Long-term traffic aspects of the NSFNET [1]

Kimberly C. Claffy
kc@cs.ucsd.edu
George C. Polyzos
polyzos@cs.ucsd.edu
Computer Systems Laboratory
University of California, San Diego
La Jolla, CA 92093-0114

Hans-Werner Braun
hwb@sdsc.edu
San Diego Supercomputer Center
San Diego, CA 92186-9784

Abstract

We present the architecture for data collection for the T3 NSFNET backbone service, and difficulties with using the collected statistics for long-term network forecasting of certain traffic aspects. We describe relevant aspects of the T3 backbone architecture, describe the instrumentation for the statistics collection process, and how it differs from that on the T1 backbone. We then present long-term NSFNET data to elucidate long term trends in both the reachability of Internet components via the NSFNET as well as the growing cross-section of traffic. We focus on the difficulties of forecasting and planning for these two traffic aspects in an infrastructure whose protocol architecture and instrumentation for data collection was not designed to support such objectives.

I. Introduction

NSFNET, the National Science Foundation Network, is a general purpose packet-switching network supporting access to scientific computing resources, data, and interpersonal electronic communications.[2] Claffy *et al.* [8] present a description of the national backbone networking environment and the instrumentation for the data collection process for the now dismantled T1 NSFNET backbone. Evolved from a 56kbps network among NSF-funded supercomputer centers in the mid-1980s to today's 45Mbps network which serves a much broader clientele. The current larger NSFNET encompasses not only the transcontinental backbone connecting the NSF-funded supercomputer centers and mid-level networks, but also mid-level networks themselves, and 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 1 shows the logical topology of the backbone.

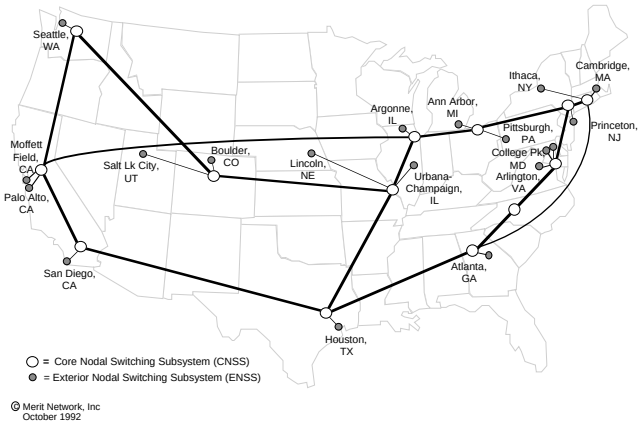

Figure 1: 1992 NSFNET T3 Backbone Service Logical Topology

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. The T3 network provides 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 has 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

[1] This research is supported by a grant of the National Science Foundation (NCR-9119473), and a joint study agreement with the International Business Machines, Inc.

[2] Chinoy and Braun [5] offer a more detailed description of the NSFNET.

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 paper we refer to the "T3 NSFNET backbone" with the understanding that we are referring to a service provided to NSF, not a dedicated physical network.

The purpose of this paper is to present the architecture for data collection for the T3 NSFNET backbone service, and difficulties with using the collected statistics for long-term forecasting of certain traffic aspects. In the next section we describe relevant aspects of the T3 backbone architecture. Section III describes the instrumentation for the statistics collection process on T3 backbone, and how it differs from that on the T1 backbone. Section IV presents a discussion of the IP network address structure, and how the status of an IP address relates to the evolution of available Internet network numbering space. In Section V we discuss the growth in application/service diversity on the Internet as measured by TCP/UDP port numbers.

Our statistics reflect operational collection of traffic and network registration statistics: IP network number growth, which provides a metric for the increasing geographic and administrative scope of the Internet; and port numbers, which provide a metric for planning for the growing cross-section of traffic.[3] We focus on the limitations of these two methodologies, both of which were initially designed to support short term engineering and planning needs, such as routing and tracking the rough cross-section of traffic. Suboptimalities in their architecture and implementation prevent their effective usage for some long-term forecasting and planning objectives. For example, the Internet architecture makes it inherently difficult to track many applications by TCP/IP port number. The situation will pose a serious obstacle to long-term planning with the growth of continuous real-time media applications, which are able to continuously block significant fractions of the available bandwidth.

