of DOE, DoD, NASA, NSF, and ARPA, respectively.[1] We briefly examine the model as a logical tree rooted at the Federal Interexchange points (FIXes), with branches representing the federal agency networks. This abstraction is not wholly accurate, as it ignores commercial IP providers, international links, and bypass interconnections, all of which continue to increase steadily. Nonetheless, it provides an architectural understanding for how networks are interconnected and local responsibility is compartmentalized. This architecture also allows local sites and mid-level networks autonomy with respect to their own routing and management decisions, and to rely on networks hierarchically above them in the tree for transit.

A.1 US Federal Agency Interconnection Points

Two Federal Interexchange points, FIX-West at NASA-Ames in Moffet Field, CA, and FIX-East, at a SURANET location outside of the University of Maryland at College Park, are the principal interconnection points for United States federal agency research and education IP networks. At these interconnection points routers are co-located on an Ethernet and FDDI LANs, allowing decoupled routing policies through the use of interdomain protocols, e.g., EGP and BGP, to exchange routing information among peer agencies.

These hubs do not exactly constitute a separate level of the U.S. Internet hierarchical infrastructure, but rather overlay the existing three-level architecture. Their strategic locations make them ideal points for data collection of interagency traffic, as well as for location of specific servers, such as for domain names and network time.

[1] This list corresponds to Department of Energy, Department of Defense; National Air and Space Administration; National Science Foundation; and Advanced Projects Research Agency.

A.2 US Agency networks

Historically, United States federal agencies, such as ARPA, NASA, and DOE, connect individual clients to their backbones according to their programmatic network requirements. At the same time, the NSF has implemented infrastructure in broad support of the networking community. We describe NSF's infrastructure in some detail, and briefly mention the others.

NSFNET, the National Science Foundation Network, is a general purpose packet-switching network supporting access to scientific computing resources, data, and interpersonal electronic communications. Evolved from a 56kbps six-node network in the mid-1980s to today's 45Mbps network, the current NSFNET includes three different levels: the transcontinental backbone connecting the NSF-funded supercomputer centers and mid-level networks, the mid-level networks themselves, and the campus networks. The hierarchical structure includes a large fraction of the research and educational community, and even extends into a global arena via international connections. Figure 3.2 shows the logical topology of the backbone.

Since July 1988, Merit Network, Inc. has administered and managed the T1 NSFNET backbone, and in late 1990, in conjunction with its partners IBM and MCI, began to deploy in parallel a replacement T3 network.[2] The T3 network provided a 28-fold increase in raw capacity over the T1 network (from 1.544 Mb/sec to 44.736 Mb/sec), and by November 1992 had completely replaced the T1 network.

In the interim, the status of the NSFNET shifted through organizational restructuring among original participants in the backbone project. In 1991, Advanced Network Services (ANS) began official operation and management of the national T3 backbone described above. Merit Network, Inc. still holds a cooperative agreement with NSF to provide NSFNET backbone services, although Merit no longer provides these services via a dedicated infrastructure. Merit now subcontracts these services to ANS, who provides them over ANSnet, their own backbone infrastructure. The "NSFNET backbone" now refers to a virtual backbone service, i.e., a set of services provided across the ANSnet physical backbone. In this thesis we refer to the "T3 NSFNET backbone" with the understanding that we are referring to a service provided to NSF, not a dedicated NSFNET physical infrastructure.

Other agency infrastructures include the NASA Science Internet (NSI) and DOE's Energy Sciences network (ESnet) which each support programmatic mission goals of their agency. ARPA supports two network testbeds, DARTnet and Terrestrial Wideband Network (TWBnet), chartered for the research and development of new protocol, switch and router technologies to support a wide range of advanced network services such as packet video, multicasting, distributed multicomputer simulation, network resource management and control.

A.3 Network access points

NSF is also supporting the development of Network Access Points (NAPs), which will allow federal, non-federal, mid-level, and international service providers to peer with each other without restrictions of the NSF Appropriate Use Policy. FIXes are a current instantiation of NAPs that focus on federal networks, and in fact the FIX concept inspired the NAP model. NAP objectives include:

- integrate and interconnect different network technologies, services, protocols and routing strategies
- provide a mechanism for new network service providers, including commercial networks, to connect to and peer with the R&E oriented infrastructure
- improve the robustness of the system in the face of the increasing complexity of interconnectivity requirements
- allow for evolution of network layer routing in the Internet
- monitor and ensure correct routing exchanges between peering networks
- provide a mechanism for generating default routing information for networks which do not want to carry the entire routing topology of the Internet withing their network

[2] The original cooperative agreement between Merit and NSF in 1987 allowed for this optional upgrade to T3 speeds, which a later follow-on proposal to the original agreement more clearly specified.

