Proceedings of the

July 27-29, 1987

Internet Engineering Task Force

Edited by
Allison Mankin and Phillip Gross

July 1987

SEVENTH IETF

The MITRE Corporation
Washington C^3I Operations
7525 Colshire Drive
McLean, Virginia 22102

TABLE OF CONTENTS

1.0 Introduction

The Internet Engineering Task Force met at MITRE Washington (7525 Colshire Drive, McLean Virginia) for the three days of July 27 through July 29, 1987. The meeting was hosted by Ann Whitaker (head of the Protocol Engineering Group) and David Wood (head of the MITRE-Washington Network Center).

The meeting followed a new format, allocating more time to working groups. The first day and the morning of the second day were dedicated to working group meetings. One of the groups meeting on the first day was a plenary meeting of the Network Management Working Group. The afternoon of the second day and the third day was composed of technical presentations and working group reports.

Allison Mankin wrote the main body of the meeting report. Various working group Chairs contributed to the reports in Section 5. Individual contributions are noted there. Several other members, particularly Coleman Blake (MITRE), were instrumental in assembling these Proceedings.

2.0 IETF Attendees

(Note: Unfortunately, the attendance list did not include a complete listing with emailing addresses and affiliation.)

Phillip Almquist
John Anderjaska
B. Appelman
Ramesh Babu
Amatzia Ben Artzi
Mary Bernstein
Len Bosack
Hans-Werner Braun
Ed Cain
Ross Callon
Jeff Case
Boots Cassel
Stephen Castro
Vint Cerf
Hubert Chang
Mike Chernick
Noel Chiappi
David Crocker
Hassan Dastivar
Chuck Davin
John Day
Doug Elias
Robert Enger
Todd Fellela
Joseph Fowler
Peter Fuson
Marianne Gardner
Jeremy Greene
Olafur Gudmundsson
Jack Hahn
Charles Hedrick
Sergio Heker
Robert Hinden
Roxana Hoadley
Steve Holmgren
Ole Jacobsen
Van Jacobson
Mike Karels
Frank Kasterholz
Dave Kaufman
Norm Kincl
Doug Kingston

Peter Kirstein
Tam Kok
Robert Kolacki
Lee LaBarre
Anne Lam
John Lekashman
John Leong
Mike Little
Mark Lottor
Paul Love
Dan Lynch
Charlie Lynn
Louis Mamakos
Kevin Martin
Keith McCloghrie
Milo Medin
Don Merrit
Rod Merry
Lynn Monsanto
John Morgan
Donald Morris
John Moy
John Mullen
Ron Natalie
Gerard Newman
Hung Nguyen
Michael J. O'Connor
Craig Partridge
Drew Perkins
Chris Perry
Michael Petry
Susan Poh
Ed Preston
Brendan Reilly
James Robertson
Jon Rochlis
Jose Rodriguez
Jeff Schiller
Marty Schoffstall
Paul Schragger
John Shaffer
Robert Slaski
S. Soo
Weldon Showalter
Mary Stahl
David Staudt
Zau-Sing Su
Pat Sullivan

Dean Throop
James Tontonoz
Glenn Trewitt
Daniel VanBelleghem
Asher Waldfogel
David Wasley
Jil Westcott
Steve Wolff
William Yascavage
Ron Zahaui
John Zorning

3.0 Final Agenda

MONDAY, July 27th

9:00am - Opening Remarks, Local Arrangements (Phill Gross, Anne Whitaker)

9:15am - Working Groups convene in separate rooms

- EGP2 (Mike Petry, UMD/Marianne Gardner, BBN)
- Short-Term Routing (Chuck Hedrick, Rutgers)
- Name Domain Planning (Doug Kingston, BRL)
- Net Management/Gateway Monitoring (Craig Partridge, BBN/Lee LaBarre,MITRE)

For additional information on the anticipated activities of these working groups, please contact the appropriate Chair. There may be additional working groups organized at the meeting.

10:45 ~11:00am Break

1:00pm - Lunch (Scheduled late to avoid cafeteria crowds)

2:00pm - Working Groups reconvene

5:00pm - Recess until morning

TUESDAY, July 28th

9:00am - Working Groups reconvene

- EGP2 (Mike Petry, UMD/Marianne Gardner, BBN)
- Short-Term Routing (Chuck Hedrick, Rutgers)
- Name Domain Planning (Doug Kingston, BRL)

Note: Net Management/Gateway Monitoring will not be meeting on Tuesday.

10:45-11:00am Break

1:00pm - Lunch

2:00pm - IETF Plenary Convenes

- BBN Status Report (Bob Hinden/Marianne Gardner, BBN)
- NSFnet Status Report (Doug Elias, Cornell/Hans-Werner Braun, UMich)
- DDN Measurement Status (Phill Gross/Rob Coltun, MITRE)
- Gateway Monitoring/Network Mgmt Working Group Report (Craig

Partridge, BBN/ Lee LaBarre, MITRE)

5:00pm - Recess until morning

WEDNESDAY, July 29th

9:00am - Working Group Reports and Discussion

- EGP2 (Mike Petry, UMD/Marianne Gardner, BBN)
- Short-Term Routing (Chuck Hedrick, Rutgers)
- Name Domain Planning (Doug Kingston, BRL)
- and other Groups as convened

10:45-11:00am Break

1:00pm - Lunch

2:00pm - Other Presentations

- Dissimilar Gateway Protocol (Dave Mills, UDel/Mike Little, MACOM)
- Round Trip Delay Estimation (Van Jacobson, LBL)
- Landmark Routing (Paul Tsuchiya, MITRE)

5:00pm - Adjourn

4.0 Meeting Notes

4.1 Monday, July 27

4.1.1 Working Groups

The first day and the morning of the second day were devoted to meetings of the Working Groups, as well as (on the first day only) a plenary meeting of the Network Management task force. Reports from these meetings are given in Section 5.

4.2 Tuesday, July 28

4.2.1 Discussion of Long Term Routing Issues: Bob Hinden (BBN)

At Phill Gross's suggestion, a new working group will be formed to develop proposals for long-term routing solutions. Bob Hinden of BBN will chair the Open Routing Working Group. In an extra hour before the start of the IETF plenary session, Hinden moderated a discussion of the charter and some directions to take.

4.2.2 BBN Status Report: Bob Hinden, Marianne Gardner (BBN)

Bob Hinden showed a graph of the gateways peering with the core in recent months. The number will soon reach 300, the limit based on the current GGP update size limits. BBN is in the process of implementing IP fragmentation and reassembly in the core gateways, so larger updates will be handled.

In terms of EGP update sizes, BBN has already seen fragmented updates arriving. The core gateways had been truncating EGP updates that were too large for them to send, but now they will fragment them. The upshot of this is that all external gateways must now do IP reassembly.

The transition of the core from LSI-11s to Butterfly gateways is scheduled for roughly the end of the year. The mail bridges conversion is to occur about the same time. There is no reason to use autonomous system number 1 for the Butterfly core gateways.

There was discussion of the excessive cycling of routes that Dave Mills observes, and whether there is fundamental instability in EGP. BBN believes routes are flapping due to the congestion problems in the Arpanet; too many neighbors are declared down, then the routes are changed when they come back up again.

Marianne Gardner then reported on the dramatic improvements of Arpanet performance at the end of July. There was a new cross-country VSAT (Very Small Aperture Terminal) link between MIT and SRI. This had been much delayed by

circumstances, but still beat the terrestrial line which will continue to be on order. It provides a 112Kb trunk (2 parallel 56Kb channels).

Right on the heels of the new link was a change to the routing algorithm used by the PSNs. The update thresholds used before led to unstable metrics if there were long queuing delays in the PSNs. Essentially metric changes were seen too often and had a spuriously large range. The change to SPF consists of a filtering mechanism using more history, and clipping the range and the rate of change of the metrics.

There was evidence (number of performance-related traps) that the routing fix was especially helpful. A break in two major Arpanet lines and seven major Milnet lines for six hours had resulted in only a small leap in congestion once the routing fix was in. (The break in nine trunks resulted from a single fiber-optic line failure in Oakland, Calif. The discussion was lively on this point. Col. Ross Mundy explained that the DDN PMO has found that it is too expensive to buy service from diverse carriers.)

PSN Release 7 is scheduled to be installed in the Arpanet within the next weeks. It features better X.25 and the new End-to-End Protocol. The old EE Protocol will still be supported. There may be a decline in performance due to the extra code size of supporting both version, but cutover to the new will not be completed for a few months. (Note: they do not interoperate).

4.2.3 Arpanet Measurement Status Report: Phill Gross, Rob Coltun (MITRE)

Gross displayed graphs of the results of the baseline portion of the Arpanet measurements that he and Rob Coltun. The baseline measurement consisted of repeated ICMP Echo 'pings' to hosts at increasing path lengths in the Arpanet. Three different interfaces were used (X.25, HDH and 1822). Various sized packets were used and tests were conducted during different traffic density periods.

The graphs condensed much information into a three dimensional format (see slides in hardcopy version of Proceedings). The graphs showed the expected increase in median and variability of roundtrip delay over increasing path lengths. The difference between performance for long and short packets was startling. X.25 was the poorest of te three but this was not unexpected due, in part, to the current methods for interoperation with 1822.

There are are number of continued measurements planned. For example, it will be interesting to see X.25 performance under the PSN 7.0 release.

4.2.4 NSFnet Status Report: Hans-Werner Braun (UMich), Doug Elias (Cornell)

Hans-Werner Braun first spoke briefly on an issue he sees rising from the recent NSF solicitation for new NSFnet sites. The solicitation specifies that the backbone will go to T-1. The transition could take place as early as mid-1988. Other nets such as the NASA Science Internet are planning for T-1. The issue is what relative significance the ARPANET will be when NSFnet adds so much capacity.

Several people asked about the status of the big expansion of the ARPANET that NSF has paid for. Braun said that the installations were very delayed, so that the NSFnet and regional nets had not been able to wait for them. The orders for new PSNs have not been cancelled.

Other points raised in the discussion included: there is already more capacity in the Eastern half of the NSFnet now than in the whole ARPANET. The ARPANET expansion to link to the NSFnet is needed because many of the new NSFnet hosts want to talk to DDN hosts. Mike St. Johns said that collapsing multiple hosts' 56Kb lines into one T-1 had been considered, but there were doubts about reliability. Summing up in this area, Braun said that connectivity at one site affects others, so interconnect engineering is critical.

The NSFnet backbone now sees 63 nets. Congestion problems are beginning to appear due to having only two cross-country trunks, but with much less traffic than the ARPANET, so far the problems have been much less. Mills installed a fair preemption algorithm in the fuzzballs which dramatically relieved congestion. Mills and Braun are writing a paper on this research.

There is a need for more monitoring, but reluctance to load the working switches with tasks like keeping a traffic matrix.

Doug Elias presented the current system of monitoring the backbone. A central station polls each interface of each switch once an hour, with the poll and response using UDP. It would be possible to poll every fifteen minutes, if this did not use too much overhead. Discussion suggested that the increased polling would be of interest, and that based on experience with HMP, it would not waste bandwidth.

The statistics gathered include a count of preemptions. The the maximum values of preemptions (bursts) give a measure of congestion. It appears that the NSFNet does not yet experience severe congestion.

A plot of the total packets carried each month by the NSFNet backbone showed an exponential increase over nine months. There were comments about how this worked; NSFNet grows differently from the ARPANET in that additions to the NSFNet tend to be already established large networks.

Nine months of data have been collected, totalling about 7 Mbytes, at 4K bytes per day. Further information and copies of the data are available from elias@tcgould.tn.cornell.edu.

A final discussion centered on the following analysis offered by Mills: The Mail Bridges drop 3.9% of packets on average. The busiest of these gateways switches 7 Mpackets per week. The busiest of the NSFnet gateways switches 4-5 Mpackets per week. The NSFnet gateways currently have an average drop rate of 0.08%. This shows that NSFnet has excess capacity. Can we predict when the NSFnet capacity will be used up, based on our experience with gateways in the DDN?

4.3 Wednesday, July 29

4.3.1 The Simple GW Monitoring Protocol: Marty Schoffstall (RPI), et al

Marty Schoffstall of Rensellaer Polytech and NYSERNET gave a presentation on the Simple Gateway Monitoring Protocol. The work is a collaboration among groups at Cornell, University of Kentucky, Proteon and RPI. There is a draft RFC of the protocol specification. Two implementations each of the gateway-resident module and the NOC module have been fielded.

SGMP differs from the High Level Entity Management system (HEMS) in that it places most complexity in the NOC instead of in the monitored entity. Like HEMS, it formulates queries using ASN.1 data representation.

The goals of the SGMP project are to gain experience in gateway monitoring and in the production of multiple interoperable protocol implementations. SGMP is a "concept prototype."

The SGMP RFC has been submitted to the RFC Editor. In discussion, it was suggested that the RFC should be expedited, so that it can be considered alongside the HEMS RFCs. Chuck Davin of Proteon stated that Proteon intends to follow Internet consensus on gateway monitoring. It may be that SGMP will be a transitional protocol.

4.3.2 A Plea from Vendors: Dave Crocker (TWG)

Dave Crocker of The Wollongong Group presented a short list of concerns of the vendor community. He prefaced it with a question as to whether the IETF saw itself as doing research or engineering? If the former, the vendors have relatively little interest, but if the latter, the IETF must understand that vendors' engineering is focused on making products.

A case in point is that vendors are being forced to implement two network management protocols now, one for the TCP-IP world and one for ISO. Vendors are in the position that most customers think ISO is here. This means that it is difficult to justify expenses for TCP-IP products, so there can be only one shot at a TCP-IP network management product. Coding of commercial products is generally a slow process.

Some projects which the IETF could undertake in support of vendors:

1. Protocol Feature Checklist. This would realistically document the options and non-optional features a protocol implementation must have. RFPs include these already, but they tend to reflect poor knowledge of the protocols. Discussion of this turned into a "Rat's Nest" having to do with protocol conformance in general.

2. Implementation Details. This would list in an official manner protocol points such as silly window avoidance, on which there is consensus beyond the specifications. Again, RFPs need this

information, but customers have difficulty obtaining the facts.

3. LAN Login Security. This asks for a Telnet option or mechanism for encrypting the login password. Several in the audience panned this on the grounds that Ethernets can't be secure. But many agreed that there useful remedies by Telnet to the common situation where a PC or workstation owner intercepts root passwords of all machines using the Ethernet.

4.3.3 The International Internet: Peter Kirstein (UCL)

Peter Kirstein of University College London spoke about a gap in expectations about the DDN Internet between the U.S. and Europe. Each European country identifies one person to be responsible for the DDN in that country. This person (Kirstein in the U.K.) deals as well as possible with all problems of connection. Since many European networks have gateways to the DDN, the number of problems is large. The common problems are different from those familiar to the IETF. For instance, RTs are generally long due to routing over SATNET.

The responsible person serves on the International Collaboration Board. DARPA and DCA participate in the ICB, but U.S. attendance to its meetings has been spotty. There is a need for some centralization of the U.S. networks. In particular, the planning of new transatlantic links has become "alarmingly uncoordinated."

Discussion pointed out that the U.S. is very different from Britain. Britain has a strong tradition of central administration of computer networking. Up until recently, there were mandatory protocols (JANET) for government procurement. The transition to ISO in Britain will be able to rely strongly on central administrative means, such as the Name Registration Scheme.

4.3.4 Working Group Reports and Discussions: Chairs

Doug Kingston, Marianne Gardner and Charles Hedrick summarized the actions and conclusions of their working groups. Due to his travel plans, Craig Partridge gave his report the day before. See Section 5 for the reports.

4.3.5 Landmark Routing: Paul Tsuchiya (MITRE)

Paul Tsuchiya from MITRE gave a presentation of his routing research, called Landmark Routing. Landmark Routing is designed to operate in arbitrarily large networks with changing topologies. Landmark Routing automatically responds to any topology change by (when necessary) redefining the routing hierarchy. This hierarchy, called the Landmark Hierarchy, is different from the well-known area hierarchy, and is much easier to manage dynamically. The main benefits of Landmark Routing, then, are durability (in the face of topology changes), and automatic configuration (addresses are not known in advance of configuration).

Since this was the first presentation of Landmark Routing, only the basic concepts of Landmark Routing and some of the research results were presented. The research includes simulations that show that Landmark Routing is comparable in performance to the area hierarchy, in terms of routing table sizes and path lengths.

A technique has been developed for binding non-changing names to changing addresses is response to hierarchy changes. This technique, called Assured Destination Binding, very efficiently accomplishes this binding in fully distributed fashion.

Other features of Landmark Routing, include administrative zoning, and dynamic hierarch managment techniques. There was not enough time to detail them or to discuss implementation and transition issues. These will be presented at future meetings.

4.3.6 Round Trip Delay Estimation: Van Jacobson (LBL)

TCP, given half a chance, will become self clocking and regulate the packet transmission rate to the capacity of the slowest link in the path. The conditions that need to be met are:

1) an acknowledgement strategy that does not distort the timing information (which is derived from the arrival times of the ACKs) by delaying or concentrating the ACKs.

2) the ability to probe the path and determine the capacity of the slowest link (i.e., to get the clock started).

3) conservative round-trip-time (RTT) estimation.

The first topic was mentioned at the February and April IETFs. The last two topics as well as a new retransmission algorithm were discussed at the July IETF.

TCP Slow Start

Most current TCP implementations start by sending a full window of data. If the gateway input buffer cannot accommodate a full window or is already partially full, this will cause the gateway to overflow and start a stable cycle of transmit-overflow-retransmit-overflow. This results in low throughput for the sender, wastes network bandwidth on retransmissions and prevents both round trip time estimation and the use of ack arrival times to regulate the flow of the data.

The slow start implementation starts with a window size of one packet and increases the window size in response to each ACK received. This generally prevents the overflow-retransmit cycle and gives the "clock" a chance to establish itself.

The original slow start algorithm (increment by one packet on each ack) doubles the window size each round trip time, leading to an exponential increase which works well for small window sizes (up to roughly 4 KB or 8 packets) but quickly overwhelmed the gateway if the window was sized appropriately for a satellite connection (e.g., 16 to 64KB).

The latest slow-start algorithm opens the window exponentially until it is a 1/2 of the size that caused the last overflow. The window is then opened linearly in response to subsequent ACKs, delaying the onset of overflow. (To achieve linear increase in window size per round trip time, the increment per ack is made proportional to 1/W, where W is the current window size. An appropriate constant of proportionality is still under investigation. The prototype implementation uses MSS^2 which results in the window increasing by one MSS packet per RTT). If a packet is dropped, half the current window size is recorded as the new threshhold for the exponential/linear transition, then the window is reduced to one packet and the process starts over.

These improvements to TCP increased the throughput on a heavily loaded SATNET link from 70 bps to 1 kbps.

Fast Retransmit: A New Loss Detection Method

This is a method of detecting packet loss in approximately one round trip time instead of the two required by the round trip time out (RTO). The method depends on the fact that the net rarely resequences packets. Thus a burst of ACKs for the same sequence number and with the same receive window probably indicate that a packet was dropped. The fast retransmit algorithm detects these bursts by incrementing a counter for each duplicate ack (zeroing the counter on any change in the ack) and retransmitting when the counter exceeds a threshhold. The packet that needs to be retransmitted is simply the one starting with the sequence number contained in the ACKs.

Tests using SATNET echo servers (so both sides of the conversation could be observed and false retransmits detected), showed that this algorithm reliably detected about 80% of the single packet losses in one RTT and never sent an unnecessary retransmission (even though the SATNET path being tested frequently reordered packets).

Experiments have shown that the window should be closed on this type of retransmit but not down to one packet (as is done for a timeout retransmit). An appropriate amount to close the window is still under investigation. The current prototype closes to half the window size at the time of the loss.

Improved Round-Trip Time Estimation

Measuring round-trip times (RTT) allows us to probe the state of the Norton equivalent queue (the series equivalent representation of the network). Changes in RTT imply changes in queue length and bandwidth. By using this information, we can accurately predict whether a packet has been dropped or delayed and whether drops are due to damage (bit errors) or congestion.

Three RTO estimators were discussed. The first, the current (RFC 793) RTO estimator, was shown for comparison purposes.

The second model estimated RTT as RTT(n) = a RTT(n-1) + b where a and b are recursively estimated by a linear regression using each RTT and its predecessor.

The third model used recursive estimates of both the mean and variance of the RTTs to calculate RTO.

While both new models out-performed the RFC 793 algorithm, the second model was more accurate than the third model but required more computations. The advantage of the second model is that it allows us to estimate the bandwidth (1/b) and utilization (a) of the limiting link in the path.

Prototype TCP Available For Beta-Test

A tcp incorporating most of these algorithms has been developed by Mike Karels and Van Jacobson. It should be possible to run this tcp with any 4.3bsd or 4.2bsd Unix system. The tcp is available via anonymous ftp from lbl-rtsg.arpa (Internet host 128.3.254.68 or 128.3.255.68), file xtcp.tar.Z. Van Jacobson (van@lbl-csam.arpa or van@okeeffe.berkeley.edu) would be very interested in reports of IETF experience, good or bad, with this tcp.

4.3.7 DGP And Other Issues: Dave Mills (UDel), Mike Little (MACOM)

Dave Mills and Mike Little gave a presentation on the current status of the Dissimilar Gateway Protocol design and prototype implementation. A detailed RFC is under review now by the Autonomous System Task Force. MACOM is in the process of modelling DGP.

Mills started out with an announcement. The IAB intends to revive INARC (the Internet Architecture Task Force) in the form of a workshop. Its topic will be the next generation of IP networks. Its proceedings will be published in ACM Computer Communication Review. The membership will not be large, but those interested may contact him.

5.0 Working Group Reports

On Monday July 27th and Tuesday July 28th, the following groups met:

Group	Convened by:
- Name Domain Planning	Doug Kingston (BRL)
- EGP Enhancements	Mike Petry (UMd)/Marianne Gardner (BBN)
- Short-Term Routing	Charles Hedrick (Rutgers)
- Network Management	Craig Partridge (BBN)/Lee LaBarre (MITRE)

This section reproduces the combined reports from these working group meetings (some previously distributed by electronic mail).

5.1 Name Domain Planning

Convened by Doug Kingston (BRL)
Reported by Doug Kingston (BRL)

Participants:

Doug Kingston (BRL),
Walt Lazear (MITRE)
Mark Lottor (SRI),
Louis Mamakos (UMD),
Mary Stahl (SRI)

The Name Domains Working Group met on the first day of the past IETF meeting at Mitre. We reviewed and exchanged comments on three proposed RFC's which will/have been submitted for "publishing" by the IAB as official RFCs. We finalized a proposed new resource record for the domain nameserver system, the responsible person record. Finally, on second day we held an expanded meeting to discuss and propose new root nameservers for the Internet, specifically to help out NSFNET but with the aim to provide more reliable service for all.

Walt Lazear offered up his MILNET Name Domain Transition document. It was approved with minor changes. This document describes the phases of implementation (Stone Age, Bronze Age, and Iron Age in our earlier discussions) and what is required at each stage. A proposed schedule was give subject to review and approval. It also includes pointers to other relevant documents.

Mary Stahl provided a complement document to RFC-920, the "Domain Requirements" document, titled "Establishing a Domain - Guidelines for Administrators" There were a few changes made and it was then approved. The RFC has a revised application for domain delegation and a better description of how it should be filled out and what one should and should not expect from the NIC. This should be available from the IETF directory on SRI-NIC as "admin.guide".

Mark Lottor provided a new RFC titled "Domain Administrators Operations Guide". This RFC provides guidelines for domain administrators in operating a domain server and maintaining their portion of the hierarchical database. Several changes were made and the resulting document approved for publishing. The document contains examples drawn from both Jeeves and BIND as examples. This should be available from the IETF directory on SRI-NIC as "rfc.zone".

The remainder of the first day's meeting was spent on designing the responsible person (RP) record. This was a carry over from our last meeting when Louis Mamakos initially presented the idea. We all agreeded the the basic idea was sound but we had not yet agreed on the details. At this meeting we agreed that the mailbox should definitely be part of the data in the same manner that it is provided in the SOA records. The question was how to get at more specific information such phone numbers, addresses, full names, and other data that might be useful in contacting the responsible people. We decided that the whois service was the kind of data we wanted although in some cases in a more well defined output format. We decided that the second data field in the RP record should be a whois pointer consisting of a key and a whois server host in the spirit of the format for the mail address. Louis will update his earlier RFC proposal and make it available for review to the IETF before we ask for it to be published. We expect little problems with getting it available quickly as we would like to see this in use as soon as possible.

On the second day we held a one hour meeting with a wider attendance to discuss root domain servers. In addition to the earlier attendees, we also had Steve Wolff (NSF), Marty Schoffstall (RPI) Hans-Werner Braun, and a few others. The impetus for this was the poor root nameserver service available on NSFNET and one goal of this meeting was to get some nameservers established that would provide good service to the NSFNET. We discussed and finally agreed on three new nameservers. Maryland and RPI were chosen fairly early on. Maryland was chosen in large part because it is in a position to service NSFNET, ARPANET, MILNET, and SURANET all equally well. After a bit more discussion we nominated NASA Ames and the third in absentia. Ames is an ideal location due to its connection to MILNET, ARPANET, NASA-Sci-Net, NSFNET?, and BARNET?. Milo already had one of everything else, so he was happy to take on a root nameserver too. These three servers and the server at Gunter Adam are expect to be fully operation by the next IETF meeting.

Having concluded these items, the working group has decided to dissolve. If future issues may require the formation of a similar group later, so be it.

5.2 EGP Enhancements

Convened by Mike Petry (UMd) and Marianne Gardner (BBN)
Reported by Coleman Blake (MITRE)

Participants:

Coleman Blake, MITRE
Len Bosack, cisco Systems

Marianne Gardner, BBN
Bob Hinden, BBN
Mike Karels, UCB
John Moy, Proteon
Mike Petry, Univ of MD
Jose M Rodriguez, Unisys
Mike St. Johns, DCA
Jim Tontonoz, DCA

Nomenclature

The Exterior Gateway Protocol version 3 (EGP3) is a new implementation of the current EGP which is referred to variously as "The EGP" or "EGP904". Since the current implementation uses a value of 2 in the version number field, the new implementation is given version number 3. This is a little confusing since there has only been one implementation of EGP prior to this one and a draft RFC describing an "EGP2" was circulated earlier. However, EGP3 is the replacement for the current EGP and hopefully this will damp out the start-up transient in EGP version numbering. For clarity, the current implementation of EGP will be referred to as EGP904.

Design Philosophy

The basic philosophy of EGP3 was to make the simplest set of changes necessary to solve the current urgent problems and add enough hooks so that most future changes could be accommodated. The basic problem of the current EGP implementation is that, due to growth in the Internet, the routing updates are growing too large into fit in a single packet. Since some of the gateways do not perform reassembly, the message is dropped. This problem is compounded by the fact that there is no easy way to introduce a new version of EGP into the system since EGP904 will drop any message it receives with a version number different from its own (which is 2). Since no error message is sent when this happens, it is impossible to distinguish between a gateway that does not understand a new version and one that is down.

Features of EGP3

There is one mandatory change and three optional changes needed to upgrade EGP904 so that it can interoperate with EGP3 and its successors. The mandatory change is that all EGPs implement a new error message code, code 6. This message will be sent when the gateway does not understand the version number of a received message. This is the minimum change that will allow EGP904 to interoperate with EGP3 and its successors. A version of EGP904 that implements only this change has been designated EGP2.2.

The optional changes are version negotiation, incremental updates, and combined reachability and poll messages. The first two deal with the problems mentioned above, the need to introduce new version of EGP into the system and the need to handle the

ever growing routing updates. The third change reduces overhead traffic by combining Hello/IHY with Poll/Update messages. The old Hello is now a Poll that doesn't contain data. These new features will be described briefly.

Version Negotiation

Version negotiation begins with gateway A sending a request to gateway B. Gateway A always starts the process with the highest implemented version, k, it has. Three things can happen:

1. Gateway B understands version k, in which case the exchange of data can begin.

2. Gateway B understand version n $>$ k, but not version k. In this case, Gateway B sends a code 6 error message to Gateway A using version n. Upon receipt of this message, gateway A knows that it cannot communicate with gateway B and stops trying

3. Gateway B understands version n $<$ k. Gateway B sends a code 6 error message to gateway A using version n. If gateway A understands version n, then it sends a new request. If gateway A doesn't understand it stops trying.

One additional step will be added to this procedure for the period of transition to version negotiation. If gateway A does not receive a response to a Request in version k, it sends a new request in version 2.2. This step can be dropped at the end of the cut-over period.

