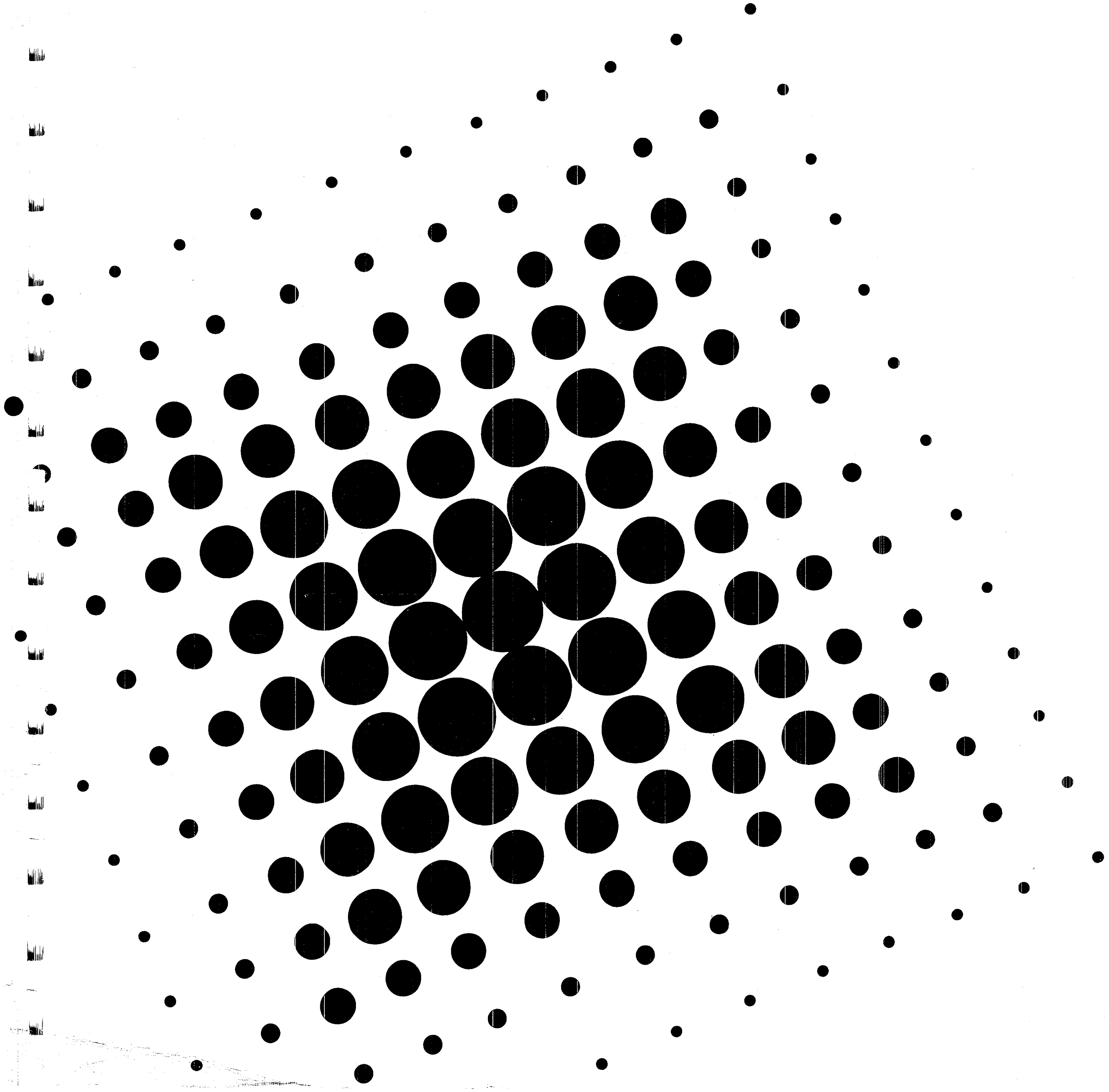
Proceedings of the Fourteenth
Internet Engineering Task Force

July 25-28, 1989

Stanford University, California

Corporation
for
National
Research
Initiatives



PROCEEDINGS OF THE FOURTEENTH INTERNET ENGINEERING TASK FORCE
JULY 24-28, 1989
STANFORD UNIVERSITY, CALIFORNIA

Compiled and Edited
by

Phillip G. Gross
Karen L. Bowers

Corporation for National Research Initiatives
1895 Preston White Drive, Suite 100
Reston, Virginia 22091

ACKNOWLEDGEMENTS

The Fourteenth IETF session was held on July 25-28, 1989 at Stanford University in California. This was a most incredible session! The Working Groups experienced a new burst of energy accompanied by an "all time high" level of productivity. Without a doubt, this productiveness was encouraged by the special atmosphere we experienced at Stanford. On behalf of the IETF, I would like to thank our host, Paul Baer, for making all the special arrangements to accommodate our fast pace. We all felt quite at home.

Susie Karlson and Valarie Collins worked hard to provide all the necessary administrative support and to minimize the few inconveniences we experienced. They did a top notch job, especially considering the large attendance and the increase in the number of Working Group sessions. Keeping track of all of us was a full time job in itself!

I would also like to extend a thank you to all the Stanford folks who provided special tours and presentations for the IETF members: Ms. Kathy O'Shaughnessy for the tour of the Stanford Linear Accelerator Center; Ms. Annette Richards and Messrs. Robert Bates and Roy Stegman for the demonstration of the 16th century machine; and Messrs. Stu Grossman and Martin Frost for the WAITS presentation. We certainly enjoyed the time we spent with you.

Finally, once again I want to thank Monica Hart (NRI) for her ever-enduring patience and the effort she expended to further improve the quarterly IETF Proceedings.

Karen L. Bowers

Table of Contents

I. Chairman's Message

Phill Gross

NRI

CHAIRMAN'S MESSAGE

July was an exciting month for both the IETF and the IAB. Both organizations made important changes to their organizational structure. In a following article, Vint Cerf, the new chair of the IAB, will summarize the new overall IAB/IETF/IRTF restructuring. I will give more specific detail about the IETF changes here.

IETF Growth and Highlights

The IETF has seen a ten-fold growth since its founding in January 1986. As late as the July 1986 meeting at Merit, the IETF was still essentially a 25 person group of government contractors. There was very little vendor involvement in those days. There were no working groups and each meeting tended to be composed of network status reports and technical reports by contractors.

The first working groups did not appear until the meeting at Ames Research Center in February 1987 (see table of meetings below). That meeting was also notable for kicking off a continuing focus on network management in the Internet (or, more properly, the lack thereof) and for the introduction of Van Jacobson to the IETF. At the Ames meeting, Van presented his analysis of how TCP/IP was self-clocking in the Internet. He also presented work that would later blossom into the TCP slow-start algorithm and improved round-trip time estimation. Since that meeting, he has continued to present important work at IETF meetings on such issues as TCP header prediction, congestion control by random (rather than most-recently-arrived) packet dropping, and pathological instances of network synchronization.

The first IETF meeting to have greater than 100 attendees was at MITRE in July 1987. This was due in part to a great interest in network management that had been growing since the Ames meeting. I believe it was at MITRE that Marty Schoffstall (VP of Research and Technology at NYSERnet, Inc) made the first detailed presentation on SGMP to the IETF.

Over the next few meetings, the IETF begin to hit a technical stride with the formation of the Host Requirements, Open Shortest-Path-First IGP (OSPF), and the Point-to-Point Protocol (PPP) working groups. The Host Requirements effort was meant to examine conventional wisdom and codify current practices in the implementation of host protocols. The goal of this fundamentally important work was to define a standard for host interoperability. It would accomplish this good by working together the separate RFCs on host protocols with a finishing veneer of technical developments. The goal of the OSPF and PPP efforts was also open interoperability, but, in this case, between routers. These important efforts have started to pay off with the recent publishing of the Host Requirements RFCs (RFC 1122 and RFC1123), and with the completion of specifications, and testing of implementations, for both OSPF and PPP.

Interoperability is a key word here, and points up a major focus of IETF activity. Following closely on the heels of these efforts will be the creation of a Router Requirements working group. This group will have the goal to codify and refine standards for router interoperability in much the same way as the Host Requirement RFCs.

Although the IETF began with a concentration on TCP/IP, multi-protocol interoperability has also become an important goal. We discussed OSI addressing schemes for the Internet in our very first meeting in January 1986, with a stronger focus at the October 1986 meeting at SRI. In April 1987 at BBN, we had a joint meeting with X3S3.3, the ANSI group dealing with OSI network and transport layer issues. The OSI Interoperability working group had its first meeting at the January 1989 IETF at the University of Texas. It's worth noting that some recent sessions of this single working group have been as widely attended as previous entire IETF meetings.

IETF attendance has hovered just over 100 until the Stanford meeting in July 1989, where it hit 220. This growth in attendees is a graphic illustration of the growth of interest in TCP/IP and networking in the last few years. Regular IETF attendees will be familiar with the reports from BBN and MERIT on the growth in the number of networks, and with reports from SRI on the growth in the number of Internet hosts. These reports, along with reports from other network operations groups have become a regular feature on the IETF plenary agenda. Working group activity has also seen an accompanying steep increase from the four originally formed at the Ames meeting to the current list of over 25.

Formation of Technical Areas and The Steering Group

As both the number of working groups and the number of attendees began to soar, it became clear that additional structure was needed to assist the IETF chair in providing technical and managerial leadership. In coordination with the IAB, we developed the concept of a steering group based on technical areas. The existing working groups were divided into appropriate areas of concentration. A technical director would be chosen to head each area, and the technical directors, with perhaps some at-large members, would compose the Internet Engineering Steering Group (IESG).

The function of the IESG would be to take a critical view of how the Internet may evolve in 3-5 years, and then create and guide the appropriate working groups to satisfy these requirements. New Internet requirements might be based on:

- Technology (e.g., faster local and metropolitan networks, more economic carrier tariffs, faster hosts and routers)
- New Service Requirements
- Continued growth in attached networks
- Agency Programs (e.g., FRICC activities and their impact on the Internet and IETF, particularly FRICC planning for the National Research and Education Network (NREN))

I originally felt that the six areas listed below provided the right alignment:

- Host Services (Transport and application infrastructure)
- Internet Services (IP and Internet infrastructure)
- Routing (Internet routing protocols and routing architecture)
- Network Management (NM protocols, NM applications, NOC services)
- OSI Interoperability (Coexistence of OSI in Internet)
- User Services (NIC services)

At the first IESG meeting we discussed the mission of the IETF and how it is split between protocol development and operations. In conjunction with the IAB, we defined the principal charter of the IETF as:

- Responsible for short- and mid-term evolution of Internet protocol architecture, with explicit role in Internet standards process (i.e., the IAB ratifies Internet standards, but IETF is recognized as a body where Internet standards may be developed or refined.)
- Focus for Internet operational stability
- Technology Transfer from Internet Research Task Force (IRTF)
- Forum for interchange between vendors, users, agency contractors, network managers, and researchers

The six original areas listed above focus heavily on protocol development, and we realized that a fundamental strength of the Internet community was its operational experience. We realized that we needed to maintain close contact with the operational community. Therefore, we decided to add a separate area for Operations.

Further, the host-services area was originally conceived to include Internet applications. However, we recognized a growing need to focus in this area on the underlying infrastructure (e.g., RPC, data representation) for networked applications. Therefore, we formed an entirely separate area to focus purely on the applications themselves. In connection with this, we realized that the proposed User Services area actually contained both protocol-based and operations-based services. We felt the needs of these different types of functions could best be satisfied by splitting these two sides of user services between the new Applications area and the Operations area. However, so as not to let important user services issues languish under a new area without a director, we assigned them (at least temporarily) to the Host-services area. Finally, we formed a Security area in realization that we needed a real commitment to security issues. In this regard, we have opened an interchange with the Computer Emergency Response Team (CERT) at SEI.

The current alignment of technical areas and their directors are given below:

- Host-based Services (Transport and application infrastructure) Craig Partridge, BBN
- Internet-based Services (IP and Internet infrastructure) Noel Chiappa, Consultant/Proteon
- Routing (Internet routing protocols and routing architecture) Robert Hinden, BBN
- Network Management (NM protocols, NM applications, NOC services) David Crocker, DEC
- OSI Interoperability (Coexistance of OSI in Internet) Rob Hagens, University of Wisconsin and Ross Callon, DEC
- Applications (e.g., Email, file transfer, directory services, etc.) TBD
- Security Services (e.g., Authentication, access control, secure configuration management) TBD
- Operations TBD

I am very pleased that the important players above were able to join the IESG in their present roles. The IESG represents a large commitment to the Internet community, and I am both appreciative and somewhat astonished that we were able to recruit such a high level of talent. We will see increased visibility from the IESG in the coming meetings as I begin to delegate interactions with the working groups to the appropriate directors.

Please join me in welcoming this new group to the IETF. In addition to guiding the existing working groups under their individual areas, the IESG is currently looking at the future requirements that will lead to the formation of new working groups. In particular, we want to begin an open planning process for Internet protocol evolution, OSI coexistence and interoperation, and an Internet routing architecture. We will be relying on the important input and continued support of all IETF participants to make it all happen.

IETF Meetings

Date	Location	Highlights
1986		
January	Linkabit, San Diego	(Joint with Dave Mill's TF)
April	Army Ballistics Research Lab, Aberdeen MD	
July	U. of Michigan, Ann Arbor	
October	SRI, Menlo Park	(Joint with Dave Mill's TF, initial OSI discussions)
1987		
February	NASA Ames Research Center, CA	(Initial WGs, Net Mgt)
April	BBN, Boston	(Joint with ANSI X3S3.3)
July	MITRE, McLean VA	(Focus on Net Mgt, slow-start)
November	NCAR, Boulder	(Initial Host Reqs session)
1988		
March	San Diego Supercomputer Center	(Initial OSPF and PPP WGs)
June	U.S. Naval Academy, Annapolis	
October	U. of Michigan	(Showcase new NSFnet backbone)
1989		
January 18	U. of Texas	(OSI Working Group)
April 11	Kennedy Space Center, Cocoa Beach	(NASA Sponsored)
July 24	Stanford University, CA	(Announce Steering Group)
October 31	University of Hawaii	(NASA Sponsored, Focus on Pacific networking)
1990		
February 6	FSU Supercomputer Center, Tallahassee	
May 1	Pittsburgh Supercomputer Center	
July 31	University of British Columbia	(FRICC/NACCIRN tie-in)
November	Princeton and JVNC (tentative)	

WG NAME CHAIR/ORGANIZATION EMAIL ADDRESS TELEPHONE NUMBER	MAILING LIST ADDRESS MAILING LIST REQUEST ADDRESS INTERNET DRAFT* OR 1989 RFC
Alert Management Louis Steinberg/IBM louiss@ibm.com 203-783-7175	alert-man@merit.edu alert-man-request@merit.edu <draft-ietf-alertman-asyncalertman-00.txt> "Managing Asynchronously Generated Alerts," September 1989
Authentication Jeff Schiller/MIT jis@athena.mit.edu 617-253-4101	awg@bitsy.mit.edu <draft-ietf-auth-ipauthoption-00.txt> "The Authentication of Internet Datagrams", August 1989
CMIP-over-TCP (CMOT) Lee LaBarre/Mitre cel@mitre-bedford.arpa 617-271-8507	netman@gateway.mitre.org RFC1095: Common Management Information Services and Protocol over TCP/IP (CMOT), April 1989
Domain Name System (DNS) Paul Mockapetris/USC-ISI pvm@isi.edu 213-822-1511	namedroppers@sri-nic.arpa RFC1101: DNS Encoding of Network Names and Other Types, April 1989
Dynamic Host Configuration Ralph Droms/Bucknell (NRI) droms@nri.reston.va.us 703-620-8990	host-conf@rutgers.edu
Host Requirements Bob Braden/ISI braden@isi.edu 213-822-1511	ietf-hosts@nnsc.nsf.net <draft-ietf-hostreq-hrll-00.txt> "Requirements for Internet Hosts -- Communication Layers; <draft-ietf-hostreq-hrul-00.txt> "Requirements for Internet Hosts -- Application Layer" June 1989
Interconnectivity Guy Almes/Rice almes@rice.edu 713-527-6038	iwg@rice.edu RFC1105: Border Gateway Protocol (BGP) June 1989
Internet User Population (IUP) Craig Partridge/BBN craig@nnsc.nsf.net 617-873-2459	ietf@venera.isi.edu(temporary)
Joint NSFnet Regional Monitoring (JOMANN) Susan Hares/Merit skh@merit.edu 313-936-2095	njm@merit.edu njm-interest@merit.edu njm-request@merit.edu
LAN Manager	lanmanwg@spam.istc.sri.com Draft RFCs: LANMAN-MIB and LANMAN-MIB-EXPER
Network Information Services Infrastructure (NISI) Karen Bowers/NRI kbowers@nri.reston.va.us 703-620-8990 Phill Gross/NRI pgross@nri.reston.va.us 703-620-8990	nisi@merit.edu
NOC-Tools Bob Enger/Contel enger@sccgate.scc.com 301-840-4040 Bob Stine/Sparta stine@sparta.com 703-448-0210	noctools@merit.edu
Open SPF-Based IGP (OSPF) Mike Petry/UMD petry@trantor.umd.edu 301-454-2943 John Moy/Proteon jmoy@proteon.com 508-898-2800	ospfigp@trantor.umd.edu <draft-ietf-ospfigpwg-spec-07.ps> "Draft OSPF Specification", July 1989
Open Systems Routing Marianne Lepp/BBN mlepp@bbn.com 617-873-2458	open-rout-interest@bbn.com

OSI Interoperability Ross Callon/DEC callon@erlang.dec.com 508-486-5009 Rob Hagens/UWisc hagens@cs.wisc.edu 608-262-1017	ietf-osi@cs.wisc.edu ietf-osi-request@cs.wisc.edu
PDN Routing Group CH Rokitansky/Fern Uni-Hagen roki@isi.edu or roki@dhafeu52.bitnet roki%dhafeu52.bitnet@cunyvm.cuny.edu	pdn-wg@bbn.com pdn-request@bbn.com pdn-interest@bbn.com <draft-ietf-pdn-clusterscheme-00.txt> "Internet Cluster Addressing Scheme", August 1989 <draft-ietf-pdn-pdncluster-00.txt> "Application of the Cluster Addressing Scheme to X.25 Public Data Networks and Worldwide Internet Network Reachability Information Exchange", August 1989 <draft-ietf-pdn-pdnclusternet assignm-00.txt>, "Assignment / Reservation of Internet Network Numbers for the PDN-Cluster", August 1989
Performance and Congestion Control Allison Mankin/Mitre mankin@gateway.mitre.org 703-883-7907	ietf-perf@gateway.mitre.org <draft-ietf-perfcc-gwcc-00.txt> "Gateway Congestion Control Policy" July 89
Point-to-Point Protocol Drew Perkins/CMU ddp@andrew.cmu.edu 412-268-8576 Russ Hobby/UC Davis rdhobby@ucdavis.edu 916-752-0236	ietf-ppp@ucdavis.edu ietf-ppp-request@ucdavis.edu <draft-ietf-ppp-req-00.txt>, "Requirements for an Internet Standard Point-to-Point Protocol", <draft-ietf-ppp-ipdatagramstx-01.txt> "The Point-to-Point Protocol (PPP), A Proposed Standard for the Transmission of IP Datagrams over Point-to-Point Links" June 1989
ST-Connection IP Claudio Topolcic/BBN topolcic@bbn.com 617-873-3874	cip@bbn.com
TELNET Dave Borman/Cray dab@cray.com 612-681-3398	telnet-ietf@cray.com RFC 1116: "Telnet Linemode Option" August 1989
User Documents Karen Roubicek/BBN (NNSC) roubicek@nnsc.nsf.net 617-873-3361 Tracy LaQuey/U of Texas tracy@emx.utexas.edu 512-471-3241	user-doc@nnsc.nsf.net user-doc-request@nnsc.nsf.net
User Services Karen Bowers/NRI bowers@sccgate.scc.com or kbowers@nri.reston.va.us 703-620-8990	us-wg@nnsc.nsf.net us-wg-request@nnsc.nsf.net

Other Activities:

Independent Submission Steve Kille/UCL s.kille@cs.ucl.ac.uk	<draft-ucl-kille-x400RFC822-01.txt>, "Mapping between X.400 (1988) and RFC 822", June 1989
Independent Submission Steve Kille/UCL s.kille@cs.ucl.ac.uk	<draft-ucl-kille-uucpmapping-00.txt>, "Mapping between Full RFC 822 and RFC 822 with Restricted Encoding", July 26, 1989

*These individual Internet-Drafts are stored on-line in the directory "Internet-Drafts:" at NIC.DDN.MIL.

II. Internet Activities Board

THE INTERNET ACTIVITIES BOARD

1. Introduction

In 1968, the U.S. Defense Advanced Research Projects Agency (DARPA) initiated an effort to develop a technology which is now known as packet switching. This technology had its roots in message switching methods, but was strongly influenced by the development of low-cost minicomputers and digital telecommunications techniques during the mid-1960's [BARAN 64, ROBERTS 70, HEART 70, ROBERTS 78]. A very useful survey of this technology can be found in [IEEE 78].

During the early 1970's, DARPA initiated a number of programs to explore the use of packet switching methods in alternative media including mobile radio, satellite and cable [IEEE 78]. Concurrently, Xerox Palo Alto Research Center (PARC) began an exploration of packet switching on coaxial cable which ultimately led to the development of Ethernet local area networks [METCALFE 76].

The successful implementation of packet radio and packet satellite technology raised the question of interconnecting ARPANET with other types of packet nets. A possible solution to this problem was proposed by Cerf and Kahn [CERF 74] in the form of an internetwork protocol and a set of gateways to connect the different networks. This solution was further developed as part of a research program in internetting sponsored by DARPA and resulted in a collection of computer communications protocols based on the original Transmission Control Protocol (TCP) and its lower level counterpart, Internet Protocol (IP). Together, these protocols, along with many others developed during the course of the research, are referred to as the TCP/IP Protocol Suite [RFC 1100, LEINER 85, POSTEL 85, CERF 82, CLARK 86].

In the early stages of the Internet research program, only a few researchers worked to develop and test versions of the internet protocols. Over time, the size of this activity increased until, in 1979, it was necessary to form an informal committee to guide the technical evolution of the protocol suite. This group was called the Internet Configuration Control Board (ICCB) and was established by Dr. Vinton Cerf who was then the DARPA program manager for the effort. Dr. David C. Clark of the Lab for Computer Science at Massachusetts Institute of Technology was named the chairman of this committee.

In January, 1983, the Defense Communications Agency, then responsible for the operation of the ARPANET, declared the TCP/IP protocol suite to be standard for the ARPANET and all systems on the network converted from the earlier Network Control Program (NCP) to TCP/IP. Late that year, the ICCB was reorganized by Dr. Barry Leiner, Cerf's successor at DARPA, around a series of task forces considering different technical aspects of internetting. The re-organized group was named the Internet Activities Board.

As the Internet expanded, it drew support from U.S. Government organizations including DARPA, the National Science Foundation (NSF), the Department of Energy (DOE) and the National Aeronautics and Space Administration (NASA). Key managers in these organizations, responsible for computer networking research and development, formed an informal Federal Research Internet Coordinating Committee (FRICC) to coordinate U.S. Government support for and development and use of the Internet system. The FRICC sponsors most of the U.S. research on internetting, including support for the Internet Activities Board and its subsidiary organizations.

At the international level, a Coordinating Committee for Intercontinental Research Networks (CCIRN) has been formed which includes the U.S. FRICC and its counterparts in North America and Europe. The CCIRN provides a forum for cooperative planning among the principal North American and European research networking bodies.

2. Internet Activities Board

The Internet Activities Board (IAB) is the coordinating committee for Internet design, engineering and management. The Internet is a collection of over a thousand of packet switched networks located principally in the U.S., but also includes systems in many other parts of the world, all interlinked and operating using the protocols of the TCP/IP protocol suite. The IAB is an independent committee of researchers and professionals with a technical interest in the health and evolution of the Internet system. Membership changes with time to adjust to the current realities of the research interests of the participants, the needs of the Internet system and the concerns of the U.S. Government, university and industrial sponsors of the elements of the Internet.

IAB members are deeply committed to making the Internet function effectively and evolve to meet a large scale, high speed future. All IAB members are required to have at least one other major role in the Internet community in addition to their IAB membership. New members are appointed by the chairman of the IAB, with the advice and consent of the remaining members. The chairman serves a term of two years. The IAB focuses on the TCP/IP protocol suite, and extensions to the Internet system to support multiple protocol suites.

The IAB has two principal subsidiary task forces:

- Internet Engineering Task Force (IETF)
- Internet Research Task Force (IRTF)

Each of these Task Forces is led by a chairman and guided by a Steering Group which reports to the IAB through its chairman. Each task force is organized, by the chairman, as required, to carry out its charter. For the most part, a collection of Working Groups carries out the work program of each Task Force.

All decisions of the IAB are made public. The principal vehicle by which IAB decisions are propagated to the parties interested in the Internet and its TCP/IP protocol suite is the Request for Comment (RFC) note series. The archival RFC series was initiated in 1969 by Dr. Stephen D. Crocker as a means of documenting the development of the original ARPANET protocol suite [RFC 1000]. The editor-in-chief of this series, Dr. Jonathan B. Postel, has maintained the quality of and managed the archiving of this series since its inception. A small proportion of the RFCs document Internet standards. Most of them are intended to stimulate comment and discussion. The small number which document standards are especially marked in a "status" section to indicate the special status of the document. An RFC summarizing the status of all standard RFCs is published regularly [RFC 1100].

RFCs describing experimental protocols, along with other submissions whose intent is merely to inform, are typically submitted directly to the RFC editor. A Standard RFC starts out as a Proposed Standard and may be promoted to Draft Standard and finally Standard after suitable review, comment, implementation and testing.

Prior to publication of a Proposed Standard, Draft Standard or Standard RFC, it is made available for comment through an on-line Internet-Draft directory. Typically, these Internet-Drafts are working documents of the IAB or of the working groups of the Internet Engineering and Research Task Forces. Internet-Drafts are either submitted to the RFC editor for publication or discarded within 3-6 months.

The IAB performs the following functions:

- (a) Sets Internet Standards
- (b) Manages the RFC publication process
- (c) Reviews the operation of the IETF and IRTF
- (d) Performs strategic planning for the Internet, identifying long-range problems and opportunities
- (e) Acts as a technical policy liaison and representative for the Internet community, and
- (f) Resolves technical issues which cannot be treated within the IETF or IRTF frameworks.

To supplement its work via electronic mail, the IAB meets quarterly to review the condition of the Internet, to review and approve proposed changes or additions to the TCP/IP suite of protocols, to set technical development priorities, to discuss policy matters which may need the attention of the Internet sponsors, and to agree on the addition or retirement of IAB members and on the addition or retirement of task forces reporting to the IAB. Typically, two of the quarterly meetings are by means of video teleconferencing (provided, when possible, through the experimental Internet packet video-conferencing system).

The IAB membership is currently as follows:

Vinton Cerf	Chairman
David Clark	IRTF Chairman
Phillip Gross	IETF Chairman
Jonathan Postel	RFC Editor
Robert Braden	Executive Director
Hans-Werner Braun	Member
Barry Leiner	Member
Daniel Lynch	Member
Stephen Kent	Member

3. The Internet Engineering Task Force

The Internet has grown to encompass a large number of widely geographically dispersed networks in academic and research communities. It now provides an infrastructure for a broad community with various interests. Moreover, the family of Internet protocols and system components has moved from experimental to commercial development. To help coordinate the operation, management and evolution of the Internet, the IAB established the Internet Engineering Task Force (IETF).

The IETF is chaired by Mr. Phillip Gross and managed by its Internet Engineering Steering Group (IESG). The IAB has delegated to the IESG the general responsibility for making the Internet work and for the resolution of all short- and mid-range protocol and architectural issues required to make the Internet function effectively.

The charter of the IETF includes:

- (a) Responsibility for specifying the short and mid-term Internet protocols and architecture and recommending standards for IAB approval.
- (b) Provision of a forum for the exchange of information within the Internet community.
- (c) Identification of pressing and relevant short- to mid-range operational and technical problem areas and convening of Working Groups to explore solutions.

The Internet Engineering Task Force is a large open community of network designers, operators, vendors, and researchers concerned with the Internet and the Internet protocol suite. It is organized around a set of eight technical areas, each managed by a technical area director. In addition to the IETF Chairman, the area directors make up the IESG membership. Each area director has primary responsibility for one area of Internet engineering activity, and hence for a subset of the IETF Working Groups. The area directors have jobs of critical importance and difficulty and are selected not only for their technical expertise but also for their managerial skills and judgment. At present, the eight technical areas and chairs are:

Applications	TBD
Host-Based	Craig Partridge
Internet Services	Noel Chiappa
Routing	Robert Hinden
Network Management	David Crocker
OSI Coexistence	Ross Callon and Robert Hagens
Operations	TBD
Security	TBD

The work of the IETF is performed by subcommittees known as Working Groups. There are currently more than 20 of these. Working Groups tend to have a narrow focus and a lifetime bounded by completion of a specific task, although there are exceptions. The IETF is a major source of proposed protocol standards, for final approval by the IAB.

The IETF meets quarterly and extensive minutes of the plenary proceedings as well as reports from each of the working groups are issued by the IAB Secretariat, at the Corporation for National Research Initiatives.

4. The Internet Research Task Force

To promote research in networking and the development of new technology, the IAB established the Internet Research Task Force (IRTF).

In the area of network protocols, the distinction between research and engineering is not always clear, so there will sometimes be overlap between activities of the IETF and the IRTF. There is, in fact, considerable overlap in membership between the two groups. This overlap is regarded as vital for cross-fertilization and technology transfer. In general, the distinction between research and engineering is one of viewpoint and sometimes (but not always) time-frame. The IRTF is generally more concerned with understanding than with products or standard protocols, although specific experimental protocols may have to be developed, implemented and tested in order to gain understanding.

The IRTF is a community of network researchers, generally with an Internet focus. The work of the IRTF is governed by its Internet Research Steering Group (IRSG). The chairman of the IRTF and IRSG is David Clark. The IRTF is organized into a number of Research Groups (RGs) whose chairs of these are appointed by the chairman of the IRSG. The RG chairs and others selected by the IRSG chairman serve on the IRSG.

These groups typically have 10 to 20 members, and each covers a broad area of research, pursuing specific topics, determined at least in part by the interests of the members and by recommendations of the IAB.

The current members of the IRSG are as follows:

David Clark	Chairman
Robert Braden	End-to-End Services
Douglas Comer	Member-at-Large
Deborah Estrin	Autonomous Networks
Stephen Kent	Privacy and Security
Keith Lantz	User Interfaces
David Mills	Member-at-Large

5. The Near-term Agenda of the IAB

There are seven principal foci of IAB attention for the period 1989 - 1990:

- Operational Stability
- User Services
- OSI Coexistence
- Testbed Facilities
- Security
- Getting Big
- Getting Fast

Operational stability of the Internet is a critical concern for all of its users. Better tools are needed for gathering operational data, to assist in fault isolation at all levels and to analyze the performance of the system. Opportunities abound for increased cooperation among the operators of the various Internet components [RFC 1109]. Specific, known problems should be dealt with, such as implementation deficiencies in some versions of the BIND domain name service resolver software. To the extent that the existing Exterior Gateway Protocol (EGP) is only able to support limited topologies, constraints on topological linkages and allowed transit paths should be enforced until a more general Inter-Autonomous System routing protocol can be specified. Flexiblity for Internet implementation would be enhanced by the adoption of a common internal gateway routing protocol by all vendors of internet routers. A major effort is recommended to achieve conformance to the Host Requirements RFCs which are to be published early in the fourth quarter of calendar 1989.

Among the most needed user services, the White Pages (electronic mailbox directory service) seems the most pressing. Efforts should be focused on widespread deployment of these capabilities in the Internet by mid-1990. The IAB recommends that existing white pages facilities and newer ones, such as X.500, be populated with up-to-date user information and made accessible to Internet users and users of other systems (e.g. commercial email carriers) linked to the Internet. Connectivity with commercial electronic mail carriers should be vigorously pursued, as well as links to other network research communities in Europe and the rest of the world.

Development and deployment of privacy-enhanced electronic mail software should be accelerated in 1990 after release of public domain software implementing the private electronic mail standards [RFC 1113, RFC 1114 and RFC 1115]. Finally, support for new or enhanced applications such as computer-based conferencing, multi-media messaging and collaboration support systems should be developed.

The National Network Testbed (NNT) resources planned by the FRICC should be applied to support conferencing and collaboration protocol development and application experiments and to support multi-vendor router interoperability testing (e.g., interior and exterior routing, network management, multi-protocol routing and forwarding).

With respect to growth in the Internet, architectural attention should be focused on scaling the system to hundreds of millions of users and hundreds of thousands of networks. The naming, addressing, routing and navigation problems occasioned by such growth should be analyzed. Similarly, research should be carried out on analyzing the limits to the existing Internet architecture, including the ability of the present protocol suite to cope with speeds in the gigabit range and latencies varying from microseconds to seconds in duration.

The Internet should be positioned to support the use of OSI protocols by the end of 1990 or sooner, if possible. Provision for multi-protocol routing and forwarding among diverse vendor routes is one important goal. Introduction of X.400 electronic mail services and interoperation with RFC 822/SMTP [RFC822, RFC821, RFC987] should be targeted for 1990 as well. These efforts will need to work in conjunction with the White Pages services mentioned above. The IETF, in particular, should establish liaison with various OSI working groups (e.g., at NIST, RARE, Network Management Forum) to coordinate planning for OSI introduction into the Internet and to facilitate registration of information pertinent to the Internet with the various authorities responsible for OSI standards in the United States.

Finally, with respect to security, a concerted effort should be made to develop guidance and documentation for Internet host managers concerning configuration management, known security problems (and their solutions) and software and technologies available to provide enhanced security and privacy to the users of the Internet.

REFERENCES

BARAN 64
P. Baran, et al, "On Distributed Communications," Volumes I-XI, RAND Corporation Research Documents, August 1964.

CERF 74
V. G. Cerf and R. E. Kahn, "A Protocol for Packet Network Interconnection," IEEE Trans. on Communications, Vol. COM-22, No. 5, pp. 637-648, May 1974.

CERF 82
V. Cerf and E. Cain, "The DoD Internet Architecture Model," Proceedings of the SHAPE Technology Center Symposium on Interoperability of Automated Data Systems, November 1982. Also in Computer Networks and ISDN, Vol. 17, No. 5, October 1983.

CLARK 86
D. D. Clark, "The Design Philosophy of the DARPA Internet Protocols," Proceedings of the SIGCOMM '88 Symposium, Computer Communications Review, Vol. 18, No. 4, August 1988, pp. 106-114.

HEART 70
F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther and D. C. Walden, "The interface message processor for the ARPA computer network", AFIPS Conf. Proc. 36, pp. 551-567, June 1970.

IEEE 78
R. E. Kahn (Guest Editor), Keith W. Uncapher and Harry L. Van Trees (Associate Guest Editors), Proceedings of the IEEE, Special issue on Packet Communication Networks, Volume 66, No. 11, November 1978, pp. 1303-1576.

IEEE 87
Mobile Radio Special Issue of Proceedings, January 1987.

LEINER 85
B. M. Leiner, R. Coles, J. B. Postel, D. Mills, "The DARPA Internet Protocol Suite," IEEE Communications, Vol. 23, No. 3, March 1985, pp. 29-34.

METCALFE 76
R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet for local computer networks," Communications of the ACM, Vol. 19, No. 7, pp. 395-404, July 1976.

POSTEL 85
J. B. Postel, "Internetwork Applications Using the DARPA Protocol Suite," IEEE INFOCOM 85, Washington, D.C., March 1985.

RFC 821
J. B. Postel, "Simple Mail Transfer Protocol," RFC 821, USC/Information Sciences Institute, August 1982.

RFC 822
D. Crocker, "Standard for the Format of ARPA Internet Text Messages," RFC 822, University of Delaware, August 1982.

RFC 987
S. E. Kille, "Mapping between X.400 and RFC 822," RFC 987, University College London, June 1986.

RFC 1000
J. B. Postel and J. Reynolds, "The Request for Comments References Guide," USC/Information Sciences Institute, RFC 1000, August 1987.

RFC 1026
S. E. Kille, "Addendum to RFC 987: (Mapping between X.400 and RFC 822)," RFC 1026, September 1987.

RFC 1100
J. B. Postel (ed.), "IAB Official Protocol Standards," Request for Comment 1100, April 1989.

RFC 1109
V. Cerf, "Report of the Second Ad Hoc Network Management Review Group," RFC 1109, NRI, August 1989.

RFC 1113
"J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I – Message Encipherment and Authentication Protocol," RFC 1113, IAB Privacy Task Force, August 1989.

RFC 1114
S. T. Kent and J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part II – Certificate-based Key Management," RFC 1114, IAB Privacy Task Force, August 1989.

RFC 1115
J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part III – Algorithms, Modes and Identifiers," RFC 1115, IAB Privacy Task Force, August 1989.

ROBERTS 70
L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," Proc. SJCC 1970, pp. 543-549.

ROBERTS 78
L. G. Roberts, "Evolution of Packet Switching," Proc. IEEE, Vol. 66, No. 11, November 1978, pp. 1307-1313.

Note: RFCs are available from SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025 or on-line via anonymous file transfer from NIC.DDN.MIL.

III. IETF Attendees

July 1989–IETF Meeting Attendee List

Alyson Abramowitz

Hewlett-Packard
19420 Homestead Road
Cupertino, CA 95014
408-447-3829
ala@hpindda.hp.com

Ken Adelman

TGV, Inc.
15139 Old Ranch Road
Los Gatos, CA 95030
408-353-3939
adelman@tgv.com

Charles Alberts

Banyan Systems
115 Flanders Road
Westborough, MA 01581
508-898-1000
charlie@banyan.banyan.com

Guy Almes

Rice University
PO Box 1892
Dept of Computer Science
Houston, TX 77251-1892
713-527-6038
almes@rice.edu

Philip Almquist

Consultant
214 Cole Street
San Francisco, CA 94117
415-752-2427
almquist@jessica.stanford.edu

Karen Armstrong

San Diego SCC
PO Box 85608
CERFNet
San Diego, CA 92138-5608
619-534-5077
armstrong@sds.sdsc.edu

Karl Auerbach

Epilogue Technology Corp.
PO Box 217
Capitola, CA 95010-0217
408-426-8786
auerbach@csl.sri.com, karl@asylum.sf.ca.us

Paul Baer

Stanford University
Networking Systems
Pine Hall Rm. 115
Stanford, CA 94305
415-723-7520
baer@jessica.stanford.edu

Gerald Bailey

National Security Agency T4
9800 Savage Rd.
Ft. George G. Meade, MD 20755-6000
301-688-5328
c/o dpowles@bbn.com

Fred Baker

Vitalink
6607 Kaiser Drive
Fremont, CA 94555
415-794-1100
baker@vitalink.com

Trudy Barker

SRI International
Network Info Systems Ctr. Rm EJ221
333 Ravenswood Ave.
Menlo Park, CA 94025
415-859-5240
trudy@sri-nic.arpa

Bob Beach

Ultra Network Technologies
101 Daggett Drive
San Jose, CA 95131
408-972-0100
bob@ultra.com

Neal Bierbaum

Vitalink
6607 Kaiser Drive
Fremont, CA 945555
415-795-1100
bierbaum@vitalink.com, vitam6!bierbaum@vitam6

Craig Blackwood

Hewlett-Packard
8000 Foothills Blvd.
R3NF1
Roseville, CA 95678
916-785-5574
craig@hprnd.rose.hp.com

Steve Blumenthal

BBN STC
10 Moulton St.
Cambridge, MA 02238
617-873-3197
blumenthal@bbn.com

Rick Boivie

IBM
472 Wheelers Farms Road
Milford, CT 06460
203-783-7284
rboivie@ibm.com

Dave Borman

Cray Research
1440 Northland Dr.
Mendota Heights, MN 55120
612-681-3398
dab%oliver.cray.com@uc.msc.umn.edu, dab@cray.com

Leonard Bosack

cisco Systems
1360 Willow Road
Menlo Park, CA 94025
415-326-1941
bosack@mathom.cisco.com

William Bostwick

FRICC
US Department of Energy
Washington, DC
(301) 353-5800
bostwick@ccc.mfenet.gov

Karen Bowers

Corp. for National Research Initiatives
1895 Preston White Drive
Suite 100
Reston, VA 22091
703-620-8990
kbowers@nri.reston.va.us

Billy Brackenridge

Alembic Systems
23801 Upper Ramirez Motorway
Malibu, CA 90265
213-457-2026
brackenridge@isi.edu

Robert Braden

Uuniversity of Southern California/ISI
4676 Admiralty Way
Marina del Rey, CA 90292
213-822-1511
braden@isi.edu

Laura Breeden

BBN Communications
10 Moulton Street
Cambridge, MA 02138
617-873-2649
breeden@sh.cs.net

Lee Breslau

University of Southern California
Computer Science Dept.
Henry Salvatori CS Dept.
Los Angeles, CA 90089-0782
213-743-0836
breslau@jerico.usc.edu

Scott Brim

Cornell University
265 Olin Hall
Cornell Theory Center
Ithaca, NY 14853
607-255-8686
swb@devvax.tn.cornell.edu

Charles E. Brooks

Ford Aerospace
220 Henry Ford II Drive
p.o. Box 49041
San Jose, CA 95161-9041
(408) 473-4535
ceb@wdl1.fac.ford.com

Thomas Eric Brunner

Tule Network Services
801 Church St.
Suite 1327
Mountain View, CA 94041
415-855-4486

Jeffrey Burgan

NASA-Ames Research Center
M/S 233-18
Moffet Field, CA 94035
415-694-6440
jeff@nsipo.nasa.gov

John Burnett

Hewlett-Packard
19420 Homestead Road
MS-43UC
Cupertino, CA 95014
408-447-2326
ucbvax!hpda!jlb

Ross Callon

Digital Equipment Corporation
550 King Street
LKG1-2/A19
Littleton, MA 01460-1289
508-486-5009
callon@erlang.dec.com, callon@bigfut.dec.com

Glen Carter

Defense Communication Agency
US Air Force
HQ DCA Code B615
Washington, DC 20305-2000
703-285-5481
gcarter@ddn1.dca.mil

Stephen Casner

University of Southern California
4676 Admiralty Way
ISI
Marina del Rey, CA 90292
213-822-1511
casner@isi.edu

A. Lyman Chapin

Data General
4400 Computer Drive
M/S D-112
Westborough, MA 01580
508-870-6056
lyman-chapin@dgc.mceo.dgcom, lyman-chapin@dgc.mceo.dgcom

Mike Chepponis

Apple
211 Kansas Way
Fremont, CA 94539
415-438-6459
k3mc@apple.com

Debra Chi

Sun Microsystems
3147 Allen Way
Santa Clara, CA 95051
415-336-2938
debee@sun.com

Joseph Choy

NCAR
PO Box 3000
Boulder, CO 80307-3000
U S
303-497-1222
choy@ncar.ucar.edu

David Clark

MIT LCS
545 Technology Square
LCS
Cambridge, MA 02139-1986
617-253-6003
ddc@lcs.mit.edu

Danny Cohen

University of Southern California
4676 Admiralty Way #1000
ISI
Marina del Rey, CA 90292
213-822-1511
cohen@isi.edu

George Cohn

Ungermann-Bass
3900 Freedom Circle
Santa Clara, CA 95052
408-562-7676
geo@ub.com

Richard Colella

Nat. Inst. of Standards & Technology
Building 225
Room B217
Gaithersburg, MD 20899
301-975-3627
colella@osi3.ncsl.nist.gov

Michael Collins

Lawrence Livermore National Lab
PO Box 5509
M/S 561
Livermore, CA 94551
415-422-4017
collins@ccc.mfecc.arpa

Rob Coltun

University of Maryland
Computer Science Dept.
College Park, MD 20742-2411
301-454-2946
rcoltun@trantor.umd.edu

John Cook

Chipcom Corp.
195 Bear Hill Rd.
Waltham, MA 02154
617-890-6844
cook@chipcom.com

Dave Crocker

Digital Equipment Corporation
675 Spruce Drive
Sunnyvale, CA 94086
408-246-8253
dcrocker@ahwahnee.stanford.edu

Farokh Deboo

3Com
2081 North Shoreline Blvd.
Mountain View, CA 94043
415-940-7677
fjd@bridge2.esd.3com.com

Steve Deering

Stanford University
Computer Science Department
Stanford, CA 94305
415-723-9427
deering@pescadero.stanford.edu

Barbara Denny

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
415-859-2812
denny@sri.com

Way-Chi Doo

3Com
2081 North Shoreline Blvd
Mountain View, CA 94043
415-940-7690
wcd@bridge2.esd.3com.com

Ralph Droms

Corp. for Natl Research Initiatives
1895 Preston White Drive
Suite 100
Reston, VA 22091
717-524-1145
rdroms@nri.reston.va.us, droms@cs.purdue.edu, droms@sol.bucknell.edu

Tom Easterday

Ohio State University
1971 Neil Avenue
Room 406
Columbus, OH 43210
614-292-4843
tom@niscuiircc.ohio-state.edu

David Edwards

Cisco
503 Pennsylvania Avenue
San Francisco, CA
(415) 647-4107
dle@cisco.com

Robert Enger

Contel
1300 Quince Orchard Blvd.
Gaithersburg, MD 20878-4199
301-840-4040
enger@sccgate.scc.com

Deborah Estrin

University of Southern California
Computer Science Dept. mc0782
234 University Park
Los Angeles, CA 90089-0782
213-743-7842
estrin@usc.edu

Erik Fair

Apple Computer
20740 Valley Green Drive
M/S 32-E
Cupertino, CA 95014
408-974-1779
fair@apple.com

Dino Farinacci

3Com
2081 N. Shoreline Blvd.
Mountain View, CA 94043
415-940-7661
dino@bridge2.3com.com

Mark Fedor

NYSERNet Inc
Rensselear Technology Park
165 Jordan Road
Troy, NY 12180
518-283-8860
fedor@nisc.nyser.net

Thomas Ferrin

University of California
Computer Graphics Laboratory
School of Pharmacy
San Francisco, CA 94143-0446
415-476-1100
tef@cgl.ucsf.edu

Bob Fink

Lawrence Berkeley Laboratory
University of California
Bldg 50B, Room 2258
Berkeley, CA 94720
415-486-5692
rlfink@lbl.gov

Dale Finkelson

University of Nebraska-Lincoln
29 Walter Scott Engineering Center
MIDnet
Lincoln, NE 68588-0534
402-472-5032
dmf@westie.unl.edu

James Forster

cisco Systems, Inc.
1350 Willow Road
Menlo Park, CA 94025
415-326-1941
forster@cisco.com

Craig Fox

Network Systems Corp
7600 Boone Avenue North
Brooklyn Park, MN 55428
612-424-4888
foxcj@nsco.network.com

Richard Fox

Tandem Computers
950 Linden Ave., #208
Sunnyvale, CA 94086
408-865-4420
rfox@tandem.com

Vincent Fuller

Stanford University
Networking Systems
115 Pine Hall
Stanford, CA 94305
415-723-6860
fuller@jessica.stanford.edu

James Galvin

Trusted Information Systems, Inc.
3060 Washington Road
Glenwood, MD 21738
301-854-6889
galvin@tis.com

Jose Garcia-Luna

SRI International
Telecommunications Sciences Ct
333 Ravenswood Avenue
Menlo Park, CA 94025
415-859-5647
garcia@sri.com

Vicki Gary Ralls

Interlink
47370 Fremont Blvd.
Fremont, CA 94538
415-657-9800
iruucp!sun!ntrlink!ralls

Elise Gerich

Merit/NSFNet
University of Michigan
1075 Beal Avenue
Ann Arbor, MI 48109-2112
1-800-666-3748
epg@merit.edu

Chuck Gerlach

AT&T Bell Laboratories
1100 East Warrenville Rd
IW 1F305
Naperville, IL 60566-7044
312-979-7325
cag@iwlcs.att.com

Robert E. Gilligan

Sun Microsystems
2550 Garcia Ave.
Mountain View, CA 94043
415-336-1012
gilligan@sun.com

Geoff Goodfellow

Anterior Technology
PO Box 1206
Menlo Park, CA 94026-1206
415-328-5615
geoff@fernwood.mpk.ca.us

Phill Gross

Corp. for Natl. Research Initiatives
1895 Preston White Drive
Suite 100
Reston, VA 22091
703-620-8990
pgross@nri.reston.va.us

Martin Gross

DCEC
1860 Wiehle Avenue
Code R640
Reston, VA 22090-5500
703-437-2165
martin@protolaba.dca.mil

Stu Grossman

Stanford University
Computer Science Dept.
Bldg 460, Rm 040
Stanford, CA 94305
415-723-1407
grossman@score.stanford.edu

Robert Hagens

University of Wisconsin
Computer Science Dept.
1210 West Dayton Street
Madison, WI 53706
608-262-1017
hagens@cs.wisc.edu

Martyne Hallgren

Cornell Theory Center
265 Olin Hall
Ithaca, NY 14853-5210
607-255-8686
martyne@tcgould.tn.cornell.edu

Gene Hastings

Pittsburgh SCC
5000 Forbes Ave.
Pittsburgh, PA 15213
412-268-4960
hastings@morgul.psc.edu

Charles Hedrick

Rutgers University
Hill Center, POB 879
CCIS
Piscataway, NJ 08854
201-932-3088
hedrick@aramis.rutgers.edu

Robert Hinden

BBN Communications
150 CambridgePark Drive
Cambridge, MA 02140
617-873-3757
hinden@bbn.com

Russell Hobby

University of California
Computing Services
Surge II - Room 1400
Davis, CA 95616
916-752-0236
rdhobby@ucdavis.edu

Greg Hollingsworth

Mitre Corporation
7525 Colshire Drive
M/S 425
McLean, VA 22102
703-883-7739
gregh@gateway.mitre.org

Jeffrey Honig

Cornell University
Theory Center
265 Olin Hall
Ithaca, NY 14853-5201
607-255-8686
jch@tcgould.tn.cornell.edu

Steven Hunter

Lawrence Livermore Natl Lab
PO Box 5509, L-561
NMFECC
Livermore, CA 94550
415-423-2219
hunter@ccc.mfecc.arpa

Tom Hytry

AT&T Bell Labs
1100 E. Warrenville Rd
IW IF313
Naperville, IL 60566
312-979-7313
tlh@iwics.att.com, ...!att!iwics!tlh

David Iannucci

NASA - Ames Research Center
Building 258
M/S 258-5
Moffett Field, CA 94035
415-694-4325
dji@orville.nas.nasa.gov

Ski Ilnicki

Hewlett-Packard
19420 Homestead Road
Cupertino, CA 95014
408-447-2797
hpde!ski@hplabs hpde!ski@hplabs.

Ole-Jørgen Jacobsen

Advanced Computing Environments
480 San Antonio Road, Suite 100
Mountain View, CA 94040
415-941-3399
ole@csli.stanford.edu

Van Jacobson

Lawrence Livermore National Lab
One Cycloctron Road
M/S 46A
Berkeley, CA 94720
415-486-6411
van@helios.ee.lbl.gov

Charles Jerian

Sun Microsystems
14-49 2550 Garcia Ave.
Mountain View, CA 940434
415-336-1201
cpj@sun.com

William Jolitz

TeleMuse
447 61st Street
Oakland, CA 94609
415-420-1406
william@ernie.berkeley.edu

William Jones

NASA
Ames Research Center
233-17
Moffett Field, CA 94035
415-694-6482
jones@nsipo.arc.nasa.gov

Dan Jordt

University of Washington
Academic Computing Services
3737 Brooklyn Avenue NE
Seattle, WA 98105
206-543-7352
danj@blake.acs.washington.edu

Hemant Kanakia

Stanford University
221 Easy Street #7
Mountain View, CA 94043
415-967-3782
kanakia@pescadero.stanford.edu

Michael Karels

University of California
1107 Liberty Street
El Cerrito, CA 94530
415-642-4948
karels@berkeley.edu

Phil Karn

Bell Communications Research
445 South Street
Room 2P-357
Morristown, NJ 07960
201-829-4299
karn@thumper.bellcore.com

David Kashtan

TGV, Inc.
15139 Old Ranch Road
Los Gatos, CA 95030
408-353-3939
kashtan@tgv.com

Dave Katz

Merit Computer Network
1075 Beal Avenue
Ann Arbor, MI 48103
313-763-4898
dave_katz@um.cc.umich.edu

David Kaufman

Proteon, Inc.
Two Technology Dr.
Westborough, MA 01581-5008
508-898-2800
dek@proteon.com

Raman Khanna

Stanford University
Networking Systems
115 Pine Hall
Stanford, CA 94305
415 723-3909
khanna@jessica.stanford.edu

Norm Kincl

Hewlett-Packard
19047 Pruneridge Ave.
M/S 46G
Cupertino, CA 95014
408-447-7148
kincl@iag.hp.com

Mark Knopper

Merit Computer Network
2200 Bonisteel Blvd.
Ann Arbor, MI 48109
313-764-9423
mknopper@merit.edu

Stev Knowles

FTP Software
26 Princess Street
Wakefield, MA 01880-3004
617-246-0900 x270
stev@ftp.com

Mark Koopman

3Com
2081 North Shoreline Blvd.
Mountain View, CA 94043
415-940-7647
mhk@athos.3com.com

Edward Kozel

cisco Systems
1360 Willow Road
Menlo Park, CA 94025
415-688-8292
kozel@mathom.cisco.com

Alan Kullberg

BBN Communications
10 Moulton Street
Cambridge, MA 02138
617-873-3955
akullberg@bbn.com

Tracy La Quey

University of Texas
Computation Center
Austin, TX 78712
512 320-9031
tracy@emx.utexas.edu

Eliot Lear

IntelliGenetics
700 East El Camino Real
Mountain View, CA 94040
415-962-7323
lear@net.bio.net

Young Lee

Stanford University
Networking/Communications Systems
166 Pine Hall
Stanford, CA 94305
415-723-2267
youngl@jessica.stanford.edu

Barry Leiner

RIACS
Ames Research Center
M/S 230-5
Moffett Field, CA 94035
415-694-6362
leiner@riacs.edu

John Lekashman

NASA-Ames Research Center
MS 258-6
Moffett Field, CA 94035
415-694-4359
lekash@orville.nas.nasa.gov

Marianne Lepp

BBN Communications
150 Cambridge Park Drive
Cambridge, MA 02140
617-873-2458
mlepp@bbn.com

Norbert Leser

Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142
617-621-8715
nl@osf.org

Mike Little

SAIC
8619 Westwood Center Dr.
Vienna, VA 22182
703-749-5360
little@saic.com

Mark Lottor

SRI International
333 Ravenswood Ave
M/S EJ282
Menlo Park, CA 94025
415-859-2652
mkl@nic.ddn.mil

Kirk Lougheed

cisco Systems
1350 Willow Rd.
Menlo Park, CA 94025
415-326-1941
lougheed@cisco.com

John LoVerso

Xylogics, Inc.
53 Third Ave.
Burlington, MA 01803
617-272-8140
loverso@xylogics.com

E. Paul Love

San Diego Supercomputer Center
PO Box 85608
San Diego, CA 92138
619-534-5043
loveep@sds.sdsc.edu

Dan Lynch

Advanced Computing Environments
21370 Vai Avenue
Cupertino, CA 95014
415-941-3399
lynch@isi.edu

Charles Lynn

BBN/STC
10 Moulton Street
Cambridge, MA 02238
617-873-3367
clynn@bbn.com

Andy Maas

Stanford University
Networking System
115 Pine Hall
Stanford, CA 94305
415-723-6856
maas@jessica.stanford.edu

Gary Malkin

Proteon
2 Technology Drive
Westborough, MA 01581
508-898-2800
gmalkin@proteon.com

Louis Mamakos

University of Maryland
Computer Science Center - Syst
College Park, MD 20742
301 454-2943
louie@trantor.umd.edu

Allison Mankin

Mitre Corporation
7525 Colshire Drive
M/S W425
McLean, VA 22102
703-883-7907
mankin@gateway.mitre.org

April Marine

SRI International
333 Ravenswood Avenue
DDN NIC Room EJ294
Menlo Park, CA 94025
415-859-5318
april@sri-nic.arpa

Matt Mathis

Pittsburgh SCC
4400 5th Ave.
Pittsburgh, PA 15213
412-268-3319
mathis@fornax.ece.cmu.edu

Tony Mazraani

Washington University
Bryan Hall, Rm.307
Computer & Communications Research Center
St. Louis, MO 63130
314-726-4163
tonym@flora.wust.edu

Keith McCloghrie

Hughes LAN Systems
1225 Charleston Road
Mountain View, CA 94043
415-966-7300
kzm@twg.com

Paul McKenney

SRI International
333 Ravenswood Ave.
EK 360
Menlo Park, CA 94025
415-859-4910
mckenney@sri.com

Milo Medin

NASA-Ames Research Center
1121 San Antonio Road
Palo Alto, CA 943036
415-694-6440
medin@nsipo.nasa.gov

Bill Melohn

Sun Microsystems
2550 Garcia Avenue
Mountain View, CA 94043
415-336-2941
melohn@sun.com

Marjo Mercado

Hewlett-Packard
19420 Homestead Rd.
M/S 43LN
Cupertino, CA 95014
408-447-2826
marjo@hpindlm.hp.com

Donald Merritt

Ballistic Research Laboratory
Attn: AMXBR-SECAD
Aberdeen Proving Gro, MD 21005-5066
301-278-6808
don@brl.mil

Stephen Miller

Mass. Microelectronic Center
75 North Drive
Westborough, MA 01581
508-870-0312
miller@m2c.org

Paul Mockapetris

University of Southern California
4676 Admiralty Way
ISI
Marina del Rey, CA 90292
213-822-1511
pvm@isi.edu

Jeffrey Mogul

Digital Equipment Corporation
Western Research Labs
100 Hamilton Ave.
Palo Alto, CA 94301
415-853-6643
mogul@decwrl.dec.com

Douglas Montgomery

Nat. Inst. of Standards & Technology
Technology Bldg., B-217
Gaithersburg, MD 20899
301-975-3630
dougm@osi3.ncsl.nist.gov

Berlin Moore

PREPnet
530 N. Neville
Pittsburgh, PA 15213
412-268-7870
prepnet@andrew.cmu.edu

Robert Morgan

Stanford University
115 Pine Hall
Stanford, CA 94305
415-723-9711
morgan@jessica.stanford.edu

Donald Morris

NCAR
POB 3000
Scientific Computing Division
Boulder, CO 80307
303-497-1000
morris@ncar.ucar.edu

George Mundy

Trusted Information Systems
13202 Hampshire Court
Woodbridge, VA 22193
703-690-3087
mundy@tis.com

Ronald Natalie

Rutgers University
CCIS PO Box 879
Busch Campus-Hill Center
Piscataway, NJ 08855-0879
201-932-3433
ron@rutgers.edu

Rebecca Nitzan

Lawrence Livermore National Lab
PO Box 5509
M/S 561
Livermore, CA 94550
415-422-9775
nitzan@ccc.mfecc.llnl.gov

Lee Oattes

University of Toronto
Computer Services
4 Bancroft Avenue, Rm. 102
Toronto, ON M5S1A1
CANADA
416-978-5448
oattes@utcs.utoronto.ca

William Ohle

General Services Administration
GSA Info Systems, KBA
18th and F Streets NW
Rm 1234
Washington, DC 20405
202-535-7800
ohle@osi.ncsl.nist.gov

Zbigniew Opalka

BBN Communications
50 Moulton St.
Cambridge, MA 02238
617-873-2888
zopalka@bbn.com

Dave Oran

Digital Equipment Corporation
216 Lakewood Dr.
Bloomington, IN 47401
812-333-8107 617-486-7646

Raylene Pak

McDonnell Douglas Corp.
PO Box 49019
M/S D-23
San Jose, CA 95161-9019
408-922-6594
raylene%tardis.tymnet.com@tymix.tymnet.com

Howard Palmer

Interlink Computer Sciences
47370 Fremont Blvd.
Fremont, CA 93438
415-657-9800
sun!iruucp!ntrlink!palmer

Philippe Park

BBN STC
10 Moulton Street
STC
Cambridge, MA 02238
617-873-2892
ppark@wilma.bbn.com

Craig Partridge

BBN Systems & Technologies
10 Moulton Street
Cambridge, MA 02138
617-873-2459
craig@bbn.com

Gurudatta Parulkar

Washington University
Campus Box 1045
St. Louis, MO 63130
314-889-4621
guru@flora.wustl.edu, parulkar@udel.edu

Edison Paw

3Com
2081 N Shoreline Blvd
Mountain View, CA 94043
415-940-7663
esp@bridge2.esd.3com.com

Francine Perillo

cisco Systems
1360 Willow Road
Menlo Park, CA 94025
415-326-1941
perillo@cisco.com

Drew Perkins

Carnegie Mellon University
4910 Forbes Ave
M/S UCC-130
Pittsburgh, PA 15213
412-268-8576
ddp@andrew.cmu.edu

Michael Petry

University of Maryland
Computer Science Center
Room 3339
College Park, MD 20742
301-454-2943
petry@trantor.umd.edu

Shyam Pillalamarri

Hewlett-Packard
19420 Homestead Rd.
Information Networks Division
M/S 43LN
Cupertino, CA 95014
408-447-2794
hpda!shyam@hplabs.hp.com

Mel Pleasant

Rutgers University
H019 Hill Center
Busch Campus
Piscataway, NJ 08855-0879
201-932-2023
pleasant@rutgers.edu

Rex Pugh

Hewlett-Packard
8000 Foothill Blvd
M/S R3NF3
Roseville, CA 95678
916-785-4471
pugh%hprnd@hplabs.hp.com

Jon Pugh

NMFECC
POB 5509 L-561
Livermore, CA 94550
415-422-4017
pugh@nmfecc.llnl.gov

K.K. Ramakrishnan

Digital Equipment Corporation
LKG 1-2/A19
550 King Street
Littleton, MA 01460-1289
508-486-7267
rama%erlang.dec.com@decwrl.dec.com

Elizabeth Redfield

SRI International
333 Ravenswood
EJ 292
Menlo Park, CA 94025
415-859-6187
red@sri-nic.arpa

Michael Reilly

Digital Equipment Corporation
550 King Street
LKG1-3/L06
Littleton, MA 01460
508-486-5283
reily%atari.nac.dec.com@decwrl.dec.com

Jim Reinstedler

Ungermann-Bass
2560 Mission College Blvd.
Santa Clara, CA 95054
408-562-5636
jimr@ub.com

Yakov Rekhter

IBM
TJ Watson Research Center
Route 134 PO Box 218
Yorktown Heights, NY 10598
914-945-3896
yakov@ibm.com

Joel Replogle

NCSA
605 E Springfield
Champaign, IL 61820
217-244-0636
jr@ncsa.uiuc.edu

Robert Reschly

Ballistic Research Laboratory
ATTN: SLCBR-SF (Reschley)
Aberdeen Proving Gro, MD 21005-5066
301-278-6678
reschly@brl.mil

Joyce K. Reynolds

University of Southern California
4676 Admiralty Wy #1001
ISI
Marina del Rey, CA 90292-6695
213-822-1511
jkrey@venera.isi.edu

Michael Roberts

EDUCOM
1112 16th Street NW
Suite 600
Washington, DC 20036
202-872-4200
xb.mmr@forsythe.stanford.edu

Ronald Roberts

Stanford University
Pine Hall
Rm 175B
Stanford, CA 94305-4122
415-723-3352
roberts@jessica.stanford.edu

Jon Rochlis

MIT
Room E40-311
77 Massachusetts Ave
Cambridge, MA 02139
617-253-4101
jon@athena.mit.edu

John Romkey

Epilogue Technology Corp.
PO Box 217
Capitola, CA 95010-0217
408-426-8786
romkey@asylum.sf.ca.us

Milt Roselinsky

Communication Machinery Corporation
125 Cremona Drive
Santa Barbara, CA 93117
805-562-3194
cmcvax!milt@hub.ucsb.edu

Milt Rosenlinsky

Communication Machinery Corporation
125 Cremona Drive
Santa Barbara, CA 93117

cmcvax!milt@hub.ucsb.edu

Karen Roubicek

BBN Systems & Technologies
10 Moulton St.
NSF Network Service Center
Cambridge, MA 02138
617-873-3361
roubicek@nnsc.nsf.net

Timothy J. Salo

Minnesota Supercomputer Center
1200 Washington Avenue South
Minneapolis, MN 55415
612-626-0347
tjs@msc.umn.edu

Greg Satz

cisco Systems
1360 Willow Road
Suite 201
Menlo Park, CA 94025
415-326-1941
satz@cisco.com

Keely Scanlan

Hewlett-Packard
8000 Foothills Blvd.
Bldg. R3
Roseville, CA 95678
916-785-4472
keely@hprod.rod.hp.com

Jeffrey Schiller

Massachusetts Institute of Technology
1 Amherst Street
Room E40-311
Cambridge, MA 02139
617-253-4101
jis@bitsy.mit.edu

Martin Schoffstall

NYSERNet Inc
Rensselaer Technology Park
165 Jordan Road
Troy, NY 12180-3590
518-283-8860
schoff@nisc.nyser.net

Bruce Schofield

DCEC
1860 Wiehle Avenue
Reston, VA 22090
703-437-2556
schofield@edn-unix.arpa

Scott Shenker

Xerox(PARC)
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304
415-494-4840
shenker.pa@xerox.com

Jim Sheridan

IBM
PO Box 334
Whitmore Lake, MI 48189
313-393-6537
jsherida@ibm.com

Dana Sitzler

Merit Computer Network
1075 Beal Avenue
Ann Arbor, MI 48109-2112
313-936-3000
dds@merit.edu

Greg Skinner

SRI International
333 Ravenswood Ave
Rm EK396
Menlo Park, CA 94025
415-859-5851
gds@spam.istc.sri.com

Keith Sklower

University of California
Evans Hall, Rm. 570
Berkeley, CA 94720
415-642-9587
sklower@okeeffe.berkeley.edu

Charles Slater

Sunday & Associates
1119 Timberpine Court
Sunnyvale, CA 95050
408-244-8864
uunet!ibmsupt!cslater

Tom Smith

Hewlett-Packard
8000 Foothills Blvd.
R3NF3
Roseville, CA 95678
916-785-4735
toms@hprnd.rose.hp.com

Frank Solensky

InterLAN
155 Swanson Rd
Boxborough, MA 01719
508-263-9929
solensky@interlan.interlan.com

Karen Sollins

MIT-LCS
545 Technology Sq.
Cambridge, MA 02139
617-253-6006
sollins@lcs.mit.edu

Michael St. Johns

Defense Communication Agency
Code B612
Washington, DC 20305-2000
703-285-5133
stjohns@beast.ddn.mil

Mary K. Stahl

SRI International
Net Info Sys Ctr
333 Ravenswood Ave. Rm EJ 296
Menlo Park, CA 94025
415-859-4775
stahl@sri-nic.arpa

Martha Steenstrup

BBN Communications
150 CambridgePark Dr.
Cambridge, MA 02140
617-873-3192
msteenst@bbn.com

Einar Stefferud

Network Management Associates
17301 Drey Lane
Huntington Beach, CA 92647-5615
714-842-3711
stef@nrtc.northrop.com

Louis Steinberg

IBM
472 Wheelers Farms Rd
M/S 91
Milford, CT 06460
203-783-7175
louiss@ibm.com

Robert Stine

SPARTA, Inc
7926 Jones Branch Dr.
Suite 1070
McLean, VA 22102
703-448-0210
stine@sparta.com

Geof Stone

Network Systems Corporation
7600 Boone Avenue North
Brooklyn Park, MN 55428
612-424-1693
geof@network.com

Roxanne Streeter

NASA Ames Research Center/Sterling Software
M/S 233-18
Moffett Field, CA 94035
415-694-4845
streeter@nsipo.arc.nasa.gov

Zaw-Sing Su

SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
415-859-4576
zsu@tsca.istc.sri.com

James Sweeton

Merit Computer Network
1075 Beal Street
Ann Arbor, MI 48109-2112
313-936-3000
jcs@merit.edu

Vladimir Taft

Hewlett-Packard
1453 Enchanted Way
San Mateo, CA 94402
408-447-2870
vtaft@hpinddf.hp.com

Bob Tausworthe

Hewlett-Packard
19420 Homestead Rd.
Cupertino, CA 95014
408-447-2073
tozz@hpda.hp.com

Michael Tharenos

Stanford University
115 Pine Hall
Stanford, CA 94305
415-725-1392
tharenos@jessica.stanford.edu

Claudio Topolcic

BBN Communications
10 Moulton Street
STC
Cambridge, MA 02138
617-873-3874
topolcic@bbn.com

Lance Travis

Apollo Computer
330 Billerica Rd
Chelmsford, MA 01824
508-256-6600 6799
cmt@apollo.com

Glenn Trewitt

Stanford University
Center for Integrated Systems
Rm 216
Stanford, CA 94305-4070
415-725-3731
trewitt@miasma.stanford.edu

Howard Tsai

AT&T Bell Laboratories
200 Laurel Ave.
Middletown, NJ 07748
201-957-6132
hst@mtuxo.att.com

Paul Tsuchiya

Mitre Corporation
7525 Colshire Drive
McLean, VA 22102
703-883-7352
tsuchiya@gateway.mitre.org

L. Stuart Vance

TGV, Incorporated
15139 Old Ranch Road
Los Gatos, CA 95030
408-353-3939
vance@tgv.com

Lynn Vaughan

Hewlett-Packard
8000 Foothills Blvd.
R3NF3
Roseville, CA 95678
916-785-4470
lynna@hprnd.rnd.hp.com

Ross Veach

University of Illinois
1304 West Springfield
Urbana, IL 61801
217-244-4274
rrv@seka.cso.uiuc.edu

Ruediger Volk

Universitaet Dortmund
Informatik IRB
Postfach 500 500
D-4600 Dortmund 50,
GERMANY
+49-231 755 4760
rv@germany.ue.net, rv@unido.informatik.uni-dortmund.de

Steve Waldbusser

Carnegie-Mellon University
4910 Forbes Avenue
Pittsburgh, PA 15213
412-268-6628
waldbusser@andrew.cmu.edu

Asher Waldfogel

Wellfleet Communications, Inc.
12 DeAngelo Drive
Bedford, MA 01730
617-275-2400
...linus!wellflt!awaldfog

Bill Westfield

cisco Systems
1350 Willow Rd.
Suite 201
Menlo Park, CA 94025
415-326-1941
billw@cisco.com

Richard Wilder

Mitre Corporation
7525 Colshire Dr
M/S W425
McLean, VA 22102
703-883-6174
rick@gateway.mitre.org

Dan Wintringham

Ohio Supercomputer Center
1224 Kinnear Rd.
Columbus, OH 43212
614-292-0901
danw@igloo.osc.edu

Philip Wood

Los Alamos National Laboratory
MX B294
Los Alamos, NM 87545
505-667-2598
cpw@lanl.gov

Mary Youssef

IBM
472 Wheelers Farms Rd.
Milford, CT 06460
203-783-4334
mary@ibm.com

Aileen Yuan

Mitre Corporation
7525 Colshire Dr.
McLean, VA 22102
703-883-7023
aileen@gateway.mitre.org

Lixia Zhang

Xerox(PARC)
333 Coyote Hill Road
Palo Alto, CA 94304
415-494-4471
lixia@lcs.mit.edu

Total number of people in this group: 220.

IV. Final Agenda

STANFORD IETF FINAL AGENDA, 25-28 Jul 89

Chairman: Phill Gross, NRI
Host: Paul Baer, Stanford University

MONDAY, JULY 24TH

9:00 am - 5:00 pm	OSI Interoperability: Review of GOSIP V2 (Hagens, UWisc and Callon, DEC) (Stanford)
1:00 pm - 5:30 pm	Point-to-Point Protocol (Hobby, UCDavis and Perkins, CMU) (Hyatt Rickeys)
6:00 pm - 8:00 pm	Early Registration and Welcoming Reception (Hyatt Rickeys)

TUESDAY, JULY 25TH (STANFORD)

8:15 am	IETF Registration
8:55 am	Call to Order
9:00 am	Opening Plenary, Introductions and Local Arrangements (Phill Gross and Karen Bowers, NRI)
9:15 am	Internet Activities Board (Dave Clark, MIT)
9:30 am - 12:00 pm	Morning Working Group Sessions
	• Host Requirements (Braden, ISI)
	• OSI Interoperability: Inter Domain Routing (Hagens, UWisc and Callon, DEC)
	• Point-to-Point Protocol (Hobby, Davis and Perkins, CMU)
	• ST-Connection IP: CO-IP (Topolcic, BBN)
9:30 am -10:30 am	User Services (Bowers, NRI)
10:45 am -12:00 pm	USER-DOC (Roubicek, BBN and LaQuey, UTexas)
12:00 pm	LUNCH
1:00 pm - 4:00 pm	Afternoon Working Group Sessions
	• Authentication (Schiller, MIT)
	• Host Requirements (Braden, ISI)
	• Interconnectivity (Almes, Rice)
	• OSI Interoperability: Intra Domain Routing (Hagens, UWisc and Callon, DEC)
	• ST-Connection IP: CO-IP (Topolcic, BBN)
	• USER-DOC (Roubicek, BBN and LaQuey, UTexas)
4:15 pm - 5:30 pm	Technical Presentations and Network Status Reports
	• NSFnet Status Report (Elise Gerich, Merit) 15 min
	• Fair Queuing Revisited (Scott Shenker, XEROX PARC) 1 hr.
5:45 pm	RECESS
6:00 pm	A Demonstration of a 16th Century Machine (Jacobsen, ACE and Bates, Richards and Stegman, Stanford Memorial Church)
7:30 pm	Interconnectivity (Almes, Rice) (Hyatt Rickeys)

WEDNESDAY, JULY 26TH (STANFORD)

9:00 am — Opening Plenary (Phill Gross and Karen Bowers, NRI)

9:15 am - 12:00 pm — Morning Working Group Sessions

- Alert Management (Steinberg, IBM)
- Autonomous Network Task Force (Estrin, USC)
- Interconnectivity (Almes, Rice)
- NISI (For Members Only) (Bowers and Gross, NRI)
- OSI Interoperability: General Meeting (Hagens, UWisc and Callon, DEC)
- Performance and CC (Mankin, MITRE)
- Point-to-Point Protocol (Hobby, Davis and Perkins, CMU)
- ST-Connection IP: ST (Topolocic, BBN)

12:00 pm — LUNCH

- JOMANN (Gerich, MERIT)

1:00 pm - 4:00 pm — Afternoon Working Group Sessions

- Autonomous Network Task Force (Estrin, USC)
- Dynamic Host Configuration (Droms, Bucknell)
- NOC-Tools (Stine, SPARTA and Enger, Contel)
- Open Systems Routing (Lepp, BBN)(For Members Only)
- OSI Interoperability: X.500 and DEC DNS (Hagens, UWisc and Callon, DEC)
- Performance and CC (Mankin, MITRE)
- ST-Connection IP: ST (Topolcic, BBN)

4:15 pm - 5:30 pm — Technical Presentations and Network Status Reports

- How Slow is One Gigabit per Second? (Craig Partridge, BBN) 45 min.
- Network Performance Impact of X Window System Protocol (Ralph Droms, NRI) 45 min.

5:45 pm — RECESS

6:15 pm — Tour of the Stanford Linear Accelerator Center (Kathy O'Shaughnessy, Stanford U)

8:00 pm

- OSI Interoperability: OR (Hagens, UWisc and Callon, DEC) (Hyatt Rickeys)
- Point-to-Point Protocol (Hobby, Davis and Perkins, CMU)
- Point-to-Point Protocol (Hobby, Davis and Perkins, CMU)

8:30 pm — Joint USER-DOC and NOC-Tools Session (Roubicek, BBN, Stine, SPARTA and Bowers, NRI)

THURSDAY, JULY 27TH (STANFORD)

9:00 am — Opening Plenary (Phill Gross and Karen Bowers, NRI)

9:15 am - 12:00 pm — Morning Working Group Sessions

- Domain Name System (Mockapetris, USC-ISI)
- Open Systems Routing (Lepp, BBN) (For Members Only)
- Open SPF-Based IGP (Petry, UMD and Moy, Proteon)
- TELNET (Borman, Cray)
- User Services (Hallgren, Cornell)
- IETF Workshop (Gross, NRI)
- Ad Hoc T1 Point-to-Point Group (Cohen, USC-ISI)

12:00 pm — LUNCH

12:40 pm — National Network (MCI video) 19 min.

1:00 pm - 5:30 pm — Technical Presentations and Network Status Reports

- IAB Update - Status and Priorities (Vint Cerf, NRI) 15 min.
- IETF Restructuring (Phill Gross, NRI) 15 min.
- Gateway Congestion Control (Allison Mankin, MITRE) 45 min.
- IP Option for Crypto Summing (Jeff Schiller, MIT) 20 min.
- The Terrestrial Wideband Network (Claudio Topolcic, BBN) 45 min.
- High Speed Networking using OSI Protocols (Bob Beach, Ultra Network Technologies) 30 min.
- Domain Name Status Report (Mary Stahl and Mark Lottor, SRI) 10 min.
- Point-to-Point Protocol Specs (Drew Perkins, CMU) 30 min.
- Some European Internet Activities (Ruediger Volk, Universitat Dortmund, Germany)
- State of the Internet (Zbigniew Opalka, BBN) 10 min.
- ESNet Status Report (Steven Hunter, LLNL) 10 min.

5:45 pm — RECESS

6:00pm — Tour of a WAITS System (Almquist, Grossman and Frost, Stanford U)

FRIDAY, 28TH JULY (STANFORD)

9:00 am	Opening Plenary (Phill Gross, NRI)
9:15 am - 11:30 am	Working Group Reports

- User Services (Bowers, NRI)
- NISI (Bowers, NRI)
- NOC-Tools (Stine, SPARTA)
- USER-DOC (Roubicek, BBN)
- Dynamic Host Configuration (Droms, NRI)
- Alert Management (Steinberg, IBM)
- Authentication (Schiller, MIT)
- Open SPF-Based IGP (Petry, UMD)
- Open Systems Routing (Little, SAIC, for Lepp, BBN)
- ST and Connection IP (Toplocic, BBN)
- OSI Interoperability (Hagens, UWisc)
- JOMANN (Hastings, PSCC, for Gerich, MERIT)
- TELNET (Borman, Cray)
- Domain Name System (Gross, NRI, for Mockapetris, ISI)
- Host Requirements (Braden, ISI)

11:30 am	Concluding Plenary Remarks and Group Discussions (Phill Gross, NRI)
12:00 pm	Adjournment

V. Working Group Summaries

- Charters
- Status Updates
- Current Meeting Reports

Alert Management Working Group
Chairperson: Louis Steinberg/IBM

CHARTER

Description of Working Group:

The Alert Management Working Group is chartered with defining and developing techniques to manage the flow of asynchronously generated information between a manager (NOC) and its remote managed entities.
The output of this group should be fully compatible with the letter and spirit of SNMP (RFC 1067) and CMOT (RFC 1095).

Specific Objectives:

1. Develop, implement, and test protocols and mechanisms to prevent a managed entity from burdening a manager with an unreasonable amount of unexpected network management information. This will focus on controlling mechanisms once the information has been generated by a remote device.
2. Write an RFC detailing the above, including examples of its conforment use with both SNMP traps and CMOT events.
3. Develop, implement, and test mechanisms to prevent a managed entity from generating locally an excess of alerts to be controlled. This system will focus on how a protocol or MIB object might internally prevent itself from generating an unreasonable amount of information; examples of such techniques might include limiting number of alerts per time period, delayed reporting of "good news" (as in the link up sgmp trap on NSFNET), or the use of thresholds.
4. Write an RFC detailing the above. Since the implementation of these mechanisms is protocol dependent, the goal of this RFC would be to offer guidance only. It would request a status of "optional".

Estimated Timeframe for Completion:

A draft of the first RFC (alert flow control) will be written and reviewed by the July IETF meeting, with final review expected at the October IETF meeting. The second RFC draft will be submitted for initial review at the October IETF meeting. A date for final review of this document has not yet been determined.

STATUS UPDATE

1. Chairman: Louis Steinberg, louiss@ibm.com
2. WG Mailing List: alert-man@merit.edu
   alert-man-request@merit.edu
3. Last Meeting: July 26, 1989, Stanford, CA
4. Next Meeting: October 31 - November 3, 1989 - Hawaii IETF
5. Pending or New Objectives: see objectives in Charter
6. Progress to date (e.g., documents produced):

   First document (alert flow control) to be posted as an Internet-Draft shortly after the July IETF Meeting.