United States NSFNET Mid-level Networks
BARRNET (Bay Area Regional Research Network, California)
CERFnet (California Education and Research Federation Network, California)
CICNet (Committee on Institutional Cooperation Network, Mid-West)
CO Supernet (Colorado Supernetwork, Colorado)
CONCERT (Communications for North Carolina Education, Research, and Technology Network, North Carolina)
INet (Indiana Network, Indiana)
JVNCnet (John von Neumann Center Network, Northeast)
Los Nettos (Southern California)
MichNet (Michigan Network, Merit-statewide, Michigan)
MIDnet (Midwestern States Network, Mid-West)
MRnet (Minnesota Regional Network, Minnesota)
NEARnet (New England Academic and Research Network, North-East)
netILLINOIS (Illinois)
NevadaNet (Nevada)
NorthwestNet (Northwestern States Network)
NYSERnet (New York State State Education and Research Network)
OARnet (Ohio Academic Research Network)
PREPnet (Pennsylvania Research and Economic Partnership Network)
PSCNET (Pittsburgh Supercomputing Center Network, Mid-West)
SDSCnet (San Diego Supercomputer Center Network)
SESQUINET (Texas Sesquicentennial Network)
SURAnet (Southeastern Universities Research Association Network, South-East)
THEnet (Texas Higher Education Network)
VERnet (Virginia Education and Research Network)
Westnet (Southwestern States Network)
WiscNet (Wisconsin)
WVNET (West Virginia Network for Educational Telecomputing)

A.4 Midlevel Networks

Mid-level (regional) networks form the next layer of branching below the NSFNET backbone and other national transit network.[3] (See Table A.4.) These mid-level networks connect to the NSFNET and/or other federal agency backbones, and provide connectivity among sites in the academic environment. Federal mission agencies also assume the services of mid-level networks for connectivity to their university-based researchers. The NSFNET backbone allows for peer network connectivity to networks such as national backbones, e.g., Advanced Network and Services, Alternet, Performance Systems International, and SprintLink. The backbone also supports international connections to national backbones of foreign countries.

A.5 Campuses

The transit networks we have described support connectitivity among individual sites which include university and college campuses, research laboratories, private companies, educational sites such as K-12 school districts, etc. These sites attach as clients to mid-level networks, or sometimes directly to backbone service providers. The aggregated investment at these network distribution sites dramatically surpasses that of federal government investment in backbone and mid-level networks. For example, the annual cost of operation of an individual large campus network can approach that of an entire government agency T-1 based backbone. These site networks form an integral part of the overall infrastructure, largely financing

[3] Mid-level networks have also been called "regionals", reflecting their geographical span, but we will use the term "mid-level" to reflect their hierarchical position in the architecture.

their expenses out of internal funding, and are quite independent in both internal as well as interconnection decisions.

A.6 Commercial Providers

Restrictive usage policies of federal agency networks have contributed to the introduction of commercial service providers to provide unrestricted IP connectivity on a national scale. Commercial providers include Advanced Network Services (ANS), Performance Systems International, Inc. (PSI), Sprintlink, and UUNET Technologies' Alternet. Mid-levels, e.g., CERFnet, have also begun to offer commercial services. Commercial service providers can enable subscribers, regardless of government, academic, or commercial affiliation, to integrate PCs, mainframe, or LANs into the global Internet. Generally included in the range of services are domain and network number registration, 24-hour customer support, Usenet access, usage statistics, and extensive consulting and training. Connectivity is available at various speeds via dedicated leased circuits, and customers may attach their own routers and communications equipment to the commercial provider's network.

Several existing but independent attempts to provide commercial connectivity led to the need for commercial interchange points (CIXes) analogous to the FIXes described above, to provide broader connectivity.

A.7 International components

The NSF supports extensive network connectivity to international R&E sites and networks. This is accomplished through direct funding of international connections as well as through interagency cooperatively funded links.

Appendix B

Tools for Network Management and Operation

For our present purpose, however, it is only necessary to recognize that, whatever degree of care and experimental skill is expended in equalizing the conditions, other than the one under test, which are liable to affect the result, this equalization must always be to a greater or less extent incomplete, and in many important practical cases will certainly be grossly defective. We are concerned, therefore, that this inequality, whether it be great or small, shall not impugn the exactitude of the frequency distribution, on the basis of which the result of the experiment is to be appraised.