[3] The data for the statistics presented in this report were gathered by an NSFNET NSS software package which aggregates information using the ARTS software package. This compilation, greatly assisted by ANS, Merit Network, Inc., and other installations, reflects an effort to capture as much and as accurate data as possible. However, no guarantee is given for the completeness of the data or its accuracy.

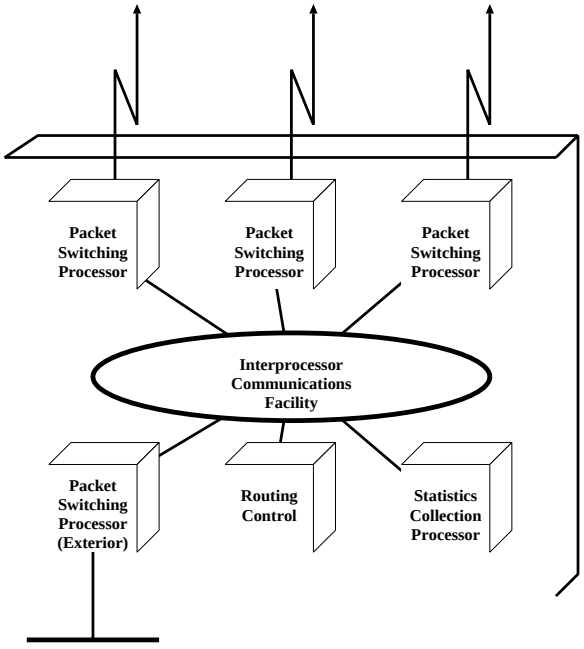

Figure 2: T1 NSS architecture

II. 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 [6] 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 client 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 cloud 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 with minimal, or no 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, facilitating path redundancy.

The serial line interfaces to each node on the T3 backbone are of "T3", or DS3 speed, 44.736

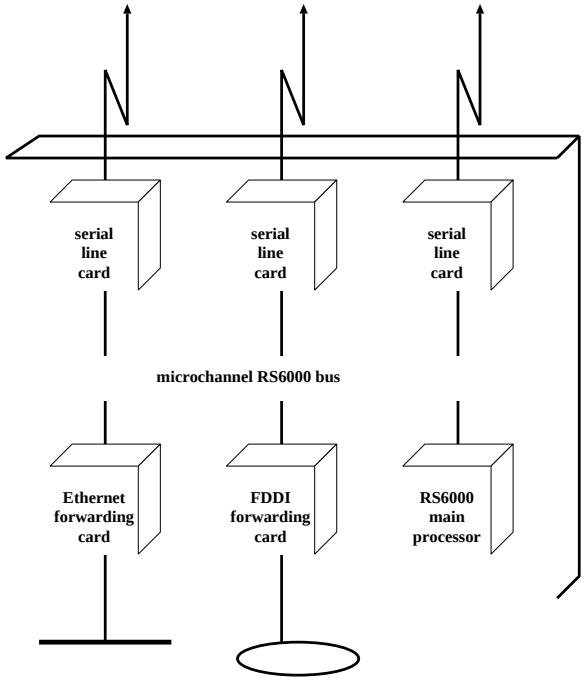

Figure 3: T3 C-NSS architecture

Mbits/second.[4] 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 [6]. To access external client networks, the T3 backbone nodes currently use Ethernet and FDDI interfaces, with packet size limits of 1.5 and 4 kilobytes, respectively. Each packet is also embedded within an Ethernet or FDDI frame, which the LAN drivers at 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 [19]. 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 2 and 3 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[5] 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[6] bus, without the intervention of the main processor.

III. Statistics collection mechanisms

In this section we describe the mechanisms for data collection on the T3 backbone, and how they differ from those used on the T1 backbone. The principal sources of information about the T3 backbone come from routine collection of three classes of network statistics: interface statistics; packet categorization; and internodal delays. Interface statistics derive from programs using the Simple Network Management Protocol (SNMP) [4]. Specialized software packages perform packet categorization: the T1 backbone utilized the NNStat [3] package for collection; the T3 backbone utilizes the ARTS (ANSnet Router Traffic Statistics) [1] package, which encompasses similar functionality.

III.A. Interface performance

The mechanism for collecting interface performance statistics did not change from the T1 to the T3 backbone. To maintain data regarding packets and bytes transmitted and received, errors, delay times, and down times, all backbone nodes record statistics about the packets which traverse each of their interfaces. Each backbone node runs SNMP servers which respond to queries regarding SNMP Management Information Base (MIB) variables. Centralized collection of the data 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 SNMP, include those for

[4] There are T1 backup links as well; we will not focus on these backup links in our discussion of the architecture.