Incremental Updates

Incremental updating allows a routing update to be broken into several messages, making the size of an update message independent of the size of the internet. This solves the problem of the updates growing too large for a single packet and also reduces overhead since only new information is exchanged between gateways.

In addition to a message sequence number, each gateway keeps a send and receive routing update sequence number for each of its peers. These numbers are sent with every Poll/Update message and are used to calculate how much data is outstanding. The information exchange between the two gateways begins with the the exchange of sequence numbers and perhaps data with the initial Poll/Update packets. The exchange continues until all outstanding information has been sent and then ceases for a polling interval.

Hello Polling

Since there are no Hello/IHY messages any more, neighbor reachability must be determined from the Poll/Update messages. A gateway can ping a neighbor with a Poll message that may or may not contains routing data. The neighbor then responds with an Update or Poll message that also may or may not contains routing information.

This technique reduces overhead by allowing routing data to piggy-back on reachability packets. If all of the new information that two gateways need to exchange will fit into a single packet from each, then the update can be completed with two packets instead of the four required before.

Data Compaction

There is one additional significant feature of EGP3. There is no data compaction in the gateway IP address field of the update message. This reduces the processing required to uncompact the data, allows greater flexibility in routing and makes less restrictive topologies possible. However, the topology restriction is explicitly retained in EGP3 since the routing algorithm cannot resolve loops if the tree structure constraint is relaxed.

In addition to the major areas described above, the working group came to agreement on a number of technical details. These will be incorporated in the draft that will be distributed to the task force after the working group members have commented on it. The working group expects to complete its charter between now and the next IETF.

5.3 Short-Term Routing

Convened and Reported by Charles Hedrick (Rutgers)

This is a report on the Short Term Routing meetings at the July IETF. I should start this report with a list of attendees. Unfortunately, I forgot to get a list. Also, there were so many sessions that by the time we were finished, we probably had half the membership of the IETF there at one time or another. Attendance at the first sessions, where NSFnet was discussed, included at least briefly people familiar with BARnet, JvNC, NYsernet, PSC, and Suranet. Discussions of the RIP protocols had as the primary participants Noel Chiappa (who keeps assuring us that he does not represent Proteon), Mike Karels (Berkeley), and Jeff Schiller (MIT). Just so you know the extent to which representatives of existing RIP implementations were present, I note that Len Bosack (cisco) was present, but as far as I can recall, no one was present who was responsible for gated or the Ungermann-Bass routers. Dave Mills was not present either.

We began by looking at the routing problem presented by NSFnet and the regionals. It is impossible to reproduce the map here. But what we have in effect is a number (approx. 10) of regional networks, with diameters of up to 5 or 6, connected by the NSFnet backbone, the Arpanet, USAN, and a number of connections directly between sites in one regional and a site in an adjacent regional. (These were referred to as "back doors". The largest diameter appears to be Suranet, which has a diameter of about 9 if

one line happens to be down. The backbone has a diameter, when converted by gated to RIP hop counts, that can get as high as 6. After looking at these configurations, we came to the following rather obvious conclusions:

- RIP as it exists now can't be run over this whole set of networks as a single system. The effective diameter is greater than 16. Rutgers and other sites have already seen networks become inaccessible because they are more than 15 hops away.

- The system is sufficiently decentralized and uncontrolled that it would probably be unsafe to run it as a single RIP system even if RIP's metric were increased and stability problems fixed.

Based on this, we finally concluded that it would be best to think of each regional as an autonomous system, and to use EGP or something similar at the boundary between each regional and the NSFnet backbone, and also at all backdoors. There was no clear formulation of what should happen at these boundaries, but I think people have in mind roughly the following things:

- The autonomous systems should have separate metrics. Metrics are in effect "regenerated" at the boundary between two AS's.

- We need a set of rules to control what information passes where. Otherwise routing loops will occur.

Ideally, we would have some sort of meta-routing system to control the routing between AS's. A number of discussions happened during lunch and at other informal times, to see whether we could come up with a plausible system that avoided lots of manual configuration tables. These didn't lead anywhere. In my opinion, we are going to have to live with manually-updated configuration tables for some time. Some additional technical detail will be put in an appendix to this report, which will be circulated separately.

The second set of meetings was directed towards producing an RFC describing RIP. An agreement was reached with all of the implementors who chose to speak up. Note that this agreement is quite different from the conclusions of the previous IETF. Part of this is because the previous IETF envisioned a single-level RIP system covering the entire country. This would require a version of RIP that can handle larger metrics, and that is more stable than the existing one. In this meeting, we agreed that this approach would not work, and instead propose breaking the network up into autonomous systems. RIP is envisioned as (at most) the IGP within one AS. As such a metric of 16 may be large enough, and some responsiveness/stability tradeoffs will go differently. Here are the features agreed to at the meeting:

- a variant of split horizon is required. Probably the briefest term is "infinite split horizon", though my personal preference

is "split horizon with poisoned reverse". Conventional split horizon says that update messages must be calculated separately for each interface, and must omit any networks whose next hop is through the interface out which the update is being sent. Split horizon with poisoned reverse says that instead of omitting such networks, they should be included with an infinite metric. With this provision, any two-gateway loop will be broken immediately. Without split horizon, two-gateway loops get broken by counting to infinity. With conventional split horizon, there are situations where it may be necessary to wait for a timeout to get rid of the loop.

- holddowns are not included. Loops with more than two members are expected to be resolved by counting to infinity. Simple calculations show that counting to infinity is actually faster than waiting for a reasonable holddown to expire, with networks whose diameter is less than 16.

- triggered updates are required. In order to avoid meltdown, they must be delayed by a random time from 1 to 5 seconds. The randomness is introduced in order to avoid having a system with a large number of gateways on one network create collisions when there is an update.

- provisions are required to prevent the regular 30-sec updates from self-synchronizing. This will happen if the updates are triggered by a timer that is started when the previous update finishes. Implementors are required to adopt one of two approaches: (1) updates are triggered by a clock whose rate is not affected by system load, at precalculated points of time separated by 30 sec; or (2) the 30 sec has a small random time added to it.

- support for host routes is optional

- the trace command is removed

- the poll command is removed. [According to Mike Karels, poll was not part of any version of routed distributed by Berkeley. Since I was using Sun for a 4.2 source, that suggests that poll was added by Sun. There was some discussion about Sun adding a command to dump the route table. This would require that each route would include not only the metric but the gateway. Had Sun's routed done this, it was agreed that this would be included in the spec. However I just checked, and it does not. Thus I believe the agreement calls for poll not to be included at all.]

- messages with version 0 are to be discarded

- messages with version 1 are to be discarded if any of the "must be zero" fields are non-zero. This spec documents version 1.

- messages with version greater than 1 are to have the "must be zero" fields ignored. This allows implements that conform to this spec to process packets from possible new versions that may include additional data.

- messages must have the IP source address corresponding to the interface out which they are being sent.

- messages from a gateway that is not on a directly connected network are to be ignored.

- administrative controls will be suggested. They will include a list of allowed neighbors, and restrictions on networks allowed in messages sent or received.

- the draft document also suggests a provision for changing metrics of networks. This will be prohibited. Note that the routing strategy proposed for NSFnet will require such a feature. The implementors feel that this feature is so dangerous that even though it may be needed for certain applications, and thus may actually be implemented, implementing it is to be regarded as a violation of the specifications. (Mr. Phelps, if any of your updates are captured or killed, the RFC-writers will disavow any ...)

- a cautionary note will be added saying that there may be performance problems for 9600 baud and slower lines. If the entire NSFnet comes up or goes down, and somehow a loop is creating involving this list of networks, we could end up counting to infinity with very large update messages. This could create a disaster for slow lines. There is no solution to this problem for which we could reach a consensus.

I will produce an updated draft of the document, including all of these features. It will be reviewed by Mike Karels. We will attempt to produce a version that is acceptable to both of us. Should this miracle occur, the rest of the committee agrees to bow 7 times to West and accept the result.

Dave Mills asked an interesting question during the IETF plenary session. He asked in effect, "With this combination of features, it appears that RIP will not be stable. Do you believe that the version of RIP described herein will in fact be safe for use by

NSFnet? Are you worried about the fact that counting to infinity with updates containing several hundred routes will kill performance on 9600 baud lines?" My answer is that I do not believe that it is entirely safe, but I do not think it will cause any disasters, and I believe it is the strongest protocol on which a concensus can be reached. The primary issue on which there was no consensus is holddowns. I believe the situation is the following:

holddowns are implemented by cisco, the fuzzballs [not quite RIP, but in the NSFnet context they function as part of a RIP network], Ungermann-Bass [holddowns can be disabled], and gated [I think]

holddowns are not implemented by routed or Proteon

I do not believe that this RFC will change this situation. Omission of holddowns does not indicate a concensus that they are a bad idea. It is certainly possible that organizations may choose to specify them in RFP's. Thus vendors may still wish to supply them as an option. Since it is unlikely that we will be able to reach concensus on an RFC that mentions them -- even as an option -- holddowns will remain as another secret optional feature, like metric modification.

5.4 Network Management

Convened by Craig Partridge (BBN) and Lee LaBarre (MITRE)
Reported by Craig Partridge

These are the minutes for the meeting of the Network Management Working Group meeting on July 27th, at Mitre in Washington, DC.

These notes are in two parts: a synopsis of the general meeting, which took the form of a series of presentations, and then a list of the issues for which resolution was announced at the meeting.

Meeting Synopsis:

Report of the Network Management Working Group -- Lee LeBarre of Mitre and Amatzia Ben-Artzi of Sytek.

Automated Network Management (ANM) -- Jill Wescott of BBN.

ANM is a network management system which can operate on internetworks which support multiple management protocols. The system is distributed. A collection of cooperating distributed database managers, called DMM, coordinate in the retrieval and storage of monitoring information from the network. The DMMs are capable of collecting their information using whatever protocol is appropriate (provided that a collection agent for that application has been integrated into the ANM system). The information stored in the DMMs is made available to management applications using an

ANM protocol. (This means that an application can query ANM about any device using a single protocol -- ANM does any translations required). Work is also progressing in developing intelligent graphics programs to display the information stored in DMMs.

Simple Gateway Monitoring Protocol (SGMP) -- Chuck Davin of Proteon.

Chuck along with Jeff Case of U. Tenn., Marty Schoffstall and Mark Fedor of NYSERNET, have developed a simple monitoring protocol which is being implemented on several different nodes. The protocol offers a tree-shaped data space to applications, and provides facilities to do a simple, in-order, tree walk to extract data. Data is encapsulated in the ASN.1 data format.

Report of the Gateway Monitoring Working Group -- Glenn Trewitt of Stanford and Craig Partridge of BBN.

Glenn and Craig presented an overview of the High-Level Entity Management System (the system which the Gateway Monitoring Group was originally formed to oversee). The system specification has reached a roughly stable point and the four RFCs describing the system have been sent to the RFC Editor. Copies of the drafts are available on SRI-NIC in the <IETF> directory.

Craig is now working on an implementation to verify the specification. At the same time Glenn and Craig are working with the members of the core group of the Network Management Working Group to integrate suggestions for improvements to the HEMS spec. As a result of this effort, they expect that a revised HEMS spec (verified by implementation experience and reviewed by the larger audience which has access to the RFCs) will eventually be developed.

Issues:

- The relationship between the Network Management Working Group and the Gateway Management Working Group has been settled. The Network Management Working Group has decided to focus on developing functional requirements for and a service interface to management protocols. This work will be used to make suggested changes and improvements to the HEMS system (which is being done under the auspices of the Gateway Management Working Group). Trewitt and Partridge have agreed to incorporate the suggested changes, subject to implementation experience. The Network Management Working Group no longer plans to attempt to develop a CMIP-based system for the Internet, although the service interface they develop will not preclude the use of such a system.

- There were questions about the relationship between SGMP and HEMS. Proteon said several times that it views SGMP as meeting an immediate need, and that when an Internet standard solution is developed, they will follow it.

- Some people were curious about the relationship between the various management groups, inside and outside the Internet community that are springing up. There seems to be good informal coordination between the HEMS group and the ANSI/ISO communities (several members of the Network Management Working Group are on the ANSI and ISO groups -- and other members of these groups are in contact with the HEMS developers). But Peter Kirstein mentioned that he was aware of at least one other activity ("TTP"?), and it was pointed out that we don't seem to have any contact with the IEEE standards bodies.

- It was decided that a recommendation be made to the RFC Editor that the drafts of the SGMP and the HEMS RFCs be issued as official RFCs.

6.0 Presentation Slides

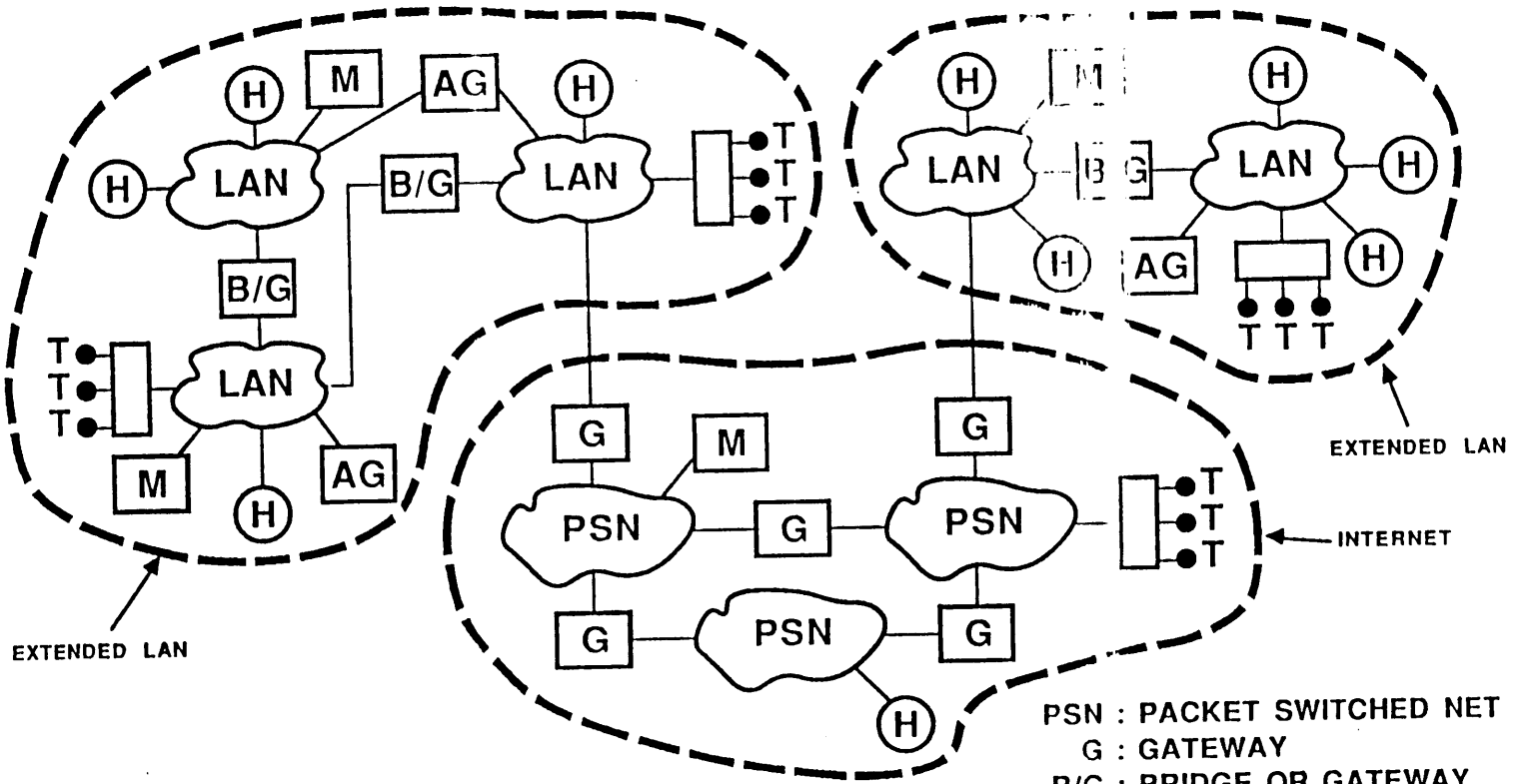
This section contains the slides for the following presentations made at the July 27-29, 1987 IETF meeting:

- TCP/IP Network Mgmt	LaBarre (MITRE), Ben-Artzi (Sytek)
- Simple Gateway Mgmt Protocol	Davin (Proteon)
- Automated Network Management	Wescott (BBN)
- High-Level Entity Mgmt Sys	Partridge (BBN), Trewitt (Stanford)
- Long Term Routing Issues	Hinden (BBN)
- Arpanet Status Report	Gardner (BBN)
- Arpanet Performance Measurement	Gross (MITRE)
- NSFnet Status Report	Braun (UMich), Elias (Cornell)
- SGMP	Schoffstall (RPI)
- A Plea from Vendors	Crocker (TWG)
- The International Internet	Kirstein (UCL)
- Landmark Routing	Tsuchiya (MITRE)
- EGP Wkg Group Report	Gardner (BBN)
- Round Trip Delay Estimation	Jacobson (LBL)
- Dissimilar Gateway Protocol	Little (MA/COM)

Presentation Outline

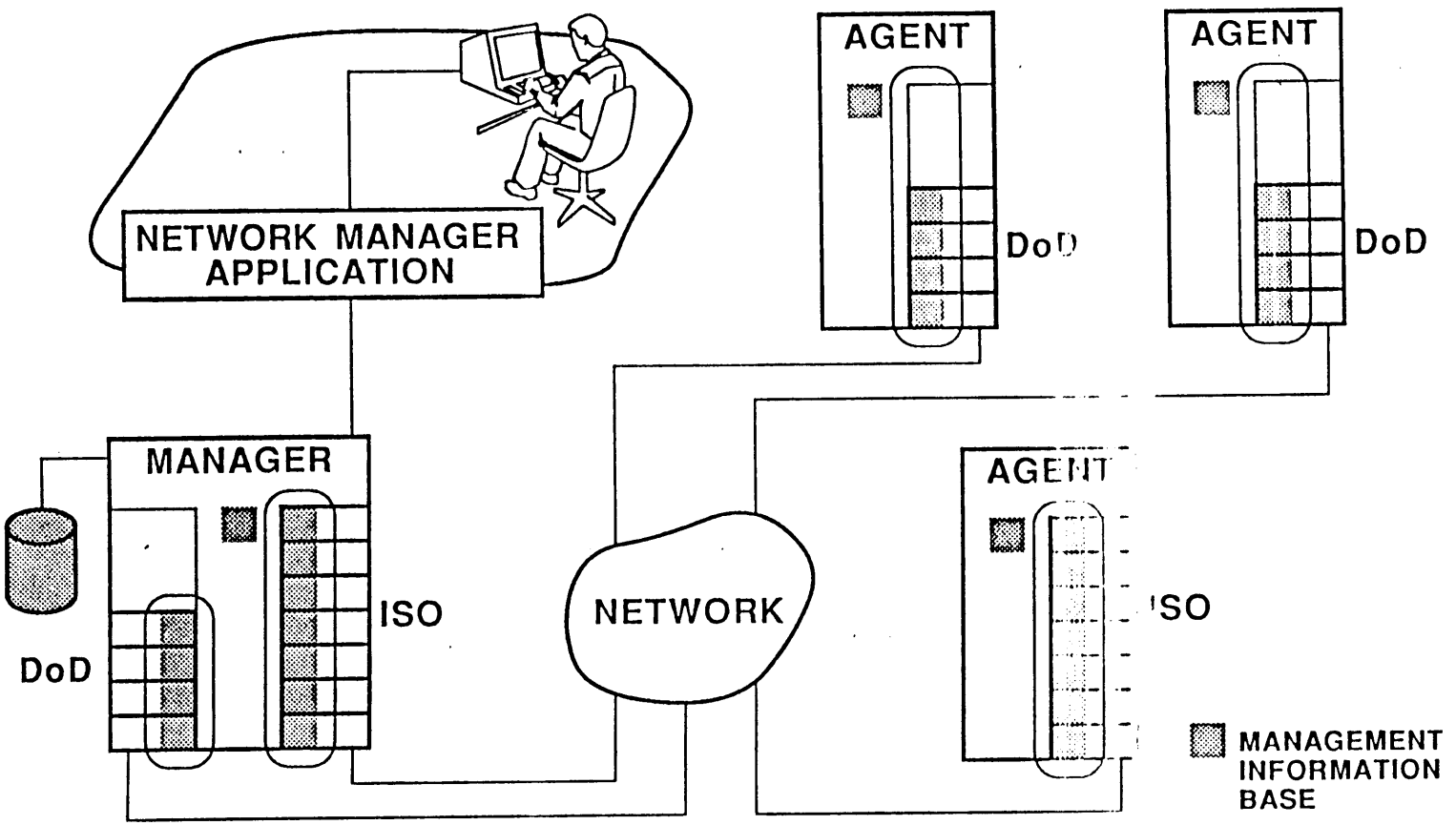
- Working group goals and scope
- Approach
- Solution
- Status

Environment

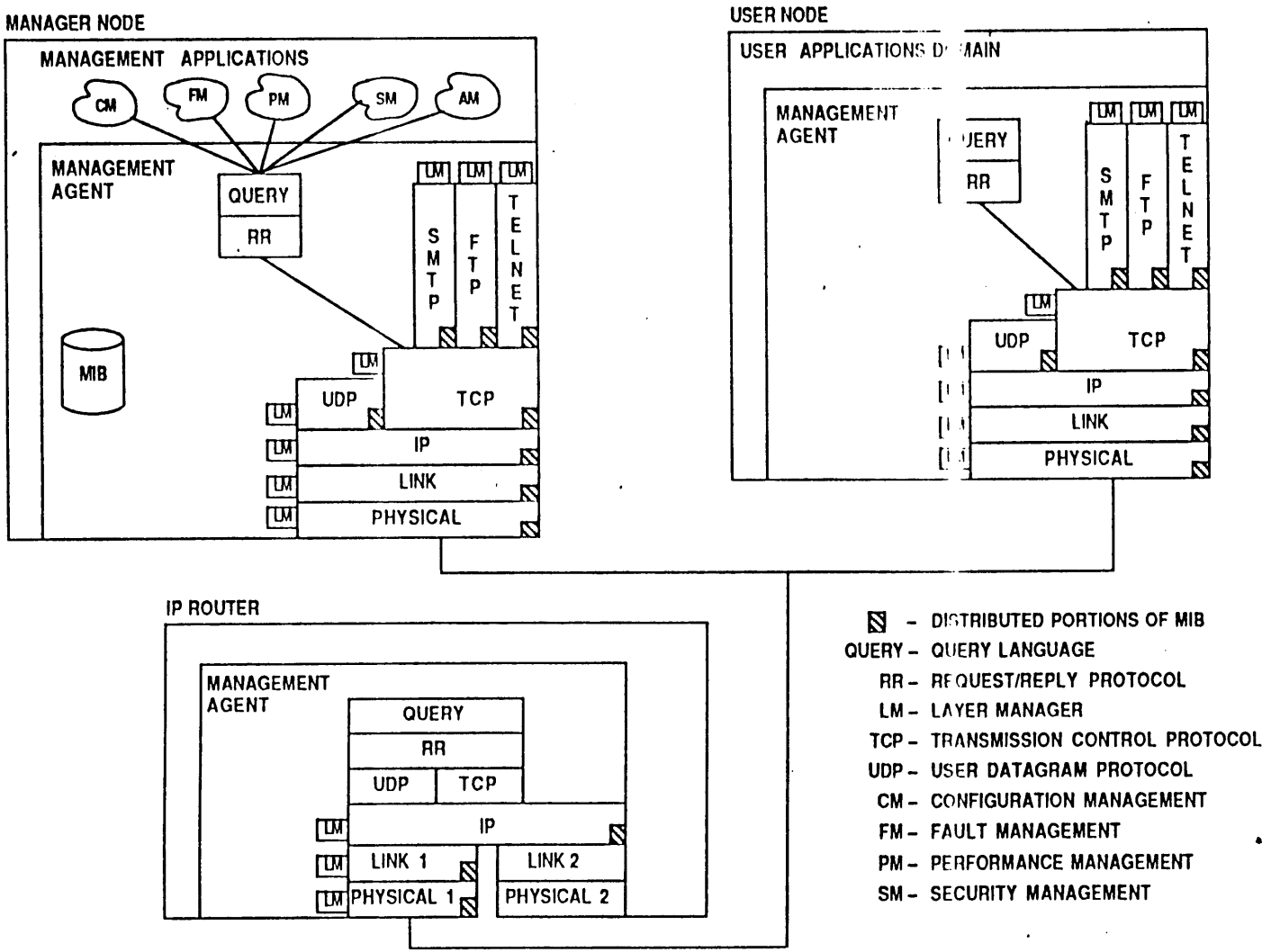


MITRE

Integrated ISO/DoD Network Management



DoD Protocol Management Architecture



MITRE

Specific Working Group Objectives

- Manage end-systems and intermediate systems that can be identified with IP address
- Manage systems without an IP address through agent by proxy - agent monitors/controls IP-less nodes with proprietary protocol
- Produce RFCs for:
  - Management overview and framework
  - Management service definition

Specific Working Group Objectives (Cont)

- Transport and network layer management
- Management of layers below IP
- FTP, Telnet, SMTP management
- Node management

- Schedule
  - Working draft by August 1987
  - Initial distribution of draft RFCs by September 1987

Specific Working Group Objectives (Cont)

- Draft standard by December 1987
- Revised RFCs (if necessary) May 1988

- RFCs stable through end of 1990
- Simple product implementation
- Guarantee clear migration path to ISO
- Management within a single domain
- Approach and tasks outlined in scope and goals document

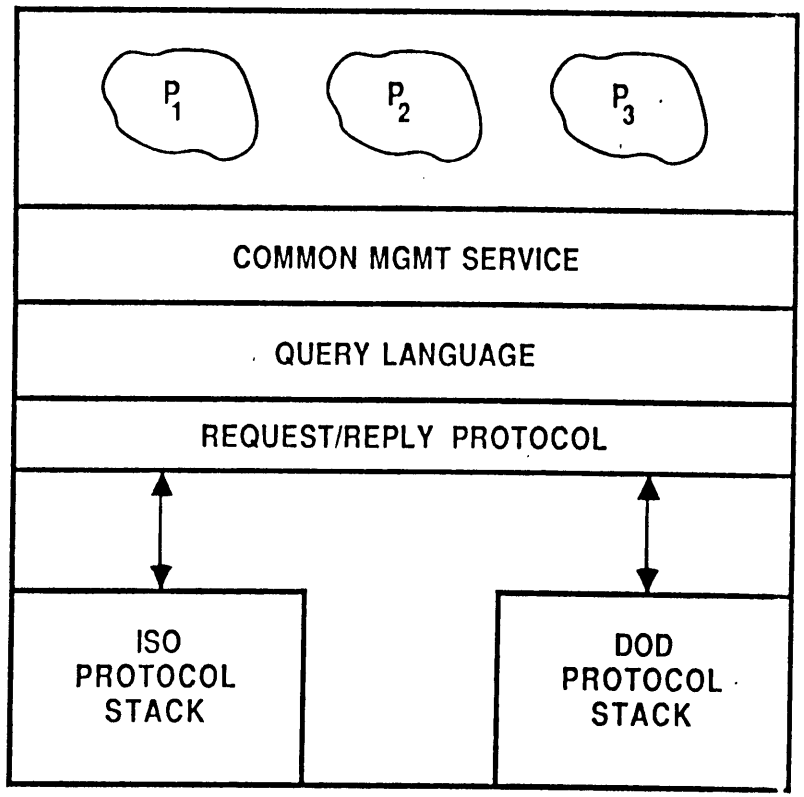
Notice

- We have developed a solution consistent with goals of:
  - TCP/IP network management working group
  - Gateway monitoring working group