– *Sir Ronald A. Fisher,* Experimental Design

Generally you don't see that kind of behavior in a major appliance.

– *Bill Murray,* GhostBusters

As important as traffic totals and the reports characterizing traffic characteristics are, it is also necessary to have tools which analyze traffic patterns and apply this information to future network planning. This section focuses on such tools for effective network management. Stine [122] describes many of these tools in more detail.

We differentiate network tools into a two major categories: tools in support of real-time work, and non-real-time tools to analyze a collected data set. Real-time network management tools are typically needed during general operation of the network, such as to retrieve error information from the environment and then react to it, e.g., *traceroute*, SNMP queries. Other real-time tools allow for specific performance tests on existing networks, e.g., *ping.* Non-real-time tools allow for the analysis of network behavior following the collection of network information, or for simulation given emprical or analytic traffic data.

Statistics collection tools may be either intrusive or non-intrusive. Intrusive tools typically conform to some request/response function, where a networking entity responds to a request/query. *Ping*, an ICMP echo request/reply tool, is an example of a real-time, intrusive tool frequently used to determine reachability, latency and quality of the connectivity. In contrast, non-intrusive tools passively monitor traffic without disturbing the networking environment. Examples include Ethernet monitors, e.g., Netwatch, NNStat, *iptrace*, and *netsnoop*, all of which run without introducing queries as part of the monitoring activity. ARTS (ANSnet Router Traffic Statistics), which ANS designed based on NNstat functionality for collecting statistics on T3 NSFNET backbone nodes, is somewhat hybrid; each node has several subsystem cards that can switch packets in parallel, but each of them forwards every fiftieth packet they switch to the main CPU for statistics gathering.

B.1 NNStat

The day-to-day SNMP statistics gathering of the T1 NSFNET Backbone was augmented by a software package, NNStat [71], that allows further insight into specific traffic characteristics. NNStat, a set of programs and utilities developed by the Information Sciences Institute of the University of Southern California, includes a module for the collection and aggregation of traffic statistics, as well as two programs for remote access to results of the collection module. The NNStat collection module, *statspy*, gathers traffic statistics via a promiscuous Ethernet tap on the local network, rather than instrumentation on the gateways. The statistic gathering agent can passively access and thus profile all traffic traversing that Ethernet regardless of its original source or ultimate destination. *Statspy* can also use a filter to gather information only on certain traffic, e.g., traffic from a specific source or destination IP address (typically a gateway address), or of a certain type, all of which the user can specify. A program called *collect* retrieves and manages collected data by periodically polling various *statspy* processes in the domain of interest to retrieve locally logged statistical data. Another utility, *rspy*, supports requests to retrieve single items of data.

B.2 Ping

Ping sends an ICMP echo request packet to a host or gateway which, if functioning properly, answers with an ICMP echo reply packet. By default the *ping* command sends one datagram per second and prints one line of output for every response received, although it supports options to invoke alternative behavior. The *ping* command is useful for determining whether a host is up as well as current delay and jitter to that host. *Ping* can assist in tracking and isolating hardware and software problems, and testing, measuring, and managing networks.

B.3 Netsnoop

Netsnoop is the SGI utility that passively collects packet headers on a broadcast medium, e.g. Ethernet, while allowing the specification of filters to only capture packets that match certain criteria. Interpretation of the packet headers can either happen in real-time, by displaying a result on a requestor's terminal, or by writing the data into a binary file, with offline parsing at some later time. *Netsnoop* uses a system call to a kernel routine called *snoop*. To collect traces for this thesis we wrote our own program using the *snoop* facility directly to avoid the application overhead of the *netsnoop* program.

B.4 Traceroute

Traceroute is a tool that reveals the route taken by packets from a given source to a given destination. It is useful in situations where the IP record route option would fail, such as if intermediate gateways discard packets, if a route is longer in hops than the time-to-live capacity of a datagram allows, or if intermediate gateways do not support the record route option. Round trip delays between the source and intermediate gateways are also reported, indicating how each gateway contributes to end-to-end delay. Other options include loose source routing, which allows one to examine the return path from a remote machine back to the local host.

Traceroute relies on the ICMP time exceeded error reporting mechanism. When a gateway receives an IP packet with a time-to-live value (TTL) of 0, it sends an ICMP TTL exceeded message to the host that generated the original packet. By sending packets to the ultimate destination with successively increasing TTLs starting from 0, one can invoke a ICMP time exceeded message from each intermediate gateway and thus identify each hop in the path. Each gateway also includes the timestamp of the original packet in its return ICMP time exceed message, so one can calculate round trip delay to each hop.