[5] RISC System/6000 is a trademark of IBM Corporation.

[6] Microchannel is a trademark of IBM Corporation.

Table 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

packets, bytes, and errors transmitted in and out of each interface.[7] Table 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.[8]

III.B. 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 2, 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 would examine the header of every packet traversing the intra-NSS processor intercommunication facility, and use a modified version of the NNStat package [3] to build statistical objects based on the collected information. 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 [7].

The design of the T3 backbone required significant modification to 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 illustrates the current design of the backbone nodes, which offloads the forwarding capability to the cards as described in Section II. 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*,[9] to the main CPU, where the collection software performs the traffic characterization based on these sampled packets. Note that multiple subsystems, including those connected to T3, Ethernet, and FDDI external interfaces, forward to the RS/6000 processor in parallel.

Because the main CPU card performs the categorization, 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. The collection host is an IBM RS/6000 at the ANS NOC; as an example of the memory requirements, this machine collected approximately

[7] Error conditions on the interface include HDLC checksum errors, invalid packet length, and queue overflows resulting in discards. The single counter does not distinguish among these error conditions.

[8] Object definitions found in McCloghrie and Rose [14] [15].

[9] 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 [2].

Table 2: Packet categorization objects on T1 and T3 backbone nodes

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

25 MB of ARTS traffic characterization data during a typical workday in mid-February 1993.[10] Table 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 the 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 according to protocol (e.g., TCP, UDP, ICMP), and TCP/UDP port (application).

III.C. 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 [17] to record the 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 II. 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. 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. The new report format 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.

IV. What's in an IP address

In the Internet environment network clients must acquire IP network number assignment. These number assignments indicate to some degree the growth in scope of the network, although some network clients use multiple network numbers. Over the years the number of allocated net-

[10] ANS is now using a more efficient binary format to store data than that used on the T1 backbone.

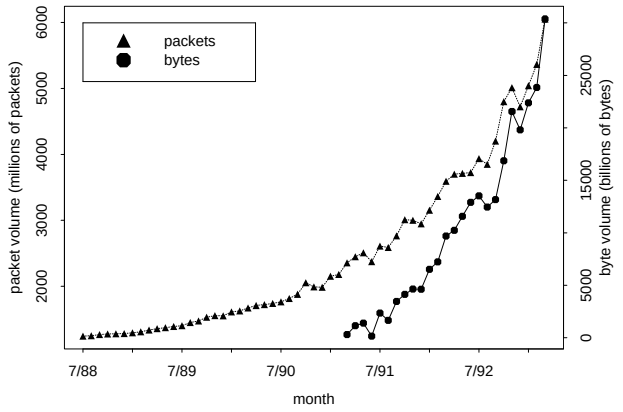

Figure 4: Long-term growth of NSFNET traffic *(Data source: Merit/NSFNET operations)*

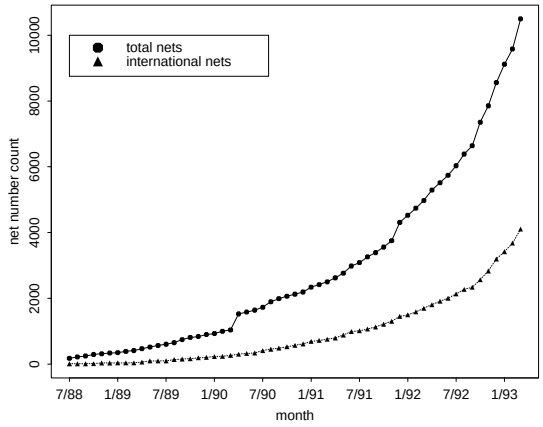

Figure 5: Long-term growth of NSFNET network numbers *(Data source: Merit/NSFNET operations)*

work numbers has grown from below 100 to many thousands.

Figure 4 shows the long term growth of NSFNET backbone traffic, in terms of both packets and bytes. Both curves exhibit quadratic growth during the last five years of the NSFNET backbone operation. The graph begins at the establishment of the T1 NSFNET backbone in mid-1988; the T3 gradually replaced the T1 during 1992, and as of November 1992 assumed all NSFNET backbone traffic forwarding responsibilities.

Each packet was generated at and destined to specific *IP host addresses*, as specified in the header of each IP packet. Three commonly used classes determine the size of the host address component within the four byte IP address fields: Class A, B, and C networks with three, two, and one-byte host field, 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 [13].