Key Decisions

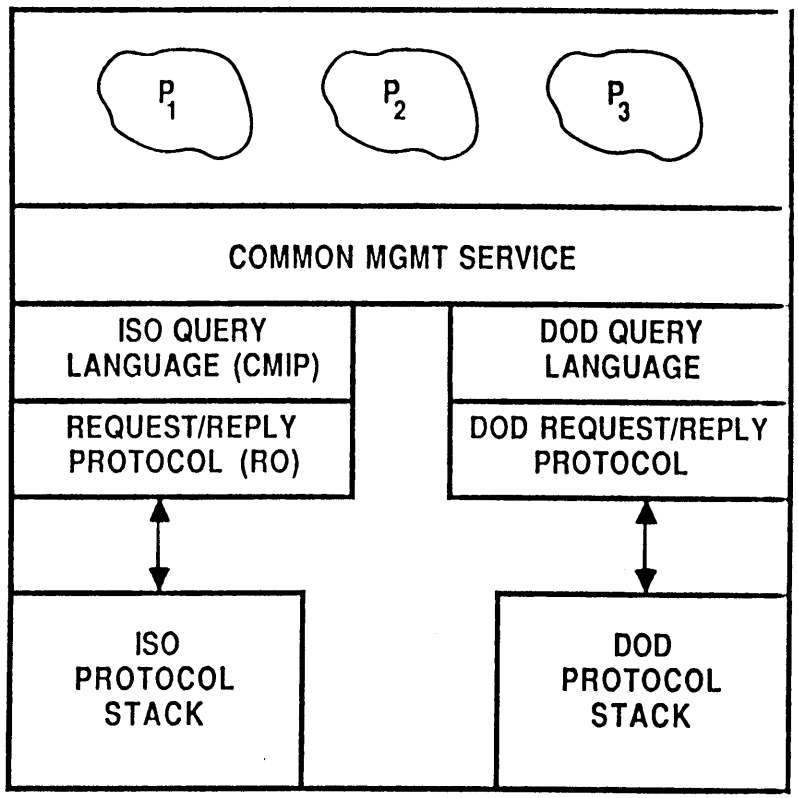
- Protocol architecture
  - Allows development of management applications independent of management protocols
  - Allows management of DoD and ISO products on same network
- Resource identification consistent for DoD and ISO
- Structure of management information consistent for DoD and ISO

Dual-Suite Manager Node Architecture With A Common Query Language and Request/Reply Protocol



MITRE

Dual-Suite Manager Node Architecture With A Different Query Language and Request/Reply Protocol



Management Services Primitives

- GET (attribute) constrained by (filter) synchronization
- SET (attributes) constrained by (filter) synchronization
- EVENT
- CONFIRMED EVENT
- BLOCKING multiple operations with synchronization
- CREATE

Managment Services Primitives (Cont)

- DELETE
- ACTION for defining new operations
- Association control services
  - INITIALIZE
  - TERMINATE
  - ABORT

Structure of Management Information

- Status
- Counter
- Gauge (or meter)
- Tidemark
- Threshold
- Internal event information
- Report control information
- Log
  - •
  - •
  - •

Status

- Draft management overview and framework RFC

  - Amatzia Ben-Artzi, Sytek

- Draft management services RFC

  - Lee LaBarre, MITRE

- Draft resources document (SMI and resources for layers)

Status (Cont)

- Critique of HEMS for management services compatibility
  - Resource identification
  - Filters
  - Service interface (constraints?)
  - Association information
  - Clarity and consistent terminology
- Develop specs for layers below IP, Telnet, FTP, SMTP

Core Working Group Members

Phil Almquist, Stanford
Stan Ames, MITRE
Karl Auerbach, Epilogue Technologies
Amatzia Ben-Artzi, Sytek
Ramesh Babu, Excellan
Eric Benhamou, Bridge
Dave Crocker, Wollongong
Steve Homlgren, CMC
Norm Kincl, HP

Lee LaBarre, MITRE
Dan Laddermann, Wollongong
Dan Lynch, Advanced Computing Env.
Lynn Monsanto, SUN
Keith Morgan, Data General
Jim Robertson, Bridge
Glenn Trewitt, Stanford
Others not in my notes - apologies

MITRE

RFC Set

- Overview
  - System overview
  - HEMs subsystem overview
- MS
  - MS services
  - Programming interface

RFC Set (Cont)

- Protocols
  - HEMP
  - QUERY
- Management information
  - MI overview (layer guidelines)
  - Transport
  - Network

RFC Set (Cont)

- Data link
- Physical

TCP/IP NETWORK MANAGEMENT WORKING GROUP: GOALS AND SCOPE

Revision 3 - 6/18/87

1. REVISION HISTORY

0: Lee Lebarre (Mitre) - 5/19/87

1: Phil Almquist (Stanford, ACIS), Amatzia Ben-Artzi (Sytek), Eric Benhamou (Bridge), David Crocker (Ungermann-Bass), Ramesh Babu (Excelan) - 5/22/87

2: Eric Benhamou (Bridge), Working Group meeting attendees - 5/28/87

3: Eric Benhamou (Bridge), Working Group meeting attendees - 6/18/87

2. INTRODUCTION

Within the Internet community of researchers, users, and vendors, there is a recognized need to address the problems of initiating, terminating, monitoring, and controlling communications activities and assisting in their harmonious operation as well as handling abnormal conditions. The activities that address these problems are collectively considered network management.

The overall objective of the Network Management Working Group of the IETF is to generate a set of specifications, in the form of RFCs, which describe standard mechanisms for network management. The specific goals and scope of the effort are perhaps best described in the context of the networking environment and a framework for network management. Section 3. and 4. describe the networking environment and a framework for network management. Section 5. then suggests some specific working group objectives and the scope of the problems to be addressed. A suggested approach to achieve the working group objectives is provided in Section 6. Finally, Section 7. proposes a prioritized list of specific network management tasks (definitions, network management functions and associated mechanisms) on which this Working Group intends to focus.

The purpose of this document is to serve as a working (rather than a final) statement of goals, objectives, approaches and priorities. For that reason, the definition of the specific words and phrases used in this document will not be attempted here.

3. NETWORKING ENVIRONMENT

The networking environment that is expected to exist during the period when the TCP/IP network management specifications would be used may be described as follows: the Internet of packet switched networks (PSNs) interconnected by IP gateways attaches extended local area networks (LANs) consisting of LANs connected by bridges or gateways. Hosts

and terminal servers from different vendors, containing either the TCP/IP suite or ISO protocol suite (perhaps both), are connected either directly to a PSN or to a LAN. Dual suite application layer gateways may exist to provide translation between comparable TCP/IP and ISO applications (FTP/FTAM, SMTP/X.400, Telnet/VTP). Alternatively, nodes may exist that contain ISO upper layer protocols marbled on top of TCP/IP protocols. Network management stations (possibly dual suite) are on each extended LAN. Other network management stations are connected to the PSNs. Each network management station monitors and controls the nodes within an administratively defined domain (e.g., LAN or PSN) and may interact with management stations of other domains to form hierarchical relationships for global network management.

4. MANAGEMENT FRAMEWORK OVERVIEW

Network management activities can be categorized into the following general administrative functional areas:

- Configuration and Name Management
- Fault Management
- Performance Management
- Security Management
- Accounting Management

The components of network management are the following:

Network Management Entities. Entities, in this document, refer to the objects being managed. In most cases, the notions of an entity and a node will coincide. In the general sense however, an entity need not be restricted to a node, but requires that an Agent participate in the network management environment on its behalf. An Agent must have an IP address. An Entity must have a resource id. The specification of resource id's is part of the task of this Working Group. The mechanisms by which an entity and its agent exchange information are considered to be outside the scope of of this Working Group.

Management applications residing in a specified node (or nodes) that monitor and control activities of other entities within the network to accomplish the above administrative functions.

Management agent applications that may reside in the entities being managed and provide monitoring information to the management applications and effect control actions as specified by the management applications.

Layer management entities (LME) that manage the individual protocol layers and provide monitoring information to the management agent and effect control actions on the layer from the management agent.

Manager-Agent protocols for exchanging monitoring and control information between managers and agents.

Manager-Manager protocols for exchanging information between management domains and maintaining hierarchical relationships for global network management.

5. WORKING GROUP OBJECTIVES AND SCOPE

Given the network environment and management framework described in Sections 3.0 and 4.0, the following specific Working Group objectives and scope are identified:

1. Provide the capability to manage end-systems (e.g. access machines such as hosts, terminal servers, PCs, etc.) and intermediate systems (e.g. gateways) that are identified by an IP address and are attached to or internal to the Internet or associated LANs. The capabilities will be described in RFCs for the following:

   Management framework describing management components, their relationships, and their associated services.

   Definition and representation of management objects, including parameters, actions and events in nodes. Specifically:

   - Management of layers below IP
   - Transport and Network layer management for common protocols
   - FTP, Telnet, and SMTP management
   - Management information pertaining to a node as a whole and maybe specific to that node (e.g., gateway or host).

2. Develop a working draft of the RFCs by August 1, 1987, an initial distribution draft of the RFCs by September 1, 1987, and draft standard by December 1, 1987. If needed based upon implementation experience, a revised specification with clarifications will be published by May 1, 1988.

3. The RFCs should be stable through the end of 1990.

4. The RFCs should lend themselves to simple product implementations.

5. Develop the specifications in a manner that guarantees a clear migration path to the ISO network management standard.

6. Address only management of nodes within a single management domain. Relationships among management domains are beyond the scope of the working group efforts.

6. SUGGESTED APPROACH

The following guidelines are suggested for the Working Group efforts:

1. Base the effort to the greatest extent possible on previous network management efforts within the Internet and standards communities.

2. Use existing Commercial Off The Shelf (COTS) protocols to the maximum extent possible. Avoid development of new protocols, if possible. Specifically, lower layers management standardization will consist in evaluating and integrating existing standards.

3. Provide for functional extensibility with upward compatibility.

4. Allow for implementation-specific extensions to the management facilities.

5. Allow for the distinction amongst different classes of entities with varying degrees of network management intelligence.

7. PRIORITIZATION OF STANDARDS TASKS

The following list itemizes and prioritizes the Working Group standards tasks. When a network management function is listed, the generic mechanisms (Set/Poll/Trap) used to accomplish this function are also mentioned. (Note that the same function may appear in multiple function-mechanisms pairs depending upon the perceived standardization priority of that pair):

0. Architecture and Structure of Management Information.

1. Definition of Activity (Performance and Configuration) Parameters.

2. Definition of Node Parameters.

3. Definition of Network Parameters.

4. Activity Monitoring - Poll.

5. Definition of Faults.

6. Fault Monitoring - Trap.

7. Activity Monitoring - Trap.

8. Fault Monitoring - Poll.

9. Security of Network Management Actions.

10. Download.

11. Layer Management Parameterization - Set/Poll.

12. Name Resolution.

13. Definition of Accounting Parameters.

14. Accounting - Poll.

15. Accounting - Trap.

16. Network User Validation.

Work on items 1-16 will be performed according to the following layer priorities:

1. Transport and Network Layers.

2. Lower Layers.

3. FTP, Telnet and SMTP.

Due to operational constraints, the following topics, although relevant to network management are specifically excluded from the list as non-goal items:

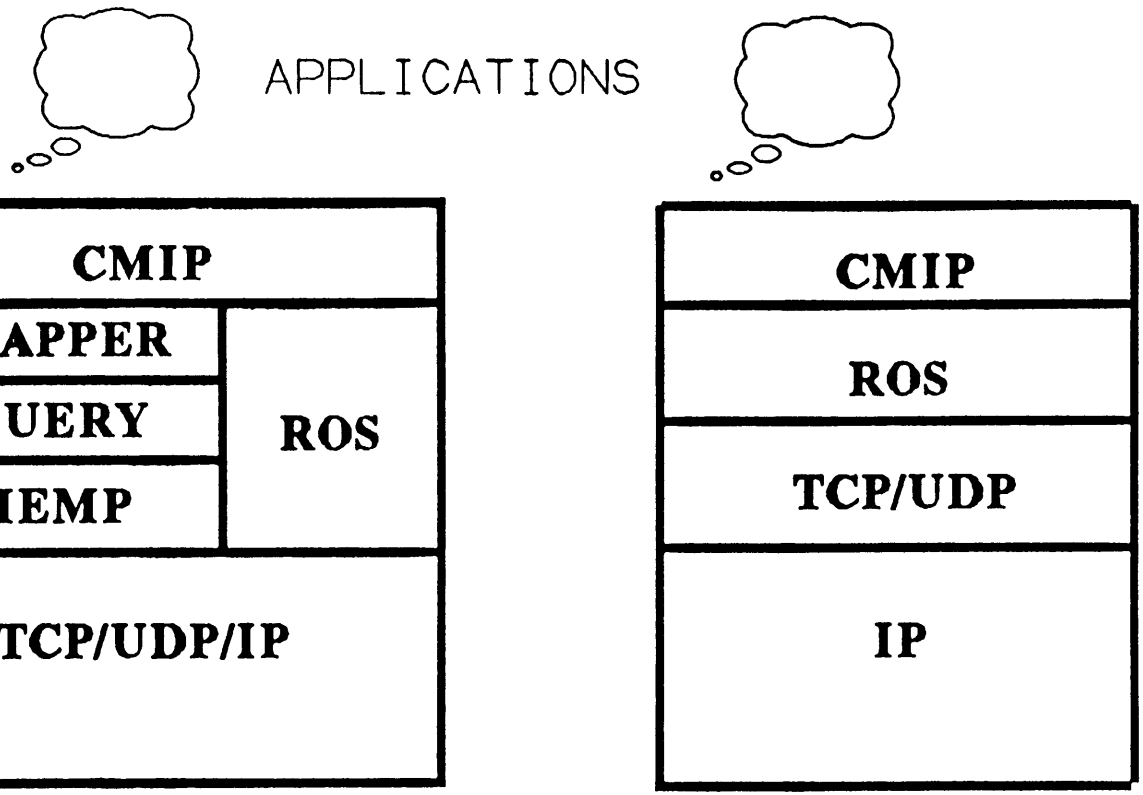
1. Activity Analysis.

2. Fault Analysis.

3. Accounting Analysis.

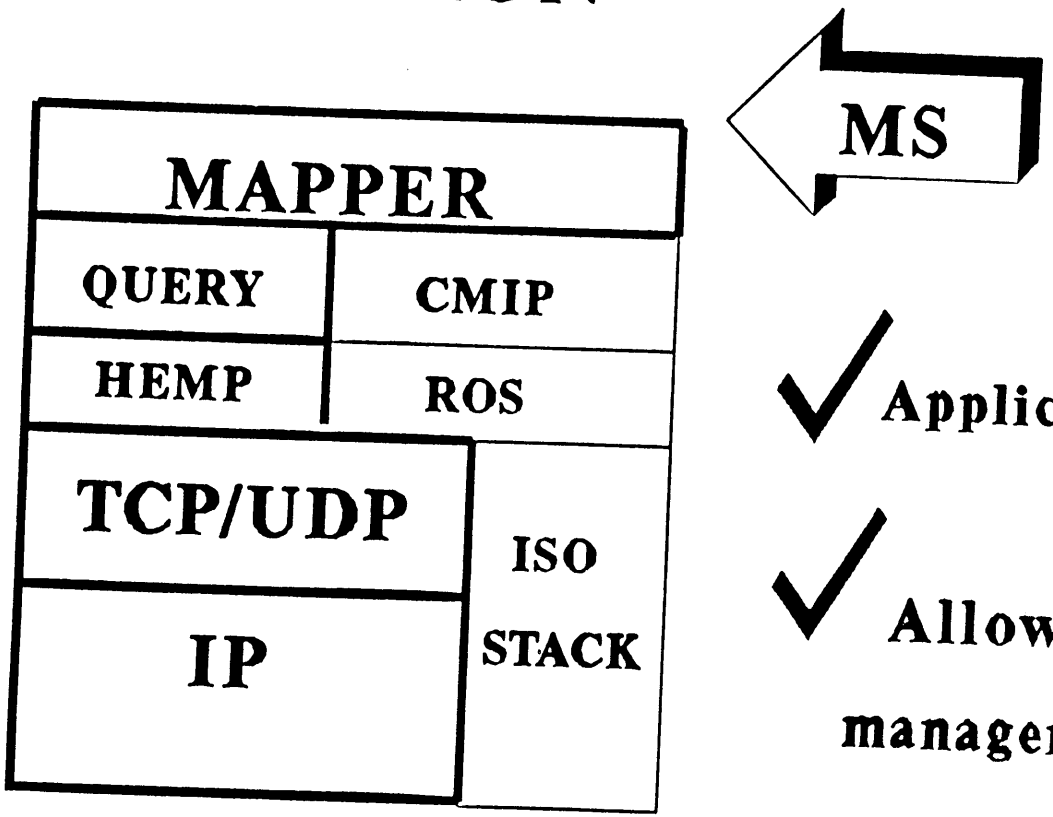
4. User Specific Parameters.

HOW TO DO ISO/NM on TCP



Sytek

SOLUTION



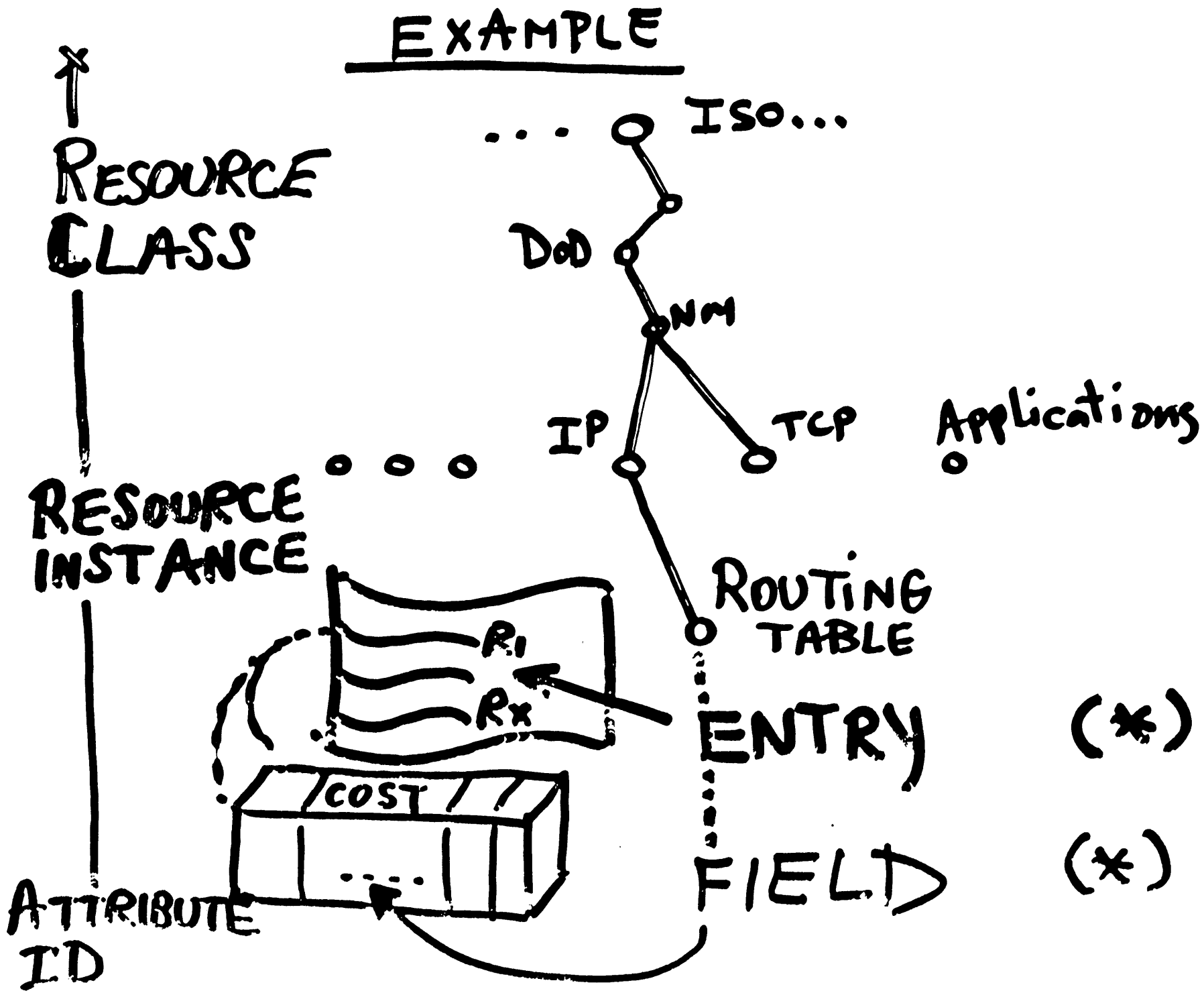
✓ Applications are preserved

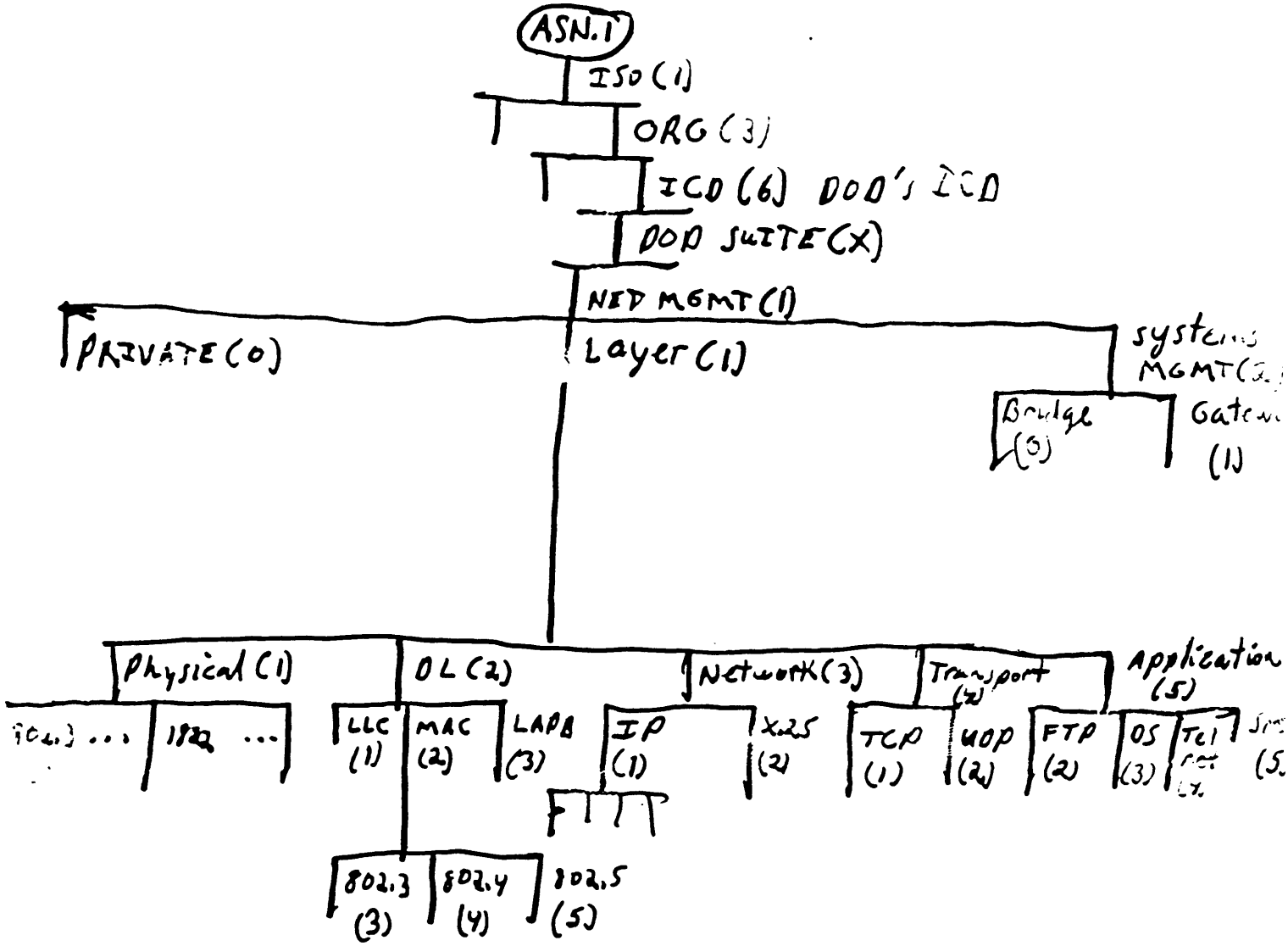
✓ Allows simultaneous management of ISO/HEMS

Sytek

Structure of Management Information

- Status
- Counter
- Gauge (or meter)
- Tidemark
- Threshold
- Internal event information
- Report control information
- Log
- ⋮


EXAMPLE
RESOURCE CLASS
RESOURCE INSTANCE
ATTRIBUTE ID
ISO...
DoD
NM
IP
TCP
Applications
ROUTING TABLE
R1
Rx
COST
ENTRY (*)
FIELD (*)



ObjectID

```
ObjectClass ::= CHOICE {
        fullpath [0] OBJECT IDENTIFIER
        relativepath [1] OCTET STRING
}
ObjectInstance ::= SEQUENCE {
        instance OCTET STRING
```

e.g. TCP = fullpath (1 3 6 x 1 1 4) relative path

MANAGEMENT INFORMATION

- ORGANIZED IN A TREE STRUCTURE
- REFERRED TO THROUGH THE PATH IN THE TREE FROM THE ROOT
- SAME RESOURCE DEFINITION FOR SAME RESOURCE IN DIFFERENT "BOXES"
- RESOURCE CLASS, INSTANCE & ATTRIBUTE ID

Sytek

A Simple Gateway Monitoring Protocol

(aka "Simple-Mon" or "SGMP")

Motivation:

- o Concern about multiple standards efforts (e.g. ISO, NSF)
- o Pressing network management needs
- o Desire for implementation experience

CHUCK DAVIN
jrd@monk.proteon.com

Simple Gateway Monitoring Protocol

- o UDP-based -- adaptable to other transports
- o Retrieval of individual variables by name
- o Limited number of unsolicited trap messages
- o ASN.1 data representation

Current Project Status

- Four distinct implementations in progress
  - Two gateway implementations
  - Two host implementations
- Working prototype for a Sun workstation (Proteon)
- Working monitoring tools for both Ultrix and MS-DOS (U. Tennesee at Knoxville)
- Monitoring tools under development (RPI)
- Gateway implementation under development (Cornell)
- p4200 gateway implementation working -- still some bugs (Proteon)

Authentication Protocol

Message Format:

Octet	Interpretation
0 - 1	Message Length (Big-Endian integer)
2	Session Name Length (value = n)
3 - (n + 2)	Session Name
(n + 3) -	User Data

Three Functions (selected by Session Name):

- Authentication Function

  Boolean-valued; verifies that message is "authentic"

- Representation Function

  Maps user data into protocol representation (e.g. your favorite checksum/encryption algorithm)

- Interpretation Function

  Maps protocol representation to user data (e.g. your favorite checksum/decryption algorithm)

Trap Messages

- Four currently defined
  - "Warm" boot
  - "Cold" boot
  - Link Failure
  - Authentication Failure

Trap Message Format:

- An integer that specifies the type of the trap
- Zero or more values of integer or octet string type that provide additional information

Get Request/Response Message Format

Field	Interpretation
Request Id	Specified by user to match Request with Response
Error Status	In a Response message, indicates the result of the corresponding Request
Error Index	In a Response message, indicates the component of the corresponding Request that may be in error
Variable Name	An octet string that names some node of the variable name space tree; in a Response, the name of the variable actually retrieved
Variable Value	In a response message, the value of the variable retrieved

- Multiple Variable Name-Variable Value pairs may appear in a single Get Request message

Variable Naming Conventions

- Symbolic representation of variable names

  Used by humans

- Numerical representation of variable names

  Used on the network

Example:

The variable whose name is represented symbolically as "GW_version_rev" might be represented numerically as 01 01 02

Protocol Variable Space

- Variable space is conceptually a tree with named edges
- Variables are at the leaves of the tree
- Name for an individual variable is the concatenation of edge names along the path from the root to the leaf
- For a given node of the tree, its edges are ordered lexicographically from left to right according to name