Inconsistencies among some IP implementations cause problems with the *traceroute* tool. First, some gateways forward packets with a TTL of 0, thus escaping identification. Second, some gateways use the TTL field in the arriving packet as the TTL for the ICMP error reply, which delays identification. Third, sending datagrams with the source route option will cause some gateways to crash.

Appendix C

Glossary of acronyms used

It's not what you say; it's what they hear.

ACK acknowledgement

AD administrative domains

ANS Advanced Network Services

ARIMA autoregressive integrated moving average

ARPAnet Advanced Research Projects Agency network

ARTS ANSnet router traffic statistics

AS autonomous system

ATM asynchronous transfer mode

BGP border gateway protocol

CERFnet California research and education network

CIDR classless interdomain routing

CNSS central nodal switching subsystem

CPU central processing unit

dns domain name system

DoD Department of Defense

DOE Department of Energy

DS3 Digital Signal Hierarchy 3, equivalent to T3, 44.736 Mbits/s

DSU data service unit

EGP exterior gateway protocol

ENSS external nodal switching subsystem

ESnet Energy Sciences network

FDDI fiber distributed data interface

finger finger application

FIN TCP packet that signals the closure of a TCP session

FIX federal interexchange point

ftpcontrol file transfer protocol, control

ftpdata file transfer protocol, data

ftp file transfer protocol

HDLC high level data link control

IANA Internet assigned numbers authority

IBM International Business Machines

ICMP Internet control message protocol

IETF Internet engineering task force

IGMP Internet gateway management protocol

InterNIC Internet network information center

IPIP IP encapsulated in IP

IP Internet protocol

IQD interquartile delay

IRDS information resource discovery services

IRIX SGI Unix-based operating system

IS-IS intermediate system to intermediate system routing protocol

ISI Information Sciences Institute at the University of Southern California

LAN local area network

LRU least recently used

MAC medium access control

MB megabyte

MIB management information base (used with SNMP)

Milnet operational backbone network for United States Department of Defense

NAP network access points

NASA National Aeronautics and Space Administration

NCSA National Center for Supercomputing Applications

ND network disk protocol

NFS network file system

NNStat NSF Network Statistics package, developed at ISI

nntp network news transfer protocol

NOC network operations center

NSFNET National Science Foundation network

NSF National Science Foundation

NSI NASA science internet

NSS nodal switching subsystem

ntp network time protocol

PRDB policy routing database

PVC permanent virtual circuits

RSVP resource reservation protocol

RTT round trip time

SDSC-DMZ San Diego Supercomputer Center, demilitarized zone (data set used for sampling study in Chapter 4

SDSC San Diego Supercomputer Center

SDSC-int San Diego Supercomputer Center, internal FDDI LAN (data set used for flow profiling)

SDSC-viz San Diego Supercomputer Center, visualization laboratory (data set used for flow profiling)

SD-NSF San Diego NSFNET node, traffic going into the backbone (data set used for flow profiling)

SGI Silicon Graphics, Incorporated

smtp simple mail transfer protocol

SNMP simple network management protocol

SVC switched virtual circuit

SYN TCP packet to signal beginning of session

T1 trunk speed of 1.5 Mbits/s

T3 trunk speed of 44.736 Mbits/s

TCP transport control protocol

TTL time to live

TWB terrestrial wideband network

UC-NSF Urbana-Champaign NSFNET node, traffic going into the backbone (data set used for flow profiling)

UCSD University of California, San Diego (also refers to UCSD data set used for flow profiling)

UDP user datagram protocol

UIUC University of Illinois, Urbana-Champaign

VC virtual circuit

WAN wide area network

Bibliography

[1] J. Becker, "personal communication (e-mail)," January 1993. Advanced Network Services.

[2] D. R. Boggs, J. C. Mogul, and C. A. Kent, "Measured capacity of an Ethernet: Myths and reality," in *Proceedings of ACM SIGCOMM '88*, pp. 222–234, August 1988.

[3] R. Gusella, "A measurement study of diskless workstation traffic on an Ethernet," *IEEE Transactions on Communications*, vol. 38, pp. 1557–68, September 1990.

[4] M. Davis, "Analysis and optimization of computer network routing," Master's thesis, University of Delaware, 1988.

[5] S. Heimlich, "Traffic Characterization of the NSFNET National Backbone," in *Proceedings of the 1990 Winter USENIX Conference*, December 1988.