Figure 5 shows the long term growth of network numbers configured for communication via the NSFNET backbone [16]. These configured NSFNET numbers are the only destinations to which the NSFNET backbone will route packets, and also exhibits quadratic growth over the last few years, including substantial increases in the international area.

A visible jump in the total net number count in early 1990 reflects a change in NSFNET treatment of Arpanet routes. Since its introduction, the T1 NSFNET backbone had always retained full routing knowledge in each backbone node rather than relying on a default routing scheme. However the NSFNET backbone nodes accepted all Arpanet routes until early 1990 even though they were not included in the NSFNET policy routing database. In early 1990 the NSFNET explicitly configured these Arpanet routes in order to be consistent with the configuration of all other NSFNET backbone clients. This addition caused the visible jump in the graph, but we note that the NSFNET knew about and routed traffic to many of those networks before.

Figure 6 shows the long term growth of network number assignments by class. As mentioned above, Class A, B, and C networks differ in the number of host addresses they can support. Over the years the Network Information Center has assigned IP network numbers to clients according to the number of hosts required. The growth of the Class B number space is of particular interest; about 40% of the currently available space is assigned. The Class B space is a very attractive one to use if one expects to eventually use subnetting of IP network numbers. Anticipated depletion of the Class B address space has led to significant efforts toward augmenting the IP architecture, including a methodology to forestall depletion and routing table explosions within the framework of the currently deployed architecture. This methodology, called Classless Inter Domain Routing (CIDR), uses clustered Class C numbers as an alternative to Class B numbers, with network masks allowing for a number aggregation [9].

Although these graphs give some sense of the increasing geographic and administrative scope of the NSFNET, a discussion of the significance of the IP addresses structure to the infrastructure will allow clearer understanding of the growth in service reachability of the NSFNET. We present IP addresses as they are registered with the NSF Internic at the top level, and then delegated to other authorities from there. Our primary focus here is the distribution and registration of IP network numbers, and not issues such as network classes, multicasting, or CIDR [9].

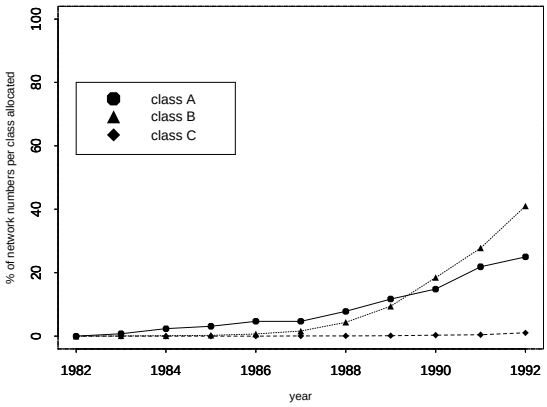

Figure 6: Long-term growth of assigned Class A, B, and C IP network numbers *(Data source: Internic, personal communication)*

The IP address space architecture originated with RFC 791 [18], 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 Registrar, on behalf of the Internet community, now formally registers these *assigned* network numbers in a data base that also includes address information of the institution responsible for the network, and other attributes. With the advent of RFC1366 [10] [11] in October 1992, the Internic began to assign addresses according to the geographic location of the requestor, with a strong preference for assigning single or multiple class C addresses rather than a Class B address. Since one or several class C addresses are typically sufficient to fulfill the request of a single geographically defined network domain, this methodology throttled the class B depletion but accelerated the class C depletion.

The Internic also in certain cases *delegates* blocks of class C IP network numbers to other authorities for further assignment. For example, the NIC assigned a large portion of the class C space to Europe for further redistribution within their network community. From the NIC's point of view, these *delegated* 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 IP networks.

Networks that are NIC-*assigned* do not by definition actively exchange traffic on the Internet. In fact, the set of communicating, or Internet *active* network numbers, is not even necessarily a proper subset of the set of *assigned* network numbers (although in a frictionless world, it would be). Some organizations consider their local network environments wholly disconnected from the Internet, and with no plans for future connection, they sometimes even choose their own IP network numbers, independent of the NIC's registry, to satisfy their isolated TCP/IP protocol needs. Unfortunately, experience has shown that such disconnected environments often turn out to be quite *leaky*. When traffic from these networks manages to find its way into the Internet, often much to the surprise (or ignorance) of the local network administrators, these network numbers join the set of *leaky unassigned* numbers. *Leaky unassigned* numbers are members of the *active* set of numbers that are not in the *assigned* set.