Alert Management Working Group
Chairperson: Louis Steinberg/IBM

CURRENT MEETING REPORT

Reported by Lee Oattes

AGENDA

- Introduction
- Discussion of draft flow control document
- Preliminary discussion of alert-generation document note: this was shelved due to a lack of time

ATTENDEES

1. Bierbaum, Neal/vitam6!bierbaum@vitam6
2. Carter, Glen/gcarter@ddn1.dca.mil
3. Cohn, George/geo@ub.com
4. Cook, John/cook@chipcom.com
5. Denny, Barbara/denny@sri.com
6. Easterday, Tom/tom@nisca.ircc.ohio-state.edu
7. Edwards, David/dle@cisco.com
8. Fedor, Mark/fedor@nisc.nyser.net
9. Hunter, Steven/hunter@ccc.mfecc.llnl.gov
10. Kincl, Norman/kincl@iag.hp.com
11. Malkin, Gary/gmalkin@proteon.com
12. Oattes, Lee/oattes@utcs.utoronto.ca
13. Paw, Edison/esp@esd.ecom.com
14. Replogle, Joel/replogle@ncsa.uiuc.edu
15. Salo, Tim/tjs@msc.umn.edu
16. Sheridan, Jim/jsherida@ibm.com
17. Taft, Vladimir/vtaft@hpinddf.hp.com
18. Waldbusser, Steve/sw0l@andrew.cmu.edu
19. Wintringham, Dan/danw@osc.edu
20. Steinberg, Louis/louiss@ibm.com

MINUTES

1. The meeting of the Alert Management Working Group began with an introduction from the Chairman (Lou Steinberg).
2. A discussion of several independent implementations of feedback/pin and polled, logged alerts led to an agreement to adopt these mechanisms in some form.
3. The following questions were answered by discussion and consensus:
   (a) Can we have a read-only alerts_enabled mib object, by limiting the transmission rate of alerts (no shutoff) and not use feedback? No.
   We need a total shutoff mechanism in case a number of alert generators are "screaming" all at once. The total traffic might be too much for the manager, and this "stable" situation cannot improve (while a disabling mechanism would tend to be self-correcting).
   Total shutoff implies the use of a resetable, read-write mib object.
   An automated, timer-based reset mechanism was discussed but it was felt that such a system might tend to sync resets of multiple generators and could still lead to an over-reporting condition.
   (b) Might an automated-reset of alerts_enabled from the manager station create a "blast-off-blast-off..." alert traffic pattern?
   Yes, but such a manager would still tend to only get as much traffic as he could handle. A re-enable would only be sent when the manager isn't swamped (i.e., is capable of sending one).
   A manager experiencing such a traffic pattern should readjust his window prior to setting alerts_enabled TRUE.
   (c) When pin disables alerts due to the generation of many similar alerts (e.g., link flapping) might we also lose an unrelated alert from the same system prior to resetting alerts_enabled?
   Yes, but the rate limiting (as opposed to shutoff) technique has the same problem; the probability of sending a single, specific alert is much lower than the probability of sending any one of many identical alerts.
   This problem is minimized by using polled, logged alerts along with feedback/pin (could still lose alerts if log is overwritten).
   (d) Should we allow the implementation to decide if alerts are totally disabled or limited to a max rate? No.
   Implementations should be consistent since this affects the way we manage our alert generators.
   (e) Can the alert log in polled, logged alerts be overfilled?
   Yes, but the standard suggests that a manager should attempt to keep the log empty by removing known alerts.
   If an individual implementation has no mechanism for removing old alerts (no set) then the log must wrap when full and the manager might lose alerts.

(f) If using the SNMP get-next, do we want the oldest logged element first, or the newest first?
Clearly the manager wants the oldest first if a full log will wrap...this gives him the most chance to see the oldest alert (in a full log) before losing it.
No real concensus here. It seems as though this should be implementation specific since it only applies to SNMP, and since the log, actually being a table, makes this a question of "are new table entries added at the table top or bottom?".

(g) Can we shrink the log size by stripping out only the "important" information from each alert?
We can, but this is something we decided we shouldn't do. It requires a different parser at the manager (can't run it through the alert parser), and we did not know how do decide what information might be needed (it varied with the protocol and alert type).

(h) How about only logging alerts, and sending an "alert logged" alert for each new log entry? The manager gets the asynch. "alert logged" notice and reads the alert log to determine what happened.
While this is an interesting concept, it was felt that it might tend to aggravate some of the other logging problems (e.g., if the log is filled and not over-writing, the only chance of getting the alert information is from the async alert...this removes the asynch alert information and replaces it with "see the log" information).

(i) A discussion of the cpu cycles and memory needed for keeping a log followed. Since the log size might be settable (to 0) it was felt that systems could allow managers to disable logging. It was also felt that the performance and memory hits were not large, but numbers to confirm this were not available.

4. The following were decided by vote:

(a) Feedback/Pin
Mandatory mib objects:

alerts_enabled	read/ write
window (time)	read/ optional write
max_alerts	read/ optional write

Do not include alert counters as mib objects for this document. Individual implementors will decide if they need total dropped and/or sent, but not everybody likes the idea of adding more counters as (even optional) mib objects.
Do not optionally allow a reduced rate mode on the over reporting condition...require total async. Alerts to be shutoff for reasons given in earlier discussion.

(b) Polled, Logged Alerts Remove time field from the table, as most alerts are time stamped and the information in an alert should be defined by the protocol...not us.

ALERT-MAN

ALERT-MAN @ MERIT.EDU
ALERT-MAN-REQUEST @ MERIT.EDU

ALERT MANAGEMENT
OR
SUPER HERO ?

MEETING GOALS

→ NO HOLY WARS (LOU ISN'T A TARGET)

a) USE OBSCURE MEETING ROOM

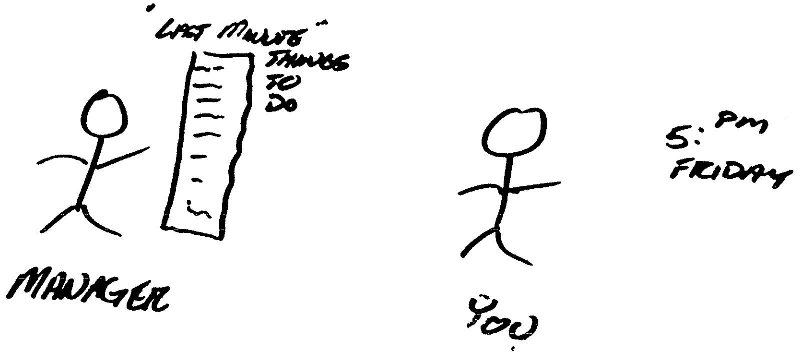
b) LOU'S DISGUISE - NO SUIT

→ DON'T BREAK SNMP OR CMOT

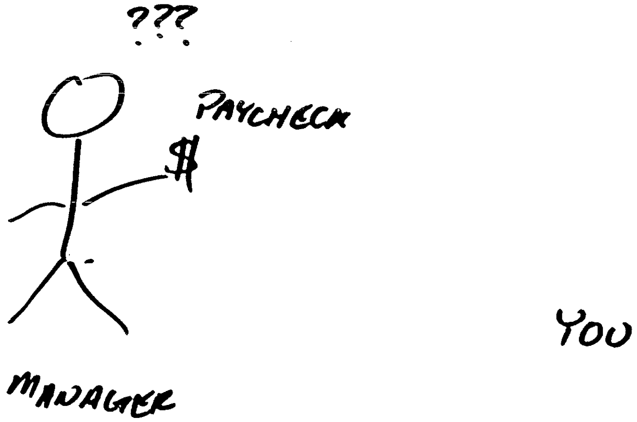
→ "WRAP UP" 1st DOC. (FLOW CONTROL) DRAFT

FOR DETAILS...
READ THE DRAFT

OVER INFORMING:



UNDER INFORMING



PROTOCOL STATUS

→ REQUIRED FEEDBACK MECHANISM TO HELP PREVENT "OVER-INFORMING" MANAGER - TOO MANY ALERTS

→ ALERT LOG TO FIX "UNDER-INFORMING" PROBLEMS CREATED BY ABOVE (OTHER BENEFITS INCLUDE HISTORICAL PERSPECTIVE, SYNCHRONOUS MECHANISM, ETC).

STATUS

→ AT LEAST 3 DISTINCT WORKING IMPLEMENTATIONS

→ DRAFT REVISIONS POSTED WITHIN A WEEK TO LIST

→ DRAFT POSTED ≈ 2 WEEKS

"THERMO GRAM" (WHY 2 WEEKS?)
AVOIDANCE

→ WAIT UNTIL OTHERS JUST POSTED THEIR DRAFTS

→ FIGURE OUT HOW TO SHUT OFF OUR READERS

→ CHANGE DRAFT NAME TO CONFUSE READER
(e.g. How I spent my SUMMER VACATION)

COMING SOON TO AN IETF NEAR YOU:

→ DOC #2 : HOW TO CONTROL GENERATION (RATE) OF ASYNCH. ALERTS.

Authentication Working Group
Chairperson: Jeffrey Schiller/MIT

CHARTER

Description of Working Group:

To brainstorm issues relating to providing for the security and integrity of information on the Internet, with emphasis on those protocols used to operate and control the network. To propose open standard solutions to problems in network authentication.

Specific Objectives:

1. RFC specifying an authentication format which supports multiple authentication sy tems.
2. Document discussing the cost/benefit tradeoffs of various generic approaches to sol ing the authentication problem in the Internet context.
3. Document to act as a protocol designers guide to authentication.
4. RFC proposing A Key Distribution System (emphasis on "A" as opposed to "THE" MIT's Kerberos seems the most likely candidate here.

Estimated Timeframe for Completion:

This working group will hopefully complete its current objectives within one year. At this point the group will either disband or will move on to other related problems/issues.

STATUS UPDATE

1. Chairperson: Jeffrey Schiller, jis@bitsy.mit.edu
2. WG Mailing list: AWG@BITSY.MIT.EDU
3. Last Meeting: July 25-28, 1989, Stanford IETF
4. Next Meeting: October 31 - November 3, 1989, Hawaii IETF
5. Pending or New Objectives: see Charter
6. Progress to Date (e.g., documents produced):

A draft RFC proposing a standard authentication format for multiple protocols was circulated at the last meeting (addresses object [1] above).

A draft "Authentication Requirements" document was also circulated. This is the beginning of an effort that will lead to writing a protocol designers guide to authentication (addresses objective [3] above).

A draft RFC for the Kerberos Authentication system was circulated as well (addresses objective [4] above).

Authentication Working Group
Chairpersons: Jeff Schiller/MIT and Jon Rochlis/MIT

CURRENT MEETING REPORT

Reported by Jeff Schiller

ATTENDEES

1. Carter, Glen/gcarter@ddn1.dca.mil
2. Cook, John/cook@chipcom.com
3. Galvin, James M./galvin@tis.com
4. Jerian, Chuck/cpj@eng.sun.com
5. Kincl, Norman/kincl@iag.hp.com
6. Lee, Young/youngl@jessica.stanford.edu
7. Leser, Norbert/nl@osf.org
8. LoVerso, John/loverso@xylogics.com
9. Mamakos, Louis/louie@trantor.umd.edu
10. McCloghrie, Keith/kzm@twg.com
11. Merritt, Don/don@brl.mil
12. Mundy, Russ/mundy@tis.com
13. Natalie, Ron/ron@rutgers.edu
14. Pak, Raylene/raylene@tardis.tymnet
15. Petry, Mike/petry@trantor.umd.edu
16. Rochlis, Jon/jon@mit.edu
17. Schiller, Jeff/jis@bitsy.mit.edu
18. Schofield, Bruce J./schofield@edn-vax.dca.mil
19. Tharenos, Michael/tharenos@jessica.stanford.edu

MINUTES

The AWG met briefly at the July IETF at Stanford. The meeting was short and basically consisted of announcements. The following was announced by myself:

- The IP Authentication Option document will be submitted as an Internet Draft
- Work is progressing on a document entitled "Authentication and Privacy in the SNMP" which attempts to address the issue of adding authentication to SNMP. This document is currently being worked on by its authors and should be ready to be circulated to the AWG on or before the October IETF meeting.

The next planned meeting of the AWG is for the October IETF.

CMIP-over-TCP (CMOT) Working Group
Chairperson: Lee LaBarre/Mitre

CHARTER

Description of Working Group:

- Develop a long term approach to management of the Internet based on the OSI Network Management Framework and the Common Management Information Protocol (CMIP).
- Provide input to the OSI standards process based on experience in the Internet, and thereby influence the final form of OSI International Standards on network management, in particular CMIS/P.

Specific Objectives:

1. Develop prototype implementors agreements on CMIP over TCP.
2. Develop prototype implementations based on the CMOT agreements and IETF SMI and MIB agreements.
3. Experiment with CMOT and extensions to the SMI and MIB.
4. Develop final implementors agreements for CMOT.
5. Promote development of products based on CMOT.
6. Provide input to the OSI Network Management standards process in time to effect the International Standards.

Estimated Timeframe for Completion:

The group's work should be completed by June 1989.

STATUS UPDATE

1. Chairperson: Lee LaBarre, cel@mitre.org.com
2. WG Mailing List: netman@gateway.mitre.org
3. Last meeting: 19 January, 1989, Austin, Texas
4. Next Meeting: TBD as required
5. Pending or New Objectives:

   The remaining tasks for the group include:

   - updating the specification when the OSI standards reach international standard (IS) status,
   - specification of event generation and event report control mechanisms.
6. Progress to date (e.g., documents produced):
   - RFC1095, "The Common Management Information Services and Protocol over TCP/IP (CMOT", edited by U. Warrier and L. Besaw
   - The group has completed a major portion of its charter to develop a long term approach to network management, namely an specification of an architecture and protocol that is consistent with OSI and will facilitate management of future networks containing TCP/IP and OSI components. That specification, contained in RFC1095, is based on the DIS version of CMIP, and on Internet RFCs. The RFC1095 and the new SNMP RFC1098 have been given equal status by the IAB.

CURRENT MEETING REPORT

Did not meet

Domain Working Group
Chairperson: Paul Mockapetris/USC/ISI

CHARTER

Description of Working Group:

The goal of the Domain Working Group is to advise on the administration of the top levels of the DNS ("the root servers"), consider proposed extensions and additions to the DNS structure and data types, and resolve operational problems as they occur.

Specific Objectives:

The specific short-term objectives are:

- Adding load balancing capability to the DNS.
- Adding DNS variables to the MIB.
- Implementation catalog for DNS software.
- Responsible Person Record.
- Adding network naming capability to the DNS.
- Evaluate short term measures to improve, or at least describe the security of the DNS.

Estimated Timeframe for Completion (for above objectives):

1. The preferred method for Load Balancing was decided upon at the April '89 IETF meeting at Cocoa Beach. A short RFC will be written before the next meeting in July '89.
2. End of 1989
3. Questionaire sent, responses data being organized, summary and detail to appear. (PVM)
4. July '89.
5. RFC issued April 89, implementations to follow.

STATUS UPDATE

1. Chairperson: Paul Mockapetris / pvm@isi.edu
2. WG Mailing Lists(s): namedroppers@sri-nic.arpa
3. Date of Last Meeting: July '89, Stanford University
4. Date of Next Meeting: Oct-Nov '89, Hawaii IETF
5. Pending or New Objectives: Efficiency Problems and Enhancements
6. Progress to Date (e.g., documents produced):
    - RFC 1101 - on Network Name Mapping
    - Advice to Internet Host Requirements Editor

Domain Working Group
Chairperson: Paul Mockapetris/USC/ISI

CURRENT MEETING REPORT

Reported by Paul Mockapetris

AGENDA

- Redeployment of high level servers.
- Short and Long Term fixes for excessive DNS usage reported in the NSFNET and elsewhere.
- What should the DWG suggest to the Host Requirements WG.
- Addition of dynamic add and delete to the DNS.
- Enhancements to the DNS in general.

ATTENDEES

1. Almquist, Phil/almquist@jessica.stanford.edu
2. Brackenridge, Billy/brackenridge@isi.edu
3. Burgan, Jeffrey/jeff@nsipo.nasa.gov
4. Crocker, Dave/dcrocker@ahwahnee.stanford.edu
5. Edwards, David/dle@cisco.com
6. Fedor, Mark/fedor@nisc.nyser.net
7. Kincl, Norman/kincl@iag.hp.com
8. Lottor, Mark/mkl@nic.ddn.mil
9. Natalie, Ron/ron@rutgers.edu
10. St. Johns, Mike/stjohns@beast.ddn.mil
11. Stahl, Mary/stahl@sri-nic.arpa
12. Volk, Ruediger/rv@germany.eu.net
13. Woods, C. Philip/cpw@lanl.gov

MINUTES

The Domain Working Group met at Stanford University IETF. Mike St. Johns discussed some possibilities for offloading some of the top-level domains, such as EDU and COM, from management by the NIC.DDN.MIL. Some preliminary thoughts were presented, but a firm plan has not yet been made. The majority of the meeting was spent discussing recent DNS usage problems, cures, and the most needed repairs to BIND.

Problems:

The best known aspect of the usage problems was NSFNET observations of 20% DNS packets on some links at certain times. Traffic monitoring revealed that these large packet fluxes were from relatively few sites, the so called "screamers". The screamers are typically sites with Sun's YP using the DNS as a backstop, i.e. configured so that queries which cannot be answered by YP drop into the DNS. The trouble is that under certain cases YP retries DNS queries as fast as possible, so a simple failure is repeated over and over.

The same problem also caused more severe consequences in local environments. In one case, DNS screaming leading to gateway overload, leading to gated cycle starvation, leading to EGP problems, leading to connectivity loss. In another, the same traffic which was 20% of a NSFNET T1 was more than 100% of a 56Kbit link.

In addition to the screaming phenomena, others noted low level useless traffic which becomes significant when multiplied by the large number of hosts, but still much less than screaming.

Cures:

DNS screaming has been fixed by new Sun YP software. However, others could easily make the same mistake, so in the future we need firewalls to stop this behavior in both the resolver and name server since we cannot always assume control of either. The method is an extension of negative caching.
The extensions and already defined negative caching mechanisms are needed even if screamers are fixed so that the system will continue to scale up.
Total load of DNS should be 1% or less.

BIND needs:

The attendees made the following list of the most important problems with existing DNS implementations, usually BIND.

- All retry mechanisms should use exponential backoff, with settable upper and lower limits.
- Negative caching of:
  - Name errors and no data as in RFCs
  - Temporary failures
  - Server failures
- Cooperation between forwarding name servers and waiting ACKs to resolvers.
- Satisfactory implementation TTL=0 RR handling.

- Correct operation in an environment without root server connectivity.
- Correct implementation of master file defaults and minimums.
- Broadcast and multicast implementation.

ACTION ITEMS

1. P. Mockapetris to produce detailed draft of problems and proposed cure
2. Group of interested parties to draft incremental update method.

DOMAIN WG

* DISCUSSION OF FUTURE EDU ZONE MIGRATION AND OTHER STEPS TO OFFLOAD THE SRI-NIC - M. StJohns

~~* [illegible]~~

* FORMULATION OF MESURES TO IMPROVE NAME SERVICE (I.E. GIVE NSFNET BACK 19+%). DOCUMENT TO DESCRIBE MEASURES TO FOLLOW. (PVM action item)

* DYNAMIC / INCREMENTAL UPDATE DISCUSSION

Dynamic Host Configuration Working Group

Chairpersons: Ralph Droms/UMD and Phill Gross/NRI

CHARTER

Description of Working Group:

> The purpose of this working group is the investigation of network configuration and reconfiguration management. We will determine those configuration functions that can be automated, such as Internet address assignment, gateway discovery and resource location, and that which cannot (i.e., those that must be managed by network administrators).

Objectives:

1. We will identify (in the spirit of the Gateway Requirements and Host Requirements RFCs) the information required for hosts and gateways to:
   (a) Exchange Internet packets with other hosts (e.g., discover own Internet address).
   (b) Obtain packet routing information (e.g., discover local gateways).
   (c) Access the Domain Name System (e.g., discover a DNS server).
   (d) Access other local and remote services.
2. We will summarize those mechanisms already in place for managing the information identified by objective 1.
3. We will suggest new mechanisms to manage the information identified by objective 1.
4. Having established what information and mechanisms are required for host operation, we will examine specific scenarios of dynamic host configuration and reconfiguration, and show how those scenarios can be resolved using existing or proposed management mechanisms.

Estimated Timeframe for Completion: (to be determined)

STATUS UPDATE

1. Chairpersons: Ralph Droms, rdroms@nri.reston.va.us
   Phill Gross, pgross@nri.reston.va.us
2. WG Mailing List: host-conf@rutgers.edu
3. Last Meeting: July 26, Stanford IETF
4. Next Meeting: Midterm video-conference or Hawaii IETF in October
5. Pending or New Objectives: Internet Draft describing dynamic host configuration problem. To be completed before Hawaii IETF.
6. Progress to Date (e.g., documents produced):

Dynamic Host Configuration Working Group
Chairpersons: Ralph Droms/Bucknell University and Phill Gross/NRI

CURRENT MEETING REPORT

Reported by Ralph Droms

AGENDA

- Review Objectives from April meeting
- Agree on statement of required configuration information
- Discuss existing configuration protocols

ATTENDEES

1. Borman, Dave/dab@cray.com
2. Brunner, Eric/brunner@monet.berkeley.edu
3. Cook, John/cook@chipcom.com
4. Deering, Steve/deering@pescadero.stanford.edu
5. Denny, Barbara/denny@sri.com
6. Droms, Ralph/rdroms@nri.reston.va.us
7. Edwards, David/dle@cisco.com
8. Fair, Erik/fair@apple.com
9. Fox, Richard/rfox@suntan.tandem.com
10. Gilligan, Bob/gilligan@sun.com
11. Gross, Phill/pgross@nri.reston.va.us
12. Jordt, Dan/danj@cac.washington.edu
13. Lear, Eliot/lear@net.bio.net
14. Lottor, Mark/mkl@nic.ddn.mil
15. LoVerso, John/loverso@xylogics.com
16. Maas, Andy/maas@jessica.stanford.edu
17. Mamakos, Louis/louie@trantor.umd.edu
18. Mockapetris, Paul/pvm@isi.edu
19. Morgan, Bob/morgan@jessica.stanford.edu
20. Mundy, Russ/mundy@tis.com
21. Natalie, Ron/ron@rutgers.edu
22. Partridge, Craig/craig@nnsc.nsf.net
23. Perkins, Drew/ddp@andrew.cmu.edu
24. Petry, Mike/petry@trantor.umd.edu
25. Rochlis, Jon/jon@mit.edu
26. Romkey, John/romkey@asylum.sf.ca.us
27. Satz, Greg/satz@cisco.com
28. Schiller, Jeff/jis@bitsy.mit.com
29. Skinner, Greg/gds@spam.istc.sri.com
30. Westfield, Bill/billw@cisco.com

MINUTES

The Stanford meeting began with a review of the objectives discussed at the April meeting. The next topic was a discussion of information required by a host to participate in Internet communications:

- IP address
- Subnet properties
  - Subnet mask
  - MTU
  - Broadcast address
- Default gateway
- DNS server
- Domain name of host
- High-level services
  - Boot services
  - Other gateways
  - Local network topology

Having agreed on the list of configuration information, the group developed a list of mechanisms required to distribute and maintain host configurations:

- IP address discovery
- IP address allocation
- Subnet properties discovery
- Gateway discovery
- Gateway reconfiguration
- DNS server discovery

- Domain name allocation
- DNS update/name, IP address registration
- High-level resource discovery

These mechanisms can be loosely grouped into two categories: discovery mechanisms, which transmit configuration information to a host, and allocation mechanisms, which determine host-specific information. For example, the mechanism used to transmit an IP address to a host can be entirely independent of the mechanism used to select that IP address.

Several members of the group described details of existing configuration mechanisms (both at the meeting and immediately afterward, through the host-config@rutgers.edu mailing list). There are several existing protocols of interest to this group:

- ICMP
- BOOTP
- Athena NIP (Schiller and Rosenstein, MIT)
- Dynamic IP Address Assignment (Morgan, Stanford)
- DRARP (Brownell, Sun)

At the next meeting, we will discuss an Internet Draft that summarizes the conclusions reached at the Stanford meeting.

CONFIGURATIONS

- IP ADDRESS
- SUBNET PROPERTIES
  - SUBNET MASK
  - MTU
- DEFAULT GATEWAYS
- DNS SERVERS
- DOMAIN NAME
- HIGH-LEVEL NAMES
- HIGH-LEVEL SERVICES

CONSTRAINTS

- CONFIGURATION/RECONFIGURATION
- DEGREE OF PERMANENCE
- AUTHORITY MECHANISMS
- SCALE
- FUNCTIONALITY OF NETWORK
- STYLES OF SYSTEM ADMINISTRATION

MECHANISMS

- IP ADDRESS DISCOVERY
- IP ADDRESS ALLOCATION
- SUBNET PROPERTIES DISCOVERY
- GATEWAY DISCOVERY
- GATEWAY RECONFIGURATION
- DNS SERVER DISCOVERY
- DOMAIN NAME ALLOCATION
- DNS UPDATE
- HIGH-LEVEL RESOURCE DISCOVERY

SCENARIOS

- NEW HOST INITIALIZATION
- EXISTING HOST REBOOT
- RELOCATED HOST INITIALIZATION
- MOBILE HOST

- DISTRIBUTED VS. CENTRALIZED IP AUTHORITY
- LOCAL VS. GLOBAL DNS

NEXT MEETING

- APPROVAL OF MECHANISMS LIST
- DISCUSSION OF EXISTING PROTOCOLS
  - ICMP
  - BOOTP
  - ATHENA - SCHILLER
  - DEERING/PARTRIDGE ICMP GATEWAY DISCOVERY
- APPROVAL OF SCENARIOS LIST
- BEGIN DEVELOPMENT OF NEW PROTOCOLS

Host Requirements Working Group
Chairperson: Robert Braden/ISI

CHARTER

Description of Working Group:

The Host Requirements Working Group has the goal of producing an RFC defining the official requirements for the software on a host which is to be part of the Internet.

Specific Objectives:

1. Produce a document that is the host equivalent of RFC-1009, "Requirements for Internet Gateways", providing guidance for vendors, implementors, and users of host software for internet applications.
2. Enumerate the protocols required, referencing the RFC's and other documents describing them in detail.
3. Provide further clarification, discussion, and guidance in those areas of the referenced specifications that contain ambiguous or incomplete information.
4. Define the current architecture as completely and carefully as possible, don't invent new architecture.
5. As a secondary task, provide a forum for discussing particular solutions to pressing host problems.

Estimated Timeframe for Completion:

Our objective is to publish the document early in 1989.

STATUS UPDATE

1. Chairperson: Bob Braden, braden@isi.edu, (213) 822-1511
2. WG Mailing List: ietf-hosts@NNSC.NSF.NET
3. Last Meeting: Stanford IETF, July 1989
4. Next Meeting: None
5. Pending or New Objectives: see Charter
6. Progress to date (e.g., documents produced):

   The Host Requirements RFC has been split into two sections, one covering the link, Internet, and transport layers, and the other covering the application and support programs. They are approximately equal size and total slightly over 200 pages.

   The final versions are now being prepared for submission as RFC's.

Host Requirements Working Group
Chairperson: Robert Braden/ISI

CURRENT MEETING REPORT

Reported by Robert Braden

At Stanford, the WG met and dealt with all the issues that have been raised since January 1989. Where the issue could be agreed upon, new text was drafted; other issues were deferred for some future revision of the document.

ATTENDEES

1. Almquist, Phil/almquist@jessica.stanford.edu
2. braden, Bo/braden@isi.edu
3. Borman, Dave/dab@cray.com
4. Brooks, Charles E./ceb@wdl1.fac.ford.com
5. Burgan, Jeffrey/jeff@nsipo.nasa.gov
6. Burnett, John/ucbvax!hpda!jlb'
7. Carter, Glen/gcarter@ddn1.dca.mil
8. Cook, John/cook@chipcom.com
9. Crocker, Dave/dcrocker@ahwahnee.stanford.edu
10. Deboo, Farokh/fjd@bridge2.esd.3com.com
11. Deering, Steve/deering@percadero.stanford.edu
12. Gross, Martin/martin@edn-unix.dca.mil
13. Grossman, Stu/grossman@score.stanford.edu
14. Hedrick, Charles/hedrick@aramis.rutgers.edu
15. Jerian, Chuck/cpj@eng.sun.com
16. Karels, Mike/karels@berkeley.edu
17. Karn, Phil/karn@thumper.bellcore.com
18. Lear, Eliot/lear@net.bio.net
19. Lee, Young/youngl@jessica.stanford.edu
20. Lekashman, John/lekash@orville.nas.nasa.gov
21. Lynn, Charles/clynn@bbn.com
22. Mathis, Matt/mathis@fornax.ece.cmu.edu
23. Mockapetris, Paul/pvm@isi.edu
24. Pak, Raylene/raylene@tardis.tymnet
25. Partridge, Craig/craig@bbn.com
26. Reilly, Michael/reilly@atari.nac.dec.com
27. Skinner, Greg/gds@spam.istc.sri.com
28. Solensky, Frank/solensky@interlan.interlan.com
29. Westfield, Bill/billw@cisco.com

Interconnectivity Working Group
Chairperson: Guy Almes/Rice

CHARTER

Description of Working Group:

We aim to improve practical inter-autonomous system routing in the Internet.

Specific Objectives:

Produce a practical system for Inter-Autonomous System routing that is (a) significantly better than the current system based on EGP-2 and the Stub Model, and (b) significantly more timely than we expect the outcome of the Open Routing Working Group to be. We hope to produce:

- a Mid-Term Inter-AS Routing Architecture, and
- a Border Gateway Protocol both implemented and deployed.

Estimated Timeframe for Completion:

April 1990

STATUS UPDATE

1. Chairperson: Guy Almes, almes@rice.edu
2. WG Mailing List(s): IWG@rice.edu
3. Date of Last Meeting: July 1989 IETF at Stanford.
4. Date of Next Meeting: 31 Oct - 3 Nov 89, Hawaii IETF
5. Pending or New Objectives:

   Revision of the draft RFCs by the autumn IETF meeting.

6. Progress to Date (e.g., documents produced):

   We have draft RFCs of both the MIRA architecture and the BGP protocol.

Interconnectivity Working Group
Chairperson: Guy Almes/Rice

CURRENT MEETING REPORT

Reported by Guy Almes

AGENDA

- Tuesday afternoon: Open meeting to do a review of concept with other IETFers and obtain feedback on the appropriateness of our objectives.
- Tuesday evening: Work on MIRA Architecture.
- Wednesday morning: Work on BGP Usage.

ATTENDEES

1. Almes, Guy/almes@rice.edu
2. Breslau, Lee/breslau@jerico.usc.edu
3. Brim, Scott/swb@devvax.tn.cornell.edu
4. Burgan, Jeffrey/jeff@nsipo.nasa.gov
5. Carter, Glen/gcarter@ddn1.dca.mil
6. Choy, Joe/choy@ncar.ucar.edu
7. Crocker, Dave/dcrocker@ahwahnee.stanford.edu
8. Deboo, Farokh/fjd@bridge2.esd.3com.com
9. Denny, Barbara/denny@sri.com
10. Doo, Way-Chi/wcd@bridge2.esd.3com.com
11. Edwards, David/dle@cisco.com
12. Enger, Robert/enger@sccgate.scc.com
13. Estrin, Deborah/estrin@usc.edu
14. Fair, Erik/fair@apple.com
15. Farinacci, Dino/dino@bridge2.3com.com
16. Fedor, Mark/fedor@nisc.nyser.net
17. Fuller, Vince/vaf@jessica.stanford.edu
18. Gerich, Elise/epg@merit.edu
19. Grossman, Stu/grossman@score.stanford.edu
20. Hastings, Gene/hastings@morgul.psc.edu
21. Hedrick, Charles/hedrick@aramis.rutgers.edu
22. Honig, Jeffrey/jch@sonne.tn.cornell.edu
23. Ilnicki, Ski/ski
24. Jones, Bill/jones@nsipo.arc.nasa.gov
25. Jordt, Dan/danj@cac.washington.edu
26. Katz, Dave/dkatz@merit.edu
27. Kaufman, David/dek@proteon.com
28. Lepp, Marianne/mlepp@bbn.com
29. Lougheed, Kirk/lougheed@cisco.com
30. Mathis, Matt/mathis@fornax.ece.cmu.edu
31. Medin, Milo/medin@nsipo.nasa.gov
32. Mundy, Russ/mundy@tis.com
33. Nitzan, Rebecca/nitzan@ccc.nmfecc.llnl.gov
34. Rekhter, Yakov/yakov@ibm.com
35. Replogle, Joel/replogle@ncsa.uiuc.edu
36. Roberts, Ron/roberts@jessica.stanford.edu
37. Satz, Greg/satz@cisco.com
38. Schoffstall, Martin/schoff@nisc.nyser.net
39. St. Johns, Mike/stjohns@beast.ddn.mil
40. Steinberg, Lou/louiss@ibm.com
41. Tsuchiya, Paul/tsuchiya@gateway.mitre.org
42. Veach, Ross/rrv@seka.cso.uiuc.edu
43. Volk, Ruediger/rv@germany.eu.net
44. Wintringham, Dan/danw@osc.edu

MINUTES

Tuesday afternoon we had an open meeting to review MIRA and BGP concepts. The notion of Route Server, the structure of MIRA, and the notion of Full-AS-Path were all discussed in detail, and comments were solicited. Was this doable? Would it advance the state of connectivity and quality/stability of Inter-AS routing? In all these cases, we heard no substantive negative remarks. This enabled us to proceed with our more technical sessions, confident that MIRA and BGP would be useful if properly designed and implemented.

Tuesday evening we met to discuss detailed questions related to the implementability of MIRA.

In the general MIRA case, the Route Servers and Border Gateways are not the same machine and are not even co-located. This makes the tasks of what EGP calls Neighbor Reachability difficult. We agreed to focus on the case in which each Route Server shares a network, typically an Ethernet, with one or more Border Gateways of its AS.

Another technical problem relates to the transient situation in which a transit AS's Inter-AS route to a destination changes. The AS must stop advertising its old route, then its new route must be usable and used and propagated through the Interior Gateways of its AS, and then it can advertise its new route to other ASes. Flash updates with the AS's IGP and engineering of non-huge diameters will be key. We returned to this issue on Wednesday.

Another issue was the determination of up/down status of the link between adjacent ASes. In many protocols, such as RIP, there is no up/down protocol other than the receipt of routing packets; this leads to grave problems when diode situations arise. Even in modern protocols, such as the IS-IS protocol used within the NSFnet Backbone, up/down protocols may fail. A recent case was discussed in which a non-trivial bit-error rate existed on a serial line of the Backbone. The rate was low enough to allow most of the (very small) packets used in the up/down protocol to get through. The rate was large enough, however, to corrupt most large data packets, so the link was essentially useless, even though the IS-IS up/down protocol had determined the link was up. Some of the group have concluded that the only reliable up/down protocol approach will be to use monitoring protocols such as SNMP, with careful implementations adapted to the particular kind of physical/link layers used. During the near term, however, when such monitoring implementations are not available, a conservative approach would be to insist on colocating Route Servers on an Ethernet.

We concluded the session with a discussion of the pros and cons of separating the role of Route Server from the role of Border Gateway. We noted that MIRA *allows* the two to be implemented within the same machine. Doing so in fact simplifies various RS-to-BG communications. It is crucial, however, to also allow the two to be separated:

- This will allow network engineers to continue to use existing Border Gateways and still move to MIRA with separate RSes.
- It will reduce the computational burden on the Border Gateways by doing Inter-AS routing functions in another computer.
- It will allow network engineers to choose among vendors for RS implementations. (In the current environment, users are 'captive' to gateway vendors; we should try to reduce the extent of this.)
- A vendor can add RS functionality to its gateway product on a schedule of the vendor's choosing; its customers can use separate RSes during the meantime.
- All network engineers could support MIRA/BGP with separate RSes during a period of time in which integrated RS/BG implementations were being built.

Wednesday morning we focused on the dynamics of changing Inter-AS routes. A near-worst-case occurs when AS1 functions as a transit AS for a given destination N. AS1 uses AS2 as its next AS in routing to N, and advertises this path to AS3. AS3, however, uses a path via AS5, and AS1 sees AS3 advertising this path. Now, due to a break in the direct AS1-to-AS2 link, AS1 wants to use AS3 as its next hop. Before it can do so, two things must happen:

- AS1's neighbors must learn that AS1's old path is no longer valid. (Otherwise a loop can form.)
- The Interior Gateways of AS1 must learn that a Border Gateway to AS3 is its path to N rather than the Border Gateway to AS2.

During the time when these two things are happening, routing to N will be very difficult, and dropped packets may occur. Careful sequencing of actions must take place in these and similar cases.

A second issue was to decide on Shortest-AS-Path-Length and Static Preferences as the methods of deciding which of several alternate AS Paths to use. MIRA/BGP allows for future more sophisticated techniques, but we will wait a while to push these techniques.

Internet User Population Working Group
Chairperson: Craig Partridge/BBN

CHARTER

Description of Working Group

To devise and carry out an experiment to estimate the size of the Internet user population.

Specific Objectives:

We expect to produce two documents: (1) a description of the experimental procedure and (2) an RFC that gives the results of the experiment. We may also produce a short paper for publication in a networking magazine.

Estimated Timeframe for Completion:

The firm hope is that this will only take two meetings: Hawaii to determine the experimental design and then the next meeting to report the results.

STATUS UPDATE

1. Chairperson(s): Craig Partridge, craig@nnsc.nsf.net
2. WG Mailing List: ietf@venera.isi.edu (interim address)
3. Date of Last Meeting: Just formed
4. Date of Next Meeting: Hawaii IETF, 31 Oct - 3 Nov 89
5. Pending or New Objectives: Undertake description of proposed experimental procedure.
6. Progress to Date (e.g., documents produced): First meeting will be held in Hawaii at the 31 Oct - 3 Nov 89 quarterly meeting.

JOMANN Working Group
Chairperson: Susan Hares/MERIT

CHARTER

Description of Working Group:

> This "Joint Monitoring Access for Adjacent Networks focusing on the NSFNET Community" Working Group will:
>
> - discuss how to identify problems in the next hop network
> - create a list of existing tools which can solve these problems (We will discuss to see if NOC-Tools Working Group can take over this. NSFNET will archive a list of these tools.)
> - create a list of routing topology maps of regionals (possibly prepare a MAP Internet-Draft)

Specific Objectives:

> See above

Estimated Timeframe for Completion:

> 6-9 months (extended to November 1, 1989)

STATUS UPDATE

1. Chairperson: Susan Hares/Merit, skh@merit.edu
2. WG Mailing List:
    - njm@merit.edu (Regional or National Net NOC people)
    - njm-interest@merit.edu (anyone interested)
    - njm-request@merit.edu
3. Date of Last Meeting: July 25-28, 1989
4. Date of Next Meeting: October 31 - 1 November, 1989 University of Hawaii (tentative)
5. Pending or New Objectives: to be determined
6. Progress to Date (e.g., documents produced): See Current Meeting Report

WHO SHOULD ATTEND:

> Technical representatives from mid-level or peer networks. In the future we may want to extend this to technical representatives from campus networks. However, in interest of getting a lot of work done quickly the initial working group will be limited.

JOMANN Working Group
Chairperson: Susan Hares/Merit

CURRENT MEETING REPORT

Reported by Elise Gerich/Merit and Dale Finkelson/Midnet

AGENDA

- Community Names
- Network Maps
- Mailing Lists
- NSFNET <--> ARPAnet Routing
- Misbehaving Hosts
- Outage Reports

ATTENDEES

- Guy Almes
- Phil Almquist
- Scott Brim
- Jeffrey Burgan
- Joe Choy
- Tom Easterday
- Mark Fedor
- Gene Hastings
- Jeff Honig
- Paul Love
- Matt Mathis
- Milo Medin
- Don Morris
- Rebecca Nitzan
- Jacob Rekhter
- Ron Roberts
- Vince Fuller
- Joel Replogle
- Marty Schoffstall
- Dale Finkelson
- Ross Veach
- Dave Katz
- Dan Jordt
- Jim Sheridan
- Elise Gerich

MINUTES

The JOMANN Working Group met on Wednesday, July 26, 1989. Gene Hastings of PSC and Elise Gerich of Merit acted as co-chairs in the absence of Susan Hares, chairperson.

Community Names:

Common community names exist for SGMP, but not for SNMP. Some groups have a concern about announcing a public community name. The options that were considered were:

1. Live with public community names
2. Use multiple community names
    - one for NSF
    - one for regionals

   This course opens the communities to the campuses.
3. Develop a tool that uses multiple community names

The consensus of the group was:

1. To set the NSFNET community name in all gateways not accessible to the campuses
2. That regional gateways will use monitor for now
3. To see if other backbones will use same community name.

The question was raised whether anyone had a concern about tools that deduce topology. No one who was at the meeting expressed a concern.

Maps:

Merit continues to collect and store maps on nis.nsf.net. Please send maps to Merit via device independent electronic form. Gene Hastings volunteered to take PICT files and convert them if anyone needed help. Send to Gene at hastings@morgul.psc.edu.

Mailing Lists:

The current mailing lists are: nwg
nsfnet-site-people
regional-techs

nsfnet-site-people has become a general list that is freely distributed. Since it no longer pertains only to NSFNET the name of this list has been changed to network-status-reports@merit.edu and may be exploded to all campuses. nsfnet-site-people will disappear.

nwg should be used for network discussions. Some concern was expressed that not all regionals are on nwg.

regional-techs is a new list for the nsfnet regional technical representatives. This should not be expanded to redistribution lists, or at least only to administrative entities for the regionals.

NSFNET <--> ARPAnet

The question arose as to how far along the path of the implementation is NSFNET. The plan is generally known, but its status is not.

The regionals would like to know when various links to the Milnet/ARPAnet are down so that they can respond to their users problems. Merit will attempt to announce the status of the connection to the Milnet/ARPAnet via network-status-reports.

There was some discussion as to secondary and tertiary paths to the ARPA/Milnet, with the opinion expressed by some that these routes do not always help.

Agreement seemed to be reached that if the Ames mailbridge is up connectivity is quite good, but if it is down, connectivity is not very good. The group would like to be notified via network- status-reports that Ames' mailbridge is down.

Discussion then centered on where regionals could get information on how nets are announced to the ARPAnet/Milnet. Config files are on nis.nsf.net, in the nsfsites directory, and the file name is net-comp.now. (Note: There is another directory which is password protected which lists the machine readable configuration files for each nss. If you are interested in reviewing these files, call Merit for the password.) Various other policy database files exist in this directory.

Milo Medin mentioned that on a good day the mailbridge runs about 4.5 million packets of an average of 110 bytes.

Misbehaving Hosts:

Both PSC and Merit have been noticing overly agressive domain requests, and a discussion ensued as to what actions if any would be reasonable to take. No one objected to letting people know that problems exist, and some network administrators requested that they be notified in addition to the host administrator. It was suggested that vendors should be addressed as well as individual offenders.

The meeting drew to a close without the opportunity to address the agenda item concerning outage reports. It was agreed that this could be discussed electronically.

A special thanks to Dale Finkelson for agreeing to keep the minutes for this meeting.

Jo-Mann WG

Common community names
- More than 1
- End to end debugging

MAPS!

Call for maps
Online at NIS.NSF.NET
PICT Conversion
hastings@morgul.psc.edu

Mailing Lists

NWG@merit.edu
nsfnet-site-people changing to network-status-reports
<nsr@merit.edu>
– very public,
Call your regional.

28 July 1989

NSFNET <.> MILNET/ARPANET

Connection to Ames MB
in service
Mitre connection
"in a few weeks"

(On a good day, Ames passes
4.5M packets, avg. 110 bytes

Misbehaving hosts

- NSFNET Backbone Stats
show Telnet #1,
Domain #2 or 3 ----
Merit collecting net and
host pairs for review,
host administrators may
get mail soon (speeding
tickets)

Tools:

LSR Traceroute.
further discussion deferred to email.

28 July 1989

CHARTER

Description of Working Group:

To define the MIB (and relevant related mechanisms) needed to allow management overlap between the workgroup environment (LAN Manager based) and the enterprise environment (based on TCP/IP management).

Specific Objectives:

This translates into four basic areas:

- Define a set of management information out of the existing LAN Manager objects to allow for useful management from a TCP/IP based manager.
- Define extensions to the TCP/SMI when appropriate. Develop requirements for additional network management information, as needed, and work to extend the LAN Manager interfaces to support such information.
- Define the mechanisms of exchange of management information between clients and servers so that proxies can be developed.

Estimated Timeframe for Completion:

STATUS UPDATE

1. Chairperson: To be determined
2. WG Mailing List: lanmanwg@spam.istc.sri.com
3. Date of Last Meeting: Cocoa Beach/April 1989
4. Date of Next Meeting: To Be Determined
5. Pending or New Objectives:
6. Progress to Date (e.g., documents produced):
   Documents submitted to RFC editor for RFC consideration:

   LANMAN-MIB
   LANMAN-MIB-EXPER

CURRENT MEETING REPORT

Did not meet during this quarter's IETF plenary.

NISI Working Group
Chairpersons: Karen Bowers/NRI and Phill Gross/NRI

CHARTER

Description of Working Group:

The NISI WG will explore the requirements for common, shared internet-wide network information services. The goal is to develop an understanding for what is required to implement an information services "infrastructure" for the Internet. This effort will be a sub-group of the User Services WG and will coordinate closely with other IAB and FRICC efforts in the area of Directory Services.

Specific Objectives:

1. Write a short white paper to serve as a starting point for discussions on an Internet-wide information services infrastructure.
2. Develop a more detailed statement of required information services as currently supplied by a typical network information service organization. This will initially take the form of an annotated outline of services, suitable to be expanded into a full Requirements Document.
3. Define candidate pilot projects for consideration by this or other groups to implement. Initial candidates include:
    - Define common user interface for information retrieval by electronic mail.
    - Define common user interfaces for other information services (e.g., white pages)
    - Define the minimally required information content for an Internet-wide user registration database and begin to collect such information.

Estimated Timeframe for Completion:

- Objective 1 – First draft outline to be distributed by email to the USWG mailing list by June 30, 1989.
- Objective 2 – Annotated outline, ready for volunteer writing assignments, to be distributed by email to the USWG and the NISI mailing lists by June 30, 1989
- Objective 3 – To be determined

STATUS UPDATE

1. Chairpersons: Karen L. Bowers / kbowers@nri.reston.va
   Phill Gross / pgross@nri.reston.va.us
2. WG mailing list: NISI@MERIT.EDU
3. Date of Last Meeting: 25-28 July 1989
4. Date of Next Meeting: Interim in September / U of Hawaii, 31 Oct - 3 Nov 1989
5. Pending Objectives:
   Two Immediate Actions:
    - Network Information Services Requirements Document
    - White Paper: Network Information Services Infrastructure (NISI)
6. Progress to Date (e.g., documents produced):

   The first draft outline for the requirements document was prepared. Review of and adjustments to the outline were undertaken during the IETF plenary. Due to time constraints the review was not completed and will be continued on the mailing list and at the next meeting. Plans are to hold an interim meeting in September.

NISI Working Group
Chairpersons: Karen Bowers/NRI and Phill Gross/NRI

CURRENT MEETING REPORT

Reported by Karen Bowers

AGENDA

- Review Charter and Objectives
- Examine What is Required to Implement an Information Services "Infrastructure" for the Internet
- Review the NIS Requirements Document 1st Draft Outline
- "Recruit" Interested Members for Specific Writing Assignments

ATTENDEES

1. Bowers, Karen/kbowers@nri.reston.va.us
2. Breeden, Laura/breeden@bbn.com
3. Droms, Ralph/rdroms@nri.reston.va.us
4. Garcia-Luna, J. J./garcia@sri.com
5. Gross, Phill/pgross@nri.reston.va.us
6. Moore, Berlin/bm24@andrew.cmu.edu
7. Roberts, Mike/roberts@educom.edu
8. Roubicek, Karen/roubicek@nnsc.nsf.net
9. Sollins, Karen/sollins@lcs.mit.edu
10. Stahl, Mary/stahl@sri-nic.arpa
11. Sweeton, Jim/sweeton@merit.edu
12. Yuan, Aileen/aileen@gateway.mitre.org

MINUTES

The second session of the NISI WG convened for two hours on 26 July 1989 at Stanford University. The meeting opened with a quick review of the charter and objectives. This was followed by a recapitulation of the 5 basic ingredients (previously identified) necessary to implement an information services "infrastructure" for the Internet and related projects/issues:

1. Set of commonly defined Information Services supported by NICs (Related project: define starting set of common Information Services)
2. Commonly defined User Interfaces to 1. (Related project: As a pilot project, define common user interface for email document retrieval)
3. Commonly defined Administrative Support for 1. (Related project: Define collection tools for administering a Whois White Pages Service)
4. Common Service Interfaces between NICs or special interfaces, as required, for 1. (Related project: Define common or special interfaces required for different NIC White Pages Services to interoperate (X.500?))
5. Compatible policies and co-operative agreements between NICs

The remainder of the time was spent discussing what the scope and approach of the Requirements Document should be and reviewing the first draft outline. The following points were made with respect to the scope and approach:

- This requirements document must be written with other relevant documents in mind: the Program Plan for NREN, the EDUCOM NTTF report (The National Research and Education Network: A Policy Paper) and the FCCSET report released by the Office of Science and Technology Policy (and one mentioned at the first meeting, A Plan for Internet Directory Services). With respect to the Program Plan for NREN, attention must be paid to the NREN model of NICs and their future functions. Action Item: K. Bowers to secure permission to release the NREN document to the NISI members and, once permission is secured, to distribute the most recent version to the NISI WG membership.
- This document should take the user view of Network Information Services - "what every user should have access to". Specifically, the document should address what information services currently exist, those currently under development, and future requirements foreseen.
- Furthermore, this document must examine the delivery mechanism (technology) and "who" should do the delivery.
- Finally, emphasis must be placed on automated information delivery.

As the review of the draft outline was undertaken, it became very apparent that care must be taken to include the assumptions on which the document is based and full definitions of ambiguous terms such as "information services", "user services", etc. The limited time remaining for reviewing the draft outline was spent discussing types of on-line and off-line assistance and what they should be called.

Changes to the draft outline will be posted to the NISI mailing list shortly. A follow-on meeting is planned for September and will be scheduled and announced shortly. Two hours of time only allowed us to begin work on the outline; at least one full day must be secured for NISI activities at the next IETF plenary.

Related Issues:

- Nysernet is doing a pilot project on commercial white pages.
- We must be cognizant of commercial or proprietary constraints on "compatible policies and co-operative agreements between NICs".
- What are the timelines for the "infrastructure"?
- Should the User Services Area have the full goal of defining, guiding or doing the Network Information Services Infrastructure? How do the other IETF WGs come into play?

NOC-Tools Working Group
Chairpersons: Robert Enger/Contel and Robert Stine/Sparta

CHARTER

Description of Working Group:

The NOC-Tools Working Group will develop a catalog to assist network managers in the selection and acquisition of diagnostic and analytic tools for TCP/IP Internets.

Specific Objectives:

1. Identify tools available to assist network managers in debugging and maintaining their networks.
2. Publish a reference document listing what tools are available, what they do, and where they can be obtained.
3. Arrange for the central (or multi-point) archiving of these tools in order to increase their availability.
4. Establish procedures to ensure the ongoing maintenance of the reference and the archive, and identify an organization willing to do it.
5. Identify the need for new or improved tools as may become apparent during the compilation of the reference document.

Estimated Timeframe for Completion:

The first edition of the catalog will be submitted for final review at the October-November IETF meeting. Preliminary versions will be made available earlier.

STATUS UPDATE

1. Chairpersons: Robert Enger, enger@sccgate.scc.com
   Robert Stine, stine@sparta.com
2. WG Mailing List: noctools@merit.edu
3. Date of Last Meeting: July 25-28, 1989 / Stanford U
4. Date of Next Meeting: October 31 - November 1, 1989 University of Hawaii
5. Pending or New Objectives:

   Draft ready for first publication on or about December 12, 1989
6. Progress to Date (e.g., documents produced):

   First draft in progress

NOC-Tools Working Group
Chairpersons: Robert Stine/Sparta and Robert Enger/Contel

CURRENT MEETING REPORT

Reported by Robert Enger and Robert Stine

ATTENDEES

1. Armstrong, Karen/armstrongk@sds.sdsc.edu
2. Auerbach, Karl/karl@asylum.sf.ca.us
3. Bierbaum, Neal/vitam6!bierbaum@vitam6
4. Bowers, Karen/kbowers@nri.reston.va.us
5. Brunner, Eric/brunner@monet.berkeley.edu
6. Carter, Glen/gcarter@ddn1.dca.mil
7. Crocker, Dave/dcrocker@ahwahnee.stanford.edu
8. Deboo, Farokh/fjd@bridge2.esd.3com.com
9. Easterday, Tom/tom@nisca.ircc.ohio-state.edu
10. Enger, Robert M./enger@sccgate.scc.com
11. Finkelson, Dale/dmf@westie.unl.edu
12. Hastings, Gene/hastings@morgul.psc.edu
13. Hunter, Steven/hunter@ccc.mfecc.llnl.gov
14. Karn, Phil/karn@thumper.bellcore.com
15. Malkin, Gary/gmalkin@proteon.com
16. Mathis, Matt/mathis@fornax.ece.cmu.edu
17. Miller, Stephen/miller@m2c.org
18. Moore, Berlin/bm24@andrew.cmu.edu
19. Morris, Don/morris@ncar.ucar.edu
20. Nitzan, Rebecca/nitzan@ccc.mfecc.llnl.gov
21. Oattes, Lee/oattes@utcs.utoronto.ca
22. Pleasant, Mel/pleasant@rutgers.edu
23. Pugh, Jon/pugh@nmfecc.llnl.gov
24. Replogle, Joel/replogle@ncsa.uiuc.edu
25. Roberts, Ronald/roberts@jessica.stanford.edu
26. Salo, Tim/tjs@msc.umn.edu
27. Sheridan, Jim/jsherida@ibm.com
28. St. Johns, Michael/stjohns@beast.ddn.mil
29. Stahl, Mary/stahl@sri-nic.arpa
30. Steinberg, Lou/louiss@ibm.com
31. Stine, Robert/stine@sparta.com
32. Streeter, Roxanne/streeter@nsipo.arc.nasa.gov
33. Veach, Ross/rrv@seka.cso.uiuc.edu
34. Waldbusser, Steve/sw01@andrew.cmu.edu
35. Yuan, Aileen/aileen@gateway.mitre.org

MINUTES

The July session was the second formal meeting of the NOC-Tools WG. We received a number of additional suggestions for tools that should be included in the catalog, as well as suggestions on matters of procedure and distribution. Things seem to be picking up.

The following were suggested as means to ferret out existence of additional tools:

- utilize the ACE mailing list in some capacity
- consult the "DATA PRO" reference
- consult the Data Communications Guide
- examine Nysernet's public software repository
- post a request to the following mailing lists:
  - IAB
  - JOMANN
  - CERT
  - DDN Management Bulletin
  - NameDroppers
  - CERF-OPS
  - NWG
  - NSFnet regional techs
  - SNMP
  - GATED
  - BYTE-COUNTERS

It was suggested that the preparation and distribution issues be coordinated with USER-DOC WG, since they are also facing them. In particular, how can we publicize the existance of the catalog? It was suggested that we ask that our catalog be referenced within the New Users Guide and the Vendors Guide distributed by SRI-NIC. However, the general question of how to disseminate information to the user and user-helper community is being deferred to the entire USWG.

The following tool/category suggestions were voiced at the recent meeting:

- TWG and HP Ethernet monitors
- BBN's ANM product
- XLAN from Rutgers
- SNMP Topology tools
- Nysernet's SNMP Lookup tool
- Point-to-Point monitors
- Throughput measurement tools
- Modified traceroute that generates a topology map
- Data reduction scripts (generate statistics of various sorts)
- KA9Q HopCheck program
- Third party traceroute
- Traceroute for the Mac (listens to routing info, makes a map)
- High level (applications) debugging. (telnet negotiation, etc) (it was suggested that we should not do Conformance testing)
- The Sniffer
- Promiscuous Expert System network monitor
- Karn's Bogon Trap
- Karn's Arp monitor
- Karn's Extended Lan Station Locator (LanBridge-100, etc)
- Eon LanProbe

It was suggested that all attendees be added to the NOC-Tools mailing list.

It was suggested that NOC-Tools set up an anonymous FTP directory.

A flyer was given out to the attendees of the Thursday morning plenary. It asked the attending network gurus to give us their suggestions for tools that should be included in the catalog, as well as referrals to other gurus who we should contact. We received a modest response.

NOC-Tools WG

1) CHARTER:

The NOC-Tools Working Group will develop a catalog to assist network managers in the selection and acquisition of diagnostic and analytic tools for TCP/IP Internets.

2) Objectives:

A) Identify tools available to assist network managers in debugging and maintaining their networks.

B) Publish a reference document listing what tools are available, what they do, and where they can be obtained.

C) Arrange for the central (or multi-point) archiving of these tools in order to increase their availability.

D) Establish procedures to ensure the ongoing maintenance of the reference and the archive, and identify an organization willing to do it.

E) Identify the need for new or improved tools as may become apparent during the compilation of the reference document.

3) Timeframe:

The first edition of the catalog will be submitted for final review at the October-November IETF meeting. Preliminary versions will be made available earlier.

NOCTools WG Meeting

Wednesday PM, 26 July

Progress in soliciting entries—

- Request emailed to TCP-IP, IETF, BIGLAN.
- Individual requests to old boys
- Requests to publications/trade rags.

Future solicitation—

- NWG, SNMP, Namedroppers, Cerf-ops, FARNET, IAB, byte counters.
- Individual requests to vendors.
- Net manager/System administrator interviews.

NOC Tools

Progress in compiling catalog –

Before meeting,
- 28 known tools,
- 17 draft entries.

During meeting –
- ~11 more tools identified,
- ~3 volunteers to compose entries.

NOC Tools

Goals

Polished draft by next IETF
- Tool descriptions
- Short tutorial on practical network management.

Procedures established for document maintenance, dissemination

NOCTools

We need:

- Tool descriptions!
  If useful & available,
  we want it!
- Pointers to tools
- Pointers to pointers

Send to:

noctools@merit.edu
(requests to join to
noctools-request)

Open SPF-based IGP Working Group
Chairpersons: Mike Petry/UMD and John Moy/Proteon

CHARTER

Description of Working Group:

The OSPF working group will develop and field test an SPF-based Internal Gateway Protocol. The specification will be published and written in such a way so as to encourage muliple vendor implementations.

Specific Objectives:

- Design the routing protocol, and write its specification.
- Develop multiple implementations, and test against each other.
- Obtain performance data for the protocol.
- Make changes to the specification (if necessary) and publish the protocol as an RFC

Estimated Timeframe for Completion:

We have a complete protocol specification. Implementation experience and performance data should be obtained during the summer of 1989. The specification should be ready for final review by the October-November IETF.

STATUS UPDATE

1. Chairpersons: Mike Petry, petry@trantor.umd.edu
   John Moy, jmoy@proteon.com
2. WG Mailing List(s): ospfigp@trantor.umd.edu
3. Date of Last Meeting: July 1989, Stanford University
4. Date of Next Meeting: October 31 - November 3, 1989 Hawaii IETF
5. Pending or New Objectives: see charter
6. Progress to Date (e.g., documents produced):
   - The OSPF Specification, first revision (1/89)
   - First revision of the OSPF specification finished (1/89)
   - OSPF presentation given during IETF plenary (4/89)
   - Two working OSPF implementations
   - The OSPF specification is now an Internet Draft

Open SPF-based IGP Working Group
Chairpersons: Mike Petry/UMD and John Moy/Proteon

CURRENT MEETING REPORT

Reported by Rob Coltun

AGENDA

The OSPFIGP working group met for a half day on July 28th at Stanford. The agenda was as follows:

- Implementations
- Spec Changes
- Net Management Items
- What's Next

ATTENDEES

1. Baker, Fred/baker@vitalink.com
2. Bierbaum, Neal/bierbaum@vitalink.com
3. Blackwood, Craig/craig@hprnd.rose.hp.com
4. Blumenthal, Steve/blumenthal@bbn.com
5. Coltun, Rob/rcoltun@trantor.umd.edu
6. Deboo, Farokh/fjd@bridge2.3com.com
7. Deering, Steve/deering@pescadero.stanford.edu
8. Doo, Way-Chi/wcd@bridge2.esd.3com.com
9. Farinacci, Dino/dino@bridge2.3com.com
10. Fuller, Vince/vaf@jessica.stanford.edu
11. Honig, Jeffrey/jch@sonne.tn.cornell.edu
12. Hytry, Tom/tlh@iwlcs.att.com
13. Ilnicki, Ski/ski
14. Jones, Bill/jones@nsipo.nasa.gov
15. Jordt, Dan/danj@cac.washington.edu
16. Karn, Phil/karn@thumper.bellcore.com
17. Medin, Milo/medin@nsipo.nasa.gov
18. Moy, John/jmoy@proteon.com
19. Oattes, Lee/oattes@utcs.utoronto.ca
20. Oran, David/oran@oran.dec.com
21. Petry, Mike/petry@trantor.umd.edu
22. Pugh, Rex/pugh@hprnd.rose.hp.com
23. Reilly, Michael/reilly@atari.nac.dec.com item Smith, Tom/toms@hprnd.rose.hp.com
24. St. Johns, Mike/stjohns@beast.ddn.mil
25. Stone, Geof/geof@network.com
26. Veach, Ross/rrv@seka.cso.uiuc.edu

MINUTES

1. Milo Medin presented John Moy's slides of the Proteon OSPF implementation in John's stead.
2. Rob Coltun presented slides of UMD's BSD OSPF implementation.
3. During the presentations questions came up concerning the migration from RIP to OSPF. It was agreed that the best way to approach this is to initially put OSPF at the borders of the Autonomous System and work towards the center. We are considering writing a paper addressing this issue.
4. Questions also came up regarding running OSPF on BSD systems that currently do not support Multicast. We explained that the BSD version allows configuring non-broadcast, multi-access networks over broadcast networks to serve as a temporary fix for those systems that do not support Multicast; this option will remain in the implementation until Multicast is generally available.
5. There were discussions on conditions that may occur in very large Autonomous Systems such as how to handle routing updates when a gateway has run out of memory and on possible consequences of running the spf algorithm each time a new LSA arrives when a partition has been repaired.
6. We talked about some of the changes to the specification such as packet formats, the Fletcher checksum, hello packets on non-broadcast multi-access networks, and hello packets over point-to-point links.
7. We presented some possible MIB variables. These basically were state, database and packet summary information. Not much discussion.
8. Some discussion followed about the OSPF working group being close to (or at) the end. We talked about starting a Multicast routing working group.

Agenda

IMPLEMENTATIONS

SPEC CHANGES

NET MANAGEMENT ITEMS

MODELING PARAMETERS

WHAT'S NEXT

UMD's OSPF IMPLEMENTATION

(the Peoples' Implemetation)

- Overview
- Configuration
- Some Implementation Details

---

umd <> ospf

OVERVIEW

Runs on Uvax

- 4.3 BSD
- Ultrix 3.1

OSPF Version 1

- Needs To Be Upgraded To Version 1

Talks to PROTEON

- Establish Full Adjacency, Exchange LSAs, etc.

7500 Lines of C

- Including Comments and .h

TESTING

Most Testing on Ethernet

- Fake NBMA and Point-to-Point

Simulated Larger AS To Test Routing Table Calculations

- 16 Nodes (Gateways), 17 Nets
  - Change Source Node
  - Change Link Costs
  - LSA Age > MaxAge
  - Bad Backlinks
  - No Net or RTR LSA

CONFIGURATION

Configuration Parameters Read from File

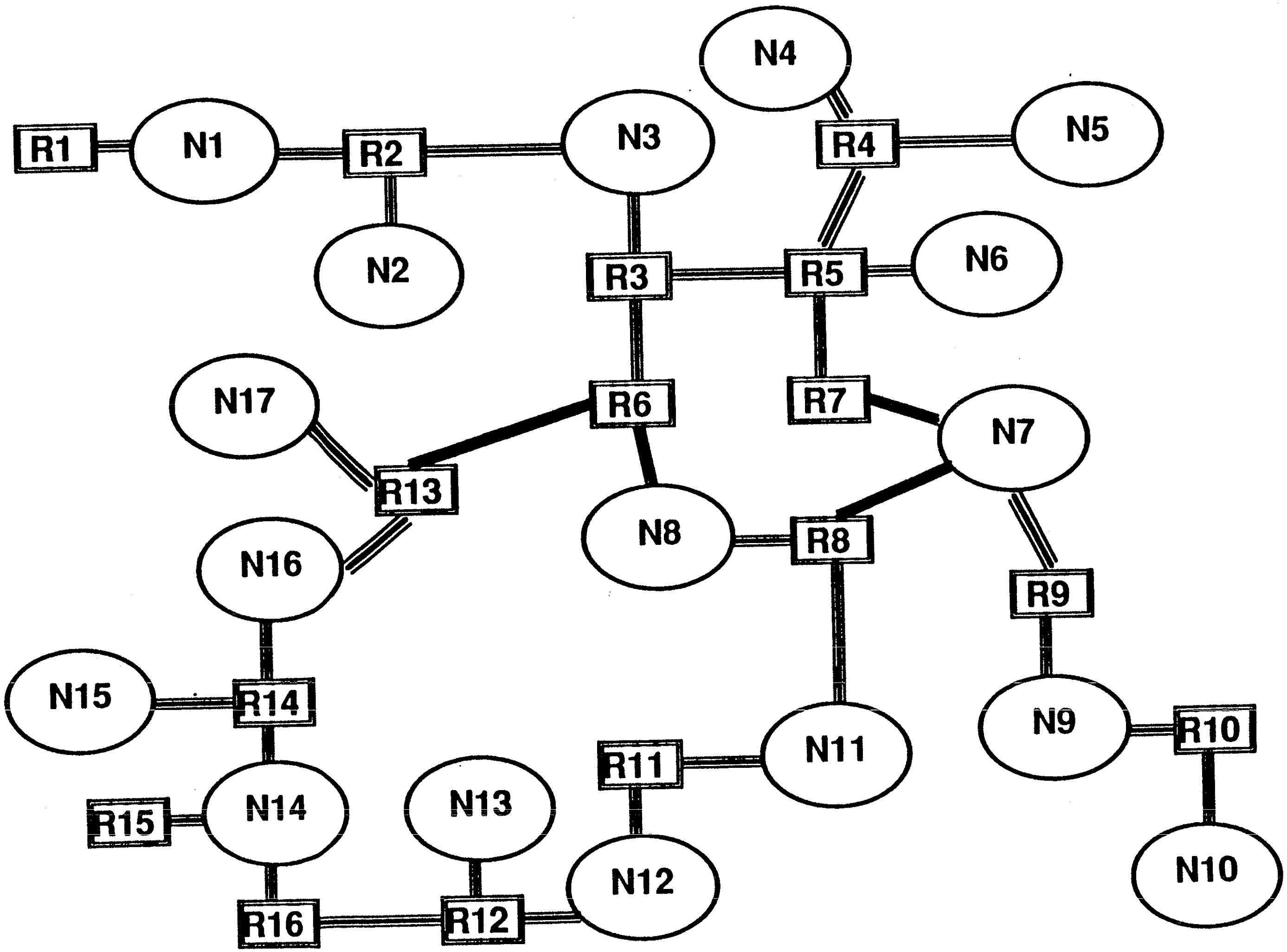
- Plan To Add Interactive Configuration

Each OSPF Interface Is Bound To UNIX Interface

External Routes Are Configured

```
##################################################################################
#                                                                                #
# INTERFACE DEFINITION                                                           #
# Note: the area associated with each interface must be defined                  #
#        before the interface is; also all intervals are in seconds              #
#                                                                                #
# SINTAX -                                                                       #
#       IF: <Associated area id> <IP addr> <IP mask> <type> <cost0> <cost1>      #
#           <cost2> <cost3> <rxmt interval> <trans delay> <priority>             #
#           <Hello interval> <rtr dead interval>                                 #
#           <Auth key 1st 32 bits> <Auth key 2nd 32 bits>                        #
#       type could be 1 = broadcast, 2 = nonbroadcast multi access or            #
#               3 = point to point                                               #
#                                                                                #
##################################################################################
IF: 0   '128.8.10.14     FFFFFF00        2       20      0       0       0
5       20      14      20      80

IF: 0   128.8.2.127     FFFFFF00        2       30      0       0       0
5       20      127     20      80
```

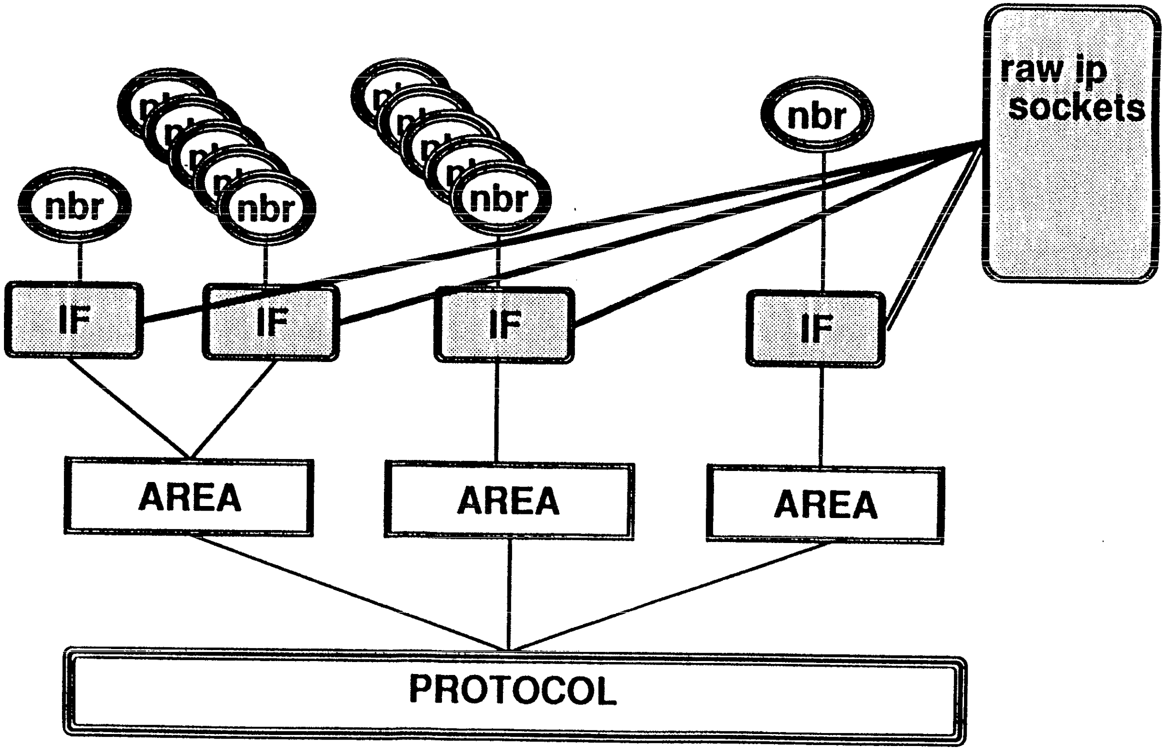
umd <> ospf


N4
R1
N1
R2
N3
R4
N5
N2
R3
R5
N6
N17
R6
R7
N7
R13
N8
R8
N16
R9
N15
R14
N11
N9
R10
R11
N14
R15
N13
N12
N10
R16
R12

IMPLEMENTATION DETAILS

- GENERAL STRUCTURES
- TIMERS
- STATE MACHINE
- LINK-STATE DATABASE
- ROUTING TABLE UPDATES
- LOGGING

GENERAL STRUCTURES



raw ip sockets

nbr nbr nbr nbr

IF IF IF IF

AREA AREA AREA

PROTOCOL

TIMERS

```
struct TIMER {
        struct TIMER *next;
        u_char type     /* Type of alarm */
        u_long key;     /* Ptr to INTF, NBR or AREA */
        struct NBR *nbr;
        struct INTF *intf;
        struct AREA *area;
        int (*routine)();
        struct itimerval set;
        struct itimerval reset;   /* 0 if one shot */
};
```

- Timers kept in priority queue
- Struct passed as routine param

TIMER TYPES

- Interface Check - per interface
- Hello Interval - per interface
- Wait One Shot - per multi-access interface
  - Elect DR
- Inactivity Timer - per interface
  - Declare Nbr Down
- Hold D-D One Shot - per nbr
  - Hold Last D-D pkt when slave
- Retrans Interval - per interface
  - Checks Queue Depending On State
    - Db Sum
    - LS Req
    - LS Adv

TIMER TYPES (cont'd)

- LSA Age Interval - global
  - Set To CheckAge - Every 3x Check For FlushAge
- LSA Interval Send - global
- Lock For RTR, NET, SUM and ASE
  - Set To MinLSInterval
- Delayed Ack Interval Send

State Machine

State Space is Small

- 9 Nbr States x 14 Nbr Events
- 8 Intf States x 7 Intf Events

Array of Routines

- (*nbr_trans)[NNBR_STATES][NNBR_EVENTS]
- (*intf_trans)[NINTF_STATES][NINTF_EVENTS]
- e.g., (*nbr_trans)[nbr->state][ADJ_OK]

Link-State Database

LSDB STRUCT IS KEY

- LSDB Holds Link-State Advertisements
  - Hash - List Stored In Order
  - Advertisements Are Time Stamped When They Arrive
    - No Need To Age LSDB
  - LSDB Per Area - External LSA In Protocol Struct
- LSDB Contains List Of Nbrs That Contain This LSDB In Their Retrans List
  - Nbr Contains List Of LSDBs To Retrans
  - Easy To Keep Track Of When LSDB Is Changed
    - New LSDB Has Arrived
    - Old LSDB Has Reached FlushAge

Link-State Database (cont'd)

- LSDB Struct Is Used As Vertex In Spf Algorithm
- LSDB Struct Has Ptr To Routing Table Entry Relating To Its LSA

Routing Table Updates

- Kernel Has Routing Table So OSPF Has To Keep Track Of What Has Changed
  - List Of Routing Table Entries Is Moved From Current To Tmp List
  - New Spf Algorithm Is Run And Routing Table Rebuilt
  - As *addroute(vertex,area)* Is called, a flag is set when Distance Or First Hop Is Changed
  - *addroute()* Puts Routing Entry on Current List (removing entry from tmp list if entry exists)
  - Tmp List Contains What Has To Be Deleted
  - Current List Contains What Is In Routing Table and What Has To Be Changed

LOGGING

Current Version Sends Log Msgs To Files

- Intf and Nbr State Trans Logs
- Rx and Tx Log By Packet Type
- Error Log
  - Packet Type and Error
  - Unusual Events
- Cumulative Log Of Rx, Tx and Errors
- Packet Dump (by type with timestamp)
- Routing Table Dump

p4200 OSPF IMPLEMENTATION

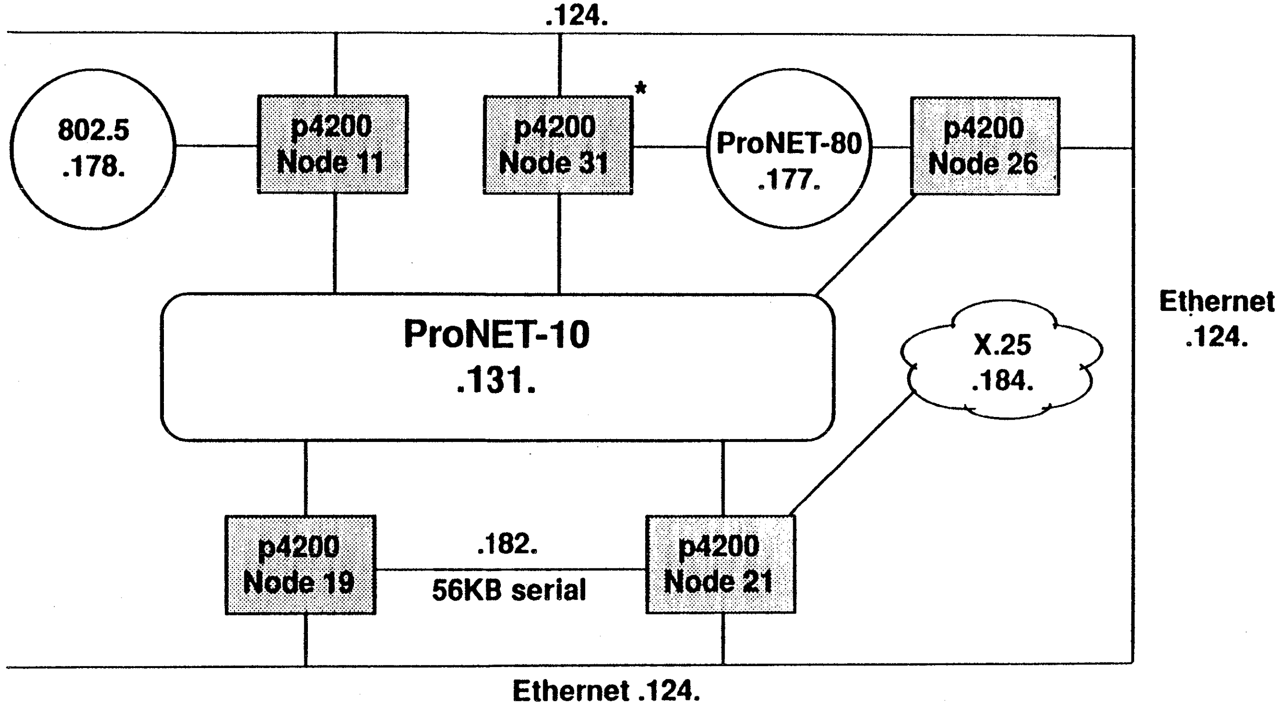
- Three components:
  - Configuration Console
  - Monitoring Console
  - Logging messages

IMPLEMENTATION STATISTICS

- 8,000 lines of C
  - comments and .h files included
  - EGP is 3,500 lines
  - RIP is 1,200 lines
- Trial implementation in Release 8.1A
  - available now
- Complete implementation in Release 8.2

proteon

OSPF TESTBED



* ASBR router

OSPF CONFIGURATION ITEMS

- Per interface:
  - OSPF enable/disable
  - interface cost
  - subnet mask
- Area groupings
- Virtual links, if needed
- Non-broadcast neighbors

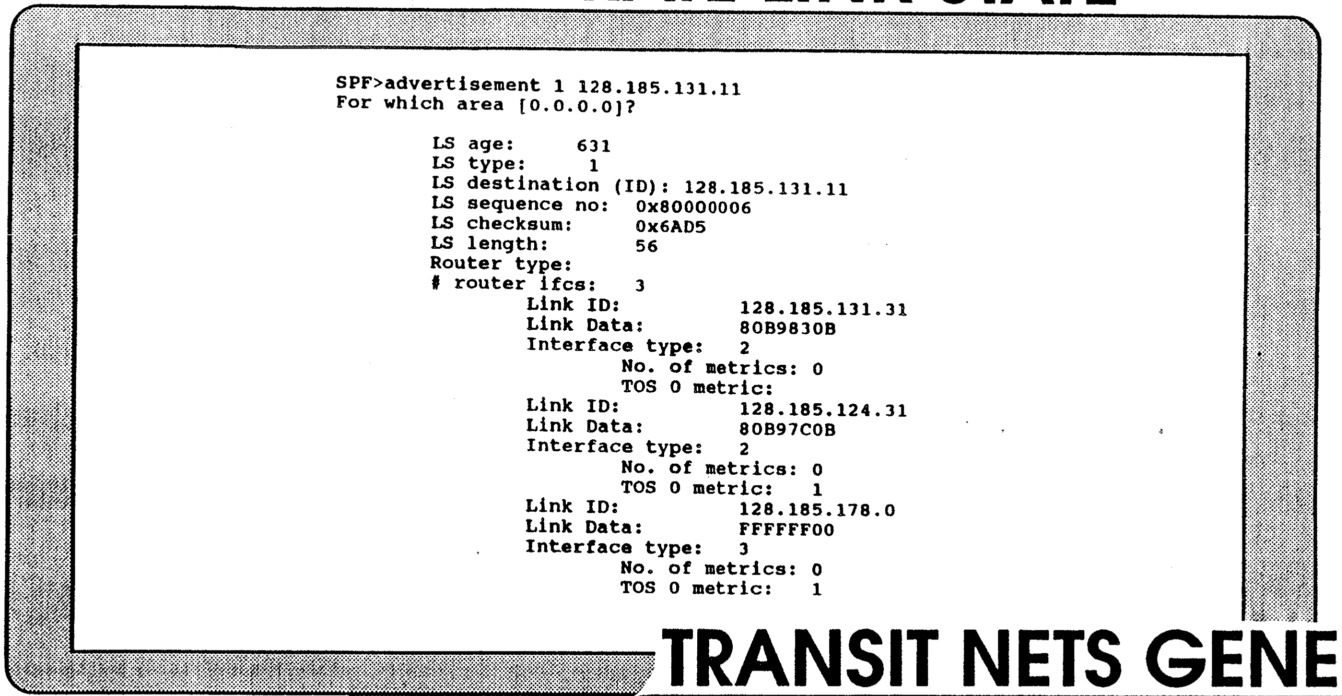
RIP TRANSITION

- RIP routes can be imported to OSPF
  - configurable
  - always as Type 1 external
- Internal OSPF routes advertised by RIP
  - with hop count 1
- Allows gradual conversion to OSPF