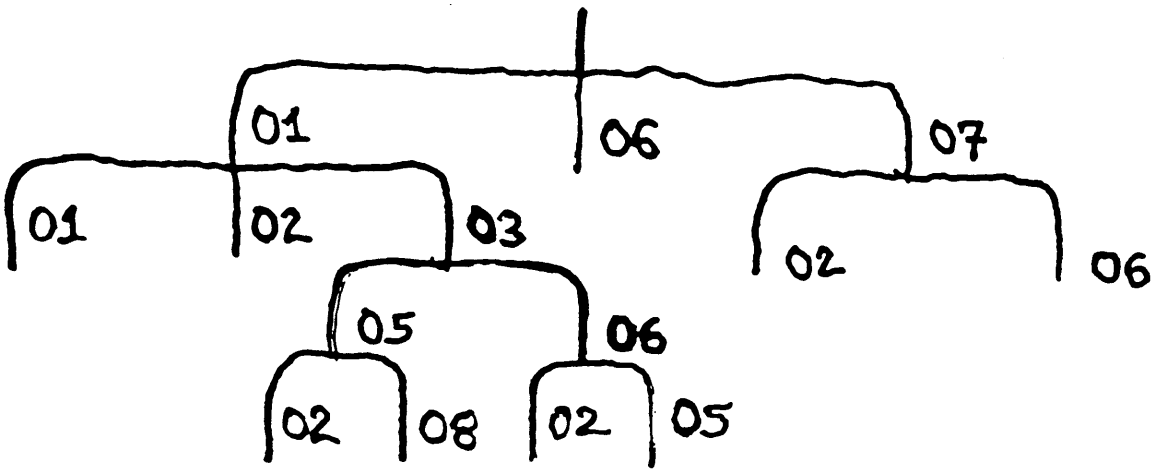
- If an internal node of the variable space tree is named in a Get Request message, then the server returns the variable that is at the leftmost leaf of the named subtree

- If a leaf of the variable space tree is named in a Get Request message, then the server returns the variable that is at the next leaf to the right in the tree

Protocol Operation Examples

- Request for name "01 03 05" in the tree below returns the value for variable "01 03 05 02"

- Request for name "01 03 06 02" in the tree below returns the value for variable "01 03 06 05"



LEAVES:

01 01
01 02
01 03 05 02
01 03 05 08
01 03 06 02
01 03 06 05
06
07 02
07 06

Using the Protocol: Example 1

Find the gateway for destination 128.185.123.16

- Send a Request for the name (symbolically)

  "GW_pr_in_rt_gateway_128_185_123_16"

  or (numerically)

  01 04 01 02 01 80 B9 7B 10

- Receive Response and display answer

Using the Protocol: Example 2

Dump the routing table

- Send a Request for the names

(symbolic)	(numeric)
"GW_pr_in_rt_gateway"	01 04 01 02 01
"GW_pr_in_rt_type"	01 04 01 02 02
"GW_pr_in_rt_metric"	01 04 01 02 03

- Receive Response
- If the prefix of the returned variable names is not as "expected," then all routes have been retrieved
- Display the three retrieved values as a row of the routing table
- Send a request for the three names returned in the last Response
- Repeat from the second step above

What We Learned

- ASN.1/X.409 parsing is not impossible
- ASN.1/X.409 constructs that pertain to multiple protocol layers are difficult
- Easily extensible protocols are easier to specify and standardize

Automated Network Mgmt Westcott (BBN)

Automated Network Management

- Problem Definition
- Goals
- ANM System Architecture
- Network Components
- Distributed Management Modules
- Client Processes

Problem Definition

learn to manage the DARPA Internet:

Size

- 250 networks
- + 5-10 networks/month

Complexity

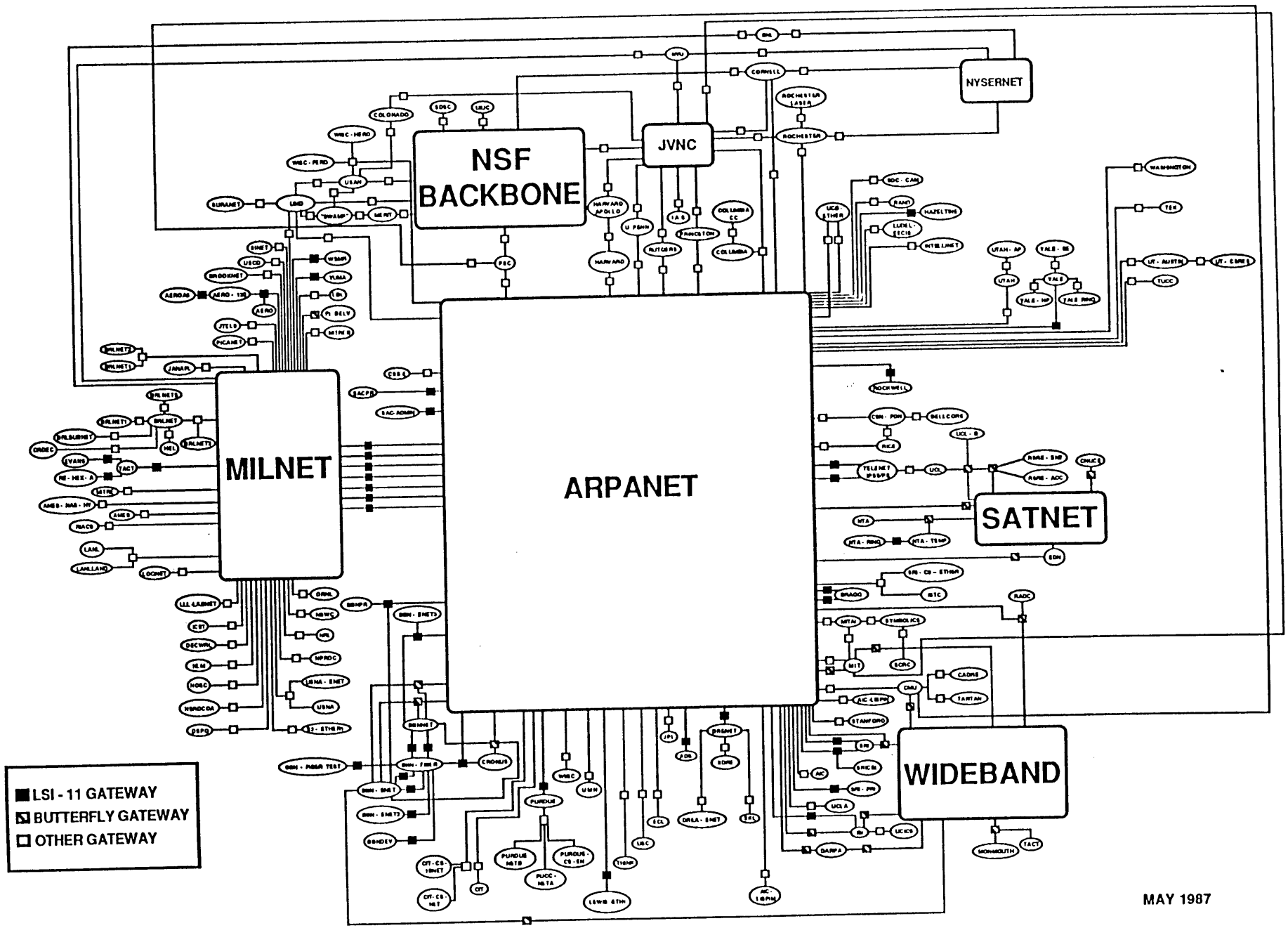
- diverse components
- diverse protocols
- wide geographic range

Performance

- widely varying by route

  - throughput range 9.6Kb to 80 Mb
  - forwarding delay from microseconds to seconds

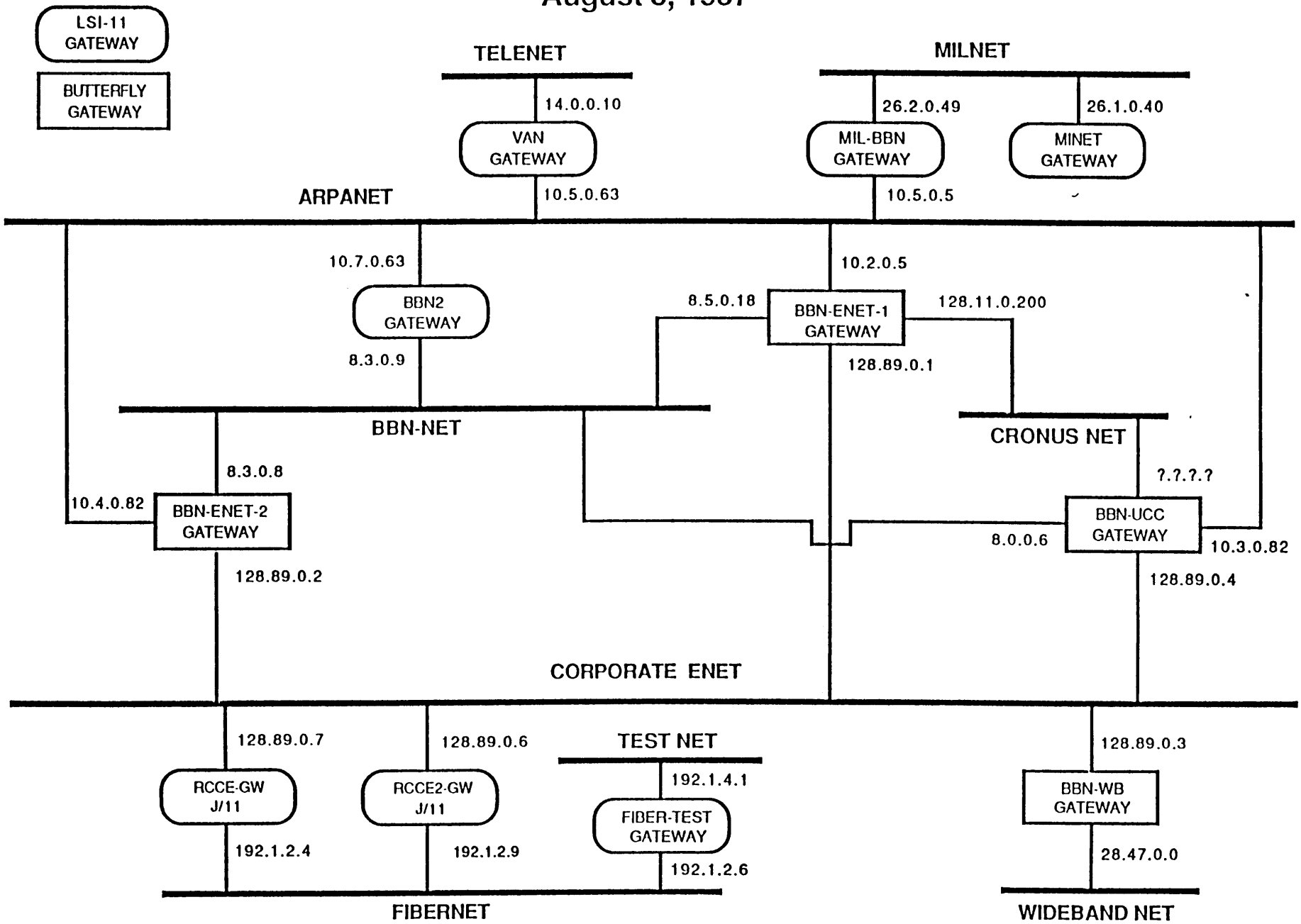
- protocols don't know expected performance



BBN Communications Corporation

BBN Networks and Gateways

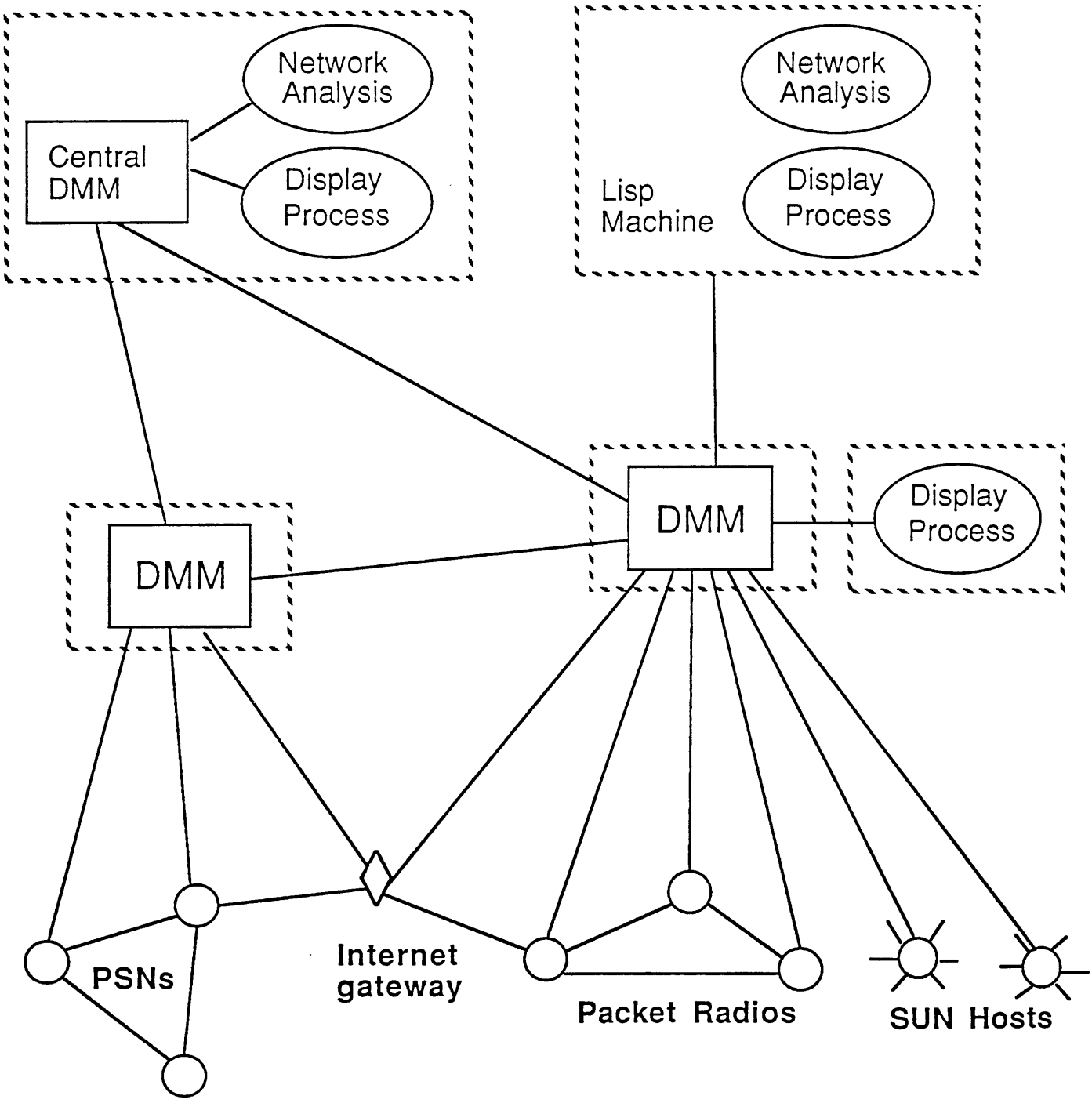
August 3, 1987



Automated Network Management Goals

- Diverse Components
- Distributed Architecture
- Intelligent Assistance
- User-Friendly Interface

ANM Distributed Architecture



Network Components

- Current Capabilities

  -Mobile Packet Radios

  -LSI-11™ and Butterfly™ Gateways

  -UNIX™ Hosts (SUNs)

  -C/30™ Packet Switch Nodes

- Planned Capabilities

  -Wideband Network

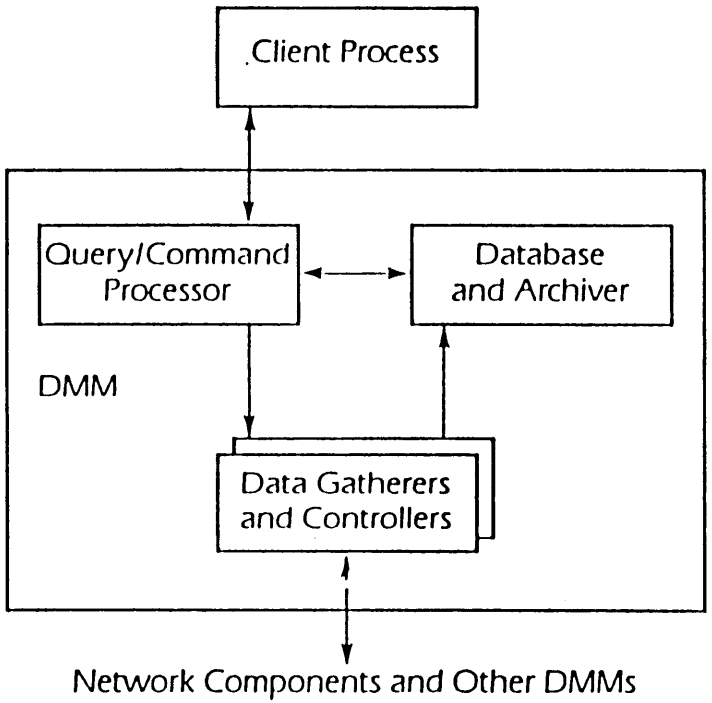
  -Multiple Satellite System (MSS)/ Cooperating Space Systems (CSS)

  -LAN Management

Distributed Management Module Function

- Translates and Forwards Queries and Control Commands to Network Components
- Forwards Queries and Control Commands to Other DMMS When Necessary
- Stores Data Collected From Components and Other DMMs
- Archives Network Management Data
- Maintains Data Catalogs to Support Distributed System

Distributed Management Module Architecture



Client Process - User Interface

- Retrieval
- Presentation
- Alerting
- Explanation

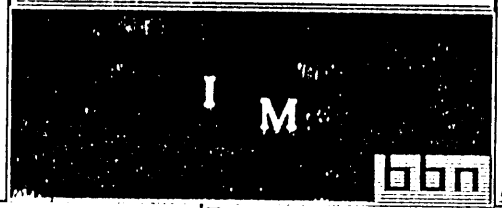
sri-milnet-gw.arpa

Main Object Buffer

Other Object Buffers

Clear Buffer
Save Buffer
Select Tuples
Select View
Find GW Obj by Name
Select Gateways
Select GW Interfaces
Poll Initial Gateway
Poll All Gateways

IM Commands



Pane Not Used

gw:sri-milnet-gw.arpa neighbor:yale-gw.arpa

IVAR-NAME	IVAR-VAL
BYTSENT	7067
DGLANDRP	0
DGOFULDRP	0
EGP_FLAG	0
NAME	yale-gw.arpa
PKTSFORW	8
PKTSFROMUS	46
ROUTUP2CNT	7
ROUTUPFROMCNT	9
UPDWN_FLAG	2

Main Tuple Buffer

IM Lisp Interaction Pane

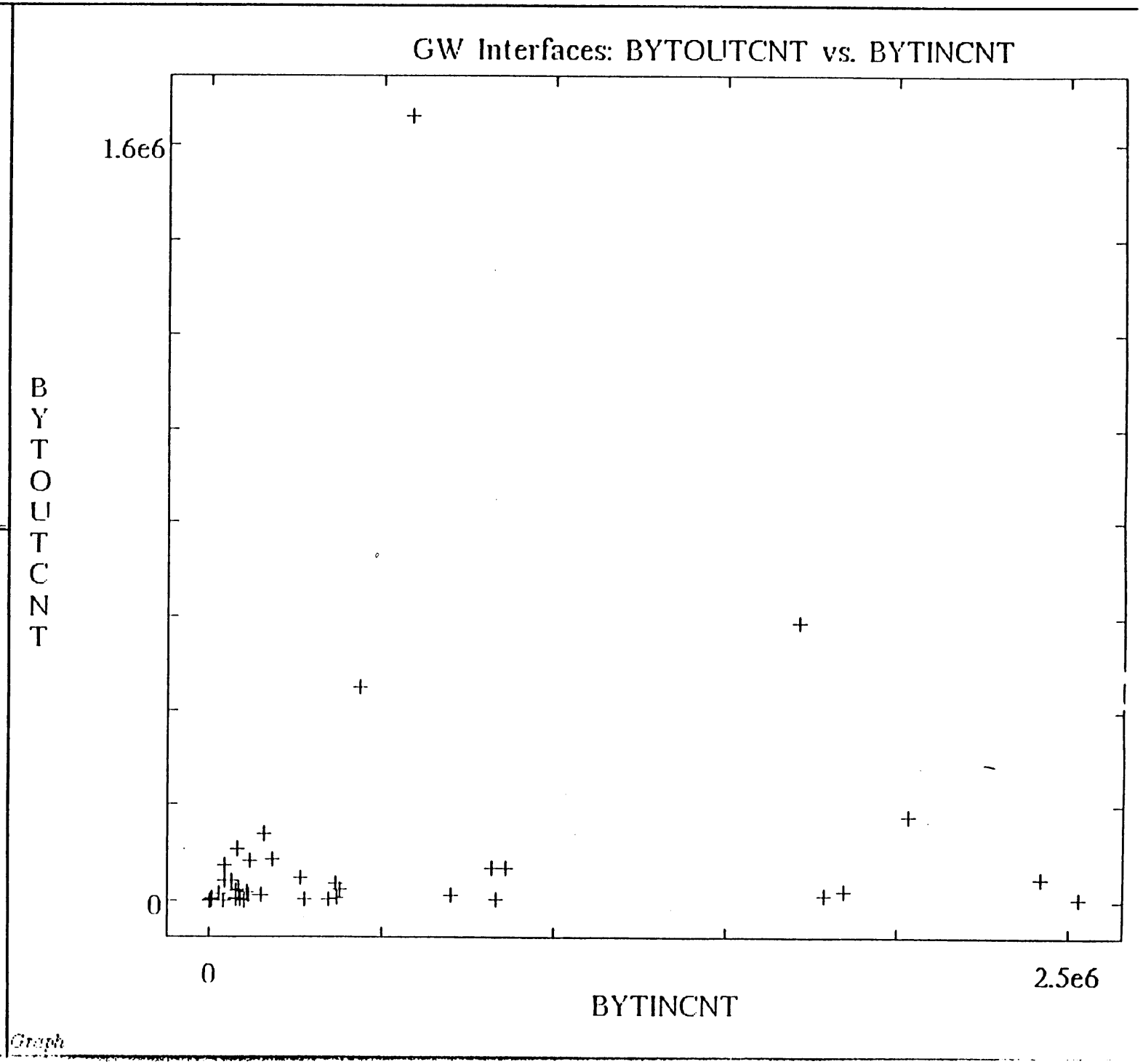
Graph View

GW Histogram
GW Scatter Plot
GW Interface Histogram
GW Interface Scatter Plot ×
History of Datum
Kill View
New Command

Commands

70 unknown values

Messages



Graph

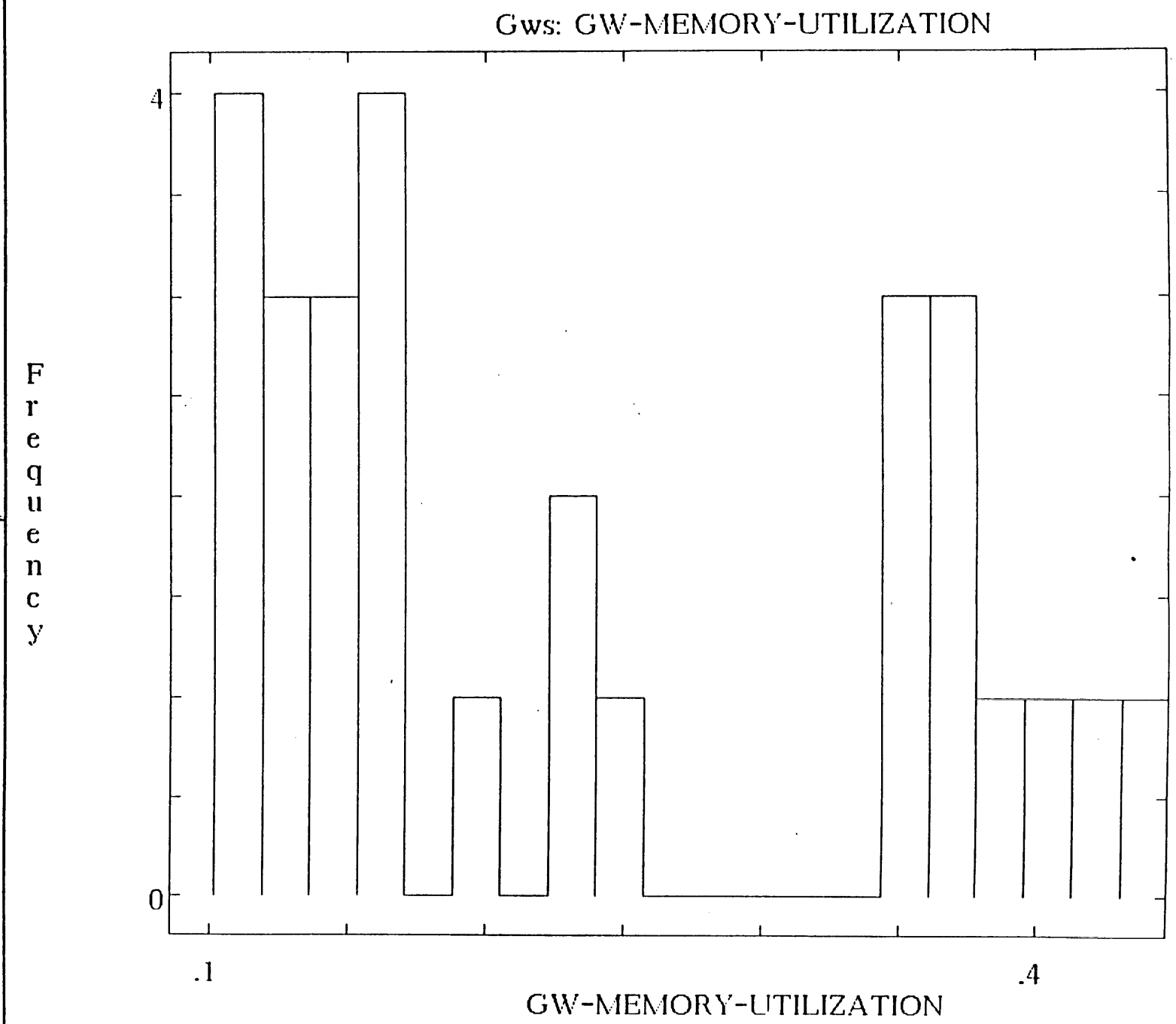
Graph View

GW Histogram ×
GW Scatter Plot
GW Interface Histogram
GW Interface Scatter Plot
History of Datum
Kill View
New Command

Commands

1 unknown values

Messages



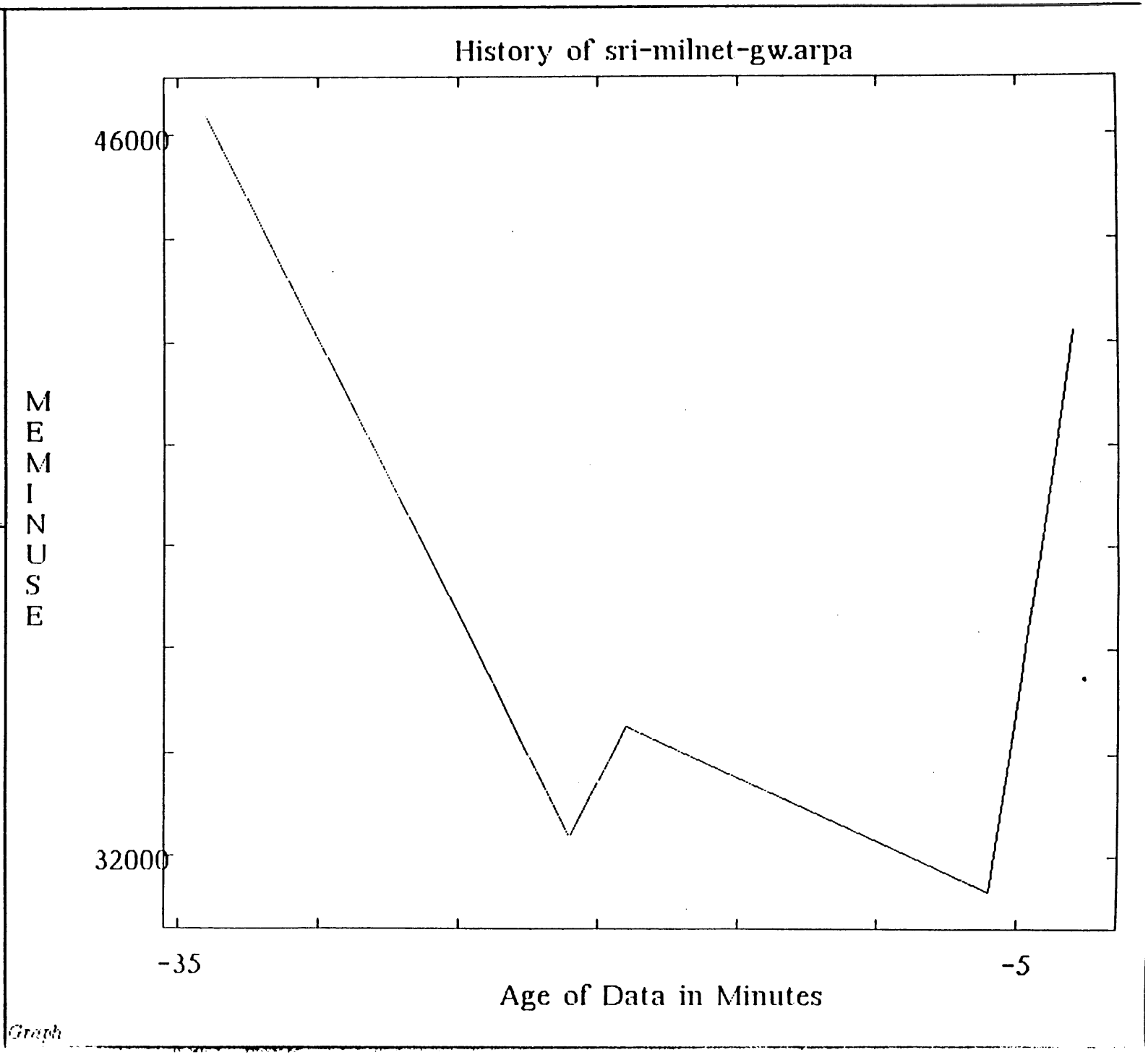
Graph

Graph View

GW Histogram
GW Scatter Plot
GW Interface Histogram
GW Interface Scatter Plot
History of Datum
Kill View
New Command