[6] R. Caceres, P. Danzig, S. Jamin, and D. Mitzel, "Characteristics of wide-area TCP/IP conversations," in *Proceedings of ACM SIGCOMM '91*, pp. 101–112, September 1991.

[7] A. Schmidt and R. Campbell, "Internet protocol traffic analysis with applications for ATM switch design," *Computer Communiations Review*, vol. 23, pp. 39–52, April 1993.

[8] P. B. Danzig, K. Obraczka, and A. Kumar, "An analysis of wide-area name server traffic," in *Proceedings of ACM SIGCOMM '92*, August 1992.

[9] S. Hares and D. Katz, "Administrative domains and routing domains: A model for routing in the Internet." Internet Request for Comments Series RFC 1136, December 1989.

[10] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load." Technical Report, December 1992.

[11] K. Claffy, G. C. Polyzos, and H.-W. Braun, "Measurement considerations for assessing unidirectional latencies," *Internetworking: Research and Experience*, vol. 4, pp. 121–132, September 1993.

[12] S. Floyd and V. Jacobson, "On traffic phase effects in packet-switched gateways," *Internetworking: Research and Experience*, vol. 3, pp. 115–156, September 1992.

[13] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages," in *Proceedings of ACM SIGCOMM '93*, pp. 33–44, September 1993.

[14] L. Zhang, S. Shenker, and D. D. Clark, "Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic," in *Proceedings of ACM SIGCOMM '91*, pp. 133–148, September 1991.

[15] J. Mogul, "Observing TCP dynamics in real networks," in *Proceedings of ACM SIGCOMM '92*, pp. 305–317, August 1992.

[16] Z. Wang and J. Crowcroft, "Eliminating periodic packet losses in the 4.3 Tahoe BSD TCP congestion control algorithm," *Computer Communications Review*, April 1992.

[17] D.Sanghi, A. Agrawala, O. Gudmundsson, and B. Jain, "Experimental assessment of end-to-end behavior on the Internet," in *Proceedings of IEEE Infocom 93*, pp. 867–874, March 1993.

[18] A. K. Agrawala and D. Sanghi, "Network dynamics: an experimental study of the Internet," in *Proceedings of Globecom '92*, December 1992.

[19] V. Rutenburg and R. G. Ogier, "Fair charging policies and minimum-expected-cost routing in internets with packet loss," in *Proceedings of IEEE Infocom 91*, pp. 279–288, April 1991.

[20] B. Kumar, "Effect of packet losses on end-user cost in internetworks with usage based charging," *Computer Communiations Review*, August 1993.

[21] R. Jain, *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[22] L. Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks," in *Proceedings of ACM SIGCOMM '90*, pp. 19–29, August 1990.

[23] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," in *Proceedings of ACM SIGCOMM '93*, September 1993.

[24] K. Claffy, G. C. Polyzos, and H.-W. Braun, "Traffic characteristics of the T1 NSFNET backbone," in *Proceedings of IEEE Infocom 93*, pp. 885–892, 1993.

[25] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann Publishers, 1990.

[26] R. Jain, "Characteristics of destination address locality in computer networks: a comparison of caching schemes," *Computer networks and ISDN systems*, vol. 18, pp. 243–54, May 1990.

[27] N. Gulati, C. Williamson, and R. Bunt, "Local area network traffic locality: Characterization and application," in *Proceedings of the first International Conference on LAN Interconnection*, pp. 233–250, October 1993. Research Triangle Park, Raleigh, NC.

[28] R. Jain and S. A. Routhier, "Packet trains—measurement and a new model for computer network traffic," *IEEE Journal on Selected Areas in Communiactions*, pp. 986–995, September 1986.

[29] K. Claffy, G. C. Polyzos, and H.-W. Braun, "Internet traffic flow profiling." UCSD TR-CS93-328, SDSC GA-A21526, November 1993.

[30] V. Paxson, "Growth trends of wide area TCP conversations," *IEEE Network*, 1994. to appear.

[31] D. Estrin and D. Mitzel, "An assessment of state and lookup overhead in routers," in *Proceedings of IEEE Infocom 92*, pp. 2332–42, 1992. Florence, Italy, 4-8 May.

[32] P. B. Danzig, S. Jamin, R. Caceres, D. J. Mitzel, and D. Estrin, "An empirical workload model for driving wide-area TCP/IP network simulation," *Internetworking: Research and Experience*, vol. 3, no. 1, 1991.