MONITORING DISPLAYS

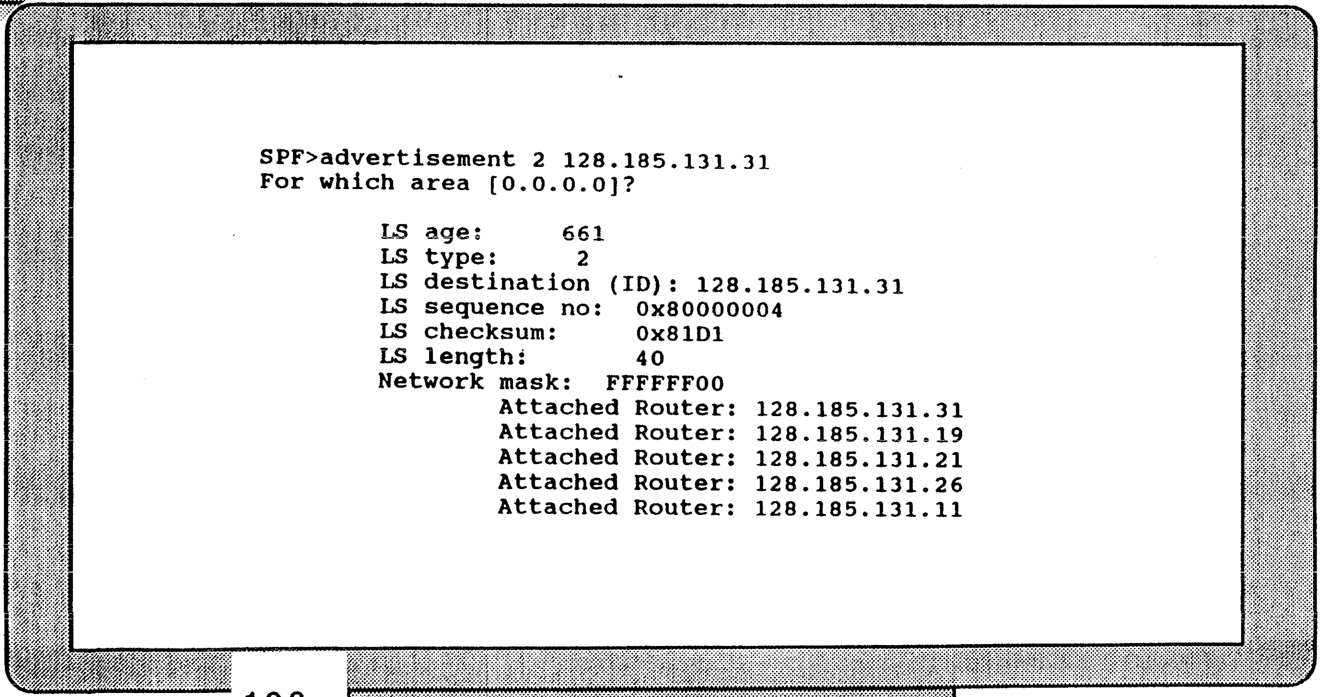
- Taken from node 19
- Node 31 importing external routes
  - learned via RIP
- Displays shows
  - link state database
  - D.R. election
  - flooding procedure

ROUTERS GENERATE LINK STATE



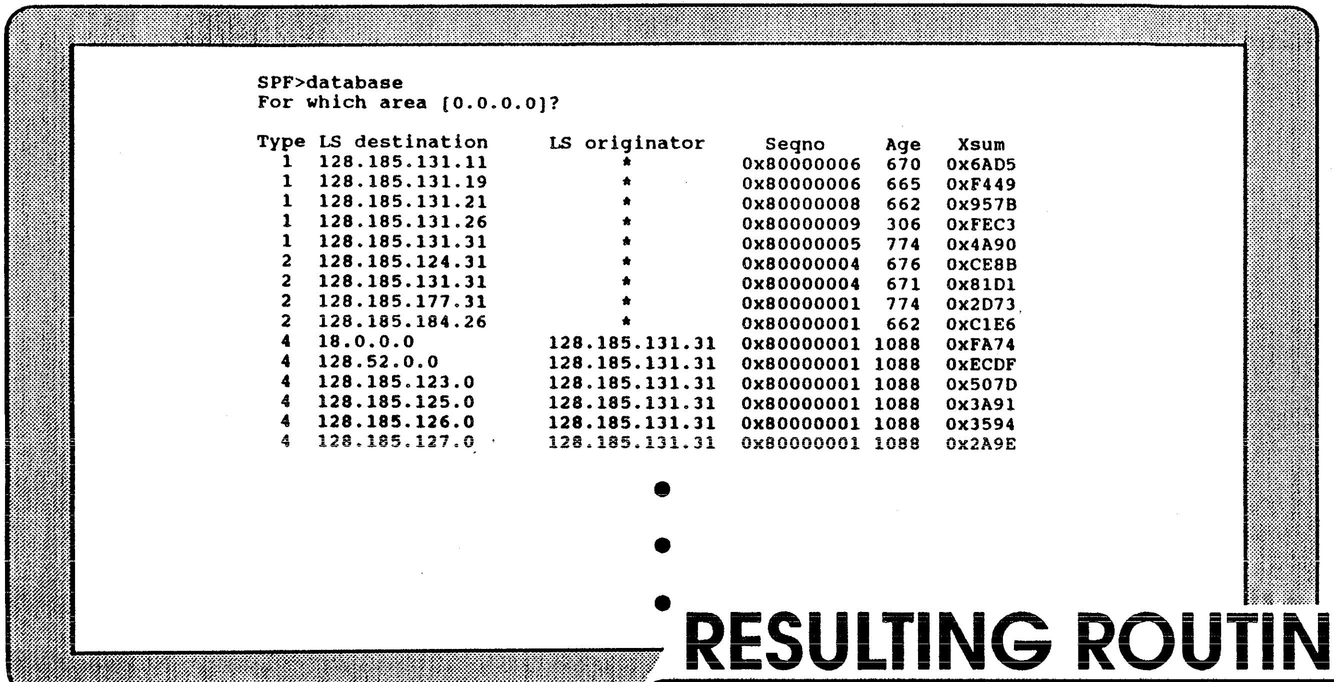
```
SPF>advertisement 1 128.185.131.11
For which area [0.0.0.0]?

        LS age:      631
        LS type:       1
        LS destination (ID): 128.185.131.11
        LS sequence no:  0x80000006
        LS checksum:     0x6AD5
        LS length:       56
        Router type:
        # router ifcs:   3
                Link ID:            128.185.131.31
                Link Data:          80B9830B
                Interface type:   2
                        No. of metrics: 0
                        TOS 0 metric:
                Link ID:            128.185.124.31
                Link Data:          80B97C0B
                Interface type:   2
                        No. of metrics: 0
                        TOS 0 metric:   1
                Link ID:            128.185.178.0
                Link Data:          FFFFFF00
                Interface type:   3
                        No. of metrics: 0
                        TOS 0 metric:   1
```

TRANSIT NETS GENERATE LINK STATE



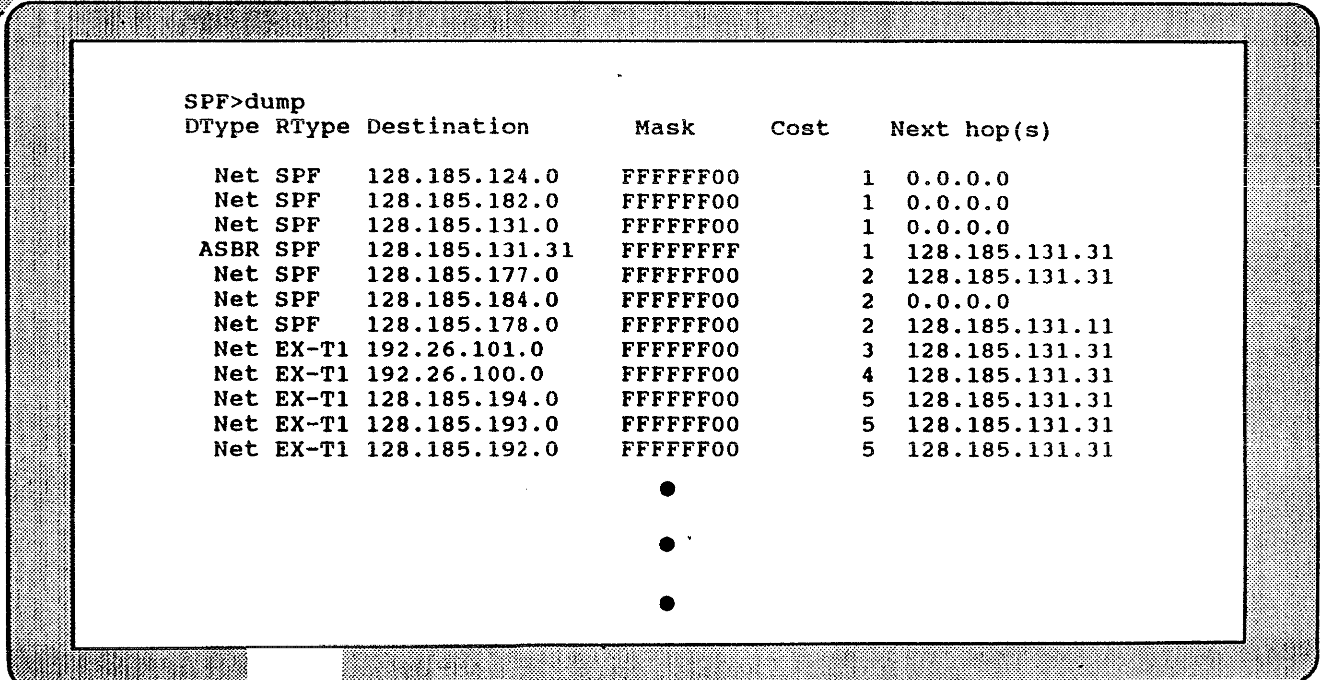
```
SPF>advertisement 2 128.185.131.31
For which area [0.0.0.0]?

        LS age:      661
        LS type:       2
        LS destination (ID): 128.185.131.31
        LS sequence no:  0x80000004
        LS checksum:     0x81D1
        LS length:       40
        Network mask:  FFFFFF00
                Attached Router: 128.185.131.31
                Attached Router: 128.185.131.19
                Attached Router: 128.185.131.21
                Attached Router: 128.185.131.26
                Attached Router: 128.185.131.11
```

COLLECTED LINK STATE FORMS ROUTING DATABASE



```
SPF>database
For which area [0.0.0.0]?

Type LS destination      LS originator      Seqno     Age   Xsum
  1  128.185.131.11            *          0x80000006  670  0x6AD5
  1  128.185.131.19            *          0x80000006  665  0xF449
  1  128.185.131.21            *          0x80000008  662  0x957B
  1  128.185.131.26            *          0x80000009  306  0xFEC3
  1  128.185.131.31            *          0x80000005  774  0x4A90
  2  128.185.124.31            *          0x80000004  676  0xCE8B
  2  128.185.131.31            *          0x80000004  671  0x81D1
  2  128.185.177.31            *          0x80000001  774  0x2D73
  2  128.185.184.26            *          0x80000001  662  0xC1E6
  4  18.0.0.0            128.185.131.31   0x80000001 1088  0xFA74
  4  128.52.0.0          128.185.131.31   0x80000001 1088  0xECDF
  4  128.185.123.0       128.185.131.31   0x80000001 1088  0x507D
  4  128.185.125.0       128.185.131.31   0x80000001 1088  0x3A91
  4  128.185.126.0       128.185.131.31   0x80000001 1088  0x3594
  4  128.185.127.0       128.185.131.31   0x80000001 1088  0x2A9E

                               ●

                               ●

                               ●
```

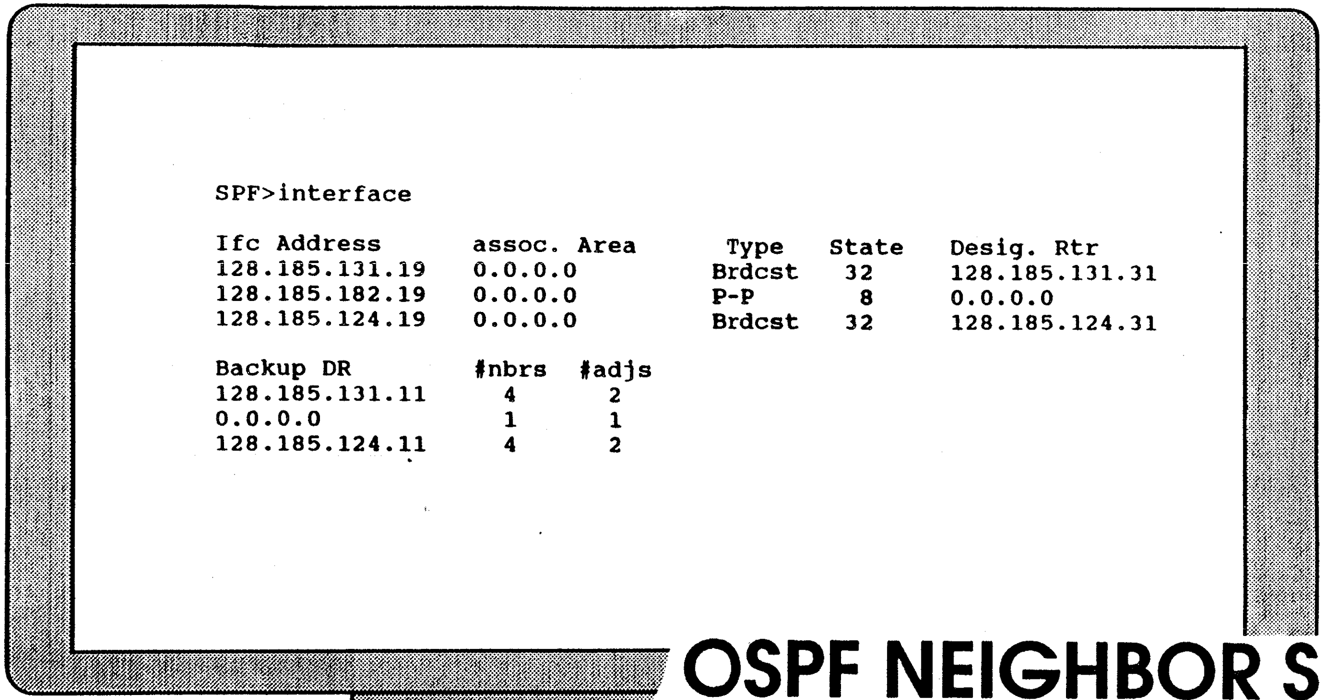
RESULTING ROUTING TABLE



```
SPF>dump
DType RType Destination       Mask      Cost    Next hop(s)

  Net SPF   128.185.124.0    FFFFFF00       1  0.0.0.0
  Net SPF   128.185.182.0    FFFFFF00       1  0.0.0.0
  Net SPF   128.185.131.0    FFFFFF00       1  0.0.0.0
 ASBR SPF   128.185.131.31   FFFFFFFF       1  128.185.131.31
  Net SPF   128.185.177.0    FFFFFF00       2  128.185.131.31
  Net SPF   128.185.184.0    FFFFFF00       2  0.0.0.0
  Net SPF   128.185.178.0    FFFFFF00       2  128.185.131.11
  Net EX-T1 192.26.101.0     FFFFFF00       3  128.185.131.31
  Net EX-T1 192.26.100.0     FFFFFF00       4  128.185.131.31
  Net EX-T1 128.185.194.0    FFFFFF00       5  128.185.131.31
  Net EX-T1 128.185.193.0    FFFFFF00       5  128.185.131.31
  Net EX-T1 128.185.192.0    FFFFFF00       5  128.185.131.31

                                 ●

                                 ●

                                 ●
```

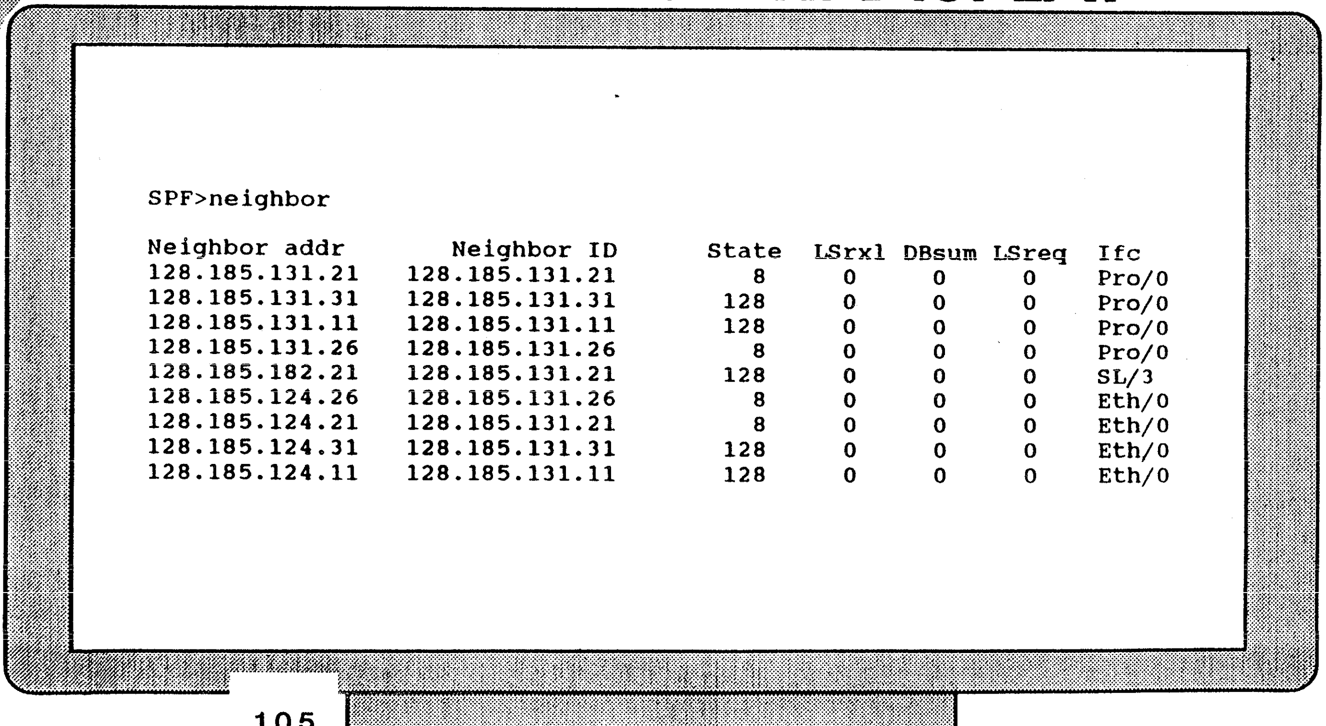
OSPF INTERFACE STATE DISPLAY

proteon



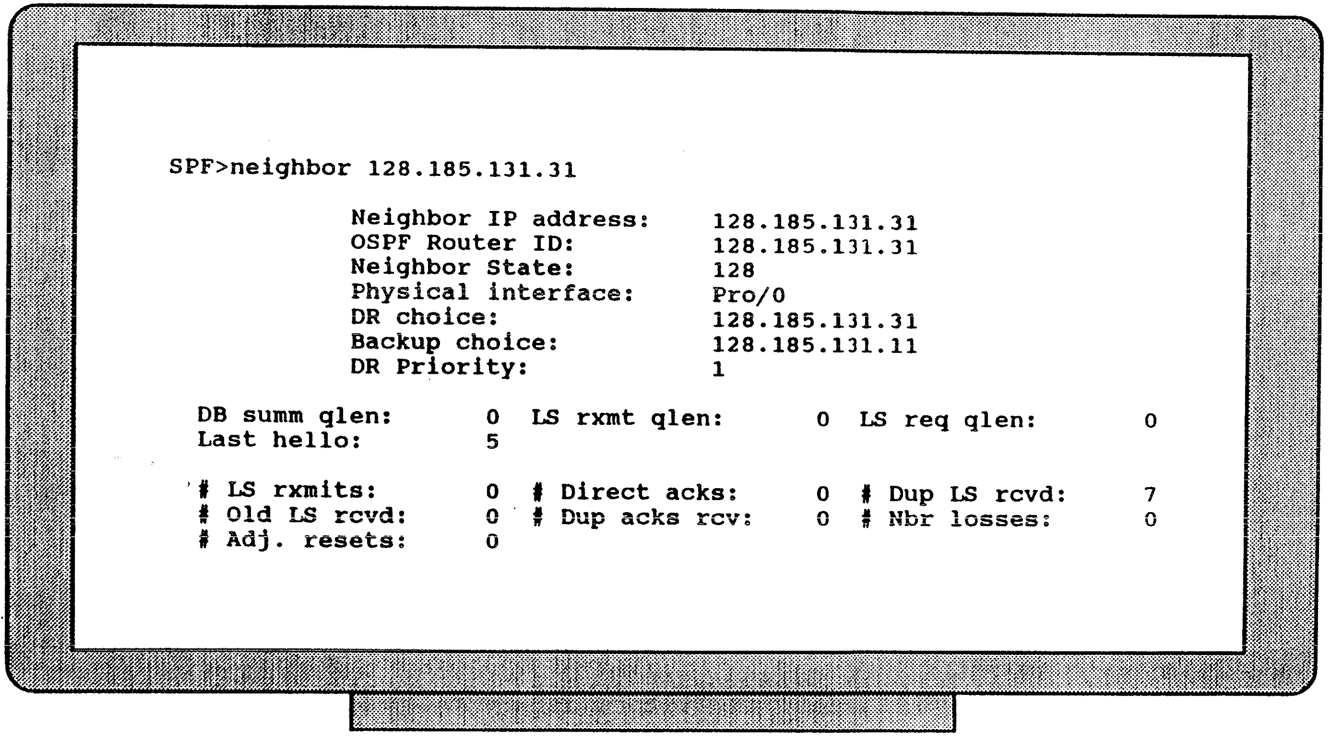
```
SPF>interface

Ifc Address       assoc. Area        Type    State    Desig. Rtr
128.185.131.19    0.0.0.0           Brdcst    32     128.185.131.31
128.185.182.19    0.0.0.0           P-P        8     0.0.0.0
128.185.124.19    0.0.0.0           Brdcst    32     128.185.124.31

Backup DR         #nbrs   #adjs
128.185.131.11      4       2
0.0.0.0             1       1
128.185.124.11      4       2
```

OSPF NEIGHBOR STATE DISPLAY



```
SPF>neighbor

Neighbor addr        Neighbor ID        State  LSrxl  DBsum  LSreq  Ifc
128.185.131.21    128.185.131.21          8      0      0      0    Pro/0
128.185.131.31    128.185.131.31        128      0      0      0    Pro/0
128.185.131.11    128.185.131.11        128      0      0      0    Pro/0
128.185.131.26    128.185.131.26          8      0      0      0    Pro/0
128.185.182.21    128.185.131.21        128      0      0      0    SL/3
128.185.124.26    128.185.131.26          8      0      0      0    Eth/0
128.185.124.21    128.185.131.21          8      0      0      0    Eth/0
128.185.124.31    128.185.131.31        128      0      0      0    Eth/0
128.185.124.11    128.185.131.11        128      0      0      0    Eth/0
```

RELIABLE FLOODING PER ADJACENCY



```
SPF>neighbor 128.185.131.31

            Neighbor IP address:    128.185.131.31
            OSPF Router ID:         128.185.131.31
            Neighbor State:         128
            Physical interface:     Pro/0
            DR choice:              128.185.131.31
            Backup choice:          128.185.131.11
            DR Priority:            1

  DB summ qlen:       0  LS rxmt qlen:      0  LS req qlen:       0
  Last hello:         5

  # LS rxmits:        0  # Direct acks:     0  # Dup LS rcvd:     7
  # Old LS rcvd:      0  # Dup acks rcv:    0  # Nbr losses:      0
  # Adj. resets:      0
```

Changes

→ We HAVE DRAFT ←

- PKT FORMAT
  - LSAs ALL HAVE ORIG ID
  - Reflected IN LS Update, D-D + Ack
- 2 TYPES OF LS Sum PKT TYPE 3 IS NET, 4 IS ASBR
- NBMA HELLOS ARE closer TO BROADCAST
  - Sent at INTERVALS TO KNOWN ADJACENCY
  - ALSO send Hello to new, DR ELIG ROUTER
- Fletch is back
  - 2 chksums are better than 1
- Hello over Point-to-Point
- DR DECISION STATE IS GONE
  - FROM WAITING TO DR, BDR or DROther

MODELING

- METRICS
  - # CHANGES PER TIME
  - OVERHEAD
    - NO CHANGES
    - WITH CHANGES
  - NORMALIZATION TIME
  - PROCESSING TIME
- PARAMS
  - NODE COUNT
  - NET DIAM
  - CHANGE IN EXTERNAL ROUTE
  - Change IN INTERNAL ROUTE
  - MIN LINK SPEED
  - MEMORY USED

Net Management

- > Time Up
- > # Interfaces
- > Type of Machine
- > Which Protocol is Running
- > List of Interfaces and Neighbors
- > States of Interfaces and Neighbors
- > How Long Interface and Neighbor has been Up
- > Interface and Neighbor Down (when, how many times since last clear)
- > Cost of Interfaces
- > Type and Address of Network per Interface
- > Line Speeds
- > MTU of Net
- > Cumulative Stats - protocol packets tx and rx by type, errors

- > Routing Table
- > Next Hops
- > Cost to Network
- > Cost to Border Routers
- > Cost to External Routers
- > Enough Info To Be Able To Build Network Map

gregorio.stanford.edu

vmtp-ip/ipmulticast.README
.tar
.tar.Z

deering@pescadero.stanford.edu

nsipo.nasa.gov: pub/ospf.ps (7/28)
ospf-request@trantor.umd.edu

Open Systems Routing Working Group
Chairperson: Marianne Lepp/BBN

CHARTER

Description of Working Group:

The Open Systems Routing Working Group is chartered to develop a policy-based AS-AS routing protocol that will accommodate size and general topology.

Specific Objectives:

- Architecture
- Functional Specification
- Draft Protocol Specification

Estimated Timeframe for Completion:

December 1989

STATUS UPDATE

1. Chairperson: Marianne Lepp, mlepp@bbn.com
2. WG Mailing List: open-rout-interest@bbn.com
3. Last meeting: IETF- Cocoa Beach, April, 1989
4. Next Meeting: IETF- U of Hawaii, Oct 1 - 3 Nov 1989
5. Pending or New Objectives:
6. Progress to date (e.g., Documents Produced):
   - IDEA 007 Requirements
   - Functional Specification
   - Architecture in draft

Open Systems Routing Working Group
Chairperson: Marianne Lepp/BBN

CURRENT MEETING REPORT

Reported by Marianne Lepp

ATTENDEES

Combined Attendee Roster Open Routing/ANTF

1. Almes, Guy/almes@rice.edu
2. Braden, Bob/braden@isi.edu
3. Breslau, Lee/breslau@jerico.usc.edu
4. Brim, Scott/swb@devvax.tn.cornell.edu
5. Clark, David/ddc@lcs.mit.edu
6. Estrin, Deborah/estrin@usc.edu
7. Farinacci, Dino/dino@bridge2.3com.com
8. Lepp, Marinne/mlepp@bbn.com
9. Little, Mike/little@saic.com
10. Mogul, Jeffrey/mogul@decwrl.dec.com
11. Sollins, Karen/sollins@lcs.mit.edu
12. Steenstrup, Martha/msteenst@bbn.com
13. Su, Zaw-Sing/zsu@sri.com
14. Tsuchiya, Paul/tsuchiya@gateway.mitre.org
15. Wood, C. Philip/cpw@lanl.gov
16. Zhang, Lixia/lixia@lcs.mit.edu

MINUTES

We met jointly with ANTF. The first day was spent discussing issues from the architecture, in particular virtual links. On the second day we went through the architecture paper, pointing out places where it was confusing and insufficiently detailed. The second topic of the day was a discussion of experiments that can be undertaken now to support our work. The rest of the time was spent on ANTF discussions which will be described in the ANTF meeting report.

ORWG Status Report

- met jointly with ANTF
- Discussed draft of architecture RFC
  Sept-Oct
- Discussed experiments and experimental enviroment
- Discussed + reached agreement on Functional Requirements RFC

OSI Interoperability Working Group

Chairpersons: Ross Callon/DEC and Robert Hagens/Univ of Wisc

CHARTER

Description of Working Group:

Help facilitate the incorporation of the OSI protocol suite into the Internet, to operate in parallel with the TCP/IP protocol suite. Facilitate the co-existence and interoperability of the TCP/IP and OSI protocol suites.

Specific Objectives:

The following are specific short-term goals and objectives for the OSI WG. Other mid-term objectives have also been identified and are available from the chairs.

- Specify an addressing format (from those available from the OSI NSAP addressing structure) for use in the Internet. Coordinate addressing format with GOSIP version 2 and possibly other groups.
- Review the OSI protocol mechanisms proposed for the upcoming Berkeley release 4.4. Coordinate efforts with Berkeley folks.
- Review GOSIP. Open liaison with Government OSI Users Group (GOSIUG) for feedback of issues and concerns that we may discover.
- What routing should be used short term for (i) intra-domain routing; and (ii) inter-domain routing?
- For interoperability between OSI end systems and TCP/IP end systems, there will need to be application layer gateways. Are there outstanding issues remaining here?
- Review short term issues involved in adding OSI gateways to the Internet. Preferably, this should allow OSI and/or dual gateways to be present by the time that Berkeley release 4.4 comes out.

Estimated Timeframe for Completion:

Still being determined

STATUS UPDATE

1. Chairpersons:

   Ross Callon (DEC) callon@erlang.dec.com

   Rob Hagens (UWisc) hagens@cs.wisc.edu
2. Name of WG Mailing List(s):

   ietf-osi@cs.wisc.edu - submissions to list

   ietf-osi-request@cs.wisc.edu - addition/deletions
3. Date/Site of Last Meeting: Stanford University/July 24-28, 1989
4. Date/Site of Next Meeting: U of Hawaii/October 31 - November 1, 1989
5. Pending or New Objectives:
   - Complete the RFC for CLNP Echo (circulate draft before the next meeting).
   - Continue to explore architecture and mechanisms for routing and network management of encapsulated CLNP inside DoD IP for production purposes.
   - Prepare the IETF-OSI "OSI documents to read" list (ongoing).
   - Follow-up on the Gosip V2 comments in order to determine if they are accepted (report status at next meeting).
6. Progress to Date (e.g., documents produced):
   - RFC 1069, 1070, Comments on GOSIP V2
   - We have reviewed the Gosip Version 2 NSAP address format and support its current structure. We plan to issue a new version of RFC 1069 that conforms to the NSAP structure in Gosip Version 2.
   - We have continued definition of an echo-request/echo-reply function to propose as an RFC and possibly as an addendum to ISO 8473.
   - We have reviewed Gosip V2 and prepared official IETF-OSI comments on the document.

OSI Interoperability Working Group
Chairpersons: Ross Callon/DEC and Robert Hagens/U of Wisconsin

CURRENT MEETING REPORT

Reported by Ross Callon, Rob Hagens and Richard Colella

AGENDA

Monday, July 24th

- Discussion and review of GOSIP V2

Tuesday, July 25th

- Inter-domain routing
- Intra-domain routing

Wednesday, July 26th

- General Meeting
    1. BSD 4.4 Update
    2. Review of the CLNP Echo proposal
    3. Review of GOSIP comments
    4. Strategies for encapsulating CLNP in DoD IP
- X.500
- DEC DNS

ATTENDEES

1. Abramowitz, Alyson/ala@hpindda.hp.com
2. Alberts, Charlie/charlie@banyan.banyan.com
3. Barker, Trudy/trudy@sri-nic.arpa
4. Bierbaum, Neal/bierbaum@vitalink.com
5. Blackwood, Craig/craig@hprnd.rose.hp.com
6. Brackenridge, Billy/brackenridge@isi.edu
7. Breslau, Lee/breslau@jerico.usc.edu
8. Brim, Scott/swb@devvax.tn.cornell.edu
9. Callon, Ross/callon@erlang.dec.com
10. Chapin, Lyman/lyman_chapin@dgc.mceo.dg.com
11. Chi, Debra/debee@sun.com
12. Colella, Richard/colella@osi3.ncsl.nist.gov
13. Collins, Mike/collins@ccc.mfecc.llnl.gov
14. Coltun, Rob/rcoltun@trantor.umd.edu
15. Deboo, Farokh/fjd@bridge2.esd.3com.com
16. Denny, Barbara/denny@sri.com
17. Doo, Way-Chi/wcd@bridge2.esd.3com.com
18. Farinacci, Dino/dino@bridge2.3com.com
19. Fedor, Mark/fedor@nisc.nyser.net
20. Fink, Bob/rlfink@lbl.gov
21. Forster, Jim/forster@cisco.com
22. Galvin, James/galvin@tis.com
23. Garcia-Luna, J. J./garcia@sri.com
24. Gary Ralls, Vicki/iruucp!sun!ntrlink!ralls
25. Gerich, Elise/epg@merit.edu
26. Gerlach, Chuck/cag@iwlcs.att.com
27. Gross, Martin/martin@protolaba.dca.mil
28. Hagens, Rob/hagens@cs.wisc.edu
29. Hollingsworth, Greg/gregh@gateway.mitre.org
30. Hytry, Tom/tlh@iwlcs.att.com
31. Ilnicki, Ski/ski
32. Jacobsen, Ole/ole@csli.stanford.edu
33. Jarza, Peggy
34. Jones, Bill/jones@nsipo.arc.nasa.gov
35. Jordt, Dan/danj@cac.washington.edu
36. Katz, Dave/dkatz@merit.edu
37. Kincl, Norman/kincl@iag.hp.com
38. Knopper, Mark/mak@merit.edu
39. Kullberg, Alan/akullberg@bbn.com
40. LaQuey, Tracy/tracy@emx.utexas.edu
41. Lebeck, Sue/lebeck@tandem.com
42. Lee, Young/youngl@jessica.stanford.edu
43. Lepp, Marianne/marianne@bbn.com
44. Leser, Norbert/nl@osf.org

45. Little, Mike/little@saic.com
46. Love, Paul/loveep@@sds.sdsc.edu
47. Lynch, Dan/lynch@isi.edu
48. Maas, Andy/maas@jessica.stanford.edu
49. Malkin, Gary/gmalkin@proteon.com
50. Mankin, Allison/mankin@gateway.mitre.org
51. Merritt, Don/don@brl.mil
52. Mockapetris, Paul/pvm@isi.edu
53. Mogul, Jeff/mogul@decwrl.dec.com
54. Montgomery, Doug/dougm@osi3.ncsl.nist.gov
55. Mundy, Russ/mundy@tis.com
56. Nitzan, Rebecca/nitzan@ccc.mfecc.llnl.gov
57. Ohle, Bill/ohle@osi.ncsl.nist.gov
58. Opalka, Zbigniew /zopalka@bbn.com
59. Oran, Dave/oran@oran.dec.com
60. Pak, Raylene/raylene@tardis.tymnet
61. Palmer, Howard/sun!iruucp!ntrlink!palmer
62. Pillalamarri, Shyam/shyam
63. Pugh, Jon/pugh@nmfecc.llnl.gov
64. Pugh, Rex/pugh@hprnd.rose.hp.com
65. Ramakrishnan, KK/rama
66. Reilly, Michael/reilly@atari.nac.dec.com
67. Reinstedler, Jim/jimr@ub.ubcom.com
68. Rekhter, Yakov/yakov@ibm.com
69. Replogle, Joel/replogle@ncsa.uiuc.edu
70. Reschly, Robert/reschly@brl.mil
71. Roberts, Mike/roberts@educom.edu
72. Roselinsky, Milt/cmcvax!milt@hub.ucsb.edu
73. Scanlan, Keely/keely@hprnd.rose.hp.com
74. Schoffstall, Martin/schoff@nisc.nyser.net
75. Schofield, Bruce/schofield@edn-vax.dca.mil
76. Sheridan, Jim/jsherida@ibm.com
77. Sklowear, Keith/sklower@okeeffe.berkeley.edu
78. Smith, Tom/toms@hprnd.rose.hp.com
79. Sollins, Karen/sollins@lcs.mit.edu
80. St. Johns, Mike/stjohns@beast.ddn.mil
81. Stahl, Mary/stahl@sri-nic.arpa
82. Steenstrup, Martha/msteenst@bbn.com
83. Stefferud, Einar/stef@nrtc.northrop.com
84. Sweeton, Jim/sweeton@merit.edu
85. Tausworthe, Bob/tozz@hpda.hp.com
86. Tharenos, Michael/tharenos@jessica.stanford.edu
87. Travis, Lance/cmt@apollo.com
88. Tsai, Howard/hst@mtuxo.att.com
89. Vance, L. Stuart/vance@tgv.com
90. Vaughan, Lynn/lynna@hprnd.rose.hp.cpm
91. Volk, Ruediger/rv@germany.eu.net
92. Wilder, Rick/rick@gateway.mitre.org
93. Youssef, Mary/mary@ibm.com

MINUTES

The meeting was convened by co-chairmen Ross Callon and Rob Hagens. The major issues discussed at this meeting included: GOSIP version 2, inter-domain and intra-domain routing, CLNP encapsulation, and directory services.

GOSIP V2 COMMENTS (Monday)

We went through the draft GOSIP version 2 document more or less front to back, and agreed on the following comments (to be submitted officially as OSIIWG comments):

Congestion Recovery:

It was suggested that the congestion recovery mechanisms should be mandatory in GOSIP, rather than "strongly recommended". It was pointed out that there is a difference between congestion recovery and congestion avoidance, and that only the congestion recovery need be mandatory. After a brief discussion this proposal was accepted.

Inclusion of CO/CL indicator in SEL part of NSAP address:

Several people raised concern about the bit in the SEL part of the NSAP address which indicates whether the network service is connectionless or connection-oriented. It was explained that in some cases, in the absence of directory services, an ES which is to initiate communications may have the remote NSAP address available, but not know whether to use the connectionless or connection oriented network service. By looking at the bit in the remote NSAP address, it would know what protocol/service type to use. This was described as an interim measure.

This explanation raised the level of interest from mild displeasure to serious concern. In particular, it was clear that some people were planning to write code which relied upon specific meaning in the SEL fields of remote NSAP addresses. Software written with this assumption would never be able to interact with any end system which happened to assign the wrong value to that bit of the NSAP address (such as non-Gosip OSI-compliant systems, or systems implementing future or past versions of GOSIP). We quickly agreed that this was undesirable.

NSAP format:

Other than the CO/CL bit in the SEL field, people were quite happy with the NSAP format. We agreed that RFC 1069 should be re-written to be compatible with GOSIP version 2.0.

NSAP Assignment and Administration:

There was a lengthy discussion about who was to administer NSAPs. Doug Montgomery suggested that since they were already setting up administrative procedures for the bulk of the Government, perhaps the same procedures should be used for the DoD. There was also some talk about whether the same procedures should be used for the entire Internet community, including educational institutions and government contractors. There was no agreement on this last group, except that in general there was no clear distinction between what was a government contractor and what was a private company (which would be expected to get their assignment from ANSI). Related to this discussion was the issues surrounding the administration of ICD 0005 and ICD 0006. Although ICD 0006 is delegated to the DoD (and therefore, part of the Internet), many felt that all addresses should be registered under ICD 0005, and ICD 0006 left empty (or for private military use).

There was also a discussion of the need for guidelines on (i) what sort of agencies should be considered an Administrative Authority; (ii) Under what conditions should specific grouping of networks be included in one domain, versus being split into several domains. We appeared to be in agreement that in many cases the specific people who are tasked to set up domain and address structures will be folks who do not fully understand the technical ramifications of these choices (such as the effect on routing). It was also suggested that commercial companies probably have the same need for information of value to clients setting up large networks. It was agreed that (i) There is a need for such guidelines; (ii) Writing these guidelines is beyond the scope of the OSIIWG, although we would like to review and comment upon any guidelines intended for the Internet; (iii) This was an important issue which should be brought to the attention of the IAB and the FRICC, but which did not result in any specific comment to GOSIP.

It was agreed that it would be preferable to describe the address administration in a separate document from the NSAP address, rather than postpone re-issuance of RFC 1069 in order to include both issues in one RFC.

O/R Names:

We were generally in agreement that the X.400 O/R Names section in Gosip has problems.

Priority Processing of PDUs:

The GOSIP 2.0 spec contains a bullet item which could be interpreted to mean that in order to conform with GOSIP you HAVE to separate incoming traffic by priority before processing the header (which would seem to imply mostly processing the header to find the priority, then queueing the packet, then re-processing the header). On the other hand, it was pointed out that in some specific environments priority forwarding of packets is very important. We proposed alternate wording which we feel preserves the possibility for individual acquisition authorities to require priority handling of packets where appropriate, while correcting the possible mis-interpretation.

Example of use of DoD Management Protocols:

In the introductory section there is a discussion of the need to use "Tertiary" sources for protocol specifications (sources which are neither standards nor proposed standards). An example was given of use of DoD management standards (designed for use with TCP/IP) for management of OSI systems. We agreed that this was a poor example, and proposed a better example (use of the ANSI MIB along with DIS version of CMIP).

General:

Various folks were tasked with writing up paragraphs describing each item, which Ross agreed to type up for submission to NIST.

INTER-DOMAIN ROUTING (Tuesday)

We had a half day for discussion of Inter-domain Routing Protocols, which was intended for two purposes: (i) For information purposes, to increase understanding of what possibilities are under development; (ii) To determine what we want to do short term in the Internet.

Marianne Lepp (chair of the IETF Open Routing Working Group) gave a presentation of the inter-domain architecture on which the ORWG is working, and presented a schedule for more concrete written architectural and protocol specifications. Doug Montgomery gave a presentation of the NIST scheme, which ECMA and NIST are bringing to OSI. Finally Jacob Rehkter of IBM gave a presentation of the short-term proposal of the Interconnectivity Working Group.

We then had a discussion of what to do for short term use in the Internet. Yacob Rechkter asked: "How many routing domains do we have currently in the Internet?" (obviously the answer is none). He then asked: "How many will we have in two years?" (probably not very many). He suggested that the number of domains will probably be very small for several years, and that we have much more pressing problems. So, why not just use fixed tables for now, and in a couple of years re-visit the question (with the benefit of the work of the other Internet groups working on this problem for TCP/IP). We quickly agreed on this.

There ensued a brief discussion that essentially was of the question "How fixed are fixed routing tables, and what might one offer to allow remotely updating them". There was no clear conclusion.

INTRA-DOMAIN ROUTING (Tuesday)

Dave Oran gave a presentation of the DEC/ANSI intra-domain IS-IS routing protocol, with emphasis on the changes made since the older (October 1987) version. Dave had a hard copy of the brand new updated proposal, which was sent to be copied and distributed. The new version of the ANSI IS-IS routing spec, which will be submitted to ISO in Sept. 1989 will follow, with luck, the following progression through ISO:

- DP Jan, 1990
- DIS Oct, 1990
- IS June, 1991

GENERAL MEETING (Wednesday)

BSD 4.4 STATUS REPORT

A brief status report on 4.4 BSD was given by Rob Hagens. The ISODE is being ported to run over the Wisconsin lower layers. Testing of the now complete OSI stack will begin shortly.

ECHO OPTION FOR ISO 8473

Two mechanisms for realizing an 8473 echo request/reply were discussed by Rob Hagens: using a SEL value to indicate that a DT pdu should be sent to an echo entity, or using a new type code in the pdu itself to distinguish a DT from an echo request/reply.

The OSIIWG felt that the use of the SEL is a good short term solution. However, the new type field is the best long term solution. The echo draft (not yet public) should be edited to suggest that the SEL field be used in the short term. Concurrent to this, the new type code should be described in detail and submitted to ANSI as a work item.

Finally, the source route option was discussed. Some people would like to use it in the echo-request and have it reversed in the echo-reply. Others would like it not copied from echo-request to echo-reply. Since the source route option is currently incorrectly specified in ISO 8473, it was suggested that the best approach is to discourage use of a source route option when using an echo facility.

ENCAPSULATION

The OSIIWG agreed that a production encapsulation method was a necessary transition aid. EON, as specified in RFC 1070 is insufficient.

The act of wrapping and unwrapping CLNP in DoD IP should be performed by gateways. The CLNP should run in native mode as far as possible.

There are actually 3 sub-problems:

a) The wrapper/unwrappers must know of each other of the purpose of network layer routing. b) The wrapper (when acting as a SNDCP for CLNP) must obtain the mapping from NSAP address to SNPA (DoD IP address) of the unwrapper. c) It is not clear if the CLNP packet should be placed directly into the DoD IP data field, or if a small header should be used. The header might contain the NSAP/DoD IP address mapping of the wrapper. The general consensus was not to include this extra header.

The routing problem (a) is similar to that experienced when X.25 is used as an COSNS for CLNP. The group looked to the ANSI IS-IS proposal for support. The IS-IS solution does not provide a magic solution. A general opinion of the group was that static tables should be used. However, opinions varied considerably. The question really becomes: how static is static? Could we utilize a network management protocol to adjust static tables? This topic requires more discussion.

The method of mapping the NSAP to DoD IP address (b) was not determined. Again, the ANSI IS-IS spec. was not helpful. Possibilities include: embed the DoD IP address in the NSAP address, use static tables, use an SNPA server. This topic requires more discussion.

DIRECTORY SERVICES AND NAMING

Dave Oran gave a presentation on the DEC DNS naming scheme. Karen Sollins gave an ad hoc presentation on the X.500 name service.

OSI WG REPORT

- GOSIP V2 Comments
  - NSAP address format
  - SEL Field
  - Administration authority for NSAPs
  - Congestion recovery
  - X.400 ORAddress organization
- Inter-domain routing ...

STATIC

OSI WG REPORT (cont)

- Intra-domain routing

DP	Jan 1990
DIS	OCT 1990
IS	Jun 1991

- 8473 ECHO

  Short term - use Selectors

  Long term (ISO) - use type field

- 8473 encapsulation in DoD IP

  - unresolved

  issues: - what is sent?

  - routing? (branchy general topology net)

  - NSAP -> DoD IP address map

OSI WG REPORT (cont)

- Naming
  - DEC DNS
  - X.500

OSI-OR REPORT

- Deployment of X.400 in the Internet

- Addressing (3 phases)
  register
  bind
  Advertize

- Will register "com", "edu", ...
  and "com foo", "edu foo" .... foo=?

OSI - Routing

Inter - Domain Routing

Marianne Lepp

Doug Montgomery

Jacob Rehkter

Discussion

Lunch

Intra Domain Routing

Dave Oran

STATUS REPORT

INTER-AUTONOMOUS SYTEMS POLICY ROUTING

Marianne Lepp

BBN Communications Corporation

OVERVIEW

- The Problem
- Architecture
- Status

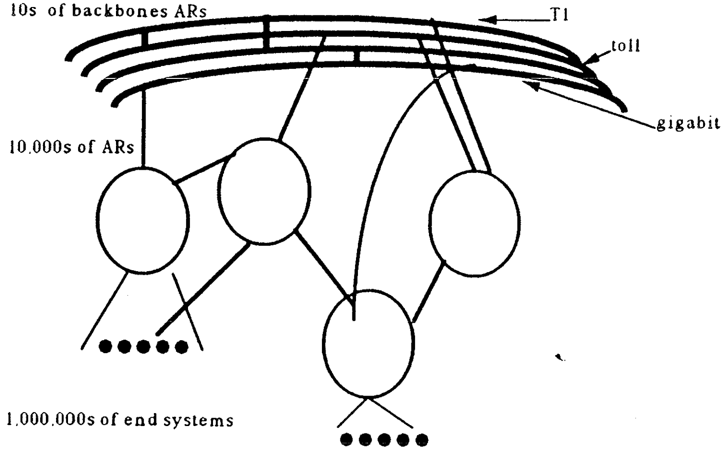
THE PROBLEM

- Inter-Autonomous System Routing
- Policy-Based Routing
- General Topology
- Grow Gracefully as the Internet Grows

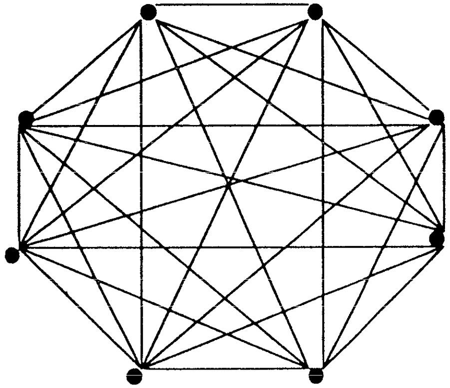
DRIVING REQUIREMENTS

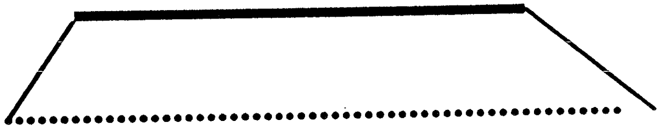
- 10,000s Autonomous Regions
  - 10,000,000 Networks
- General Topology
- Complex Policy Requirements
- Heterogenaity
- Limited Cooperation
- Privacy
- Performance
  - CPU
  - Bandwidth
  - Robustness

EVOLUTION OF THE INTERNET



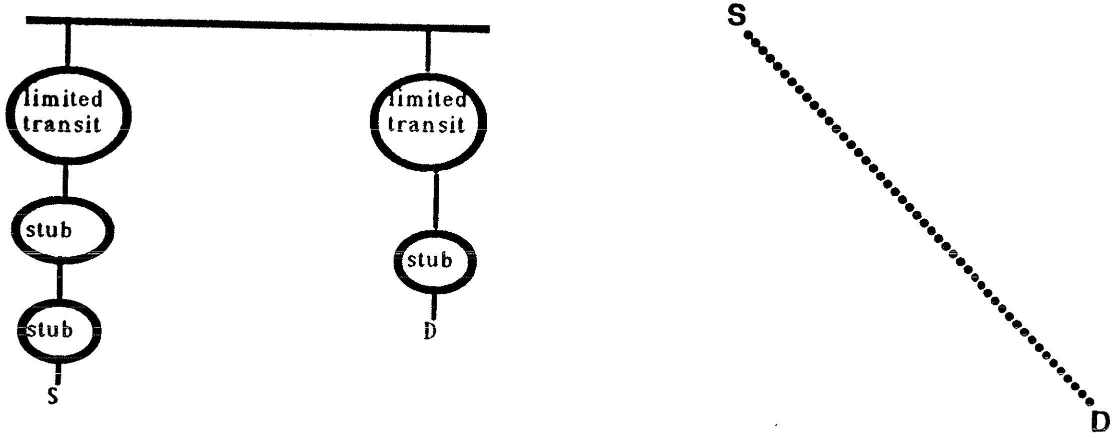
OTHER FUTURES





24 July1989

THE ROUTING PROBLEM IS AFFECTED BY THE SHAPE OF THE INTERNET



STRATEGY FOR SOLVING THE PROBLEM

√• Identify Requirements

√• Outline Architectural Elements

√• Develop Architecture

*• Flesh out the Architecture into a Design

• Specify the Details of the Design

• Test Implementations

ARCHITECTURE

- Data Reduction
- Local Autonomy
- Source Routing
- Source Routing Mechanism
- Route Failure Detection

DATA REDUCTION

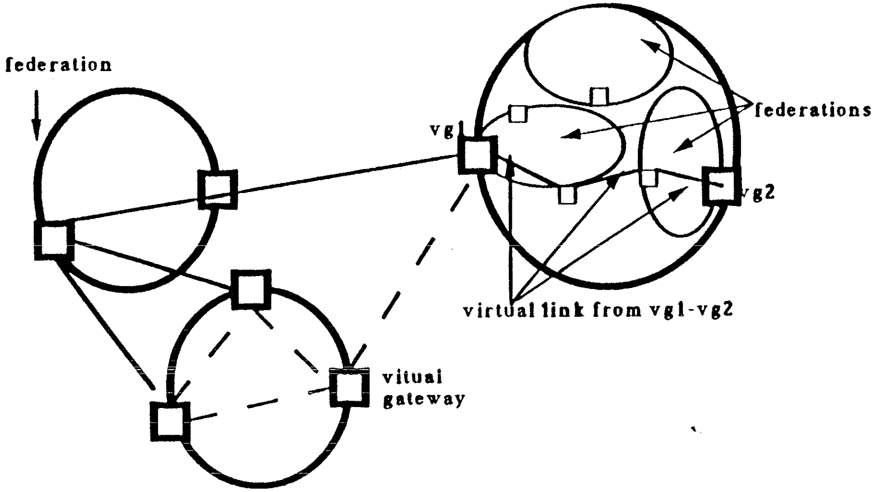
- Scaling the Problem to Size

- Why
  - 10,000,000 Networks
  - Robustness
  - Local Autonomy
  - Reduction of Information Exchange

- How
  - Object-oriented Routing
  - Federatons, Virtual Links, and Virtual Gateways

DATA REDUCTION

24 July 1989



Federations of Federations of Autonomous Regions

LOCAL AUTONOMY VS GLOBAL KNOWLEDGE

- Local
  - Maintenance of Virtual Links
  - Maintenance of Virtual Gateways
  - Caching of Routes and Addresses
  - Routing to Ultimate Destinations

- Global
  - "Virtual" routes
  - Address Service

SOURCE ROUTING

- Why
  - Source Can Enforce User Policies
  - Source Control
  - Simplifies Routing Consistency Problems

- What Kind of Source Route?
  - Routes consist of Entrance(VG, Parcel AR/FAR) Pairs

MECHANISMS FOR SOURCE ROUTING

- Route in Each Packet
  - Packet-Level Overhead
  - Processing Overhead
- Interface List in Each Packet
  - Global Synchronization Needed

- Route Set-up
  - Fast Forwarding
  - Can be Used to Reserve Resources
  - Vehicle for Implementing Policy

- Route Set-up Is Best
  - Local Decision (staight-forward)
  - Specify the Interface
  - Pointer in Database with Policy Info

ROUTE FAILURE DETECTION

24 July1989

- Very Low-Level Status Topology Updates
- Virtual Gateways Run Link Up/Down
  - Virtual Links Repaired with No Notification for the Source
- If Source Route Fails, Source notified
  - Call Back Service

FUNCTIONAL ORGANIZATION

- User Agent
- Policy Agent
- Routing Agent
- Forwarding Agent
- Data Collection Agent

DOCUMENT STATUS

24 July1989

- √• Design Issues RFC
- √• Architectural Decisions paper
- *• Functional Requirements RFC
- * • Architecture RFC
- • Protocol Specification for Prototyping

MILESTONE STATUS

- Requirements Defined - Completed
- Achitecture Defined - Completed
- Requirements RFC - August 1989
- Architecture RFC - September 1989
- Design Defined- October, 1989
- Protyping Specification Draft - December 1989
- Prototype Specification, version 1 RFC - March 1990

NIST / ECMA
INTER-DOMAIN ROUTING PROPOSAL
OVERVIEW

DOUG MONTGOMERY (NIST)
NIST
TECHNOLOGY, B-217
GAITHERSBURG, MD 20899
dougm@osi3.ncsl.nist.gov
(301) 975-3614

TASSOS NAKASSIS
nakassis@ecf.ncsl.nist.gov

ASSUMPTIONS / GOALS

- RESTRICTING PATHS TO ADDRESS LEGAL, CONTRACTUAL, AUTONOMY PRESERVING CONCERNS. "POLICIES"
- POLICIES ARE RELATIVELY STATIC
- SOLUTION SUITABLE FOR ISO:
  - NOT ENGINEERED FOR PARTICULAR ENVIRONMENT
  - COMPATIBLE WITH EXISTING / PROPOSED ROUTING + DATA PROTOCOLS
  - WELL UNDERSTOOD PARADIGMS / TECHNOLIGIES
  - SCALABLE

NON GOALS

- ROUTE OPTIMIZATION AT THE EXPENSE OF POLICY.
- FINE GRAINED POLICY BASED UPON PERFORMANCE, USER CLASSIFCATION.
- ADAPTATION TO LOAD
- DYNAMIC POLICY CHANGES.

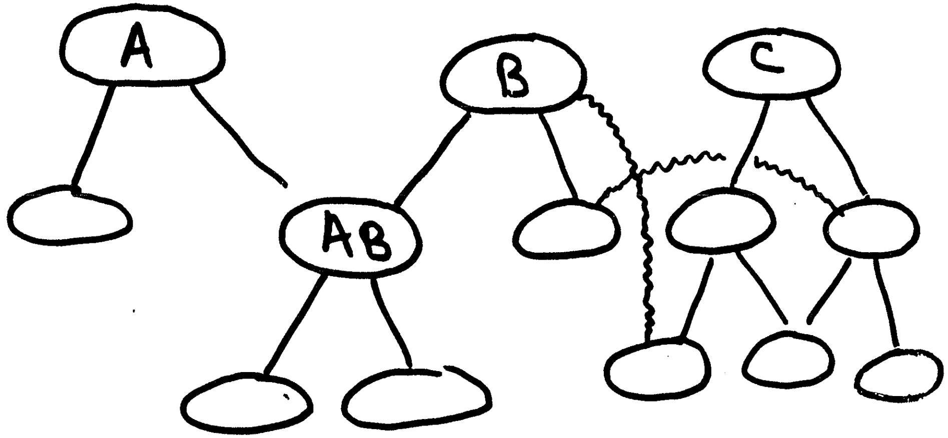
ARCHITECTURE PHILOSOPHY

- EFFICIENTLY DECOMPOSE "MEANINGFUL" POLICIES INTO SIMPLE LOCAL AGREEMENT
- EASILY ASSESS THE GLOBAL "IMPACT" OF THE COMPOSITION OF LOCAL AGREEMEN
- ISOLATE POLICIES;
  - ADD / REMOVE POLICIES WITH MINIMAL IMPACT ON OTHER POLICIES
- LOCALIZE COMPLEXITY / COST OF IMPLEMENTING POLICIES TO THEIR INSTIGATORS
- NO UNJUSTIFIED TOPOLOGY RESTRICTIONS

NIST/ECMA ARCHITECTURE

- MOST ISs SHOULD NOT BE INVOLVED IN INTER DOMAIN ROUTING (IDR).
- ISs THAT SERVE MULTIPLE ROLES SHOULD BE MODELED AS MULTIPLE DISTINCT LOGICAL ENTITIES.
- ISs ARE GROUPED INTO "CLUSTERS". EACH LOGICAL IS RESIDES IN A SINGLE CLUSTER.
- DOMAINS ARE LINKED BY DEPLOYING CLUSTERS AND ENTERING RELATIONSHIPS WITH OTHER CLUSTERS
- RELATION AMONG CLUSTERS FORMS A PARTIAL ORDER

EXAMPLE



∿∿∿ = "J" LINK.

ARCHITECTURE + POLICY

- INTER CLUSTER LINKS TAGGED: "UP, DOWN, JUMP" BASED UPON RELATIONSHIP OF CLUSTERS.
- INFORMATION ACROSS A "D" OR "J" LINK SHALL BE FOLLOWED BY "D" ONLY
- INTER CLUSTER LINKS MAY HAVE FILTERS THAT RESTRICT INFORMATION EXCHANGES IN ADDITION TO TOPOLOGY RULES
- POLICIES IMPLEMENTED BY RELATIVE POSITION OF CLUSTERS AND FILTERS.

PROTOCOL PHILOSOPHY

- ARCHITECTURE PROVIDES GENERAL MODEL OF OVERLAPPING ADMINISTRATIONS, COMMUNITIES OF INTEREST, SERVICE USERS AND PROVIDERS
- STRAIGHT FORWARG GENERALIZATION OF "HIERARCHICAL ROUTING" SCHEMES (TELEPHONY, EGP, NSFNet, ANSI INTERDOMAIN).
- ROUTING TECHNOLOGIES (INFORMATION DISTRIBUTION, ROUTE SYNTHESIS, FORWARDING) FOR ARCHITECTURE:

  HIERARCHICAL ROUTE SERVERS?

  "TYPICAL" ALGORITHMS MAPPED ONTO ARCHITECTURE.

NIST / ECMA Protocol

- Specifies inter cluster operation:
  - Link verification
  - Information Exchange
  - Route synthesis
- Inter Cluster "distance vector" Algorithm:
  - IS's maintain static information about local environment.
  - Border ISs exchange NSAP Reachability information
- Intra Cluster Algorithm must:
  - Distribute Inter cluster DV information within cluster
  - Identify Partitions of clusters.

Protocol Considerations

- Topology + Flow Rules → No loops
- Clusters + Distance vectors → most equipment failures will only induce Local Perturbations:
  - Intra cluster
  - Inter cluster attenuates

Support of Information abstraction and hiding

- Well understood technology
- Compatible with ANSI IS-IS

STATUS

- ECMA TR FINALIZED JUNE 1989
- 7TH DRAFT CONTRIBUTED TO SC6/WG2 MAY 1989
- DISTRIBUTED FOR MEMBER BODY COMMENT ALONG WITH NWI
- NIST SIMULATION STUDIES.

OSI Interdomain Routing Proposal

YAKOV REKHTER

T.J. WATSON RESEARCH CENTER
IBM Corp.

OSI Interdomain Routing

- How many OSI routing domains do we have today?
- How many OSI routing domains are we going to have by 1992?
- Do we have no other more important issues to worry about between now and 1992?
- current year + Network layer = 1992
  (layer 3 from OSIRM)

IWG

YAKOV REKHTER
T.J. WATSON RESEARCH CENTER
IBM Corp.

OSI Interdomain Routing

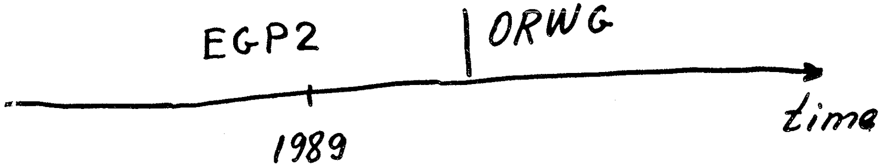
Proposal:

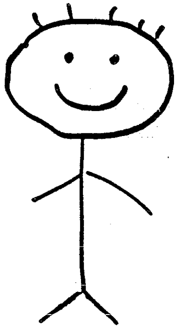
1. Use STATIC Interdomain Routing now – consistent with ANSI IS-IS Intra-domain proposal

2. Monitor routing activities in IETF and ISO

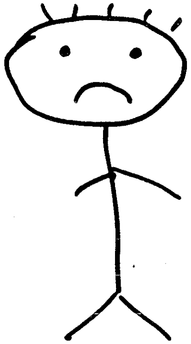
3. Concentrate on more pressing problems (e.g. intra-domain routing, performance improvements in 4.4 BSD, etc...)

Optimistic view





Pessimistic / Realistic view



Medium term Solution

- Not a longterm solution
- Does not solve <u>all</u> problems for <u>all</u> people
- Expected implementation cost $\neq \infty$
- Expected deployment efforts $\neq \infty$

Long term Solution

"One that would be right for generations to come"

(from "The Energy Crisis in Oceanview by Prof. J. Finnegan)

IWG progress

- How to fix EGP2 ?
- How to fix EGP3 ?
- Border Gateway Protocol

BGP Design Objectives

- Routing protocol
- Minimize changes in the current Internet

  $\Rightarrow$

  expected deployment efforts $\neq \infty$
- Not hard to implement

  $\Rightarrow$

  expected implementation cost $\neq \infty$

Border Gateway Protocol
(BGP)

- Initial design - January 89

RFC 1105 - June 89

- Independent implementations by:
  - GATED (public domain)
  - cisco
  - IBM (for NSFNET)
- Deployed and operational in the NSFNET Test Network

BGP

- RFC 1105 - defines protocol:
  - PDU Format
  - Protocol FSM
- "Application of the Border Gateway Protocol in the Internet"
- MIRA
- RFC 1104 - defines policy based routing model supported by BGP

Distance Vector versus Link State

Link state:

Choices:

1. Complete topology knowledge
2. Some form of source routing

   Choices:

   a) Source route in every packet

   b) Route setup

"Packet switching means that each packet can go by a different route to the same destination as the last packet headed there went by"

(from "The elements of networking style" by M. Padlipsky")

Distance Vector versus Link State

Distance vector:

- Gracefully handes partial knowledge (no source routing required)
- Immediatly associated with RIP
- Issue of routing table information looping

BGP

- Based on distance-vector
- Allows to construct routing spanning tree (similar to link-state)
- Fast and efficient suppression of routing table information looping

BGP

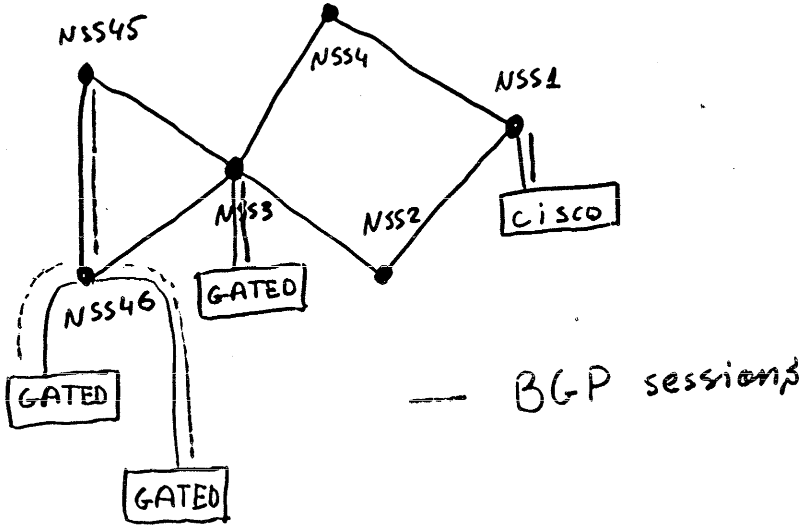
- Complete AS path
- Incremental updates
- Can run on top of reliable transport (e.g. TCP, VMTP)
  - current implementation uses TCP
- Trivial to extend for ISO

BGP Status

NSS1 - CISCO

NSS3 - GATED

NSS46 - GATED



State of the World in Iso Routing

Presentation for IETF OSI Transition Working Group

Dave Oran

Mike Shand

20-May-1989

Current Status

- DEC Phase V Routing Spec still the base text
- At stage of “Working Draft”
- Changes since previous draft:
  - Partial updates supported instead of fragmenting LSPs
  - Lollipop sequence space replaced by linear (*not circular)* space
  - LAN update throttling added
  - Memory exhaustion algorithms added
- All of these changes accepted technically at recent SC6 meeting

Current Status (cont)

- Pressure building (inexorably) to move this forward
  - -- IETF in particular
- Some issues outstanding
  - -- Multiple routing Metrics
  - -- Handling of security
  - -- Specification of addressing requirements
  - -- Overspecification/Formal Description concerns
  - -- Need for Conformance and PICS material

  (Later slides will deal with some of these in more detail)

Next Steps

- Checkpoint document last week of July to circulate informally at IETF
- Send out to "WG2 experts" in early August
- Editing meeting in Reading, U.K. last week of September
- Send out reworked spec to all of SC6 in early October
- DP from WG2 meeting in January/ February 1990
- DIS (hopefully) from SC6 meeting in September/ October 1990
- Full Standard by June 1991.

Multiple Routing Metrics

- Strong push to include limited support for multiple routing metrics (ToS routing)
  - Lots of people want it
  - ISO8473 implies you can do some of this with QoS bits
  - Most are aware of (theoretical) problems and will accept something simple
- We have something workable that:
  - Is simple
  - Is optional
  - Meshes well with ISO IP

Proposal for Multiple Routing metrics

- Current “cost” becomes the *default* routing metric — measures *throughput*
- Add three optional routing metrics:
  - *Delay*
  - *Monetary cost*
  - *Residual error / reliability*
- All routers must compute routes based on default metric
- Router may support any of the optional metrics on a per-link basis
- Reporting an optional metric on a link indicates that the router is computing routes based on that metric

IETF-OSI WG
July 26, 1989

- BSD 4.4 Update
- Results of the GOSIP V2 discussion
- 8473 ECHO
- 8473 Encapsulation
- DEC DNS

8473 ECHO

- Semantics
  1. Send a DT to the remote system
  2. Remote system sends a new DT back (containing the old DT
- How is ECHO DT identified ?

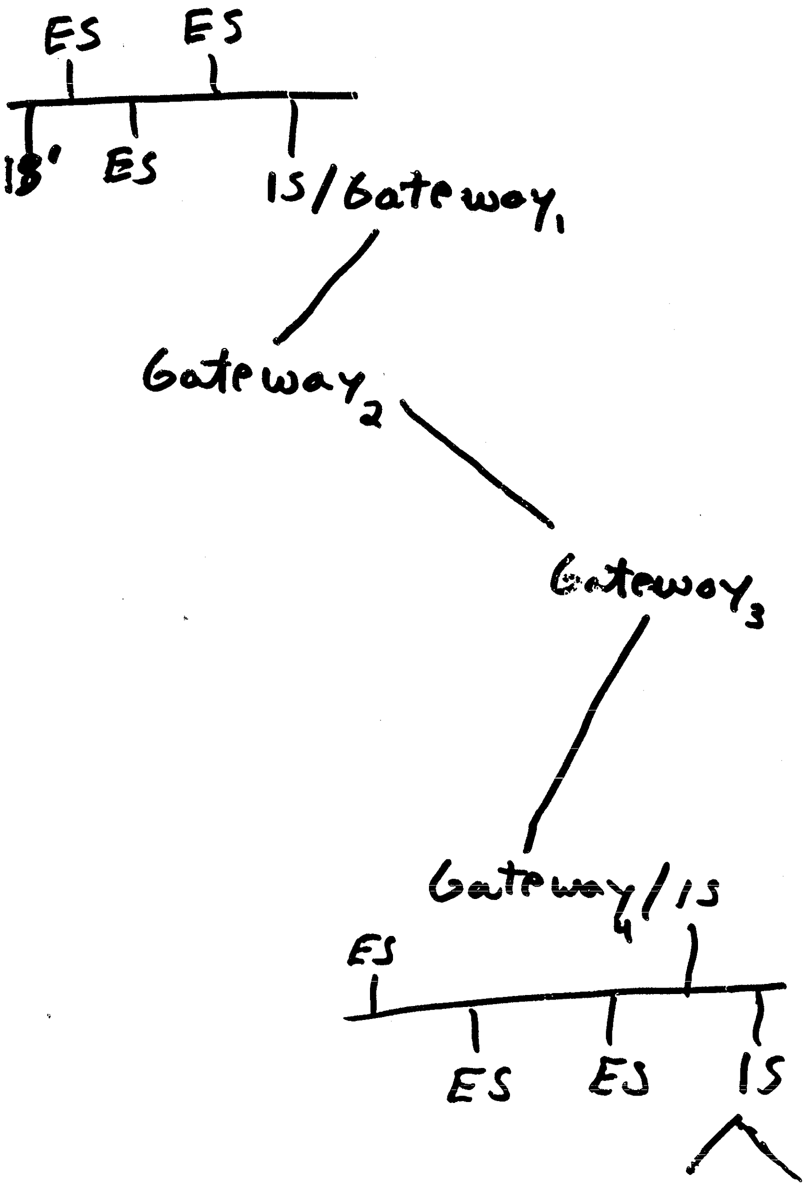
A) Special PDU Type (REQ/RPLY)
- \+ works
- \+ unambigious
- \- Protocol change
- \- LOSE backward compat

B) Special NSAP
- \+ works
- \+ backward compat
- \- xtra Dir lookup ?

8473 Encapsulation

- Transition tool
- Assume we don't have/want
  - A TS-bridge
  - An Application Gateway
- Assume we do have
  - a Full OSI Stack-LAN
  - IP routers in between

8473 Encapsulation



ISSUES

- What is sent ?
    - PDU
    - wrapper NSAP/INET address
- How is unwrapper Identified ?
    - Static table
    - route server
    - IS-IS protocol
      (auto config problem)
    - "Trace route" approach

PDN Routing Working Group
Chairperson: Carl-Herbert Rokitansky/Fern University of Hagen

CHARTER

Description of Working Group:

The DoD INTERNET TCP/IP protocol suite has developed into de facto industry standard for heterogenous packet switching computer networks. In the US, several hundreds of INTERNET networks are connected together; however the situation is completely different in Europe: The only network which could be used as a backbone to allow interoperation between the many local area networks in Europe, now subscribing to the DoD INTERNET TCP/IP protocol suite, would be the system of Public Data Networks (PDN). However, so far, no algorithms have been provided to dynamically route INTERNET datagrams through X.25 public data networks. Therefore, the goals of the Public Data Network Routing working group are the development, definition and specification of required routing and gateway algorithms for an improved routing of INTERNET datagrams through the system of X.25 Public Data Networks (PDN) to allow worldwide interoperation between TCP/IP networks in various countries. In addition, the application and/or modification of the developed algorithms to interconnect local TCP/IP networks via ISDN (Integrated Services Digital Network) will be considered.

Specific Objectives and Estimated Timeframe for Completion:

1. Application of the INTERNET Cluster Addressing Scheme to Public Data Networks. (Already done, see produced documents)
2. Development of hierarchical VAN-gateway algorithms for worldwide INTERNET network reachability information exchange between VAN-gateways (Already done, see produced documents)
3. Assignment of INTERNET/PDN-cluster network numbers to national public data networks. (Mapping between INTERNET network numbers and X.121 Data Network Identification Codes (DNICs) (Already done, see produced documents)
4. Assignment of INTERNET/PDN-cluster addresses to PDN-hosts and VAN-gateways according to the developed hierarchical VAN-gateway algorithms (Almost done, see produced documents)
5. Definition of the PDN-cluster addressing scheme as an Internet standard (Already done, [earlier than expected - a case that happens very seldom!] see produced documents)
6. Specification of an X.121 Address resolution protocol (RFC-Draft, expected to be completed by October '89)
7. Specification of an X.25 Call Setup and Charging Determination Protocol (RFC-Draft, expected to be completed by Fall '89)
8. Specification of an X.25 Access and Forwarding Control Scheme (to be written up as an RFC-Draft by Fall '89 or later)
9. Specification of routing metrics taking X.25 charges into account (to be written up as an RFC-Draft by Fall '89 or later)
10. Delayed TCP/IP header compression by VAN-gateways and PDN-hosts (new objective, will be considered Fall '89 or later)
11. Provide a testbed for worldwide interoperability between local TCP/IP networks via the system of X.25 public data networks (PDN) (starting June '89)
12. Implementation of the required algorithms and protocols in a VAN-BoX (Test version towards End '89)
13. Interoperability between ISO/OSI hosts on TCP/IP networks through PDN (1989/90)
14. Consideration of INTERNET Route Servers (1990)
15. Interoperability between local TCP/IP networks via ISDN (1990)
16. Development of Internetwork Management Protocols for worldwide cooperation and coordination of network control and network information centers (starting 1990).

STATUS UPDATE

1. Chairperson:

   Dr. Carl-Herbert Rokitansky, Fern University of Hagen, D-5860 ISERLOHN, FRG
   E-Mail: roki@DHAFEU52.BITNET, roki@A.ISI.EDU;
   Tel: ++49/2371/566-235

2. WG Mailing List(s):
   - pdn-wg@BBN.COM: For internal discussions and information exchange between members of the PDN Routing working group.
   - pdn-interest@BBN.COM: For information about:
     - Status report and proceedings of the PDN Routing WG
     - Draft proposals of documents and papers
     - Documents and papers published by PDN WG members
     - Important discussion on PDN Routing issues.
   - pdn-request@BBN.COM: For people interested in being put on the "pdn-interest" mailing list.
3. Date of Last Meeting: April IETF 1989, Cocoa Beach, FL
4. Date of Next Meeting: Oct 31 - Nov 2, 1989, IETF/Univ of Hawaii
   (Intensive information exchange via e-mail meanwhile)
5. Pending or New Objectives:
   (a) Specification of an X.121 Address Resolution Protocol: Expected to be completed as an RFC-Draft by October '89.

   (b) Specification of X.25 Call Setup and Charging Determination Protocol: Functionality, data structure and state diagram have already been defined, will allow reverse charging on international (!) X.25 connections, is currently in the progress to be written up as an RFC-Draft, and is expected to be completed by Fall '89.

   (c) Specification of an X.25 Access and Forwarding Control Scheme: Functionality and data structure have already been defined, might be included in the RFC-Draft above on X.25 Call Setup and Charging Determination Protocol, and is expected to be completed by late Fall '89.

   (d) Specification of routing metrics taking X.25 charges into account: Proposal is expected to be written up as an RFC-Draft by Fall '89 or later.

   (e) Delayed TCP/IP header compression: As a result of Van Jacobsons' presentation (IETF, April 13, 1989), a delayed version will be considered to be used on X.25 connections by VAN-gateways and PDN-hosts, as a new objective (Fall '89)

   (f) Provide a testbed for worldwide interoperability between local TCP/IP networks via the system of X.25 public data networks (PDN): Several institutes and research establishments in Europe, USA and Australia have already agreed to participate in these tests, which are expected to start in June '89.

   (g) Implementation of the required algorithms and protocols in a VAN-BoX: All the algorithms and protocols specified above are intended to be implemented in a VAN-BoX (or on a workstation) towards the end of 1989 (first test version).

   (h) Interoperability between ISO/OSI hosts on TCP/IP networks through PDN: Will be tested and demonstrated in connection with national and international PDN-tests (see below).

   (i) Consideration of Route Servers: Already discussed, but no detailed specification so far; will be considered with regard to results from international PDN-tests (see below).

   (j) Interoperability between local TCP/IP networks via ISDN: First discussions and proposals were already made, will be considered in detail in 1990.

   (k) Internetwork Management Protocols (cooperation of NOCs and NICs): Will be considered with regard to results from international PDN-tests (see above).
6. Progress to Date (e.g., documents produced):
   (a) Rokitansky, C.-H., "Internet Cluster Addressing Scheme and Its Application to Public Data Networks", in Proceedings of the 9th International Conference on Computer Communication (ICCC'88), pp. 482-491, Editor: J.Raviv, Tel Aviv, Israel, Oct 30 - Nov 4, 1988.

   (b) Rokitansky, C.-H., "Hierarchical VAN-Gateway Algorithms and PDN-Cluster Addressing Scheme for Worldwide Interoperation Between Local TCP/IP Networks Via X.25 Networks", in Proceedings of ITG/GI Conference on "Communication in Distributed Systems" (Informatik Fachberichte 205, Kommunikation in verteilten Systemen, ITG/GI Fachtagung, Stuttgart, P.J. Kuehn (Hrsg.), ISBN 3-540-50893-7 Springer-Verlag Berlin Heidelberg New York, ISBN 0-387-50893-7 Springer-Verlag New York Berlin Heidelberg), pp. 758-774, Stuttgart, Feb 22-24, 1989.

   (c) Assignment of default INTERNET/PDN-cluster addresses to VAN-gateways: Currently in the progress of being written up as an RFC-Draft, expected to be completed by October '89.

   (d) X.121 Address Resolution Protocol (first version has already been written up as an RFC-Draft to be discussed between members of the PDN Routing WG, and is expected to be completed by October '89).

   (e) Three Internet-Drafts produced in August 1989 (to be submitted for RFC consideration shortly):
   - "Internet Cluster Addressing Scheme"
     <draft-ietf-pdn-clusterscheme-00.txt>
   - "Application of the Cluster Addressing Scheme to X.25 Public Data Networks and Worldwide Internet Reachability Information Exchange"
     <draft-ietf-pdn-pdncluster-00.txt>
   - "Assignment/Reservation of Internet Network Numbers for the PDN-Cluster"
     <draft-ietf-pdn-pdnclusternetassignm-00.txt>

CURRENT MEETING REPORT

Did not meet during this quarter's IETF plenary.

Performance and Congestion Control
Chairperson: Allison Mankin/Mitre

CHARTER

Description of Working Group:

> The charter of the IETF Performance and Congestion Control Working Group is to collect and develop short-term techniques for improving Internet performance, methods which like TCP Slow-start are retrofittable and inexpensive to implement. After a preliminary draft of a white paper documenting such performance enhancements for hosts and gateways, it was decided to sharpen the focus and divide the material into two papers.
>
> One of the resulting papers is the RFC on gateway congestion control policies and algorithms. The intent of this paper is to present what is now known about the difficult problem of avoiding congestion in Internet gateways. It describes proposed policies such as Random Drop, Congestion Indication, and Fair Queuing, and sketches ground-rules for their adoption. An additional goal of the paper (achieved during the writing) is to generate dialogue on longer-term Internet gateway performance problems.
>
> The other paper is an RFC on TCP performance. This describes TCP algorithms such as Retransmit Backoff, Slow-start, Nagle (Small-Packet Avoidance), and Delayed Ack, as well as their correct interaction. The scope is to expand the treatment of TCP performance found in the Host Requirements RFC.

STATUS UPDATE

1. Chairperson: Allison Mankin/mankin@gateway.mitre.org
2. Name of WG Mailing List(s): ietf-perf(-request)@gateway.mitre.org
3. Date of Last Meeting: Stanford, July 26, 1989
4. Date of Next Meeting: TBD
5. Pending or New Objectives:

   The pending WG objective is the Transport Performance RFC. Existing draft is being expanded now that Stanford meeting is over.
6. Progress to Date (e.g., documents produced):
   - INTERNET-DRAFT: DRAFT-IETF-PERFCC-GWCC-00.TXT.1
   - Corrections and minor changes will result in a revision 01 within the near future (August).

Performance and Congestion Control Working Group
Chairperson: Allison Mankin/Mitre

CURRENT MEETING REPORT

Reported by Allison Mankin

AGENDA

The Performance and Congestion Control Working Group met in Palo Alto on July 26, 1989. The final agenda was:

GW Congestion Control Wrap-up	9:15-3:00
Work-in-progress: MIT Fair Queueing	Zhang
Discussion of INTERNET-DRAFT	Mankin
Work-in-progress: BBN Congestion Control	Steenstrup
Discussion of Fair Queueing Revisited	Shenker
Work-in-progress: Stochastic Fair Queuing	McKenney
TCP Performance	3:15-4:00

ATTENDEES

1. Coltun, Robert, rcoltun@trantor.umd.edu
2. Deboo, Farokh, fjd@bridge2.esd.3com.com
3. Fox, Richard, rfox@suntan.tandem.com
4. Hedrick, Chuck, hedrick@aramis.rutgers.edu
5. Hollingsworth, Greg, gregh@gateway.mitre.org
6. Kanakia, Hemant, kanakia@pescadero.stanford.edu
7. Karn, Phil, karn@thumper.bellcore.com
8. Lougheed, Kirk, lougheed@cisco.com
9. Lynn, Charles, clynn@bbn.com
10. McKenney, Paul, mckenney@sri.com
11. Parulkar, Guru, guru@flora.wustl.edu
12. Pugh, Rex, pugh@hprnd.rose.hp.com
13. Ramakrishnan, K.K., rama@erlang.dec.com
14. Reschly, Robert J., reschly@brl.mil
15. Schofield, Bruce J., schofield@edn-vax.dca.mil
16. Shenker, Scott, shenker@xerox.com
17. Skinner, Greg, gds@spam.istc.sri.com
18. Solensky, Frank, solensky@interlan.interlan.com
19. Steenstrup, Martha, msteenst@bbn.com Vance, L. Stuart, vance@tgv.com
20. Youssef, Mary, mary@ibm.com
21. Zhang, Lixia, lixia@lcs.mit.edu

MINUTES

The paper "Gateway Congestion Control Policies" was submitted as an INTERNET-DRAFT shortly before the Stanford meeting. It has been extensively revised and expanded from the draft discussed in Florida. In a nutshell, we draw the following conclusions:

1. Congestion is important and we can neither ignore it nor buy out of it.
2. There are multiple metrics for performance goals; but for any of them, finding an appropriate interval of measurement is critical.
3. Congestion recovery is a must; congestion avoidance is important, but much harder to get right.
4. Internet congestion recovery is simple to do, as demonstrated by Jain and Jacobson and by widespread practice.
5. Random Drop for congestion recovery is a 'win.'
6. There are many reasons to field Fair Queueing in gateways, assuming we solve implementation problems.