Commands

Messages

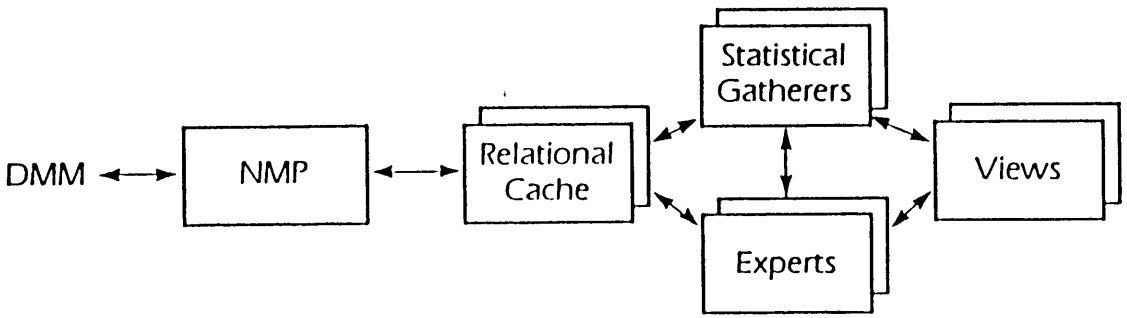


Graph

Client Process - Data Analysis

- Arithmetic Calculations
- Statistical Analyses
- Network Algorithms
- AI-Based Reasoning

Intelligent Network Manager



ANM Status

Release 2.0

- Delivered in January '87
- In System Test at SRI, Ft. Monmouth & Ft. Bragg
- Monitors SUN Workstations, LSI-11™ Gateways, Packet Radios, and Packet Radio Stations

Release 3.0

- Will Deliver in Summer '87
- Contains New User Interface
- Adds Monitoring for C/30 PSNs

ANM Status

Release 4.0

- Will Deliver in Spring '88
- Adds Monitoring for Butterfly Gateways
- Contains new relational database with report generating capabilities
- Deliver to NOC for DARPA Internet

High-Level Entity Mgmt Sys Partridge (BBN), Trewitt (Stanford)

The High-Level Entity Management System (HEMS)

- Motivation and Philosophy (Partridge)
- HEMP (Partridge)
- ASN.1 (Trewitt)
- Overview of Data Organization (Trewitt)
- Query Language (Trewitt)
- ISO Compatibility (Trewitt)
- Detail Data Organization (Partridge)
- Events/Traps (Partridge)

HEMS: Motivation

- Increasing Heterogeneity Making Local and Global Network Management Difficult

- Lack of A Generally Accepted Solution

- MULTIPLE MONITORING CENTER

HEMS: Philosophy (Extensibility)

- Types of Extensibility: Architectural and System

- Architectural Extensibility: How Easily Can We Revise the Overall Design?
  - — Had To Assume We Wouldn't Get Things Quite Right the First Time

- System Extensibility: Supporting Extensions Within the System As Designed
  - — Don't Want Extensions That Destroy Homogeneity
  - — Problem of Confining All Systems To One Abstract Model
  - — IP Networks Keep Evolving

HEMS: Philosophy (System Model)

- Entity Being Managed by an Application at Another Entity
  - — Decided to Model as Application to Application Link

- Network Distance Between Entities is Potentially Long or Flakey or Both
  - — Implies a Reliable Transport Protocol
  - — Simple RPC Probably Won't Work (Delays Between RCP Calls Intolerable)

- Must Be Possible to Manage Network Without Continuous Polling
  - — From Operational Experience
  - — Implies Need for Events/Traps

HEMS: Overall Architecture

- Three Parts to Architecture:
  - — Message Protocol
  - — Query Language
  - — Data Set

- Can Tinker to a Large Degree with Each Piece Without Disturbing the Others

- Language Has Explicit Support (Discovery and Definition Methods) for Entity-Specific Extensions to the Data Set

High-Level Entity Management Protocol (HEMP)

- A Message Protocol:
  - — Each Message is Distinct (No Long Term Association)
  - — Assumes a Lot is Being Done in A Message

- Provides Hooks Required For Network Management:
  - — Standard Encapsulation
  - — Authentication/Access Control
  - — Encryption

High-Level Entity Management Protocol (HEMP): Message Formats

```
HempMessage ::=   [0]   IMPLICIT SEQUENCE {
                  [0]   IMPLICIT EncryptSection OPTIONAL,
                  [1]   IMPLICIT ReplyEncryptSection OPTIONAL,
                  [2]   IMPLICIT AuthenticateSection OPTIONAL,
                  [3]   IMPLICIT CommonHeader,
                  [4]   IMPLICIT Data }
```

High-Level Entity Management Protocol (HEMP): Encryption

- Required to Protect Data From Eavesdroppers
- Does Not Protect Against Traffic Analysis
- Request and Response May Use Different Methods
- No Encryption Schemes Defined — Simply Defined Hooks

High-Level Entity Management Protocol (HEMP): Authentication/Access Control

- Required to Protect Data From Intruders
- Needed to Protect Entities From Unauthorized Processing Requests
- Needed to Authenticate Critical Management Information
- Hooks Defined, and Two Systems (Password and by Encryption)

Detailed Data Organization

- Needed Some Way to Subdivide the Data
  — Choose Classic Protocol LayerCake but...
  — Had to Add A Few Things
- Current Definition Needs Considerable Refinement
- It Has Been Suggested That We Will Have to Start Requiring Management Parameters to Be Defined in New Protocol Specifications.

Detailed Data Organization: Issues

- No Data Reduction Should Be Done by Entity.
  - — Entity Stores Raw Data (e.g., Counters) Used by Applications To Get More Complex Statistics.
  - — Limits Impact On Entity Performance (Minimal New Overhead Per Packet).

- Have To Assume Multiple Users At a Time
  - — Can't Allow An Application To Change a Counter
  - — Event (Trap) Control Is *Very* Difficult

- Don't Want To Dictate Machine Architectures
  - — Flexibility on Roll-Over Counter Sizes, etc.

Detailed Data Organization: The Root Directory

- Top-Level Directory Is Divided Into Seven Groups
  - — System Variables: General System Values Such as Clocks and Buffer Management.
  - — Event Controls: Mechanisms For Managing When Events (Traps) Are Sent.
  - — Interfaces: Information About Network Interfaces.
  - — IP Layer: Information on IP (statistics on fragmentation, packets switched, traffic matrices, checksum problems, etc).
  - — Routing: The Routing Table. Information on Routing Protocols Are Stored At Transport Layer (e.g. EGP) or Above (e.g., RIP).

Detailed Data Organization: The Root Directory (cont.)

— Transport: All Transport Protocols (e.g., ICMP, TCP, UDP, RDP) Which Are Used by This Entity.

— Applications: Information on Applications Such as the Domain Name Server (currently not defined).

Events (Traps)

- The Least-Well Developed Portion of HEMS.
- Required To Limit the Amount of Polling We Must Do to Manage Network.
- Must Be Standardized. The Same Trap Should Mean The Same Thing Everywhere.

Events (Traps):
What We Have So Far

- Events Are Sent To A List of One or More Addresses Whenever A Certain Condition Occurs a Certain Number of Times.

- Events Have Assigned Codes, Which Are Standardized.
  - — Per-Entity SubCode That Can Be Used To Identify Where Event Occurred
  - — Each Event Has A Fixed Set of Data That Must Be Returned With It.

- Events Also Contain Text Descriptions, So There Can Be Entity-Specific Events Which We Can Interpret.

Events (Traps): What Is Missing

- Management Centers Will Want To Customize Their Event Stream
  - — Add Additional Information to Event Message
  - — Select Which Events They Will Accept, and When

- Hard To Provide Powerful Tailoring Facilities Without Making Event Processing Very Expensive and Cumbersome.

- Difficult To Determine What the Generic Event Codes Should Be.
- Difficult To Determine What Data Must Be Returned With Each Event Code.

ISO ASN.1

(a.k.a. CCITT X.409)

Glenn Trewitt
Stanford University

ASN.1 defines both

Notation	– printed form
Representation	– binary encoding, *e.g.* in a packet

I will discuss only the representation.

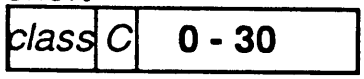
Each "data element" consists of 3 components:

Identifier	– identifies "type" of element
Length	– length of *contents*, in octets
Contents	– actual data for the element

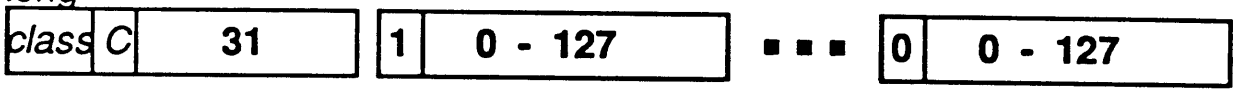
Representation Format

Identifier formats:

short



long

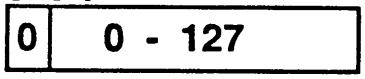


Class defines scope of identifiers:

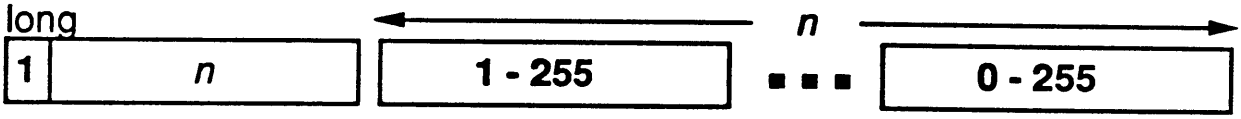
00 Universal	"well-known" types
01 Application	application-specific
10 Context	within some data item
11 Private	not defined

Length formats:

short



long



indefinite

Data Types

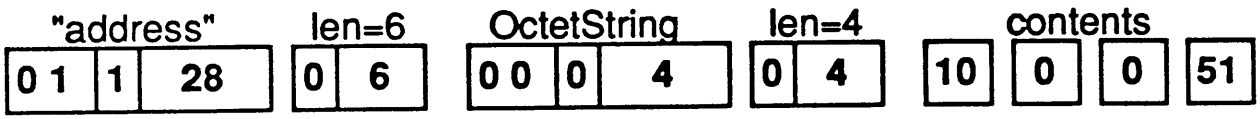
Type	code	Constructor/Primitive
reserved	0	
Boolean	1	primitive
Integer	2	primitive
BitString	3	either
OctetString	4	either
Null	5	primitive
Sequence	16	constructor (record or array)
Set	17	constructor (tagged values)
Tagged		constructor (explicit)
		primitive (implicit)
Choice		*notational only*
Any		*notational only*

Also, various string types and date formats are built on top of these.

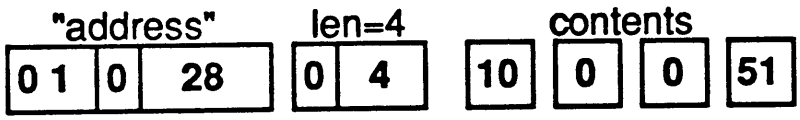
Example Representations

Internet Address — Choose an identifier, say 28 in the "application-specific" class. Two choices:

Explicit: The value is explicitly an *OctetString*:



Implicit: Everyone "knows" that 28 implies *OctetString*:



We have chosen to use implicit representations for HEMS data.

HEMS Data Organization

Glenn Trewitt — Stanford University
Craig Partridge — NNSC at BBN Laboratories

Vast amount of data to be monitored.

- Most is in tables (routes, arp, ...)

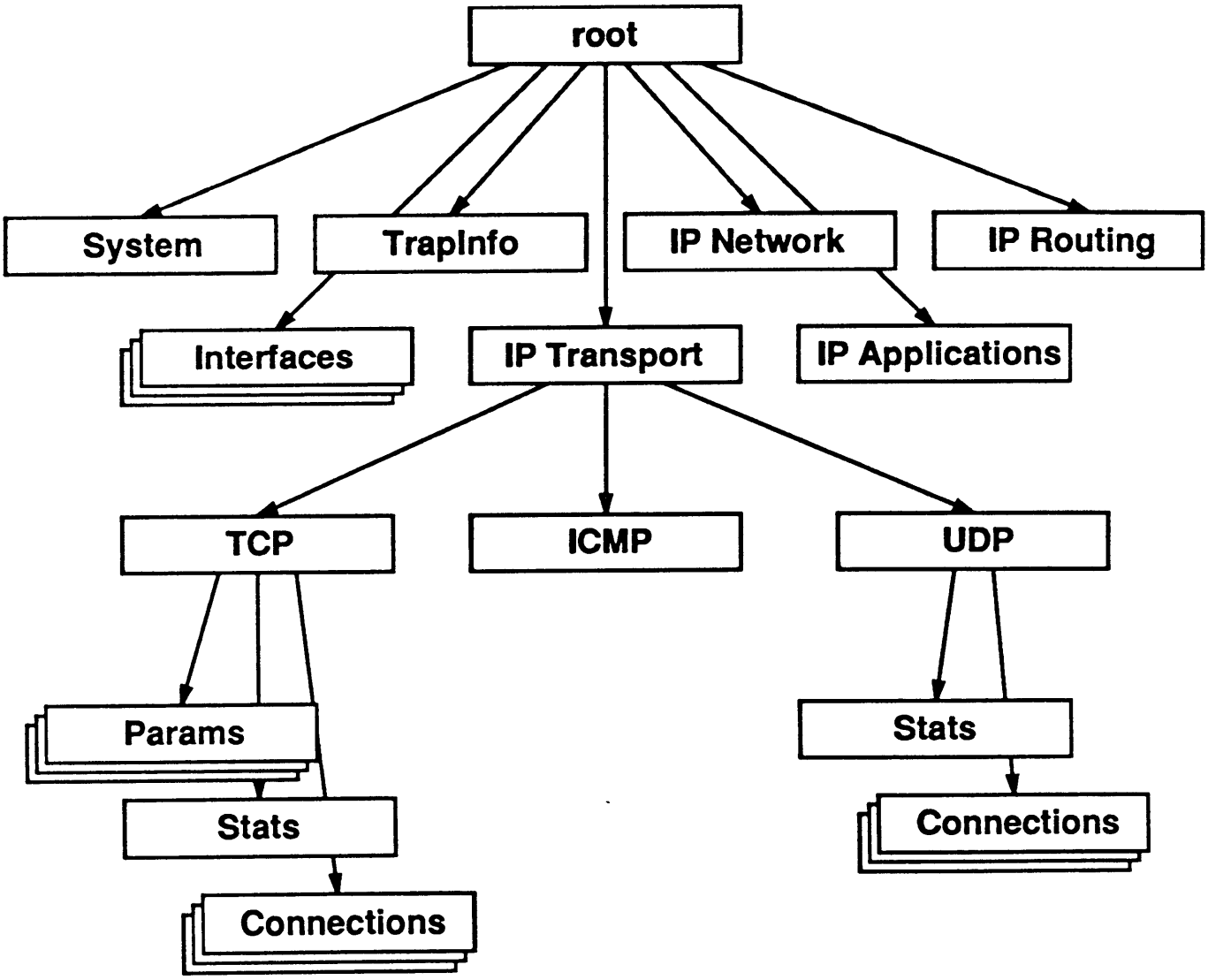
Organization should provide structure.

- Group related data together.
- Allow data to be named.

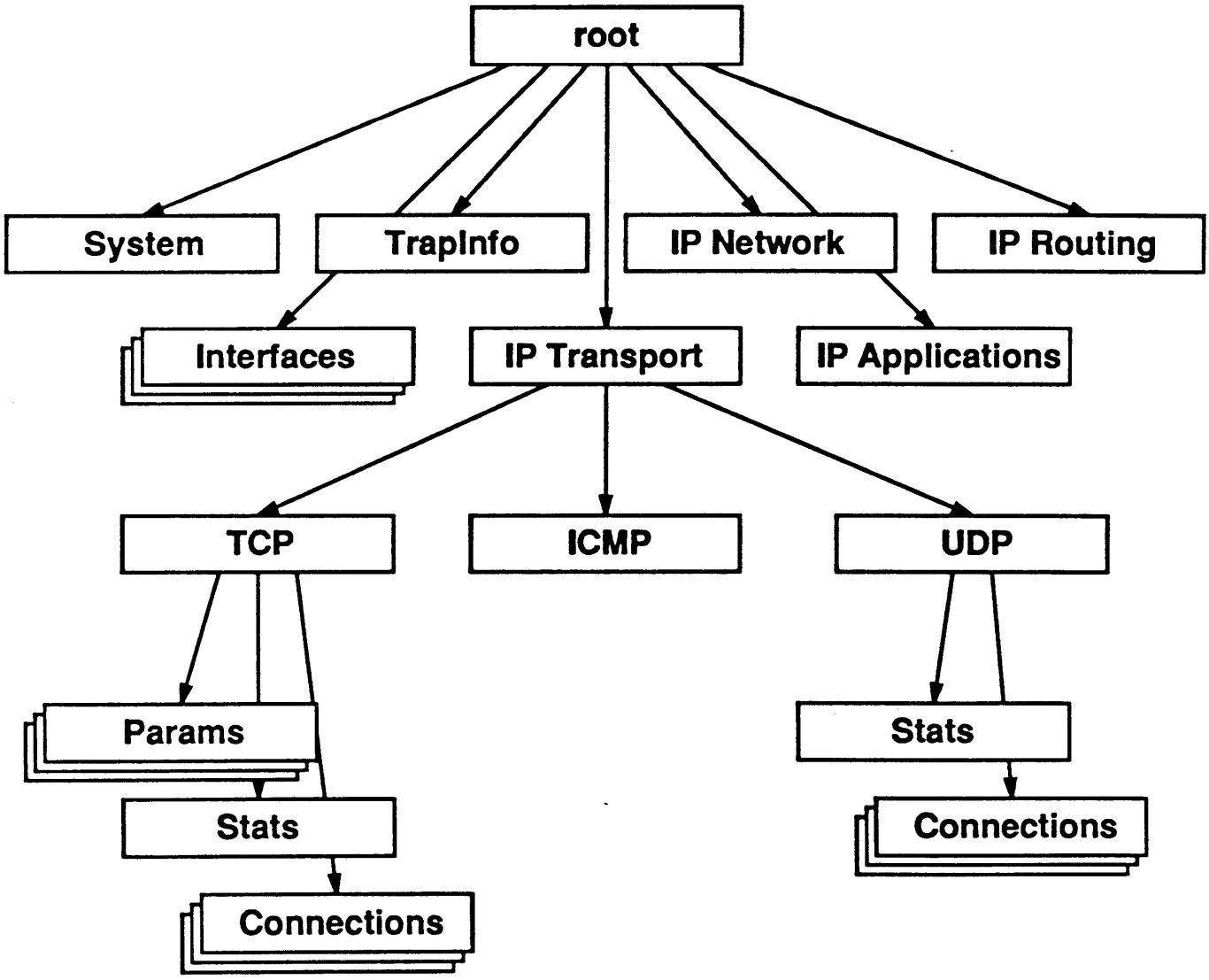
HEMS uses a heirarchical structure.

- Data is named by giving path through tree.
- Maps onto ASN.1 easily.

Partial Data Tree Skeleton



Partial Data Tree Skeleton



Representing Data

Internal nodes are called dictionaries.

- Set of key / value pairs.
- Values may be data or other dictionaries.
- "dictionary" is taken from PostScript™.

The data in the "TCP Stats" dictionary is represented by:

```
root{ IPTransport{ TCP{
                Stats{ . . . } } } }
```

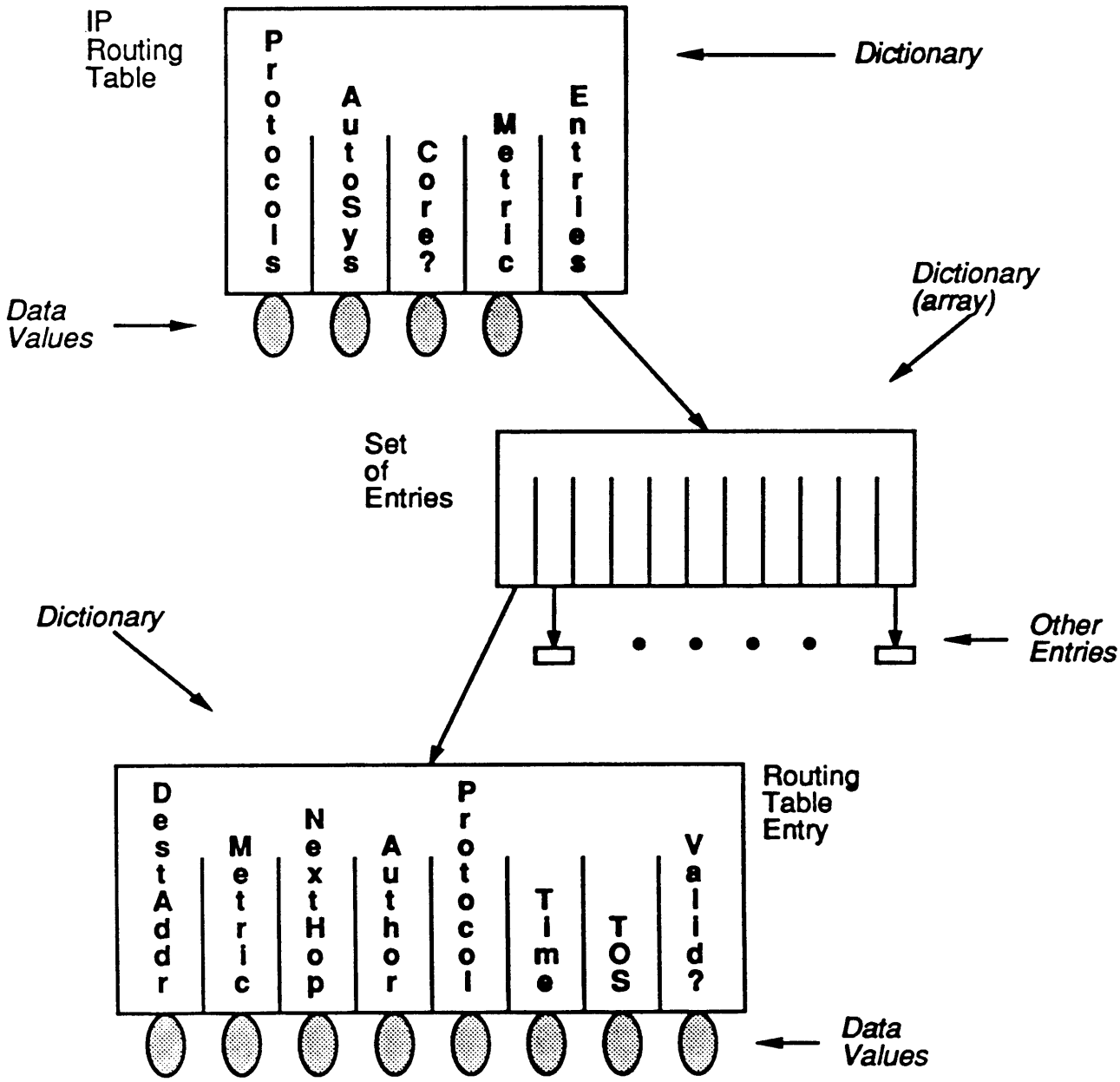
Omit the root name, leaving:

```
IPTransport{ TCP{
        Stats{ . . . } } }
```

The **name** of the data is the same, except with the data removed, and the path left intact:

```
IPTransport{ TCP { Stats } }
```

Routing Table Detail



Examples of Names

The entire Routing Table subtree is named by:

```
IPRouting
```

The autonomous system number would be given by:

```
IPRouting{ AutoSys }
```

All of these names have referred to only one object in the tree.

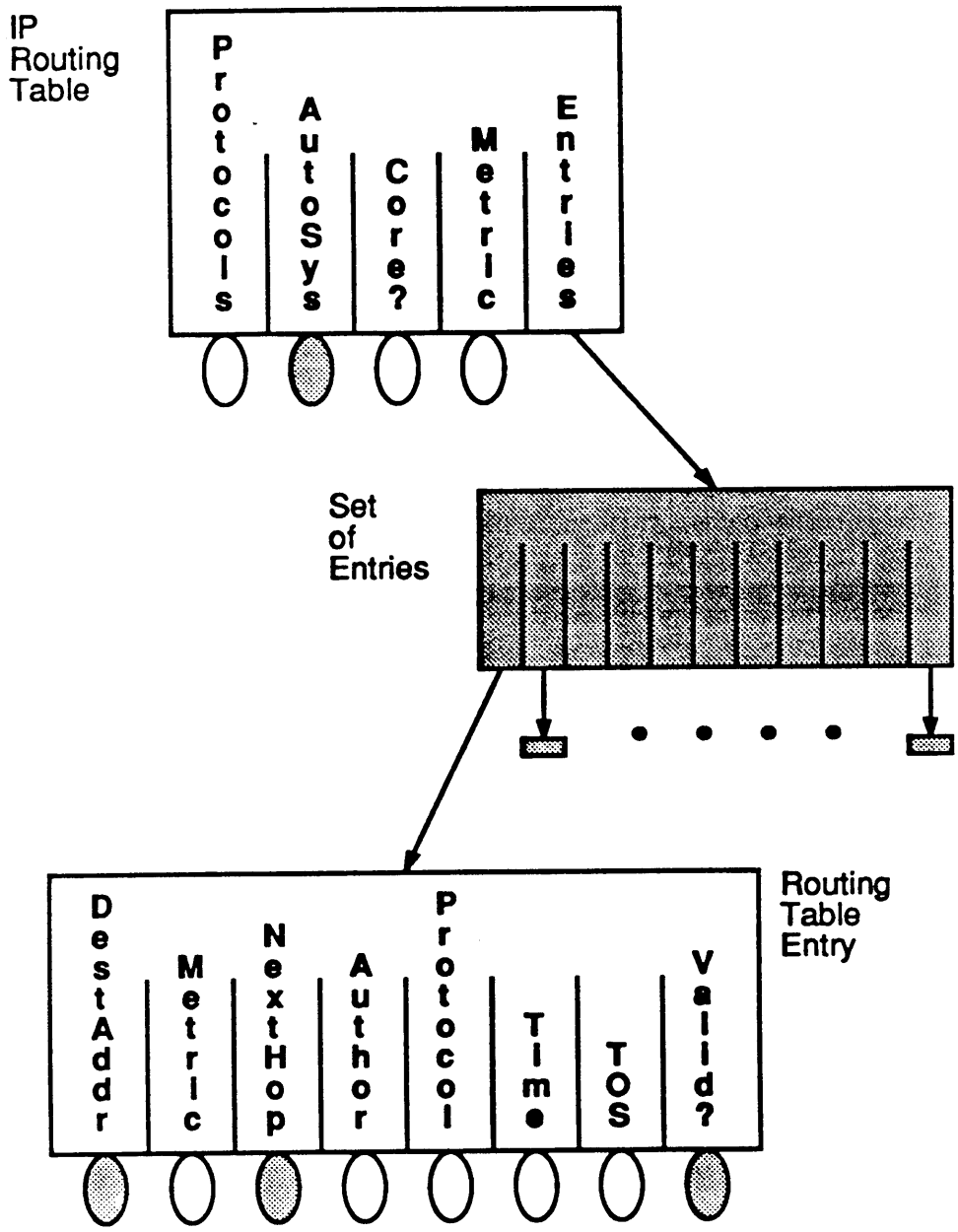
In general, a **template** can name multiple objects:

```
IPRouting{
    Entries{ Entry{ DestAddr, NextHop,
                    Valid? } } }
```

or even

```
IPRouting{
    AutoSys,
    Entries{ Entry{ DestAddr, NextHop,
                    Valid? } } }
```

Template Example



Dealing With Tables

Tabular data reists simple naming.