[33] M. Acharya, R. Newman-Wolfe, H. Latchman, R. Chow, and B. Bhalla, "Real-time hierarchical traffic characterization of a campus area network," in *Proceedings of the Sixth International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 1992.

[34] M. Acharya and B. Bhalla, "A flow model for computer network traffic using real-time measurements," in *Second International Conference on Telecommunications Systems, Modeling and Analysis*, Nashville, TN March 24-27, 1994.

[35] J. Mogul, "Network locality at the scale of processes," in *Proceedings of ACM SIGCOMM '91*, pp. 273–285, September 1991. Zurich, Switzerland.

[36] D. C. Feldmeier, "Improving gateway performance with a routing table cache," in *Proceedings of IEEE Infocom 88*, pp. 298–307, March 1988.

[37] Y. Rekhter and B. Chinoy, "Injecting inter-autonomous system routes into intro-autonomous system routing. a performance analysis," *Internetworking: Research and Experience*, vol. 3, pp. 198–202, July, 1992.

[38] Y. Rekhter, "Forwarding database overhead for inter-domain routing," *ACM Computer Communiations Review*, vol. 23, pp. 66–81, January 1993.

[39] L. Breslau, D. Estrin, D. Zappala, and L. Zhang, "Limited distribution updates to reduce overhead in adaptive internetwork routing." unpublished, February 1993.

[40] D. Estrin, Y. Rekhter, and S. Hotz, "Scalable inter-domain routing architecture," in *Proceedings of ACM SIGCOMM '92*, pp. 40–52, August 1992.

[41] D. Estrin, J. Mogul, G. Tsudik, and K. Anand, "Visa protocols for controlling inter-organizational datagram flow," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 486–98, May 1989.

[42] M. Steenstrup, "An architecture for inter-domain policy routing." Internet Request for Comments Series RFC 1478, June 1993.

[43] M. Steenstrup, "Inter-domain policy routing protocol specification and usage: version 1." Internet Request for Comments Series RFC 1479, July 1993.

[44] C. Topolcic, "Experimental Internet stream protocol, version 2 (ST-II)." Internet Request for Comments Series RFC 1190, October 1990.

[45] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 368–79, April 1990.

[46] D. Verma, H. Zhang, and D. Ferrari, "Guaranteeing delay jitter bounds in packet switching networks," in *Proc. Tricomm 91, IEEE Conference on Communications Software: Communications for Distributed Applications and Systems*, pp. 35–43, April 1991.

[47] A. Lazar and G. Pacifici, "Control of resources in broadband networks with quality of service guarantees," *IEEE Communications Magazine*, vol. 29, no. 10, pp. 66–73, October 1991.

[48] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proceedings of ACM SIGCOMM '89*, pp. 1–12, September 1989.

[49] D. D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: Architecture and mechanism," in *Proceedings of ACM SIGCOMM '92*, pp. 14–26, August 1992.

[50] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: a new resource reservation protocol," *IEEE Network*, vol. 7, pp. 8–18, September 1993.

[51] S. J. Floyd, "Hierarchical link sharing." draft manuscript, available as `ftp.ee.lbl.gov:papers/link.ps.Z`, April 1994.

[52] S. Shenker, D. C. Clark, and L. Zhang, "A scheduling service model and a scheduling architecture for an integrated services packet network." preprint, March 1994.

[53] D. J. Mitzel, D. Estrin, S. Shenker, and L. Zhang, "An architectural comparison of ST-II and RSVP," in *Proceedings of IEEE Infocom 94*, March 1994.

[54] A. Milliken, "Resource coordination object: a state distribution mechanism," tech. rep., BBN, 1993.

[55] W. Willinger, "Variable-bit-rate video traffic and long-range dependence," *IEEE Transactions on Communications*, 1994. forthcoming.

[56] H. Heffes and D. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE Journal on Selected Areas in Communication*, vol. 4, pp. 856–868, April 1986.

[57] D. Anick, D. Mitra, and M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *Bell System Technical Journal*, 1984.

[58] V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling," tech. rep., Lawrence Berkeley Laboratory, February 1994. submitted for publication.

[59] H.-W. Braun and K. Claffy, "Network analysis in support of Internet policy requirements," in *Proc. of INET '93*, pp. FAC:1–11, June 1993.

[60] J.-S. Ahn, P. B. Danzig, D. Estrin, and B. Timmerman, "Hybrid technique for simulating high bandwidth delay computer networks," in *Proceedings of ACM Sigmetrics '93*, May 17-21 1993. Santa Clara, CA.

[61] V. Paxson, "Empirically-Derived Analytic Models of Wide Area TCP Connections." submitted to *IEEE/ACM Transactions on Networking*, May 1993.