Our meeting at Stanford was intended to wrap up the effort associated with the INTERNET-DRAFT. Point 6 was the major discussion area for the day. Scott Shenker gave his paper "Fair Queueing Revisited" (soon to appear in the Proceedings of SIGCOMM '89) at the plenary session on July 25. It was scheduled then so that the WG could discuss it with him the following day.

DISCUSSION

How much is gained in gateway performance by refining the original Nagle algorithm with the Bit-Round and promptness allocation (delta) computations Shenker describes? Straight Nagle has at least two problems:

1. A user's turn which is sending a very long packet obviously gets more bandwidth than one which is sending a short one.
2. Users whose packets arrive just after their turn wait through a whole round-robin; if these are users whose bandwidth requirements have been low, they receive particularly unfair delay and bandwidth.

A quantitative comparison between Nagle and Bit-Round Fair Queuing has not been done. It was suggested that in the Internet we have had such bad delay behavior for the most part (with First Come First Serve gateways and no congestion avoidance), the improvements from Nagle might overshadow further improvements from solving 1 and 2. Scott pointed out that there are easy ways to improve Nagle Fair Queueing without doing full-blown Bit-Round and delta calculations. In the full-blown algorithms, deciding which packet to transmit requires examining state information for all the queues. One shortcut is to give up the low latency for new arrivals and restrict the examination to the current queue in the round-robin sequence. Going around, the decision of whether to send the packet or skip its turn is based on a Bits Sent value for the queue.

Several other algorithms were presented that look for shortcuts. One is Fair Queueing Fixed Quota (FQFQ) by Davin and Heybey (MIT). This differs from Bit-Round in dropping packets on congestion based on a fixed maximum length for each queue, rather than dropping only after a search for the longest queue. According to the MIT simulations presented to the WG, the fairness of this simplified buffer management is comparable to the original.

Another is Paul McKenney's (SRI) Stochastic Fairness Queueing (SFQ), in which packets are queued using a hash function. Each source-destination pair is not guaranteed a unique queue, but the algorithm leads to a fixed maximum round-robin cycle in addition to the queue location speedup of the hash.

In the simulations McKenney presented to us, SFQ did not include computations of when to send the packet. Maintaining a list of queues of each length was suggested for controlling the overhead of the search for the longest queue. A suggestion for how to simulate the delta computation (probabilistically) was to place candidate packets in the queue which had the next turn, instead of in the queue they hashed to.

How can we decide how much processing for Fair Queueing would be acceptable gateway overhead? Our INTERNET-DRAFT points out that we probably can't tolerate CPU requirements in a congestion control algorithm that increases as congestion increases. Counter-arguments included the possibility that Internet stability simply requires us to swallow this overhead (Scott's analyses of matrix models of networks of gateways point this way). Another argument Scott made is that the increase of CPU work with increasing congestion is alleviated by FQ variations that bound the number of queues; SFQ is one of these, as is using FQ largely for gateway policy enforcement, for then there is a small number of user classes.

How much can the processing overhead of FQ be squeezed down? Would it be possible to implement FQ in hardware or firmware? McKenney estimated that SFQ could be tuned to require only twenty memory references per packet for maintaining the queueing data structures. Parallel memory support can speed things further. The WG felt that a hardware-supported implementation of SFQ was quite possible. Note, though, that the queueing discipline in SFQ is that of Nagle Fair Queueing, so this instruction estimate did not include computations of when to send the packet.

On the subject of implementation speed, Kirk Lougheed (Cisco) reminded us that the Internet is growing rapidly in breadth with slow links (e.g. SLIP over 9.6kbps lines) during the same era that high speed links are arriving. For gateways serving the low bandwidth population, the processing for Fair Queueing would be essentially free – done while the CPU waits for the link to be available for the next packet.

PRESENTATIONS

Davin and Heybey's draft paper "Router Algorithms for Resource Allocation" on their Fair Queueing work was made available ahead of time. Contact jrd@lcs.mit.edu for information on obtaining an up-to-date copy. Lixia Zhang stood in for Chuck Davin and gave a presentation on the work. Their vugraphs follow these minutes.

Martha Steenstrup (BBN) gave us a blackboard presentation on the BBN gateway congestion control scheme. This is based on the ARPANET congestion control running now in PSN version 8.0. Link and CPU resources are controlled by allowing any flow (an Internet source-destination pair) a certain ration of each. The target utilization of the resources will be determined a priori from simulation results (to come). Buffers are not rationed, but are viewed as secondary to the other resources.

Each gateway makes a computation of the ration for each resource, based on a formula:

- ration(t) = min target, (target/load * ration(t-1))

The load in this expression is not the flow's but a measured aggregate for the gateway. Martha explained that measuring this load is very easy; the impact on fairness is acceptable since a greedy flow is given its correct fair share in one cycle, though other flows have to wait a few cycles until they find out they can increase.

The flow gets as its ration the minimum of the rations available along its path. In BBN's first phase of implementation, gateways collect the flow information from all the other gateways. Later, feed forward (sending an anticipated load ahead of your flow) will take its place.

The algorithm communicating and enforcing the rations for the current phase is called Smart Drop. Based on load measurements made at the gateway on a heuristically determined interval (probably about five seconds), and the collected flow information, the decision is made as to whether each flow is exceeding its ration.

The talk concentrated on the gateway processing. Similarities were noted between the BBN scheme and Selective Feedback Congestion Indication. Martha is looking into the availability for IETF members of a white paper she has written on Smart Drop. For more information, contact her at msteenst@bbn.com.

Finally, Paul McKenney presented SFQ. His vugraphs follow these minutes. Paul has completed a draft paper on his results. For more information contact him at mckenney@sri.com.

MISCELLANEOUS

The Chair and K.K. Ramakrishnan (DEC) submitted comments in the area of congestion control requirements to the OSI Interoperability Working Group Review of GOSIP Version 2, on July 24. We recommended:

- That GOSIP make congestion recovery algorithms mandatory, but for now leave congestion avoidance an option. (The reason for the qualification is that connections doing congestion avoidance (the DEC Bit policy) give up resources to connections that do not. Until there is more protection of quality of service at the gateways, it is premature to require congestion avoidance of hosts.)
- That GOSIP make dynamic retransmission timer estimation mandatory (it is currently optional in the TP specification and the NIST Implementors' Agreements).
- That GOSIP make the Quality of Service parameter a required part of the CLNP header. This would enable a near-future transition to using QOS Bits (including the DEC Bit, but also the bits needed for OSI IS-IS routing) in host and router processing. This recommendation was not adopted by the OSIIWG because the need was in the future.

Following discussion, we submitted a text for the OSIIWG-prepared GOSIP comments. The final version of the text is available by contacting rcallon%erlang.dec@decwrl.dec.com.

TCP PERFORMANCE

A short discussion (45 mins.) was devoted to the TCP performance paper. The attendees of that part of the meeting agreed that we would undertake to write some sections on the parallel performance requirements of TP4. For example, since TP4 does not have resegmenting (sequence numbers are by segment), the Nagle algorithm is even more significant in TP4 than in TCP. We will only write up the very clear cases, and the editor will keep in close contact with the OSIIWG, having already discussed the planned scope with Rob Hagens during a lunch hour at Stanford. A number of new writing volunteers appeared.

STOCHASTIC FAIRNESS QUEUING
FOR
CONGESTION CONTROL

Paul E. McKenney

ITSTD
SRI International

July 1989

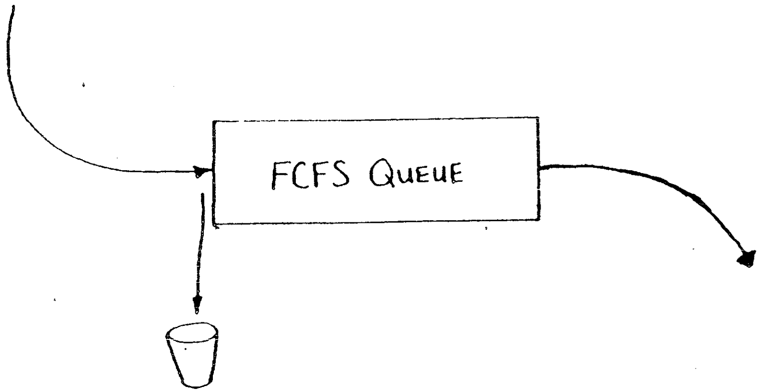
STOCHASTIC FAIRNESS QUEUING

OVERVIEW

- Congestion control
- Queuing disciplines
    - First-come-first-served
    - Random drop
    - Fairness queuing
    - Stochastic fairness queuing
- Results

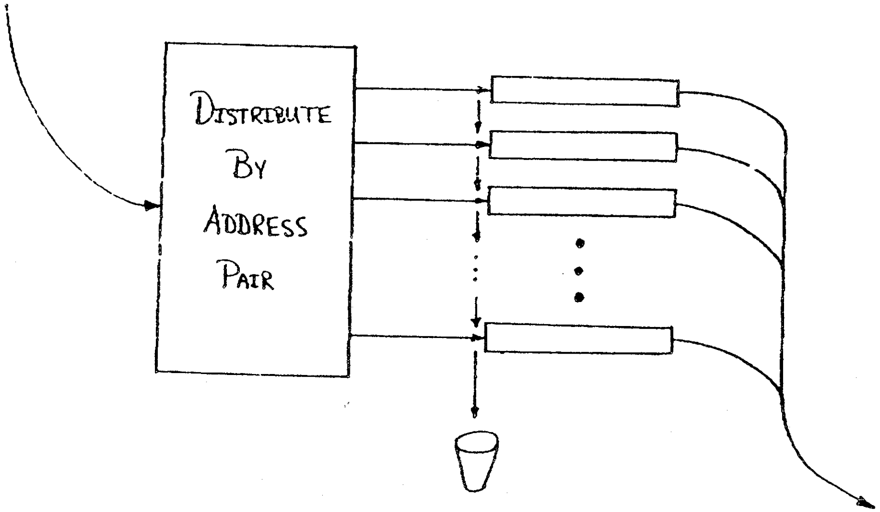
STOCHASTIC FAIRNESS QUEUING

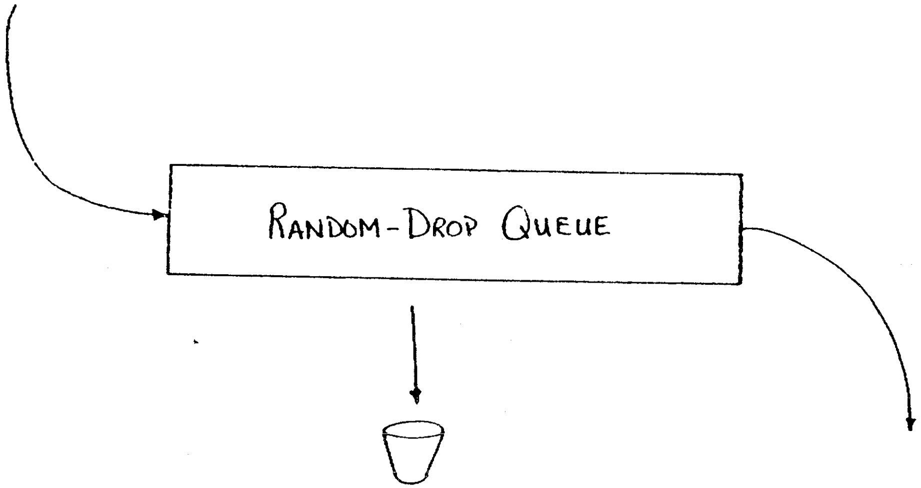
CONGESTION CONTROL



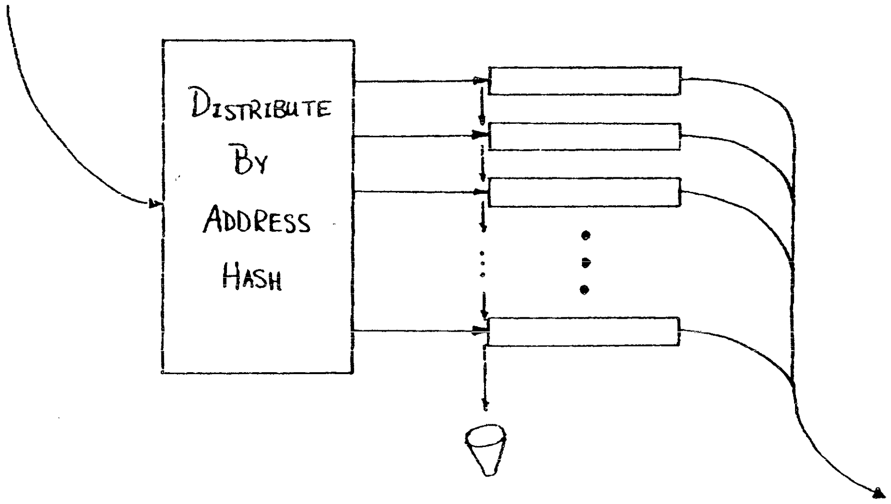
- Use network efficiently
- Approaches:
  - End-to-end
  - Local to gateway
  - Combination

FAIRNESS QUEUE





STOCHASTIC FAIRNESS QUEUE



STOCHASTIC FAIRNESS QUEUING

SFQ DIFFERENCES

- Simple mapping from address-pair to queue
- Addresses only scanned once
- All operations are time complexity O(1)
- Not guaranteed separate queue for each address pair

STOCHASTIC FAIRNESS QUEUING

SIMULATED ALGORITHMS

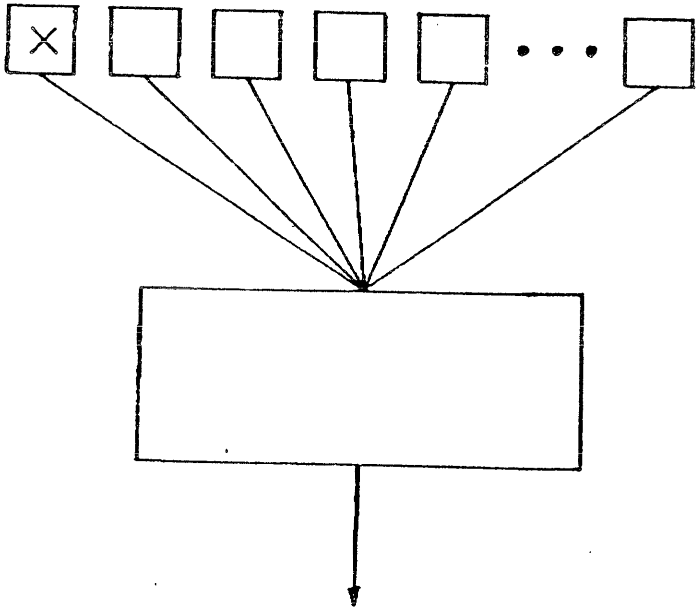
- First-come-first-served
- Random drop
- Fairness queuing
- Stochastic fairness queuing
  - Constant hash function
  - Switched hash function

STOCHASTIC FAIRNESS QUEUING

SIMULATION SETUP

- 20 concurrent "sessions" (one ill-behaved)
- Queues of length 5
- Four-times overload
- No protocol action

SIMULATION SETUP



STOCHASTIC FAIRNESS QUEUING

PERFORMANCE REQUIREMENTS

- About ten times as many queues as sessions
- Enough buffers for each session to have a couple
- Adequate rest between hash switching

STOCHASTIC FAIRNESS QUEUING

CONCLUSIONS

- Performance approximates fairness-queuing
- Adaptable for high-speed software/firmware
- Graceful degradation under overload
- Protocol-independent

Router Algorithms for Resource Allocation

(Chuck Davin & Andrew Heybey)

- Definitions and Assumptions
- Three Algorithms
- Long-Term Enforcement Results
- Short-Term Enforcement (WIP)

Some Definitions

(See Paper)

- Policy in terms of *User Class:* a single TCP connection or the traffic of an entire corporation; For this study: aggregates of TCPs
- *Uniform Policy:* each user class is entitled to the same share
- *Non-Uniform Policy:* some user classes are more equal than others

Some Assumptions

(See Paper)

- No architectural changes
- Plenty of buffers; plenty of CPU
- Criminals are everywhere but should not profit
- Packets can be reliably, quickly identified (SEP)
- No denial of idle resources (TDM, anyone?)

Three Algorithms

- FCFS (First Come First Served)
- FQNP (Fair Queueing No Punishment) Variation on Shenker in which user classes are NOT punished for dropped packets
- FQFQ (Fair Queueing Fixed Quota) Variation on Shenker in which each user class is afforded a fixed number of buffers; Motivation: reduce synchronization and overhead of original buffer mgmt policy

Long-Term Enforcement Results

- FCFS loses whenever demand deviates from policy
- FQNP and FQFQ enforce policy equally well over the long term
- FQNP and FQFQ effectively enforce non-uniform policies over the long term
- Relative performance of individual TCPs in each user class is no more pathological under FQNP or FQFQ than under FCFS.

Short-Term Policy Enforcement (WIP)

Approach: posit a magical, ideal policy enforcement machine and compare the behavior of each algorithm to it

This ideal machine is based on familiar definitions of *fairness* in the literature (e.g. Hayden, Hahne)

Short-Term Policy Enforcement (WIP)

- In the short term, FQFQ and FQNP enforce policy better than FCFS whenever the network is loaded

- No conclusions (yet) on relative merit of FQNP and FQFQ

- No conclusions (yet) on cases of underloaded network

- No conclusions (yet) on effect of each algorithm upon demand (TCP feedback mechanisms)

Point-to-Point Protocol Working Group
Chairpersons: Drew Perkins/CMU and Russ Hobby/UC Davis

CHARTER

Description of Working Group:

The working group is defining the use of serial lines in data networks. While the main intent is to standardize the connection of IP networks over point-to-point links, the protocol is being designed to be extensible to other network protocols as well. The protocol will provide the capability of establishing the link parameters, authentication, link encryption, link testing, as well as control of the link while it is up. The protocol will also allow configuration and control of the higher level protocols such as IP, OSI, 802.3 bridging, and others.

Specific Objectives:

The main objective of the workgroup is to produce an RFC defining the protocol for the link and IP levels.

Estimated Timeframe for Completion:

The final draft of the RFC will be completed for the Fall 89 IETF Meeting.

STATUS UPDATE

1. Chairpersons: Russ Hobby/University of California/Davis, rdhobby@ucdavis.edu
   Drew Perkins/Carnegie Mellon University, dpp@andrew.cmu.edu
2. WG Mailing lists: ietf-ppp@ucdavis.edu - main mail list
   ietf-ppp-request@ucdavis.edu - requests
3. Last meeting: Stanford, July 25-28, 1989
4. Next meeting: University of Hawaii, October 31 - November 3, 1989
5. Pending or New Objectives:

   Produce an RFC on protocol definition, final draft expected Fall 89, IETF meeting.
6. Progress to date (e.g., documents produced):
   - Requirements for a Point-to-Point Protocol, Perkins September 1988.
   - Complete protocol definition of link configuration and control. Definition of IP configuration and control being finished.

Point-to-point Protocol Work Group
Chairpersons: Russ Hobby/UC Davis and Drew Perkins/CMU

CURRENT MEETING REPORT

Reported by Russ Hobby

ATTENDEES

1. Cohen, Danny/cohen@isi.edu
2. Coltun, Rob/rcoltun@trantor.umd.edu
3. Deboo, Farokh/fjd@bridge2.esd.3com.com
4. Edwards, David/dle@cisco.com
5. Fair, Erik/fair@apple.com
6. Farinacci, Dino/dino@bridge2.3com.com
7. Fox, Craig/foxcj@nsco.network.com
8. Gross, Phill/pgross@nri.reston.va.us
9. Hobby, Russ/rdhobby@ucdavis.edu
10. Hollingsworth, Greg/gregh@gateway.mitre.org
11. Jolitz, William/william@ernie.berkeley.edu
12. Kaufman, Dave/dek@proteon.com
13. Khanna, Raman/khanna@jessica.stanford.edu
14. Kullberg, Alan/akullberg@bbn.com
15. LoVerso, John R./loverso@xylogics.com
16. Lottor, Mark/mkl@sri-nic.arpa
17. Maas, Andy/maas@jessica.stanford.edu
18. Mamakos, Louis A./louie@trantor.umd.edu
19. McKenney, Paul E./mckenney@sri.com
20. Melohn, Bill/melohn@sun.com
21. Merritt, Don/don@brl.mil
22. Natalie, Ron/ron@rutgers.edu
23. Opalka, Zbigniew/zopalka@bbn.com
24. Perkins, Drew /ddp@andrew.cmu.edu
25. Petry, Mike/petry@trantor.umd.edu
26. Satz, Greg/satz@cisco.com
27. St. Johns, Mike/stjohns@beast.ddn.mil
28. Tsai, Howard/hst@mtuxo.att.com
29. Waldfogel, Asher/wellflt!awaldfog

MINUTES

The PPP WG met on July 24, 25 and 26 at the IETF meeting at Stanford. Review of the latest draft of the specifications required discussion on the following areas:

1. An "Executive Summary" needs to be written for the beginning.
2. The PPP document should have less details of the HDLC protocol and have references to the appropriate documents on HDLC. The PPP document should include text of specifications that are unique to the PPP protocol application of HDLC.
3. All discussion of LAPB will be dropped from the document. The Enable LAPB option will also be removed.
4. There was again discussion of what protocol numbers to use, the ethernet numbers or new numbers. It was decided to let Jon Postel make the final decision with arguments presented for each case.
5. There was clarification of the wording in steps 3 and 4 of the description of the LCP sequence.
6. The Configure Request Request packet and the Character Generator Request/Reply packets were determined to be unnecessary and would be dropped.
7. A better description of Async Character Mapping is needed and how it relates to sync lines.

Many other minor editing changes were suggested and will be incorporated in the next draft.

The state diagram of the configuration exchange was examined in detail and the final form will be written up.

There was a lengthy discussion on the best method for doing keepalives. The final conclusion was that a keep-alive request would be sent to the remote end containing the number of packets sent. The remote end would send a keep-alive reply containing the difference in the number of packets sent and the number of packets received. Policy on when to take the line down could be determined at each end independently based on the information provided by the keep-alive packets. A more detailed description of the mechanism will be written.

There was discussion on what is the minimal implementation of PPP. The conclusion was: the minimum would be LCP configuration exchange with no options included. This would be followed by an IP configuration exchange with no options. The line would then be ready for IP traffic.

Areas in need of further work are:

- Stronger Authentication Protocols
- Definition of encryption methods
- Stronger IP address exchange methods
- Definition of the use of other high level protocols

The group plans to have a document with the agreed specifications finalized in two weeks followed with a video conference for verification of the text.

ST and Connection IP Working Group
Chairperson: Claudio Topolcic/BBN

CHARTER

Description of Working Group:

> Define the next version of the ST protocol, explore future connection oriented internet protocol, use the former as a testbed to perform experiments in support of the latter.

Specific Objectives:

- Produce a new specification of ST
- Produce a specification of a next generation connection oriented protocol

Estimated Timeframe for Completion:

1. Produce a new specification of ST. (2-3 months)
2. Produce a specification of a connection oriented protocol. (6-12 months)

STATUS UPDATE

1. Chairperson: Claudio Topolcic, BBN Labs, topolcic@bbn.com
2. Name of WG Mailing List(s): cip@bbn.com
3. Date of Last Meeting: July 27, 1989 Stanford California
4. Date of Next Meeting: October 31, 1989 University of Hawaii
5. Pending or New Objectives: none
6. Progress to Date (e.g., documents produced)
    - Internal draft of ST Specification
    - Numerous e-mail messages describing issues in connection oriented protocols

ST and Connection IP Working Group
Chairperson: Claudio Topolcic/BBN

CURRENT MEETING REPORT

Reported by Steve Casner, Allison Mankin and Claudio Topolcic

ATTENDEES

1. Casner, Steve/casner@isi.edu
2. Clark, David/ddc@lcs.mit.edu
3. Fedor, Mark/fedor@nisc.nyser.net
4. Fox, Richard/rfox@suntan.tandem.com
5. Mankin, Allison/mankin@gateway.mitre.org
6. Mazraani, Tony/tonym@flora.wustl.edu
7. Park, Philippe/ppark@bbn.com
8. Parulkar, Guru/guru@flora.wustl.edu
9. Ramakrishnan, KK/rama@erlang.dec.com
10. Su, Zaw-Sing/zsu@sri.com
11. Toplocic, Claudio/topolcic@bbn.com
12. Wood, C. Philip/cpw@lanl.gov
13. Zhang, Lixia/lixia@lcs.mit.edu

MINUTES

The working group held three meetings. The meeting held during the day of Tuesday 25 July covered the high level and long term issues of connection oriented internet protocols. A second meeting was held on 26 July and covered a number of short term issues that need to be discussed to finalize the ST specification. Since all such issues hadn't been addressed, we held a third meeting during the morning of 27 July.

Connection oriented internet protocol meeting of 25 July 1989

Three presentations were made. Lixia Zhang described the Flow Protocol (FP). Guru Parulkar gave a high level description of McHIP and Tony Mazraani described the McHIP protocol in more detail. These interactive presentations took the bulk of the day. Their content is not described here because they are better described in other documents.

The suggestion that the working group adopt McHIP as its protocol led to a discussion about the future of McHIP, ST and the working group. Adopting a specific protocol would provide direction and structure. It would help keep the working group from endless debating. It would cause us to look at practical tradeoffs, etc.

If a protocol is to be selected, then what should it be? The three choices appear to be McHIP, FP, and ST-2. It is somewhat easier to consider the relation between McHIP and ST-2. It would be optimal for both to evolve to a single protocol. This is reasonable since they are very similar. A significant difference is that ST-2 uses simplex connections to support conferences and McHIP uses omniplex connections. Another issue is that ST-2 is a very short term effort that will be operational in approximately six months, whereas McHIP is being developed in a somewhat longer schedule.

McHIP is harder to compare with FP. They seem to be addressing different issues. Guru felt that FP is more a network protocol than an internet protocol because it does not fully address the use of pure connectionless networks. Claudio felt that McHIP addresses a number of protocol issues, while FP provides a resource management algorithm. Claudio felt that we could reasonably implement a version of McHIP that incorporates an FP style resource management scheme.

Since time was running short, we decided to review our earlier ideas about the kinds of applications we wanted to support and the implications they have on the protocol. We decided to do this my E-mail. Phil will redistribute his messages entitled "Connection Oriented Protocol" and "Application Characterization" and Claudio will redistribute the minutes of the October 88 meeting. We will all read the first three sections of Guru's paper "The Next Generation of Internetworking". We will continue this discussion by E-mail. Phil and Guru will be in charge of writing a resulting paper.

ST protocol specification meeting of 26 July 1989

We went over the decisions we had made at the previous meeting and made a number of new decisions.

Reviewing the agreements from the evening meeting on ST at Cocoa Beach in April:

- IP encapsulation: adopt Steve Casner's description.
- Using IP-like headers: tabled for now; this can be retrofitted later.
- Interface to next higher protocol: it is OK to make changes as long as there is a good reason for them.
- We will need to write more documents than this one.
- Control messages: it is OK to define new ones if they have different functions from the old ones.
- ST.DG: eliminated.
- Source route option: it can be an option in the connect message. For multipoint this might be too hard, but one could at least do incremental connections with a source route option on each.
- Security: use SDNS.

New decisions to be made:

- Aggregation: we just get rid of it.
- Routing: the routing protocol is separate, just as for IP.
- Next protocol field: should this just be part of the Extension field (EXT= PROTOCOL & PORT)? But should ST carry only the IP address and protocol field, as does IP, and let the higher-level protocol carry the port number, as does TCP? This assumes that the "open" message of the higher-level protocol is carried in the ST connect message. Then, in multipoint the ST header must have a list of addresses and NVP must have a list of ports. We have not answered how the elements of these two lists are associated nor decided this issue yet.
- Keep PTP, or have a flag for automatic establishment of a reverse connection? Really, there should be three flags:
  1. "Don't assign a group address, I promise not to have more than 2 parties in the connection."
  2. "Do reverse HID assignment because there are only 2 parties now."
  3. "Allocate bandwidth for the reverse path, i.e., automatically set up two connections at once. For multipoint, set up N-1 individual reverse connections." Would have to carry two flowspecs in the connect.

  An issue is that multiple connection names must be assigned. If the source does them all, then the connection name could wind up corresponding to (having the IP address of) a host that's no longer in the connection:

  Step 1. Host A opens a multipoint connection to hosts B and C with the reverse bit on. This creates connections named A1(A -> B,C), A2(B -> A), and A3 (C -> A).

  Step 2. Host C incrementally adds to connection A3 a path to host D, so we have A3 that connects C->A,D.

  Step 3. Host A drops out, leaving A3 as a connection from C to D (but the connection name includes A).

  Is this a killer? "A detail for the RFC writer to work out."
- HID negotiation: Claudio will write up a reasonable one from his proposals for review, and we will use the static assignment subset implementation in practice.
- Robustness measures: We need more link-state information exchange to determine if ST connectivity is still established. That is, we must be able to tell if ST state was lost at the next agent. If there's a temporary net outage but no state loss, then just try a network repair (reallocate stream in WBnet). If a link goes down long enough to declare it dead, then back up hop by hop to do a pruned, depth-first search for alternate connection paths. The pruning is based on whatever routing information is available, so paths that are known to be insufficient are not tried. We will use Claudio's BREAK proposal for repairing the connection.

ST protocol specification meeting of 27 July 1989

Continuing the previous day's discussion:

> Robustness-level timeouts to keep track of agents going down: Exchange of keepalive control packets is per ST agent pair, which means per IP address pair. This only goes on if there is a connection up. An implementation may choose not to timeout if data is flowing but keepalives don't come in (fall behind). May want to have a separate timeout for each connection that is determined by the application: the connection is not flushed immediately upon declaring the link down, but only when the timeout expires after that (timeout may be zero). Applications should/may have their own data-dependent timeout, and then disconnect on failure.

ST and connection oriented internet protocols

- Connection oriented protocols
  - Presentation of Flow Protocol
    Lixia Zhang
  - Presentation of McHIP
    Guru Parulkar
  - Adopt McHIP?
    - give us a framework
    - work with something concrete
    - are we ready for something concrete?
    - can McHIP support what we want?
    - continue by E-mail

- ST-2
  - All outstanding issues discussed
    - Mostly resolved
  - Approach
    - KISS
    - When in doubt, leave it out
    - Subsettability
  - plan
    - Write up strawman solutions
    - Review by e-mail
    - Merge

TELNET Working Group
Chairperson: Dave Borman/Cray

CHARTER

Description of Working Group:

The TELNET working group is to look at RFC 854, "Telnet Protocol Specification", in light of the last 6 years of technical advancements, and determine if it is still accurate with how the TELNET protocol is being used today. This group will also look at all the numerous TELNET options, and decide which of them are still germane to current day implementations of the TELNET protocol.

Specific Objectives:

- Either re-issue RFC 854 to reflect current knowledge and usage of the TELNET protocol, or issue a companion RFC to update and expand on fuzzy areas of RFC 854.
- Create or update RFCs for TELNET options to clarify or fill in any missing voids in the current option set. (Most noteably, some method to allow automatic user authentication is needed).
- Act as a clearing house for all proposed RFCs that deal with the TELNET protocol.
- When the above objectives have been met, go dormant, and will be re-activated as needed to fullfill the objective of being a clearing house for future extensions to the TELNET protocol.

Estimated Timeframe for Completion:

Will be determined during the next meeting.

STATUS UPDATE

1. Chairperson: Dave Borman, dab@cray.com
2. WG Mailing List(s): telnet-ietf@cray.com
3. Date of Last Meeting: July 25-28, 1989
4. Date of Next Meeting:

   October 31 - November 1, 1989 University of Hawaii (tentative)
5. Pending or New Objectives: see Charter
6. Progress to Date (e.g., documents produced):

   Internet-Draft: "Telnet Linemode Option", July 1989

TELNET Working Group

Chairperson: Dave Borman/Cray

CURRENT MEETING REPORT

Reported by J.K. Reynolds, modified by Dave Borman

AGENDA

- Does RFC 854 (Telnet) need to be updated and re-issued?
- Do any of the option RFCs need to be updated and re-issued?
- What new options are needed?
- What about international character sets?
- What does BINARY mode really mean?
- How do you avoid option negotiation loops?
- What Telnet options are MUST? SHOULD? MAY? DONT?
- How do you flush input and output?
- 7 bit NVT vs 8 bit NVT vs 8 bit BINARY
- Telnet to other protocol translation

ATTENDEES

1. Adelman, Kenneth A./adelman@tgv.com
2. Borman, Dave/dab@cray.com
3. Hedrick, Charles/hedrick@aramis.rutgers.edu
4. Karels, Mike/karels@berkeley.edu
5. LoVerso, John/loverso@xylogics.com
6. Mamakos, Louis A./louie@trantor.umd.edu
7. Mercado, Marjo F./marjo@hpindlm.hp.com
8. Reinstedler, Jim/jimr@ub.ubcom.com
9. Replogle, Joel/replogle@ncsa.uiuc.edu
10. Reynolds, Joyce K./jkrey@isi.edu
11. Roselinsky, Milt/cmcvax!milt@hub.ucsb.edu
12. Salo, Tim/tjs@msc.umn.edu
13. Schofield, Bruce J./schofield@edn-vax.dca.mil
14. Solensky, Frank/solensky@interlan.interlan.com
15. Vance, L. Stuart/vance@tgv.com
16. Westfield, Bill/billw@cisco.com
17. Wilder, Rick/rick@gateway.mitre.org
18. Wintringham, Dan/danw@osc.edu

MINUTES

Opening Comments:
Telnet Option draft RFCs - What are in the queue??

- Borman's Telnet Linemode: This is in the queue now for becoming an RFC. It has been handed off to Phill Gross.
- Berstein's Q-Method: For later discussion in this meeting, see item 6

Borman presented proposed agenda to group and asked what else should be included:
Bill Westfield lobbied for a document on Telnet with X.3 negotiations – he was overruled. It was decided that this along with item 10, was out of the scope of this group.
RFC 854 and Postel - Is there a justification for a "revised" Telnet spec?? There seemed to be general agreement that a better approach would be to answer all the other questions first, and that would decide this question for us.
The next item up for discussion was possible future options for Telnet that are needed.

Pursue??	What to include:
Yes	User Name (who you're going in as, i.e., name, acct, etc.)
Yes	Authentication (get rid of RLogin) (Authentication and encryption are somewhat related.)
Yes	Environment
Possibly	System Type
Yes	Encryption (Encryption and authentication are somewhat related.)
Maybe	Compression (data) (A subcase of encryption?? A maybe, depending upon encryption.)
Yes	don't Telnet Option (Bill Westfield working on this one.)

Big Topics:

Go through which Telnet options are not needed.
Send a message out to a mailing list asking who currently uses what telnet options. The following list is what we came up with at the meeting. Those marked with YES were changed from "no", those

marked with a ? no one was sure on. (This is re-constructed from memory, so please let me know if I made a mistake... -Dave B.)

Number	Name	RFC	NIC	DPH	USE
0	Binary Transmission	856	—	yes	yes
1	Echo	857	—	yes	yes
2	Reconnection	...	15391	yes	no
3	Suppress Go Ahead	858	—	yes	yes
4	Approx Message Size Negotiation	...	15393	yes	no
5	Status	859	—	yes	yes
6	Timing Mark	860	—	yes	yes
7	Remote Controlled Trans and Echo	726	39237	yes	no
8	Output Line Width	...	20196	yes	no
9	Output Page Size	...	20197	yes	no
10	Output Carriage-Return Disposition	652	31155	yes	no
11	Output Horizontal Tabstops	653	31156	yes	no
12	Output Horizontal Tab Disposition	654	31157	yes	no
13	Output Formfeed Disposition	655	31158	yes	no
14	Output Vertical Tabstops	656	31159	yes	no
15	Output Vertical Tab Disposition	657	31160	yes	no
16	Output Linefeed Disposition	658	31161	yes	no
17	Extended ASCII	698	32964	yes	no
18	Logout	727	40025	yes	no
19	Byte Macro	735	42083	yes	no
20	Data Entry Terminal	732	41762	yes	no ?
21	SUPDUP	734 736	42213	yes	no
22	SUPDUP Output	749	45449	yes	no
23	Send Location	779	—	yes	no
24	Terminal Type	1091	—	yes	YES
25	End of Record	885	—	yes	no
26	TACACS User Identification	927	—	yes	no ?
27	Output Marking	933	—	yes	no
28	Terminal Location Number	946	—	no	no
29	3270 Regime	1041	—	no	no ?
30	X.3 PAD	1053	—	no	no
31	Window Size	1073	—	no	YES
32	Terminal Speed Option	1079	—	no	YES
33	Remote Flow Control	1080	—	no	YES
34	Linemode	TBA	—	no	YES
35	X Display Location	1096	—	no	no
255	Extended-Options-List	861	—	yes	yes

Clarifying Timing Mark RFC
Does anyone use STATUS??

What's wrong with the current Telnet spec:

- old stuff
- what to update

Other Issues:
Borman's concept of the new Telnet Working Group:

> This group is not to disband, but upon completion of their activities, go dormant from time to time, and start up and become available as a group to review Telnet draft RFCs, etc....as needed.

Discussion/Issues of 7 bit, 8 bit binary:

1. Delay problem between client and server, interrupt character, interrupt systems, interrupt marker - Linemode really helps you here in this realm.
2. Interrupt - telnet process can control things, output prompt between the two.
3. Host Requirement RFC document - discussion regarding "clean wording" of Telnet in the Host Requirement RFC. In particular, a statement on 7, 8 bit data passing; 8 bit should NOT be used for parity bit.
4. Should anything be said in the Host Requirement RFC re: 7, 8 bit?? What about the statement of "SHOULD or MUST" negotiate binary??
5. Should the Telnet standard be changed/updated to reflect the context of Host Requirements RFC??
6. Items d and e were not resolved at this meeting. There is a need to soften the wording on the Telnet statement that's going into the Host Requirements RFC. Borman to talk to Braden.
7. Bernstein's Q-Method RFC. Postel asked the Telnet WG to review and comment. Group comment is that it should not be issued as an RFC. Part of it should be rewritten, and incorporated with whatever we release for a replacement/update to the Telnet RFC. It was felt that the real world was not having problems with option negotiation loops, so it isn't a problem that requires an immediate solution.

Conclusion of meeting:

- Telnet WG will meet in Hawaii.
- Interim discussions will continue on the

Telnet

7/27/89

1st meeting (this is NOT Linemode)

Goals: Revisit RFC 854 & friends
Clearing house for new options
(get rid of rlogin)

New options:
- Username (account name, other info)
- Authentication (e.g. Kerberos)
- Generic Environment passing
- System Type
- Encryption
- Data Compression
- Don't Telnet Anymore

Data flushing: use TIMING-MARK in client, server should send SYNCH at appropriate times

Binary: Some people really need 8bit NVT
Need to soften HR statement

Option Negotiation Loop avoidance:
Should be documented, not critical.

USER-DOC Working Group
Chairpersons: Tracy LaQuey/Univ of Texas and Karen Roubicek/NSF

CHARTER

Description of Working Group:

> The USER-DOC Working Group will prepare a bibliography of on-line and hard copy documents/reference materials/training tools addressing general networking information and "how to use the Internet". (Target audience: those individuals who provide services to end users and end users themselves.)

Specific Objectives:

1. Identify and categorize useful documents/reference materials/training tools.
2. Publish both an on-line and hard copy of this bibliography.
3. Develop and implement procedures to maintain and update the bibliography. Identify an organization or individuals to accept responsibility for this effort.
4. As a part of the update process, identify new materials for inclusion into the active bibliography.
5. Set up procedures for periodic review of the biblio by USWG.

Estimated Timeframe for Completion:

- Format for the bibliography will be decided upon by the July IETF session, as well as identification of "sources of information" (e.g. individuals, mailing lists, bulletins, etc.)
- Draft bibliography will be prepared by mid-December 89.

STATUS UPDATE

1. Chairpersons: Tracy LaQuey, tracy@emx.utexas.edu
   Karen Roubicek, roubicek@nnsc.nsf.net
2. WG Mailing List: user-doc@nnsc.nsf.net
3. Date of Last Meeting: Stanford IETF / 25 July 1989
4. Date of Next Meeting: Hawaii IETF / October-November 1989
5. Pending or New Objectives:

   Produce first draft of bibliography by Hawaii IETF.
6. Progress to Date (e.g., documents produced):

   Several documents have been collected, categories chosen, preliminary format selected.

USER-DOC Working Group
Chairpersons: Tracy LaQuey/Univ of Texas and Karen Roubicek/BBN,NNSC

CURRENT MEETING REPORT

Reported by Karen Roubicek

AGENDA

- Review Charter, Objectives, Timeframe
- Review and revise bibliography outline
- Review the documents we have to-date and make additions
- Clarify location and maintainer of bibliography as it is being written
- Define format, contents of entries
- Identify volunteers to write abstracts
- Define the review process for selecting documents for inclusion in bibliography.
- Determine where bibliography will live after initial publication and discuss mainte nance/review/update in the future
- Identify sources for information and determine who will take responsibility for solic iting it.
- Clarify how to distribute bibliography to users

ATTENDEES

1. Armstrong, Karen, armstrongk@sds.sdsc.edu
2. Bowers, Karen, kbowers@nri.reston.va.us
3. Breeden, Laura, breeden@bbn.com
4. Easterday, Tom, tom@nisca.ircc.ohio-state.edu
5. Enger, Robert M., enger@sccgate.scc.com
6. Finkelson, Dale, dmf@westie.unl.edu
7. Hallgren, Martyne M., martyne@tcgould.tn.cornell.edu
8. LaQuey, Tracy, tracy@emx.utexas.edu
9. Marine, April, april@nic.ddn.mil
10. Miller, Stephen, miller@m2c.org
11. Moore, Berlin, bm24@andrew.cmu.edu
12. Morris, Don, morris@ncar.ucar.edu
13. Oattes, Lee, oattes@utcs.utoronto.ca
14. Perillo, Francine, perillo@cisco.com
15. Pleasant, Mel, pleasant@rutgers.edu
16. Pugh, Jon, pugh@nmfecc.llnl.gov
17. Redfield, Elizabeth, red@nic.ddn.mil
18. Reynolds, Joyce K., jkrey@venera.isi.edu
19. Roubicek, Karen, roubicek@nnsc.nsf.net
20. Sitzler, Dana, dds@merit.edu
21. Stahl, Mary, stahl@nic.ddn.mil
22. Steinberg, Lou, louiss@ibm.com
23. Sweeton, Jim, sweeton@merit.edu
24. Veach, Ross, rrv@seka.cso.uiuc.edu
25. Wintringham, Dan, danw@osc.edu
26. Yuan, Aileen, aileen@gateway.mitre.org

MINUTES

A few members of the User Services Working Group, who were attending a FARNET meeting on May 31, met with Karen Bowers for a short time to form a distinct group to create the userdoc bibliography. The July 25 IETF session therefore was the first formal meeting of the USER-DOC Working Group.

The meeting began with a review of the charter of the USER-DOC Working Group. To summarize, the purpose of the group is to prepare a bibliography of online and hardcopy documents, reference materials, and training tools addressing general networking information and "how to use the Internet". End users and people who help end users are the targeted audience. The group had already identified several documents since the first USWG meeting in Texas and described some broad categories to cover prior to this meeting. We had originally set mid-December 1989 as the publication date of the bibliography but by the end of this meeting we agreed to issue a first draft by the Hawaii IETF.

A discussion of subject areas to be covered in the bibliography identified the following topics:

- Introduction to TCP/IP
- Guide for network administrators
- Electronic Mail
- Services Documents (Directories, Libraries)
- Mailing Lists
- RFCs
- Map Collections (pointers to)
- Workshops and Conferences
- Servers
- Security
- Newsletters

This list of categories is not exhaustive, nor is it set in stone. Tracy suggested that including indexes of keywords will help us out with documents or materials that will be difficult to fit into specific categories. Several members felt that where possible, entries should have a designated level of expertise associated with them. The group agreed that it would only be realistic to rank a document as "beginner" or "novice" and "intermediate and above".

Don Morris expressed an interest in listing helpful tools or applications, such as the NCAR Internet Remote Job Entry System. The group decided that although the bibliography was not the right place to include such references, another working group could be formed for that purpose.

An important feature of the bibliography is that it be comprehensive enough to be useful but not so large as to be overwhelming. Toward this goal, Joyce Reynolds will provide a list of key RFCs to serve as a basic set. We identified a category for documents that describe how to use available servers, particularly those servers on non-Internet networks such as BITNET and SPAN. Since large numbers of users on the Internet want to use these facilities, the group decided to include such descriptive documents as are available and concluded that some may have to be written specifically for the bibliography.

The question of including articles from periodicals was raised and some members expressed concern that it would be difficult to keep up with all journal articles and that the bibliography might become unwieldy. We agreed that unless an article were very unusual (of the variety: Notable Computer Networks by John Quarterman), we would not include it. A special issue of a periodical devoted entirely to one particular topic or issue may be included.

We reviewed the current list of references and added some new ones. The materials referenced in the bibliography will not be limited to text, but will include audiotapes, videotapes and regularly scheduled workshops and conferences.

Tracy is currently maintaining the bibliography at emx.utexas.edu and will continue to do so through the first draft in October. The group will decide on the future of the bibliography at the next IETF meeting.

We provisionally chose Refer as a format for the bibliography. Standard bibliographic requirements will be included (name, author, keywords, etc) as well as some additional fields such as copyright information, length, order source/pathname, version, obsoletes/updates, type of media and format. Elizabeth Redfield and April Marine from SRI discussed the current work on a NIC bibliography database. Their efforts are considerably larger in scope than ours (current number of documents is greater than 2,000). The importance of keeping in touch with developments in the library community and documentation retrieval technology was mentioned. Laura Breeden referred to CMU's Andrew project and Karen Bowers cited NRI's Digital Library project. Two committees were established to facilitate the publication process: an "editorial board" (Jon Pugh, Karen Bowers, Tracy LaQuey, Francine Perillo, Joyce Reynolds, and April Marine) will review the entries for appropriateness and accuracy; and Dale Finkelson, Jon Pugh and representatives from the DDN NIC will constitute a group which will do the work to flesh out the entries.

To facilitate the discussion about format and obtaining further bibliographic references, the attendees formed two groups to go off and pursue those issues and reconvene after an hour. April and Elizabeth had expressed some concern over the compatibility of the USER-DOC bibliography in Refer with the NIC's bibliographic efforts, so the group decided that the NIC would include the USER-DOC entries in their database. They will also pass on their template for possible use by the USER-DOC WG. The group focusing on obtaining entries for the bibliography identified a list of individuals, groups and organizations, and mailing lists from which we will solicit information. Members of the working group will take advantage of meetings that we attend to gather information from the participants, a specific example being the IETF meeting itself. We decided that together with members of the NOC-Tools Working Group we would compose a brief handout requesting material for both the NOC-Tools catalog and the bibliography. This was passed out at the Plenary session on Thursday morning with a request to return suggestions that afternoon. We received a handful of contributions.

In discussing how to effectively distribute the bibliography to users, we concluded that the issue has relevance beyond just the USER-DOC Working Group, and Karen Bowers volunteered to set up a separate session on Thursday under the USWG umbrella to investigate methods of distribution (see USWG report).

ACTION ITEMS:

- Write up and pass out questionnaire at Plenary Session: Enger, Roubicek, Bowers
- Develop Template (LaQuey, Marine, Redfield, Roubicek)
- Define set of key RFCs (Reynolds)
- Research Andrew system and Federal databases (Breeden)
- Contact Paul re: SOCRATES project (Breeden)
- Choose set of mailing lists (Roubicek, Bowers)
- Liaison with library community (Roubicek)
- Schedule distribution meeting (Bowers)
- Write article about bibliography for ConneXions soliciting information (Perillo)

PRELIMINARY AGENDA for next meeting:

- Review first draft of bibliography
- Make final choice of format
- Establish permanent location(s) and maintainer(s) for bibliography

USER-DOC

CHARTER

The USER-DOC Working Group will prepare a bibliography of online and hardcopy documents/reference materials/training tools addressing general networking information and "how to use the Internet".

Target Audience: those individuals who provide services to end users and end users themselves.

SPECIFIC OBJECTIVES

1. Identify and categorize useful documents/reference materials/training tools.
2. Publish both an online and hardcopy version of this bibliography.
3. Develop and implement procedures to maintain and update the bibliography. Identify an organization or individuals to accept responsibility for this effort.
4. As a part of the update process, identify new materials for inclusion into the active bibliography.
5. Set up procedures for periodic review of the bibliography by US-WG

TIMEFRAME

1st Draft bibliography will be prepared by next ietf ~~mid-December~~ 1989

SUBJECT AREAS

- INTRODUCTION TO TCP/IP
- GUIDES FOR NETWORK ADMINISTRATORS
- ELECTRONIC MAIL
- SERVICES DOCUMENTS
- MAILING LISTS
- RFCs
- MAP COLLECTIONS
- WORKSHOPS/CONFERENCES
- SERVERS
- SECURITY
- NEWSLETTERS

MEDIA

- ARTICLES
- BOOKS
- JOURNALS
- VIDEOTAPES
- AUDIOTAPES

MECHANICS

REVIEW BOARD
RESEARCH BOARD

USER-DOC@NNSC.NSF.NET

FORMAT

- INTERIM - REFER
- COORDINATE WITH SRI

LOCATION

EMX.UTEXAS.EDU
in user.wg/documents

SOURCES FOR INFORMATION

- INDIVIDUALS
- GROUPS/ORGANIZATIONS
- MAILING LISTS
- OTHER BIBLIOGRAPHIES

DISTRIBUTION

User Services Working Group
Chairperson: Karen Bowers/NRI

CHARTER

Description of Working Group:

The User Services Working Group will identify and address critical service requirements needed by "those people who help end users" (e.g. local net managers) and develop tools and materials to aid in the productivity of end users. The purpose is to answer the needs of the lower levels (*) within this hierarchy:

NATIONAL NETWORK
NET MANAGERS (NSF, DCA, ETC.)
NICs/NOCs
REGIONAL NET MANAGERS
LOCAL NET MANAGERS*
END USERS*

Specific Objectives:

1. Assemble a non-static cadre of interested experts within an open forum to exchange user services information, to share problem-solving techniques, and to select critical projects to be undertaken on behalf of the local net manager and end user.
2. Select projects based on production-oriented criteria. The Project(s)
    - must lend itself to accomplishment within a reasonable timeframe
    - must culminate in a measurable/quantifiable end result
    - must address user assistance needs = be user oriented
    - must yield products/tools designed to be both easily maintained and updated (with built in accountability)
    - must not duplicate efforts (This will be pre-empted by surveying exisitng resources.)
3. Determine the most appropriate approach to a respective project (s):
    - produce a totally new product
    - enhance/improve/influence an existing resource
    - table action for future consideration
4. Spin off various small WGs (tiger teams) to address very specific, short term projects (EX: NOC-Tools WG and NISI WG). Once the respective project(s) is completed, members of the tiger team(s) will reassemble within the USWG to participate in the identification of the next project(s) to be undertaken.

Estimated Timeframe for Completion:

Selection and completion of projects will occur on a continuous basis, with timelines established for each individual tiger team formed.

STATUS UPDATE

1. Chairperson: Karen L. Bowers
   kbowers@nri.reston.va.us
2. WG Mailing List(s): US-WG@NNSC.NSF.NET and
   US-WG-REQUEST@NNSC.NSF.NET
3. Date of Last Meeting: Stanford University, 25-28 July 1989
4. Date of Next Meeting:
    - Interim Meeting Planned Mid-September (VT Conference)
    - Quarterly Meeting: Hawaii, 31 Oct - 3 Nov 1989
5. Pending or New Objectives:

   1) As part of the recently announced restructuring of the IETF, transition the current USWG into the forthcoming User Services Area. 2) Announce the soon-to-be-completed bibliography and catalog to users via bulletins, mailing lists and liaisons with other established organizations, such as FARNET, SIGUCCS, EDUCOM, etc.
6. Progress to Date (e.g., documents produced):

   Drafts are underway in each of the User Services subgroups. The USERDOC bibliography first draft will be ready 3 November 1989; the first edition NOC-Tools catalog will be completed by 12 December and ready for publication; and the NISI Requirements Document draft outline is in-progress.

User Services Working Group
Chairperson: Karen Bowers/NRI

CURRENT MEETING REPORT

Reported by Karen Bowers and Martyne Hallgren

AGENDA

- Welcome New Members
- Review Current Projects in Progress/Planned
- Brief Discussion of Ralph Drom's Universal Directory Service for the Internet
- Participation in USER-DOC, NISI and NOC-Tools WGs
- Reconvene to Discuss Issues of Distribution for the NOC- Tools catalog, USER-DOC Biblio and other USWG Undertakings

ATTENDEES

1. Bowers, Karen/kbowers@nri.reston.va.us
2. Breeden, Laura/breeden@bbn.com
3. Easterday, Tom/tom@nisca.ircc.ohio-state.edu
4. Enger, Robert M./enger@sccgate.scc.com
5. Finkelson, Dale/dmf@westie.unl.edu
6. Gerich, Elise/epg@merit.edu
7. Hallgren, Martyne M./martyne@tcgould.tn.cornell.edu
8. Hastings, Gene/hastings@morgul.psc.edu
9. Jacobsen, Ole/ole@csli.stanford.edu
10. Kincl, Norman/kincl@iag.hp.com
11. LaQuey, Tracy/tracy@emx.utexas.edu
12. Malkin, Gary/gmalkin@proteon.com
13. Marine, April/april@sri-nic.arpa
14. Miller, Stephen/miller@m2c.org
15. Moore, Berlin/bm24@andrew.cmu.edu
16. Morris, Don/morris@ncar.ucar.edu
17. Mundy, Russ/mundy@tis.com
18. Oattes, Lee/oattes@utcs.utoronto.ca
19. Pak, Raylene/raylene@tardis.tymnet
20. Partridge, Craig/craig@nnsc.nsf.net
21. Perillo, Francine/perillo@cisco.com
22. Pleasant, Mel/pleasant@rutgers.edu
23. Pugh, Jon/pugh@nmfecc.llnl.gov
24. Redfield, Elizabeth/red@sri-nic.arpa
25. Reynolds, Joyce K./jkrey@venera.isi.edu
26. Roberts, Mike/roberts@educom.edu
27. Roberts, Ronald/roberts@jessica.stanford.edu
28. Roubicek, Karen/roubicek@nnsc.nsf.net
29. Schoffstall, Martin/schoff@nisc.nyser.net
30. Sitzler, Dana/dds@merit.edu
31. Sollins, Karen/sollins@lcs.mit.edu
32. Stahl, Mary/stahl@sri-nic.arpa
33. Steinberg, Lou/louiss@ibm.com
34. Stine, Robert/stine@sparta.com
35. Sweeton, Jim/sweeton@merit.edu
36. Veach, Ross/rrv@seka.cso.uiuc.edu
37. Wintringham, Dan/danw@osc.edu
38. Youssef, Mary/mary@ibm.com
39. Yuan, Aileen/aileen@gateway.mitre.org

MINUTES

The purpose of the fairly brief USWG session on Tuesday morning, 25 July 1989, was to welcome new members and acclimate them to the recently established activities of User Services. A brief review of the current charter and organizational structure was provided and a quick synopsis given on the three currently active USWG projects: NISI (Network Information Services Infrastructure), User-Doc and NOC-Tools. During this IETF plenary, full concentration of efforts was placed on the three ongoing projects scheduled the remainder of Tuesday and all day Wednesday. (Separate meeting reports covering those activities have been prepared by the respective Chairs and are enclosed within the July 1989 Proceedings and the IETF: directory.) A new item brought to light at the close of this short session was an Internet Directory Service (IDS) being proposed by Ralph Droms (NRI). Discussion on the immediate impact of this IDS and the associated implications is to be continued during the next (interim) USWG meeting, at which Ralph Droms has agreed to provide a presentation and field audience questions.

On Thursday morning, 27 July 1989, the User Services Working Group reconvened to discuss the issues of distribution of Internet information such as the NOC-TOOLS catalog and the USER-DOC bibliography.

As a way to approach this complex issue, the group spent some time defining the user community it was trying to reach, the goals of the distribution effort and what methods were currently in use or could be used.

The audience is extremely broad. It includes end users (defined as someone who uses the network as a tool, such as a researcher), site support staff including user, network or technical, and administrative support (site being defined as an academic, industrial, or government organization), regional and backbone network providers, political groups such as state or federal legislatures, and specialized groups. These specialized groups may be associated with a specific discipline or interest or computer vendor. Such groups include EDUCOM, SIGUCCS, SHARE, DECUS, RLG, OCLC, and the American Physical Society.

The goal of distributing information to this audience is to share information and to provide guidelines on using and supporting the Internet. The information can educate or be used as reference material to both new and old participants in the Internet community. The USER-DOC bibliography is useful to the entire audience. The NOC-TOOLS catalog is targeted more towards those groups which do technical management of a network.

At present, the main method of information distribution on the Internet is via personal networking, i.e., "the old boy" system; to find out something, one asks somebody else if they know the answer or who else to ask. This personal networking will always continue. In addition, MERIT and the NSF Network Service Center (NNSC) work to provide information on specific areas. (While not mentioned in the discussion, SRI-NIC should also be included with MERIT and NNSC). As the regional networks have blossomed, a rough hierarchy has formed, where information flows vertically (generally from a top-level "NIC" to a mid-level or regional organization to some number of sites) and horizontally (between the organizations at each level of the hierarchy). Specialized groups tend to share information amongst their own members, without regard to any hierarchy.

The basic issue then, is to define or open new paths of communication through which a broad audience can be reached and be provided with the how's and why's of finding information about the Internet. There are several possibilities on how to do this. Organizations such as SIGUCCS, SIGCOMM, IEEE, ACE, Nysernet, MERIT, and NSF use conferences, workshops, and publications as information distribution mechanisms. Vendors/industry have both formal (documentation) and informal means of sharing information with their customer and internal user base. Newsletters and trade publications also provide opportunities to share information.

The role of IETF User Services in distributing information, such as the NOC-Tools catalog and USER-DOCS bibliography, is to make use of communication avenues already developed by other organizations. This can be accomplished through liaison relationships with other organizations such as EDUCOM and SIGUCCS and use of existing distribution avenues such as the NNSC Bulletin.

User Services plans the following activities. An announcement of the IETF User services working group and how to participate will be made at the EDUCOM, SIGUCCS, and INTEROP conferences this fall (Martyne Hallgren, Karen Roubicek and Karen Bowers). There will be announcements of the USER-DOC bibliography and the NOC-Tools catalog via the following avenues: NNSC bulletin (with a special mailing to the Computer Center directory list), on-line mailing lists, and through FARNET. Karen Bowers (chr), Karen Roubicek, Mary Stahl, and Gene Hastings will discuss and chose what mailing lists shall be used for distribution. Martyne Hallgren (chr), Laura Breeden, Karen Roubicek, and Dale Finkelson will pursue the liaison relationships with other organizations, including EDUCOM, SIGUCCS, NSF, and FRICC.

Additional Action Items:

- Connection Checklist (Laura Breeden and Craig Partridge)
- BOF at Interop 89
- Contact PM at NSF for the Outreach program

Other Items:

- Dan Wintringham has raised the issue of establishing a WG to address Configuration Tools.
- Tom Easterday is interested in defining procedures on 'How to Set-Up A Campus NIC/NOC', an activity in which Tracy LaQuey expressed interest earlier.
- If concentrated help in future projects is required, Joyce Reynolds has kindly volunteered to Chair a WG as needed.

Audience

1

End users

Site Support (Academia
Industry
Government)

User Support
* Technical Support
Admin Support

* Regional Networks

* Backbone Networks

Political Entities

Specialized Groups (discipline
vendor)

SIGUCCS
EDUCOM
SHARE
SUN usergroups
DECUS SIGs
Amer. Phy Society

all Docs to ALL
NOC Tools

libraries
RLG
OCLC

Purpose: Distribution of Information | NOC tools

DEFINE USER Community

DEFINE GOALS of EFFORT | Guidelines
Dore to Science

METHOD of. Distribution

new user
Education/Refer
Sharing

Current | "old boy" Network | merit NNSC

Others

industry | vendors

Library Systems

NIC liasons

Interops (BOF)
Workshops
Sigucss
Sigcomm
IEEE

Nipernet
Merit
ACE
Allumni?
NSF Outreach

newsletters (canned Articles)

Trade papers

CCR

Connections

BBNIC
reg
SIG

☺

where it is
"marketing" of knowledge
hierarchy
NICS [what does it mean]
specialized groups (Forgotten transfer)

7-27-89
12:00 Noon

board ②

Stuff from 2nd Black board at USWG meeting

Role of IETF User services
- Liaison Relationships w/ organizations
  Public Acknowledgement
  Limited lifespan of W.G. ⇒ Evolution

<u>Mechanics</u> of effort

{information to "Avenues"
{technical contact for each Registered Network

* NNSC Bulletin: 3000-4000 distribution

* Announcement of User Services

Karen B. { Educom
Maryne { InterOp
SIGUCCS
(all Avenues)

③

Activities: Announcement of Availability /

Time lines:
NOCTools 12 Dec
User DOC 3 NOV
* Be ready by Documen
Suspense i

1) Announce 2 Documents thru Following Avenues

A. NNSC Bulletin [Special Mailing to CS Directory]

B. On Line Mailing Lists: Merit, DDN-NIC, NNSC, TCP/I
Byte-counters, NWG, Federation,
IETF, IAB, RFC, BIG-LA
Info-Nets, DDN Mgt Bulletin,
(CERT?)

C. FARNET

Karen Roubicek
Karen* Bowers
Mary Stahl
Gene Hastings

D. Liaisons w/ other organizations
Educom, SIG UCCS, NSF, FRICC
↓
etc.

Laura Breeden ✓
Martyne Hallgren* ✓
Karen Roubicek
Dale Finkelson ✓

USWG

Act One

Welcome/Acclimate New Members
- Review Current Charter/Structure
- Introduce Current Activities
  - NISI (infrastructure)
  - USER-DOC (bibliography)
  - NOC-Tools (catalog)
- Disuss Other Projects Planned
- ~~Internet Directory~~ Service (R. Droms)

Act Two

" Rolled up Sleeves "

NISI, USER-DOC, NOC-Tools

Act Three

Distribution of Information (Martyne Hallgren)
- Defined User Community/Audience
- Discussed Methods/Avenues for Distribution
- Examined Actual Mechanics for Announcing Availability of Biblio/catalog
- Formed "Tyger Teams"
  - Bulletins, Special/On-Line Mail. Lists
  - Liaisons w/ Other Key Organizations

VI. Network Status Briefings
and
Technical Presentations

Status of NSFNET Backbone

Presented By Elise Gerich

Merit and her joint partners, IBM and MCI, have focused their energies on three major projects during the last three months:

- The redesign of the NSFNET topology and architecture
- Connections with peer networks
- Border Gateway Protocol

The backbone that Merit, MCI, and IBM put in place in July 1988 consisted of 14 T1 circuits connecting 13 nodes. Of those 13, 6 of them were connected to the backbone by a single tail circuit, and the T1 links were sub-channeled into 448 Kb channels.

The plan that was developed to address these issues would:

- eliminate the single tail circuits
- provide a full T1 rate end-to-end bandwidth
- provide a low network diameter (3)

The implementation of this plan involved the deployment of new hardware and software as well as the installation of additional T1 circuits, and was completed by mid-July. As of July 18, 1989, the NSFNET backbone was routing traffic at full T1 rates and each node on the backbone had multiple T1s terminating at the node.

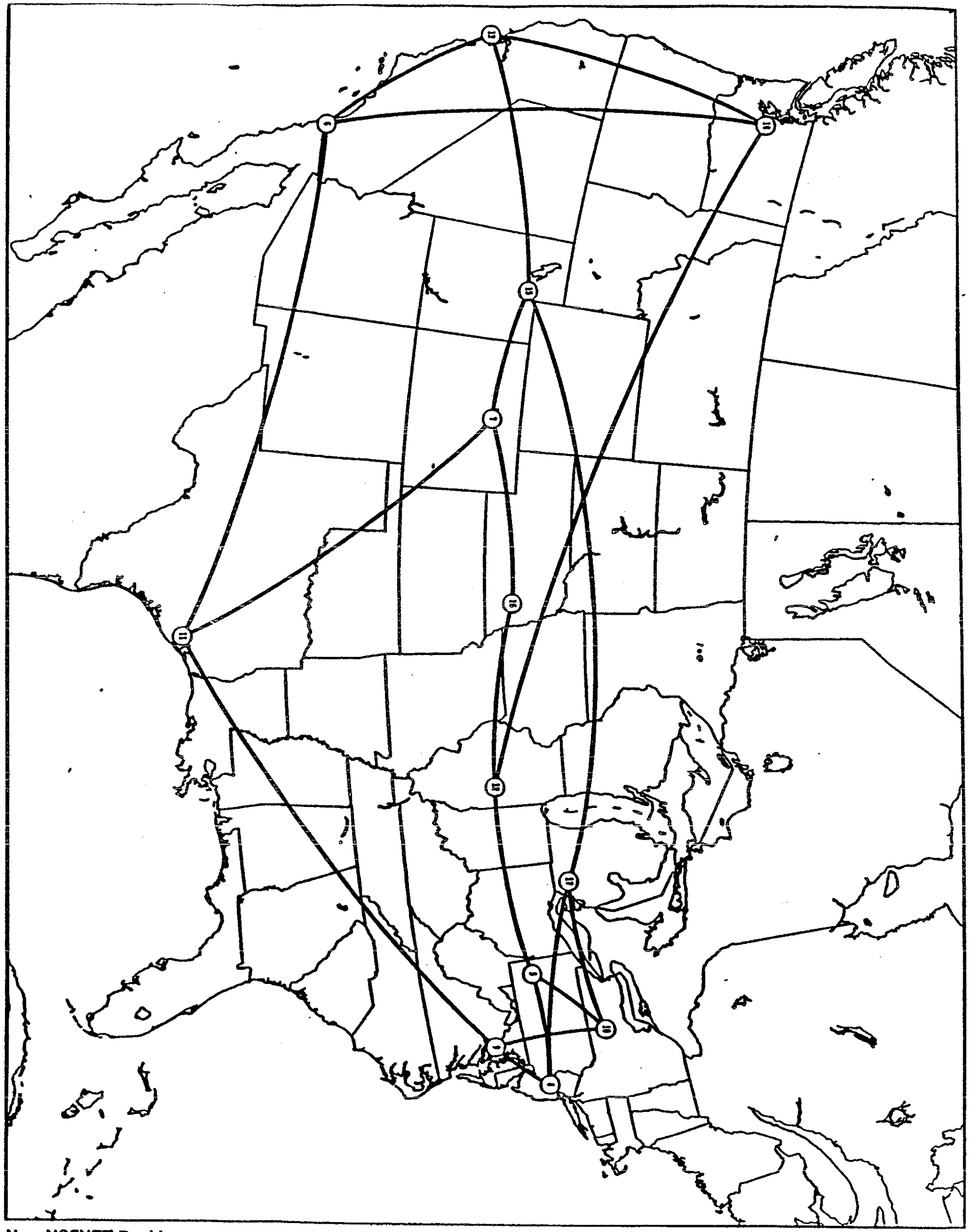
This redesign of the architecture of the backbone also lays the foundation for implementing Digital Reconfiguration Services (DRS) as it becomes available. All the circuits in the new topology were engineered to pass through MCI's Digital Cross Connects (DXC) which are specially equipped to support DRS.

In addition to completing the redesign of the backbone, Merit, NASA, DCA, and DARPA have been working to provide direct peer network connections between NSFNET, Milnet, and NSN. Two backbone nodes were targeted as the intial nodes to provide these connections, Palo Alto (BARRNet) and College Park (SURAnet). The first direct connection between the NASA Science Network and NSFNET was in place on July 15, 1989 at College Park, MD. Following shortly thereafter, we established a direct connection with the mailbridge at NASA/AMES by implementing the Split E-PSP configuration at Palo Alto, CA.

We are in the process of establishing another direct connection with the Milnet/ARPANET at College Park, MD. This will provide the NSFNET with both an east coast and west coast connection to the Milnet/ARPANET.

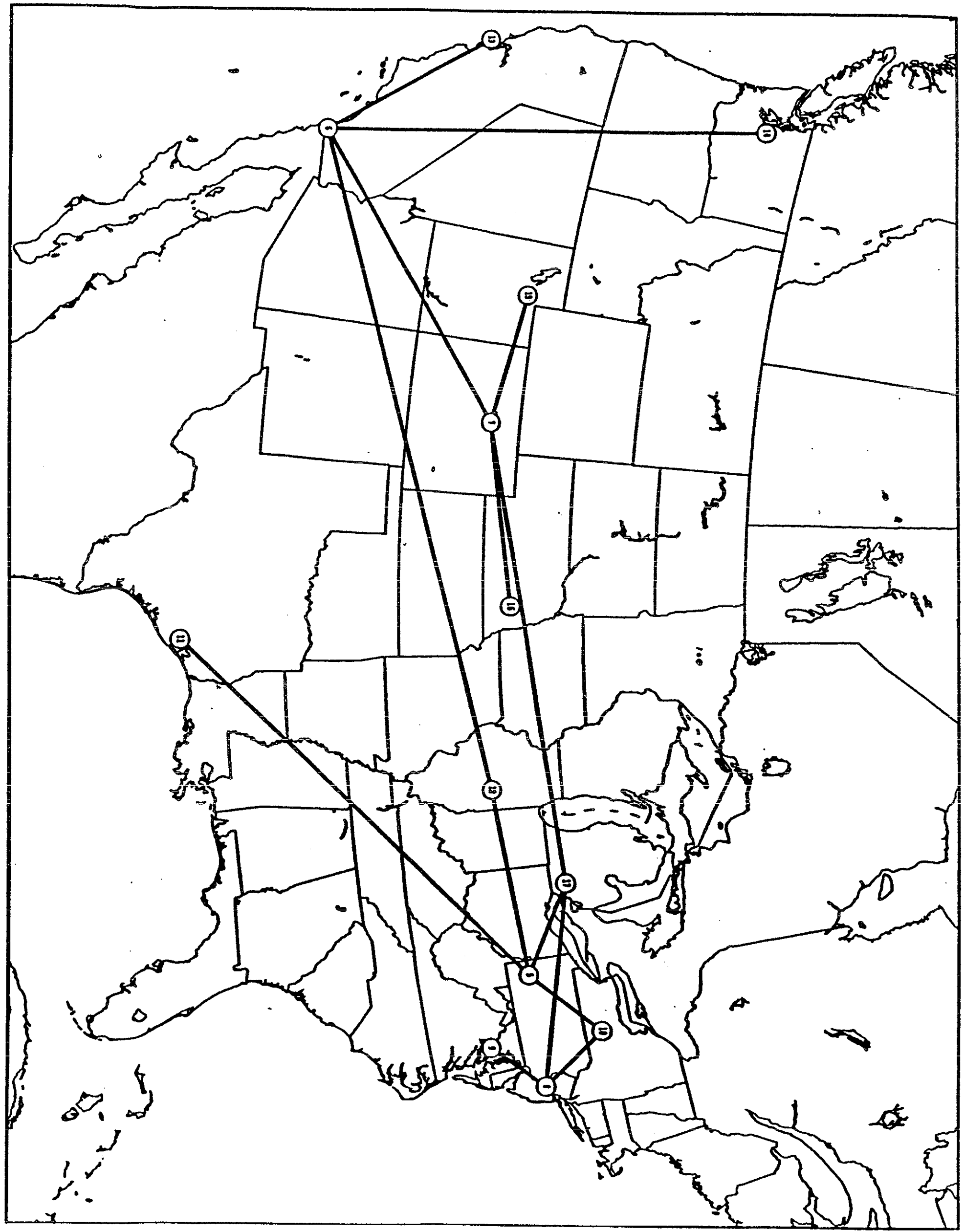
Also, development of a protocol and its implementation to provide an alternative to EGP has been under way. IBM, cisco, Cornell, and Merit, along with a few other members of the Internet community have designed the Border Gateway Protocol. The Border Gateway Protocol is documented in RFC 1105. Other related documents are RFC 1104 and Midterm Inter AS Routing Architecture (MIRA). Two documents concerning BGP Usage and Routing Domains are currently in draft form.

BGP is being tested on the NSFNET Research Network, and currently there are 3 implementations: gated, cisco, and nss.



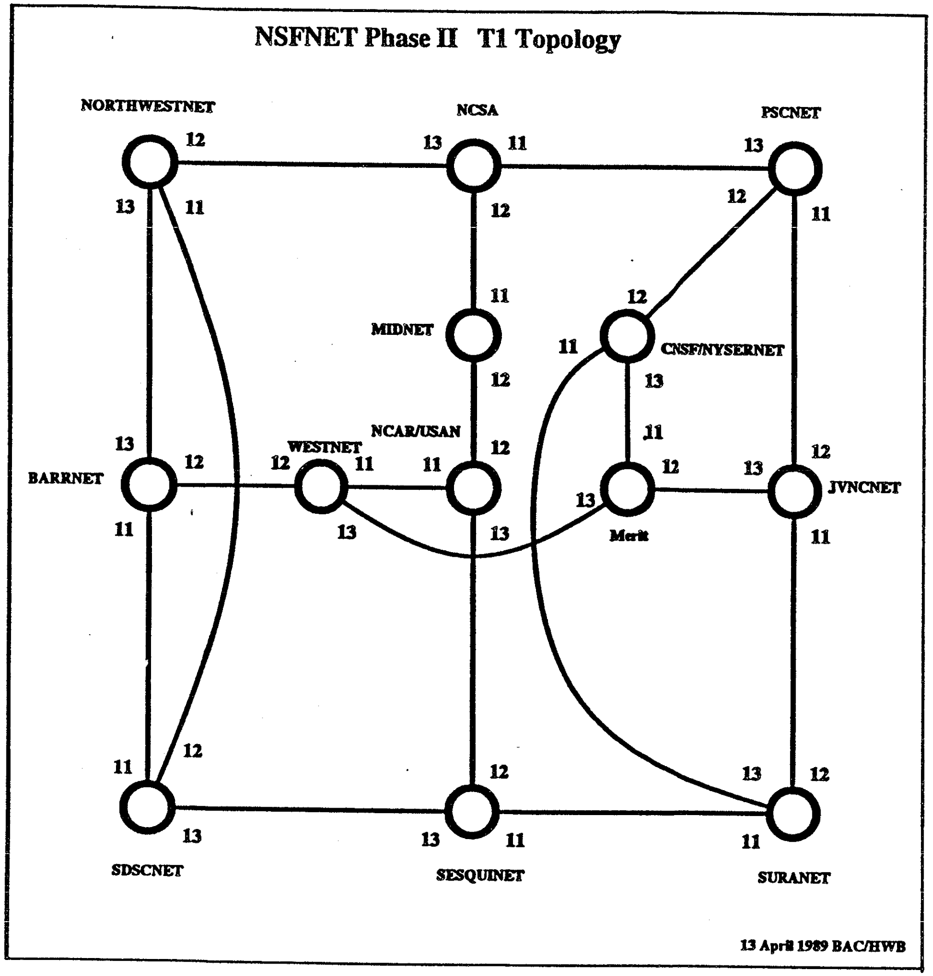
New NSFNET Backbone
Circles contain NSS number

Prepared by NSFNET-info@merit.edu at Fri Feb 24 13:58:52 1989
netmap-1.5 program by Brian Reid, map data from World Data Bank II
Lambert Conformal Projection [44°N,33°N], Map center: [40°N, 96° 30'W]
Image resolution 300/in., stroke limit 1 pixels



Physical NSFNET Topology
Circles contain NSS number

Prepared by NSFNET-info@merit.edu at Mon Mar 6 10:15:10 1989
netmap-1.5 program by Brian Reid, map data from World Data Bank II
Lambert Conformal Projection [44°N,33°N], Map center: [40°N, 96° 30'W]
Image resolution 300/in., stroke limit 1 pixels


NSFNET Phase II T1 Topology

13 April 1989 BAC/HWB

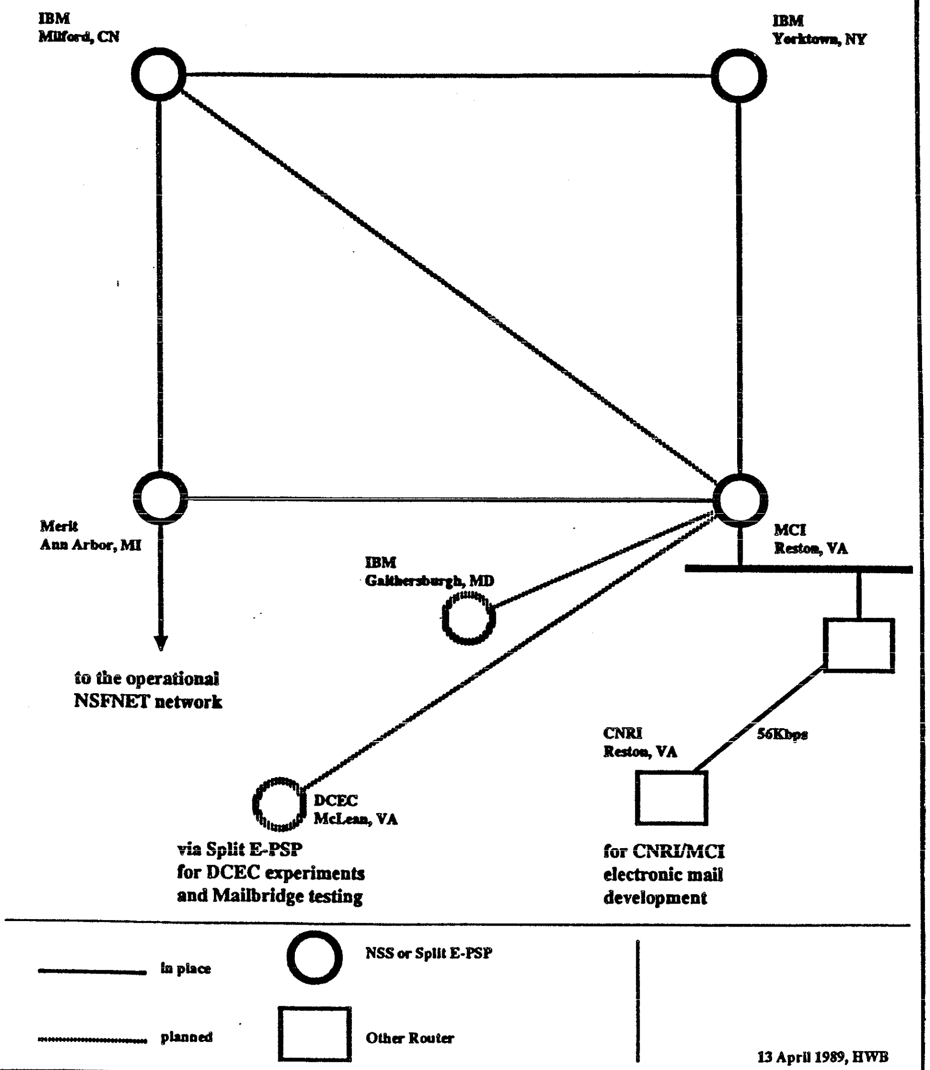
NEW TOPOLOGY

- " Fatter " pipes for packet switching
- No multiplexing and demultiplexing at sub T1 rates

  Clear channel T1
- Greater redundancy
  - No single tail circuit sites
  - No articulation points
- Optimized for MCI infrastructure
  - MCI redundancy
  - Optimum MCI routes
- Greater degree of connectivity

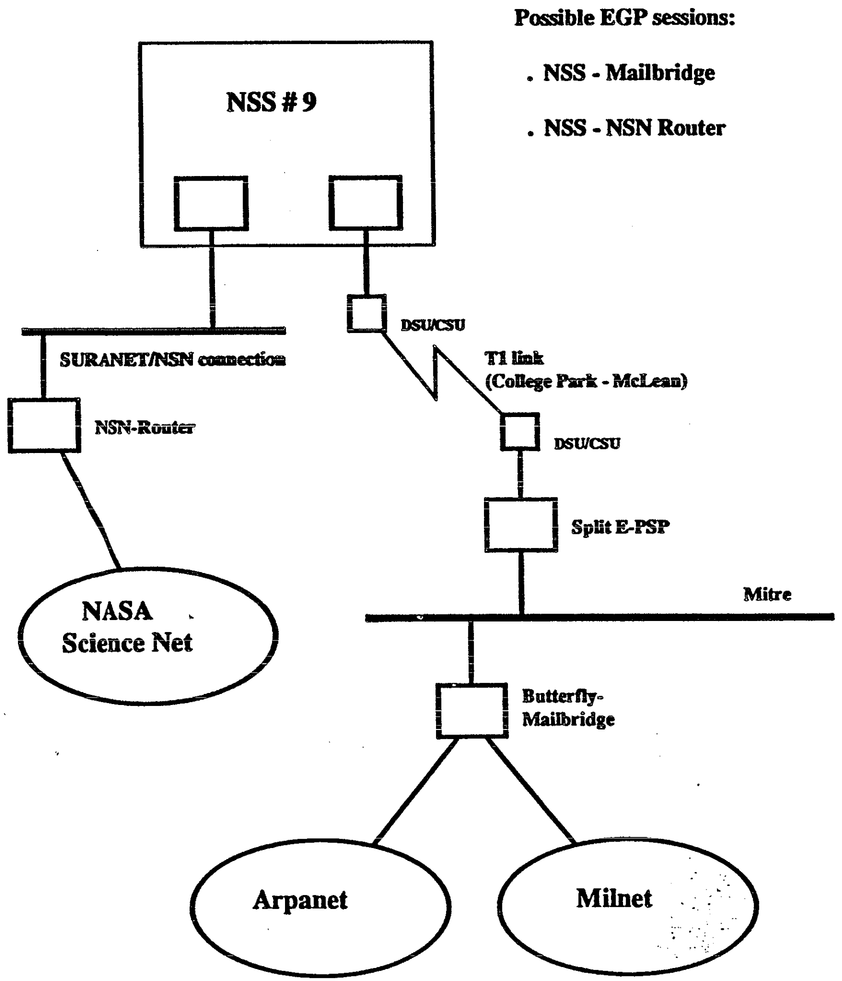
  3.07 v/s 2.15

NSFNET Research Network



13 April 1989, HWB

Planned NSFNET/NSN/DDN connection at the University of Maryland

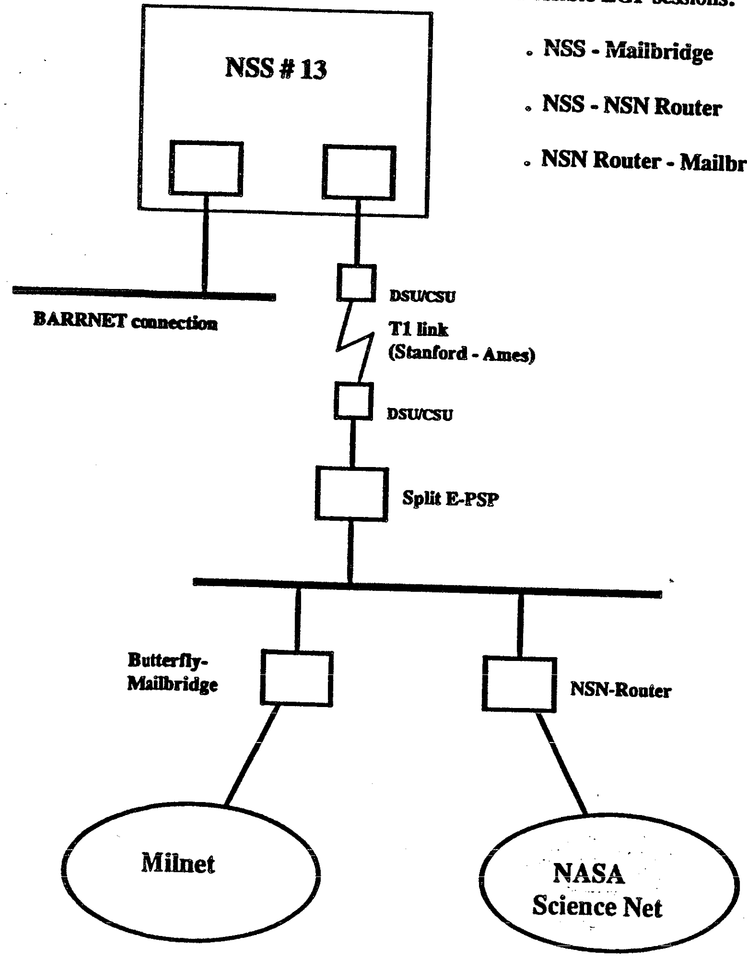


7 March 1989, HWB

Planned NSFNET/NSN/DDN connection at NASA Ames

Possible EGP sessions:

- NSS - Mailbridge
- NSS - NSN Router
- NSN Router - Mailbridge



7 March 1989, HWB

Fair Queueing Revisited

Alan Demers (Xerox PARC)
Srinivasan Keshav (Berkeley)
Scott Shenker (Xerox PARC)

Previous Work:
- Nagle (1985)
- Murray and Demers (1987)

Related Work:
- Clark, Davin, and Heybey
- Zhang (Virtual Clock)
- Other: Mills (Fuzzballs)
  - Morgan (Datakit)

Congestion Control in Datagram Networks

Three Kinds of Congestion Control Algorithms

- Flow Control Algorithms (Sources)
  - Controls traffic sent by source
- Routing Algorithms (Gateways)
  - Adaptive routing can reduce traffic on congested links
- Queueing Algorithms (Gateways)
  - *QA's not typically considered part of Congestion Control*

Four Questions

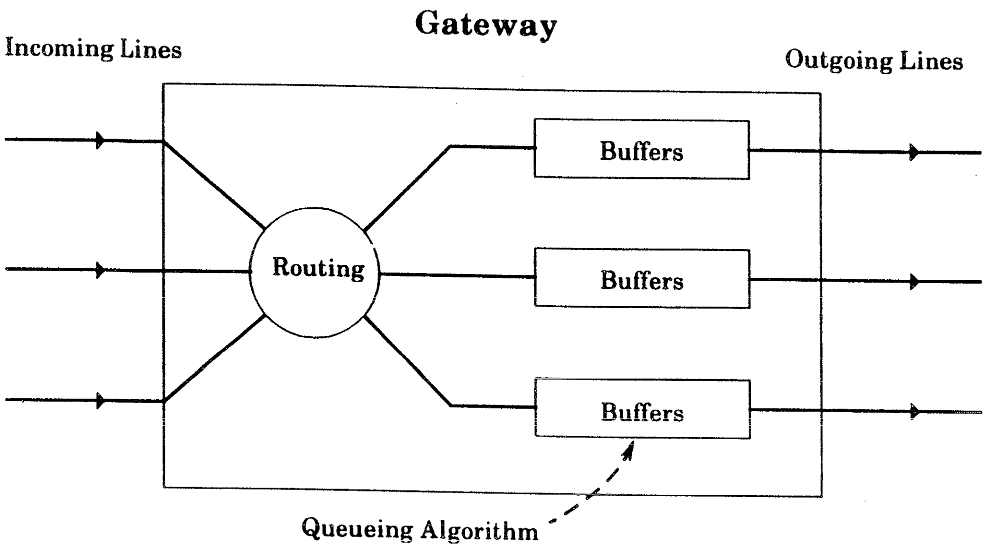
① Why are QA's relevant to congestion control?