An important component of the U.S. Internet is the NSFNET and its backbone network. Possession of an *assigned* network number is necessary but not sufficient for communication across the NSFNET backbone. The NSFNET backbone uses a policy routing database as a truth filter, to ensure that it believes only the selected dynamic routing information which its backbone clients have explicitly specified.[11] This database represents the set of *NSFNET-configured* network numbers which 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 know about and thus be able to service 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 protocol such as BGP or EGP. The 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* net. The configuration database thus 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.

Once a network is *assigned*, *configured*, and *announced*, it can both send and receive traffic over the NSFNET backbone as an *active* network. An

[11] NSFNET operators routinely update (approximately every two weeks) this database of mappings of network numbers to Autonomous System numbers, with associated preference metrics.

active network will remain active as long as connectivity exists to the destination and the appropriate service provider(s) *announces* the network directly to the backbone.

As mentioned above, Internet reality is not entirely faithful to this model. Theoretically, *any* network which sends traffic is *active*, even if it is not *assigned*, *configured*, and announced. To disambiguate the categories, 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. *Leaky* networks, particularly from *unassigned* networks, pose difficulties for network operators since they can inject bogus information even into inter-administrative domain routing protocol exchanges. Operators such as those of the NSFNET backbone must then undertake efforts to sanitize their network topology information. For this reason, the Internet Assigned Numbers Authority (IANA)[12] has since the beginning discouraged the TCP/IP community from using custom design of network numbers, considering it uncivilized behavior in an increasingly, and often seemingly transparently, interconnected world.

Equally problematic is the case of *silent* networks: networks which are *configured* or even *announced* but did not send traffic across the backbone during the month. The NSFNET project has undertaken efforts to analyze the NSFNET Policy Routing Database (PRDB) and develop methods of expediting the elimination of the *silent* nets in order to prevent potential operational difficulties due to routing table size. The problem of growth of the silent networks has intensified with the addressing guidelines of RFC1366 outlined above. Under these guidelines service providers receive large blocks of class C addresses in anticipation of and aligned with CIDR requirements, and immediately configure them in the policy routing database before actually assigning them to customers who will announce them. The result is a substantial increase in the number of *silent* networks in the NSFNET backbone configuration database. Eventual CIDR deployment will rely on net masks to reduce such blocks to a single entry in the routing table, but until that time they still pose an obstacle to efficient configuration.

Figure 7 presents a schematic of the categories of network numbers we have discussed. Engineers on operational networks must contend with these issues in the design of their traffic collection mechanisms. For example, each NSFNET T3 backbone router samples every fiftieth packet to build traffic characterization objects, in particular to create a source-destination matrix by network number. The router samples these packets before actually routing them, and thus it is conceivable that the routers will capture packets from IP network numbers which are not in the routing database, although assigned by the NIC, or even from network numbers which are neither in the routing topology nor assigned by the NIC. We will call the latter set *unassigned networks*, (although *anarchically* picked network numbers may be a more fitting term).

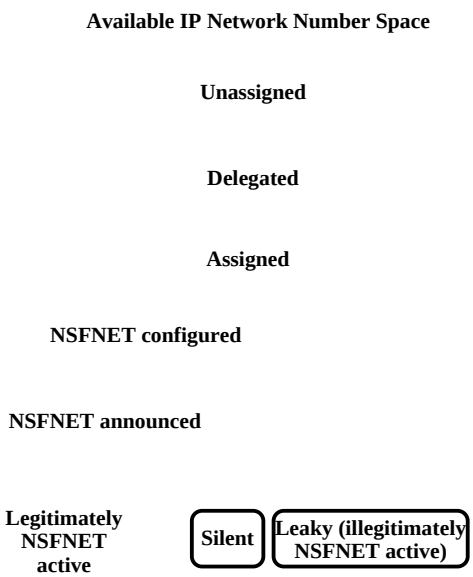

Figure 7: Descriptive categories of IP network numbers

Thus, the statistics which the NSFNET generates will include many *unassigned* or *unconfigured* networks. For the purpose of statistics analysis, there are a variety of ways to treat inactive networks, including treating them all as *unconfigured*, which risks not attributing what may be non-negligible amounts of traffic which they impart to the backbone entry point.