[62] V. Paxson, "Empirically-Derived Analytic Models of Wide Area TCP Connections: Extended Report." Technical Report LBL-34086, May 1993.

[63] I. Wakeman, D. Lewis, and J. Crowcroft, "Traffic analysis of trans-Atlantic traffic," in *Proceedings of Inet '92*, pp. 417–430, June 1992.

[64] T. Asaba, K. Claffy, O. Nakamura, and J. Murai, "An analysis of international academic research network traffic between Japan and other nations," in *Inet '92*, pp. 431–440, June 1992.

[65] L. Kleinrock and W. E. Naylor, "On measured behavior of the ARPA network," in *AFIPS Proceedings, 1974 National Computer Conference*, pp. 767–780, 1974.

[66] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications.* Wiley, 1976.

[67] B. Chinoy and H.-W. Braun, "The National Science Foundation Network." Technical report GA-A21029, 1992.

[68] B. Chinoy and P. Smith, "Final version of Aborted T3," *ANS Update*, November 1992.

[69] Y. Rekhter, "NSFNET backbone SPF-based Interior Gateway Protocol." Internet Request for Comments Series RFC 1074, 1990.

[70] J. Case, M. Fedor, M. Schoffstall, and C. Davin, "Simple Network Management Protocol (SNMP)." Internet Request for Comments Series RFC 1157, May 1990.

[71] R. Braden and A. DeSchon, "NNStat: Internet statistics collection package. Introduction and User Guide," Tech. Rep. RR-88-206, ISI, USC, 1988. Available for anon-ftp from `isi.edu`.

[72] ANS, "ARTS: ANSnet Router Traffic Statistics software," 1992.

[73] "Management Information Base for network management of TCP/IP-based internets." Internet Request for Comments Series RFC 1156, May 1990.

[74] "Management Information Base for network management of TCP/IP-based internets, MIB-II." Internet Request for Comments Series RFC 1213, March 1991.

[75] K. Claffy, G. C. Polyzos, and H.-W. Braun, "Application of sampling methodologies to network traffic characterization," in *Proceedings of ACM SIGCOMM '93*, pp. 194–203, September 13-14 1993.

[76] J. B. Postel, "Internet Control Message Protocol." Internet Request for Comments Series RFC 792, September 1981.

[77] N. K. Groschwitz, "Traffic and delay patterns on the NSFNET backbone," Master's thesis, Department of Computer Science and Engineering, University of California, San Diego, 1993.

[78] N. K. Groschwitz and G. C. Polyzos, "A time series model of long-term NSFNET backbone traffic," in *Proceedings of IEEE International Conference on Communications (ICC '94)*, May 1994.

[79] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control.* Holden-Day, 1970.

[80] "Network information services." Data available on `nis.nsf.net:/nsfnet/statistics`.

[81] J. Smith, "Nsfnet traffic visualization by U.S. state." report of NSF Research Experience for Undergraduates (REU) project, September 1993.

[82] J. Smith and B. Chinoy, "Nsfnet backbone growth trends: do more networks mean more traffic?." report of NSF Research Experience for Undergraduates (REU) project, September 1993.

[83] J. B. Postel, "Internet Protocol." Internet Request for Comments Series RFC 791, September 1981.

[84] S. Kirkpatrick, M. Stahl, and M. Recher, "Internet numbers." Internet Request for Comments Series RFC 1166, July 1990.

[85] E. Gerich, "Guidelines for management of IP address space." Internet Request for Comments Series RFC 1366 obsoleted by RFC 1466, October 1992.

[86] E. Gerich, "Guidelines for management of IP address space." Internet Request for Comments Series RFC 1466 obsoletes RFC 1366, May 1993.

[87] P. Ford, Y. Rekhter, and H.-W. Braun, "Improving the Routing and Addressing of the Internet protocol," *IEEE Network*, vol. 7, pp. 10–15, May 1993.

[88] K. Claffy, H.-W. Braun, and G. C. Polyzos, "Statistics collection for the T3 NSFNET backbone service." UCSD Technical report CS93-275, 1993.

[89] J. Reynolds and J. Postel, "Assigned numbers." Internet Request for Comments Series RFC 1340, July 1992.

[90] P. Danzig, K. Obraczka, and S.-H. Li, "Internet resource discovery services," *IEEE Computer*, pp. 8–22, September 1993.