② What is Fair Queueing?

③ How well does Fair Queueing work?

- Analytic evaluation
- Simulation

④ What are the potential problems with Fair Queueing?



Question: Why are QA's relevant to congestion control?

- QA's don't directly reduce traffic on congested links
- They merely control the order in which packets are sent

Answer:

- QA's determine how sources affect each other
- This interaction between sources determines the collective behavior of flow control algorithms

QA's allocate three quantities:

Bandwidth:	(*whose* packets get transmitted)
Buffer space:	(*whose* packets get discarded)
Delay:	(*when* packets get transmitted)

These choices are what determines how sources interact

Example: FCFS

Allocations determined completely by order-of-arrival

Congestion control relegated to sources

No insulation between sources

My performance depends on everybody else's behavior

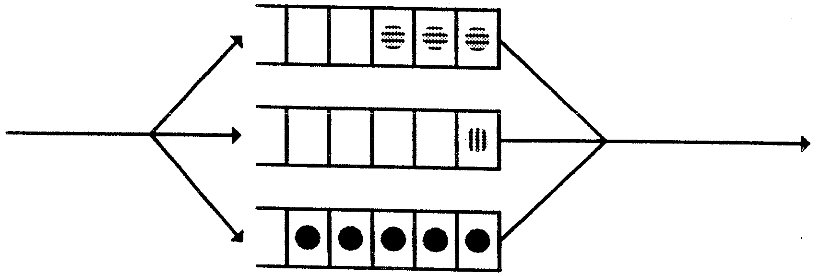
- Vulnerable to ill-behaved sources
- Congestion control requires delicate coordination among all sources

Alternative: allocate resources fairly

Fair: each user either gets full request, or no one gets more

- Not vulnerable to ill-behaved sources
- Performance: less dependent on other sources
  depends more on own behavior

Nagle's Proposal: Fair Queueing



- Separate queues for each source-destination pair
- Round-Robin service on queues

Provides insulation between sources

Problems:
1) Bandwidth allocation unfair
   (different packet sizes)
2) Discontinuity properties
3) Delay and bandwidth linked

Motivation for new algorithm

Packet-by-Packet round-robin is not fair

Bit-by-Bit Round-Robin is fair

Not feasible, but still instructive

Bit-by-Bit Round Robin (Processor Sharing)

Definitions:

$R(t)$: # of rounds up to time t

$N_{ac}(t)$: # of active conversations

$\Rightarrow \frac{\partial R}{\partial t} = \frac{\mu}{N_{ac}(t)}$ where μ is linespeed

Packet i in conversation α

Time-of-Arrival: t_i^α Size: P_i^α

Starting Round: S_i^α

Finishing Round: F_i^α

Dynamical Equations:

$S_i^\alpha = MAX[F_{i-1}^\alpha , R(t_i^\alpha)]$

$F_i^\alpha = S_i^\alpha + P_i^\alpha = P_i^\alpha + MAX[F_{i-1}^\alpha , R(t_i^\alpha)]$

Note: conversation is active if $F_i^\alpha \geq R(t)$

BbBRR: fair, but obviously impractical

Emulate BbBRR with Packet Version

- Compute quantities in terms of t_i^α and P_i^α

  S_i^α, F_i^α, $R(t)$, $N_{ac}(t)$

- Let F_i^α's determine sending order of packets

This is the basic Fair Queueing algorithm (FQ)

- Preemptive and nonpreemptive versions of FQ
- # bits-sent within $P_{\max}$ of BbBRR *(#1)*
- F_i^α continuous in t_i^α *(#2)*

Extension of Algorithm

Bid Round: $B_i^\alpha = P_i^\alpha + MAX[F_{i-1}^\alpha , R(t_i^\alpha) - \delta]$

- B_i^α's determine sending order
- δ: gives priority to new conversations *(#3)*

Evaluation of FQ

Analysis of FQ

- Throughput/Delay Calculations

    Single gateway

    Simple source model

Simulation of FQ

- Evaluate FQ in simulated network

    Real flow control algorithms

    Various network configurations

Analysis of Algorithm

Restrict to two types of sources

- FTP: Infinite source of packets (size P_F)

    FTP's care only about bandwidth

- Telnet: Random generation process (size P_T)

    Telnet's care only about delay

Performance: FTP bandwidth and Telnet delay

- Fairness of FQ determines FTP bandwidth
- Telnet delay is only nontrivial computation

Single Telnet with N FTP's ($\delta = 0$): $D_N(P_T)$

- Solution in terms of Telnet-only delay: $D_0(P_T)$
- $D_N(P_T) = D_0((N+1)P_T) + A_N(P_T)$

    $D_0((N+1)P_T)$: delay of BbBRR

    $A_N(P)$: emulation error

        independent of arrival process

Formulae for A(P):

(depends on synchronization of FTP's)

Preemptive

$$A(P) = N\left(P - \frac{P_F}{2}\right) \qquad \text{for} \qquad P \geq P_F$$

$$N\left(P - \frac{P_F}{2}\right) \leq A(P) \leq \frac{NP^2}{2P_F} \quad \text{for} \quad P_F \geq P \geq \frac{P_F}{2}\left(1 + \frac{1}{N}\right)$$

$$\frac{1}{2P_F}\left(\frac{P_F}{2} + N\left(P - \frac{P_F}{2}\right)\right)^2 \leq A(P) \leq \frac{NP^2}{2P_F} \quad \text{for} \quad \frac{P_F}{2}\left(1 + \frac{1}{N}\right) \geq P \geq \frac{P_F}{2}\left(1 - \frac{1}{N}\right)$$

$$0 \leq A(P) \leq \frac{NP^2}{2P_F} \qquad \text{for} \qquad \frac{P_F}{2}\left(1 - \frac{1}{N}\right) \geq P$$

Nonpreemptive

$$A(P) = N\left(P - \frac{P_F}{2}\right) \qquad \text{for} \qquad P \geq P_F$$

$$N\left(P - \frac{P_F}{2}\right) \leq A(P) \leq \left(\frac{P_F}{2}\right)\left\{1 + \frac{1}{N}\left[k^2 + k(2\varepsilon - 1)\right]\right\} \quad \text{for} \quad P_F \geq P \geq \frac{P_F}{2}\left(1 + \frac{1}{N}\right)$$

$$\frac{P_F}{2} \leq A(P) \leq \left(\frac{P_F}{2}\right)\left\{1 + \frac{1}{N}\left[k^2 + k(2\varepsilon - 1)\right]\right\} \quad \text{for} \quad \frac{P_F}{2}\left(1 + \frac{1}{N}\right) \geq P$$

with ε and k defined by $P_T = P_F \frac{(k + \varepsilon)}{N}$

Analysis of Fair Queueing:

Throughput/Delay curve for simple example

Single Gateway with 3 FTP's and 1 Telnet

Telnet: Poisson source

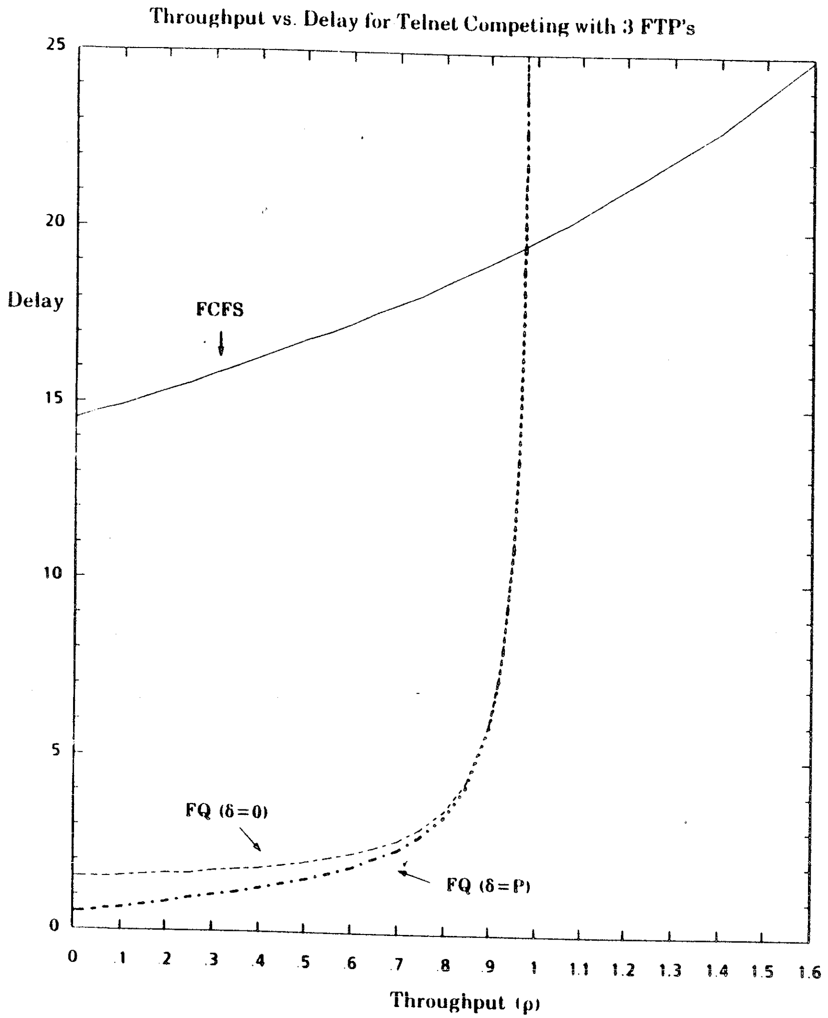
FTP's: Sliding window flow control (window size of 5)

All other delays ignored

All packets same size P

Compare FQ with FCFS

Simulations



Single Telnet source (Poisson strength λ), with three FTP conversations

Throughput measured relative to the Telnet's fair share, $\rho \equiv 4\lambda P/\mu$

FQ algorithm is nonpreemptive

FCFS case always has 15 FTP packets in the queue.

Compare FQ and FCFS in a realistic environment

- Endpoints:

FTP:	infinite supply, $P_F = 1000$ bytes	*(throughput)*
Telnet:	Poisson process, $P_T = 40$ bytes	*(delay)*
Sink:	returns ack packets $P_{ack} = 40$ bytes	

- Flow Control Algorithms

Generic:	sliding window *(max window of 5)*
J/K:	window adjusts to dropped packets
DECbit:	window adjusts to "congestion bit" gateway sets bit "selectively"

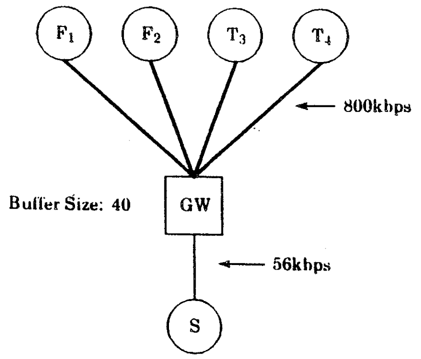
  - Tuned to keep queues small, service fair
  - State-of-the-art flow control
  - QA still FCFS

⇒ Six Protocol Pairs

- Developed FQ bit-setting algorithm for DECbit

- Networks

Several "benchmark scenarios"



Underloaded Gateway

All protocol pairs perform well:

Only difference is in average delay of Telnet packets

FCFS

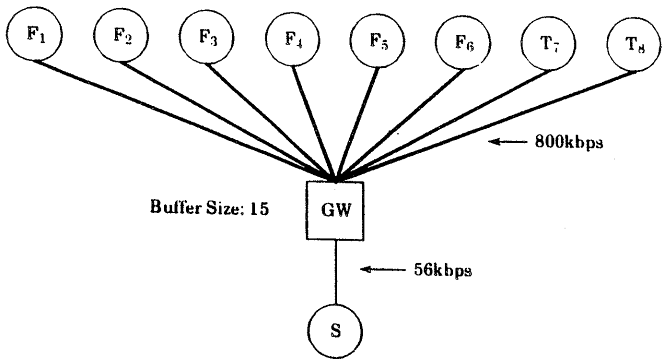
Generic: 1.35
JK: 1.35
DECbit : 0.22

FQ

All: 0.08

Lesson:

FQ delivers better service to Telnets
Does not depend on behavior of other sources



Overloaded Gateway

FCFS

Generic: FTP's "segregate" into winners and losers
(bandwidth allocation: .06 .43 .51 33 33 33)

JK: FTP bandwidth allocation reasonably fair

Both: Telnet's receive very little bandwidth

DECbit: Bandwidth allocation perfectly fair
Average Telnet delay: 0.78

FQ

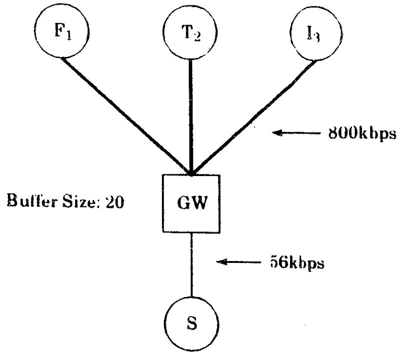
Generic: Bandwidth allocation unfair

JK: Bandwidth allocation reasonably fair

DECbit: Bandwidth allocation perfectly fair

All: Average Telnet delay: 0.087

Lesson: FQ cannot provide adequate congestion control by itself



Ill-Behaved Source

FCFS

Ill-Behaved source gets almost all of the bandwidth

FQ

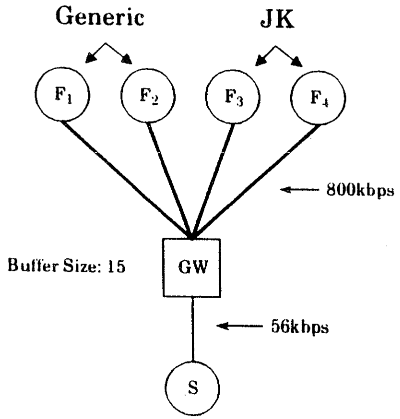
Ill-Behaved source gets almost no bandwidth

(due to penalty feature in algorithm)

Lesson:

FQ provides protection from ill-behaved sources

FCFS leaves sources vulnerable



Mixed Protocols

FCFS

Generic FTP's capture 2/3 of bandwidth

FQ

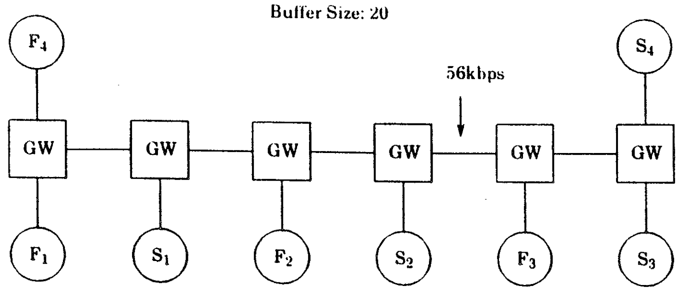
JK FTP's capture 2/3 of bandwidth

Generic FTP's "segregate"

Lesson:

FQ gives incentive for "good" behavior

FCFS gives incentive for "bad" behavior



Multihop Path

FCFS

JK + Generic:

Single-hop gets 2.5 times bandwidth of multihop

DECbit:

Single-hop gets 3.0 times bandwidth of multihop!

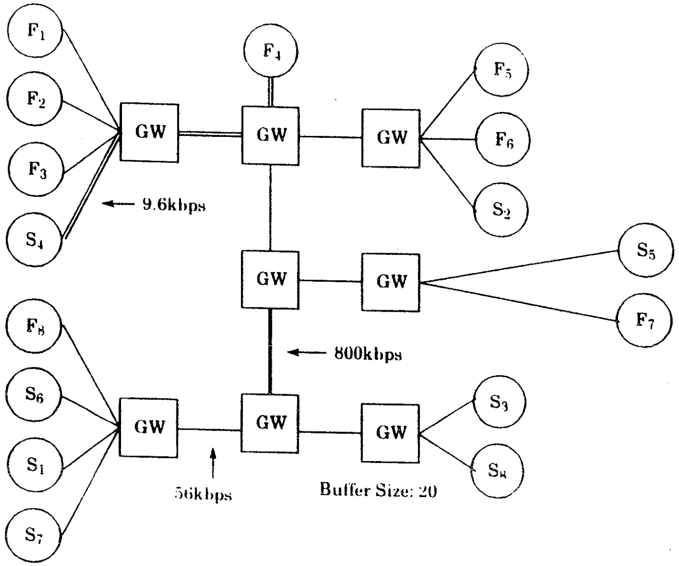
Designed for fairness, but does opposite

Tuning Problem

FQ

All: FTP's receive same bandwidth

Lesson: Hard to achieve fairness via flow control with FCFS gateways



More Complicated Network

FQ

"Counter-flow" conversations get more bandwidth!

Summary

Fair Queueing:

① Improves performance of each flow control algorithm

- Decreases Telnet delay
- Increases fairness of FTP bandwidth allocation

② Insulates conversations from each other

- Provides protection from ill-behaved sources
- Supports heterogeneity in flow control algorithms
  - Rewards well-behaved sources
  - Enables experimentation with new algorithms
  - Allows sources to make own delay/thrput tradeoff
- Suggests possibility of large stable networks

③ Needs smart flow control to function adequately

- Protects from other sources, not from self

Future Work

Practical Direction

- Implementation (Davin and Heybey)
- Buffer allocation policies (Davin and Heybey)
- Design of new rate-based flow control for FQ
- Design of new routing algorithms for FQ
- Simulation of: larger networks
  transients
  routing
  ....

Theoretical Direction

- Solvable Models:
  Poisson sources, Exponential servers
  Simple model of rate-adjusting flow control
  Poisson rate is set in response to congestion indicators
  Steady-state solution

★ Can quantify comparisons between FQ and FCFS

★ FQ is only reasonable QA that ensures linear stability

- Game-theoretic analysis of gateway algorithms

★ FQ is only reasonable QA that ensures fairness

FQ: Issues to be Resolved

Implementation

- Speed
- Scaling with number of s-d pairs

Context

- Scaling of bandwidth-delay product

Fairness

- Aggregation of users
- Assignment of resource rights

  trust model?

  billing?

Alternatives

- Other solutions must address the issues of
  Fairness
  Protection from ill-behaved sources
  Autonomy of delay/throughput tradeoffs
  Heterogeneity of flow control algorithms
  Stability
  Incentive structure

Questions about Fairness

Q: Is allocating on basis of source-destination fair?

A: No, but it is the best we can do.

- Want to allocate on basis of human users

  Might have many users per s-d pair

  This aggregation of users is unfair
- However, GW's can only identify users on s-d basis
- FQ delivers fairness to slightly incorrect entity
- Other algorithms don't deliver fairness at all

Q: Are all conversations equally important?

A: Probably not.

- Can generalize FQ algorithm to assign *shares*

  Number of slots in BbBRR
- Problem is in assigning, authenticating, and communicating allocations
- Solution would also solve the aggregation problem
- Will probably need to introduce accounting

Random Drop

Good idea, not a panacea

Original Algorithm:

If gateway must drop a packet, it picks one at random

Properties:

- **Alleviates segregation!**
  - Destroys clocking mechanism
  - Important improvement
    - but, segregation can still occur (Zhang)
- Still has problems of FCFS
  - Vulnerable to ill-behaved sources
  - Telnet's receive poor delay
  - Allocation of Bandwidth is unfair
    - Simulations (Zhang + Hamesh)
    - Theory (steady-state approximation)
      - thput $\approx 1/RTT$ *(transient analysis worse)*
      - multihop thput $\approx e^{-\#(\text{congested hops})}$
  - Stability problems

Two Visions of Future

FCFS Networks

- Universally implemented flow control
  - Endless battles over standards
  - Compatibility problems inhibit innovation
  - Improved technology renders standards obsolete
- Protocol police
  - Ruthless punishment all flow control offenders
  - Jails full of teenage hackers
- Constitutional amendment outlaws desecration of the Host Requirements RFC

FQ Networks

- Peaceful coexistence of different flow control algorithms
- Development of efficient flow control algorithms
  - Tailored to specific applications
  - Developed w/o interference of standards body
- Strong fences make for good neighbors

Analysis and Simulation of a Fair Queueing Algorithm

Alan Demers
Srinivasan Keshav†
Scott Shenker
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

Abstract

We discuss gateway queueing algorithms and their role in controlling congestion in datagram networks. A fair queueing algorithm, based on an earlier suggestion by Nagle, is proposed. Analysis and simulations are used to compare this algorithm to other congestion control schemes. We find that fair queueing provides several important advantages over the usual first-come-first-serve queueing algorithm: fair allocation of bandwidth, lower delay for sources using less than their full share of bandwidth, and protection from ill-behaved sources.

1. Introduction

Datagram networks have long suffered from performance degradation in the presence of congestion [Ger80]. The rapid growth, in both use and size, of computer networks has sparked a renewed interest in methods of congestion control [DEC87abcd, Jac88a, Man89, Nag87]. These methods have two points of implementation. The first is at the source, where flow control algorithms vary the rate at which the source sends packets. Of course, flow control algorithms are designed primarily to ensure the presence of free buffers at the destination host, but we are more concerned with their role in limiting the overall network traffic. The second point of implementation is at the gateway. Congestion can be controlled at gateways through routing and queueing algorithms. Adaptive routing, if properly implemented, lessens congestion by routing packets away from network bottlenecks. Queueing algorithms, which control the order in which packets are sent and the usage of the gateway's buffer space, do not affect congestion directly, in that they do not change the total traffic on the gateway's outgoing line. Queueing algorithms do, however, determine the way in which packets from different sources interact with each other which, in turn, affects the collective behavior of flow control algorithms. We shall argue that this effect, which is often ignored, makes queueing algorithms a crucial component in effective congestion control.

Queueing algorithms can be thought of as allocating three nearly independent quantities: bandwidth (*which* packets get *transmitted*), promptness (*when* do those packets get *transmitted*), and buffer space (*which* packets are *discarded* by the gateway). Currently, the most common queueing algorithm is first-come-first-serve (FCFS). FCFS queueing essentially relegates all congestion control to the sources, since the order of arrival completely determines the bandwidth, promptness, and buffer space allocations. Thus, FCFS inextricably intertwines these three allocation issues. There may indeed be flow control algorithms that, when universally implemented throughout a network with FCFS gateways, can overcome these limitations and provide reasonably fair and efficient congestion control. This point is discussed more fully in Sections 3 and 4, where several flow control algorithms are compared. However, with today's diverse and decentralized computing environments, it is unrealistic to expect universal implementation of any given flow control algorithm. This is not merely a question of standards, but also one of compliance. Even if a universal standard such as ISO [ISO86] were adopted, malfunctioning hardware and software could violate the standard, and there is always the possibility that individuals would alter the algorithms on their own machine to improve their performance at the expense of others. Consequently, congestion control algorithms should function well even in the presence of ill-behaved sources. Unfortunately, no matter what flow control algorithm is used by the well-behaved sources, networks with FCFS gateways do not have this property. A single source, sending packets to a gateway at a sufficiently high speed, can capture an arbitrarily high fraction of the bandwidth of the outgoing line. Thus, FCFS queueing is not adequate; more discriminating queueing algorithms must be used in conjunction with source flow control algorithms to control congestion effectively in noncooperative environments.

Following a similar line of reasoning, Nagle [Nag87, Nag85] proposed a *fair queueing* (FQ) algorithm in which gateways maintain separate queues for packets from each individual source. The queues are serviced in a round-robin manner. This prevents a source from arbitrarily increasing its share of the bandwidth or the delay of other sources. In fact, when a source sends packets too quickly, it merely increases the length of its own queue. Nagle's algorithm, by changing the way packets from different sources interact, does not reward, nor leave others vulnerable to, anti-social behavior. On the surface, this proposal appears to have considerable merit, but we are not aware of any published data on the performance of datagram networks with such fair queueing gateways. In this paper, we will first describe a modification of Nagle's algorithm, and then provide simulation data comparing networks with FQ gateways and those with FCFS gateways.

The three different components of congestion control algorithms introduced above, source flow control, gateway routing, and gateway queueing algorithms, interact in interesting and complicated ways. It is impossible to assess the effectiveness of any algorithm without reference to the other components of congestion control in operation. We will evaluate our proposed queueing algorithm in the context of static routing and several widely used flow control algorithms. The aim is to find a queueing algorithm that functions well in current computing environments. The algorithm might, indeed it should, *enable* new and improved routing and flow control algorithms, but it must not require them.

We had three goals in writing this paper. The first was to describe a new fair queueing algorithm. In Section 2.1, we discuss the design requirements for an effective queueing algorithm and outline how Nagle's original proposal fails to meet them. In Section 2.2, we propose a new fair queueing algorithm which meets these design requirements. The second goal was to provide some rigorous understanding of the performance of this algorithm; this is done in Section 2.3, where we present a delay-throughput curve given by our fair queueing algorithm for a specific configuration of sources. The third goal was to evaluate this new queueing proposal in the context of real networks. To this end, we discuss flow control algorithms in Section 3, and then, in Section 4, we present simulation data comparing several combinations of flow control and queueing algorithms on six benchmark networks. Section 5 contains an overview of our results, a discussion of other proposed queueing algorithms, and an analysis of some criticisms of fair queueing.

In circuit switched networks where there is explicit buffer reservation and uniform packet sizes, it has been established that round robin service disciplines allocate bandwidth fairly [Hah86, Kat87]. Recently Morgan [Mor89] has examined the role such queueing algorithms play in controlling congestion in circuit switched networks; while his application context is quite different from ours, his conclusions are qualitatively similar. In other related work, the DATAKIT™ queueing algorithm combines round robin service and FIFO priority service, and has been analyzed extensively [Lo87, Fra84]. Also, Luan and Lucantoni present a different form of bandwidth management policy for circuit switched networks [Lua88].

Since the completion of this work, we have learned of a similar *Virtual Clock* algorithm for gateway resource allocation proposed by Zhang [Zha89]. Furthermore, Heybey and Davin [Hey89] have simulated a simplified version of our fair queueing algorithm.

2. Fair Queueing

2.1. Motivation What are the requirements for a queueing algorithm that will allow source flow control algorithms to provide adequate congestion control even in the presence of ill-behaved sources? We start with Nagle's observation that such queueing algorithms must provide protection, so that ill-behaved sources can only have a limited negative impact on well behaved sources. Allocating bandwidth and buffer space in a *fair* manner, to be defined later, automatically ensures that ill-behaved sources can get no more than their fair share. This led us to adopt, as our central design consideration, the requirement that the queueing algorithm allocate bandwidth and buffer space fairly. Ability to control the promptness, or delay, allocation somewhat independently of the bandwidth and buffer allocation is also desirable. Finally, we require that the gateway should provide service that, at least on average, does not depend discontinuously on a packet's time of arrival (this continuity condition will become clearer in Section 2.2). This requirement attempts to prevent the efficiency of source implementations from being overly sensitive to timing details (timers are the Bermuda Triangle of flow control algorithms). Nagle's proposal does not satisfy these requirements. The most obvious flaw is its lack of consideration of packet lengths. A source using long packets gets more bandwidth than one using short packets, so bandwidth is not allocated fairly. Also, the proposal has no explicit promptness allocation other than that provided by the round-robin service discipline. In addition, the static round robin ordering violates the continuity requirement. In the following section we attempt to correct these defects.

In stating our requirements for queueing algorithms, we have left the term *fair* undefined. The term *fair* has a clear colloquial meaning, but it also has a technical definition (actually several, but only one is considered here). Consider, for example, the allocation of a single resource among N users. Assume there is an amount μ_{total} of this resource and that each of the users requests an amount ρ_i and, under a particular allocation, receives an

---

† Current Address: University of California at Berkeley

amount μ_i. What is a fair allocation? The max-min fairness criterion [Hah86, Gaf84, DEC87d] states that an allocation is fair if (1) no user receives more than its request, (2) no other allocation scheme satisfying condition 1 has a higher minimum allocation, and (3) condition 2 remains recursively true as we remove the minimal user and reduce the total resource accordingly, $\mu_{total} \leftarrow \mu_{total} - \mu_{min}$. This condition reduces to $\mu_i = MIN(\mu_{fair}, \rho_i)$ in the simple example, with μ_{fair}, the *fair share*, being set so that $\mu_{total} = \sum_{i=1}^{N} \mu_i$. This concept of fairness easily generalizes to the multiple resource case [DEC87d]. Note that implicit in the max-min definition of fairness is the assumption that the users have equal *rights* to the resource.

In our communication application, the bandwidth and buffer demands are clearly represented by the packets that arrive at the gateway. (Demands for promptness are not explicitly communicated, and we will return to this issue later.) However, it is not clear what constitutes a *user*. The user associated with a packet could refer to the source of the packet, the destination, the source-destination pair, or even refer to an individual process running on a source host. Each of these definitions has limitations. Allocation per source unnaturally restricts sources such as file servers which typically consume considerable bandwidth. Ideally the gateways could know that some sources deserve more bandwidth than others, but there is no adequate mechanism for establishing that knowledge in today's networks. Allocation per receiver allows a receiver's useful incoming bandwidth to be reduced by a broken or malicious source sending unwanted packets to it. Allocation per process on a host encourages human users to start several processes communicating simultaneously, thereby avoiding the original intent of fair allocation. Allocation per source-destination pair allows a malicious source to consume an unlimited amount of bandwidth by sending many packets all to different destinations. While this does not allow the malicious source to do useful work, it can prevent other sources from obtaining sufficient bandwidth.

Overall, allocation on the basis of source-destination pairs, or *conversations*, seems the best tradeoff between security and efficiency and will be used here. However, our treatment will apply to any of these interpretations of user. With our requirements for an adequate queueing algorithm, coupled with our definitions of *fairness* and *user*, we now turn to the description of our algorithm.

2.2. Definition of algorithm It is simple to allocate buffer space fairly by dropping packets, when necessary, from the conversation with the largest queue. Allocating bandwidth fairly is less straightforward. Pure round-robin service provides a fair allocation of packets-sent but fails to guarantee a fair allocation of bandwidth because of variations in packet sizes. To see how this unfairness can be avoided, we first consider a hypothetical service discipline where transmission occurs in a bit-by-bit round robin (BR) fashion (as in a head-of-queue processor sharing discipline). This service discipline allocates bandwidth fairly since at every instant in time each conversation is receiving its fair share. Let $R(t)$ denote the number of rounds made in the round-robin service discipline up to time t ($R(t)$ is a continuous function, with the fractional part indicating partially completed rounds). Let $N_{ac}(t)$ denote the number of active conversations, i.e. those that have bits in their queue at time t. Then, $\frac{\partial R}{\partial t} = \frac{\mu}{N_{ac}(t)}$, where μ is the linespeed of the gateway's outgoing line (we will, for convenience, work in units such that $\mu = 1$). A packet of size P whose first bit gets serviced at time t_0 will have its last bit serviced P rounds later, at time t such that $R(t) = R(t_0) + P$. Let t_i^α be the time that packet i belonging to conversation α arrives at the gateway, and define the numbers S_i^α and F_i^α as the values of $R(t)$ when the packet started and finished service. With P_i^α denoting the size of the packet, the following relations hold: $F_i^\alpha = S_i^\alpha + P_i^\alpha$ and $S_i^\alpha = MAX(F_{i-1}^\alpha, R(t_i^\alpha))$. Since $R(t)$ is a strictly monotonically increasing function whenever there are bits at the gateway, the ordering of the F_i^α values is the same as the ordering of the finishing times of the various packets in the BR discipline.

Sending packets in a bit-by-bit round robin fashion, while satisfying our requirements for an adequate queueing algorithm, is obviously unrealistic. We hope to emulate this impractical algorithm by a practical packet-by-packet transmission scheme. Note that the functions $R(t)$ and $N_{ac}(t)$ and the quantities S_i^α and F_i^α depend only on the packet arrival times t_i^α and not on the actual packet transmission times, as long as we define a conversation to be active whenever $R(t) \le F_i^\alpha$ for $i = MAX(j \mid t_j^\alpha \le t)$. We are thus free to use these quantities in defining our packet-by-packet transmission algorithm. A natural way to emulate the bit-by-bit round-robin algorithm is to let the quantities F_i^α define the sending order of the packets. Our packet-by-packet transmission algorithm is simply defined by the rule that, whenever a packet finishes transmission, the next packet sent is the one with the smallest value of F_i^α. In a preemptive version of this algorithm, newly arriving packets whose finishing number F_i^α is smaller than that of the packet currently in transmission preempt the transmitting packet. For practical reasons, we have implemented the nonpreemptive version, but the preemptive algorithm (with resumptive service) is more tractable analytically. Clearly the preemptive and nonpreemptive packetized algorithms do not give the same instantaneous bandwidth allocation as the BR version. However, for each conversation the total bits sent at a given time by these three algorithms are always within P_{max} of each other, where P_{max} is the maximum packet size (this emulation discrepancy bound was proved by Greenberg and Madras [Gree89]). Thus, over sufficiently long conversations, the packetized algorithms asymptotically approach the fair bandwidth allocation of the BR scheme.

Recall that the user's request for promptness is not made explicit. (The IP [Pos81] protocol does have a field for type-of-service, but not enough applications make intelligent use of this option to render it a useful hint.) Consequently, promptness allocation must be based solely on data already available at the gateway. One such allocation strategy is to give more promptness (less delay) to users who utilize less than their fair share of bandwidth. Separating the promptness allocation from the bandwidth allocation can be accomplished by introducing a nonnegative parameter δ, and defining a new quantity, the *bid* B_i^α, via $B_i^\alpha = P_i^\alpha + MAX(F_{i-1}^\alpha, R(t_i^\alpha) - \delta)$. The quantities $R(t)$, $N_{ac}(t)$, F_i^α, and S_i^α remain as before, but now the sending order is determined by the B's, not the F's. The asymptotic bandwidth allocation is independent of δ, since the F's control the bandwidth allocation, but the algorithm gives slightly faster service to packets that arrive at an inactive conversation. The parameter δ controls the extent of this additional promptness. Note that the bid B_i^α is continuous in t_i^α, so that the continuity requirement mentioned in Section 2.1 is met.

The role of this term δ can be seen more clearly by considering the two extreme cases $\delta = 0$ and $\delta = \infty$. If an arriving packet has $R(t_i^\alpha) \le F_{i-1}^\alpha$, then the conversation α is active (i.e. the corresponding conversation in the BR algorithm would have bits in the queue). In this case, the value of δ is irrelevant and the bid number depends only on the finishing number of the previous packet. However, if $R(t_i^\alpha) > F_{i-1}^\alpha$, so that the α conversation is inactive, the two cases are quite different. With $\delta = 0$, the bid number is given by $B_i^\alpha = P_i^\alpha + R(t_i^\alpha)$ and is completely independent of the previous history of user α. With $\delta = \infty$, the bid number is $B_i^\alpha = P_i^\alpha + F_{i-1}^\alpha$ and depends only the previous packet's finishing number, no matter how many rounds ago. For intermediate values of δ, scheduling decisions for packets arriving at inactive conversations depends on the previous packet's finishing round as long as it wasn't too long ago, and δ controls how far back this dependence goes.

Recall that when the queue is full and a new packet arrives, the last packet from the source currently using the most buffer space is dropped. We have chosen to leave the quantities F_i^α and S_i^α unchanged when we drop a packet. This provides a small penalty for ill-behaved hosts, in that they will be charged for throughput that, because of their own poor flow control, they could not use.

2.3.. Properties of Fair Queueing The desired bandwidth and buffer allocations are completely specified by the definition of fairness, and we have demonstrated that our algorithm achieves those goals. However, we have not been able to characterize the promptness allocation for an arbitrary arrival stream of packets. To obtain some quantitative results on the promptness, or delay, performance of a single FQ gateway, we consider a very restricted class of arrival streams in which there are only two types of sources. There are FTP-like file transfer sources, which always have ready packets and transmit them whenever permitted by the source flow control (which, for simplicity, is taken to be sliding window flow control), and there are Telnet-like interactive sources, which produce packets intermittently according to some unspecified generation process. What are the quantities of interest? An FTP source is typically transferring a large file, so the quantity of interest is the transfer time of the file, which for asymptotically large files depends only on the bandwidth allocation. Given the configuration of sources this bandwidth allocation can be computed *a priori* by using the fairness property of FQ gateways. The interesting quantity for Telnet sources is the average delay of each packet, and it is for this quantity that we now provide a rather limited result.

Consider a single FQ gateway with N FTP sources sending packets of size P_F, and allow a single packet of size P_T from a Telnet source to arrive at the gateway at time t. It will be assigned a bid number $B = R(t) + P_T - \delta$; thus, the dependence of the queueing delay on the quantities P_T and δ is only through the combination $P_T - \delta$. We will denote the queueing delay of this packet by $\varphi(t)$, which is a periodic function with period NP_F. We are interested in the average queueing delay Δ

$$\Delta = \frac{1}{NP_F} \int_0^{NP_F} \varphi(t)\,dt$$

The finishing numbers F_i^α for the N FTP's can be expressed, after perhaps renumbering the packets, by $F_i^\alpha = iP_F + l^\alpha$ where the l's obey $0 \le l^\alpha < P_F$. The queueing delay of the Telnet packet depends on the configuration of l's whenever $P_T < P_F$. One can show that the delay is bounded by the extremal cases of $l^\alpha = 0$ for all α and $l^\alpha = \alpha/N$ for $\alpha = 0, 1, \ldots, N-1$. The delay values for these extremal cases are straightforward to calculate; for the sake of brevity we omit the derivation and merely display the result below. The average queueing delay is given by $\Delta = A(P_T - \delta)$ where the function $A(P)$, the delay with $\delta = 0$, is defined below (with integer k and small constant ε, $0 \le \varepsilon < 1$, defined via $P_T = P_F(k + \varepsilon)/N$).

Preemptive

$$A(P)=N(P-\frac{P_F}{2}) \qquad \text{for} \qquad P\ge P_F$$

$$N(P-\frac{P_F}{2})\le A(P)\le \frac{NP^2}{2P_F} \quad \text{for} \quad P_F\ge P\ge \frac{P_F}{2}(1+\frac{1}{N})$$

$$\frac{1}{2P_F}(\frac{P_F}{2}+N(P-\frac{P_F}{2}))^2\le A(P)\le \frac{NP^2}{2P_F} \quad \text{for} \quad \frac{P_F}{2}(1+\frac{1}{N})\ge P\ge \frac{P_F}{2}(1-\frac{1}{N})$$

$$0\le A(P)\le \frac{NP^2}{2P_F} \qquad \text{for} \qquad \frac{P_F}{2}(1-\frac{1}{N})\ge P$$

Nonpreemptive

$$A(P)=N(P-\frac{P_F}{2}) \qquad \text{for} \qquad P\ge P_F$$

$$N(P-\frac{P_F}{2})\le A(P)\le (\frac{P_F}{2})\left[1+\frac{1}{N}(k^2+k(2\epsilon-1))\right] \quad \text{for} \quad P_F\ge P\ge \frac{P_F}{2}(1+\frac{1}{N})$$

$$\frac{P_F}{2}\le A(P)\le (\frac{P_F}{2})\left[1+\frac{1}{N}(k^2+k(2\epsilon-1))\right] \quad \text{for} \quad \frac{P_F}{2}(1+\frac{1}{N})\ge P$$

Now consider a general Telnet packet generation process (ignoring the effects of flow control) and characterize this generation process by the function $D_0(P_T)$ which denotes the queueing delay of the Telnet source when it is the sole source at the gateway. In the BR algorithm, the queueing delay of the Telnet source in the presence of N FTP sources is merely $D_0((N+1)P_T)$. For the packetized preemptive algorithm with $\delta=0$, we can express the queueing delay in the presence of N FTP sources, call it $D_N(P_T)$, in terms of D_0 via the relation (averaging over all relative synchronizations between the FTP's and the Telnet):

$$D_N(P_T)=D_0((N+1)P_T)+A(P_T)$$

where the term $A(P_T)$ reflects the extra delay incurred when emulating the BR algorithm by the preemptive packetized algorithm.

For nonzero values δ, the generation process must be further characterized by the quantity $I_0(P_T,t)$ which, in a system where the Telnet is the sole source, is the probability that a packet arrives to a queue which has been idle for time t. The delay is given by,

$$D_N(P_T)=D_0((N+1)P_T)+A(P_T)-$$
$$\int I_0((N+1)P_T,t)\left[A(P_T)-A(P_T-MIN(\frac{t}{N},\delta))\right]dt$$

where the last term represents the reduction in delay due the the nonzero δ. These expressions for D_N, which were derived for the preemptive case, are also valid for the nonpreemptive algorithm when $P_T\ge P_F$.

What do these forbidding formulae mean? Consider, for concreteness, a Poisson arrival process with arrival rate λ, packet sizes $P_T=P_F=P$, a linespeed $\mu=1$, and an FTP synchronization described by $l^\alpha=\alpha/N$ for $\alpha=0,1,\ldots,N-1$. Define ρ to be the average bandwidth of the stream, measured relative to the fair share of the Telnet: $\rho=\lambda P(N+1)$. Then, for the nonpreemptive algorithm,

$$\frac{D_N(P)}{P}=\frac{\rho}{2(1-\rho)}+\frac{N\rho}{2}+N(1-\rho)\left[\frac{1}{2}-\frac{N+1}{N\rho}\times\right.$$
$$\left.\left[1-\exp\left[-\frac{\rho N}{(N+1)}MIN(\frac{\delta}{P},\frac{1}{2}(1-\frac{1}{N}))\right]\right]\right]$$

This is the throughput/delay curve the FQ gateway offers the Poisson Telnet source (the formulae for different FTP synchronizations are substantially more complicated, but have the same qualitative behavior). This can be contrasted with that offered by the FCFS gateway, although the FCFS results depend in detail on the flow control used by the FTP sources and on the surrounding network environment. Assume that all other communications speeds are infinitely fast in relation to the outgoing linespeed of the gateway, and that the FTP's all have window size W, so there are always NW FTP packets in the queue or in transmission. Figure 1 shows the throughput/delay curves for an FCFS gateway, along with those for a FQ gateway with $\delta=0$ and $\delta=P$. For $\rho\rightarrow 0$, FCFS gives a large queueing delay of $(NW-\frac{1}{2})P$, whereas FQ gives a queueing delay of $NP/2$ for $\delta=0$ and $P/2$ for $\delta=P$. This ability to provide a lower delay to lower throughput sources, completely independent of the window sizes of the FTP's, is one of the most important features of fair queueing. Note also that the FQ queueing delay diverges as $\rho\rightarrow 1$, reflecting FQ's insistence that no conversation gets more than its fair share. In contrast, the FCFS curve remains finite for all $\rho<(N+1)$, showing that an ill-behaved source can consume an arbitrarily large fraction of the bandwidth.

What happens in a network of FQ gateways? There are few results here, but Hahne [Hah86] has shown that for strict round robin service gateways and only FTP sources there is fair allocation of bandwidth (in the multiple resource sense) when the window sizes are sufficiently large. She also provides examples where insufficient window sizes (but much larger than the communication path) result in unfair allocations. We believe, but have been unable to prove, that both of these properties hold for our fair queueing scheme.

3. Flow Control Algorithms

Flow control algorithms are both the benchmarks against which the congestion control properties of fair queueing are measured, and also the environment in which FQ gateways will operate. We already know that, when combined with FCFS gateways, these flow control algorithms all suffer from the fundamental problem of vulnerability to ill-behaving sources. Also, there is no mechanism for separating the promptness allocation from the bandwidth and buffer allocation. The remaining question is then how fairly do these flow control algorithms allocate bandwidth. Before proceeding, note that there are really two distinct problems in controlling congestion. Congestion *recovery* allows a system to recover from a badly congested state, whereas congestion *avoidance* attempts to prevent the congestion from occurring. In this paper, we are focusing on congestion *avoidance* and will not discuss congestion *recovery* mechanisms at length.

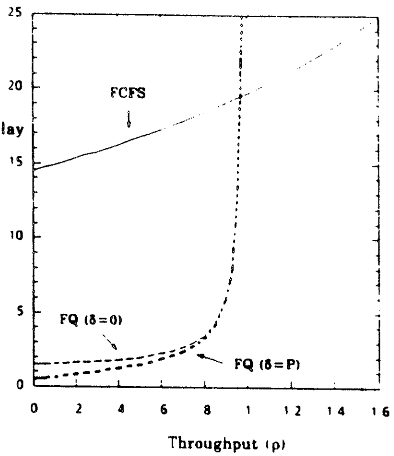


Figure 1: Delay vs. Throughput
This graph describes the queueing delay of a single Telnet source with Poisson generation process of strength λ, sending packets through a gateway with three FTP conversations. The packet sizes are $P_F=P_T=P$, the throughput is measured relative to the Telnet's fair share, $\rho=4\lambda P/\mu$ where μ is the linespeed. The delay is measured in units of P/μ. The FQ algorithm is nonpreemptive, and the FCFS case always has 15 FTP packets in the queue.

A generic version of source flow control, as implemented in XNS's SPP [Xer81] or in TCP [USC81], has two parts. There is a timeout mechanism, which provides for congestion recovery, whereby packets that have not been acknowledged before the timeout period are retransmitted (and a new timeout period set). The timeout periods are given by βrtt where typically $\beta\sim 2$ and rtt is the exponentially averaged estimate of the round trip time (the rtt estimate for retransmitted packets is the time from their first transmission to their acknowledgement). The congestion avoidance part of the algorithm is sliding window flow control, with some set window size. This algorithm has a very narrow range of validity, in that it avoids congestion if the window sizes are small enough, and provides efficient service if the windows are large enough, but cannot respond adequately if either of these conditions is violated.

The second generation of flow control algorithms, exemplified by Jacobson and Karels' (JK) modified TCP [Jac88a] and the original DECbit proposal [DEC87a-c], are descendants of the above generic algorithm with the added feature that the window size is allowed to respond dynamically in response to network congestion (JK also has, among other changes, substantial modifications to the timeout calculation [Jac88a,b, Kar87]). The algorithms use different signals for congestion: JK uses timeouts whereas DECbit uses a header bit which is set by the gateway on all packets whenever the average queue length is greater than one. These mechanisms allocate window sizes fairly, but the relation *Throughput = Window/RoundTrip* implies that conversations with different paths receive different bandwidths.

The third generation of flow control algorithms are similar to the second, except that now the congestion signals are sent selectively. For instance, the selective DECbit proposal [DEC87d] has the gateway measure the flows of the various conversations and only send congestion signals to those users who are using more than their fair share of bandwidth. This corrects the previous unfairness for sources using different paths (see [DEC87d] and section 4), and appears to offer reasonably fair and efficient congestion control in many networks. The DEC algorithm controls the delay by attempting to keep the average queue size close to one. However, it does not allow individual users to make different delay/throughput tradeoffs: the collective tradeoff is set by the gateway.

4. Simulations

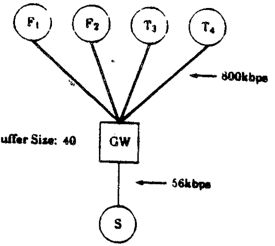
In this section we compare the various congestion control mechanisms, and try to illustrate the interplay between the queueing and flow control algorithms. We simulated these algorithms at the packet level using a network simulator built on the Nest network simulation tool [Nes88]. In order to compare the FQ and FCFS gateway algorithms in a variety of settings, we selected several different flow control algorithms: the generic one described above, JK flow control, and the selective DECbit algorithm. To enable DECbit flow control to operate with FQ gateways, we developed a bit-setting FQ algorithm in which the congestion bits are set whenever the source's queue length is greater than ½ of its fair share of buffer space (note that this is a much simpler bit-setting algorithm than the DEC scheme, which involves complicated averages; however, the choice of $\frac{1}{2}$ is completely *ad hoc*). The Jacobson/Karels flow control algorithm is defined by the 4.3bsd TCP implementation. This code deals with many issues unrelated to congestion control. Rather than using that code directly in our simulations, we have chosen to model the JK algorithm by adding many of the congestion control ideas found in that code, such as adjustable windows, better timeout calculations, and fast retransmit to our generic flow control algorithm. The various cases of test algorithms are labeled in table 1.

Label	Flow Control	Queueing Algorithm
G/FCFS	Generic	FCFS
G/FQ	Generic	FQ
JK/FCFS	JK	FCFS
JK/FQ	JK	FQ
DEC/DEC	DECbit	Selective DECbit
DEC/FQbit	DECbit	FQ with bit setting

Table 1: Algorithm Combinations

Rather than test this set of algorithms on a single *representative* network and load, we chose to define a set of *benchmark* scenarios, each of which, while somewhat unrealistic in itself, serves to illuminate a different facet of congestion control. The load on the network consists of a set of Telnet and FTP conversations. The Telnet sources generate 40 byte packets by a Poisson process with a mean interpacket interval of 5 seconds. The FTP's have an infinite supply of 1000 byte packets that are sent as fast as flow control allows. Both FTP's and Telnet's have their maximum window size set to 5, and the acknowledgement (ACK) packets sent back from the receiving sink are 40 bytes. (The small size of Telnet packets relative to the FTP packets makes the effect of δ insignificant, so the FQ algorithm was implemented with $\delta = 0$). The gateways have finite buffers which, for convenience, are measured in packets rather than bytes. The system was allowed to stabilize for the first 1500 seconds, and then data was collected over the next 500 second interval. For each scenario, there is a figure depicting the corresponding network layout, and a table containing the data. There are four performance measures for each source: total throughput (number of packets reaching destination), average round trip time of the packets, the number of packet retransmissions, and number of dropped packets. We do not include confidence intervals for the data, but repetitions of the simulations have consistently produced results that lead to the same qualitative conclusions.

We first considered several single-gateway networks. The first scenario has two FTP sources and two Telnet sources sending to a sink through a single bottleneck gateway. Note that, in this underloaded case, all of the algorithms provide fair bandwidth allocation, but the cases with FQ provide much lower Telnet delay than those with FCFS. The selective DECbit gives an intermediate value for the Telnet delay, since the flow control is designed to keep the average queue length small.

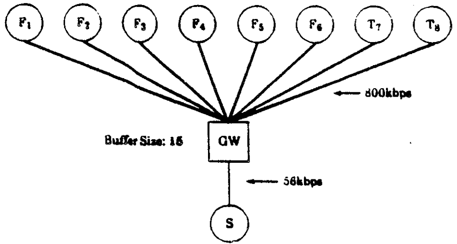


Quantity	Policy	FTP 1	FTP 2	Telnet 3	Telnet 4
Throughput (packets)	G/FCFS	1746	1746	99	96
	G/FQ	1746	1746	102	94
	JK/FCFS	1747	1746	102	104
	JK/FQ	1746	1746	105	103
	DEC/DEC	1746	1746	131	106
	DEC/FQbit	1746	1746	103	94
Average Roundtrip Time	G/FCFS	1.43	1.43	1.36	1.35
	G/FQ	1.43	1.43	079	091
	JK/FCFS	1.43	1.43	1.35	1.36
	JK/FQ	1.43	1.43	084	088
	DEC/DEC	287	288	215	214
	DEC/FQbit	1.63	1.63	080	089
Retransmitted Packets	G/FCFS	0	0	0	0
	G/FQ	0	0	2	1
	JK/FCFS	0	0	0	0
	JK/FQ	0	0	0	0
	DEC/DEC	0	0	1	0
	DEC/FQbit	0	0	2	1
Dropped Packets	G/FCFS	0	0	0	0
	G/FQ	0	0	0	0
	JK/FCFS	0	0	0	0
	JK/FQ	0	0	0	0
	DEC/DEC	0	0	0	0
	DEC/FQbit	0	0	0	0

Scenario 1: Underloaded Gateway

Scenario 2 involves 6 FTP sources and 2 Telnet sources again sending through a single gateway. The gateway, with a buffer size of only 15, is substantially overloaded. This scenario probes the behavior of the algorithms in the presence of severe congestion.

When FCFS gateways are paired with generic flow control, the sources segregate into *winners*, who consume a large amount of bandwidth, and *losers*, who consume very little. This phenomenon develops because the queue is almost always full. The ACK packets received by the *winners* serve as a signal that a buffer space has just been freed, so their packets are rarely dropped. The *losers* are usually retransmitting, at essentially random times, and thus have most of their packets dropped. This analysis is due to Jacobson [Jac88b], and the segregation effect was first pointed out to us in this context by Sturgis [Stu88]. The combination of JK flow control with FCFS gateways produces fair bandwidth allocation among the FTP sources, but the Telnet sources are almost completely shut out. This is because the JK algorithm ensures that the gateway's buffer is usually full, causing most of the Telnet packets to be dropped.

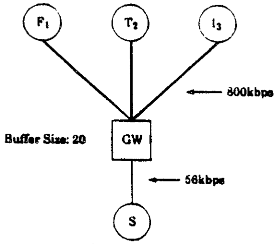


Quantity	Policy	FTP 1	FTP 2	FTP 3	FTP 4	FTP 5	FTP 6	Telnet 7	Telnet 8
Throughput (packets)	G/FCFS	18	1184	1159	3	1149	15	31	3
	G/FQ	178	836	801	600	616	621	96	98
	JK/FCFS	583	583	586	588	583	582	3	0
	JK/FQ	574	579	546	594	590	601	87	96
	DEC/DEC	582	582	583	582	582	582	99	90
	DEC/FQbit	582	582	582	582	582	582	106	97
Average Roundtrip Time	G/FCFS	408	2.18	2.16		2.18	140	116	
	G/FQ	16.8	3.31	4.88	4.83	4.83	4.47	079	078
	JK/FCFS	1.86	1.93	1.93	1.85	1.93	1.88		
	JK/FQ	1.75	1.78	1.19	1.86	2.20	2.16	091	086
	DEC/DEC	.860	860	858	859	858	859	778	783
	DEC/FQbit	860	860	860	860	860	860	088	082
Retransmitted Packets	G/FCFS	43	10	7	6	9	17	25	5
	G/FQ	73	224	176	168	243	159	2	2
	JK/FCFS	57	57	57	57	57	57	6	0
	JK/FQ	83	80	60	64	61	61	0	0
	DEC/DEC	0	0	0	0	0	0	0	0
	DEC/FQbit	0	0	0	0	0	0	2	3
Dropped Packets	G/FCFS	28	5	4	3	5	11	15	2
	G/FQ	33	139	106	88	167	98	0	0
	JK/FCFS	56	56	56	56	56	56	5	0
	JK/FQ	80	76	48	61	57	54	0	0
	DEC/DEC	0	0	0	0	0	0	0	0
	DEC/FQbit	0	0	0	0	0	0	0	0

Scenario 2: Overloaded Gateway

When generic flow control is combined with FQ, the strict segregation disappears. However, the bandwidth allocation is still rather uneven, and the useful bandwidth (rate of nonduplicate packets) is 12% below optimal. Both of these facts are due to the inflexibility of the generic flow control, which is unable to reduce its load enough to prevent dropped packets. This not only necessitates retransmissions but also, because of the crudeness of the timeout congestion recovery mechanism, prevents FTP's from using their fair share of bandwidth. In contrast, JK flow control combined with FQ produced reasonably fair and efficient allocation of the bandwidth. The lesson here is that fair queueing gateways by themselves do not provide adequate congestion control; they must be combined with intelligent flow control algorithms at the sources.

The selective DECbit algorithm manages to keep the bandwidth allocation perfectly fair, and there are no dropped packets or retransmissions. The addition of FQ to the DECbit algorithm retains the fair bandwidth allocation and, in addition, lowers the Telnet delay by a factor of 9. Thus, for each of the three flow control algorithms, replacing FCFS gateways with FQ gateways generally improved the FTP performance and dramatically improved the Telnet performance of this extremely overloaded network.

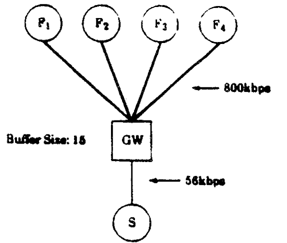


Quantity	Policy	FTP 1	Telnet 2	Ill-Behaved 3
Throughput (packets)	G/FCFS	3	11	3497
	G/FQ	3491	95	5
	JK/FCFS	0	0	3500
	JK/FQ	3489	110	5
	DEC/DEC	166	0	3334
	DEC/FQbit	3493	95	3
Average Roundtrip Time	G/FCFS	1362	2.87	2.97
	G/FQ	716	080	903
	JK/FCFS			2.83
	JK/FQ	716	085	460
	DEC/DEC	3.00		2.99
	DEC/FQbit	641	080	477
Retransmitted Packets	G/FCFS	7	139	0
	G/FQ	0	2	2
	JK/FCFS	2	0	0
	JK/FQ	0	0	0
	DEC/DEC	0	1	0
	DEC/FQbit	0	2	0
Dropped Packets	G/FCFS	7	127	3604
	G/FQ	0	0	6996
	JK/FCFS	2	0	3500
	JK/FQ	0	0	6994
	DEC/DEC	0	1	3667
	DEC/FQbit	0	0	6997

Scenario 3: Ill-Behaved Source

In scenario 3 there is a single FTP and a single Telnet competing with an ill-behaved source. This ill-behaved source has no flow control and is sending packets at twice the rate of the gateway's outgoing line. With FCFS, the FTP and Telnet are essentially shut out by the ill-behaved source. With FQ, they obtain their fair share of bandwidth. Moreover, the ill-behaved host gets much less than its fair share, since when it has its packets dropped it is still charged for that throughput. Thus, FQ gateways are effective *firewalls* that can protect users, and the rest of the network, from being damaged by ill-behaved sources.

We have argued for the importance of considering a heterogeneous set of flow control mechanisms.



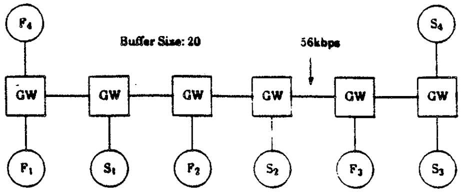
Quantity	Queueing Policy	Generic FTP 1	Generic FTP 2	JK FTP 3	JK FTP 4
Throughput (packets)	FQ	1163	12	1163	1163
	FCFS	1183	1183	560	567
Average Roundtrip Time	FQ	2.15	2.01	2.14	2.14
	FCFS	2.11	2.11	2.07	2.16
Retransmitted Packets	FQ	1	5	2	2
	FCFS	0	0	47	47
Dropped Packets	FQ	1	1	2	2
	FCFS	0	0	48	48

Scenario 4: Mixed Protocols

Scenario 4 has single gateway with two pairs of FTP sources, employing generic and JK flow control, respectively. With a FCFS gateway, the generic flow controlled pair has higher throughput than the JK pair. However, with a FQ gateway, the situation is reversed (and the generic sources have segregated). Note that the FQ gateway has provided incentive for sources to implement JK or some other intelligent flow control, whereas the FCFS gateway makes such a move sacrificial.

Certainly not all of the relevant behavior of these algorithms can be gleaned from single gateway networks. Scenario 5 has a multinode network with four FTP sources using different network paths. Three of the sources have short nonoverlapping conversations and the fourth source has a long path that intersects each of the short paths. When FCFS gateways are used with generic or JK flow control, the conversation with the long path receives less than 60% of its fair share. With FQ gateways, it receives its full fair share. Furthermore, the selective DECbit algorithm, in keeping the average queue size small, wastes roughly 10% of the bandwidth (and the conversation with the long path, which should be helped by any attempt at fairness, ends up with less bandwidth than in the generic/FCFS case).

Scenario 6 involves a more complicated network, combining lines of several different bandwidths. None of the gateways are overloaded so all combinations of flow control and queueing algorithms function smoothly. With FCFS, sources 4 and 8 are not limited by the available bandwidth, but by the delay their ACK packets incur waiting behind FTP packets. The total throughput increases when the FQ gateways are used because the small ACK packets are given priority.



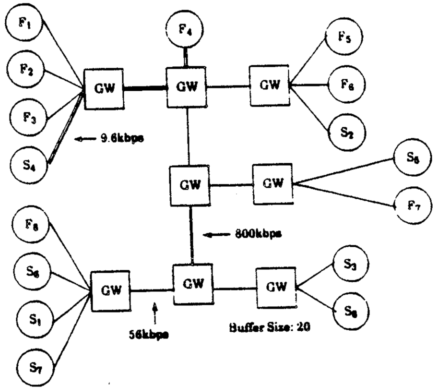
		FTP			
Quantity	Policy	1	2	3	4
Throughput (packets)	G/FCFS	2500	2500	2500	1000
	G/FQ	1750	1750	1750	1750
	JK/FCFS	2500	2500	2500	1000
	JK/FQ	1750	1750	1750	1750
	DEC/DEC	2396	2406	2377	783
	DEC/FQbit	1750	1750	1750	1750
Average Roundtrip Time	G/FCFS	1.00	1.00	1.00	2.5
	G/FQ	1.43	1.43	1.43	1.43
	JK/FCFS	1.00	1.00	1.00	2.5
	JK/FQ	1.43	1.43	1.43	1.43
	DEC/DEC	617	626	618	1.56
	DEC/FQbit	1.43	1.43	1.43	1.43
Retransmitted Packets	G/FCFS	0	0	0	0
	G/FQ	0	0	0	0
	JK/FCFS	0	0	0	0
	JK/FQ	0	0	0	0
	DEC/DEC	0	0	0	0
	DEC/FQbit	0	0	0	0
Dropped Packets	G/FCFS	0	0	0	0
	G/FQ	0	0	0	0
	JK/FCFS	0	0	0	0
	JK/FQ	0	0	0	0
	DEC/DEC	0	0	0	0
	DEC/FQbit	0	0	0	0

Scenario 5: Multihop Path

For the sake of clarity and brevity, we have presented a fairly clean and uncomplicated view of network dynamics. We want to emphasize that there are many other scenarios, not presented here, where the simulation results are confusing and apparently involve complicated dynamic effects. These results do not call into question the efficacy and desirability of fair queueing, but they do challenge our understanding of the collective behavior of flow control algorithms in networks.

5. Discussion

In an FCFS gateway, the queueing delay of packets is, on average, uniform across all sources and directly proportional to the total queue size. Thus, achieving ambitious performance goals, such as low delay for Telnet-like sources, or even mundane ones, such as avoiding dropped packets, requires coordination among all sources to control the queue size. Having to rely on source flow control algorithms to solve this control problem, which is extremely difficult in a maximally cooperative environment and impossible in a noncooperative one, merely reflects the inability of FCFS gateways to distinguish between users and to allocate bandwidth, promptness, and buffer space independently.



		FTP							
Quantity	Policy	1	2	3	4	5	6	7	8
Throughput (packets)	G/FCFS	196	202	196	187	1832	1277	1960	1946
	G/FQ	193	192	192	577	1560	1560	1622	3274
	JK/FCFS	196	202	196	190	1833	1276	1961	1946
	JK/FQ	193	193	193	577	1558	1660	1619	3273
	DEC/DEC	176	200	212	288	2034	827	1337	2684
	DEC/FQbit	193	192	192	577	1568	1620	1566	3273
Average Roundtrip Time	G/FCFS	12.8	12.4	12.8	13.3	1.36	1.96	1.27	1.26
	G/FQ	12.8	12.9	12.8	4.26	1.60	1.60	1.54	728
	JK/FCFS	12.8	12.4	12.8	13.2	1.36	1.96	1.27	1.26
	JK/FQ	13.0	13.0	13.0	4.33	1.60	1.60	1.54	729
	DEC/DEC	3.36	3.80	3.63	4.66	960	1.06	775	737
	DEC/FQbit	6.46	6.49	6.49	4.33	1.53	1.60	1.33	729
Retransmitted Packets	G/FCFS	0	0	0	0	0	0	0	0
	G/FQ	0	0	0	0	0	0	0	0
	JK/FCFS	0	0	0	0	0	0	0	0
	JK/FQ	0	0	0	0	0	0	0	0
	DEC/DEC	7	8	5	1	0	0	0	0
	DEC/FQbit	0	0	0	0	0	0	0	0
Dropped Packets	G/FCFS	0	0	0	0	0	0	0	0
	G/FQ	0	0	0	0	0	0	0	0
	JK/FCFS	0	0	0	0	0	0	0	0
	JK/FQ	0	0	0	0	0	0	0	0
	DEC/DEC	0	0	0	0	0	0	0	0
	DEC/FQbit	0	0	0	0	0	0	0	0

Scenario 6: Complicated Network

In the design of the fair queueing algorithm, we have attempted to address these issues. The algorithm does allocate the three quantities separately. Moreover, the promptness allocation is not uniform across users and is somewhat tunable through the parameter δ. Most importantly, fair queueing creates a firewall that protects well-behaved sources from their uncouth brethren. Not only does this allow the current generation of flow control algorithms to function more effectively, but it creates an environment where users are rewarded for devising more sophisticated and responsive algorithms. The game-theoretic issue first raised by Nagle, that one must change the rules of the gateway's game so that good source behavior is encouraged, is crucial in the design of gateway algorithms. A formal game-theoretic analysis of a simple gateway model (an exponential server with N Poisson sources) suggests that fair queueing algorithms make self-optimizing source behavior result in fair, protective, nonmanipulable, and stable networks; in fact, they may be the only reasonable queueing algorithms to do so [She89a].

Our calculations show that the fair queueing algorithm is able to deliver low delay to sources using less than their fair share of bandwidth, and that this delay is insensitive to the window sizes being used by the FTP sources. Furthermore, simulations indicate that, when combined with currently available flow control algorithms, FQ delivers satisfactory congestion control in a wide variety of network scenarios. The combination of FQ gateways and DECbit flow control was particularly effective. However, these limited tests are in no way conclusive. We hope, in the future, to investigate the performance of FQ under more realistic load conditions, on larger networks, and interacting with routing algorithms. Also, we hope to explore new source flow control algorithms that are more attuned to the properties of FQ gateways.

In this paper we have compared our fair queueing algorithm with only the standard first-come-first-serve queueing algorithm. We know of three other widely known queueing algorithm proposals. The first two were not intended as a general purpose congestion control algorithms. Prue and Postel [Pru87] have proposed a type-of-service priority queueing algorithm, but allocation is not made on a user-by-user basis, so fairness issues are not addressed. There is also the Fuzzball selective preemption algorithm [Mill87,88] whereby the gateways allocate buffers fairly (on a source basis, over all of the gateway's outgoing buffers). This is very similar to our buffer allocation policy, and so can be considered a subset of our FQ algorithm. The Fuzzballs also had a form of type-of-service priority queueing but, as with the Prue and Postel algorithm, allocations were not made on a user-by-user basis. The third policy is the Random-Dropping (RD) buffer management policy in which, when the buffer is overloaded, the packet to be dropped is chosen at random [Per89, Jac88ab]. This algorithm greatly alleviates the problem of segregation. However, it is now generally agreed that the RD algorithm does not provide fair bandwidth allocation, is vulnerable to ill-behaved sources, and is unable to provide reduced delay to conversations using less than their fair share of bandwidth [She89b, Zha89, Has89].

There are two objections that have been raised in conjunction with fair queueing. The first is that some source-destination pairs, such as file server or mail server pairs, need more than their fair share of bandwidth. There are several responses to this. First, FQ is no worse than the status quo. FCFS gateways already limit well-behaved hosts, using the same path and having only one stream per source destination pair, to their fair share of bandwidth. Some current bandwidth hogs achieve their desired level of service by opening up many streams, since FCFS gateways implicitly define streams as the unit of *user*. Note that that there are no controls over this mechanism of gaining more bandwidth, leaving the network vulnerable to abuse. If desired, however, this same trick can be introduced into fair queueing by merely changing the notion of user. This would violate layering, which is admittedly a serious drawback. A better approach is to confront the issue of allocation directly by generalizing the algorithm to allow for arbitrary bandwidth priorities. Assign each pair a number n_α which represents how many queue slots that conversation gets in the bit-by-bit round robin. The new relationships are $N_{ac} = \sum n_\alpha$ with the sum over all active conversations, and P_i^α is set to be $1/n_\alpha$ times the true packet length. Of course, the truly vexing problem is the politics of assigning the priorities n_α. Note that while we have described an extension that provides for different relative shares of bandwidth, one could also define these shares as absolute fractions of the bandwidth of the outgoing line. This would guarantee a minimum level of service for these sources, and is very similar to the *Virtual Clock* algorithm of Zhang [Zha89].

The other objection is that fair queueing requires the gateways to be smart and fast. There is technological question of whether or not one can build FQ gateways that can match the bandwidth of fibers. If so, are these gateways economically feasible? We have no answers to these questions, and they do indeed seem to hold the key to the future of fair queueing.

6. Acknowledgements

The authors gratefully acknowledge H. Murray's important role in designing an earlier version of the fair queueing algorithm. We wish to thank A. Dupuy for assistance with the Nest simulator. Fruitful discussions with J. Nagle, E. Hahne, A. Greenberg, S. Morgan, K. K. Ramakrishnan, V. Jacobson, M. Karels, D. Greene and the End-to-End Task Force are also appreciated. In addition, we are indebted to H. Sturgis for bringing the segregation result in section 4 to our attention, and to him and R. Hagmann and C. Hauser for the related lively discussions. We also wish to thank the group at MIT for freely sharing their insights: L. Zhang, D. Clark, A. Heybey, C. Davin, and E. Hashem.

7. References

[DEC87a] R. Jain and K. K. Ramakrishnan. "Congestion Avoidance in Computer Networks with a Connectionless Network Layer, Part I-Concepts, Goals, and Alternatives", DEC Technical Report TR-507, Digital Equipment Corporation, April 1987.

[DEC87b] K. K. Ramakrishnan and R. Jain. "Congestion Avoidance in Computer Networks with a Connectionless Network Layer, Part II-An Explicit Binary Feedback Scheme", DEC Technical Report TR-508, Digital Equipment Corporation, April 1987.

[DEC87c] D.-M. Chiu and R. Jain, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer, Part III-Analysis of Increase and Decrease Algorithms", DEC Technical Report TR-509, Digital Equipment Corporation, April 1987.

[DEC87d] K. K. Ramakrishnan, D.-M. Chiu, and R. Jain "Congestion Avoidance in Computer Networks with a Connectionless Network Layer, Part IV-A Selective Binary Feedback Scheme for General Topologies", DEC Technical Report TR-510, Digital Equipment Corporation, November 1987.

[Fra84] A. Fraser and S. Morgan, "Queueing and Framing Disciplines for a Mixture of Data Traffic Types", AT&T Bell Laboratories Technical Journal, Volume 63, No. 6, pp 1061-1087, 1984.

[Gaf84] E. Gafni and D. Bertsekas, "Dynamic Control of Session Input Rates in Communication Networks", IEEE Transactions on Automatic Control, Volume 29, No. 10, pp 1009-1016, 1984.

[Ger80] M. Gerla and L. Kleinrock, "Flow Control: A Comparative Survey", IEEE Transactions on Communications, Volume 28, pp 553-574, 1980.

[Gre89] A. Greenberg and N. Madras, private communication, 1989.

[Hah86] E. Hahne, "Round Robin Scheduling for Fair Flow Control in Data Communication Networks", Report LIDS-TH-1631, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts, December, 1986.

[Has89] E. Hashem, private communication, 1989.

[Hey89] A. Heybey and C. Davin, private communication, 1989.

[ISO86] International Organization for Standardization (ISO), "Protocol for Providing the Connectionless Mode Network Service", Draft International Standard 8473, 1986.

[Jac88a] V. Jacobson, "Congestion Avoidance and Control", ACM SigComm Proceedings, pp 314-329, 1988.

[Jac88b] V. Jacobson, private communication, 1988.

[Jai86] R. Jain, "Divergence of Timeout Algorithms for Packet Retransmission", Proceedings of the Fifth Annual International Phoenix Conference on Computers and Communications, pp 1162-1167, 1987.

[Kar87] P. Karn and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", ACM SigComm Proceedings, pp 2-7, 1987.

[Kat87] M. Katevenis, "Fast Switching and Fair Control of Congested Flow in Broadband Networks", IEEE Journal on Selected Areas in Communications, Volume 5, No. 8, pp 1315-1327, 1987.

[Lo87] C.-Y. Lo, "Performance Analysis and Application of a Two-Priority Packet Queue", AT&T Technical Journal, Volume 66, No. 3, pp 83-99, 1987.

[Lua88] D. Luan and D. Lucantoni, "Throughput Analysis of an Adaptive Window-Based Flow Control Subject to Bandwidth Management", Proceedings of the International Teletraffic Conference, 1988.

[Man89] A. Mankin and K. Thompson, "Limiting Factors in the Performance of the Slo-start TCP Algorithms", preprint.

[Mor89] S. Morgan, "Queueing Disciplines and Passive Congestion Control in Byte-Stream Networks", IEEE INFOCOM '89 Proceedings, pp 711-720, 1989.

[Mil87] D. Mills and W.-W. Braun, "The NSFNET Backbone Network", ACM SigComm Proceedings, pp 191-196, 1987.

[Mil88] D. Mills, "The Fuzzball", ACM SigComm Proceedings, pp 115-122, 1988.

[Nag84] J. Nagle, "Congestion Control in IP/TCP Networks, Computer Communication Review, Vol 14, No. 4, pp 11-17, 1984.

[Nag85] J. Nagle, "On Packet Switches with Infinite Storage", RFC 896 1985.

[Nag87] J. Nagle, "On Packet Switches with Infinite Storage", IEEE Transactions on Communications, Volume 35, pp 435-438, 1987.

[Nes88] D. Bacon, A. Dupuy, J. Schwartz, and Y. Yemini, "Nest: A Network Simulation and Prototyping Tool", Dallas Winter 1988 Usenix Conference Proceedings, pp. 71-78, 1988.

[Per89] IETF Performance and Congestion Control Working Group, "Gateway Congestion Control Policies", draft, 1989.

[Pos81] J. Postel, "Internet Protocol", RFC 791 1981.

[Pru88] W. Prue and J. Postel, "A Queueing Algorithm to Provide Type-of-Service for IP Links", RFC1046, 1988.

[She89a] S. Shenker, "Game-Theoretic Analysis of Gateway Algorithms", in preparation, 1989.

[She89b] S. Shenker, "Comments on the IETF Performance and Congestion Control Working Group Draft on Gateway Congestion Control Policies", unpublished, 1989.

[Stu88] H. Sturgis, private communication, 1988.

[USC81] USC Information Science Institute, "Transmission Control Protocol", RFC 793, 1981.

[Xer81] Xerox Corporation, "Internet Transport Protocols", XSIS 028112, 1981.

[Zha89] L. Zhang, "A New Architecture for Packet Switching Network Protocols", MIT Ph. D. Thesis, forthcoming, 1989.

How Slow Is One Gigabit Per Second?

Craig Partridge

Abstract

At first blush, one would expect that increasing data network transfer rates by two orders of magnitude (from the ubiquitous 10 Mbit speed of today's LANs to the greater than 1 gigabit-per-second speeds we expect of networks in the early 1990s) would severely impact our choice of network protocols and architectures. This report presents the strawman argument that, in fact, moving to one-gigabit data rates presents surprisingly *few* problems.

1. Introduction

Recently there has been considerable interest in gigabit-per-second networks for three reasons:

(1) The demand for bandwidth on data communications networks is growing rapidly (the current growth curve is steeply exponential) and it appears that by the mid-1990s there will be an aggregate demand for network bandwidth that approaches one gigabit per second.

(2) Innovative applications under development in research labs have large bandwidth requirements, often approaching a gigabit per second for a *single* application.

(3) It has been an article of faith for many researchers in the networking community that the current networking architectures (and in particular, the datagram-oriented architectures) would have difficulty scaling to very high speeds. The exact speed at which the architectures would break down has never been stated, but the popular impression is that one gigabit per second is sufficiently fast to exceed the limitations of the current architectures.

For several months I have participated in a research effort intended to help shape the architecture of gigabit-per-second networks. This effort involved spending considerable time trying to find the constraints that gigabit-per-second data rates placed on network architectures. In other words, Accepting the idea in point three above that gigabit-per-second networks would severely strain or break current architectures, I hoped that by discovering where those strains or breaks occurred, I would be better able to recommend changes or innovations necessary for a new, gigabit-oriented, network architecture.