Conventional approach: index into table.

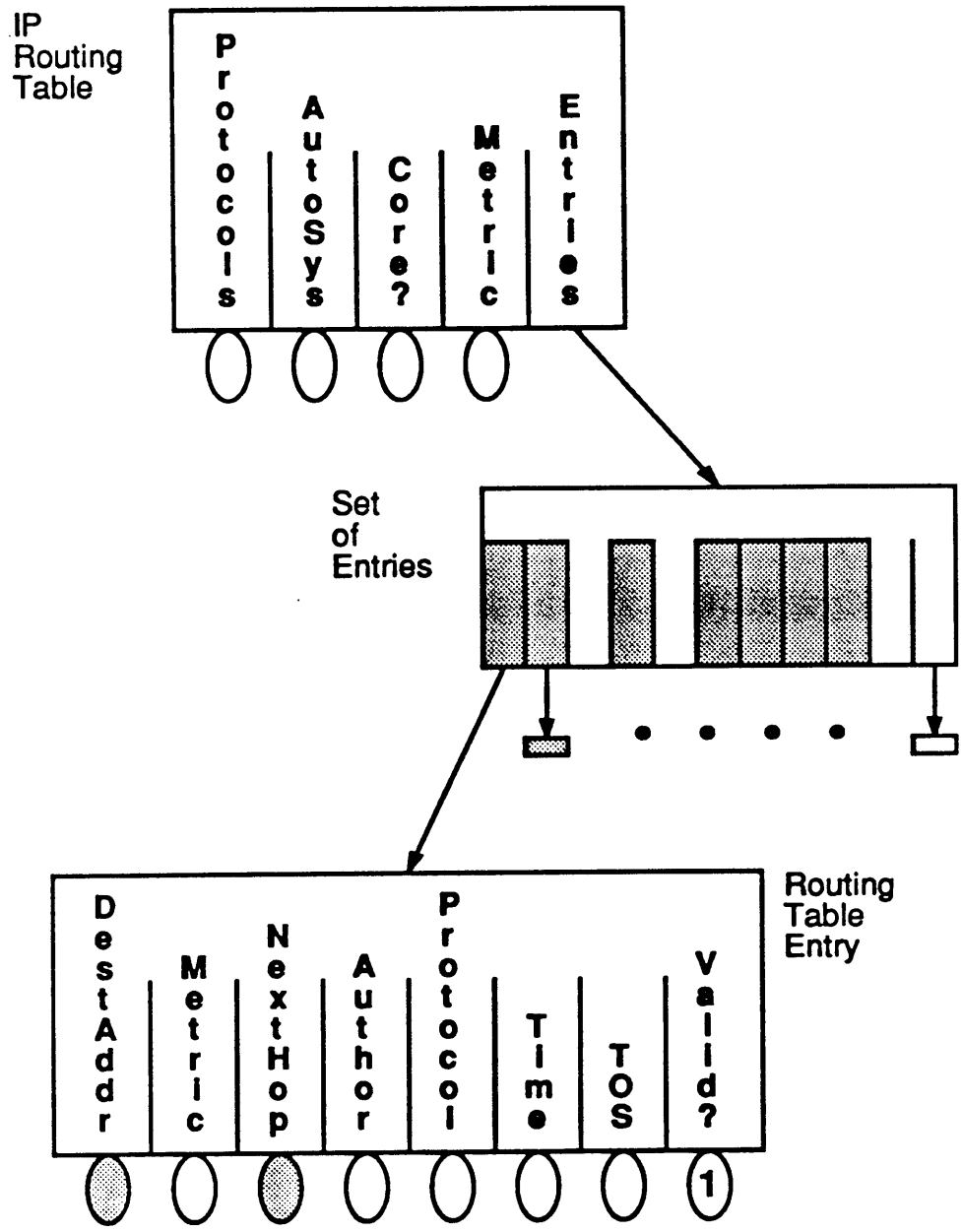
- Most tables have no useful index.

Generally want to access tabular data based upon some value in the table entry, e.g. IP Address, HopCount, etc.

For example, suppose we only want routing table entries that have the **Valid?** bit set?

The query language attempts a limited solution to this problem using **filters**.

Filtering Tabular Data



HEMS Query Language

Glenn Trewitt — Stanford University

Query Language can:

- Extract data from an entity.
- Modify (some) data in an entity.
- Perform control operations on an entity.

A **query** stands alone. It can be

- Sent in a HEMP message, generating an immediate response.
- Stored for later execution. *e.g.* when an exceptional event occurs.

A query is executed by a **query processor** running on the monitored entity.

Components of a Query

A query is composed of the following pieces:

template
: ASN.1 object naming some portion of the data tree. May be any "shape".

tag
: Same as template, but names only one object.

value
: ASN.1 object giving values for some portion of the data tree. May be any "shape". *i.e.* a template with values filled in.

opcode
: Command telling the query processor to do something.

filter
: Simple boolean expression used to select data from the tree.

Operation of Query Processor

Query processor is stack-based, driven by the sequence of ASN.1 objects in the input stream.

- All non-opcodes are pushed on the stack.
- Opcodes are executed immediately and take their arguments from the stack.
- Since incoming objects are tagged, recognition of opcodes is trivial.

The stack may contain, in addition to objects from the query, references to dictionaries.

Initially, the stack contains the root dictionary.

Query Language Operations

There are 8 operators in the language.

get Given a template, fill in the values from the data tree and return it.

set Given a value, set the data in the data tree to the supplied value(s). Not very many values are settable.

create Insert a value into a table.

delete Remove a value from a table.

Control operations are performed by **set**-ing data items that have side effects.

get-attributes
Return descriptive information about one or more objects named in a template.

get-range
Special hack to retrieve contents of memory.

begin Establish a naming context.

end Return to previous naming context.

Examples

Retrieve the autonomous system number from the routing information:

```
IPRouting{ AutoSys }
get
```

Retrieve some of the TCP Statistics:

```
IPTransport{ TCP{ Stats{
    octetsIn, octetsOut,
    inputPkts, outputPkts } } }
get
```

Another way:

```
IPTransport{ TCP } begin
Stats{ octetsIn, octetsOut,
       inputPkts, outputPkts }
get
end
```

Filters

The **get**, **set**, and **get-attributes** operators may take an additional **filter** argument to allow the data to be selected based upon values contained in a candidate object.

Filter ::=

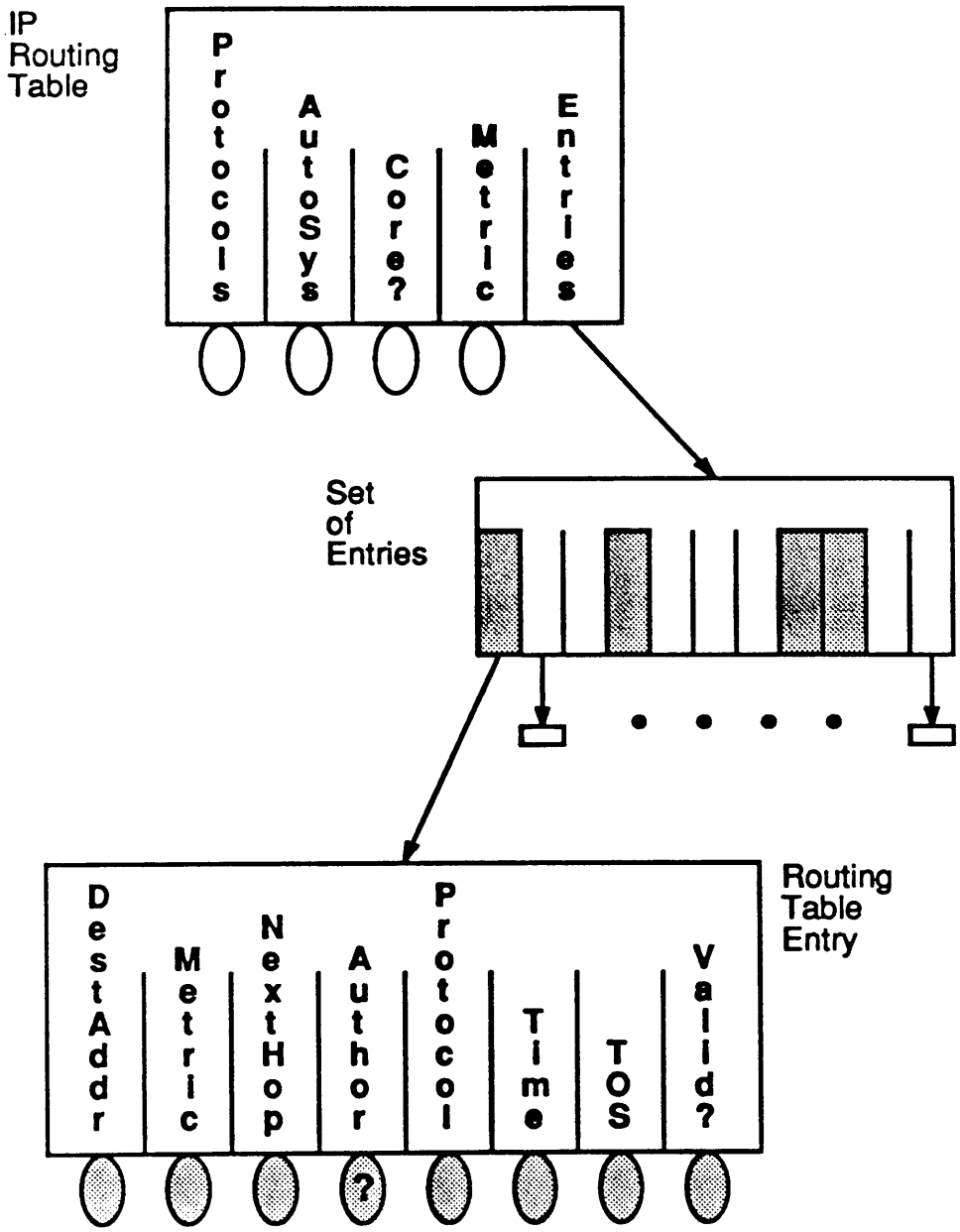
term
term AND term
term OR term
NOT term

term ::=

EQ value
GE value
LE value
EXISTS tag

Note that the binary operators EQ, GE, and LE get both the data name and data value from the same ASN.1 object.

Filtering Example



More Examples

Retrieve all routing table entries with an author of 10.2.0.2

```
IPRouting{ Entries } begin
entry
filter{ eq{
   entry{author(10.2.0.2)} } }
get
end
```

Shut down interface with address 10.1.0.11:

```
interfaces begin
interface{ status(down) }
filter{ eq{
   interface{address(10.1.0.11)} } }
set
end
```

Vendor-Specific Data

Most of the monitored data stored in an entity will be described in RFC-SSSS.

Vendor extensions are put in a special dictionary, **VendorSpecific**.

- Added to any already-defined dictionary.
- May be many, scattered through the data tree.

However, the meaning of this data will be a mystery. Two solutions:

- Always have the vendor's manual handy (correct version, too).
- Ask the entity what it means.

Data Attributes

Attribute information for each node in the tree.

- Retrieved with **get-attributes** operator.
- Often will be boilerplate.
- Useful for VendorSpecific data.

Attributes structure contains:

tag	item being described
format	ASN.1 format of data
longDesc	long string description
shortDesc	string label
unitsDesc	units (string)
properties	bitstring (*e.g.* differenceable)

A clever application could figure out:

- How to display and label the data
- Whether to subtract samples
- Full description on demand

HEMS and ISO CMIS

Glenn Trewitt — Stanford University

ISO specifies:

CMIS Common Management Information Service
(service definition)

CMIP Common Management Information Protocol

layered on top of:

ROS Remote Operations Services
(remote procedure call facility)

Layering

Major difference is how the pieces are layered.

Service	ISO	HEMS
Encapsulation	ROS	HEMP
Security	CMIP	HEMP
Operations	CMIP	QL
Grouped operations	ROS	QL

CMIP ops are much more heavyweight than QL ops.

- Each is encapsualted separately.
- Each has RPC overhead: sequence #, authentication.
- Grouped ops require RPC header *per-operation.*

Data Model

Both use tree structure.

CMIS has no notion of **template**:

- Uses lists of pathnames.
- Mechanical mapping is possible.

Common Service Definition

The services provided are essentially identical, however.

HEMS	ISO
get	M-GET
set	M-SET
create	M-CREATE
delete	M-DELETE
(set)	M-ASSOCIATE
begin	(CMIP)
get-attributes	xxx
get-range	xxx

Long Term Routing Issues Hinden (BBN)

LONG TERM ROUTING WORKING GROUP

NOT EGP, EGP2, RIP, GGP, ...

OPEN ROUTING W.G.

Autonomous System ⇔ Autonomous System

Domain ⇔ Domain

Define the Problem

or

What Problem to Solve?

Topology ?

Tree, Graph

Size ?

Networks, # A.S., # Switches, # Neighbors

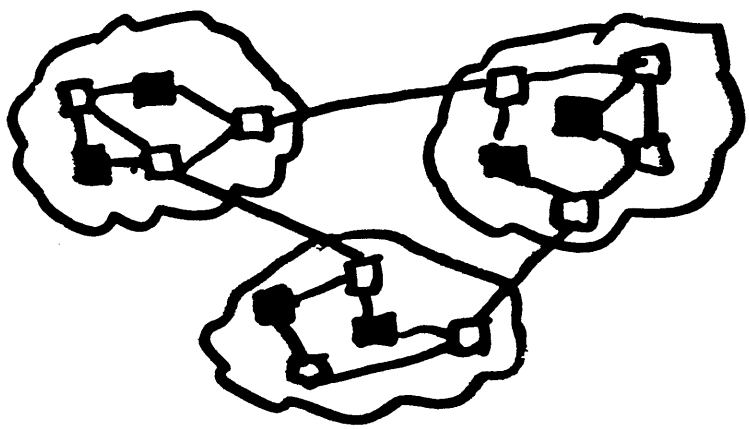
Structure ?

Flat, Multi-Level (# Levels)

Addressing ?

Current, New

Data Flow



Partitions of Autonomous Systems ?

All Gateways have All Routes ?

Ask Routing Server
Forward to Higher Level Gateway

Type of Service ?

How many ?

Multipath ?

Relationship to Congestion Control

Access Control ?

Limit Routes to Authorized Users

Security

Authentication
Firewalls

Moble Hosts ?

Logical Addressing

Disaster Perfolmance ?

When?

Now, 2 Years, Last Year

How to Specify and Engineer?

Testing
M&C
Certification

Goals of Working Group

1) Defination of Problem
2) Architectural Model for Solution(s)
3) Survey of Existing Routing Algorithms

TTL of W.G.

12-18 Months

Arpanet Status Report

What made it better?

1. New cross-country line
2. New routing metric

The Vsat saga

Prologue

- additional cross-country trunking recommended — 1985

1st act

- Severe congestion 9/86
- Line ordered 10/86
  VSAT to be operational 11/86

2nd act

Complications spanning 7 mont

eg MIT took 2 months to decide if roof could hold dish

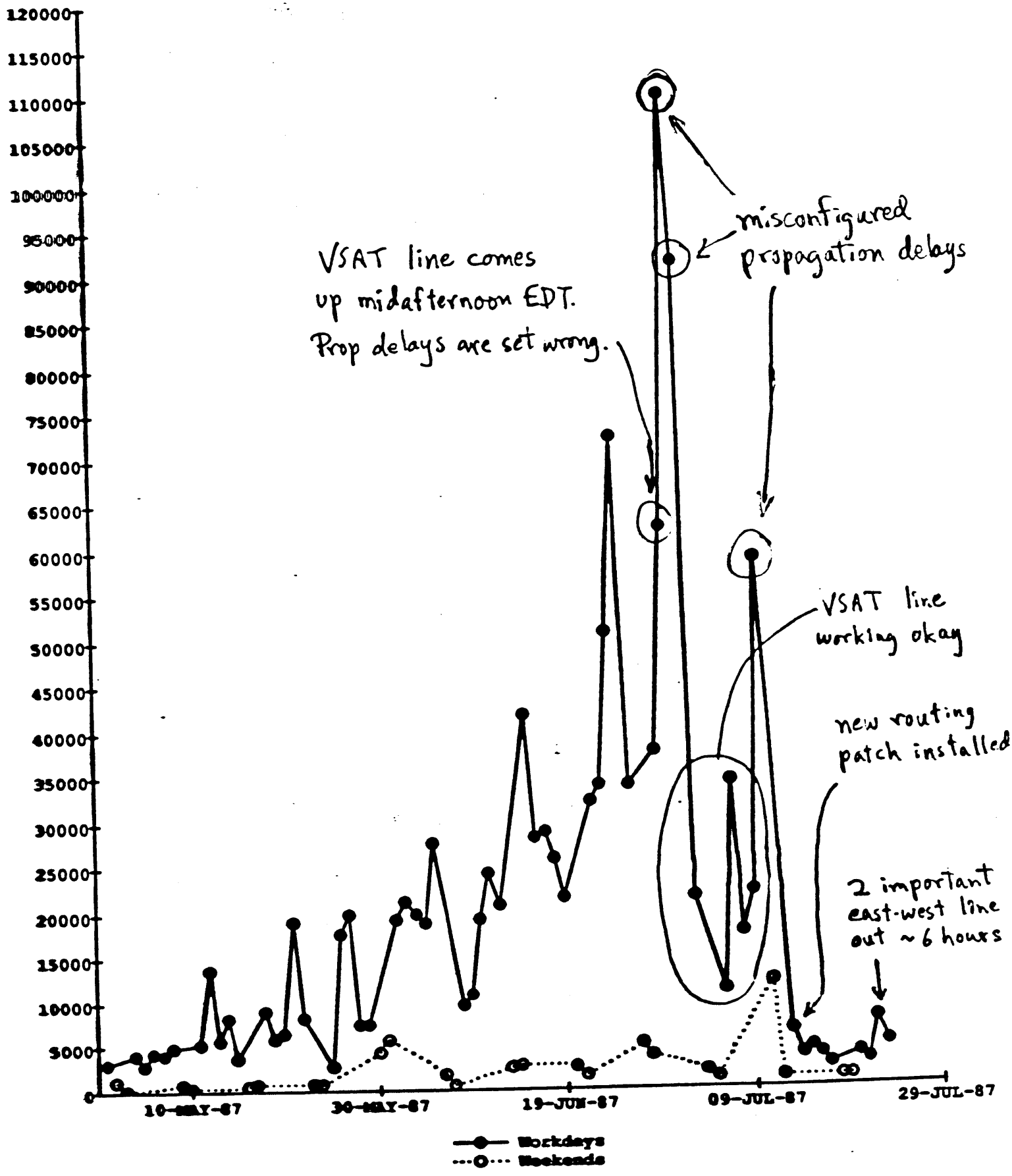
3rd act

- VSAT line installed 30/6/87
- misconfigured
- fixed
- imp restart a misconfigured
- fixed

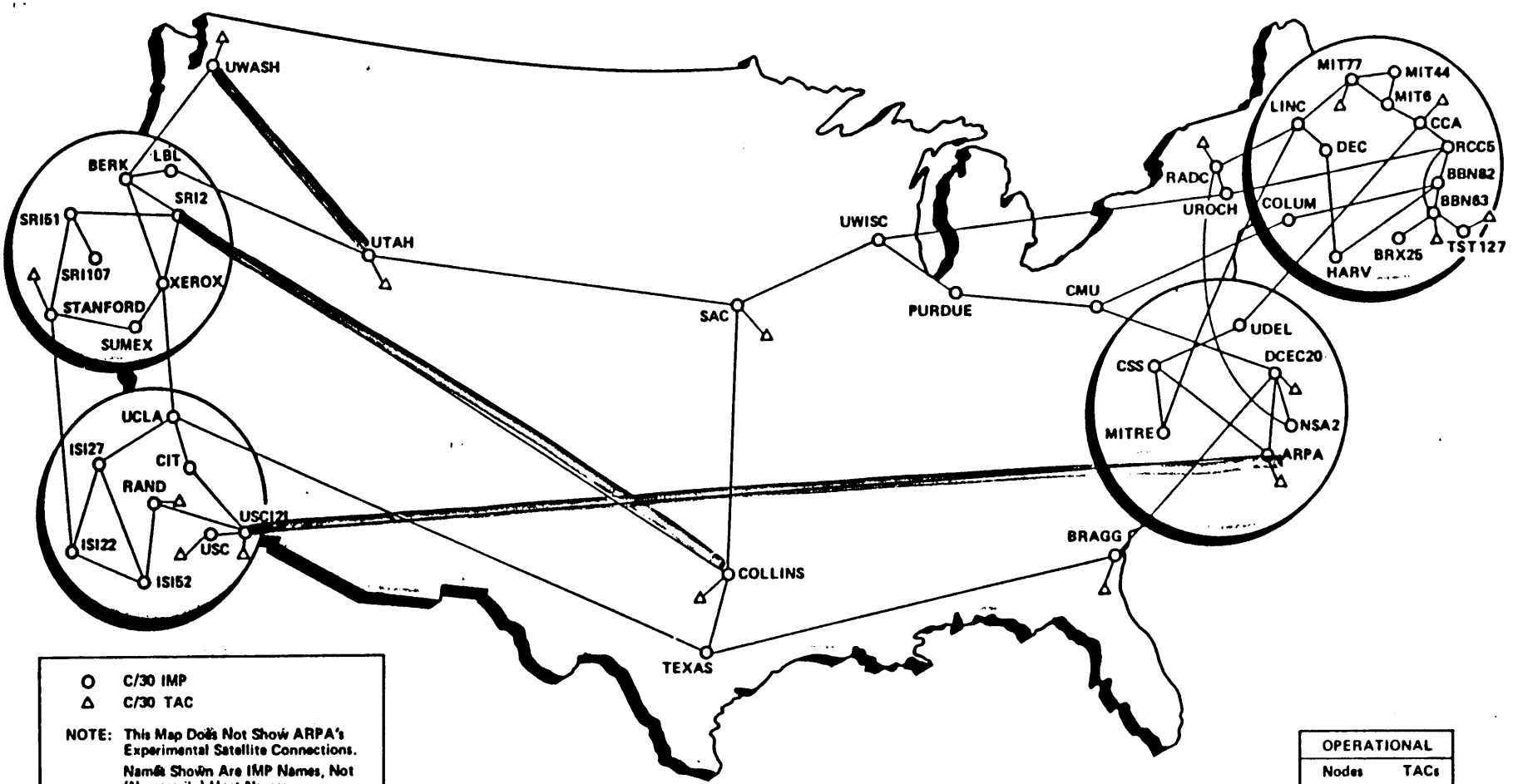
epilogue

- performance improvements
- Terrestrial line on order (still)

Performance-Related Traps per Day

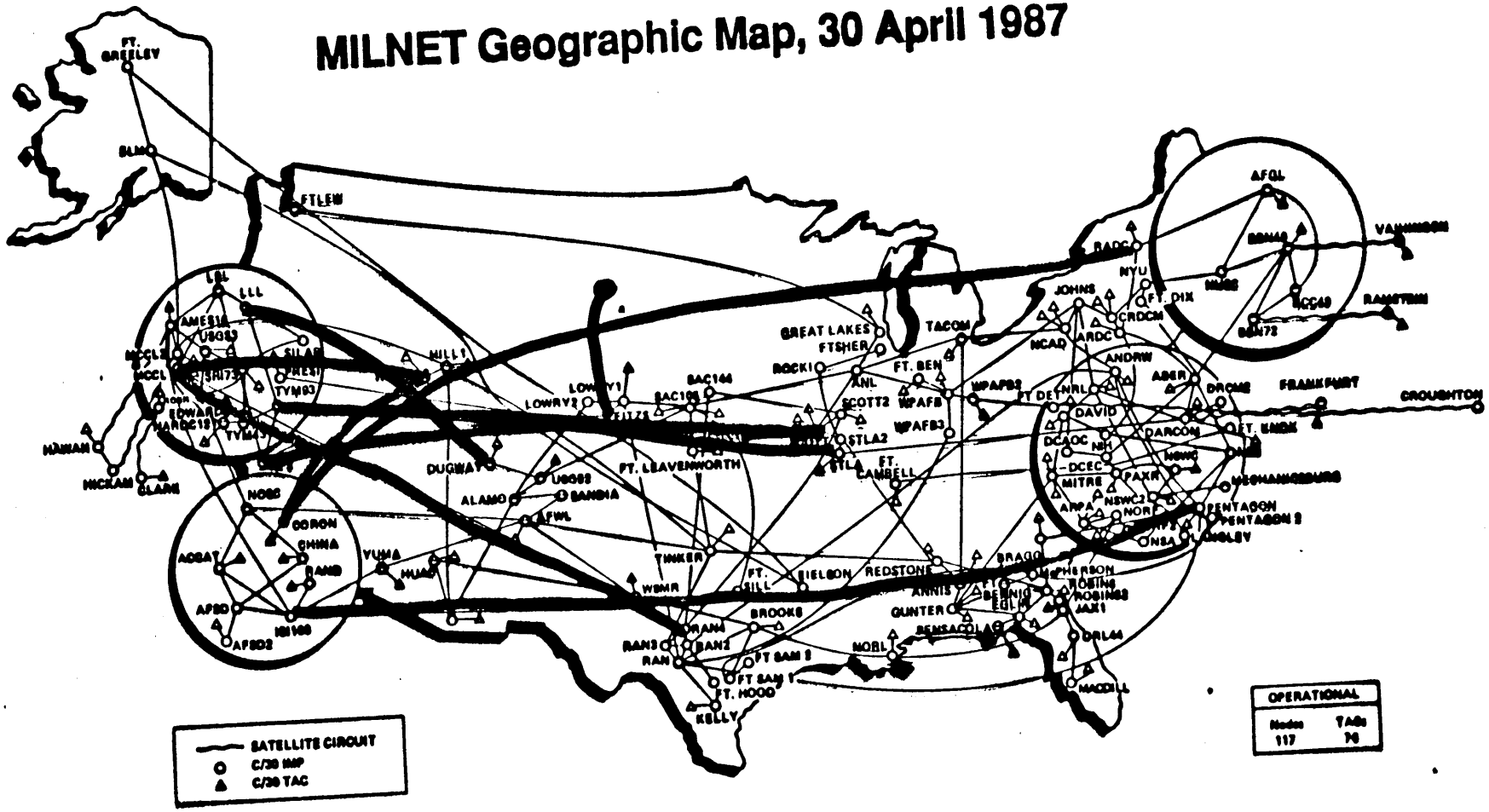


ARPANET Geographic Map, 30 April 1987



22-Jul-87
14:31-20:22 (GMT)

MILNET Geographic Map, 30 April 1987



CURRENT SPF METRIC (1)

- Packet delay measured by PSN
- Averaged over 10 second intervals
- Reported potentially every 10 seconds (depending on result of threshold check)

CURRENT SPF METRIC (2)

Current metric/algorithm fine under lightly loaded conditions — reported delay is a fairly good predictor of the delay to be expected after re-routing based on reported value. Why?