[91] R. Bohn, H.-W. Braun, K. Claffy, and S. Wolff, "Mitigating the coming Internet crunch: multiple service levels via precedence," *Journal of High Speed Networks*, forthcoming 1994. available for anon-ftp at `ftp.sdsc.edu:pub/sdsc/anr/papers/`.

[92] Y. Rekhter and T. Li, "An Architecture for IP address allocation with CIDR." Internet Request for Comments Series RFC 1518, September 1993.

[93] I. S. Organization, "Fiber distributed data interface (fddi) - media access control," 1989.

[94] L. Kleinrock, *Queueing Systems, Volume I: Theory.* Wiley, 1976.

[95] R. W. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, March 1982.

[96] W. Cochran, *Sampling Techniques.* John Wiley & Sons, 1987.

[97] P. Krishnaiah and C. Rao, *Handbook of Statistics, Volume 6: Sampling.* North-Holland, 1988.

[98] L. Goodman and W. Kruskal, "Measures of association for cross classifications," *Journal of the American Statistical Association*, pp. 732–763, December 1954.

[99] R. B. D'Agostino and M. A. Stevens, eds., *Goodness of Fit.* Marcel Dekker, Inc., 1986.

[100] T. W. Anderson and D. Darling, "Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes," *Annals of Mathematical Statistics*, vol. 23, pp. 193–212, 1954.

[101] J. Fleiss, *Statistical Methods for Rates and Proportions.* John Wiley & Sons, 1981.

[102] D. D. Clark, "The design philosphy of the Darpa Internet protocols," in *Proceedings of ACM SIGCOMM '88*, pp. 16–19, August 1988.

[103] M. Laubach, "Classical IP and ARP over ATM." Internet Request for Comments Series RFC 1577, January 1994.

[104] J. Heinanen, "Multiprotocol encapsulation over ATM adaptation layer 5." Internet Request for Comments Series RFC 1483, July 1993.

[105] D. Grossman, M. Perez, F. Liaw, E. Hoffman, and A. Mankin, "ATM signaling support for IP over ATM." Internet-Draft, April 1994.

[106] S. Deering, "Simple internet protocol plus (SIPP) specification." Available as ds.internic.net:internet-drafts/draft-ietf-sipp-spec-00.txt, February 1994.

[107] R. Caceres, "Multiplexing data traffic over wide-area cell networks." preprint, March 1993.

[108] H. Saran and S. Keshav, "An empirical evaluation of virtual circuit holding times in IP-over-ATM networks," in *Proceedings of IEEE Infocom 94*, June 1994. to appear.

[109] M. R. Macedonia and D. P. Brutzman, "Mbone provides audio and video across the Internet," *IEEE Computer*, vol. 27, pp. 30–36, April 1994.

[110] S. Casner, August 1993. personal communication.

[111] F. Baker, "personal communication (e-mail)," October 1993. Advanced Computer Communications.

[112] A. Mankin and K. Ramakrishnan, "Gateway congestion control survey." Internet Request for Comments Series RFC 1254, August 1991.

[113] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance.* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[114] R. Caceres, *Multiplexing Traffic at the entrance to wide-area networks.* PhD thesis, University of California, Berkeley, December 1992. ICSI TR 92/717.

[115] B. Stockman, "A Model for Common Operational Statistics." Operational Statistics Working Group, 1992.

[116] C. Mills, D. Hirsh, and G. Ruth, "Internet accounting: background." Internet Request for Comments Series RFC 1272, November 1991.

[117] R. Cocchi, *Pricing in multiple service class computer communication networks.* PhD thesis, University of California, Berkeley, 1992.

[118] R. Cocchi, D. Estrin, L. Zhang, and S. Shenker, "A study of priority pricing in multiple service class networks," in *Proceedings of ACM SIGCOMM '91*, September 1991.

[119] C. Parris and D. Ferrari, "A resource based pricing policy for real-time channels in a packet-switching network." ICSI TR 92-018, 1992.

[120] J. MacKie-Mason and H. Varian, "Pricing the Internet," in *Public Access to the Internet* (B. Kahin and J. Keller, eds.), Prentice-Hall, 1994. Available from ftp://gopher.econ.lsa.umich.edu/pub/Papers.

[121] H.-W. Braun, K. Claffy, and G. C. Polyzos, "A framework for flow-based accounting on the Internet," in *Proceedings of Singapore International Conference on Networks (SICON'93)*, pp. 847–851, September 1993.

[122] E. R. Stine, "FYI on a network management tool catalog: Tools for monitoring and debugging TCP/IP internets and interconnected devices." Internet Request for Comments Series RFC 1147, April 1991.