The research yielded little reason to believe that a gigabit-per-second data rate alone would force us to adopt a new network architecture. This is not to say that gigabit-per-second networks present no difficult problems to the researcher, but rather that the problems presented are less extraordinary than one might initially suspect and do not appear to require us to develop a new network architecture to solve them.[1]

This paper was written under DARPA Contract MDA903-89-C-0020. The opinions in this paper are the author's and do not necessarily reflect those of the Defense Advanced Research Projects Agency. The author's address is BBN Systems and Technologies Corporation, 10 Moulton St, Cambridge MA 02138, or via electronic mail, craig@bbn.com.

[1] We may still employ a novel network architecture when building gigabit-per-second networks. Some of the research work on gigabit applications suggests that we may need new mechanisms for communicating between network peers, and some researchers have argued that as networking matures, we must incorporate additional functionality such as policy routing into our networks. Such new mechanisms or functionality might cause us to develop a new network architecture to support them. This report simply argues that speed alone is not a good justification for change.

By examining both the performance of individual components and more systemic issues, such as problems of flow control and buffering, on gigabit-per-second networks, this paper attempts to support the contention that moving to one gigabit data rates presents relatively few problems. The discussion is meant to remain neutral about the particular architecture to use; the goal of the paper is simply to show that neither the virtual-circuit nor the datagram architectures are precluded by gigabit-per-second speed networks.[2]

2. Basic Assumptions

This argument rests on two basic assumptions:

- The average datagram or packet sizes on gigabit networks will be larger than those on current networks, such as the Internet.
- It is possible to make a good estimate of the processing capacity of future routers and hosts.

The remainder of this section explains these assumptions.

2.1. Average Datagram or Packet Sizes

The average datagram or packet[3] size on a network largely determines the rate at which routers need to be able to switch data and influences how hosts use the network. As a result, estimating the average packet size is essential to any argument about a router or host's ability to send or receive at gigabit speeds.

For the purposes of this paper, assume that the average packet sizes for hosts and routers on the Internet (about 100 octets for gateways and somewhere between 100 and 500 octets for hosts) is a conservative estimate of the average packet size on a gigabit network.[4]

The current packet size is believed to be a conservative estimate for two reasons.

First, the proportion of bulk data transfer to interactive data transfer on gigabit networks will probably increase. Currently, one of the primary reasons for building gigabit networks is that researchers want to be able to exchange large quantities of data very fast. Thus one would expect to see more large packets (carrying bulk traffic) on the network in the future than we do currently. Additional large packets will increase the average packet size.

Second, the maximum transmission units (MTU) on gigabit networks will probably be higher than the MTU on current networks. Certainly the trend is toward large MTUs.[5] The FDDI MTU is substantially larger than the Ethernet MTU. Larger MTUs imply that bulk data transfers can use fewer and larger packets; again, all other factors being equal, the average packet size will increase.

Averages don't tell the whole story, however. The average packet size can vary dramatically depending on what network users are doing at a given moment. Smaller packets make it harder to achieve high speeds, and larger packets make high speeds easier. But predicting how

---

[2] The fact that many of the numbers in this paper are derived from TCP/IP networks reflects the fact that this information was most easily available.

[3] From here on, the word *datagram* will be reserved for use only when networks using a datagram architecture are being discussed. In cases where the architecture could be either datagram or virtual circuit, the term *packet* will be used.

[4] The average packet size estimates come from a variety of sources. RFC 1009 estimates the packet size at routers to be about 100 octets. The packet size for hosts comes from observations on networks at BBN, ISI, MIT, and Sun Microsystems.

[5] Except for ATM networks, but ATM is essentially a multiplexing scheme. Hosts and routers will almost certainly use a larger transmission unit, which is then broken down into ATM cells for transmission.

gigabit networks will be used, in enough specificity to do detailed estimates of traffic mixes, is all but impossible. Therefore, estimating requirements based on conservative average packet sizes seems a good approach.

2.2. Building Future Hosts and Routers

To discuss the computing limitations of hosts and routers in a gigabit network, one must identify candidate hardware that could be used to build them. This paper discusses components that that are currently in test production, as a reasonably conservative way to predict what hardware would be available in the early 1990s. Furthermore, the focus is on inexpensive components, the sorts of processors and memory one would expect to have in a workstation class machine. Cray routers would be interesting, but probably prohibitively expensive in real life.

The key characteristics of the candidate architectures are:

- A 60 to 70 million instructions-per-second RISC CPU. The logical near-term successor to the CPUs in workstations today are the RISC CPUs currently in test production or advanced development. Intel currently has CPUs of this speed in test production.
- 64-bit data paths and word sizes. Currently most machines have 32-bit data paths, but there is plenty of room on most VLSI chips for more pins, and the industry is apparently concluding that they need the larger data paths. Note that using a 64-bit data path, a one gigabit-per-second transfer speed requires only a 16-Megahertz clock. Inexpensive RAM memory already exists that can clock out bits at this speed. Thus, modulo the problems of multiple address ports, discussed in the section on interfaces below, one can assume that a system can move data internally at gigabit rates.

3. Network Components

A likely place to start the strawman argument is with the components of the network themselves. Clearly, if our choices for gigabit speed network components are very limited, our choice of network architecture may be constrained.

This section examines three types of network components: the hosts or hosts that use the network, the switches or routers that direct the data flowing between hosts, and the physical links over which the traffic actually passes. This troika, in some form, appears in all network architectures. The discussion progresses bottom-up, starting with the links and working its way up to the hosts.

The general goal of this section is to suggest that we have enough processing or computing power in all parts of the network to handle gigabit-speed data traffic.

3.1. Link Speed And Interfaces

The first question is whether we can transmit data at one gigabit per second between systems. The answer is yes, and indeed the problems involved seem to be relatively well understood, and there is a broad choice of methods for building gigabit networks. See Acampora and Karol's article for a good discussion [2].

3.2. Routers

Routers are responsible for switching data through the network between an originating host and a destination host. In varying forms, with different mixes of hardware and software, they are a component of all major network architectures. Examples include routers between IP networks and X.75 gateways between X.25 networks.

Historically, the dominant cost in a router has been the cost of switching the packet. As a result, the key question in this section is: Do we have enough CPU bandwidth to do per-packet

processing required?

The model used here is the basic architecture used by most of today's packet switches: one or more general-purpose CPUs, memory, and network interfaces, plus the possible addition of some special-purpose hardware (often used for address or route lookup). In these configurations, the bottleneck is usually the CPU.

Why is the CPU the key bottleneck? Because routers need to examine only a small part of each packet (the header). Therefore the other obvious bottleneck, the data bus, is underloaded, unless an entire packet needs to be moved across the bus. Newer systems work very hard not to transfer entire packets; they put multiple interfaces on the same I/O board, sharing the same memory. In these configurations the CPU need simply copy the header from and to the I/O memory and set a bit on the device to switch a packet from one interface to another. Thus, in practice, the major resource that is usually in short supply is the CPU.

To estimate how much CPU time we have per packet, we estimate the number of packets per second that a router will see at gigabit speeds. The method used is to extrapolate from typical performance expected today.

Current wisdom in the TCP/IP community is that a router between two Ethernets must be able to switch at between 6,000 and 10,000 packets per second.[6] Doing a simple linear extrapolation, if we need to switch 6,000 to 10,000 packets at Ethernet speeds (10 Mbit/sec) we will need to switch 600,000 to 1,000,000 packets at gigabit rates. Or, to look at the problem another way, we need to be able to switch a packet in 1 to 1.6 microseconds. Note that if we have a pipelined architecture, we can take longer to switch the packet, but no stage in the pipeline can take longer than 1 to 1.6 microseconds.

Consider a router built with one processor (currently most routers use at least two processors). Given the expected processor speed of 60 MIPS, we have 60 to 100 instructions in our single CPU with which to handle the packet.

Sixty instructions per packet is almost certainly acceptable for virtual circuit architectures which amortize circuit set-up costs over several packets. Once a circuit is set up, switching from one interface to another can be done with a handful of instructions.[7] The real question is whether 60 to 100 instructions is enough to switch a packet.

The answer is, probably. Certainly it is fast enough to switch simple packets. The DEC LANBridge 100 switches Ethernet frames using about 50 instructions per packet (with some hardware assist).[8] More complex packets such IP packets are somewhat harder to switch.

Using 32-bit processors, current routers typically switch an IP packet using between 100 and 150 instructions plus some code to run the device drivers. That instruction count should be reduced somewhat for a 64-bit processor because many operations (such as additions in the checksum and checking header fields against bit masks) would use 64-bit integer or word sizes instead of 32 bits. So, if our device drivers are sufficiently simple to drive, or if we can make some switching functions faster through use of hardware, we should be able to switch even moderately complex packets like IP at speeds of one gigabit per second.

---

[6] This value is consistent with an average packet size of a little over 100 octets for gateways. If we assume all packets are switched off the Ethernet, these rates convert to an average packet size of between 100 and 200 octets.

[7] For example, using the Butterfly gateway-to-gateway header protocol, which is used to specify a path through a series of Butterfly gateways, switching a packet requires fewer than 30 instructions.

[8] Personal communication from Ross Callon.

3.3. Hosts

Determining what is required of a host on a gigabit network is more difficult than determining what is required of a router, because hosts and routers are very different.

At least three key differences affect this evaluation:

- Historically, a single host has not been expected to consume the entire bandwidth of the network it is attached to. The introduction to this paper notes that some applications currently being developed may require that their host systems be able to support gigabit transfer rates, but these applications also tend to have special traffic characteristics (i.e., they use maximum-size packets) that make them different from run-of-the-mill host traffic.
- Even ignoring the question of new applications, the average packet size for hosts appears to vary widely, from about 100 to over 500 bytes, depending on traffic mix. In particular, hosts that participate in distributed filesystems and do paging over the network appear to have larger average packet sizes.
- Finally, hosts usually want to look at the data, so any calculation of resource requirements must allow for some computation on the data received.[9]

Note that hosts also always need to checksum their data, while routers often do not. This difference does not play a role the discussion below, because it is reasonably easy to put a checksum into hardware, which is operated in parallel with other protocol operations.

3.4. Host Computational Requirements

To determine a host's ability to process data at one gigabit per second, we first need some estimate of the computation that an end system is expected to do on a packet. This estimate can be divided into three parts: protocol processing, application processing, and operating system overhead.

For an estimate of protocol processing costs I used the measurements by Clark, Jacobson, Romkey, and Salwen for TCP/IP processing overhead [2]. Note that their measurements were done on code that does not include many of the current innovations, such as header prediction, that can dramatically reduce processing time. Their measurements showed that sending or receiving a TCP segment took about 300 CISC instructions, inflated to 400 instructions for RISC processors. These numbers include IP processing, TCP processing (exclusive of checksum), and buffer processing, but not device driver support. The checksum does not need to be done in software, so we don't have to add this in. Device driver code can be very large or very small, depending upon the device. For this discussion, assume a device driver of moderate complexity requiring about 400 instructions. This gives a protocol processing cost of about 800 instructions.

To this estimate some instructions for context switching and other operating system overhead must be added. I've estimated this value at about 200 instructions.

Finally, some time must be allowed for processing the data. The data can be processed only as fast as every word can be touched. In other words, a segment can't be processed faster than the host can read each word. Thus the total estimated per-packet processing cost is around $1000+l/w$ instructions, where l is the packet length and w is the host's word size.

---

[9] Note that some bulk-data applications do not examine their data. For example, receivers of file or video transfers are often moving data straight from the network buffers into disk or video buffers, without further application processing. Including application processing time may overstate the host computation requirements in some cases.

Although these numbers are taken from datagram networks, I believe they are the same or lower for virtual circuit networks.[10]

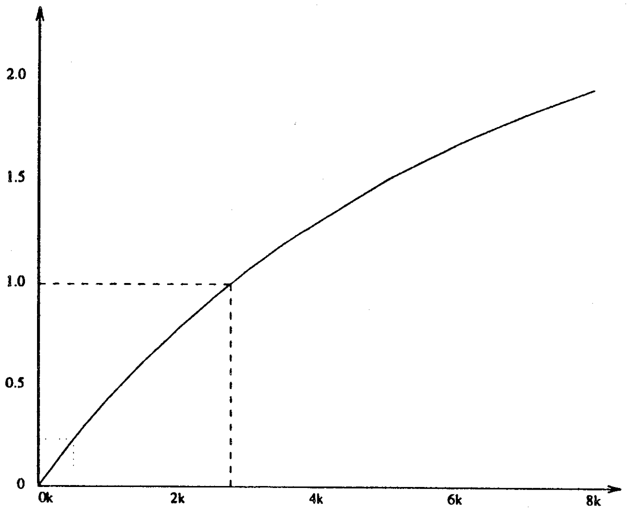


Figure 1: Packet Size (in bytes) vs. Bandwidth (in 10^9 bits)

3.5. Hosts And Variable Packet Sizes

The average packet size for hosts appears to vary dramatically depending upon the applications the host is currently running. Casual observations at various sites show average packet sizes ranging from 100 to 700 bytes and a moderate variation over time, as application mixes change during the working day. As a result, I've chosen, to make the packet size the free variable in the equation relating bandwidth, packet sizes, and available instructions.

Figure 1 shows the bandwidth a host can consume versus the packet size used to send the data. To generate the graph, the number of instructions required to process one packet of a given size was computed. This number was then used to compute the maximum number of packets that could be processed by a 60-MIPS CPU, and then the packet size was multiplied by the number of packets that could be processed to get the consumable bandwidth.[11] The packet sizes range up to 8 Kbytes, which is double the FDDI packet size. (If FDDI, which is 10 times faster than Ethernet, uses an MTU somewhat more than twice as big as Ethernet's, it does not seem unreasonable

---

[10] The transport processing (exclusive of checksum), application, and operating system costs are the same. The differences, if any, appear in the network layer and device driver code. In practice, the device drivers for circuit networks tend to be a bit more complex than those for datagram networks, because a single device supports multiple circuits, and the multiplexing and demultiplexing of data and commands for multiple circuits into the command registers of a single device is complex and often CPU-intensive. However, the network layer code in virtual circuit networks is typically somewhat simpler than the comparable code for datagram networks.

to assume that gigabit network, 10 times faster than FDDI, might use an MTU twice again as big).

At least two pieces of information are clear from the graph. First, achieving gigabit speeds will be possible for hosts. As the dashed line shows, to achieve one gigabit throughput, a host (or application) need only have an average packet size of about 3 Kbytes, which will probably be well within the MTU size of gigabit networks (FDDI, for example, has a 4 Kbyte MTU).

The second item of note is that, even with small average packet sizes, a 60-MIPS host could consume a large fraction of a gigabit. The dotted line shows that for a 512-byte average packet size, the system could use nearly a quarter of a gigabit. Even at a 100-byte average size, a system could use nearly 50,000,000 bits per second, about one twentieth of a gigabit capacity.

Both of these observations are encouraging. They suggest that in normal use (with average packet sizes below 1,000 bytes) hosts will still be able to use an appreciable fraction of a gigabit bandwidth, and that, should those systems have applications that need to fill a gigabit pipe, gigabit speeds can be achieved using a moderately large packet size.

Is it fair to ask applications to use large packet sizes to achieve gigabit speeds? In almost all cases, yes. Larger packets are more efficient for the network to process, as well as for routers and hosts to process. Even interactive traffic such as remote terminal applications can usually benefit from aggregation of single keystrokes into larger packets: witness Nagle's algorithm [5] and the DEC LAT protocol. The one exception to this rule is applications with strict time constraints, such as digitized speech. But these applications typically have low bandwidth requirements, and do not need to achieve gigabit throughput between peers.

3.6. Yet Another Way to Look at Things

Here is one more way to look at the question of processing and resources. So far, this discussion has typically measured packet costs in instructions or bandwidth consumed. But another useful way to look at the costs is in terms of time. Given a gigabit data stream, packed as tightly as possible with packets of size *n*, how long does a component have (in seconds), to process that packet?

Figure 2 shows this relationship between packet size and time, with time measured in microseconds.

This graph gives a generally comforting picture. The time available to process even small packets is not impossible to achieve, and for larger packets, such as those of 6 Kbytes or larger, the processing time available is feasible today (the DEC LANbridge switches a packet in 50 µs).

4. Flow Control and Buffering

This section of the paper concerns systemic problems. Having argued that individual components of a one gigabit network can work comfortably at gigabit speeds using either a datagram or virtual circuit model, I will now consider whether the gigabit network as a whole is stable.

This issue is not examined simply for completeness; it is a pressing question. Gigabit networks will be moving immense amounts of data at very large speeds. Errors or faults in the system may in a short time lead to megabytes or even gigabytes of misrouted, misdelivered, rerouted or, possibly, discarded data. The ease with which large amounts of data can meet such fates is often viewed as a serious problem for gigabit networks. Furthermore, this problem appears likely to get worse as speeds increase. Gigabit speed networks pass data at the speed of light. Additional bandwidth will be achieved by packing more bits per centimeter of light, not by reducing the network delay, so the delay-×-bandwidth product, the amount of data that can be in-flight at any one time, will continue to increase. Thus, if there is a problem with managing large flows, it can only get worse as networks go faster.

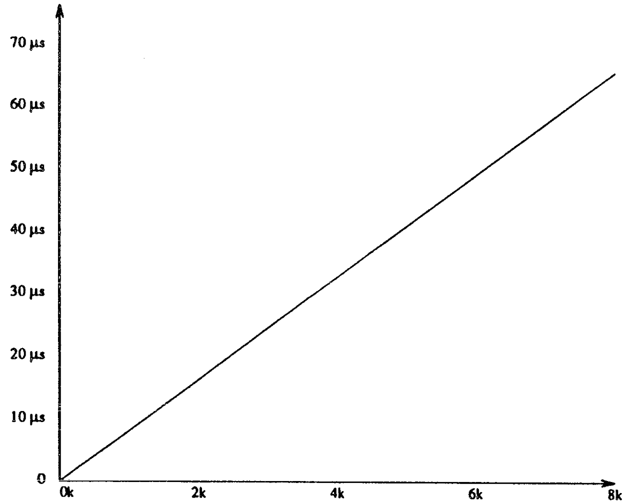


Figure 2: Packet Size (in bytes) vs. Time to Process (in µ-sec)

However, the problems of moving large data are not unmanageable. First, measured in the context of a gigabit speed network, megabytes of data, however mishandled, are much like misdirecting a few large packets in a current-day network. For example, discarding a megabyte of data in transit on a gigabit network is comparable discarding a single 576-byte IP datagram on today's 56-kilobit networks. In other words, we have to recalibrate our impressions of "big" to conform to a gigabit world. Second, once you are willing to redefine "big," our current control algorithms, perhaps with minor tweaking, actually appear to scale reasonably well to the gigabit realm.

4.1. Host Buffering

One concern about gigabit networks is how much buffer space is required by hosts. To transfer data reliably from one host to another, the sending host must buffer data which has been sent but not acknowledged. At the same time, for efficient use of bandwidth, it is important that a sender of data be able to put multiple packets of data in flight before waiting for an acknowledgement. It is also important for attaining good throughput that the receiving host buffer (as opposed to discarding) any packets received out of order [4].

---

[11] As a sanity check on this model, I also did a small test with some real life numbers. Jacobson has reported that he can saturate an Ethernet with a file transfer using a 2-MIPS workstation and 1536-byte packet sizes. He further noted that the workstation still had free cycles. Assuming the same 1000 instructions + 384 instructions for handling data and an additional 384 instructions for the software checksum (which we assume is handled in hardware at gigabit speeds), we get an expected speed of nearly 14 megabits, consistent with Jacobson's observations. (Actually, he estimates his maximum bandwidth to be somewhat higher thatn 14 megabits/second).

Clearly, if the buffering requirements are unattainable, hosts will be unable to achieve gigabit transfer speeds, because they are unable to buffer enough data to keep a gigabit link filled.

4.1.1. Basic Buffering Requirements

How much buffering is required? The maximum amount of buffering required is the delay-x-bandwidth product for one-round trip time over the longest possible path.[12] The key problem is estimating the round-trip time.

A naive method is simply to compute the round-trip time through pure fiber for a plausible longest path and determine how much buffering is required in the worst-case situation, where all data is being sent via the longest possible path. For this paper the longest path will be assumed to be a round-trip across North America, about 6,000 miles. At the speed of light through fiber, this distance takes about 47 milliseconds to traverse. The delay-x-bandwidth product is 5.9 megabytes.

A more reasonable estimate of the round-trip time is that of the long lines carriers, which reportedly promise a delay of 60 milliseconds between points-of-presence, which gives a round-trip time of no less than 120 milliseconds. The delay-x-bandwidth product is 15 megabytes.

Six megabytes of buffering is not unreasonable. Current workstations vary widely in the amount of memory they support, but a memory size of between 4 and 12 megabytes is common. A nice rule of thumb is that memory sizes roughly double every two years for constant price. So following the practice of trying to keep price constant, we might expect to see memory sizes of 8 to 24 megabytes, and in four years, 16 to 48 megabytes. Certainly, at the four-year mark, reserving 6 megabytes of memory for network buffers seems plausible.

Buffering 15 megabytes appears more difficult. It is certainly possible to add enough memory to buffer 15 megabytes (today's large DRAM chips allow us to cram enough memory on a board), but the extra memory will increase the cost of our systems.

4.1.2. The Effects of Switching Delays on Buffering

Another point to consider about buffering is how much delay is introduced by switching between networks; i.e., how much additional host buffering do we need to accomodate the delays in our gigabit equivalents of IP routers or X.25 switches?

The answer appears to be that switching delay should have a negligible effect. Recall that Section 3.2 contained the calculation that no stage in a packet switch or router could take longer than 1 to 1.6 microseconds if it was to keep up with its expected traffic load at gigabit rates. To increase the necessary host buffering by one megabyte on a gigabit network, the delay must increase by 8 milliseconds. Thus, we would have to insert 5,000 to 8,000 switching elements into the round-trip path between hosts to cause the host buffering requirements to increase by one megabyte. Or, thinking of this another way, if we put 100 switching elements into the round-trip path, the additional buffering required is only 12.5 to 20 Kbytes![13]

---

[12] Why one round-trip time? Two reasons: (1) if one uses less than one round-trip time's worth of buffering, and if the network is otherwise idle, the host will have to stop sending before the network is full; (2) if on the other hand, one uses more than one round-trip time's worth of buffering, since fiber is very reliable, the excess buffering is almost never used and is wasteful to allocate.

[13] Note that this calculation is done without considering the effects of congestion, because congestion increases the round-trip time delay, but doesn't increase the amount of data that should be outstanding. Given a network capable of moving n bits per second, sending $n+1$ bits in a second doesn't increase the delivery rate — it just causes the extra bit to be delayed in a queue somewhere. You are better off just sending n bits. In other words, when a network is congested, you should not try to put more data into it.

4.2. Flow Control

Gigabit networks have at least two important flow control problems. Both are caused by delay. Neither circuit nor datagram architectures completely solve either problem.

Because the network transmission delay cannot be faster than the speed of light in fiber, gigabit networks will pack a relatively large number of bits into a length of cable. This high concentration of bits has at least two important consequences. First, hosts that want to fill a gigabit link will need mechanisms capable of managing a connection where megabytes of data will be in transit at any given time. A second consequence, which is in some sense an inverse of the first, is that establishing a connection between two hosts becomes more expensive, in terms of data not sent. In the time that a host spends waiting for a call request or connection open to go through, several megabytes of data could have been sent. In other words, an opportunity cost is associated with connection set-up.

The first problem, managing large amounts of data in flight, appears to be the easier of the two to solve.

In the virtual circuit world, the problem is trivial. A virtual circuit with sufficient capacity is created and the data is sent through it. Enough buffering is provided in the circuit to support the maximum flow.

Datagram architectures have somewhat greater difficulty supporting large transfers. All the basic datagram control mechanisms are believed to work a gigabit speeds. The slow start algorithm can still be used to probe the bandwidth of the path, and discarding data when the network is severely congested will not have catastrophic effects. But datagram architectures have historically not provided guaranteed bandwidth, and on gigabit networks there may be applications (like real-time video) in which guaranteed bandwidth is necessary.

There have been several suggestions for how guaranteed bandwidth could be incorporated into a datagram architecture. Some of the more interesting ideas involve having routers infer the existence of guaranteed bandwidth flows from their traffic patterns or specially tagging datagrams affiliated with a guaranteed bandwidth flows. Both schemes would appear preserve the basic characteristics of datagram networks (especially recoverable state).

The problem of the opportunity cost of a connection set-up looks less tractable. This problem can be broken up into three closely related problems:

(1) How long before the first bit comes back from the remote system? If we send a request to a remote system, how long do we wait to get the start of the reply? This question is critical for distributed systems, where the delay time for network transactions can have a large impact on system performance. Unfortunately, the reply cannot come back faster than the speed of light delay. This has critical consequences for distributed systems, which will soon find that modulo small changes in host processing times, getting the first bit back in a network transaction will never get any faster.

(2) How much can we send on the first round-trip? One possible way to cope with the long delay waiting for the first bit is to send more data back in response to the original request. Again, unfortunately, this is difficult. Until at least one round-trip is made, the capacity of the network cannot be learned, and thus sending more than a trivial amount of data (a few kilobytes) risks causing mild congestion.[14]

---

[14] Note that one connection sending lots of data won't affect congestion or round-trip times very much, but several starting up at once could have catastrophic effects – particularly if there are delay-sensitive applications, like voice, using the network. So, to be conservative, each connection has to start slowly.

(3) How long before we can send at full network capacity? If we can't get more than a small amount of data back in one round-trip time, how long before we can send at whatever bandwidth the network is willing to give us. For virtual circuits, we have to wait until the virtual circuit is established, usually one-round trip time.[15] For datagram architectures, the approved method is to probe the network using exponentially larger amounts of data on each round-trip time, until the network capacity is reached [3]. This takes several round-trip times but starting with one 8-kbyte packet, the elapsed time before one gigabit is reached is well under 2 seconds. That is less time than it takes to make a phone call, which makes it likely that users would not find the delay upsetting. However, a 2-second delay is too long for some applications, and it may be that the datagram network needs to be able to provide more information about its current capacity so that applications can send at high speed after only one or two round-trips.[16]

5. Conclusions

This paper has focused narrowly on the problem of achieving transfer speeds of one gigabit per second and has suggested that speeds of one gigabit per second are within reach of both datagram and virtual circuit architectures. It is perhaps easier to achieve this speed using virtual circuits (the reader may note that much of the flow control section was concerned with scaling datagram architectures to gigabit speeds), but either architecture is capable working at one gigabit, and thus the choice of datagram or virtual circuit architecture for a gigabit network is one of preference, not technological necessity.

The next question is whether these arguments still apply at higher speeds, say 3 gigabits or 10 gigabits per second. Good arguments can be made both pro and con. Simply scaling some of the numbers to 10 gigabits gives a scary set of requirements: a 600-MIPS CPU, switching times of a few hundred nanoseconds, and 150 megabytes of buffer space. Those requirements are sufficiently stiff to make one question the network architectures that force them on us. But doomsayers have a historically poor record in computer science. Five years ago, the shift from silicon to gallium arsenide was widely predicted. It hasn't happened yet. Time was that networking experts generally felt that TCP was a processing-intensive transport protocol. Van Jacobson has shown how to process a TCP segment in under 20 instructions. More generally, processors continue to get faster (last year SUN was reportedly predicting a 300-MIPS CPU in the early 1990s); memory continues to get bigger and cheaper, and parallel processing still waits to be fully exploited.

All this leads me to be optimistic and suggest that the architectures will scale up to higher speeds. Having concluded that one gigabit is not that fast, I find it hard to accept that a mere factor of ten change will make a big difference.

Acknowledgements

Several people have helped me shape my thinking in this paper. Special thanks to Debbie Deutsch, Ira Richer, John Robinson, and members of the Internet End-To-End Services Task Force, which is chaired by Bob Braden, and the Internet Architecture Task Force, which is chaired by Dave Mills. Of course, I take full responsibility for any errors or misguided opinions.

---

[15] The exception comes when the network is being heavily used, in which case the level of service requested may not be available and the equivalent of a busy signal will result.

[16] Various mechanisms like those proposed for flow indicators could be used. Again, the basic datagram model doesn't break.

References

[1] Anthony S. Acampora and Mark J. Karol, "An Overview of Lightwave Packet Networks," *IEEE Network*, Vol. 3, No. 1, January 1989, pp. 29-41.

[2] David D. Clark, Van Jacobson, John Romkey, and Howard Salwen, "An Analysis of TCP Processing Overhead," *IEEE Communications*, Vol. 27, No. 6, July 1989, pp. 23-29. Most of the computation in this report was done with the earlier version of the paper, published in the Proc. 13th LAN Conference (Minneapolis, 1988).

[3] Van Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM '88*, Stanford, Calif., August 1988, pp. 314-329.

[4] R. Jain, "On Caching Out-of-Order Packets in Window Flow Controlled Networks," Tech-Report 342, Digital Equipment Corporation, January 1985.

[5] J. Nagle, "Congestion Control in IP/TCP Networks," *ACM Computer Communication Review*, Vol. 14, No. 4, October 1984, pp. 11-17.

How Slow is One Gigabit Per Second?

- ''Gigabits will require a new architecture....''
- This talk argues that statement isn't true.
- Given conservative assumptions, *one* gigabit clearly feasible, multiple gigabits probably so, with current architectures.
- Talk tries to stay agnostic about virtual circuit vs. datagram architectures.

Overview

- Conservative assumptions
- Bandwidth and processing limits
- Handwaving on flow and congestion control
- How well do these arguments scale?

Conservative Assumptions: Processing

- Single workstation class CPU of 60+ MIPS
- 64-bit data paths
- 60 MIPS chips due out this fall; could have multiprocessors; 64-bit paths the next logical size

Conservative Assumptions: Packet Sizes

- Future traffic mix will have same or larger average packet size than today
- Future MTUs > Today → Bigger average packet sizes
- More bulk traffic → Bigger average packet sizes

Bandwidth and Processing

- Three elements:
- Links: we know fiber can go this fast. (Note fiber-memory interface may be tricky).
- Routers/Switches: can they switch fast enough?
- Hosts: can they send and receive fast enough?

Routers: Configuration

- A component of any architecture (something to route between links)
- Lots of ways to build them
- Using simple model here: single CPU routing among multiple (~3) links

Routers: Packets per Second

- Key cost is the packet, independent of size (barring hop-to-hop error correction).
- Today's market expects 6,000 to 10,000 pkts/sec between 2-3 Ethernets
- Extrapolate to gigabit: 600,000 to 1,000,000 pkts/sec

Routers: Time per Packet

- 600,000 to 1,000,000 pkts/sec → 1 to 1.6 microseconds per packet
- Or, at 60+ MIPS, 60 to 100 instructions....
- Is that enough?

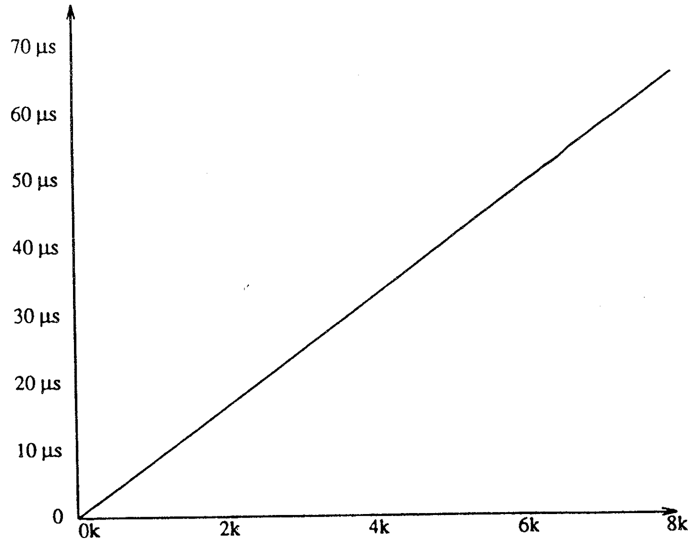


Figure 2: Packet Size (in bytes) vs. Time to Process (in µ-sec)

Routers: Is 60-100 Instructions Enough?

- Yes for circuits, if they last long enough.
- At least one implementation can switch in under 30 instructions after circuit set up. Leaves plenty of instructions for set-up amortization.
- Close for IP.
- Basic IP takes ˜120 instructions + device drivers. We'll need better designed device drivers....
- Additional data point: LANBridge 100 takes 50 instructions (with lots of hardware assist).

Hosts: Traffic Requirements

- Hosts ≠ Routers
- Hosts don't use all the bandwidth, unless packets very large.
- Hosts need time to process data.
- Packet sizes vary widely. Make packet size free variable.

Hosts: Instructions per Packet

- Clark, Jacobson, Romkey and Salwen numbers for TCP/IP, exclusive of checksum and no header-prediction.
- ~1,000 instructions before application sees data.
- Application needs to touch data. Constrained by memory to processor bandwidth. At minimum that's length of packet divided by machine word size.
- Note that file transfer and video frame buffers don't touch data, just move it.

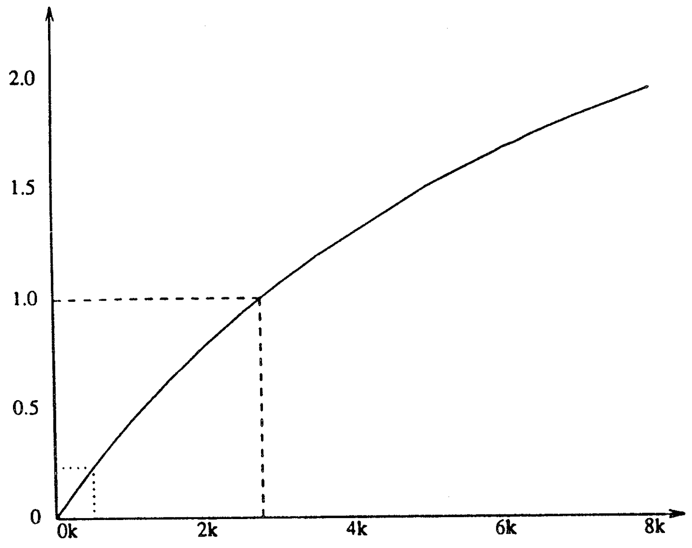


Figure 1: Packet Size (in bytes) vs. Bandwidth (in 10^9 bits)

Hosts: Reading the Graph

- A 512-byte packet size → 0.25 gigabit throughput
- A 3-Kbyte packet size (< FDDI MTU) → 1 gigabit throughput

Handwaving on Flow and Congestion Control

- Argued we can process the bits as they go by.
- But can we control them???
- Delay × bandwidth is huge
- Host-to-Host flow control
- Host-Router or Host-Network interactions

Delay × Bandwidth

- Cross country delay in fiber is 0.047 seconds
- Phone company guarantees only 0.120 seconds.
- Multiply by a gigabit and there's 5.9 to 15 megabytes in flight

Host-to-Host: Buffering

- Can we buffer 5.9 to 15 megabytes? May need to by end-to-end argument.
- Not outrageous amount to buffer. At worst, one could push some stuff to disk....
- Switching delay has no effect on this analysis – requires thousands of switching nodes in path to make an impact.
- Loss rates may screw us: bit-error-rate (BER) of 10^{-8} gives 1/3 chance a packet is lost in a given RTT.

Host-to-Host: Flow Control

- Need to determine size of pipe.
- Some architectures (both VC and datagram) can tell you. Others ask you to guess.
- Basic idea is Jacobson/J-R-C's. Exponentially expand pipe to knee (or cliff), fall back and probe again.
- Exponential growth and 8-kbyte packets means 0.7 to 1.7 seconds to reach a gigabit. Fast enough for user bulk transfers. What about programs?
- Note: connections only hunt if they are trying to maximize bandwidth or want more bandwidth than is available.

Router-Host: Feedback

- Dropping packets good enough? Unclear.
- Feedback schemes like J-R-C (1-bit scheme). Some indications it isn't as stable as one would like.
- Jacobson has mused about packet inter-arrival time estimator. Can our clocks be fine tuned enough?
- Issue of guaranteed service. Type-of-Service or flow indicators?

How Well Does This Scale?

- Doomsayers have not done well in computer science....
- There's some slack in these numbers. Faster memory, more processors, could help out...
- But scaling some numbers gets worrysome...

Network Performance Impact of the X Window System Protocol

Presented by Ralph Droms

The X Window System (X) is a machine-independent window-based graphics system that supports distributed computation. X applications generate graphics requests that are transmitted to an X server through a well-defined interface. The interface is designed to be used through a reliable byte-stream interprocess communication channel, and is therefore appropriate for use across computer networks that support such a communication abstraction.

The X Window System Protocol supports bidirectional communication between an application program and an X server. The application program sends graphics, window management and other requests to the server, and the server sends responses, notification of window-related and input events, and error notifications to the application program. The protocol requests are at a low level of detail; for example, draw a line from point (x1, y1) to point (x2, y2) or notification that a specific keystroke was just detected.

X applications may increase the communication load presented to a computing network because of:

- Overhead due to X protocol (e.g., each keystroke is 32 bytes)
- Overhead due to communication protocol packetization (e.g., each TCP segment has 54 bytes of Ethernet+IP+TCP header)
- Load on application host due to communication overhead
- Load on application host due to multiple concurrent user processes
- Load on gateways due to packet traffic

Network traffic due to X applications can be measured by writing X protocol transmissions to a log file and post-processing the log file for analysis. The protocol transmissions may be intercepted through modificiations to Xlib and the X server, by a monitor process situated between the application and the server, or by a LAN monitor.

This presentation reports on some preliminary measurements taken by a monitor process. Through several experiments, we measured the total load presented to a LAN by individual X applications, and a composite set of applications representing a normal user session. We found that on average, the load resulting from X applications was not significant; however, some applications may have significant traffic in short bursts.

NETWORK PERFORMANCE IMPACT OF THE X WINDOW SYSTEM PROTOCOL

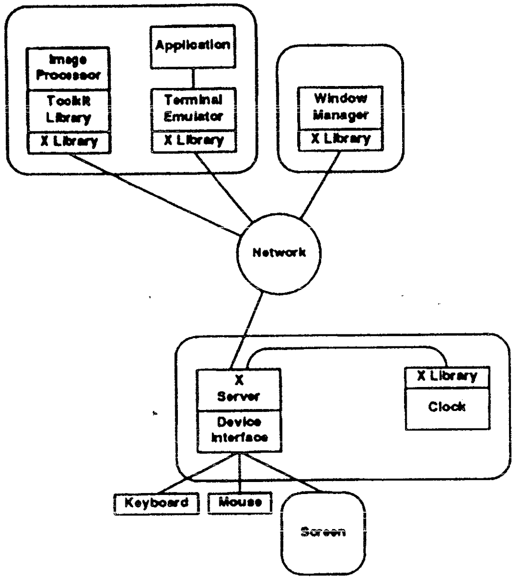
Ralph Droms
NRI
(Bucknell University)
Reston, VA
rdroms@nri.reston.va.us

The X Window System (or "X") is a *device-independent, network transparent window-based graphics system.*

- Device Independent:
  - Display hardware
  - Operating system
- Network transparent:
  - X application can run on a remote host
  - Can use many network protocols
- Window-based graphics:
  - Provides low-level graphics primitives
  - Supports overlapping window paradigm

Architecture

- Basic model: *display servers* and *clients*
- Display servers do low-level user interface
  - Graphic output
  - Windowing
  - Input management
  - Multiple simultaneous clients
- Clients implement specific applications
  - Multiple windows
  - Multiple servers
  - Device dependencies hidden by servers



Client-server Communication

- Interface between display servers and clients is a well-defined, byte-stream protocol
- Can use TCP/IP or Unix† streams

† Unix is a Trademark of AT&T Bell Laboratories.

Protocol Functions

- Graphics functions:
  - Draw line, draw shape
  - Set pixel, fill area
- Window functions:
  - Create, delete
  - Raise, lower
  - Resize, iconify, deiconify
- Input events:
  - Keystroke, mouse clicks
  - Mouse movements
  - Window edge traversal
- Replies to client requests

Possible Network Performance Impacts

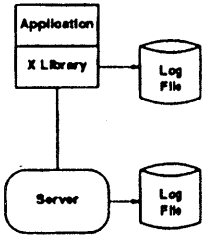
- Roundtrip Delay
  - Mouse movement
  - Keyboard input
- Transmission Delay
  - Image display

Measurements

- Roundtrip time for keystrokes
- Elapsed times for screen activity
- Throughput

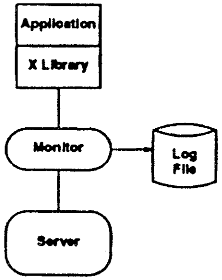
Approaches

- Log protocol activity
- Modify server and client



Approaches (continued)

- Interpose monitor agent



Implementation

- Modified server and client
  - Precise timing measurements
  - Requires access to source and programming effort
  - Requires recompilation of all measured clients
- Monitor process
  - Can monitor any client and server
  - Minimal programming effort
  - No timing information

Results

- Built two implementations
- Modified library and server
  - Xlib and Sun server
  - Logs bytes read and written to (separate) log files
  - Post-processing analyzers
  - `awk` and `grap` to display data

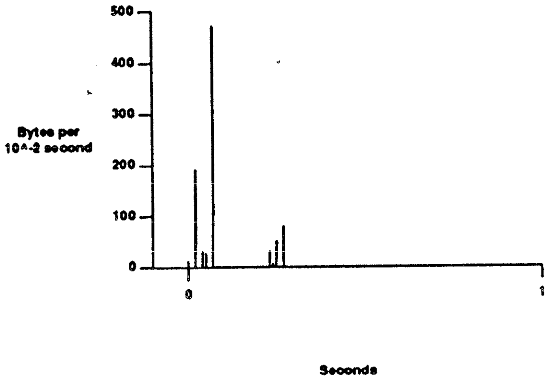
Results (continued)

- Monitor process
  - Listens for client connection to *log-host:1*
  - Makes connection to *server:0*
  - Logs protocol exchanges in both directions
  - Post processing analyzer
  - `awk` and `grap` to display data
- Gathered and analyzed round-trip and throughput data: Sun 4/280 clients and NCD-16 server

---

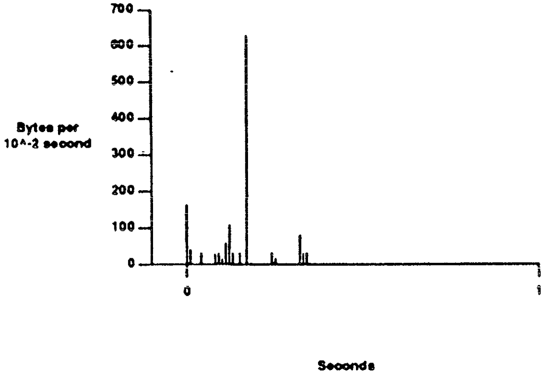
`Hello, World`

- "Simple" example program by David Rosenthal
- Opens window, displays string "`Hello, world`"



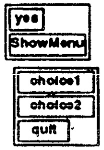
`Hello, World`
(XToolkit Version)

- Rewrite of previous example to use XToolkit facilities
- Same execution as previous



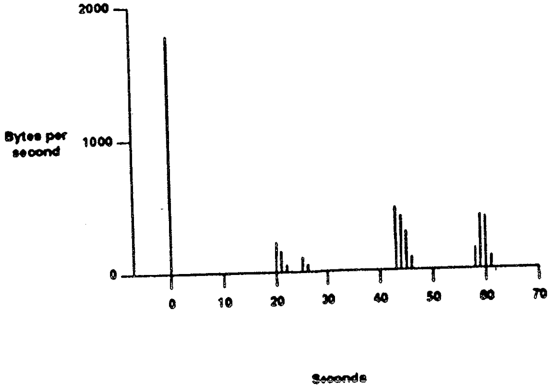
`xtpopup`

- Popup window test program
- Test of roundtrip delays



- Experiment:
  Move into window, click `yes` box
  Move between `ShowMenu` and `yes` box
  Pull down `ShowMenu`, choose `choice1`
  Pull down `ShowMenu`, choose `quit`

`xtpopup`
(results)

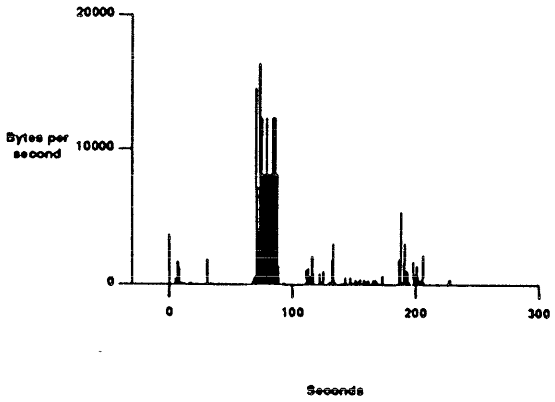


`xterm`

- Terminal emulator
  VT100, Tek emulations
  Many options (via menus)

- Experiment:
  Open window
  Type and correct line
  Type and list directory
  Cat large file to window
  Pull down menu to bring up and switch
  to Tek window
  Plot file
  Bring back vt100 mode
  Select vt100 text

`xterm`
(results)

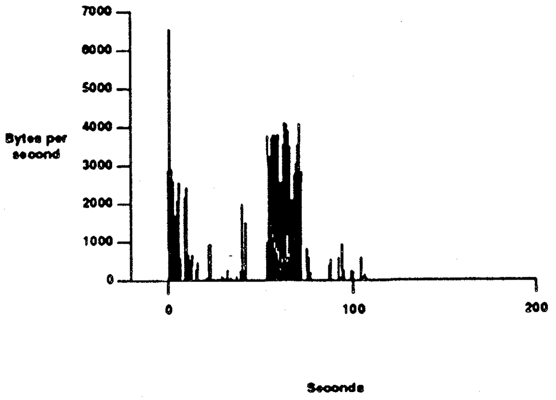


- 228 seconds, 241,480 bytes
- Maximum: 16384 bps, average: 1059 bps

`twm`

- "Tom's Window Manager" by Tom LaStrange
- Many services
  Window placement
  Iconification/deiconification
  Icon management
  Pulldown menu interface
- Experiment:
  Start `twm`
  Open `xterm` window
  Move `xterm` window
  Iconify
  Deiconify
  Close `xterm` window

`twm`
(results)

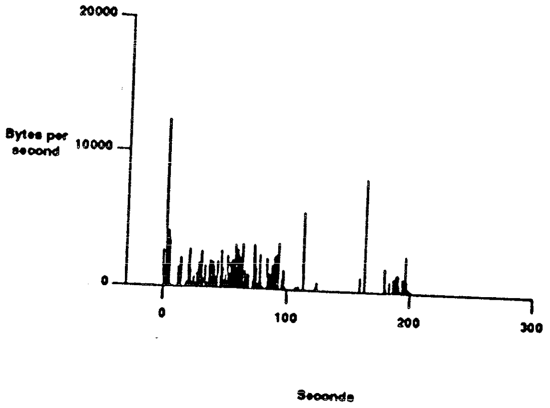


- 107 seconds, 108,848 bytes
- Maximum = 6544 bps, average = 1017 bps

`xfig`

- "Facility for Interactive Generation of Figures" by Supoj Sutanthavibul
- Object drawing application (MacDraw-ish)
- Uses *many* windows and cursor tracking
- Experiment:
  Draw line
  Draw curve
  Draw circle
  Add grid
  Iconify
  Deiconify

xfig
(results)

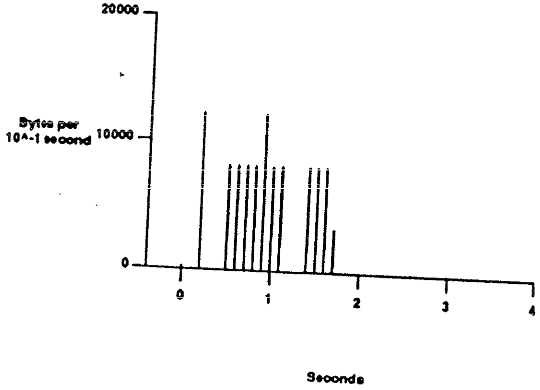


- 202 seconds, 169,712 bytes
- Maximum = 12,336 bps, average = 840 bps

---

xphoon

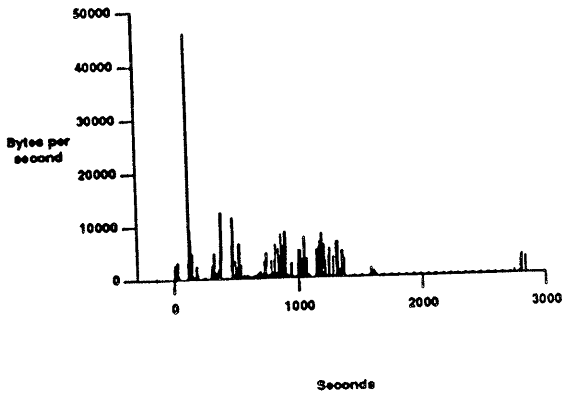
- "X Phase of the Moon" by Jef Poskanzer and Craig Leres
- Sets screen background to current phase of moon
- Ships moon image from client to server



Login Session

- Start protocol monitor
- Initiate session with `xdm`
- `xphoon, emacs, twm, xterm...`

Login Session
(results)



- 2873 seconds, 940,364 bytes
- Maximum = 45,920 bps, ave. = 2873 bps

Conclusions

- X brings graphics to the terminal
- Graphics commands replace ASCII text across network
- Typically not a significant load: certain applications may represent high transient loads

Gateway Congestion Control Policies
Presented by Allison Mankin

The paper ''Gateway Congestion Control Policies" was submitted as an INTERNET-DRAFT (DRAFT-IETF-PERFCC-GWCC- 00.TXT.1) shortly before the Stanford meeting. It was extensively revised and expanded from the draft discussed in Florida. It will receive small further revisions, based on Stanford, but will not change in scope or conclusions.

The purpose of the talk was to summarize to the IETF plenary the contents of the paper and the progress of the working group on gateway congestion control. First a bit of background. Our charter was to review Internet congestion control, and develop recommendations for the short and mid-term Internet. The initial stimulus was the success of Slow-start TCP; improvements in congestion control were suggested, and widely fielded! Also entering into the charter was the realization (due to Ramakrishnan and Jain, as well as Jacobson) that Internet gateway and host performance were coupled. We split our charter into two papers, one on TCP performance which is not yet complete, and this one on gateway congestion control.

The paper represents the outcome of many productive discussions by the working group. One highlight was our session in January with gateway implementors from BBN, IBM, Cisco, Proteon, and Network Systems. Over the working group lifetime, numerous members were energetic and active. To the extent that we've come up with a useful, accurate picture of gateway congestion control, it is due to the high quality and skepticism of the working group.

Our meetings also stimulated a number of studies: in particular, the only published results on Random Drop (so far), those of Zhang and Hashem, were from simulations done in response to the working group. Jacobson reminded the speaker to point out that these results cover only the most simplified versions of Random Drop, using simple queue thresholds and a fixed probability of drop. Although the problems identified with Random Drop Congestion Avoidance are convincing, the IETF should expect that the techniques of Random Drop will appear in future results from Van.

The paper and talk surveyed the characteristics of a number of policies, Source Quench, Random Drop, Congestion Indication, Selective Feedback Congestion Indication, and Fair Queueing. The vugraphs summarizing each of these follow this report.

The conclusions of the paper may be summed up in six points:

1. Congestion is important; we can't ignore it or buy out of it.
2. There are multiple metrics for performance goals, sometimes conflicting. Combined metrics such as power are of value, but then finding an appropriate interval of measurement is critical. We suggest several approaches.
3. Congestion recovery is a must, otherwise the Internet will be subject to congestion collapse.
4. Congestion avoidance is important. This is the effort to control end systems so that congestion can only occur as a transient phenomenon. The Internet has never had congestion avoidance, and it hard to get right.
5. Internet congestion recovery is simple to do, as demonstrated by Jain and Jacobson and by widespread practice.
6. Random Drop for congestion recovery is a 'win.' On queue overflow, drop a packet that is chosen at random from the queue (and current arrivals).
7. There are many reasons to field Fair Queueing in gateways, assuming we solve implementation problems. Promising solutions to the overhead and scaling problems are under development; one of these is Stochastic Fairness Queueing.

Key problems in Internet congestion control are still in need of research. These include demonstrating ANY gateway policy for a general (multi-gateway) topology.

There were a few questions from the audience:

> Why don't we have a recommendation either to kill Source Quench or to use it in a particular way?

We ended up viewing Source Quench as a neutral mechanism, rather than a congestion control policy. It was difficult to argue at this time that we should kill what amounted to a hook. We recommended that a host view Source Quench as advising it of unspecified (ambiguous) performance problems.

> Why didn't the working group come up with some traffic types (such as TCP segments with control flags) to be given priority during congestion?

The question went on to state that our choice not to look above the Internet layer in gateways is arbitrary. However, if you consider that the protocol above IP even now may be another network layer protocol (an encapsulation), this choice is even less arbitrary.

The Gateway Congestion Control internet-draft is available on-line at SRI-NIC.ARPA and can be accessed by anonymous ftp. The directory name is "internet-drafts:" and the file name is "draft-ietf-perfcc- gwcc-00.txt". For more information, please contact:

- ietf-request@venera.isi.edu.

GATEWAY CONGESTION CONTROL

(PERFORMANCE AND CONGESTION CONTROL WORKING GROUP WRAP-UP)

A. MANKIN

PERFORMANCE AND CONGESTION CONTROL W.G.

Charter to review methods of congestion control in Internet

W.G. has generated productive discussion and outside studies

After a year of work, we have document on Gateway Congestion Control Policies

W.G. INTERNET-DRAFT

Title: Gateway Congestion Control Policies

Purpose

- Focus discussion on some experimental solutions to Internet gateway congestion
- Survey gateway congestion control policies
- Stimulate further work
- Make modest recommendations

NOT COVERED IN DRAFT

- Routing-based methods (load-balancing)
- Economic solutions (charging)
- Connection-oriented gateway solutions

COVERED

Gateway Policies

- Source Quench (that is, current Internet)
- Random Drop Congestion Avoidance
- Random Drop Congestion Recovery
- Congestion Indication (DEC Bit Scheme)
- Selective Feedback Congestion Indication
- Fair Queueing (Shenker)

Fielded end-system congestion control schemes

- Slow-start
- End-system DEC Bit
- As these interact with gateway policies

CONGESTION AVOIDANCE AND RECOVERY

Congestion --> Demand greater than some resource

Congestion Avoidance --> Control demand so congestion is prevented

Congestion Recovery --> Restore operation after congestion starts

Internet has never had CA

In general terms, CR still needed along with CA

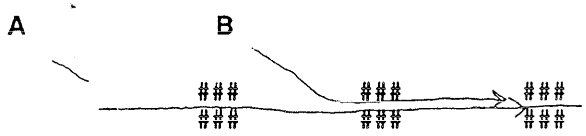
- Sudden changes in demand or resource availability

FAIRNESS

Allocation of gateway resources to users

- Not relevant if enough resources to meet every user's demand completely
- If some resources unused, give to the users with higher demand
- If no extra, give all users equal resources
- Variation -- assume unequal, set allocations

Resource Fairness



User A, User B equal

USER DEFINITIONS

One TCP connection, stream or flow

Source-destination pair is the virtual user

in SCFI, FQ

``Cooperating'' User

- One that follows effective method of adjusting demand in response to congestion

SOURCE QUENCH

Ambiguous message

Hard to distribute fairly

Uses network resources, so needs to be sent

on thresholded basis

Ongoing work by G. Finn (ISI)

- Tune SQ send, rate-based IP response to yield an effective congestion avoidance scheme

MEASUREMENT AND CONTROL INTERVAL

Design and evaluation of congestion control

requires determination of meaningful intervals

- Too small --> Transient differences are given too much importance

- Too large --> Varying population of users, for example, changes desired allocations

Draft discusses intervals based on idea of

dominant Round Trip Time

- Queue regeneration cycle, adaptively tracking RTT of dominant traffic (Congestion Indication)

- Frequency analysis to identify important RTTs (Random Drop concept)

RANDOM DROP

A concept (due to Jacobson)

A packet randomly selected from arrivals belongs to a particular user with a probability proportional to the user's demand

This selected packet is dropped for congestion recovery or congestion avoidance

Expresses good goals for fast, large population gateways:

- Scaling (independent of number of users, packets in gateway)
- Minimal computation

Idea is to choose a number A of packet arrivals, draw a random number J uniformly from 1 to A, and drop the Jth packet to arrive.

RANDOM DROP FOR CONGESTION AVOIDANCE

Hypothetical Policy

Parameters

- Control interval
- Measurement of operating point
- Probability of drop
  Function of estimated number of users
  direction of change of congestion
  detected, possibly duration of congestion

Simplified Policy

- Agreed that Random Drop alleviates stable unequal allocations to equal demand Slow-start users

RANDOM DROP CONGESTION AVOIDANCE CONS

Users with differing control get unequal bandwidth

Cooperating users with multiple-gateway paths lose bandwidth

Doesn't control delay

Sensitive to tuning

These problems identified by Shenker, Hashem, Zhang

RANDOM DROP FOR CONGESTION RECOVERY

Use Random Drop principle on overflowing queues

Simple in concept

Win (like segregation improvement) is for users which aren't tuned to fill the queue

CONGESTION INDICATION (DEC BIT)

Use a binary feedback (filtered by user) of whether congestion encountered at each gateway

Intent to optimize power

Source: Jain, Ramakrishnan, Chiu
DEC TR-506, also SIGCOMM '88

SELECTIVE FEEDBACK CONGESTION INDICATION

Concerns about Congestion Indication that small number of users causing congestion cause all users to be controlled

Selective Feedback CI enhancement with goal of separately controlling users with differing demands

Compute send rate of source-destination pairs and allocation based on total capacity

- Not applied if no congestion
- Leftover bandwidth is given to higher demand users
- Fair across users with very different sets of resources

Source: Ramakrishnan, Jain, Chiu
DEC TR-510

FAIR QUEUEING

No need to describe here, given Shenker's presentation Tuesday.

Sources: Nagle, On Packet Switches with Infinite Queueing, RFC-970

Demers, Keshav, Shenker, Analyis and Simulation of a Fair Queueing Algorithm, SIGCOMM '89

FAIR QUEUEING CONCERNS

When CPU is scarce, Fair Queueing will turn into First Come First Serve (input queues)

- Need to implement efficiently

Scaling concerns also need to be addressed in good implementations

Management, authentication of allocations poses challenges

Given allocations, treat unfairness to particular users within user class by end-system methods?

FAIR QUEUEING RECOMMENDATIONS

Power optimization for cooperating end-system pairs is possible, with much less dependency on other gateway load than other policies

Doesn't impose homogeneity on end-systems

Good for policy experiments

Try to develop needed implementation efficiencies

OPEN QUESTIONS

Gateway congestion control that will scale and not push CPU restrictions during congestion

Build in controls for congestion avoidance for Slow-start and End-system DEC Bit? (Flip question from cooperating users to cooperating gateways)

Stability of gateway congestion control in general topologies

Stability of heterogeneous gateway policies

CONCLUSIONS OF WORKING GROUP/PAPER

Congestion is important

- Can't ignore it, buy out of it

Multiple metrics for performance goals

- Several conflicting
- Some promise in combined metrics
- Appropriate interval of measurement of metrics

Avoidance Vs. Recovery

- Recovery is must, avoidance is important

Recovery is simple

- Demonstrated by Jain, Jacobson
- Practice widespread
- Recovery using Random Drop a 'win', often cheap to implement

Fairness is important issue

- Serious questions still open

Many reasons to pursue Fair Queueing algorithms

Paper identifies work that is of critical need

PERF & CC W.G. STATUS

Paper (Gateway Congestion Control Policies)

- INTERNET-DRAFT
- W.G. last revisions then IETF Review Process

Yesterday's wrap-up meeting featured presentations on:

- Fair Queueing simulations (Davin, Heybey, MIT)
- Smart Drop Congestion Control (Steenstrup, BBN)
- Stochastic Fair Queueing (McKenney, SRI)

IP Option for Crypto Summing

Presented by Jeff Schiller

The proposed IP Option RFC establishes a new IP option for storing a cryptographic checksum of an IP packet. It is intended as an "add-on" protocol to add authentication to existing IP protocols, especially those protocols designed to be used by routers for the exchange of control and routing information.

Its major advantage is that for protocols such as EGP it can be added without having to modify the EGP specification or procedures. However, as an add-on protocol it cannot provide complete or perfect authentication of datagrams. In other words it is not proposed to be *the* standard way of providing authentication for the IP protocol suite. Specifically, no attempt is made to conceal data from disclosure to third parties and no protection is provided against the replay of previous valid datagrams.

The IP Authentication Option draft is available on-line at SRI-NIC.ARPA and can be accessed by anonymous ftp. The directory name is "internet-drafts:" and the file name is "draft-ietf-auth-ipauthoption-00.txt". Comments should be sent to : jis@bitsy.mit.edu. For more information, please contact: ietf-request@venera.isi.edu.

To internet drafts:
IP Option RFC.
Kerberos V4 protocol.

Work in progress:
Authentication for SNMP
Status: draft being worked on
by authors.

The Terrestrial Wideband Network

Presented by Claudio Topolcic

In May 1989, the DARPA Terrestrial Wideband Network replaced the Wideband Satellite Network, which had been in operation for the previous 8 years. The Terrestrial Wideband Network uses one of the cross-country T1 trunks from the DARPA National Networking Testbed (NNT). The goal of the Terrestrial Wideband Network is to support research in high speed networking. Its goal is not necessarily to build the best 1.5 Mbps network, although it should be a good design. The following constraints affected the implementation. First, the most important user of this network, at least initially, is expected to be multimedia conferencing, so this application has driven the technical trade-off decisions, though not to the exclusion of other applications. Second, government fiscal realities required this network to be built quickly and at low cost re-using existing equipment.

The packet switching nodes of this network are located at unattended NNT Points of Presence (POPs) and are connected to Internet IP and ST Gateways at user sites by T1 tail circuits. These POPs are locations where DS-3 fiberoptic trunks are interconnected. These high bandwidth trunks are not available as a dense mesh across the country. Rather, they are generally installed along high utilization corridors. As a result, we are exploring link protocols that take advantage of a linear topology. The network will be built of a number of interconnected links.

The Terrestrial Wideband Network uses a type of Demand Queued Dual Bus (DQDB) protocol that we call the Dual Bus Protocol (DBP). This protocol is similar to the IEEE 802.6 Metropolitan Area Network (MAN) protocol, but with features that support wide area networking and multimedia conferencing. Whereas conventional packet store and forwarding would involve per packet forwarding processing and buffering at every intermediate node, a DQDB protocol performs processing and buffering only at the entry point and minimizes the processing and buffering at subsequent nodes along the trunk until the exit point.

The Dual Bus Protocol uses two equivalent buses, one in each direction. These buses are slotted, and each bus is used to request slots in the opposite bus. Counters that track free slots and slot requests provide datagrams FIFO access to the network. Stream traffic is supported by automatically reserved slots. The Dual Bus Protocol can also support re-use of packet slots, prior re-use of reserved slots, dynamic multicast groups and "byte-matching cut-through" for through traffic along the trunk. By byte-matching cut-through we mean that only one or possibly two bytes of a packets received from a trunk need be buffered by a site for comparison against the site's identifier. If it does not match, meaning that the packet doesn't terminate at the site, then the packet is propagated along the trunk without further buffering.

At this point we have implemented the basic Dual Bus Protocol with its enhancements for optimal use of the channel - re-use of slots, prior re-use, etc. The current implementation is tuned for the Butterfly's I/O architecture, which is optimized more for high throughput than for low delay. The Terrestrial Wideband Network currently aggregates small packets into larger slots and triple buffers all slots. This adds up to delay of 25 ms per packet and 15 ms per node traversed. The next generation I/O architecture, which will be fielded this coming winter, will implement the Dual Bus Protocol on an intelligent I/O device. This will eliminate the need for aggregation by allowing smaller slots and will decrease the forwarding delay by allowing double buffering combined with smaller slots. Hardware support for the Dual Bus Protocol could further allow byte-matching cut-through for through packets.

The Terrestrial Wideband Network includes a number of features to support operation in an unattended POP. It uses a passive fail-safe device that cuts a failed node out of the network to preserve bus continuity. It also uses remote dial-in access for a number of emergency functions that would otherwise have to be performed by local staff.

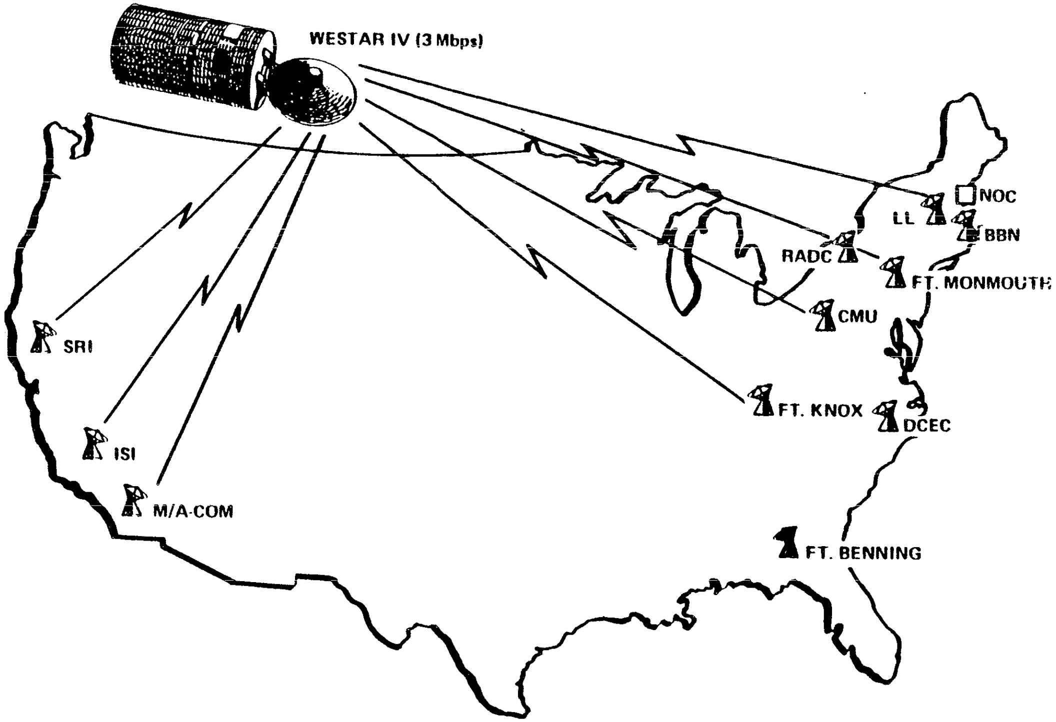
Preliminary performance measurements show that the Terrestrial Wideband network can support four-way video conferences. Of the 1.544 Mbps raw channel bandwidth, 1.236 Mbps are available for use by IP or ST gateways. Transit delay between neighbors is approximately 60 to 70 ms and end to end delay is approximately 150 ms.

The Dual Bus protocol and the next generation I/O architecture will provide the opportunity to make a number of enhancements to the Terrestrial Wideband Network. It will allow combining multiple T1 trunks to provide a single logical bus of higher bandwidth. Appropriate hardware will allow this architecture to expand to support DS-3 trunks at 45 Mbps. The Dual Bus Protocol can also provide an intelligent high speed link layer in the Internet IP or ST gateways. Finally, a number of independent intelligent links can be interconnected using packet store and forward nodes. Although a packet would require processing and would incur buffering delay at such a node, the advantages of the intelligent links would still be gained because a packet would only pass through a small number of such nodes as it traverses the network.

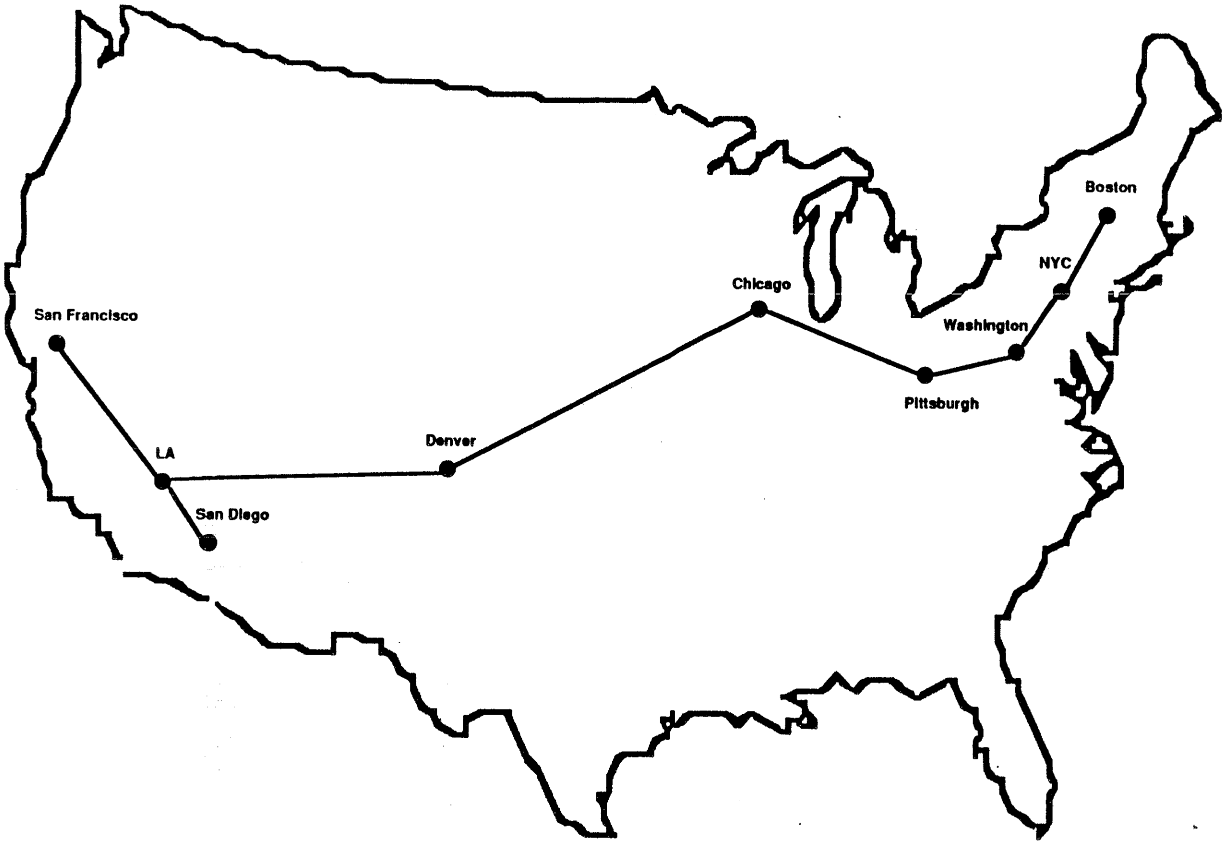
Terrestrial Wideband Network

Claudio Topolcic

BBN Systems and Technologies Corporation



Wideband Packet Satellite Network

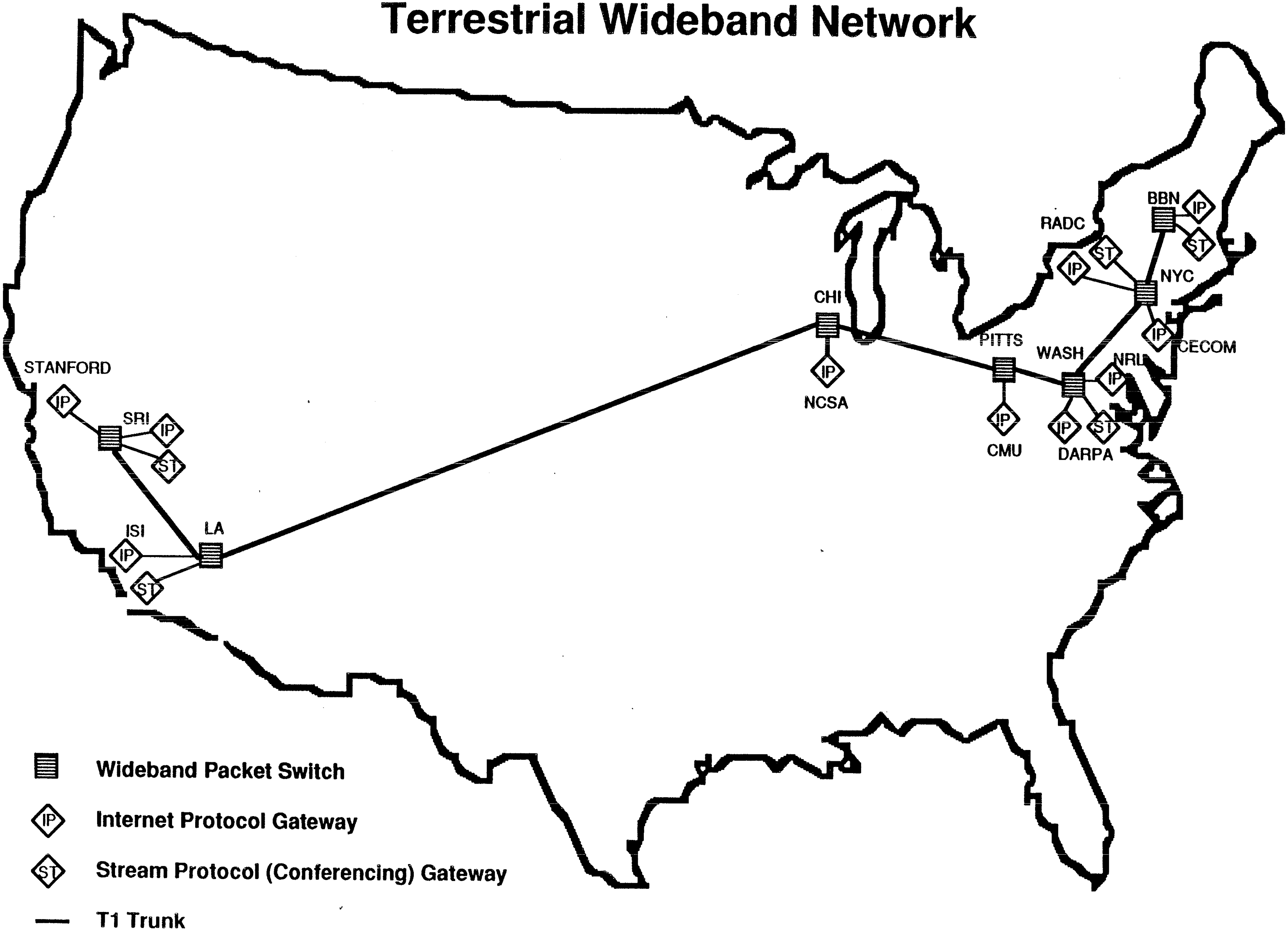


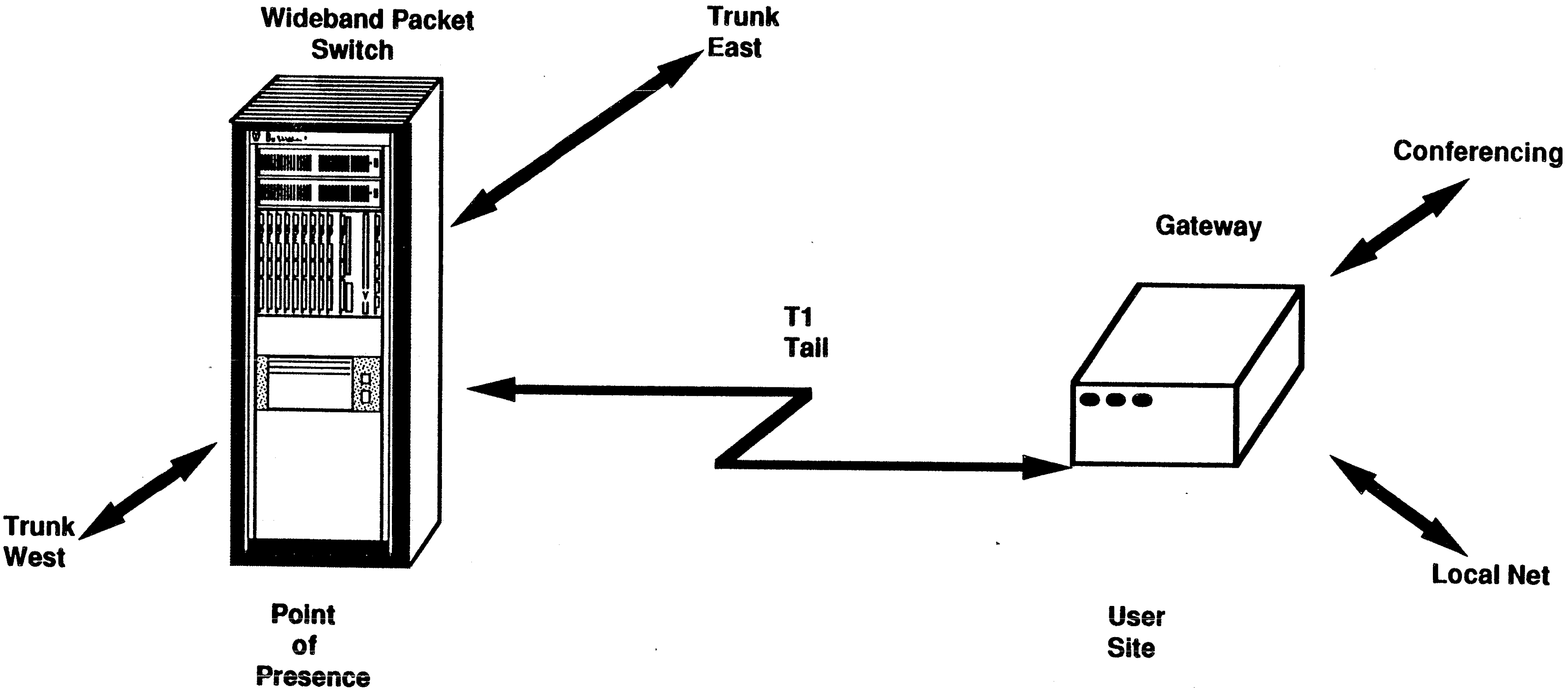
National Networking Testbed (NNT) Trunks

Goals

- The goal is to support research in networking at 45 Mbps and higher speeds
- Not necessarily to build the best 1.5 Mbps network
- Since multimedia conferencing is currently a big user, it will probably drive the trade-offs.
- Do it quickly at low cost re-using existing equipment

Terrestrial Wideband Network

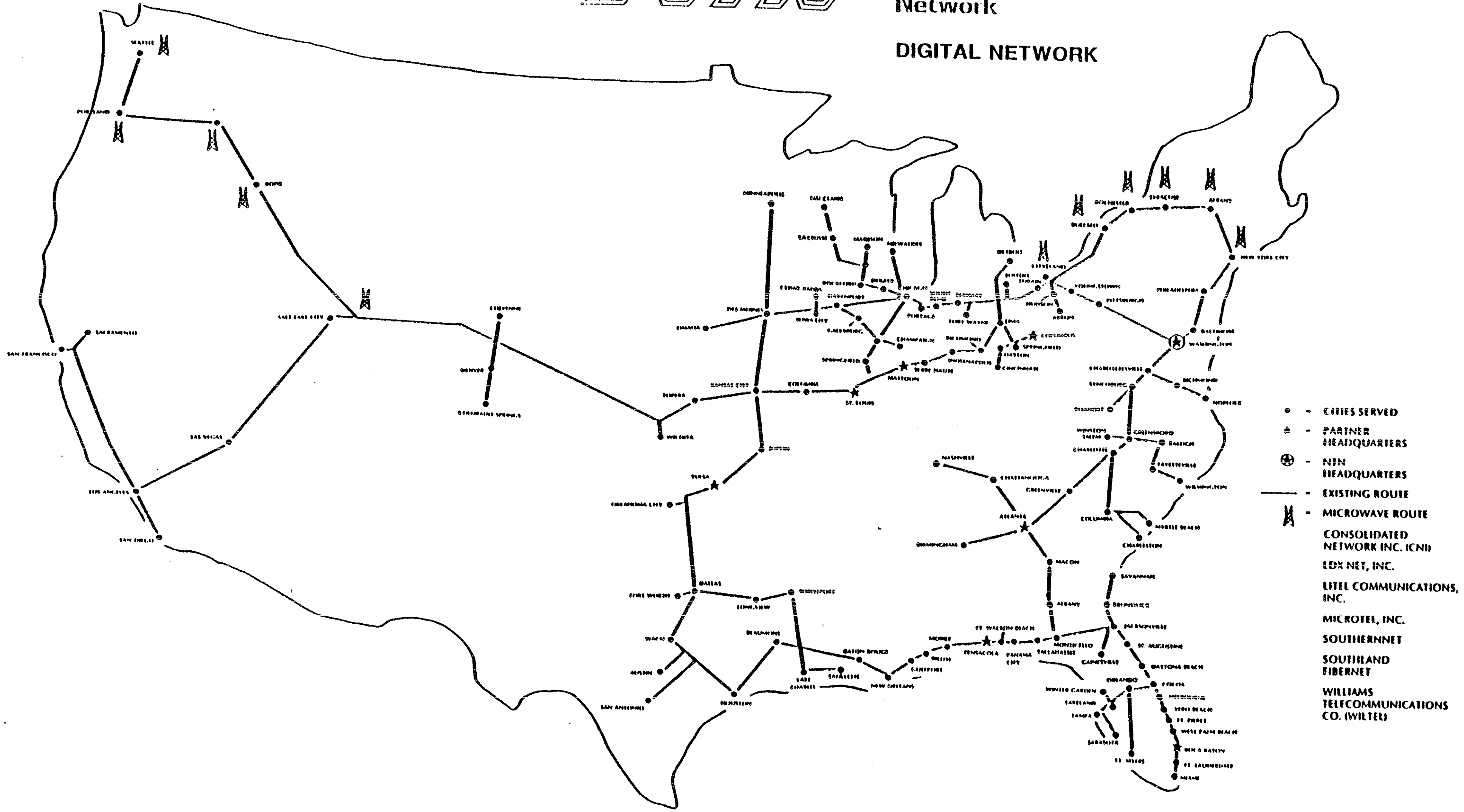




Wideband Node Configuration

National
Telecommunications
Network

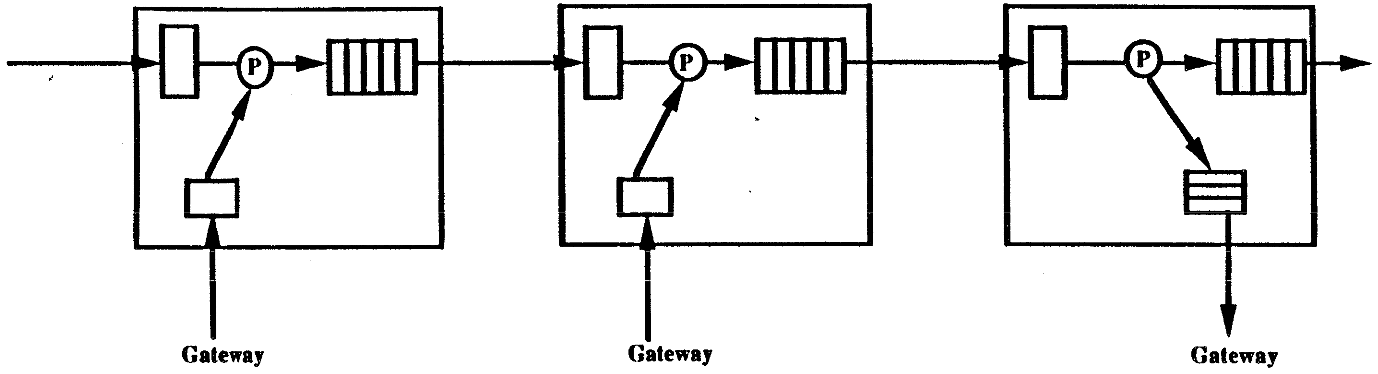
DIGITAL NETWORK



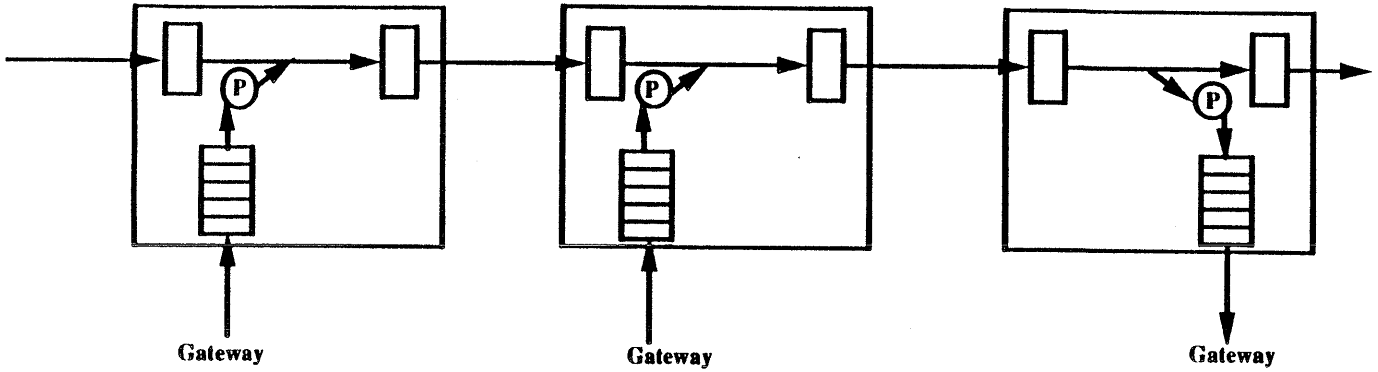
NOVEMBER 1987

The Dual Bus Protocol

- A type of Demand Queued Dual Bus (DQDB) protocol
- Similar to IEEE 802.6 MAN protocol
- Supports wide area networking, packet switching and multimedia conferencing



Conventional Packet Store and Forward

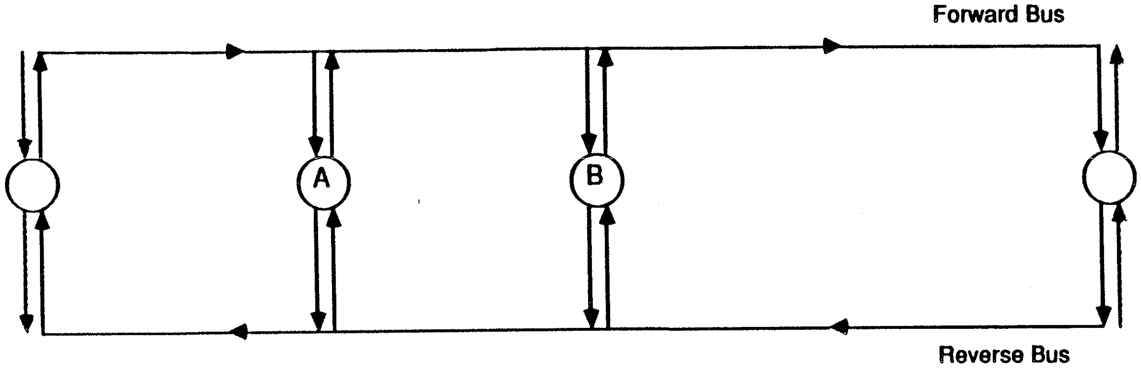


Dual Bus Protocol

Features of Dual Bus Protocol

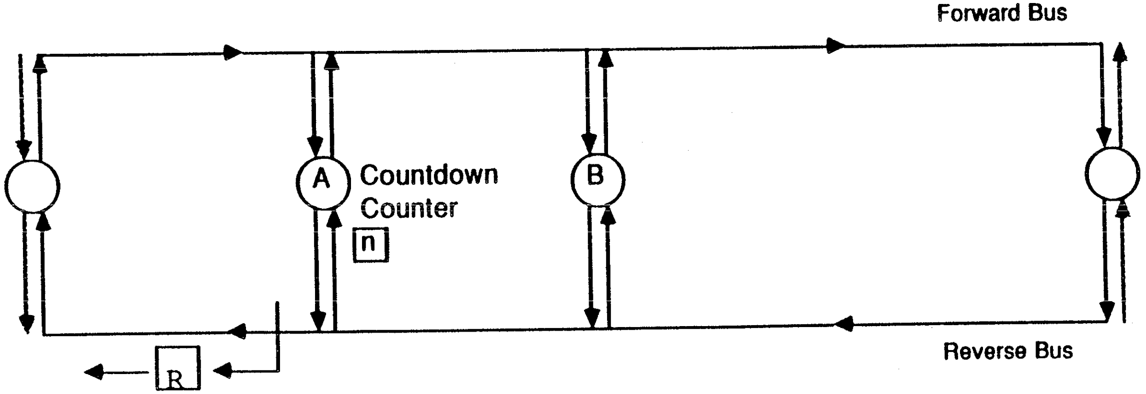
- Byte level cut-through
- Re-use of slots
- Reserved bandwidth
- Prior re-use of reserved slots
- Dynamic multicast groups

Basic Dual Bus Protocol Architecture



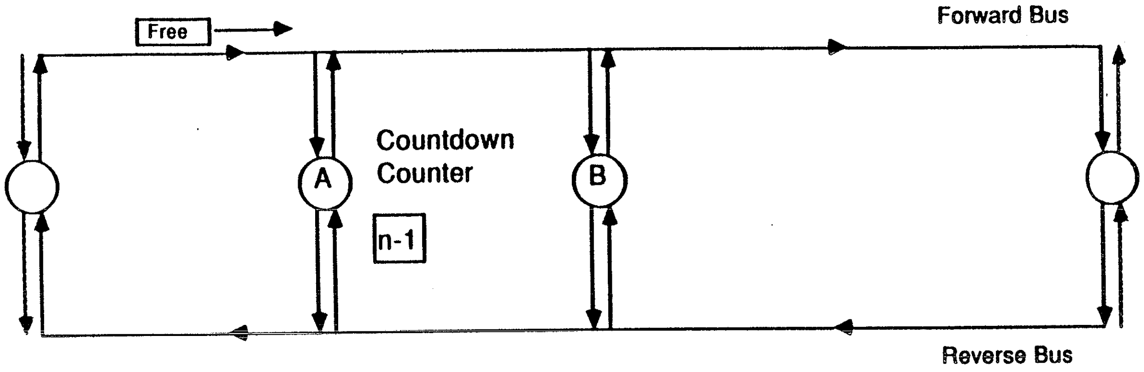
- Buses are slotted.
- The Forward and Reverse buses are equivalent.
- The Reverse Bus is used to request Free Slots on the Forward Bus.