Under light loading:

1. Link delay is essentially
   transmission delay + *propagation delay*
2. A change in reported delay results in small changes in flow. Thus, delay remains in range where *transmission delay* + *propagation delay* term dominates

CURRENT SPF METRIC (3)

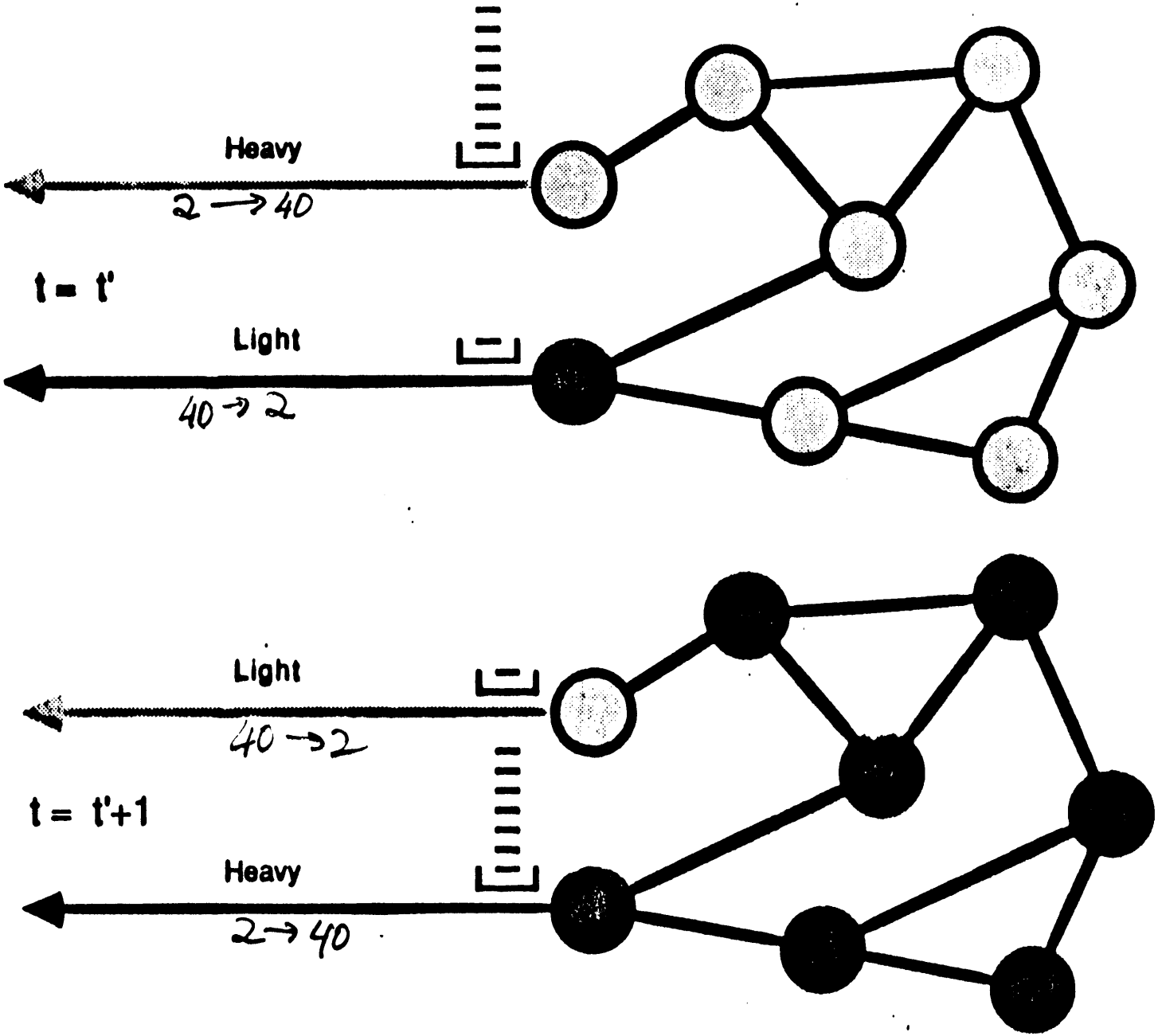
"Breaks down" under heavy loading

Culprits:

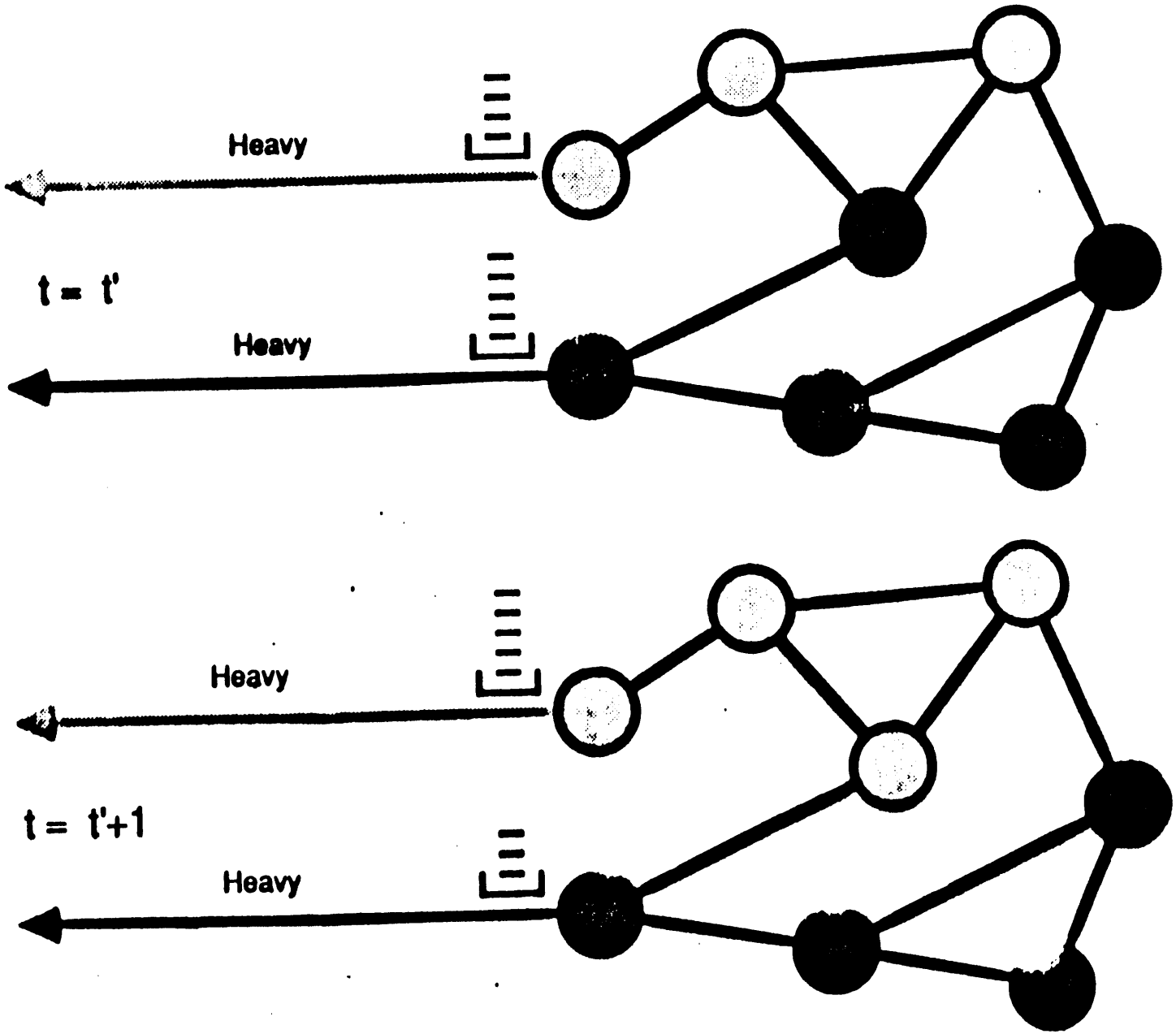
1. Too large a range of reported delay values, **e.g.**
   $254/28 \approx 9$ for 9.6 Kb network
   $40/2 = 20$ for 50 Kb network
   $254/2 = 127$ for mixed network
   Link reporting *high* looks unattractive to almost all sources — 127 link path can look more attractive than 1 link path!
2. Reported value allowed to change by too much.
3. Nodes react (change flows) simultaneously.

Network with large number of traffic sources operating in region where queueing delay significant (i.e., heavily loaded network), capitalizing on 1., 2. and 3. $\Longrightarrow$ OSCILLATION

SPF Current Behavior Under Overload

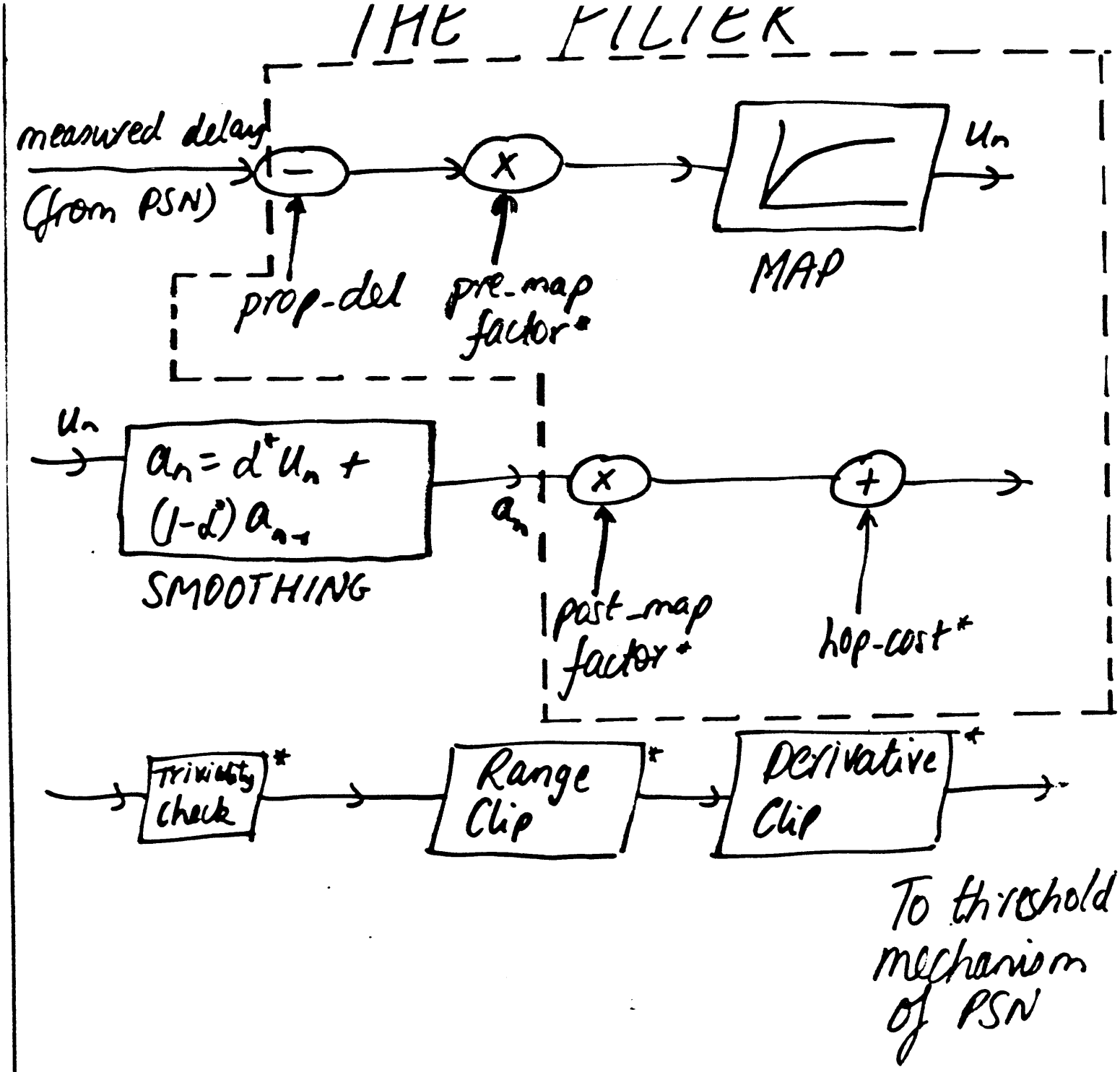


SPF Desired Behavior Under Overload



New metric

- history
- clips
- restrict speed of change (reduce the derivative)

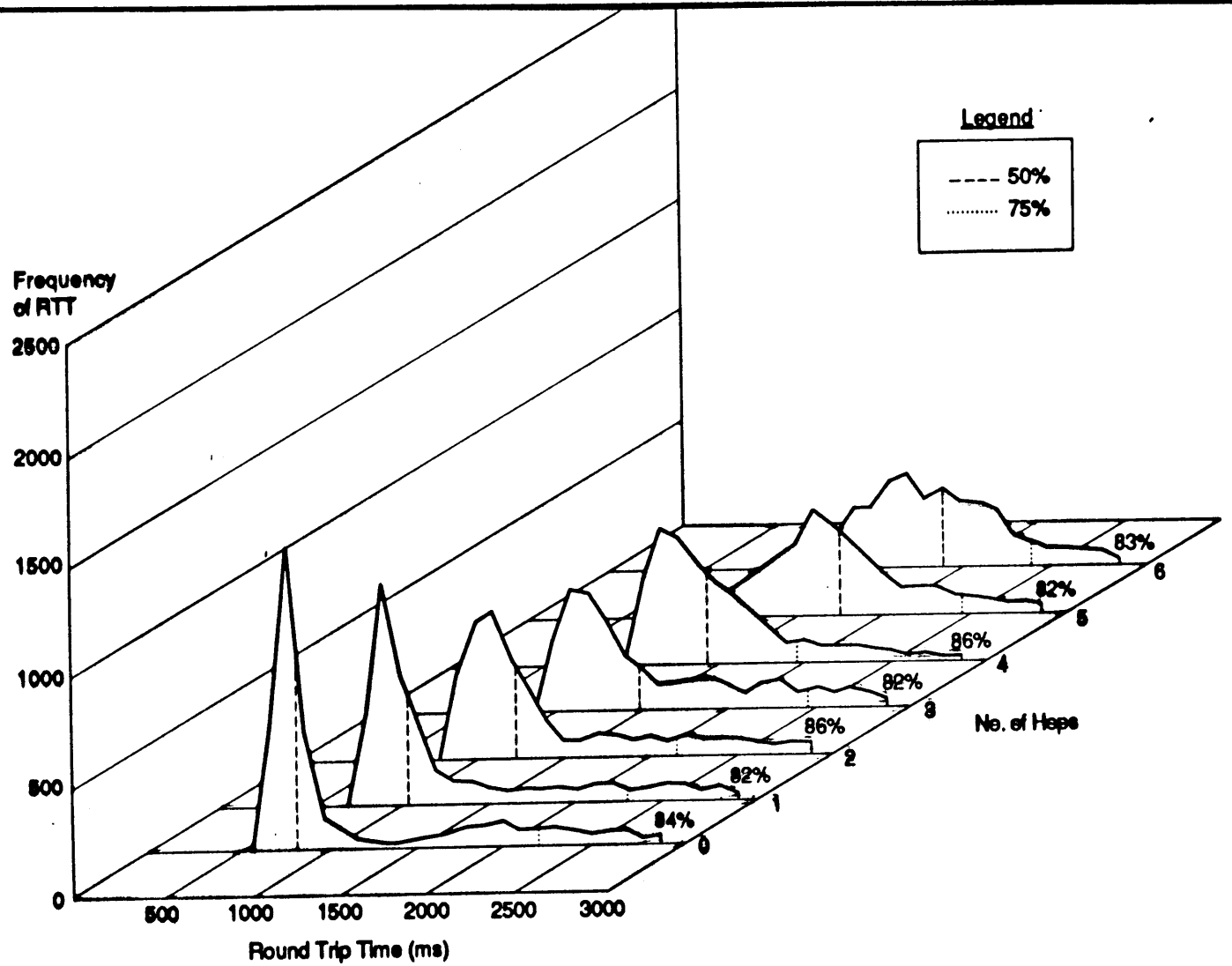


* Based on line-speed

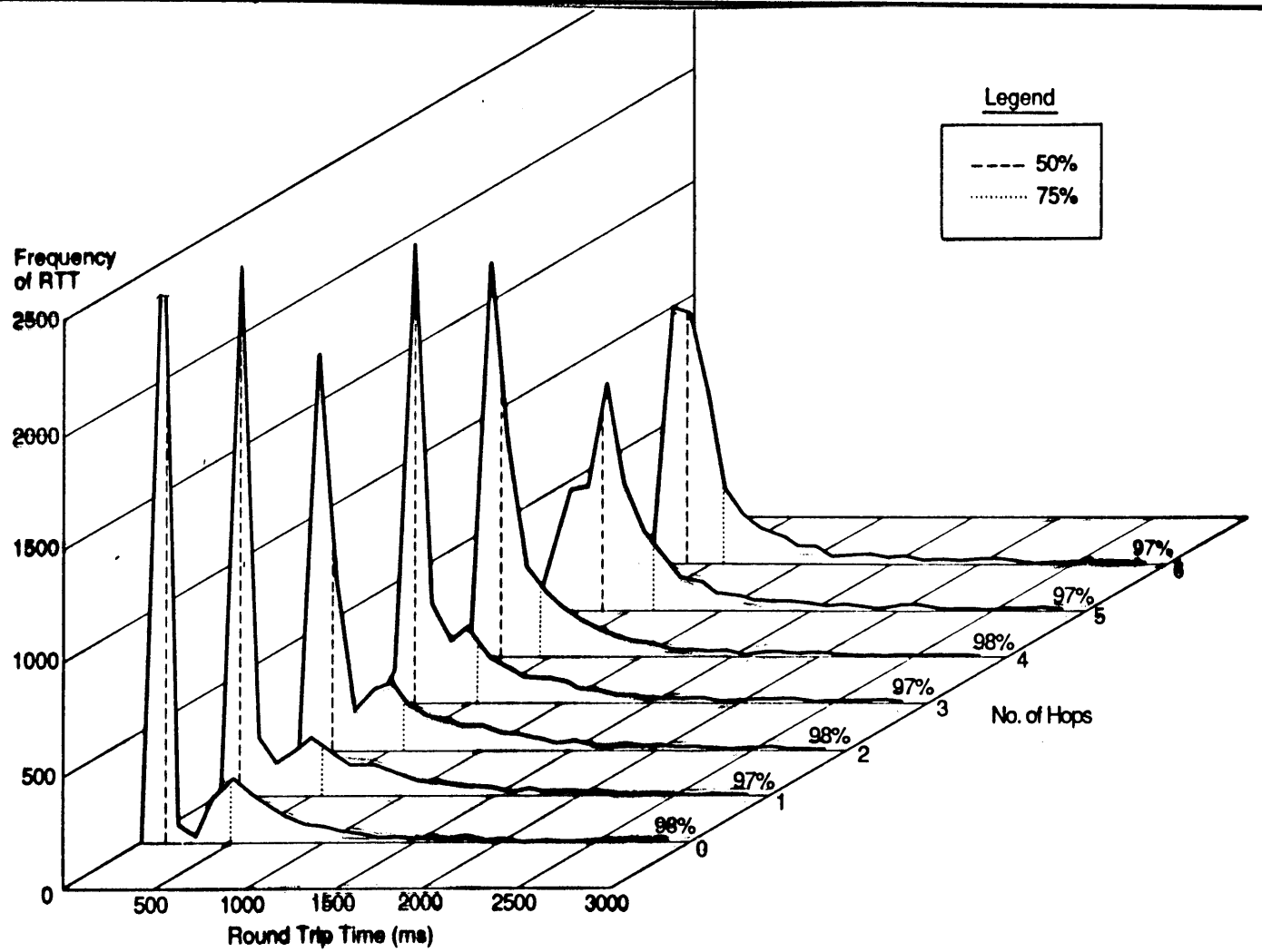
Arpanet Performance Measurements

- Results of the Arpanet Baseline measurements
- Parameters:
  - Three interfaces (X.25, HDH, 1822)
  - Packet length (long, short)
  - Interpacket gap (short, long)
  - Traffic period (high traffic, low traffic
- Timeframe- Before recent improvements
  - Currently repeating measurements
- Graphs which follow show measurements with:
  - Long interpacket gap
  - Low traffic period

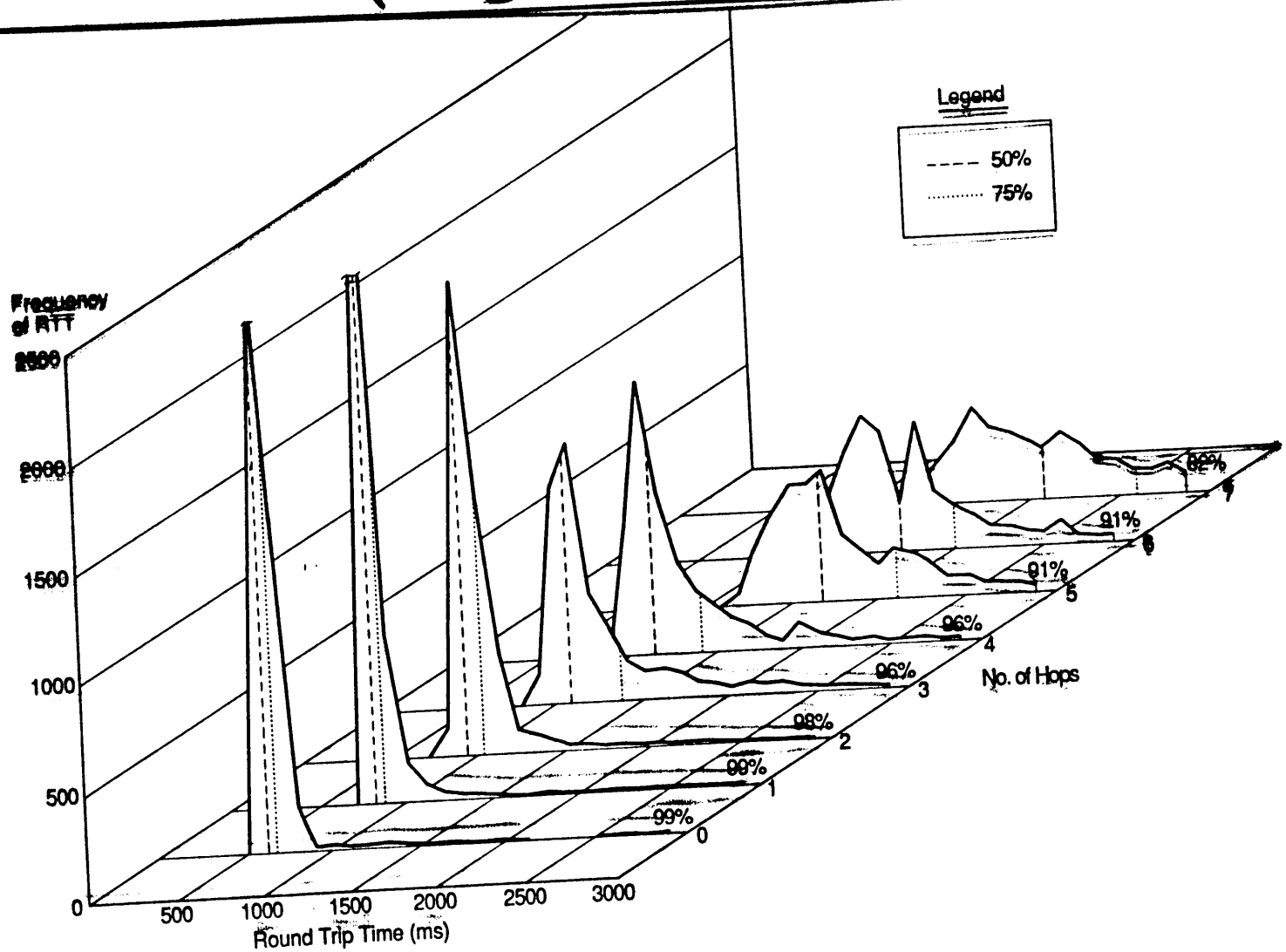
X.25- RTT Distribution over Hops (Long Packets)



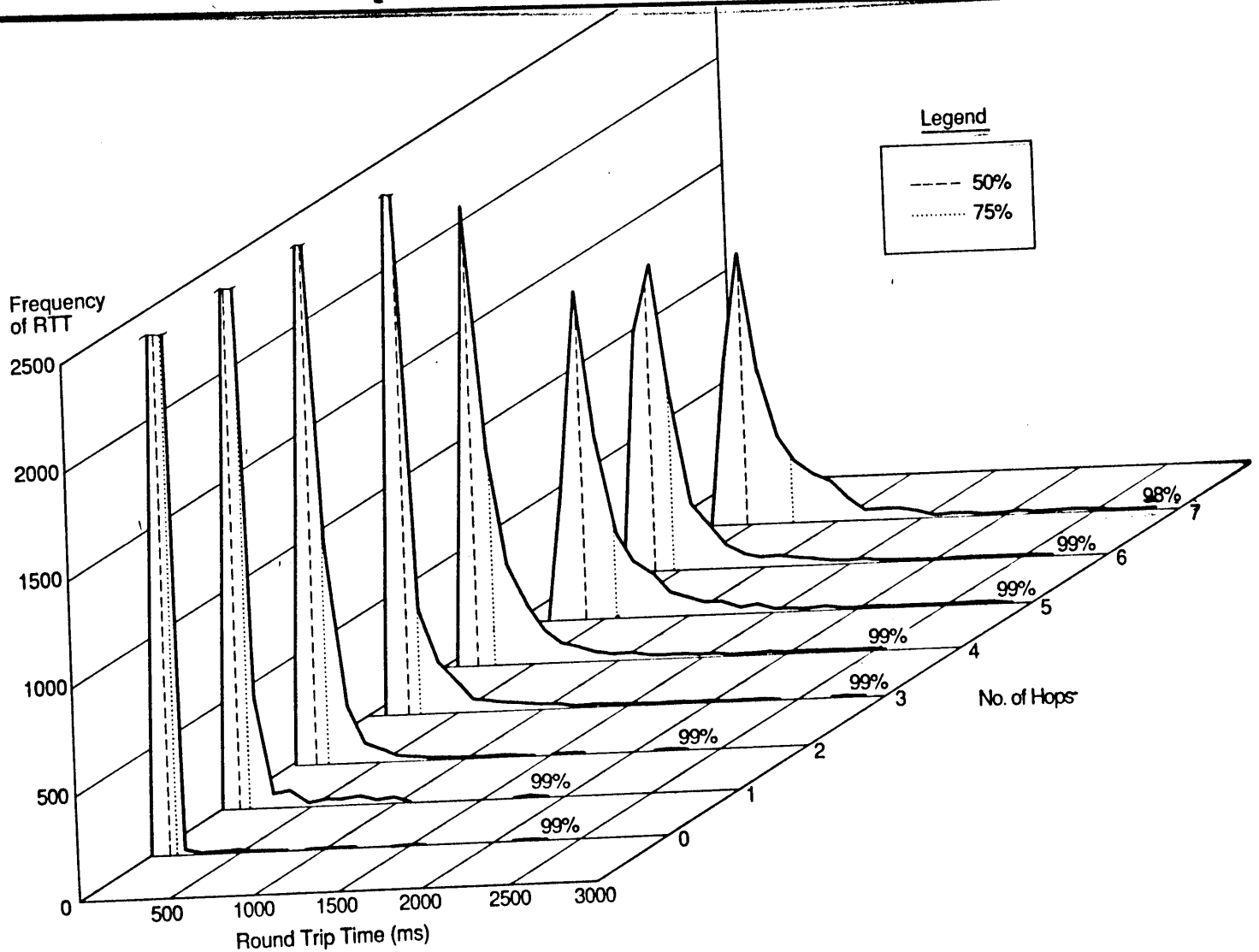
X.25- RTT Distribution over Hops (Short Packets)



HDH- RTT Distribution over Hops (Long Packets)