As an example, we now discuss the month of December 1992 on the NSFNET/ANSnet backbone. During this month, the data collection mechanism on the T3 backbone nodes recorded traffic from more than 14,000 networks. This set included networks from the entire set of *available network numbers*. Of these, about 9,700 were networks which were in the set of NIC-*assigned network numbers*. The number of *active* networks that had also been *configured* in the NSFNET/ANSnet topology data base that month was about 6,400.

To explain the large number of non-configured networks represented in the collected traffic matrix,

[12] The IANA is currently represented by Jon Postel and Joyce Reynolds of USC-ISI.

we use the terminology outlined above to describe several violations of our model. A non-configured *leaky* network may source, or send, traffic into the NSFNET backbone. Those packets may actually get delivered to the remote location, if the remote location is *assigned*, *configured* and *announced*, but traffic for the return path to the original source will not be delivered via the NSFNET.

Alternatively, a network could send traffic to another network which the NSFNET backbone does not know about for a variety of possible reasons: because the network is configured but not announced; because the network is assigned but not configured; or because the network is not even assigned. The NSFNET may see such traffic, for example when network service providers use a default route pointing to the NSFNET. However, since routing information for these destinations will not exist in the NSFNET forwarding tables, as soon as such packets reach the NSFNET, the backbone node will filter them out during the routing decision.

V. What's in a port number

Another increasing difficulty in characterizing long-term trends in the traffic on the NSFNET is the wide cross-section of applications, whose profile is ever-increasing in diversity and scope. Assessment of this profile will be critical to network service planning, e.g., as continuous flows such as audio and video impact the performance of conventional bursty traffic. In this section we describe how the collection methodology currently used to track the traffic cross-section is becoming increasingly insufficient.

The majority of applications on the NSFNET are built on top of the Transmission Control Protocol (TCP), and some on top of 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 fields for the 16 bit values identifying the source and destination ports of the packet. Originally, the Internet Assigned Numbers Authority,[13] on behalf of DARPA, administered a space of 1 to 255 as the group of *well known* or *assigned* port numbers, reserved for specific applications. For example, *telnet* received port assignment 23 [12]. 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. (In the case of telnet, the packet uses some arbitrarily assigned source port that has significance only to the originating host. Often these "return address ports" have values greater than 1000.)

[13] ISI (Information Sciences Institute, University of Southern California) houses IANA).

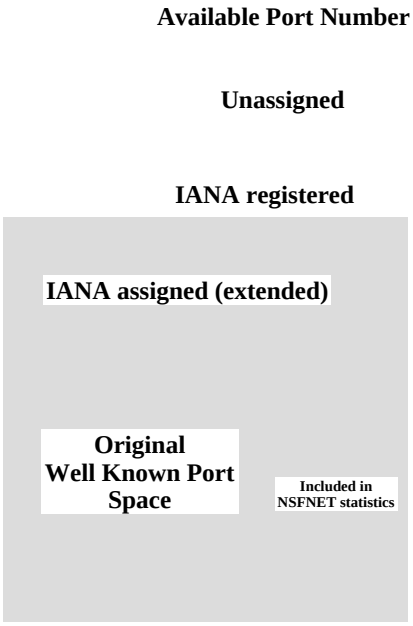

Figure 8: Descriptive categories of port numbers

Although IANA administers the number range for *well known* or *assigned* Port numbers, at some point Unix developers injected a bit of anarchy into the system by unilaterally assuming that numbers below 1024 identify specific applications. They then began to use that numbering space as they deployed applications, such as port 513 for rlogin. Eventually network users began to use numbers even above 1024 to specify further services, extending the lack of community coordination further. Examples include XDR/NFS (port 2049), and X-Windows (port 6000+), and port 4444 for (some) MBONE video multicasts. In July 1992 [12] the IANA extended the range of ports for which it manages assignments to the 0-1023 range. At this time the IANA also began to track, to the best of its ability, a set of *registered* ports within the full range of 1024-65535. IANA does not attempt to control the assignments of these ports; but only registers uses of which is is aware as a convenience to the community [12].

These port numbers are the only mechanism via which the NSFNET can monitor statistics on the aggregated distribution of applications on the backbone. Thus the proliferation of uncoordinated number assignments imposes ambiguity into this categorization of packets by application.

Figure 8 presents a schematic of the categories of port numbers we have discussed. For NSFNET statistics gathering on port distribution for the

backbone, Merit (and now ANS) specifically 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 matched one of these numbers. However even within this range not all ports have a generally known assignment, so packets using such undefined ports go into an *unknown* port category [16].