Request for Free Slot



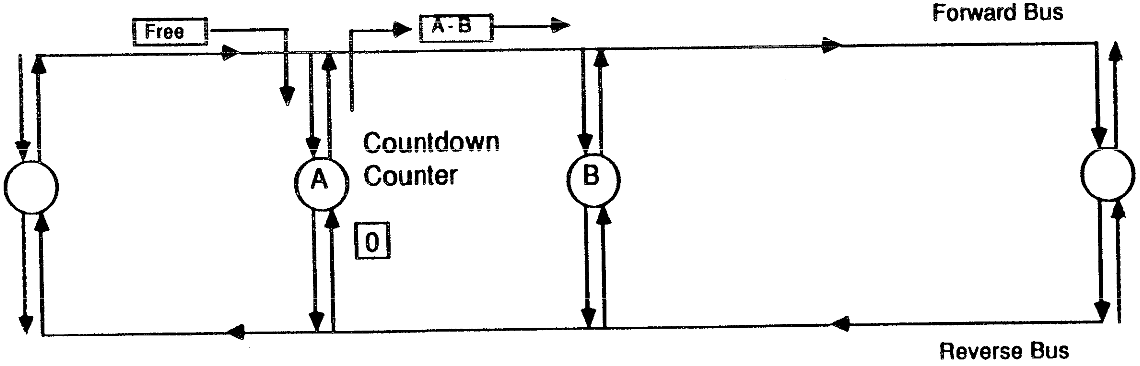
- Countdown Counter gets loaded with Request Counter when a Reservation is sent upstream.

Countdown Counter



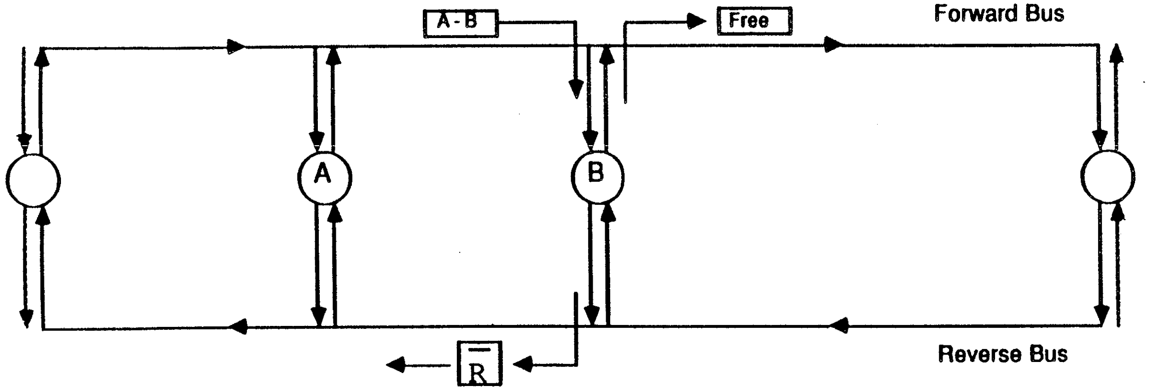
- Downstream Free Slots decrement Countdown Counter.

Packet Transmission



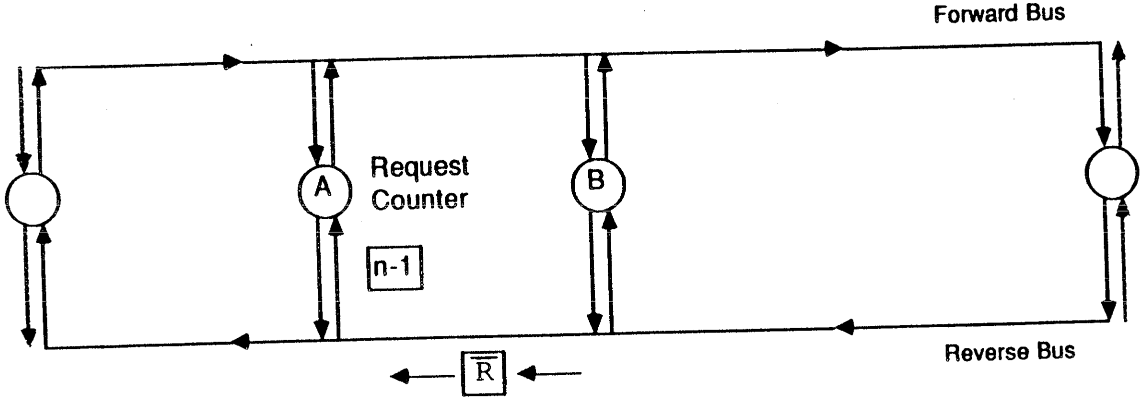
- Transmission occurs when Countdown Counter reaches zero.

Packet Reception



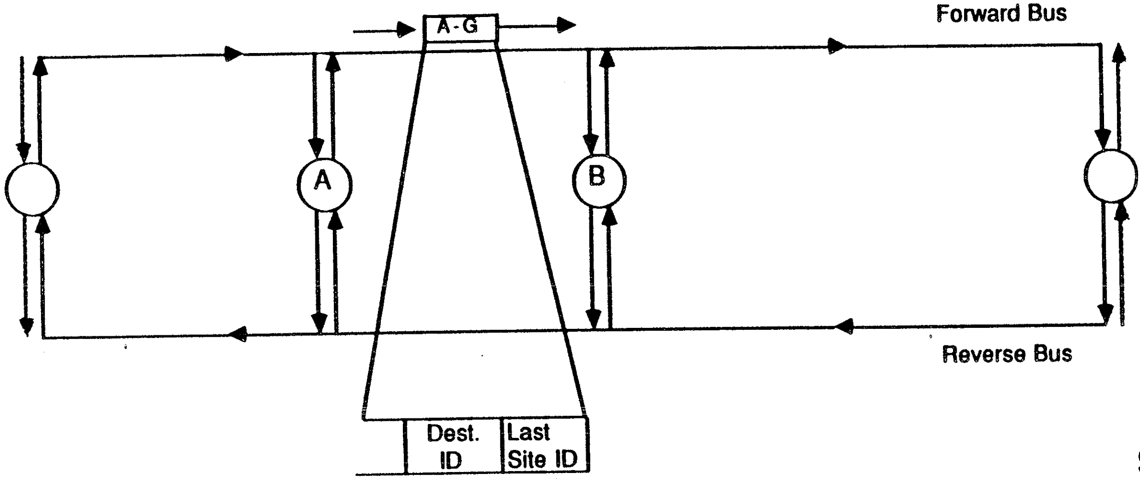
- Receiver removes packet, generates downstream Free Slot and upstream Anti-Reservation.

Request Counter cont.



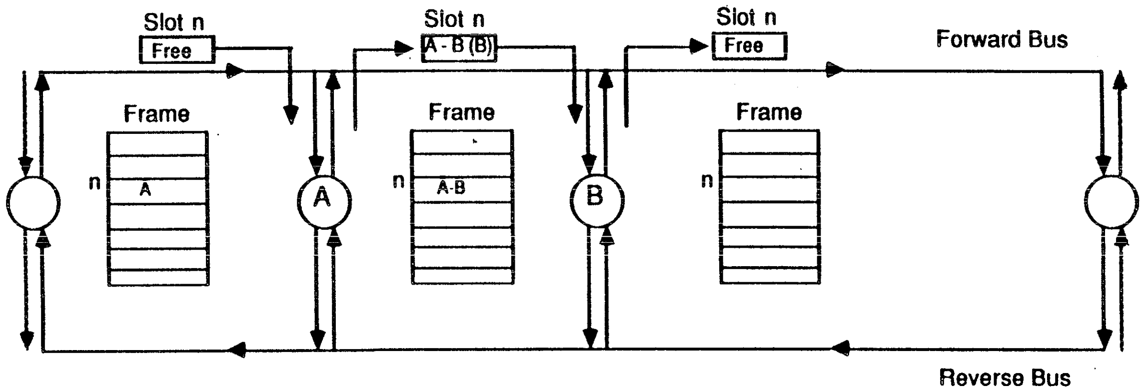
- Upstream Anti-Reservations also decrement Request Counter.

Dynamic Group Implementation



- Packet contain ID of last site to receive it.
- Last site removes packet.
- Destination may be a Group ID.

Stream Implementation



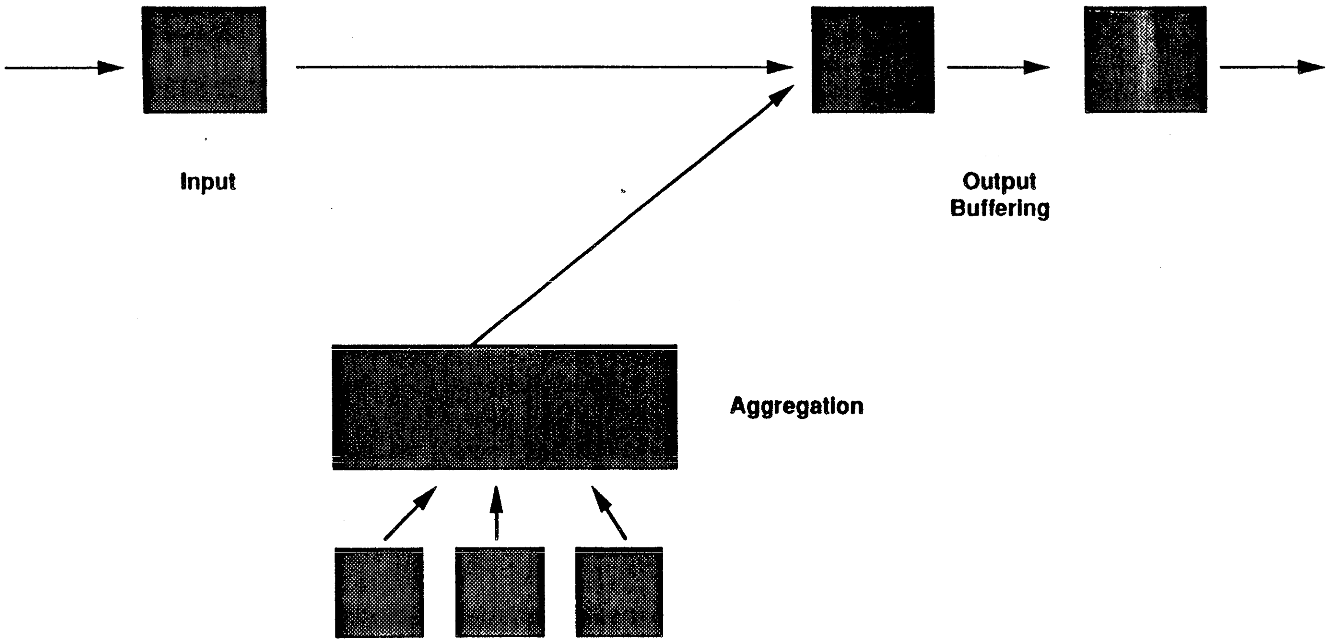
- Slots are aggregated into Frames
- Each site has a Stream Database
- Upstream sites can use stream slot as long as it is freed by the time it reaches the stream owner.
- Single slot can support multiple Streams

Implementation Issues

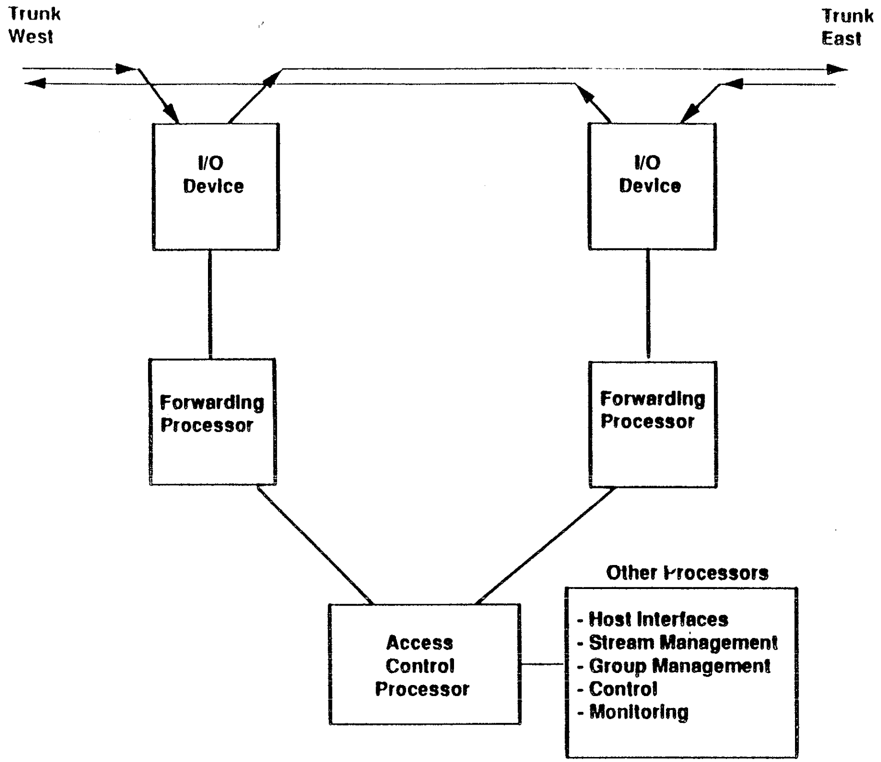
- Butterfly I/O architecture is optimized more for high throughput than low delay

- Dual bus protocol can best be supported by an intelligent I/O device

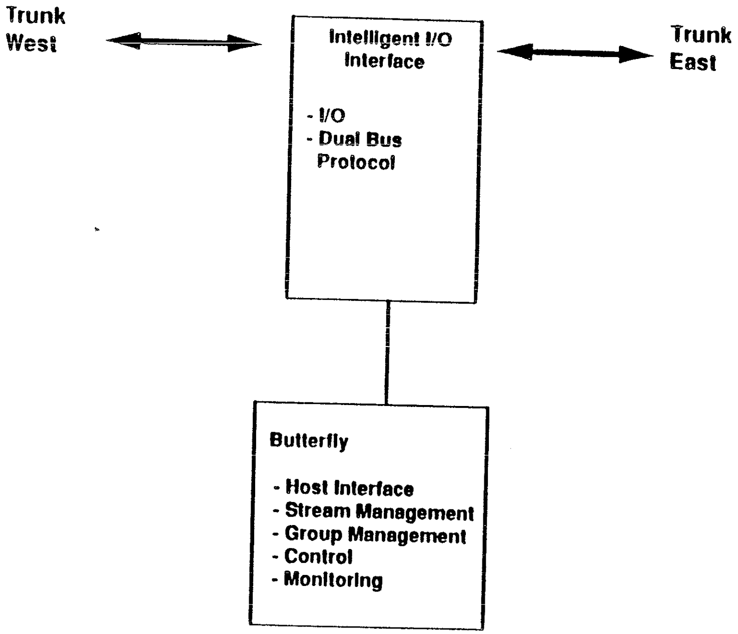
- For best performance, the I/O device should have hardware support for the Dual Bus Protocol



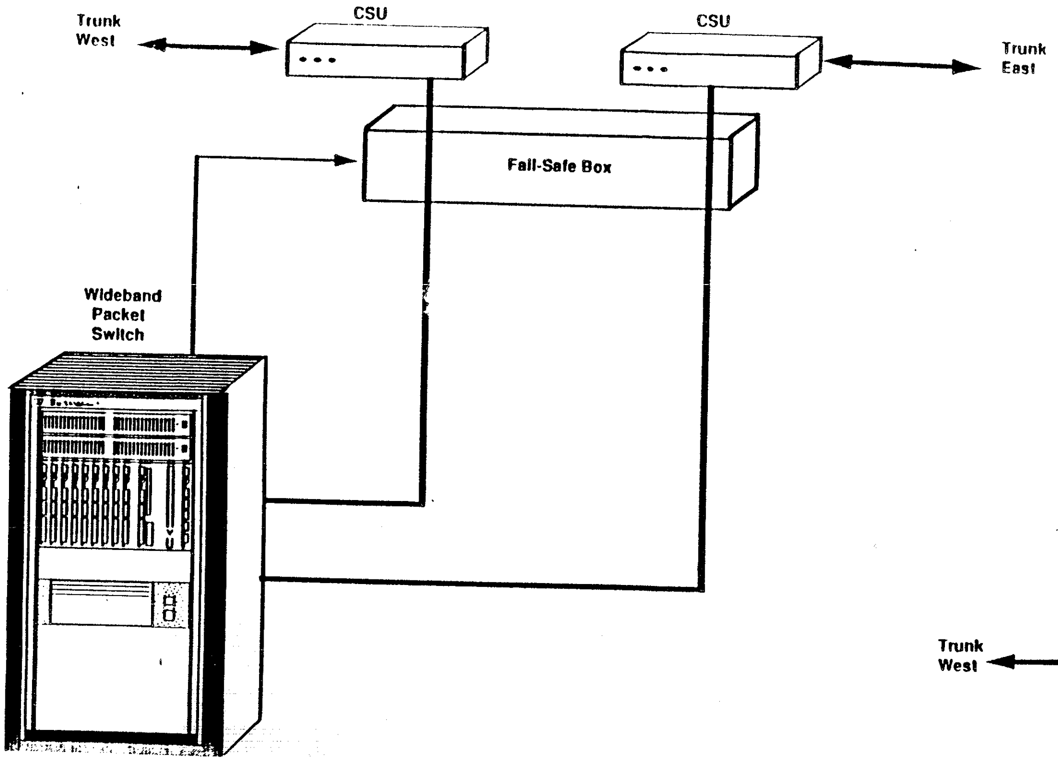
Temporary Differences from Plan



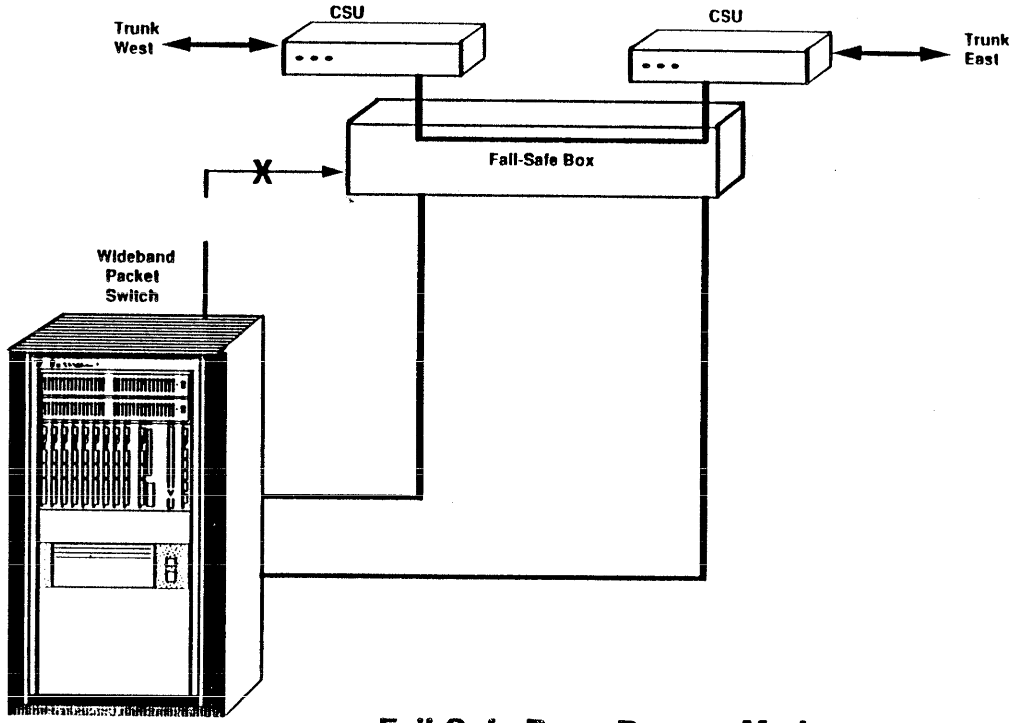
Trunk Processing Configuration



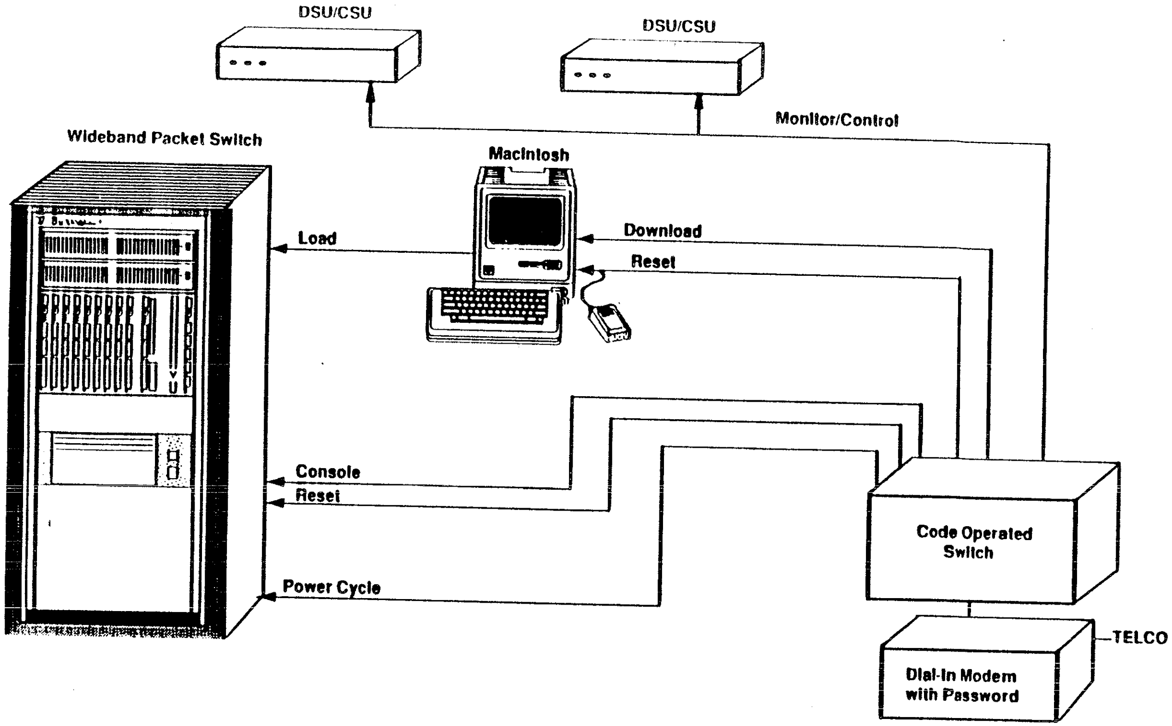
Planned I/O Architecture



Fail-Safe Box - Normal Operation



Fail-Safe Box - Bypass Mode



Support for Unattended Operation

Preliminary Performance Measurements

Throughput:

Channel Bandwidth ----------------------1.544 Mbps

Bandwidth available to IP or ST------------- 1.236 Mbps

Delay (light loading):

Delay into or out of a packet switch----------- 10 ms

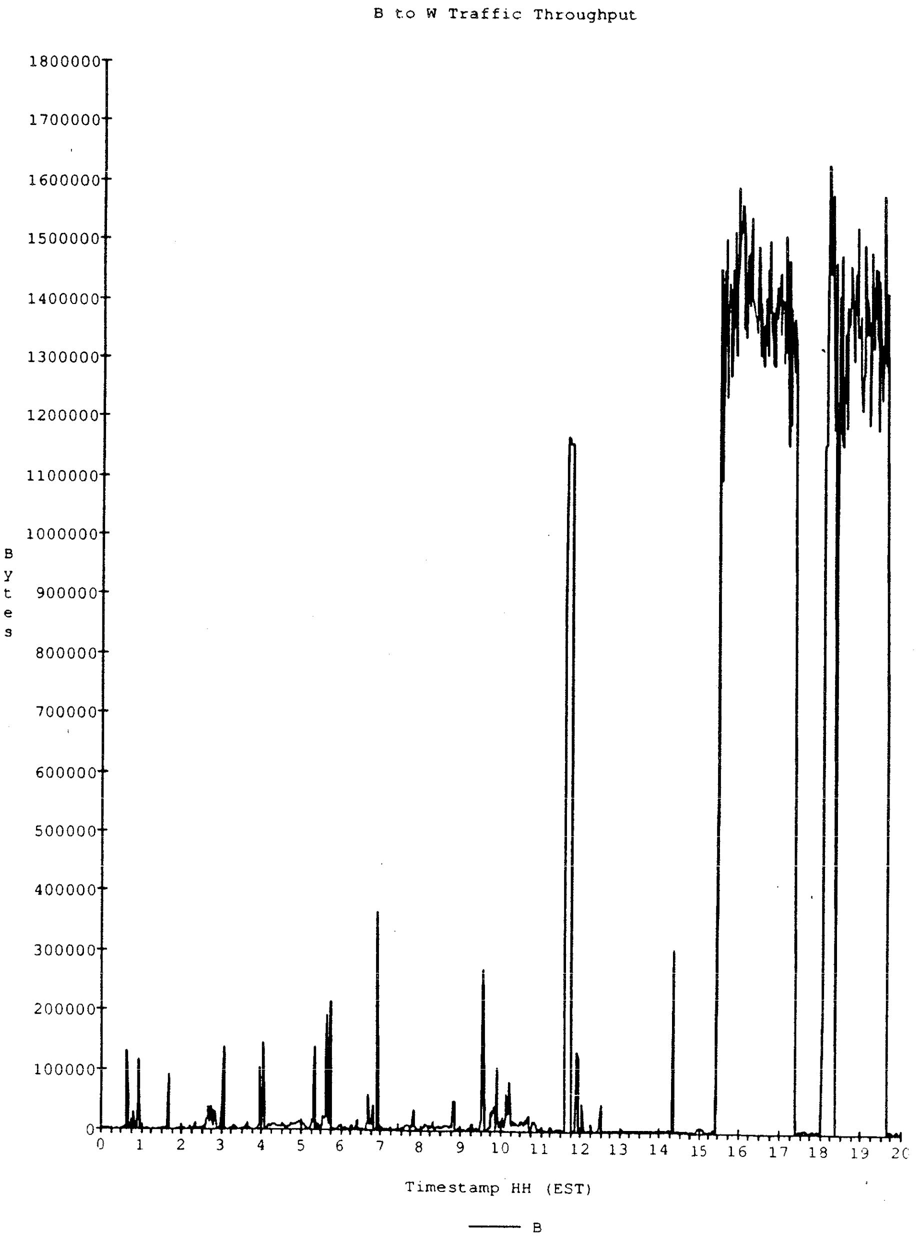
Aggregation delay for small packets-----------25 ms

Transmitter queuing delay---------------- 15-20 ms

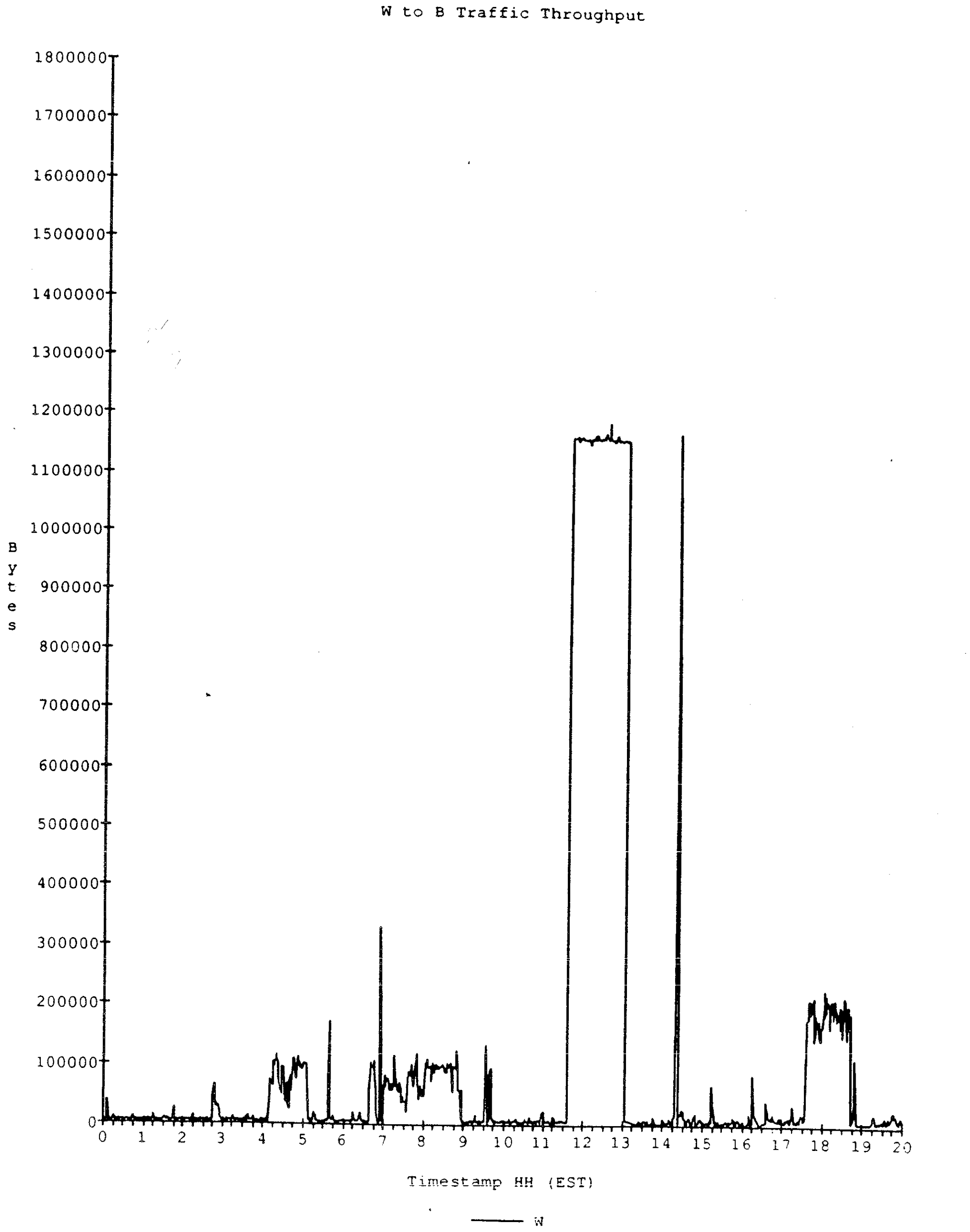
Forwarding delay at each intermediate node-----15 ms

Cross country propagation delay (?) -------- 25-30 ms

BGRAPHWSTATSLOG_JUL19_1 23-JUL-89 19:16 Page 1

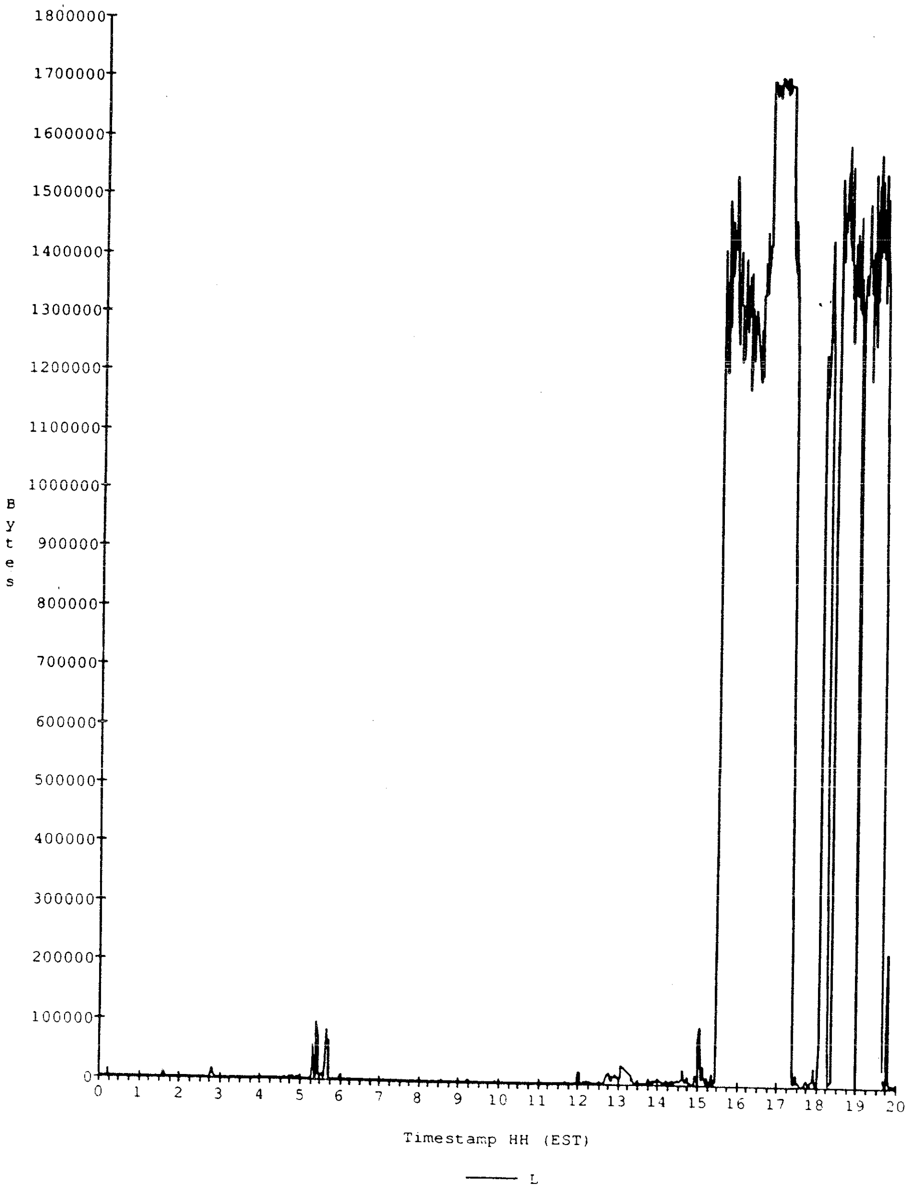


WGRAPHBSTATSLOG_JUL19_1 23-JUL-89 19:14 Page 1



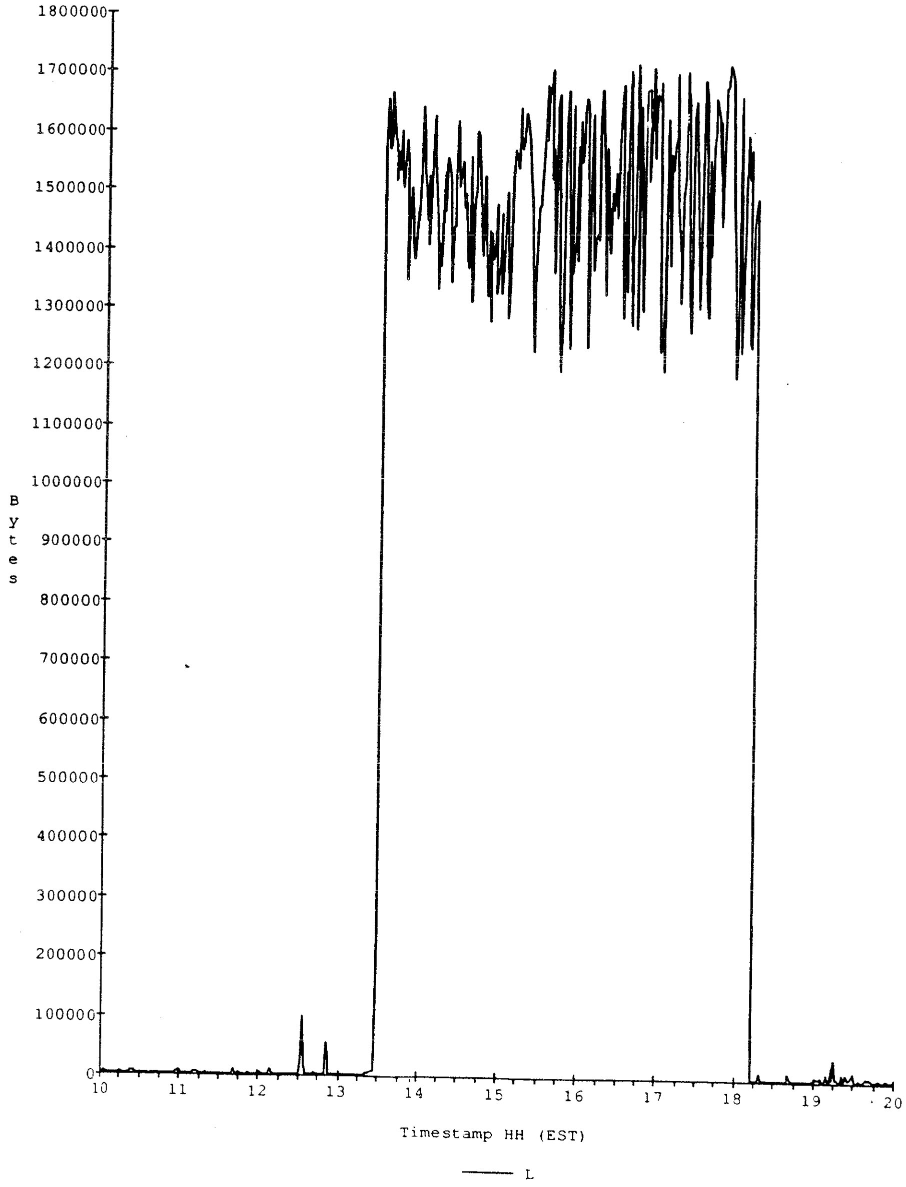
LBGRAPHBSTATSLOG_JUL19_1 23-JUL-89 19:47 Page 1

L to B Traffic Throughput



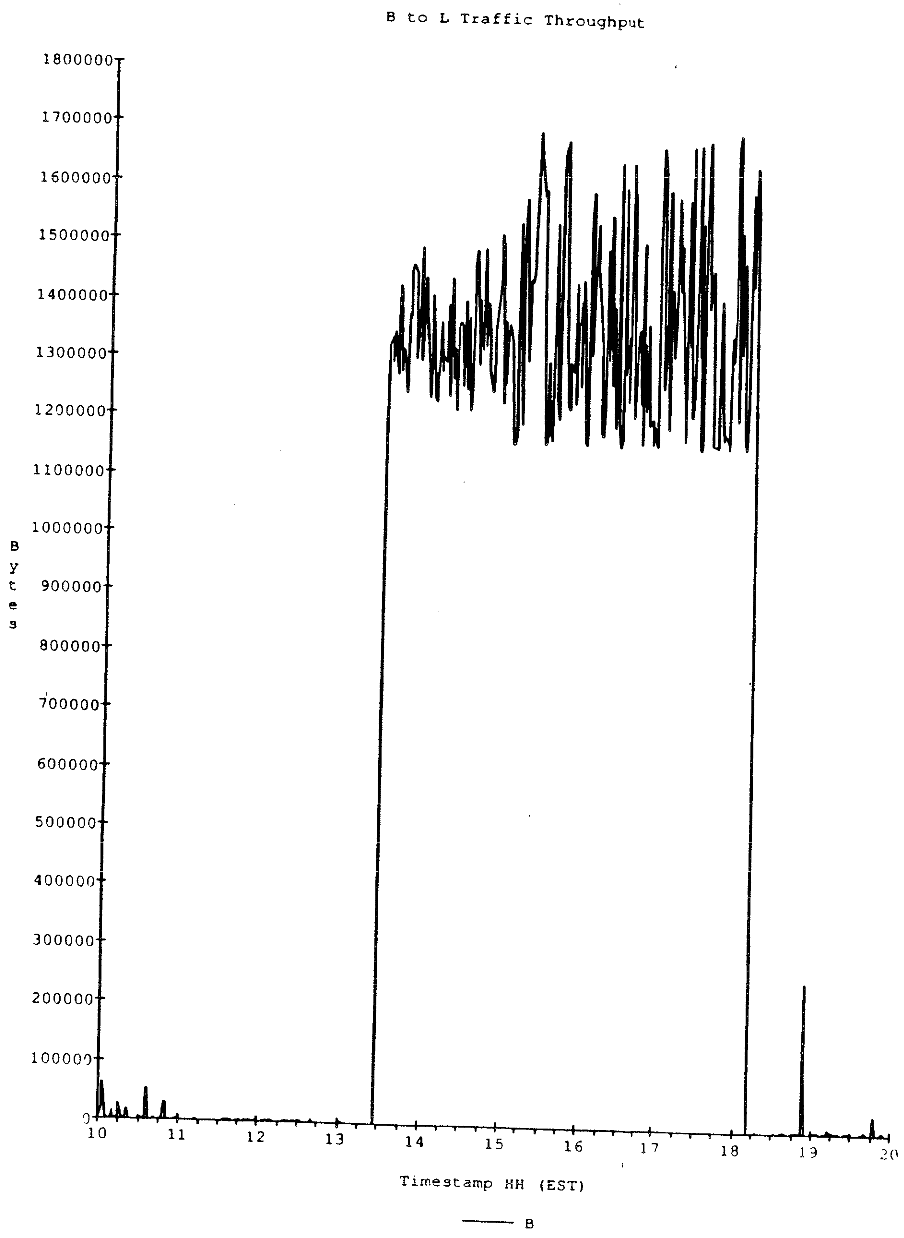
LGRAPHBSTATSLOG_JUL21_1 21-JUL-89 20:47 Page 1

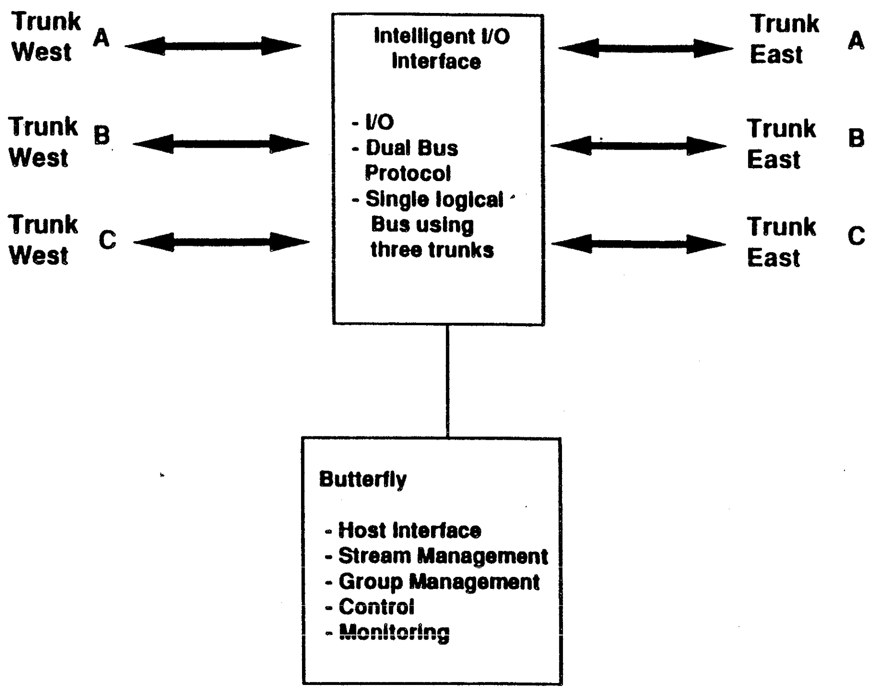
L to B Traffic Throughput



BLGRAPHXSTATSLOG_JUL21

23-JUL-89 18:34 Page 1





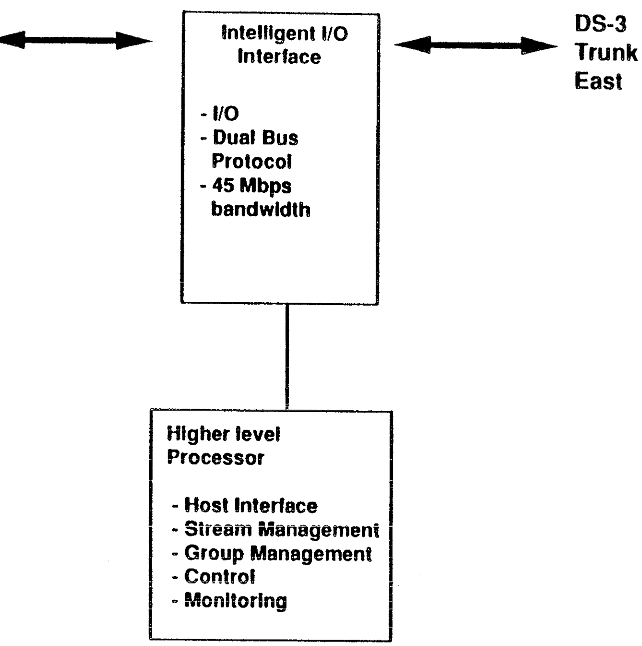
Intelligent I/O Interface

- I/O
- Dual Bus Protocol
- Single logical Bus using three trunks

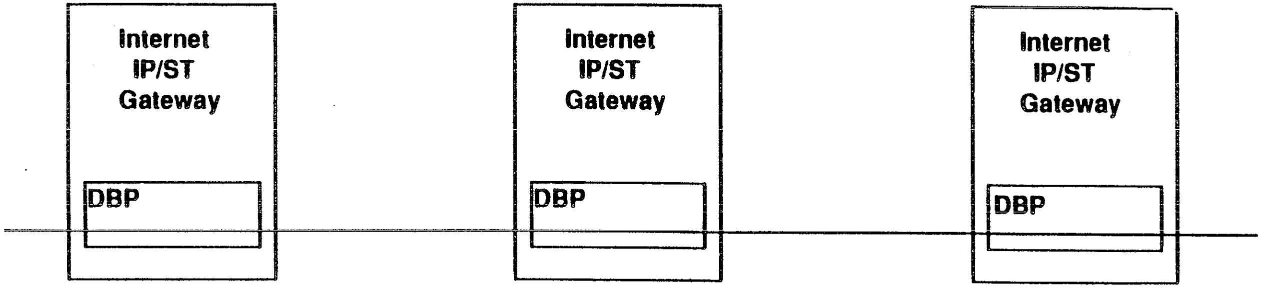
Butterfly

- Host Interface
- Stream Management
- Group Management
- Control
- Monitoring

4.5 Mbps Bus Composed of Three T1 Trunks



45 Mbps Dual Bus Protocol



Dual Bus Protocol (DBP) as "Intelligent Link"
Layer in the Internet Gateways

High Speed Networking using OSI Protocols

Presented by Bob Beach

This presentation describes the UltraNet implementation of ISO TP4 transport protocol. This implementation, which runs on data links up to 1Gigabit/sec, provides an effective transport performance of nearly 800Mbit/sec. The presentation covers the following topics:

- UltraNet architecture
- measured performance on supercomputers, minisupers, and workstations
- protocol choices
- adapter design
- host software integration
- areas of improvement
- new areas of investigation

Ultranet is a high speed local area network using intelligent offboard protocol processing. As noted above, it operates with data links of up to 1gigabit/sec. It also connects to high speed host interfaces such as Cray's HSX, various HSC implementations, and the VME bus. It offers standard host interfaces (BSD sockets) and uses ISO compatible protocols (TP4 and 8602).

UltraNet transport performance is typically 90% of the available data link and/or host interface bandwidth. When operating between a Cray and a Ultra frame buffer, the transport rate is greater than 792Mbits/sec. Between a Cray 2 and a Cray X/MP, memory to memory transfer rates are as high as 480Mbits/sec. Between minisuper computers, the memory to memory rates are in excess of 100Mbit/sec. Workstation performance is in excess of 30Mbit/sec.

The choice of TP4 over TCP/IP was made primarily on the the basis of the performance target of 720Mbit/sec. Studies indicated that given equivalent amounts of hardware, an implementation of TCP/IP would yield only about 480Mbits/sec while an ISO TP4 implementation would yield the 720Mbit/sec goal. The primary difference was due to increased difficulty in parsing TCP packets versus those of TP4. The estimated instruction count difference was as high as three times. Some enhancements were also made to the TP4 protocol to yield slightly better performance. These included increasing the maximum TPDU above the current 8K maximum, putting checksums at the end of the TPDU, and simply optimizing for one particular set of TP4 options.

The adapter hardware architecture is built around a two ported "protocol processor(PP)" and a variety of "personality modules(PM)" that supply host and data link interfaces. Each PP has two PMs attached to it that customize it to a particular host/data link or data link/data link configuration. The PPs are built around dual 125MB/sec internal buses, asymmetric multiprocessing, and a fifo based buffer structure. Different processors are utilized to handle different types of TPDUs. A fast, simple CPU is used for handling DTs and AKs while a more complex, slower CPU is used for other types of TPDUs. This design is based on the observation that greater than 99% of all frames are either data or acks and therefore a high performance implementation should optimize for these two cases. The DT/AK Processor is up to 10X faster than the Control Processor (which handles all other types of TPDUS).

The Ultra adapters are tightly coupled to the host computers to which they are attached. The Ultra host based software provides a dual path structure that allows users of the BSD sockets interface to transparently utilize either the Ultra transport implementation or the host resident implementation. The Ultranet adapter may be used as both a transport service provider and as a high speed data link for a host resident transport implementation.

In summary, Ultranet provides both a fast data link and a fast transport implementation. The level of performance provided by Ultranet allows visualization applications as well as large file transfers to take place at rates never before achieved. This has been done by use of standard protocols and standard interfaces coupled with a unique hardware and software architecture.

High Speed Networking Using TP4

Bob Beach
Ultra Network Technologies

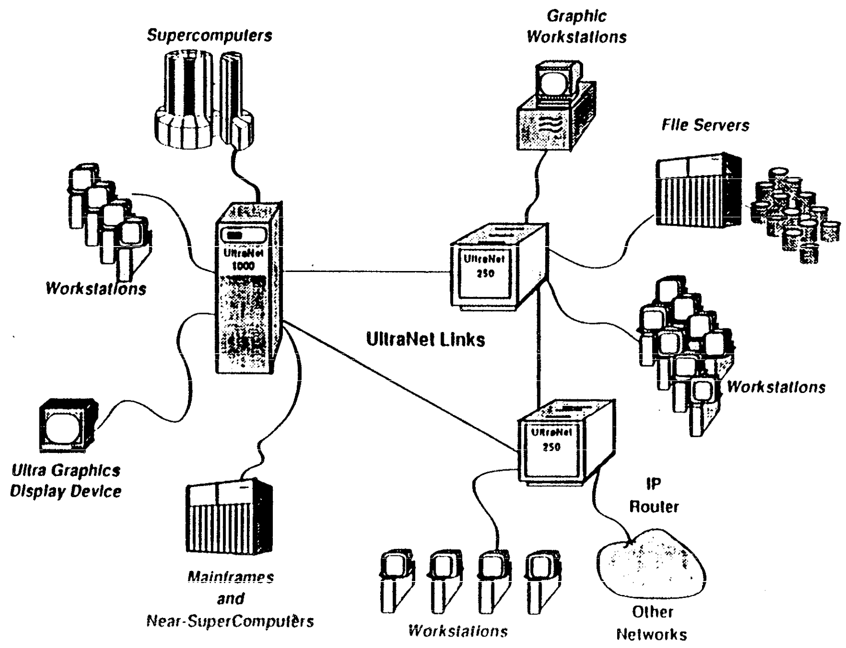
TOPICS:

- UltraNet Architecture
- Ultra Performance
- Protocol Choices
- Adapter Design
- Host Software Integration
- Areas of Improvement
- New Areas of Investigation

UltraNet Architectural Summary

- fast data links -> up to 1 gigabit
- offboard protocol processing -> unique arch
- high speed host interfaces -> HSX, HSC, VME
- operating system integration -> BSD sockets
- standard protocols -> TP4 & 8602

UltraNet Topology



UltraNet Architecture(1):

- "Hub oriented" Network
  - Each hub contains a 1 gigabit/sec bus operated as a LAN (UltraBus)
  - Hubs are independent of one another
  - Hubs contain multiple Host Adapters

- Hosts are connected to offboard transport engines via fast host interfaces (HSX, HSC)

- Two types of hosts connected

  - Channel (Cray, IBM, etc.)
  - Bus/VME (Minisupers, workstations)

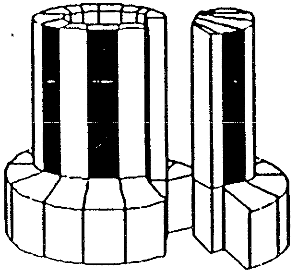
UltraNet Architecture(2):

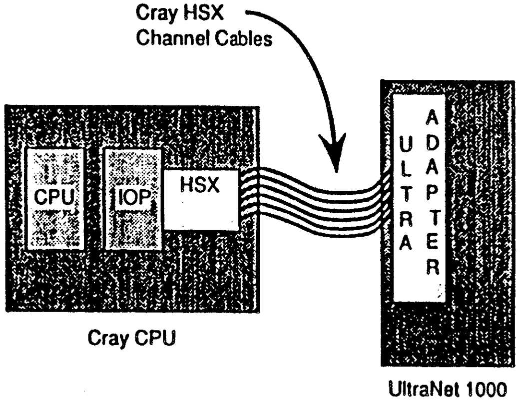
- Serial Links connect Hubs to one another and to VME systems

- Serial links operate from 62.5 Mbit/sec to 1 gigabit/sec

- Hubs, VME systems, and serial links are connected via fast MAC level bridges

Host Adapters
Cray

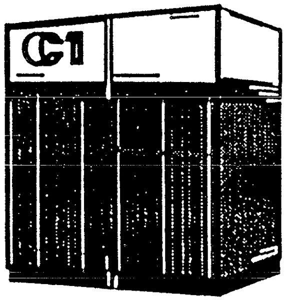


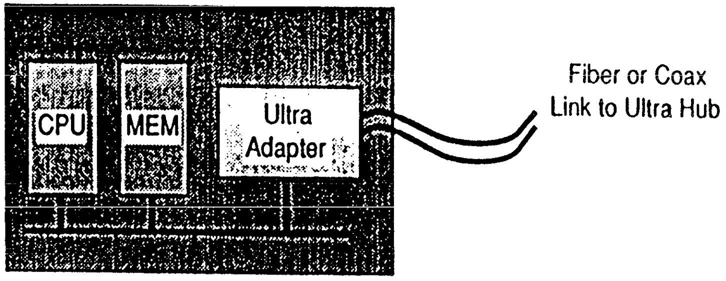


Cray CPU

UltraNet 1000

Host Adapters
VME





VME HOST

UltraNet Performance:

- end-to-end throughput
- adapter transaction rate
- bridge forwarding rate

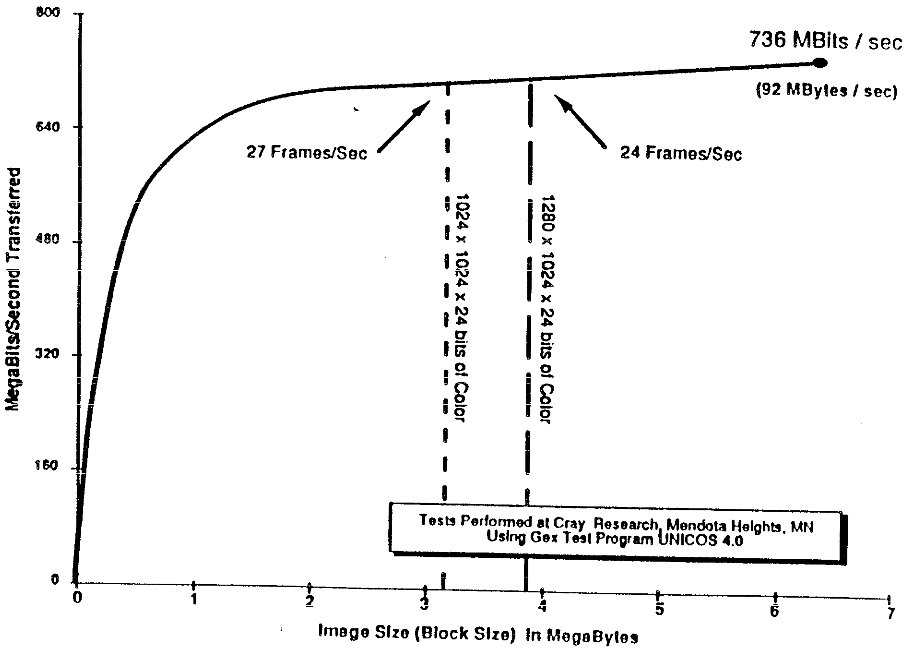
End-to-end TP4 Performance:

- Two implementations
  - Channel Oriented Machines
  - VME based
- Three classes of machines
  - Supercomputer: 100MB/sec channels
  - Minisupers: 10-15MB/sec VME
  - Workstations: 6-10 MB/sec VME
- Measurements are based on:
  - 1 MB user buffer
  - BSD sockets interface
  - Single buffering

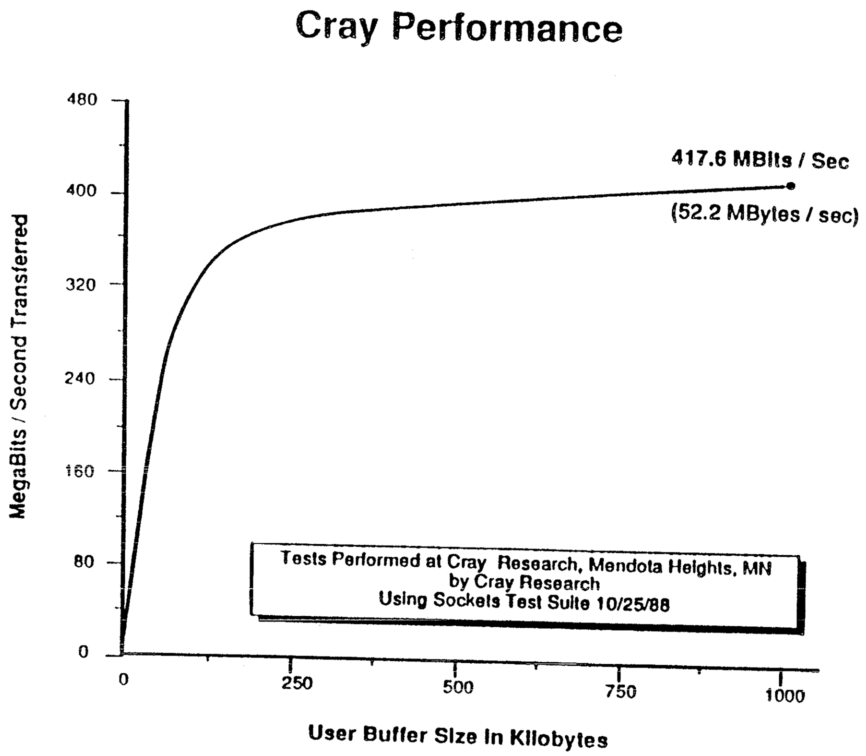
Supercomputer Performance:

- Cray (HSX) to Frame Buffer > 792 Mbit/sec
- Cray (HSX) to Cray (HSX) > 496 Mbit/sec
- CPU is 100ns/cycle bit slice
- Limitation is input channel to host, not protocol processor

Cray 2 to Frame Buffer Performance



Tests Performed at Cray Research, Mendota Heights, MN
Using Gex Test Program UNICOS 4.0

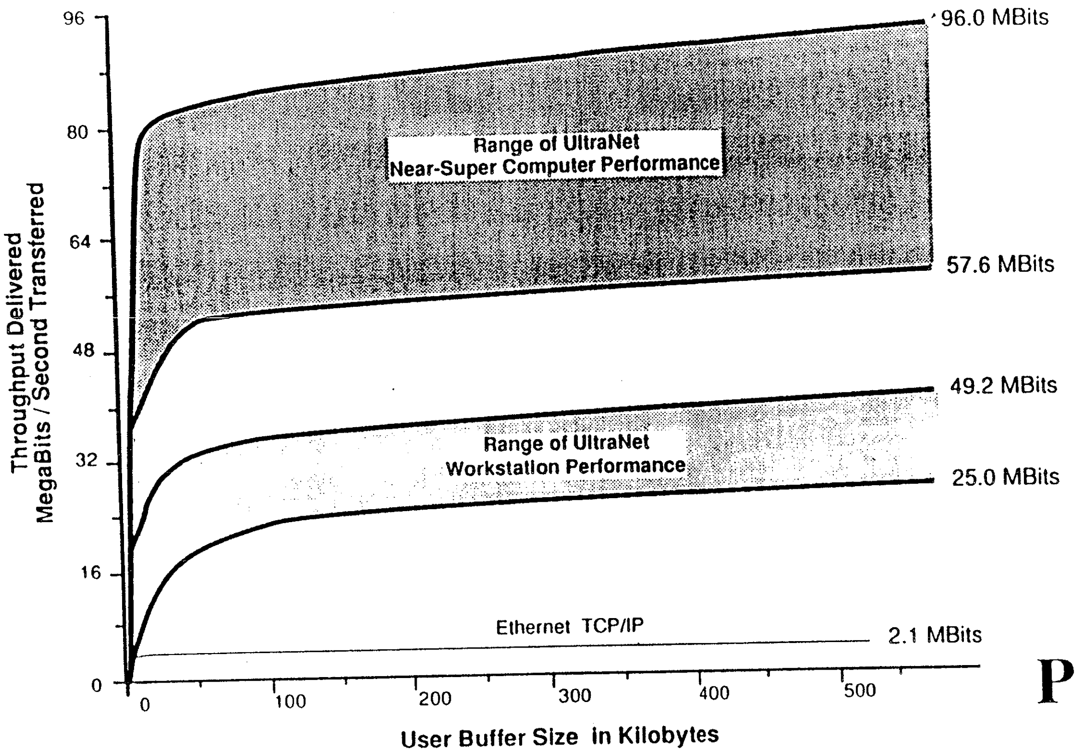


Cray Performance

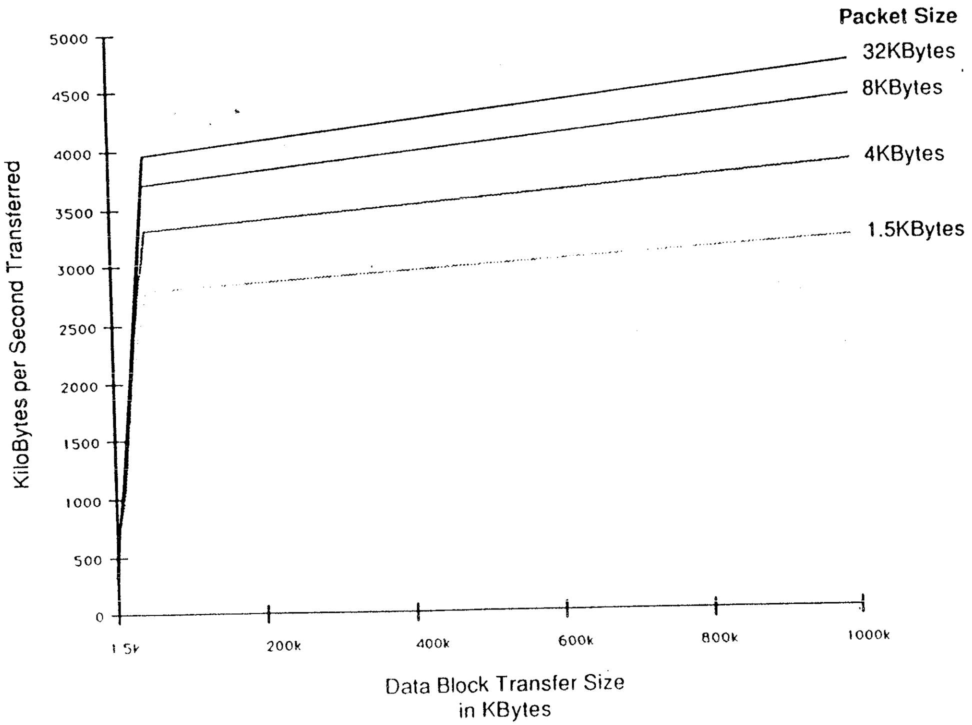
Tests Performed at Cray Research, Mendota Heights, MN
by Cray Research
Using Sockets Test Suite 10/25/88

Minisuper/workstation:

- Minisupers: Convex/Alliant ~ 64-80 Mbit/sec
- Workstations: Sun/SGI ~ 36 Mbit/sec
- Double Buffering increases throughput to 104-112 Mbit/sec on Minisupers
- CPU is 16 Mhz 80386
- Limitations are:
  - host VME bandwidth
  - OS overhead



Packet Size vs User Buffer Size



Adapter Transaction Rate:

- Transaction consists of:
  - adapter receives request from host
  - adapter fetches and formats DT
  - adapter receives AK
  - adapter returns request to host

- Channel based adapter ~ 20K transactions/sec

- VME based adapter ~ 2K transactions/sec

- Host rates are much less

- Packet Rate is 4-5X Transaction Rate

Bridge Forwarding Rate:

- MAC level bridges

- ~ 100K frames/sec

Parsing TCP vs ISO:

- Header sizes:
    - TCP: 60 bytes (40 tcp/ip + 20 subnet)
    - TP4: 32 bytes (16 tp4 + 16 subnet)

- At 1 gigabit/sec, 12 bytes every 100ns.

- Fast uProc ~ 100ns/cycle
    - -> 12 bytes rcvd/cycle
    - -> 128 bytes rcvd every microsecond

- simple field examine of TCP is 2X TP4

- additional TCP costs: port lookup

Protocol Choices: TCP vs ISO

- Choices : TCP/IP, ISO TP4, Homebrew

- Performance goal: >= 736Mbit/sec (92MB/sec)
    - memory to memory (frame buffer)
    - 100 Mbyte/sec host channel (Cray HSX)

- Estimated Performance (given constant hdw)
    - TCP/IP ~ 40-50MB/sec
    - TP4 ~ 95-100 MB/sec

- TCP Limits
    - Harder to parse, IP burden
    - byte model (seq & credit)

Ultra TP4 Extensions:

- Standard TP4 gets to about 50MBytes/sec
- Minor extensions to get to 95+Bytes/sec
    - "Favorite Format"
    - Larger TPDUs
    - Filler parameter
    - Trailer checksum
    - Selective acks
- Most are already in approval process in one form or another

One Set of Options:

- TP4 has lots of options, pick one
- simplifies parsing
- TP4 only
- 31 bit sequence numbers
- checksums on

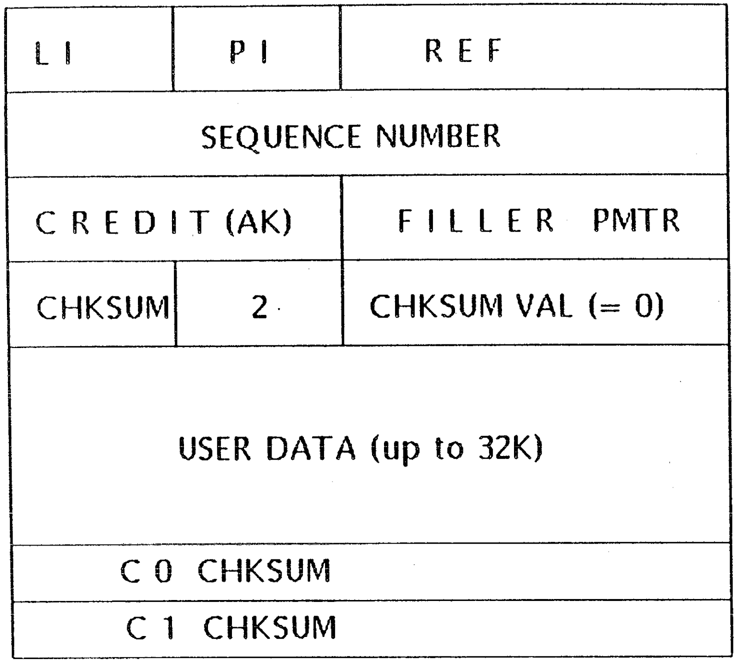
Bigger TPDU Size:

- Current TP4 has 8k byte limit
- effectively limits user data to 4K
- Ultra relaxed the limit
- Using:
    - 32K (16K user data)
    - 64K (32K user data)
    - 128K (64K user data)
- Cray submitted proposal to this effect

Filler Parameter:

- Used to make everything be on 32 bit boundaries
- New type, length = 0

Ultra TP4 TPDU (DT/AK) FORMAT:



<table>
<tr><td>LI</td><td>PI</td><td>REF</td></tr>
<tr><td colspan="3">SEQUENCE NUMBER</td></tr>
<tr><td colspan="2">CREDIT (AK)</td><td>FILLER PMTR</td></tr>
<tr><td>CHKSUM</td><td>2</td><td>CHKSUM VAL (= 0)</td></tr>
<tr><td colspan="3">USER DATA (up to 32K)</td></tr>
<tr><td colspan="3">C 0 CHKSUM</td></tr>
<tr><td colspan="3">C 1 CHKSUM</td></tr>
</table>

Trailer Checksum:

- Rationale: one pass over the data
- Ultra uses hardware checksum, striped 8 bytes wide.
- Keep checksum parameter in header, can move into place at router.
- represented as two 32 bit numbers

Ack Resets (1):

- TP4 suffers badly if destination buffer is smaller than TPDU:
    - some data received, but cannot ack (unlike TCP that does byte acks)
    - must rely on retransmission timer
    - slow and misleading
- Ack Reset is parameter in ack that means "send again" without penalty.

Ack Resets (2):

- To operate a high data rates, need large window.
- If a packet is dropped, need to:
    - not send following packets
    - resend lost frame immediately
- Ack Reset is send when a frame is received out of sequence
- Similiar to Reset TPDU available in TP3.

Congestion Control:

- started out with rate based, but removed it
  - required constant tuning
  - not stable
- moved to "basic V. Jacobson"
- operates okay at gigabit
  - problems with host interactions
  - problems with large TPDU sizes

Data Link Choices:

- Ultra Data links:
  - point-to-point
  - shared backplane/bus
- Why not "real" LAN (ring, for example)
  - Pragmatic: rings are harder
  - Cost: not everyone needs 1 Gigabit
  - History: UltraBus already existed
- Packet size: Bigger the better (almost)
  - -> 32K bytes user data

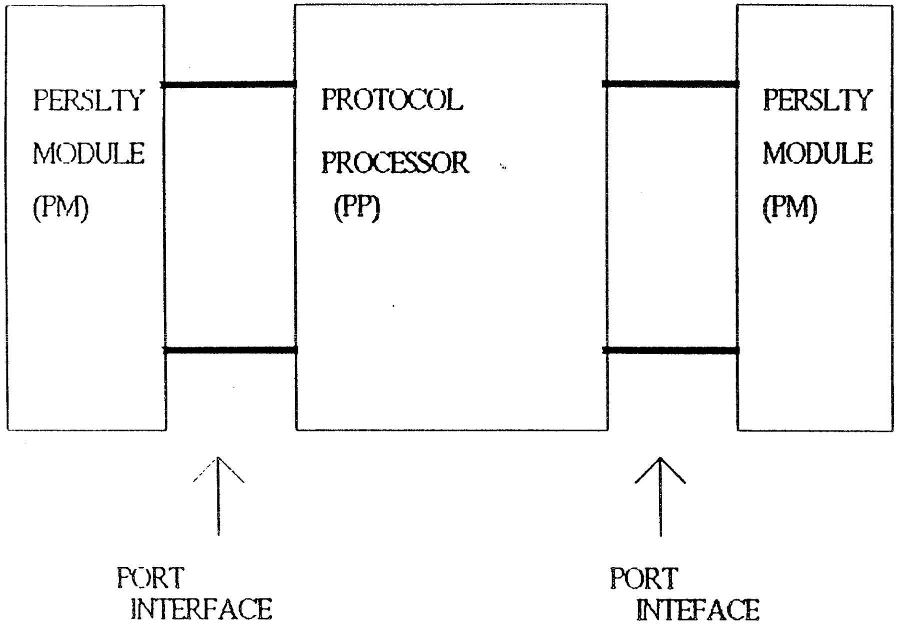
Adapter Design: primary goals

- Interface to gigabit data links
- TL performance >= 736 Mbit/sec
- Multiple Hosts (Channel oriented and Bus oriented)
- Standard user programmatic interface
- Support special devices like frame buffer and bridges
- Reasonable cost, state of the art

Adapter Design Conclusions:

- two designs: Channel & VME
- intelligent, offboard transport engine
- not a lot of onboard memory -> keep user data in host and fetch when needed
- no time to:
    - manage DL controllers
    - manage, buffers
    - copy buffers
- going to hosts is expensive, avoid it when possible

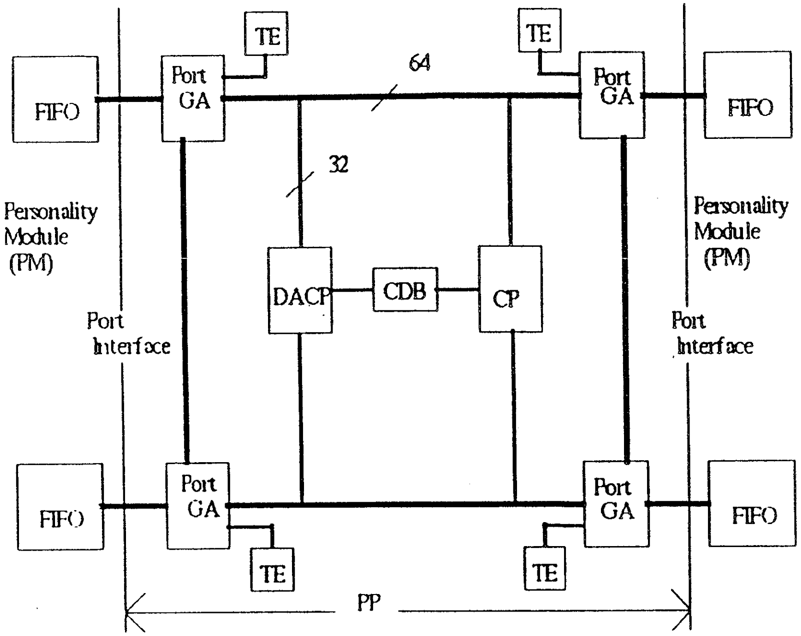
Channel Adapter Architecture:



Adapter Architecture Notes:

- Two ported, smart "protocol processor"
- Can plug in variety of personality modules
  - Host interfaces (HSC, HSC, BMC)
  - Data Link (UltraBus, serial link)
  - Specialized items (frame buffer)

PP Architecture:



PP Architecture Notes:

- 3 specialized processors
    - TE -> moves data
    - DACP -> DT, AK, Command blocks
    - CP -> connection mgmt, network mgmt

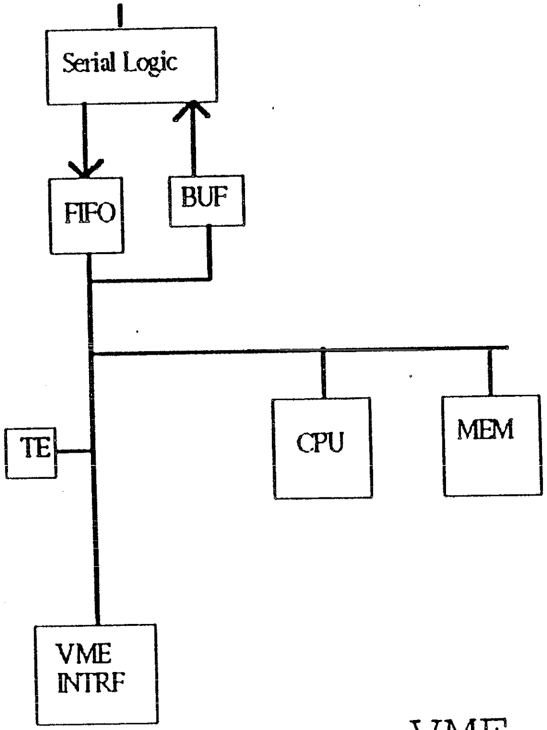
- separate 125MB/sec, 64 bit data paths

- fifos used for:
    - buffer management
    - dl controller (data just "appears")

- hardware checksum (striped 8 bytes wide)

- "flowthrough" architecture

VME Adapter Architecture:

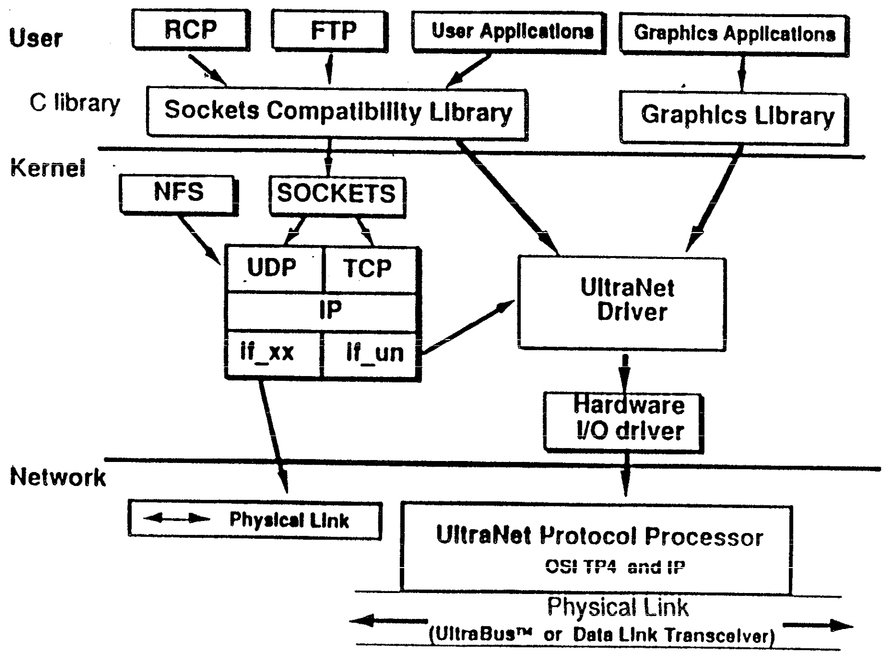


VME Adapter Notes:

- Similiar basic model to PP
- Much lower performance requirement
  - Max of 15-20 MBytes/sec
- Can get to user data directly
- Half duplex transfer to VME

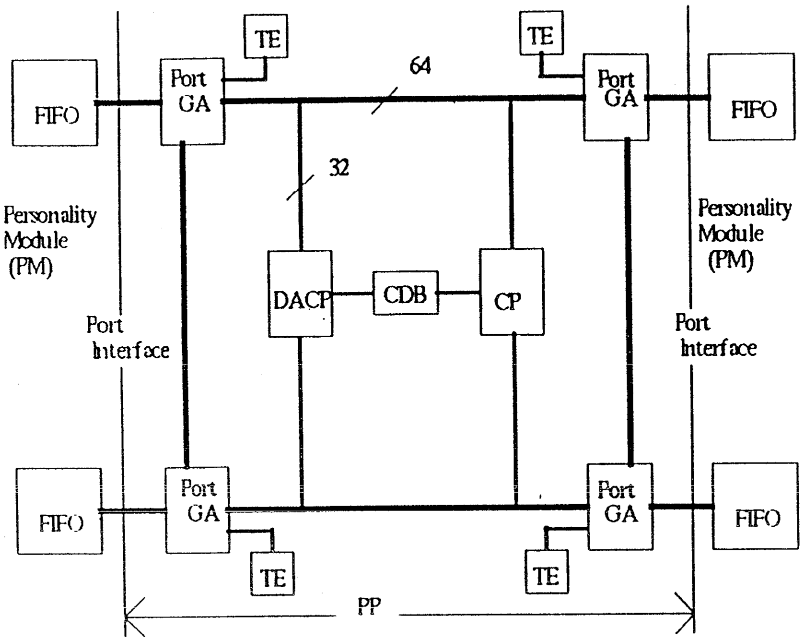
Host Software Integration:

- Runs on variety of Unix systems
- BSD sockets interface
- Designed to operate with host resident network software
- Allows user to transparently move between Ultra "stack" and host rèsident "stack"
- Can operate as subnet to TCP/IP



UltraNet Host Software Model

PP Architecture:



PP Architecture Notes:

- 3 specialized processors
  - TE -> moves data
  - DACP -> DT, AK, Command blocks
  - CP -> connection mgmt, network mgmt

- separate 125MB/sec, 64 bit data paths

- fifos used for:
  - buffer management
  - dl controller (data just "appears")

- hardware checksum (striped 8 bytes wide)

- "flowthrough" architecture

Areas of Improvement:

- minor protocol extensions
- host interfaces
    - fast, low delay path to memory
    - fast, low delay path to host process

Future areas of investigation:

- Univ of Stuttgart
    - long haul (800km), 100Mbit/sec fiber
    - two Ultra hubs and frame buffers
    - connects Cray 2, Convex, Suns
    - first trials in Sept

Conclusions:

- TP4 is highly suitable for high speed nets
- TP4 can be operated at high tl/dl ratios
- key is implementation, not protocol
- minor extensions to protocol can help
- most of our TP4 experiences can be applied to TCP

Domain Name Status Report
Presented by Mary Stahl and Mark Lottor

DOMAIN REGISTRATION AND ADMINISTRATION

The DNS continues its explosive growth, with DDN NIC registrations nearly doubling during the last year. There are now 41 top-level domains and 936 second-level domains registered, while one year ago the total number of all domains registered was only 557.

In January of this year, as a result of DDN Management Bulletin 42, the DDN NIC began the task of transitioning all MILNET hosts from the ARPA domain into the MIL domain. Because most MILNET hosts are not ready to operate their own domain servers and resolvers, this transition is one of names only until that capability exists. The transition process is affecting 2,000 hosts, TACs, and gateways. Although (at the end of June) 41in the ARPA "domain", half of those were in the first phase of the name transition, that of adopting a MIL domain nickname. We expect all MILNET hosts to be fully transitioned (a new MIL hostname is in place and the old ARPA name is a nickname) within the next two to three months.

DOMAIN SURVEY

We periodically execute our domain survey ("tree-walker") program. During the last run of this program, which took several days to complete its run, we gathered data from 4,000 domains. That data provided us with some interesting statistics. A large proportion of the hosts surveyed have only one network address. And strangely, there are some hosts with up to 30 network addresses; we assume that those hosts are carrying on connectivity experiments and do not have 30 actual network connections.

Not surprisingly, the majority of hosts in the DNS are administratively affliated with the COM and EDU domains. Sun workstations seem to be the most prevalent machine type in the DNS with over 29,000 of them sprinkled throughout. UNIX is the most widely used with over 21,000 hosts running that operating system.

DOMAIN NAME SYSTEM STATUS

Mark Lottor

Mary Stahl

DDN Network Information Center (DDN NIC)

SRI International

REGISTERED DOMAINS

COM	494
EDU	318
GOV	30
MIL	26
NET	23
ORG	43

Top-level	41
Second-level	936

MIL DOMAIN NAME TRANSITION

- Transition in progress for all major groups except NAVY
- "Exception list" hosts are in process of being renamed
- By end of June:
  - Hosts still using only ARPA names = 431 (Transition not started yet)
  - Hosts in transition "Step 1" phase = 439 (MIL nicknames assigned)
  - Hosts in transition "Step 2/3" phases = 1,116 (MIL nickname switched with old primary name)
  - Name transition completed = 480

OTHER INTERNET REGISTRATIONS

IP Networks	connected 1653 unconnected 1298
Autonomous Systems	305

NAME CHANGE FOR SRI-NIC HOST

SRI-NIC.ARPA -> NIC.DDN.MIL

6 August 1989

Domain Survey Statistics

Hosts	118,000
Domains	4000

Host Address counts:

0	7601
1	114644
2	2434
3	311
4	122
5	60
6	22
7	12
8	8
9	3
10	2
11	2
12	1
13	1
16	1
29	1
30	6

String searches on host table

mil	5680
com	54700
edu	53600
org	4090
gov	8500
us	180
hp.com	20200
sun.com	12200
sun	29100
ibm	7700
vax	8200
unix	21500

```
              SRI-NIC      Tue 25-Jul-89 20:46:31  Up 32:15:26

                                                      interval 10 secs

Traffic                     Sent     Received   %TX    %RX      dTX      dRX
IP                        890458      1181071   100    100      105      105
   Milnet                 672858       813628    75     68      101       93
   Arpanet                 93784       218709    10     18        4       12
   Nicnet                 123506       150160    13     12        0        0

ICMP                       32717        83559     3      7        0        4
   Echos                    2054         5659     0      0        0        0
   Unreachable               795        43042     0      3        0        2
   Quench                      0          392     0      0        0        0
   Redirects                   0        18204     0      1        0        1

UDP                       206753       394177    23     33       32       35
   Domain                 204350       391725    22     33       32       35
   Other                    2403         2452     0      0        0        0

TCP                       650988       703335    73     59       73       66
   SMTP                   191999       177618    21     15       49       28
   FTP                    135909       119006    15     10        0        0
   Telnet                 223408       289778    25     24       24       38
   Whois                   35807        48671     4      4        0        0
   Hostname                58341        63962     6      5        0        0
   Other                    5524         4300     0      0        0        0
```

Point-to-Point Protocol Specs
Presented by: Drew Perkins, CMU
Written by: Russ Hobby, UC Davis

INTRODUCTION

Point-to-point circuits in the form of asynchronous and synchronous lines have long been the mainstay for data communications. Even with the growing popularity of local area networks, wide area connections are still made using various point-to-point circuits. Over time the data link levels used have settled to a few standards. However, the method of use of the data link by upper level protocols, such as IP, has been largely implementation dependent. This has resulted in requiring a single vendor's equipment for both ends of a point-to-point circuit.

The Point-to-Point Protocol (PPP) seeks to remedy this problem by proposing a standard method of encapsulating IP datagrams, as well as other network layer protocol information, over point-to-point links. PPP also specifies the means for line and protocol operation and maintenance by providing the means for testing and configuring the line and each of the upper level protocols. PPP is designed to be easily extendible for adding new protocols and features.

PHYSICAL LAYER

PPP is capable of operating across virtually any DTE/DCE interface. The most common examples of interfaces are RS-232-C, EIA RS-422. EIA RS-423 and CCITT V.35. The only absolute requirement for PPP is that a duplex circuit be provided. While control signals are not required for the use of PPP, it should be recognized that they do add functionality and it is recommended that they be fully unutilized when they are available from the interface.

DATA LINK LAYER

PPP specifies for synchronous and asynchronous circuits ISO that 3309-1979 (as modified by ISO 3309:1984/PDAD1) be used. The framing specified by this document is more commonly known as HDLC and the 1984 addendum defines HDLC for asynchronous circuits. PPP describes the values to be used in the standard HDLC fields and adds a protocol field to indicate for which protocol the data is destined. There are three major types of protocols, link control, upper level protocol control, and upper level protocol data.

BRINGING UP A LINE

Before a link is considered to be ready for use by upper level protocols, a specific sequence of events must happen. These are:
Phase 1: Link Configuration Exchange - In this phase, Link Control Packets are exchanged and link configuration options are negotiated.
Phase 2: Authentication - In this phase, each end of the link authenticates itself with the remote end using authentication methods agreed to in the Link Configuration Exchange phase. If no authentication is required, this phase is skipped. PPP currently defines a simple user/password authentication protocol. Development of other protocols is encouraged.
Phase 3: Link Quality Determination - If desired the link may be tested at this point to determine if the quality of the link is sufficient for operation. PPP does not specify the policy for determining link quality but does provide low level tools, such as echo request and reply, for testing the line. If no testing is required, this phase may be skipped.
Phase 4: Link Ready - Upper level protocols may be separately configured, and may be brought up and taken down at any time. If the Link Control Protocol takes the link down, it informs the upper level protocols so that they may take appropriate action.

LINK CONTROL PROTOCOL

The Link Control Protocol (LCP) provides a method of establishing, configuring, maintaining and terminating the Point-to-Point connection. LCP handles configuration of the link itself, it does not handle configuration of individual network layer protocols. In particular, all configuration parameters which are independent of particular network layer protocols are configured by LCP.

LCP has packet types to provide link option negotiation, link up/down control, and link testing. Options that can be negotiated cover areas such as Maximum Receive Unit, async control character mapping, authentication methods, encryption methods, keepalive parameters. All configuration options are assumed to be at default values until configuration exchange is completed.

LCP provides the ability to do keepalives and determine packet losses on the link while the link is operating. Policy on when to take the link down or bring it back up is implementation dependent.

Before any other protocol packets may be exchanged, LCP must first open the connection through an exchange of configure packets. This exchange is completed once a configure ack packet has been both sent and received. Any non-LCP packets received before this exchange is complete are silently discarded.

IP CONTROL PROTOCOL

The IP Control Protocol (IPCP) is responsible for configuring, enabling and disabling the IP protocol modules on both ends of the point-to-point link. The IP Control Protocol sequence is the same as the Link Control Protocol allowing for the possibility of reuse of code. Options currently defined for negotiation by IPCP include IP addresses and compression type.

DOD INTERNET PROTOCOL (IP)

Exactly one Internet Protocol packet is encapsulated in the information field of PPP data link layer frames where the protocol field indicates DOD Internet Protocol.

FUTURE WORK

There are many areas for future work on PPP . Currently only use of the IP protocol has been defined on PPP. To make use of PPP for other protocols the control protocol and rules for the data encapsulation need to be defined. The simple authentication method currently defined is quite weak. A stronger authentication method needs to be defined. Also, while there is a means to negotiate encryption methods, no encryption methods have been defined.

PPP INTERNET-DRAFTS AVAILABLE

Two Pt-Pt Protocol internet-drafts are available on-line at SRI-NIC.ARPA and can be accessed by anonymous ftp: 1)"Requirements for an Internet Standard Point-to-Point Protocol" and 2) "The Point-to-Point Protocol (PPP), A Proposed Standard for the Transmission of IP Datagrams over Point-to-Point Links". The directory name is "internet-drafts:". The file names are "draft-ietf-ppp-req-00.txt" and "draft-ietf-ppp-ipdatagramstx-01.txt", respectively. For more information, please contact:

- ietf-request@venera.isi.edu

History

No Internet "Standard"

Defacto Standard - SLIP · Async

Anarchy - Sync

Low Functionality

Requirements

Simplicity

Transparency across all common transmission media

Efficiency

Multiple Protocols

Error Detection

Standard MTU

Switched & Non Switched Media

Symmetry

Line Quality & Connection Liveness

Loopback Detection

MisConfiguration Detection

Address Negotiation

Compression

Extensibility

Frame Format

*HDLC Frame Structure for maximum Interoperability

Flag	Address	Control	Protocol
01111110	11111111	00000011	*
8 bits	8 bits	8 bits	16 bits
	All-Stations	UI	

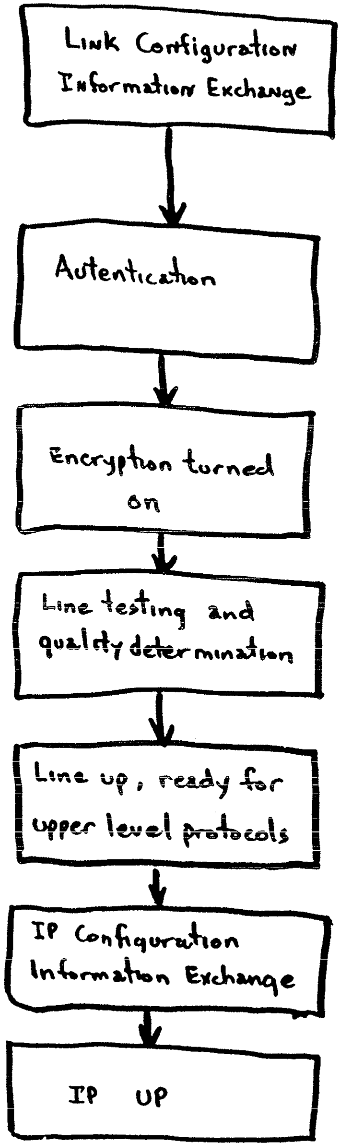
Information	FCS	Flag
n bits	16 bits	
< 1024 octets	CCITT-CRC	

Sync - Bit stuffing Transparency

Async - Character " "

- all control characters

PPP Sequence



MTU.
Async Control Character Map.
Authentication Method.
Encryption method.
Keep-alive Paramders.

Simple User/Password defined.
Development of other methods encouraged.

No methods currenty defined.
Encryption only on data fields of protocols.
Not used on Link Control Protocol.

Echo Request/Reply Packets
Keep-alive Packets.
Testing method suggested.

Upper level protocols can be brought up and taken down at any time.

IP Address
Compression method used

Link Control Protocol

Code 8 bits	Identifier 8 bits	Data n Octets

Configure Request
" Ack
" NAK
" Reject
} Data = Options

Terminate Request
Ack

Echo Request
Reply

Discard Request

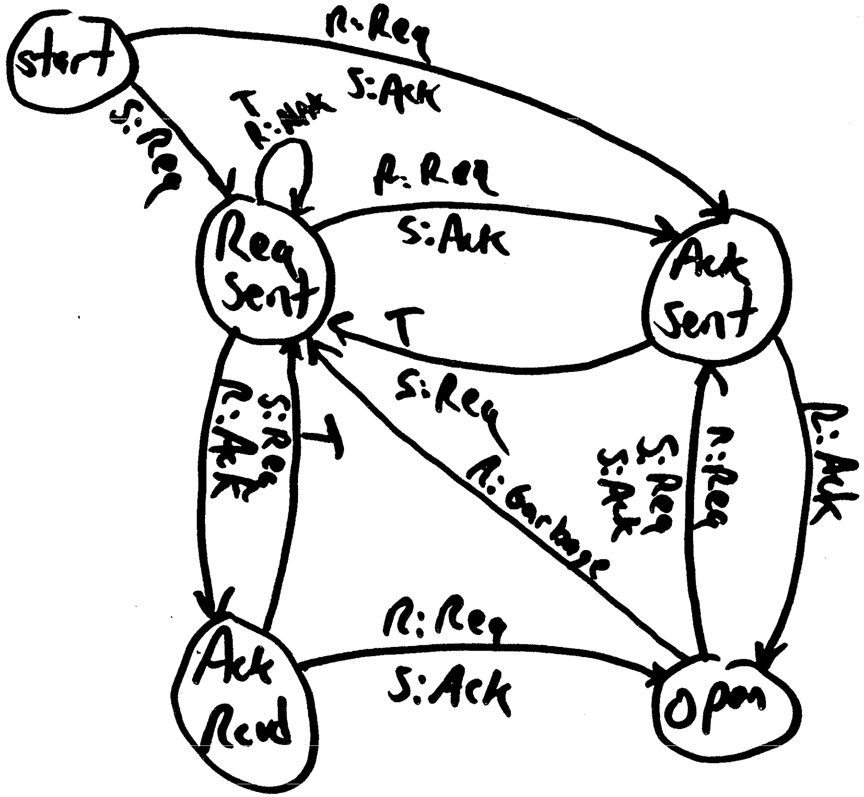
Code Reject

Protocol Reject

LCP options

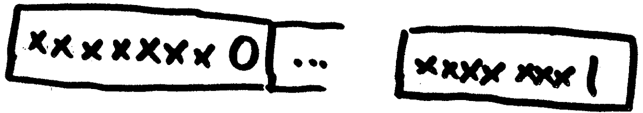
Maximum Recv Unit (MRU)
Async Control Character Map
Authentication Type
    - Weak User / Password
Encryption Type
Magic Number
    - Loopback Detection
Line Quality
Protocol Field Compression
Address / Control "

State Diagram

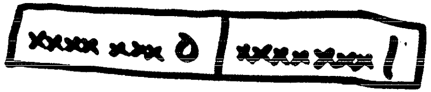


Protocol Field Compression

Encoding Scheme



we set limit to 1 extension octet so usually

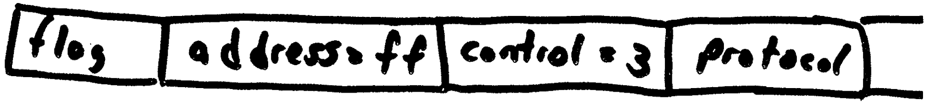


or IP =

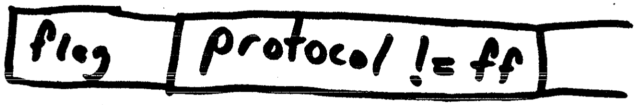
After option

Address/Control Field Compression

Before



after



Line Quality Option

Period @ which you should send me keepalives is specified in option

---

- Keepalive frame includes:

total transmits ~~since last~~ LQ packet sent

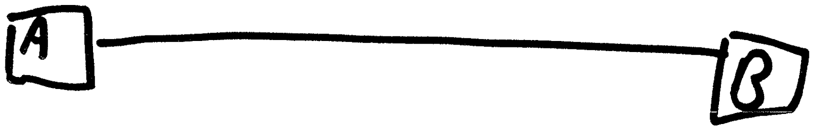
transmits specified in last LQ packet rcvd minus # rcvs measured @ reception

---

rest of decision is entirely local policy

we provide mechanism only

Example of LQ



1. req LQ P=3 →
2. ← ack LQ P=3
3. ← req LQ P=5
4. ack LQ P=5 →

---

5. LQ T=10 →
6. ← LQ T=8 T^R-R=3
7. Term Req →

Weak User/Password Auth

Packet Format

Code	Identifier	Data....
8 bits	8 bits	

Authenticate

User len	Userid
8 bits	n Octets

Pass len	Password
8 bits	n Octets

Auth Ack/NAK

msg len	message
8 bits	n octets

IP Control Protocol

IP Addresses

Compression Type

Some European Internet Activities
Presented by Rudiger Volk

Some European Internet Activities

Rüdiger Volk
Universität Dortmund, Germany
rv@germany.eu.net

messages

background

working group
current topology

Messages:

- IP is used in Europe
- Europeans want to connect to the Internet (by IP)
- Europeans have started to coordinate international IP
- invitation to cooperate

Players

national research networks

RARE

  COSINE

EARN

EUnet

special cases

user group networks

(EASInet)

- lines are expensive
- international lines are more expensive
- funding goes to OSI
- alternate network operations (sometimes) needs to be paid for by users
- improving/emerging infrastructure
  ⟹ booming user demand for IP interconnection in near future

IP coordination in Europe
(Réseaux IP Européenne)

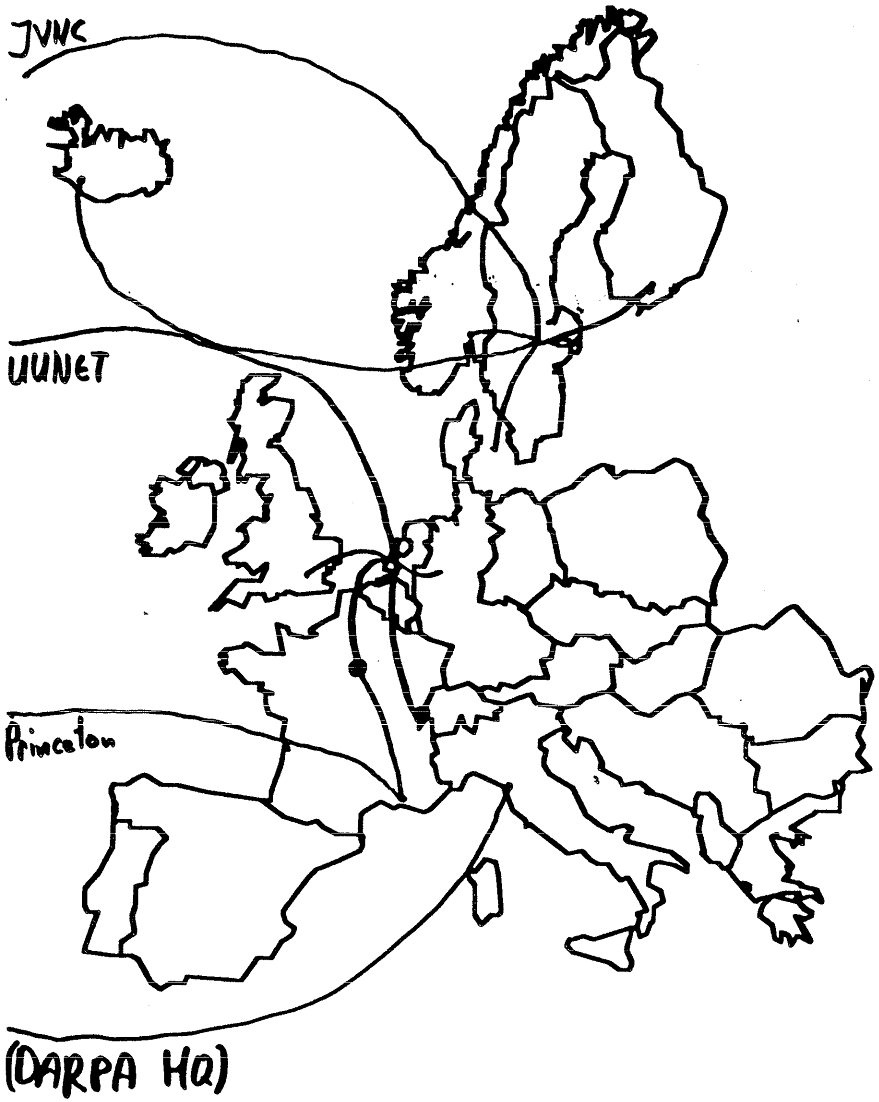
EUnet

Nordunet

HEP

outlook to national research networks

get it settled within Europe



State of the Internet

Presented by Zbigniew Opalka

The number of networks in the Internet continues to grow. Since the beginning of the year, the number of networks, as reported by the Butterfly Mailbridges, has risen from approximately 750 to 870 (July 27th figure).

Mailbridges

All six of the Butterfly Mailbridges are operational. They are currently supporting 204 neighbors (combined on both the Arpanet and Milnet). Ethernet interfaces were added to the Mitre and Ames Mailbridges. The Ames connection was up as of July 27, 1989. The Mitre Mailbridge Ethernet connection became active in the beginning of August.

There were several problems reported with the Mailbridges. These included hardware problems at Ames and routing loops. The routing loops were the most problematic. A patch was tested and deployed to filter EGP updates; this patch would be used to control routing loops.

The throughput of the Mailbridges rose significantly and is expected to rise more with the installation of the Ethernets in Ames and Mitre. The Mailbridges were passing around 8 million packets per day in May; this rose to over 12 million in June and July.

The drop rate across the Mailbridges for queuing reasons (not counting ttl expiration and unreachability) has increased slightly, due mainly to Ethernet/Milnet interconnectivity. Currently a total of .25% packets were dropped. The average packet length is around 150 bytes.

VAN Gateway

The VAN Gateway is an IP-based router that connects a PDN, in this case TELENET, with the Internet. Initially the VAN Gateway would not be allowed to make calls on the PDN only to receive them, in order to minimize cost. It would exchange EGP information with other gateways on the PDN, providing reachability information for those gateways whose sole route to the INTERNET was through a PDN.

On April 1989 BBN completed testing and deployment of the Butterfly VAN Gateway. By June, the VAN Gateway was successfully running EGP across TELENET with a CISCO gateway at CNUCE, Italy. At that time the TELENET link was providing CNUCE with its only link to the rest of the INTERNET.

The VAN Gateway provides a reliable backup for European sites that connect to the INTERNET, if the regular connection fails. During an outage of the BBN-RSRE (UK) link, the VAN Gateway exchanged EGP information across TELENET with RSRE.

The current outstanding issues in the VAN Gateway status is to provide reverse charging as requested by DND, Canada and to configure the TELENET PSN to permit window and packet size negotiation. The latter is useful when exchanging very large EGP updates across satellite links to Europe.

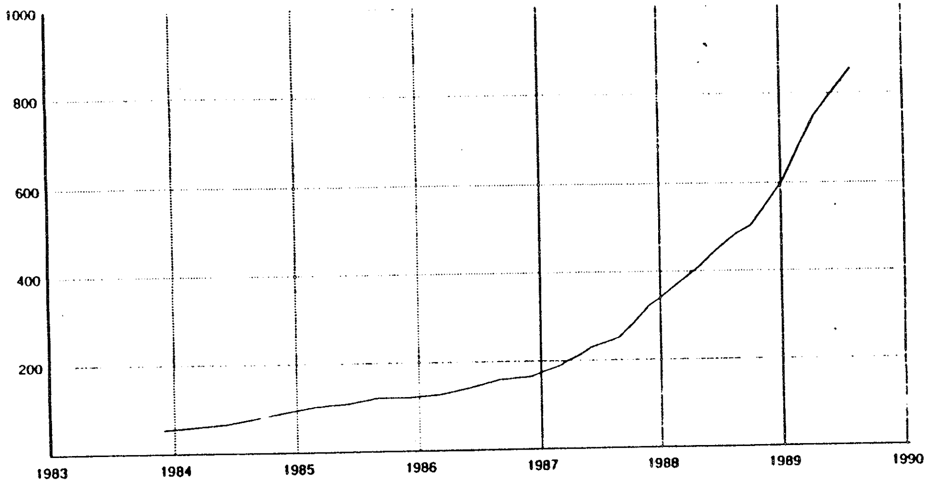
TOPICS

- Internet Growth
- DDN Mailbridges
- VAN Gateway

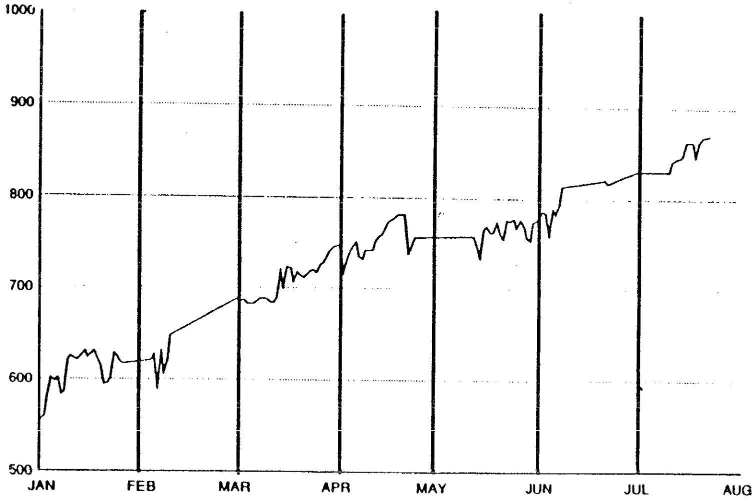
INTERNET GROWTH SUMMARY

- 870 Networks advertised
- 1710 Networks registered

NUMBER OF NETWORKS
December 1983-July, 1989



NUMBER OF NETWORKS
January 1989-July 1989



DDN MAILBRIDGES

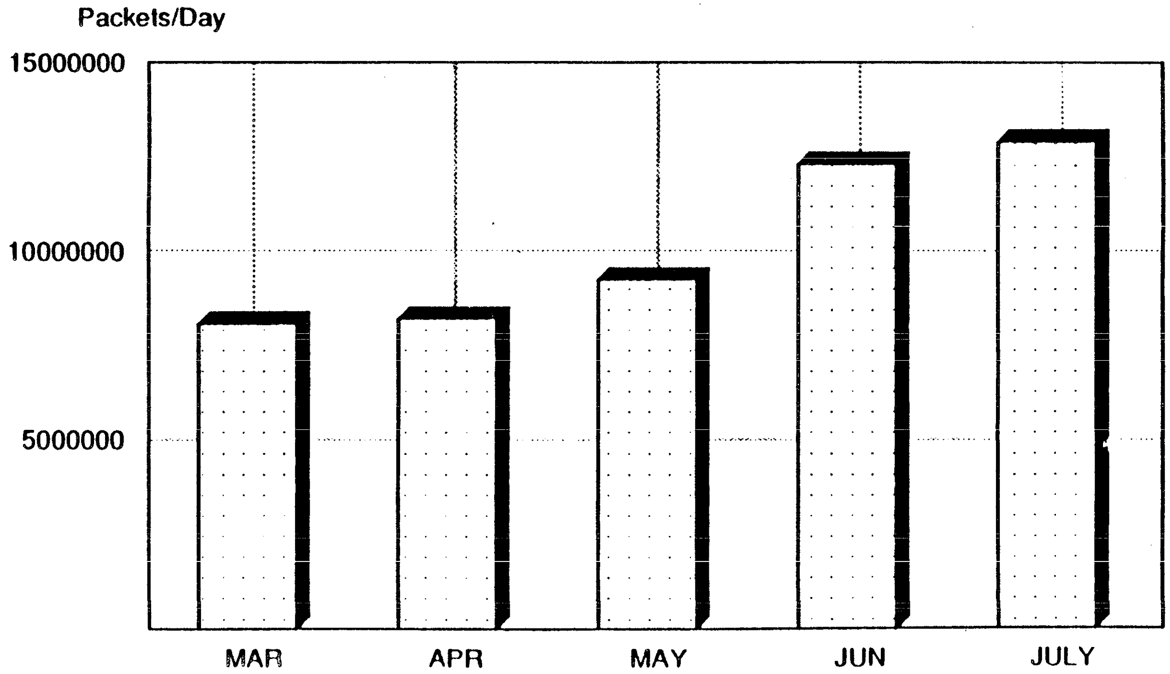
CURRENT STATUS

- Six DDN Butterfly Mailbridges operational
- 204 EGP neighbors
- Ethernet interfaces added to Mitre and Ames mailbridges
  - 192.52.194-NSFTRANSIT 5
  - 192.52.195-NSFTRANSIT 6
- Direct NSFNET connection
  - Ames - 6/27/89
  - Mitre - August

MAILBRIDGE PROBLEMS

- Ames mailbridge restarting
- Routing loops of down networks
  (patch being deployed currently)
- Hardware problems at Ames

MAILBRIDGE THROUGHPUT



TRAFFIC SUMMARY

- ~ 12,000,000 packets/day forwarded
- .25% packets dropped
- ~ 150 bytes per packet

VAN GATEWAY

STATUS

- **April 1989 - Completed testing and deployment of Butterfly VAN Gateway**
- **June 1989 - Successfully running EGP across PDN with CISCO Gateway at CNUCE. PDN is providing CNUCE with its only link to the Internet.**
- **During outage of BBN-RSRE link, VAN Gateway exchanged EGP information across PDN, with RSRE Gateway. VAN Gateway provided a backup link to the Internet.**

OUTSTANDING ISSUES

- **Configuration of Telnet PSN to permit window size and packet size negotiation**
- **Reverse charging to DND, Canada**

OBSERVATIONS

- **When running EGP with VAN, do not negotiate down to packet size of 128 bytes and window size of 2, especially over a satellite hop.**
- **Recommend 512 byte packets and window of 7**

ESNet Status Report
Presented by Steven Hunter

As we reported at the last IETF meeting, we have completely changed the proposed ESnet design because DECnet traffic may now be handled directly. We were previously not permitted to do so because it is not a public protocol, so the old design had DECnet wrapped in X.25. The new network will use all commercial hardware and software and we will become a peer network with NSFnet, NASA, and MILnet.

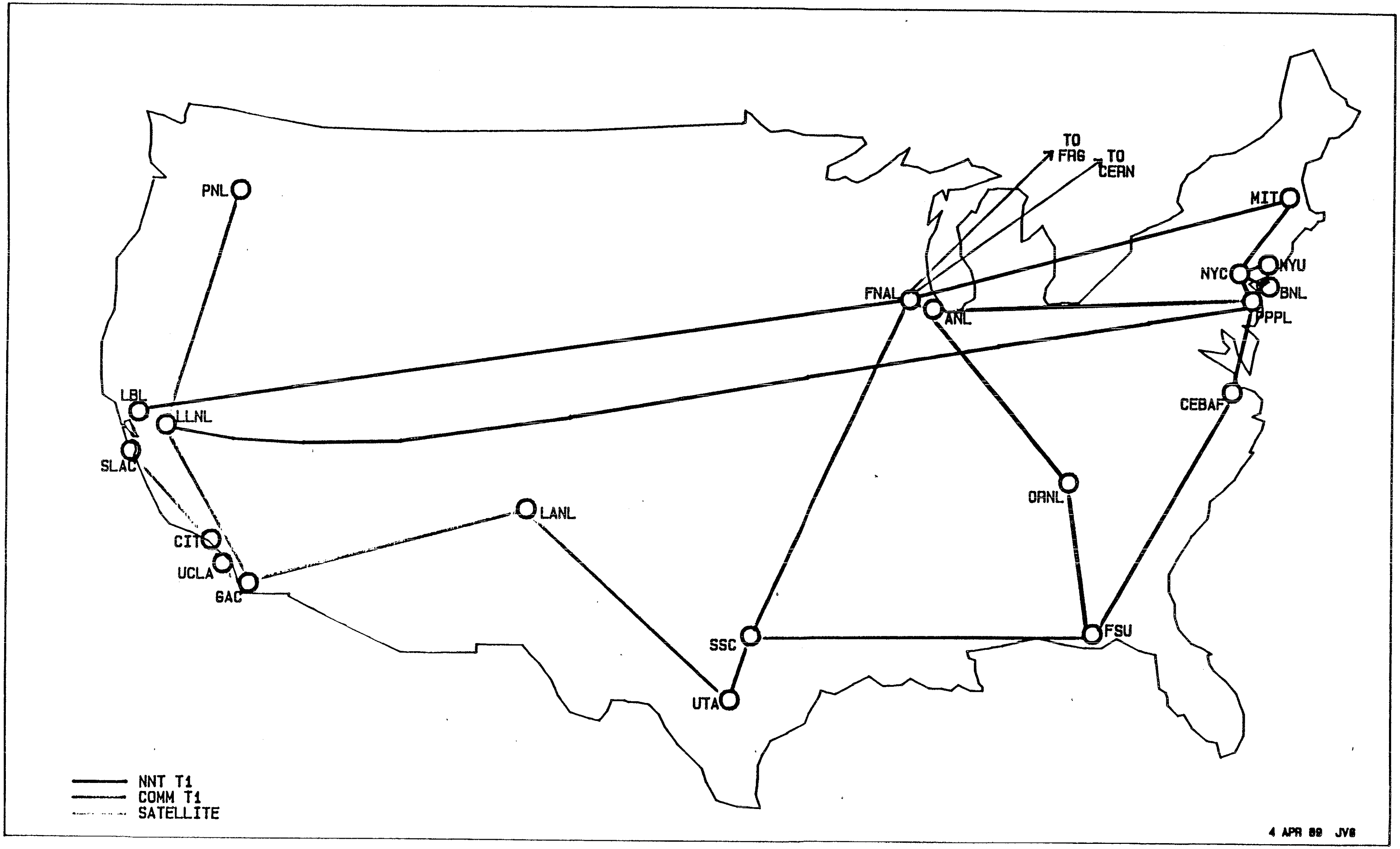
The routers have been ordered and have started to come in. We purchased cisco AGS routers after a competitive bid. They will directly handle IP, DECnet, and Appletalk traffic, although the Appletalk capability will of course not be enabled. The network will be monitored using a commercial or public domain SNMP management system.

The link from LLNL to ITER (International Thermonuclear Experimental Reactor) in Garshing, Germany is up and running. The NNT T1 links shown in the diagram are due to be ready in August. All T1 links should be ready before the end of the calendar year. The link to Japan is being studied in conjunction with other agencies.

Overview

ESnet NMFECC

- **New network will use commercial h/w & s/w**
- **Backbone links will all be T1**
- **DECnet directly supported (not wrapped in IP)**
- **Peer level communication with NSFnet (and others)**


TO FRG
TO CERN
PNL
MIT
NYC
NYU
BNL
FNAL
ANL
PPPL
LBL
LLNL
SLAC
CEBAF
ORNL
LANL
CIT
UCLA
GAC
SSC
FSU
UTA
NNT T1
COMM T1
SATELLITE
4 APR 89 JVG

Status Report

ESnet NMFECC

- cisco routers have been ordered
    - IP
    - DECnet
    - Appletalk (for experiments only)
- ITER link up (X.25 to/from Crays, IP over X.25)
- T1 links
    - NNT links due in August
- SNMP management system
- Japan link under study

VII. Special Tours and Presentations

Demonstration of a 16th Century Machine

Reported by Ole Jacobsen

Over 100 IETF members met in the organ loft at Stanford's Memorial Church to hear a short lecture on organ mechanics and tuning, followed by a "stop tour" and several musical examples. The group had the opportunity to see the inner workings of the Fisk instrument as well.

Stanford has two pipe organs, the 1901 Murray Harris organ—a rare survivor of the 1906 earthquake,—and the 1984 C.B. Fisk instrument. The Fisk instrument was built in consultation with Harald Vogel, Manuel Rosales and Herbert Nanney by the Fisk Company of Gloucester, MA. The 4-manual instrument of 73 ranks and 4422 pipes contains a special lever, used for the first time in the history of organ building, which switches the tuning of the organ from one-fifth comma meantone (appropriate to seventeenth-century music) and to a well-tempered system (appropriate to eighteenth-century music). Five extra pipes in each octave of every rank makes this change in tuning possible. The Brustpositive, however, is fixed in meantone tuning and contains two extra keys per octave: D sharp placed above E flat, and A flat above G sharp.

The eclectic design of the instrument is notable not only for its dual temperments. The organ contains stops and choruses borrowed from north German instruments, and from instruments of the French Classic tradition. No other instrument in the world provides organists with such a wide range of possibilities without the usual compromises in historical accuracy. The Fisk-Nanney organ reflects Stanford's longstanding commitment to performance and research in early music.

Pieces Presented:

- From "Livre D'Orgue, Suite Du Premier Ton" by Louis-Nicolas Clerambault (1676-1749): Recit de Cromorne et de Cornet Separe en Dialogue, and Dialogue Sur les Grands Jeux; performed by Roy Stegman
- Preludium in C, BuxWV 137 by Dietrich Buxtehude (1636-1707); performed by Roy Stegman
- A piece by Vaughan Williams (1872-1958) (on the Murray Harris organ); performed by Robert Bates
- Prelude and fugue in G major, BWV 541 by J.S. Bach (1685-1750); performed by Annette Richards

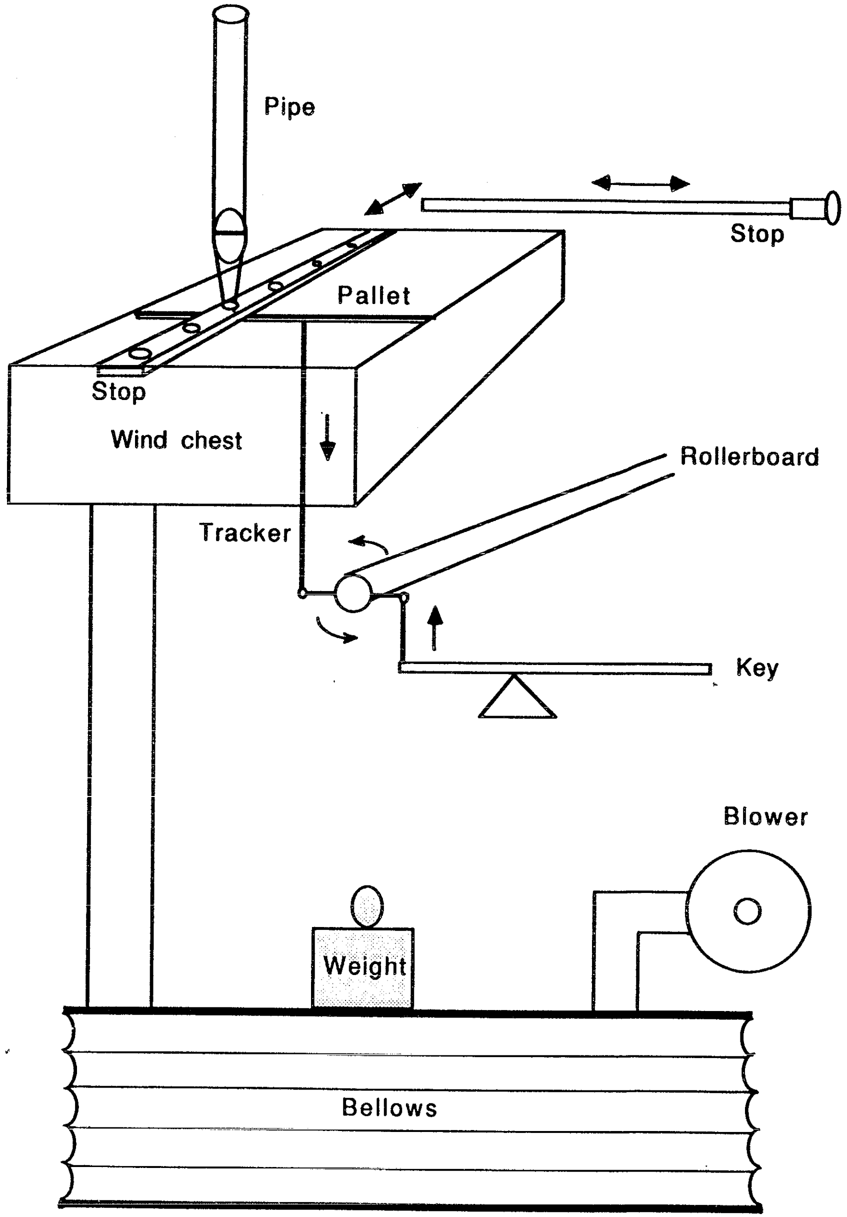
About the performers:

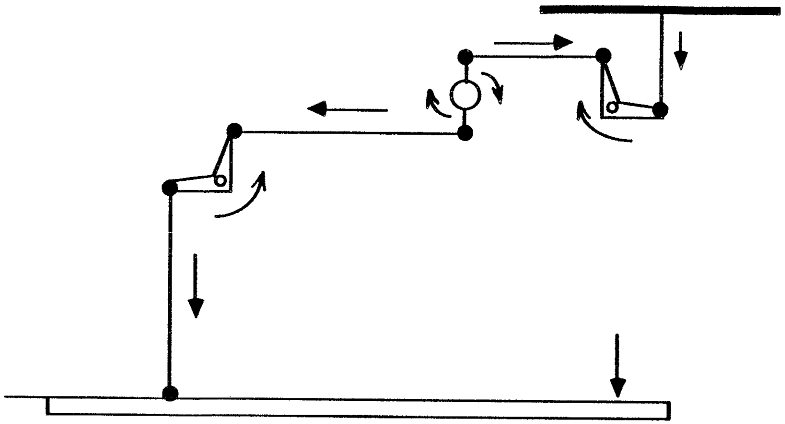
ROY STEGMAN is currently studying with Annette Richards, James Dawson and Kimberly Marshall at Stanford University. Roy can be heard playing the prelude and postlude for Stanford Memorial Church's Catholic service, every Sunday afternoon.

ROBERT BATES is Associate University Organist at Stanford and a lecturer in the Music Department, where he received his Ph.D. in musicology in 1986. He also holds degrees from Wayne State and Southern Methodist Universities. Between 1977 and 1980, he studied performance with Marie-Claire Alain, and improvisation with Daniel Roth in France. Dr. Bates has won several international competitions. He is currently continuing his research in seventeenth-century French organ music, aesthetics and the history of music theory.

ANNETTE RICHARDS is a graduate student in the Doctor of Musical Arts program at Stanford. Prior to coming to Stanford she was an organ scholar at Corpus Christi College at Oxford University, where she received her undergraduate degree. While organist and choirmaster at Corpus Christi, she was heard as organist in the BBC adaptation of Dorothy L. Sayers's Gaudy Night, which was broadcast as part of the PBS "Mystery" series in this country. She has studied organ with James Dalton, Harald Vogel, Bernhard Lagace, Christopher Stembridge and others. At Stanford she is studying with fellow Oxford alumna– University Organist Kimberly Marshall. She is a member of the Royal College of Organists. (The day after the IETF demonstration Annette left the US to spend a year in Holland studying performance practice with J. Van Ortmessen in Amsterdam; we shall miss her!)

Organ Mechanics (simplified)





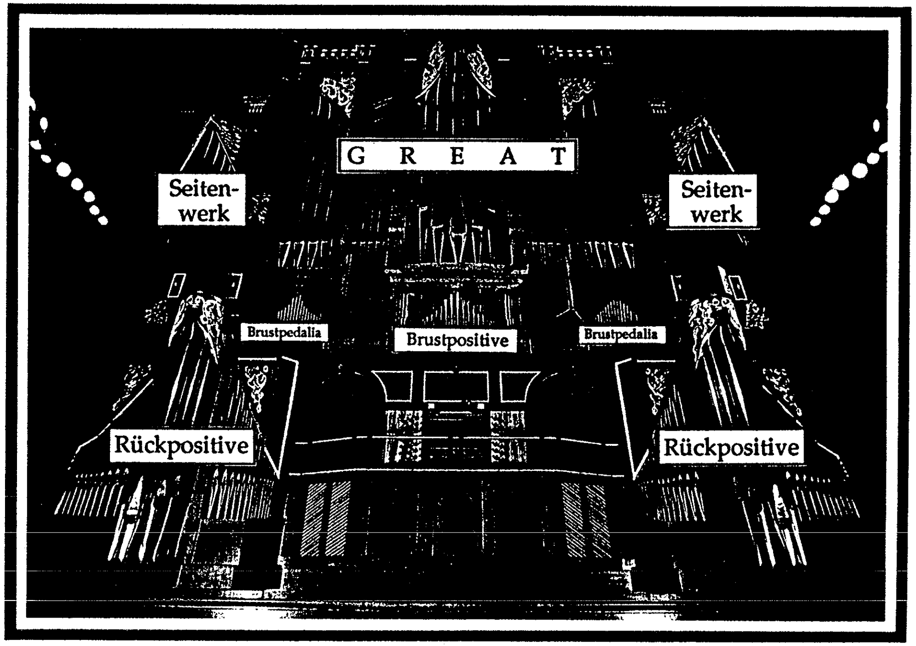
The Charles B. Fisk instrument

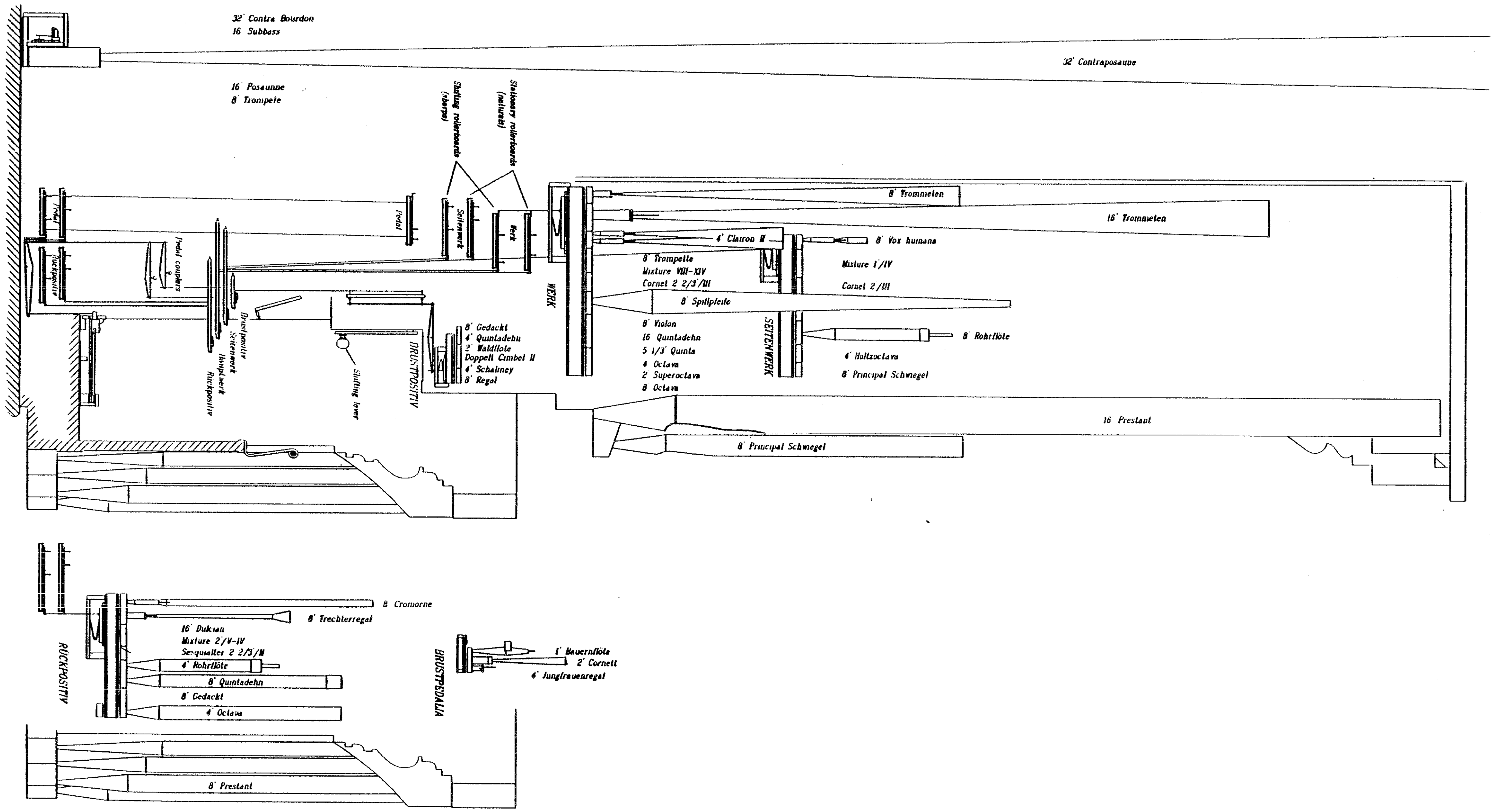
- 4 manuals/divisions:

Great
Rückpositive
Brustpositive
Seitenwerk

\+ Brustpedalia and Pedal

- 4422 pipes, 60 stops
- Unique Mean—Well shifter
  (17 pipes per octave [not 12])

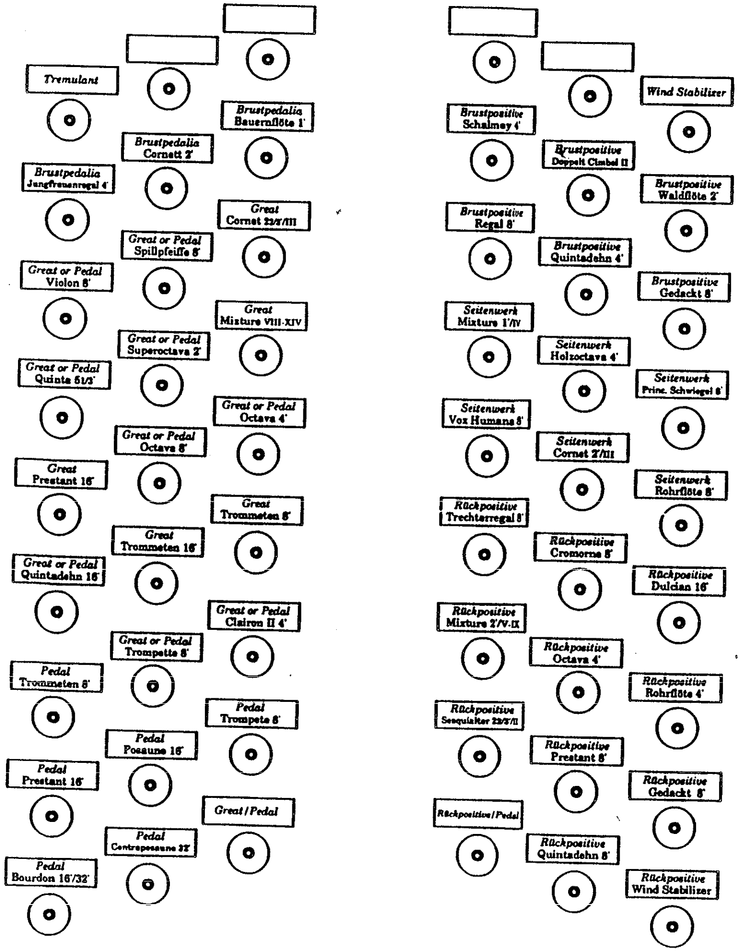




MEMORIA
STANFORI
C.B. Fisk

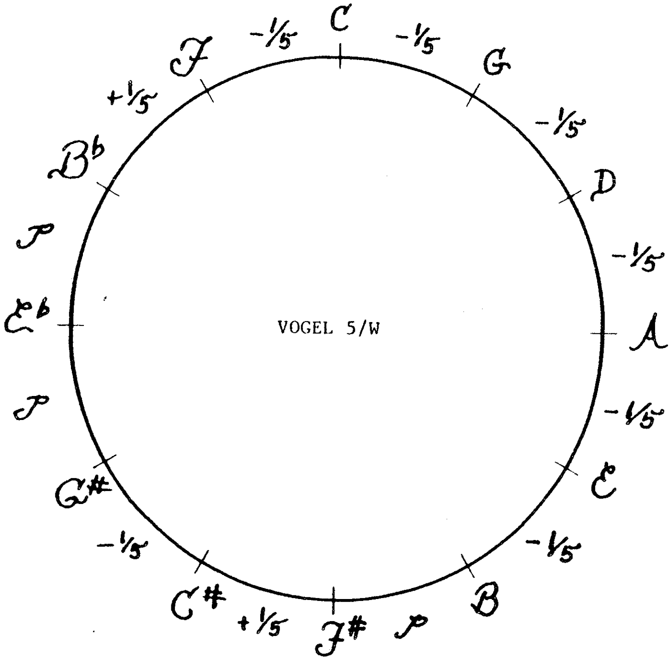
A Tour of the Stops

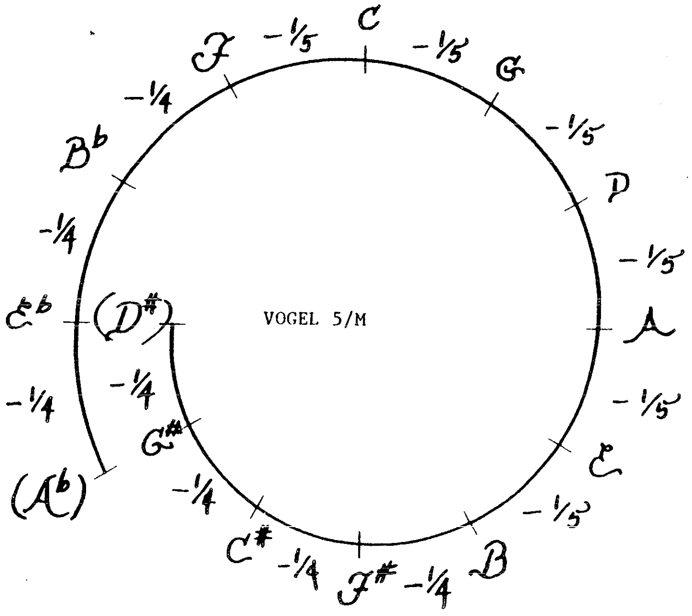
- The Principal Chorus
- Mutation Stops (and Mixtures)
- Solo Stops: Flutes and Reeds



Charles B. Fisk, Opus 85, 1984
Memorial Church, Stanford University

THE STANFORD TUNINGS

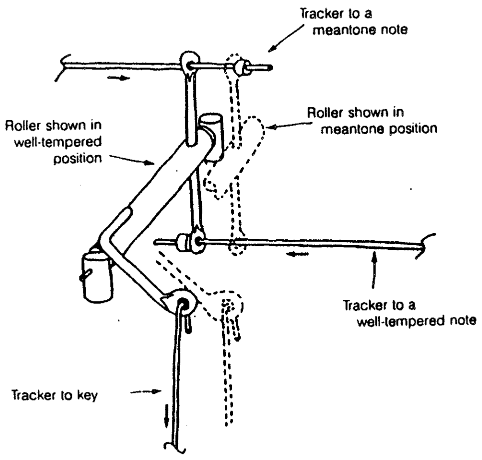




-1/5 = -1/5 Pythagorean comma
+1/5 = +1/5 Pyth

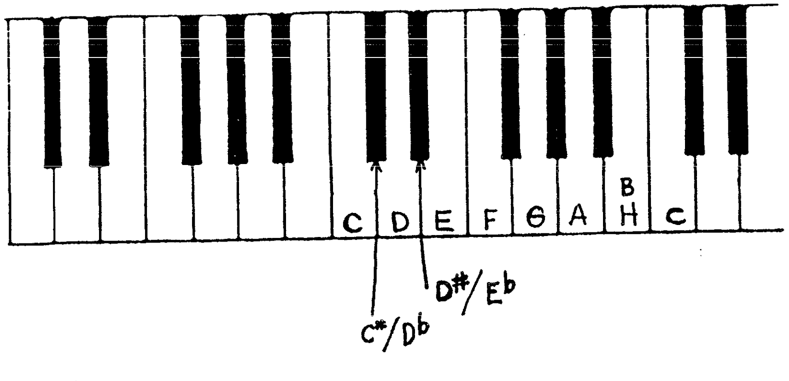
-1/4 = -1/4 syntonic comma

```
Temperament VOGEL 5/M

      PRIME-  SEVENTH-----   FIFTH------   MAJ.THIRD--   MIN.THIRD--   MAJ.SECOND-
C       0.00  1086.3  -2.0   697.3  -4.7   389.1   2.7   309.6  -6.1   194.5  -9.4
G     697.26  1085.6  -2.6   697.3  -4.7   389.1   2.7   308.9  -6.7   194.5  -9.4
D     194.53  1084.9  -3.3   697.3  -4.7   388.4   2.1   308.2  -7.4   194.5  -9.4
A     891.79  1084.3  -4.0   697.3  -4.7   387.7   1.4   308.2  -7.4   194.5  -9.4
E     389.05  1120.5  32.3   697.3  -4.7   387.0   0.7   308.2  -7.4   193.8 -10.1
B    1086.31  1119.8  31.6   696.6  -5.4   423.3  37.0   308.2  -7.4   193.2 -10.8
F#    582.89  1119.8  31.6   696.6  -5.4   423.3  37.0   308.9  -6.7   193.2 -10.8
C#     79.47  1120.5  32.3   696.6  -5.4   423.3  37.0   309.6  -6.1   230.1  26.2
G#    776.05  1121.2  32.9   733.5  31.6   423.9  37.6   310.3  -5.4   230.1  26.2
Eb    309.58  1084.9  -3.3   696.6  -5.4   387.7   1.4   273.3 -42.3   193.2 -10.8
Bb   1006.16  1085.6  -2.6   696.6  -5.4   388.4   2.1   273.3 -42.3   193.8 -10.1
F     502.74  1086.3  -2.0   697.3  -4.7   389.1   2.7   273.3 -42.3   194.5  -9.4
```



Temperament shifting mechanism (horizontally moving rollerboard)

```
Temperament VOGEL 5/W

      PRIME-  SEVENTH-----   FIFTH------   MAJ.THIRD--   MIN.THIRD--   MAJ.SECOND-
C       0.00  1086.3  -2.0   697.3  -4.7   389.1   2.7   294.1 -21.5   194.5  -9.4
G     697.26  1091.0   2.7   697.3  -4.7   389.1   2.7   298.8 -16.8   194.5  -9.4
D     194.53  1100.4  12.1   697.3  -4.7   393.7   7.4   308.2  -7.4   194.5  -9.4
A     891.79  1100.4  12.1   697.3  -4.7   403.1  16.8   308.2  -7.4   194.5  -9.4
E     389.05  1105.1  16.8   697.3  -4.7   403.1  16.8   308.2  -7.4   199.2  -4.7
B    1086.31  1109.8  21.5   702.0   0.0   407.8  21.5   308.2  -7.4   208.6   4.7
F#    588.27  1114.5  26.2   706.6   4.7   407.8  21.5   303.5 -12.1   203.9   0.0
C#     94.92  1105.1  16.8   697.3  -4.7   407.8  21.5   294.1 -21.5   199.2  -4.7
G#    792.18  1105.1  16.8   702.0   0.0   407.8  21.5   294.1 -21.5   203.9   0.0
Eb    294.13  1100.4  12.1   702.0   0.0   403.1  16.8   294.1 -21.5   208.6   4.7
Bb    996.09  1095.7   7.4   706.6   4.7   398.4  12.1   298.8 -16.8   203.9   0.0
F     502.74  1086.3  -2.0   697.3  -4.7   389.1   2.7   289.4 -26.2   194.5  -9.4
```



Tour of the Stanford Linear Accelerator Center
Presented by Kathy O'Shaughnessy

Quick facts about SLAC

S.L.A.C. = Stanford Linear Accelerator Center

Operated by Stanford University for the Department of Energy

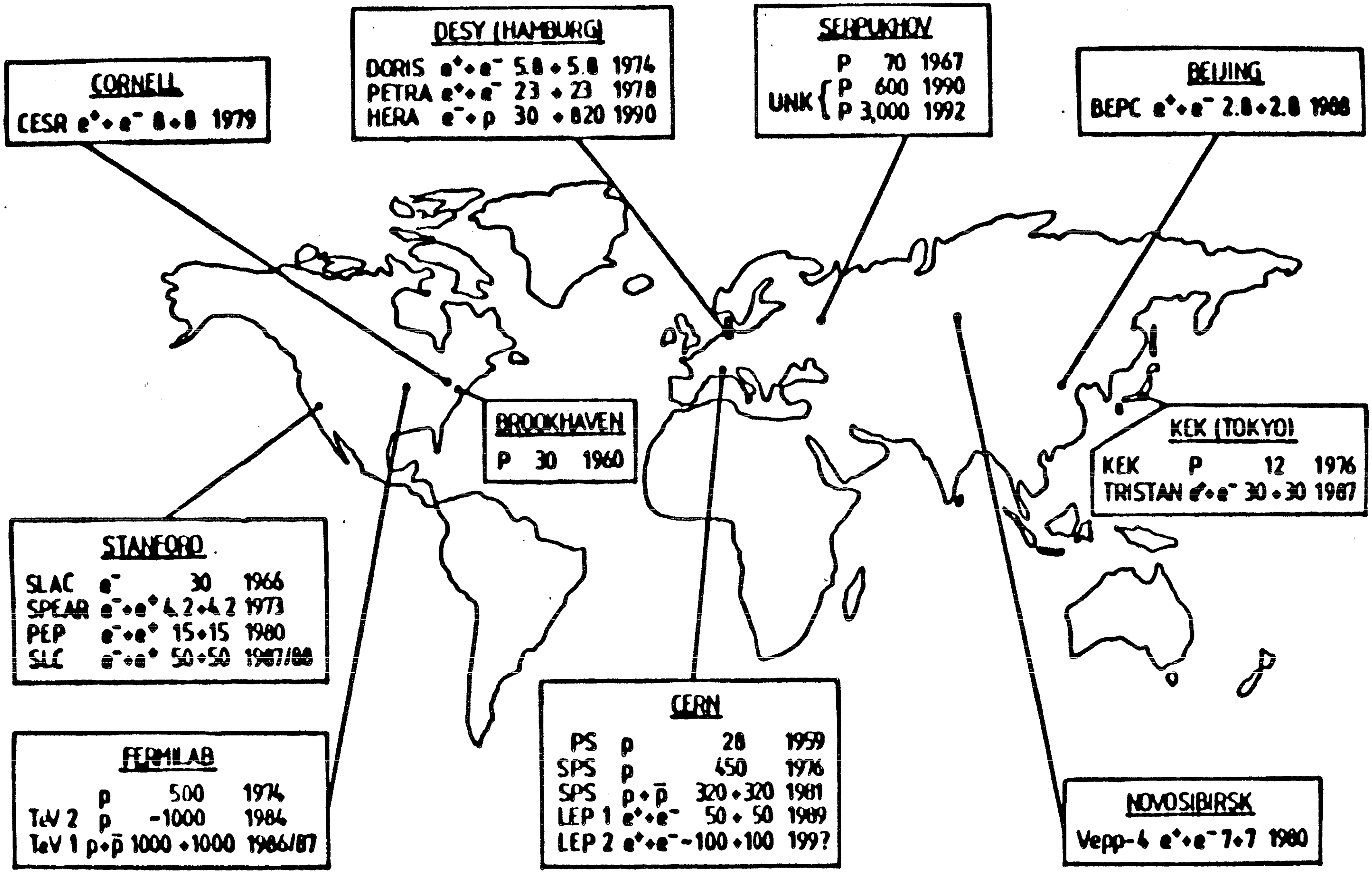
Construction began in 1962
Research began in 1966

No classified research is done here, there is a large international group of physicists who use the facility

Budget : ~ $100 million for 1988

Employees : 1350

The world of high energy accelerators



Each accelerator is defined by the particles accelerated, the energy in GeV, and the date of completion or expected completion.

What is our purpose at SLAC?

We are trying to answer two of the most fundamental questions in nature:

- What are the basic building blocks of matter?
- What holds them together?

Particles

Quarks

charge			
$+2/3$	up u	charm c	top (?) t
$-1/3$	down d	strange s	bottom b

Leptons

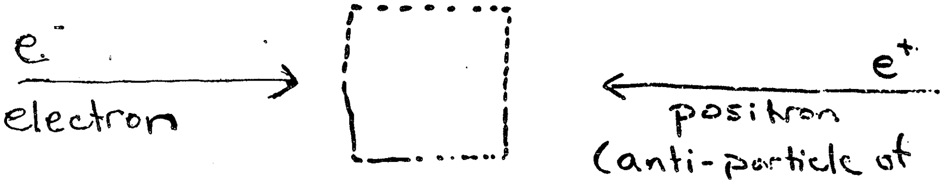
charge			
-1	electron e^-	muon μ^-	tauon τ^-
0	e^- neutrino ν_e	μ^- neutrino ν_μ	τ neutrino ν_τ

- - - - - - - - - - - - - - - - - - - -

Force carriers

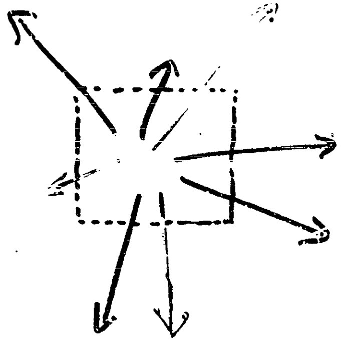
force		
gravity	graviton	
electromagnetism	photon	γ
weak	intermediate vector bosons	$W^\pm, Z^0$
strong	gluon	g

What happens in a colliding beam experiment?



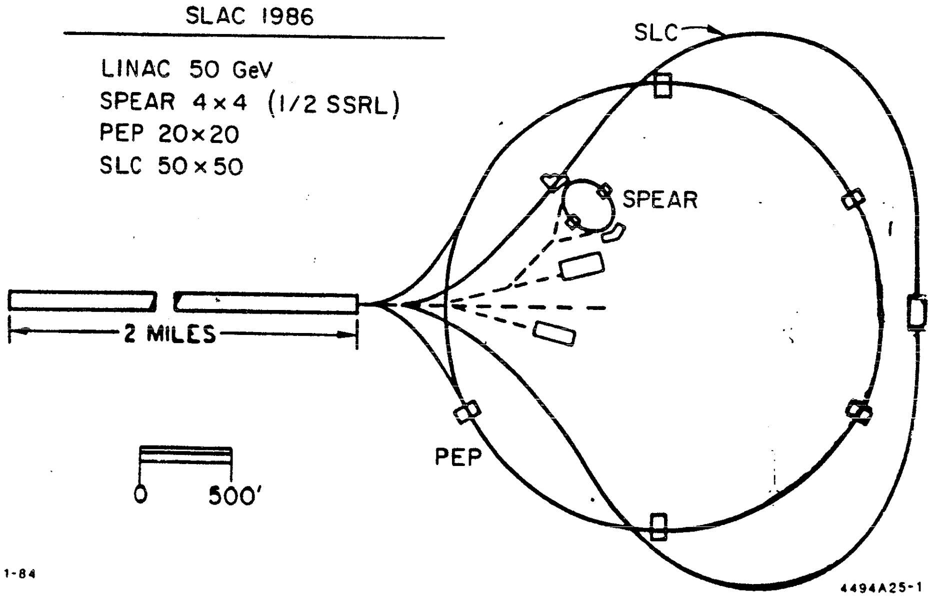
(anti-particle of an electron)

pure energy
(lasts ~ 10^{-12} sec.)



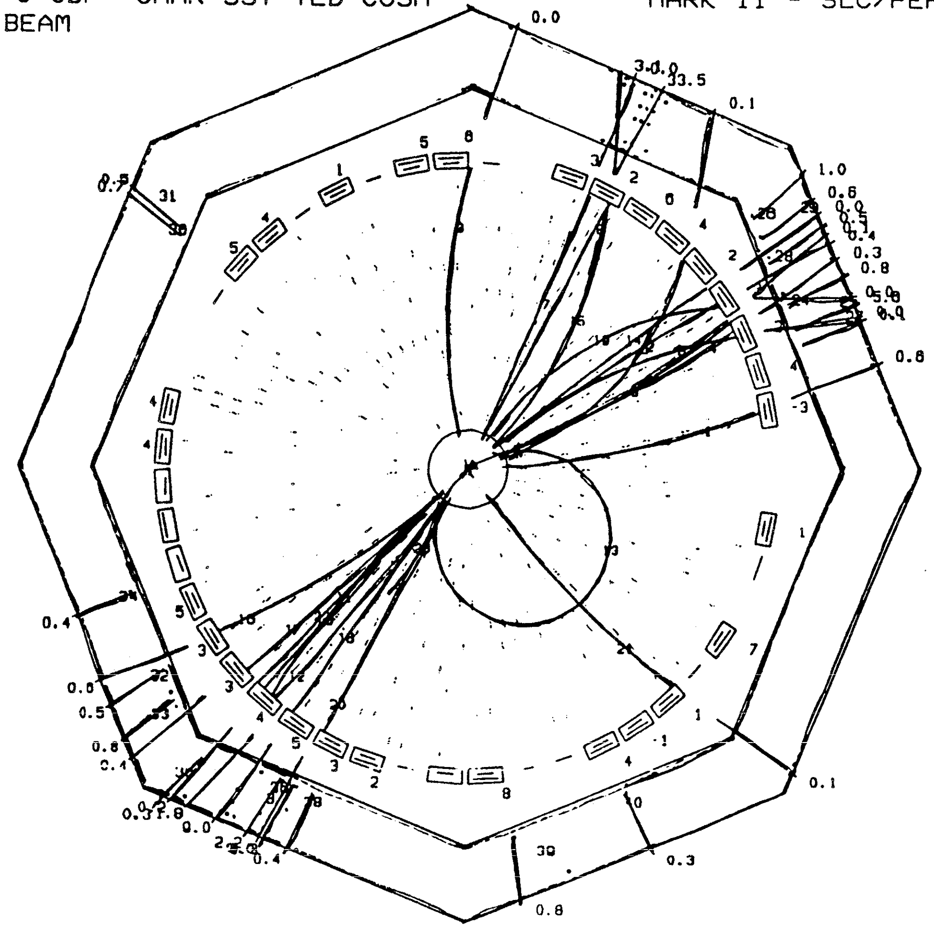
many different particles can be formed from the energy

SPEAR
PEP
SLC



RUN 17820 REC 439 E= 92.00 20 PRONG HADRON (5-0)
TRIGGER 0 0DF CHAR SST TED COSM MARK II - SLC/PEP
BEAM

TRK	P	ELATOT	ID
1	0.7	0.6	PI-
2	0.2	0.1	PI-
3	0.7	0.1	PI-
4	1.0	0.3	PI-
5	0.4	0.0	PI+
6	1.5	0.4	PI+
7	5.5	0.0	PI-
8	8.0	33.5	PI+
9	0.4	0.0	PI+
10	1.0		PI-
11	3.0	0.2	PI-
12	1.8	1.8	PI+
13	0.1		PI-
14	2.5	0.0	PI+
15	0.3	3.1	PI-
16	0.6	0.6	PI+
17	3.9	0.4	PI+
18	7.2	9.0	■■■■■
19	0.2	0.0	PI+
20	7.3	3.0	PI-
21	0.6	0.1	PI-
22	6.0		PI+
23	0.3		PI+
24		0.8	G
25		5.6	G
26		1.0	G
27		0.1	G
28		0.5	G
29		0.6	G
30		0.7	G
31		0.5	G
32		0.5	G
33		0.8	G
34		0.4	G



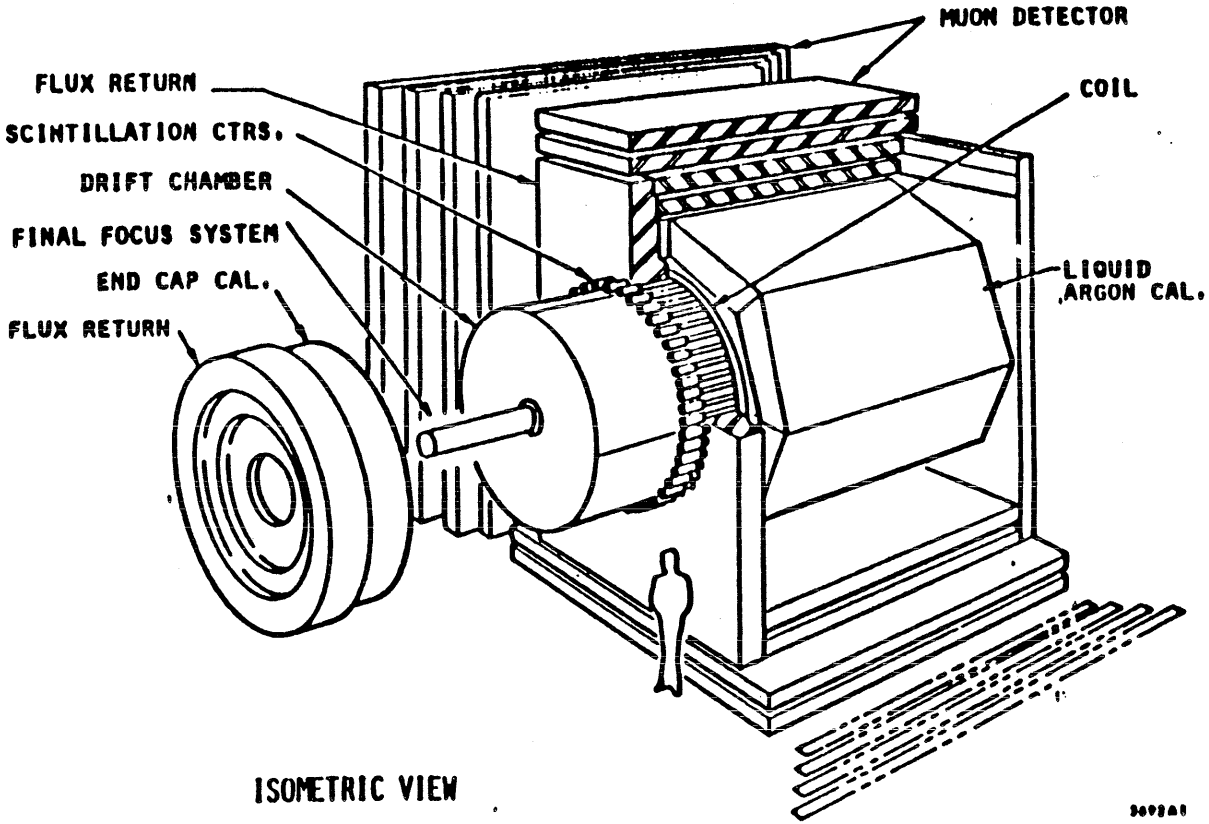


Figure 1.2 Isometric view of the Mark II detector for SLC