Figures 9 and 10 use this collected data to categorize the proportion of traffic on the network by category since August 1989. These figures illustrate the difficulty of tracking changes in the cross-section of traffic on the backbone.[14] In this figure the "non-tcp services" category corresponds to applications using a transport protocol other than TCP or UDP; the "other tcp services" category to non-standard or not well-defined ports. Both of these categories have grown much larger over the years, reflecting in the first case an increasingly multi-application environment, and in the second the diminishing ability to track individual new applications which often use non-standard or not well-defined ports. In fact, the "other tcp services" category is, as of November 1992, the largest single category of traffic (in packets) on the backbone, exposing the trend of application developers arbitrarily choosing their own port numbers for applications that collectively utilize substantial bandwidth. Since these port numbers are undefined to anyone but the end site using them, the growth of traffic volume for such applications is difficult to track; most statistics collection mechanisms can only attribute traffic to well-known port numbers, making attribution of more than the base services (telnet, ftp, etc.) close to impossible.

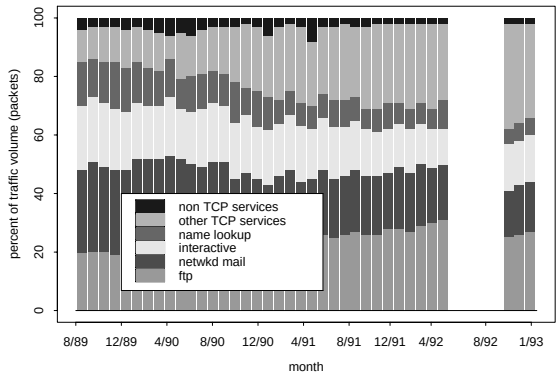

Figure 9: Distribution of packets offered into NSFNET backbone by protocol *(Data source: Merit/NSFNET operations)*

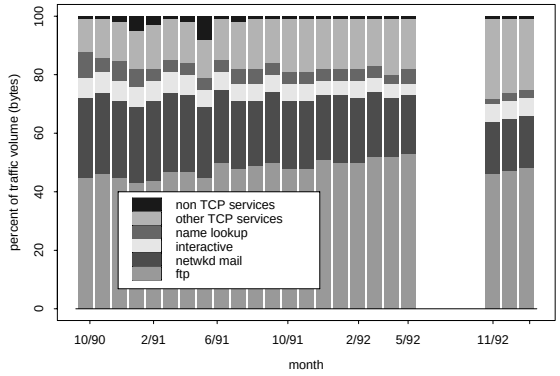

Figure 10: Distribution of bytes offered into NSFNET backbone by protocol *(Data source: Merit/NSFNET operations)*

[14] The categories in these figures correspond to:

- File exchange: ftp data and control (tcp ports 20, 21)
- Mail: smtp, nntp, vmnet, uucp (tcp ports 25, 119, 175, 540)
- Interactive: telnet, finger, who, login (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)
- Non-TCP/UDP services Internet protocols other than tcp or udp (e.g. icmp, igmp, egp, hmp, ax.25, etc.)

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 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.

VI. Forecasting traffic type

The issue of unknown applications is not by itself necessarily 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 not able to preferentially deal with certain traffic. In this section we describe the dangers of our increasing inability to monitor traffic type in a "high-end" Internet.

Today's Internet is inherently based on a datagram architecture with typically no admission control in packet forwarders. Most entrance points into transit networks can not sufficiently provide back pressure to other points of the network that inject more traffic than the network can handle. End systems can thus unfairly monopolize available bandwidth and cause significant congestion in the larger network.

During the mid-80s on the 56kbps NSFNET backbone, this state of congestion developed to a dangerous degree, and in response the NSFNET engineers deployed an emergency measure to provide certain interactive network applications, specifically Telnet, preferential treatment over other traffic. The priority transit allowed interactive users requiring better network responsiveness to continue working under highly congested circumstances.

Since that time the principal means of addressing network congestion has been to increase network capacity. However today software developers continue to build advanced network applications which can consume as much bandwidth as network engineers provide. In particular, applications using packet audio and video do not exhibit the same "burstiness" characteristics of more conventional applications such as file transfer and electronic mail, but rather require continuous delivery of large amounts of traffic in "real-time", and thus continuously consume significant bandwidth. Clearly usage of such applications will not scale in the current Internet architecture, which may potentially need to support many such continuous point-to-point connections simultaneously.

It is difficult to overestimate the dramatic impact which digital continuous media will have on the Internet fabric. No other phenomenon could more strongly drive the research community to instrument the network for admission control and multiple service levels, as well as accounting and billing. Prerequisite to accounting and billing instrumentation is a more accurate model for the attribution of resource consumption, derived from how particular applications impact network performance. Such a model may have to reliably attribute applications, or traffic profiles, to the clients if multiple levels of services exist.

VII. Summary

We have presented the architecture for data collection for the T3 NSFNET backbone service, and limitations of the approach being used for long-term network forecasting and planning. In particular, we discuss:

1. the IP address structure and its application to NSFNET transit;

2. port numbers, the implementation limitations of which prevent real tracking of service diversity.

Our statistics reflect operational collection of traffic and network registration statistics, both initially designed to support short term engineering and planning needs. Traditionally, statistics used in forecasting measure compounded traffic volume at network access points or individual network interfaces, which network planners extrapolate for indicators of future performance requirements. Although such statistics allow some tracking of Internet growth, they limit our ability to forecast capacity requirements in a network with ever richer functionality. Today's Internet aggregates traffic from among many clients, with various applications with various associated service qualities. To investigate beyond such traditional metrics of network usage we quantify for the current NSFNET environment aspects of network ubiquity, as measured by IP network numbers, and the multiplicity of services, as measured by port usage statistics.

These statistics indicate superlinear growth of IP network numbers, and therefore Internet clients, over the last several years. The trend is clearly continuing at a global scale; international clientele now account for over 40% of the network numbers known to the U.S. infrastructure. As the need to attribute network usage intensifies, e.g., for accounting and billing purposes, the currently available data sets will seem even more inadequate. Deployment of network number aggregation techniques (e.g., CIDR), which hide the interior structure of a network cluster, will further aggravate the situation.

We also investigated the 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 reassessment of network flow mechanisms such as queuing 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 are different enough to justify adaptive queue management.

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

Acknowledgements and Support

We would like to express our appreciation for the cooperation and assistance we received from Merit Network, Inc., and Advanced Network Services (ANS). In particular we would like to thank David Bolen and Jordan Becker of ANS, and Mark Knopper and Susan Horvath of Merit for many informative discussions.

References

[1] ANS. ARTS: ANSnet Router Statistics software, 1992.

[2] J. Becker. personal communication, January 1993. e-mail about T3 backbone.

[3] R.T. Braden and A. DeSchon. NNStat: Internet statistics collection package. Introduction and User Guide. Technical Report RR-88-206, ISI, USC, 1988. Available for a-ftp from `isi.edu`.

[4] J.D. Case, M. Fedor, M.L. Schoffstall, and C. Davin. Simple Network Management Protocol (SNMP). Internet Request for Comments Series RFC 1157, 1987.

[5] B. Chinoy and H.-W. Braun. The National Science Foundation Network. Technical Report GA-A21029, SDSC, 1992.

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

[7] K. Claffy, H.-W. Braun, and G. Polyzos. Application of sampling methodologies to network traffic characterization. to appear in SIGCOMM '93, September 1993.

[8] K. Claffy, G. C. Polyzos, and H.-W. Braun. Traffic characteristics of the T1 NSFNET backbone. In *Proc. INFOCOM '93, San Francisco, CA*, April 1993.

[9] P. Ford, Y. Rekhter, and H.-W. Braun. Improving the ROuting and Addressing of the Internet protocol. *IEEE Network*, May 1993.

[10] E. Gerich. Guidelines for management of ip address space. Obsoleted by RFC 1466, October 1992.

[11] E. Gerich. Guidelines for management of ip address space. Obsoletes RFC 1366, May 1993.

[12] J. Postel J. Reynolds. Assigned numbers. 138 pages, July 1992.

[13] S. Kirkpatrick, M. Stahl, and M. Recher. Internet numbers. RFC1166, July 1990.

[14] K. McCloghrie and ed. M. T. Rose. Management Information Base for network management of TCP/IP-based internets, mib-ii. Internet Request for Comments Series RFC 1213, 1991.

[15] K. McCloghrie and M. T. Rose. Management Information Base for network management of TCP/IP-based internets. Internet Request for Comments Series RFC 1156, 1990.

[16] Network information services, February 1993. Data available on `nis.nsf.net: /nsf/statistics`.

[17] J. B. Postel. Internet Control Message Protocol. RFC 792, 1981.

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

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