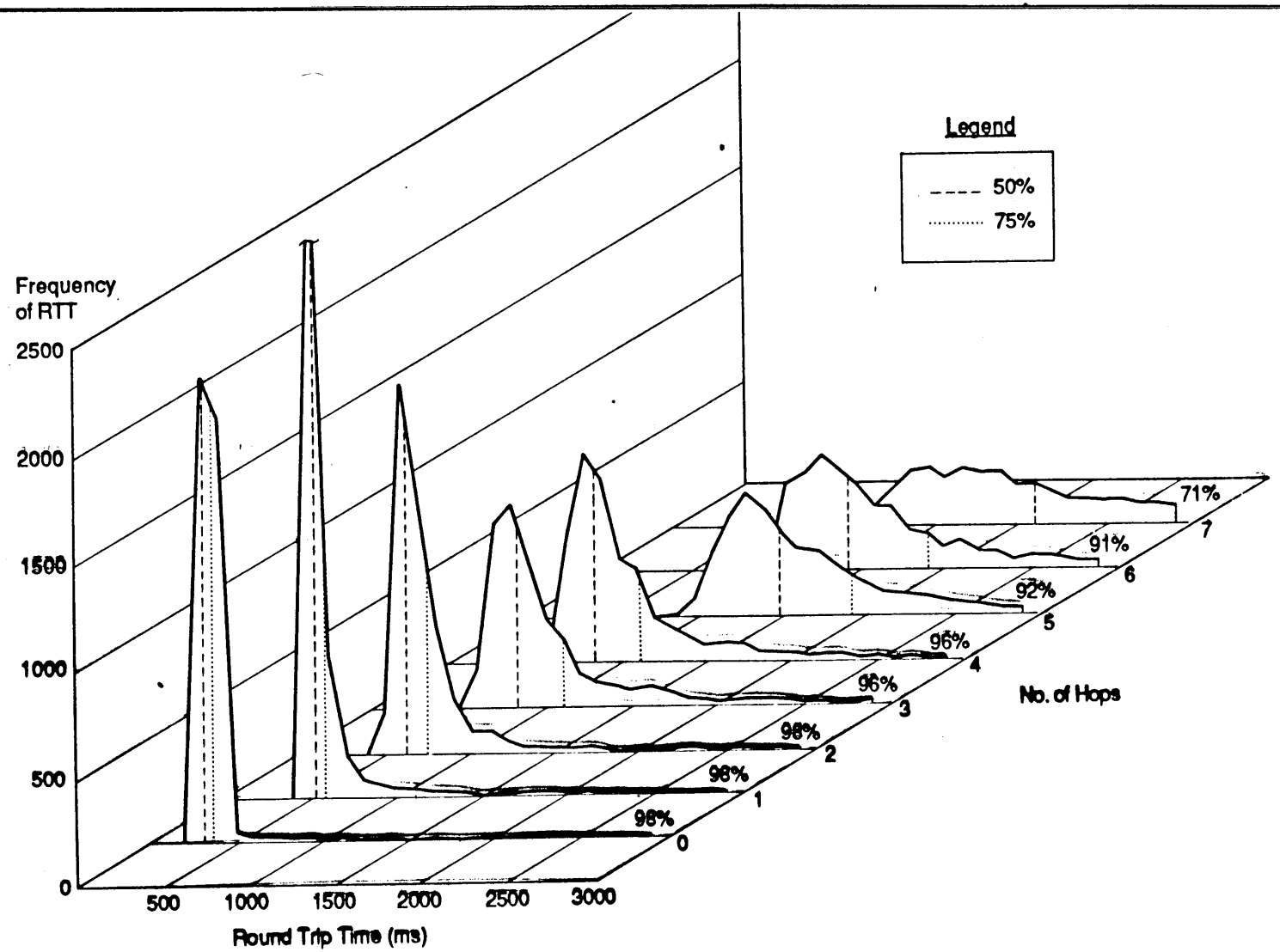
HDH- RTT Distribution over Hops (Short Packets)



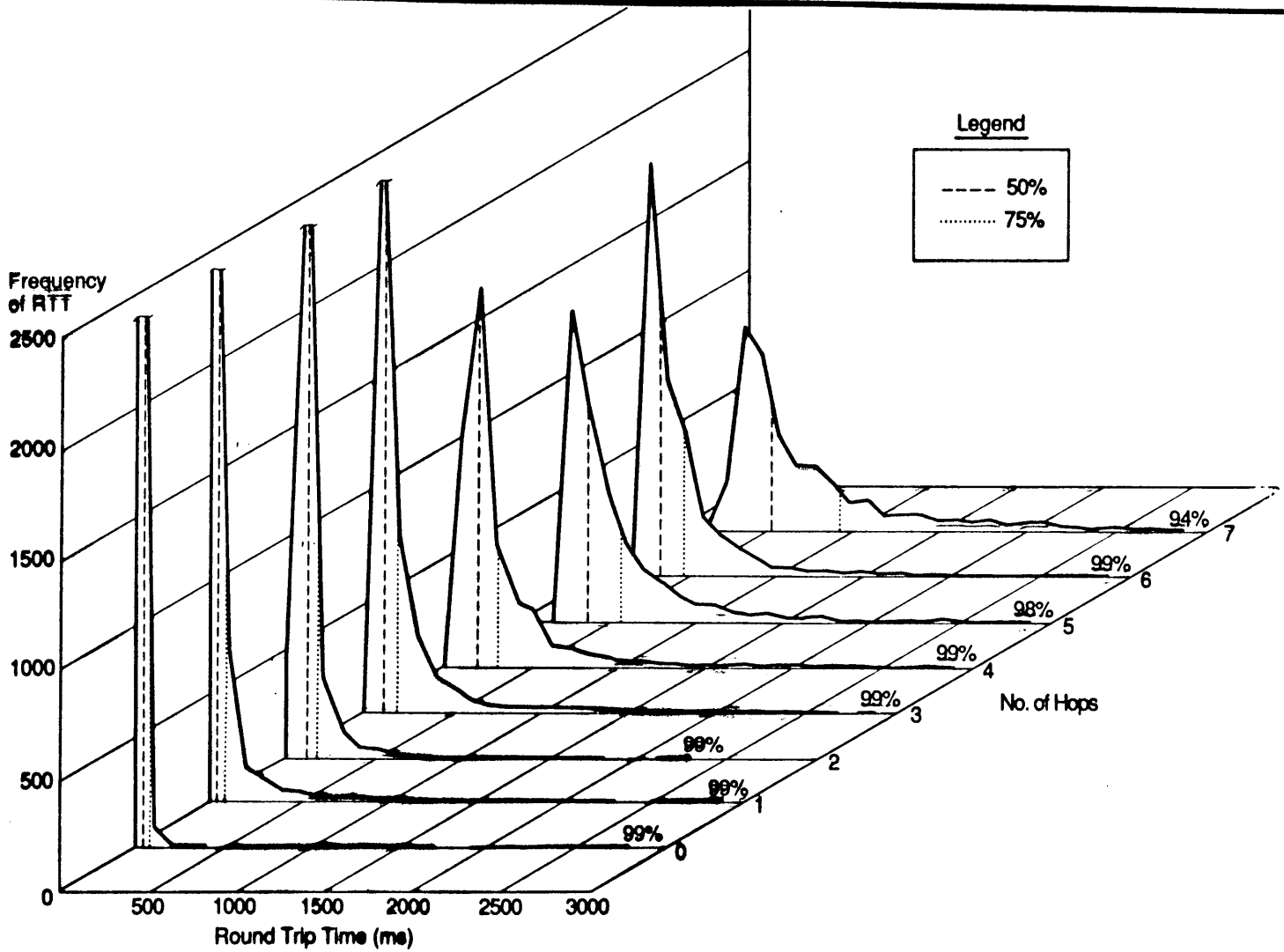
MITRE Corp.

P.Gross, R.Coltun

1822- RTT Distribution over Hops (Long Packets)



1822- RTT Distribution over Hops (Short Packets)



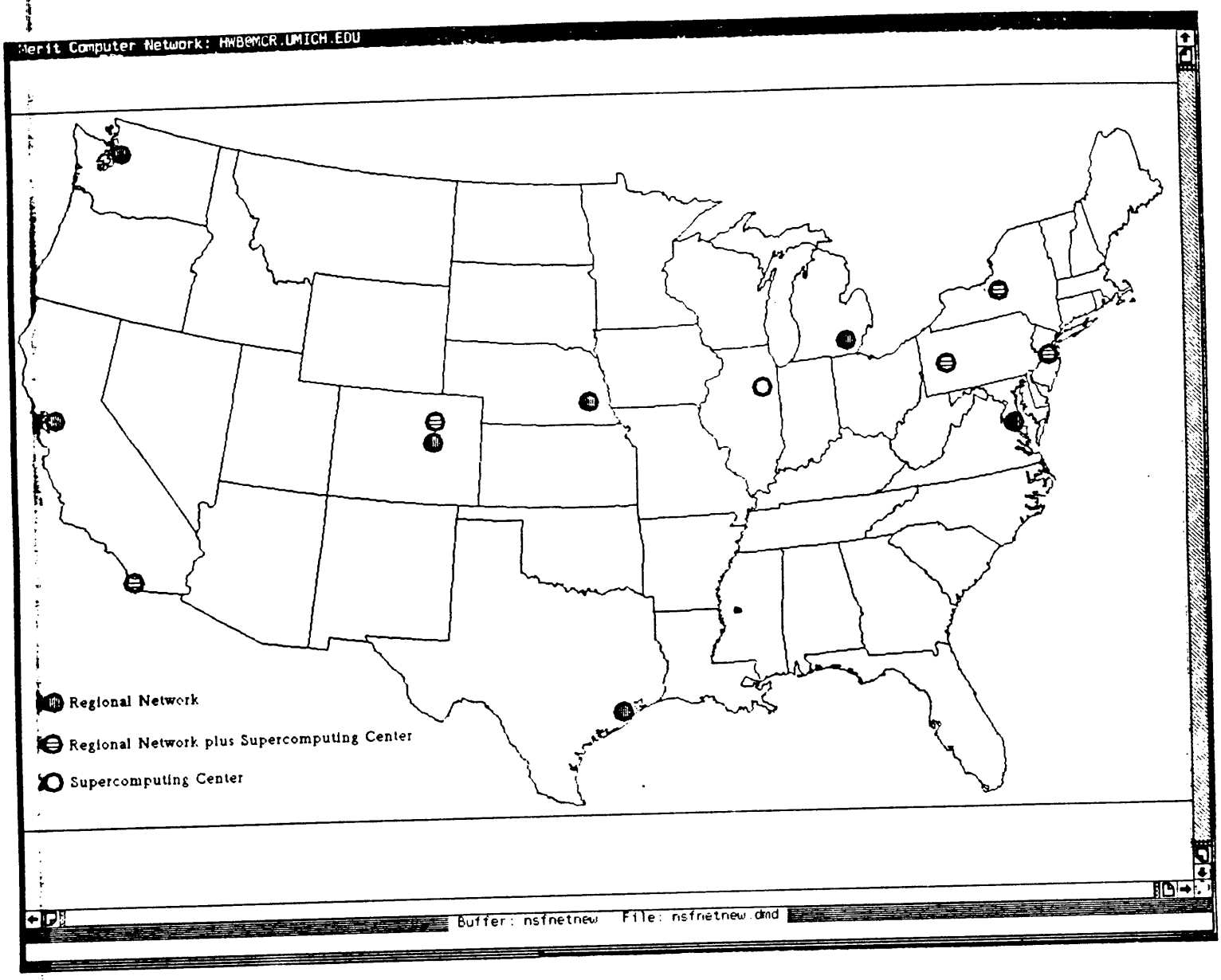
Preliminary Data

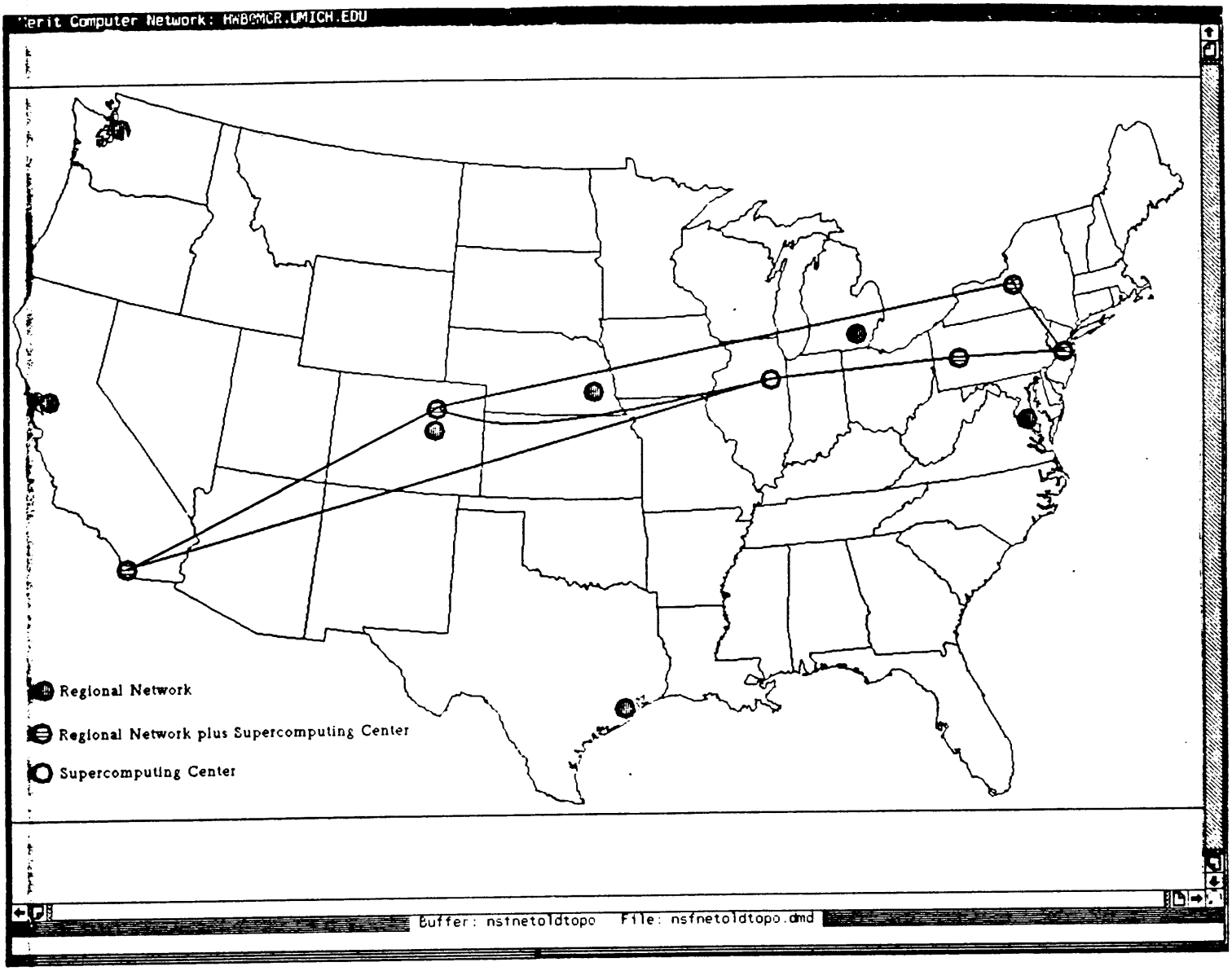
- Distribution of RTT's

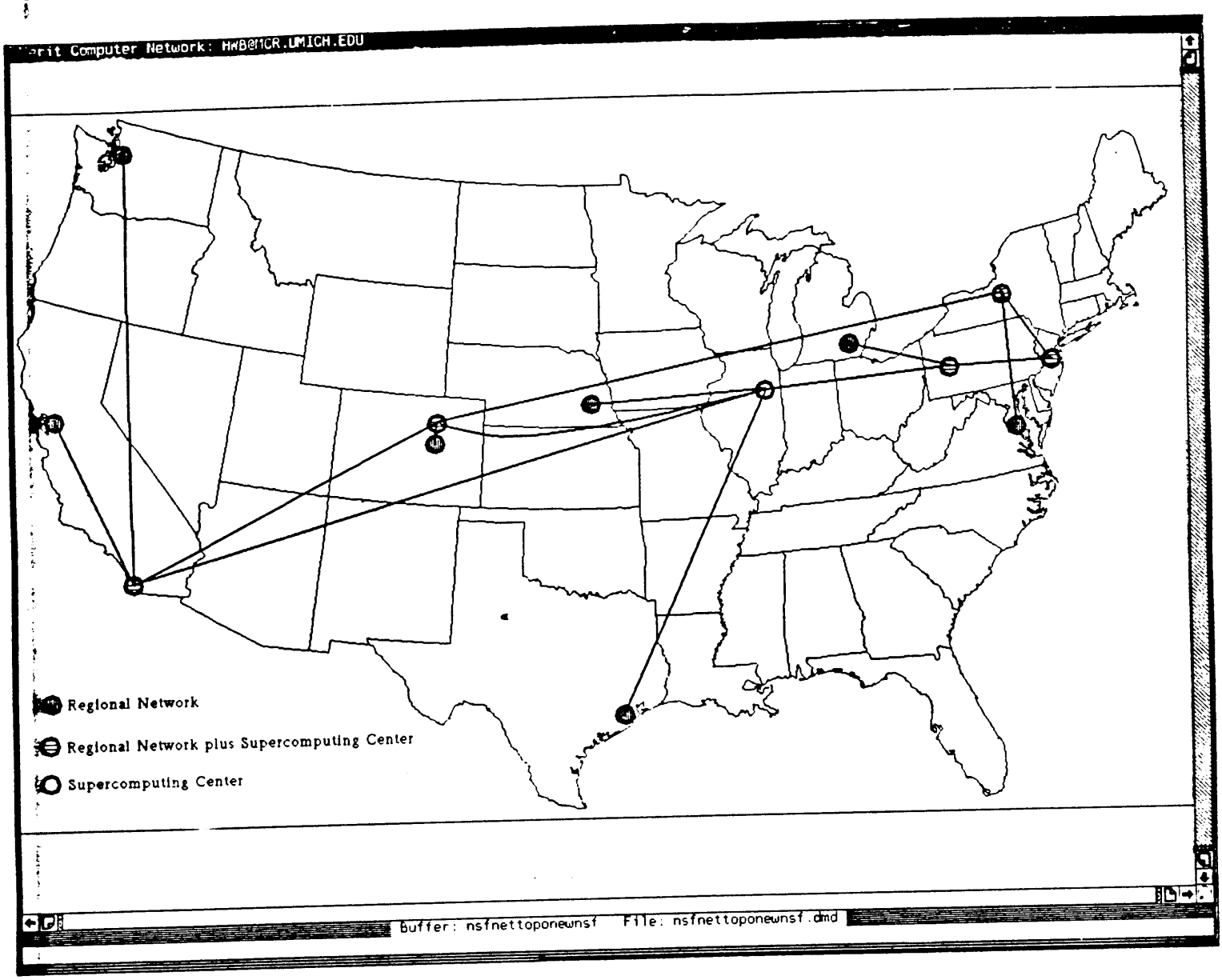
- Parameters vs. hops
  - Median
  - Variation
  - Skew

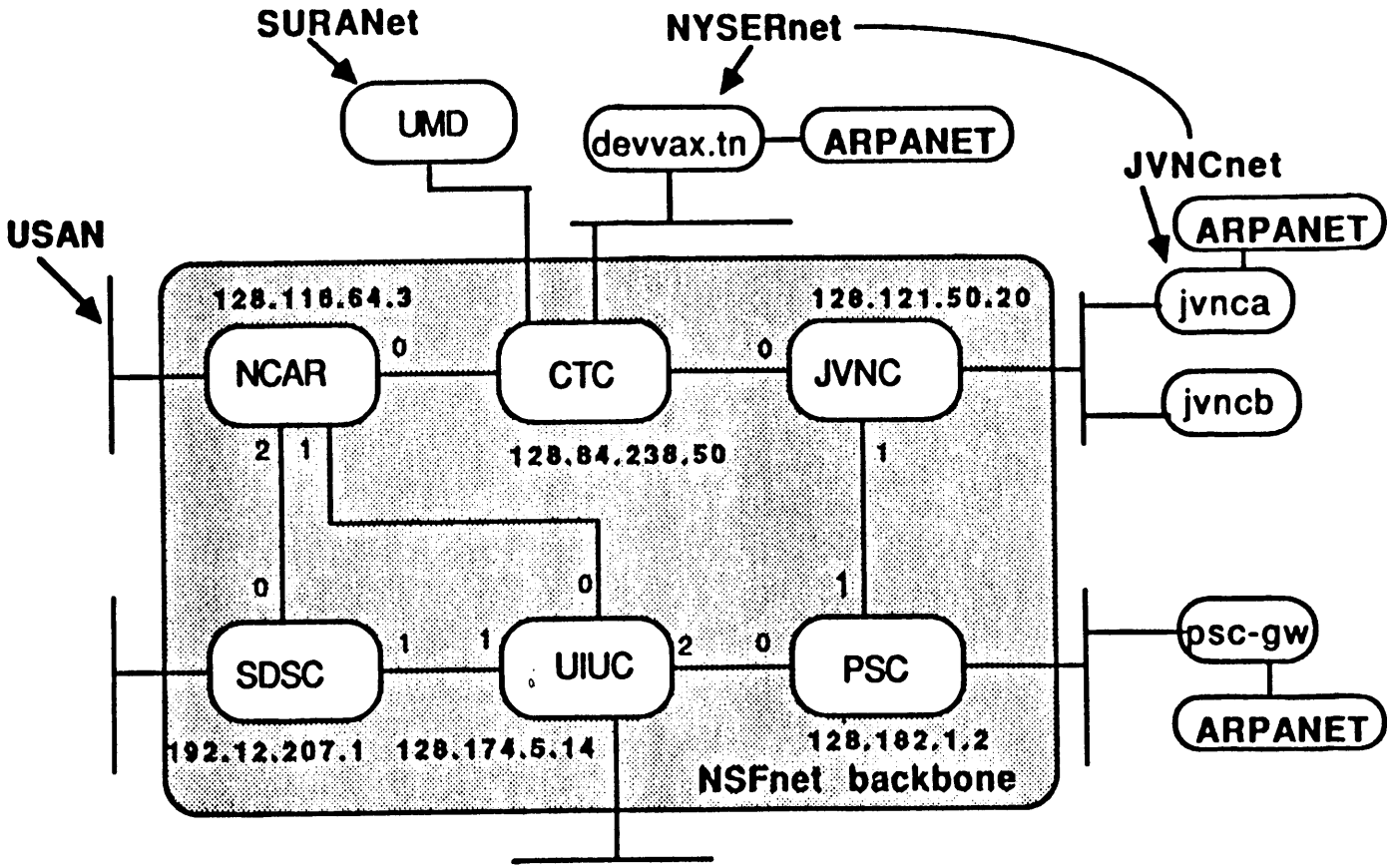
- Short term variation over single measurement
  - Median
  - single trace

NSFnet Status Report Braun (UMich), Elias (Cornell)


Merit Computer Network: HWB@MCR.UMICH.EDU
Regional Network
Regional Network plus Supercomputing Center
Supercomputing Center
Buffer: nsfnetnew File: nsfnetnew.dmd


Merit Computer Network: HAB@MCR.UMICH.EDU
Regional Network
Regional Network plus Supercomputing Center
Supercomputing Center
Buffer: nsfnetoldtopo
File: nsfnetoldtopo.dmd


rit Computer Network: HWB@MICR.UMICH.EDU
Regional Network
Regional Network plus Supercomputing Center
Supercomputing Center
Buffer: nsfnettoponewnsf
File: nsfnettoponewnsf.dmd



NSFnet Backbone Configuration

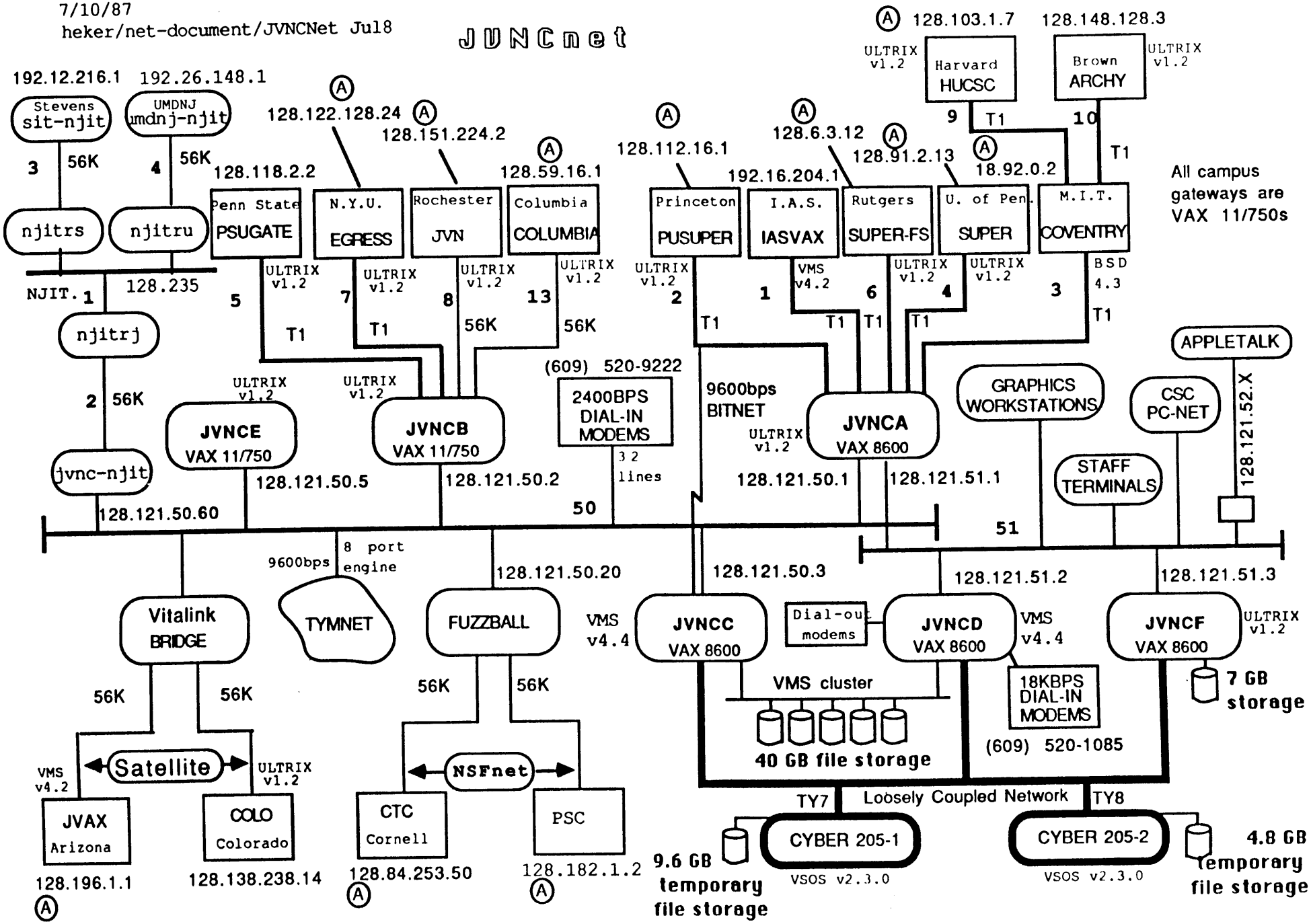
S. Heker
5/26/87
heker/nsfnet/NSFnet backbone config

The John von Neumann National Supercomputer Center

Ⓐ - ARPANET CONNECTION

7/10/87
heker/net-document/JVNCNet Jul8

JVNCnet



NSFNET TRAFFIC REPORT Period:7/13 - 7/19, '87

Significant Backbone Events:

7/14/87 Frozen servers on SURANet gateway. Inability to telnet or ftp into router, but traffic and stats not affected.

7/15/87 NCAR<->UIUC link disabled at 1430Z for BERT testing by NCAR. Test shows drop of carrier.

7/16/87 A series of tests done by AT&T on the NCAR<->UIUC link show that the line is fine.

7/17/87 UIUC switches local hardware in order to isolate NCAR<->UIUC line problem. Link enabled at 1430Z. Investigation continues.

7/15/87 - 7/17/87 Data collection base machine crashed at Cornell, badly skewing the next reading.

Total Traffic Figures

	Between Sites	Ethernet
Input	19749772	12809585
Output	20224230	13560394
In+Out	39974002	26369979

Grand 66343981

Traffic Delivered to LANs

	min	mean	max	total	%	
CMU	4293	27299.80	59506	3494374	23.70	**
Cornell	2036	14917.10	32915	1909389	12.95	**
JvNC	3539	37113.94	98031	4676357	31.72	**
NCAR	2416	13830.36	24566	1770286	12.01	**
SDSC	230	4284.66	11746	548437	3.72	**
SURA	875	9240.66	23836	1182804	8.02	**
UIUC	787	9074.62	26984	1161551	7.88	*
Overall				14743198	100.00	

"**" indicates that the reported mean-value is artifically elevated due to missing observations which skew the following reading. These values are probably 5-10% higher than actual.

Site Traffic Percentages of Grand

	%INPUT	%OUTPUT		%LINK
PSC				
UIUC	2.60	3.26		5.86
JvNC	3.93	5.11		9.04
Ether	6.63	5.27		11.90
Totals	13.16	13.64		
			%SITE	26.80

Cornell					
NCAR		2.03	1.69		3.72
JvNC		3.37	1.96		5.33
SURA		1.75	1.78		3.53
Ether		1.00	2.88		3.88
Totals		8.15	8.31		
				%SITE	16.46
JvNC					
Cornell		1.91	3.42		5.32
PSC		5.05	3.97		9.02
Ether		6.65	7.05		13.70
Totals		13.60	14.44		
				%SITE	28.04
NCAR					
Cornell		1.63	2.08		3.72
UIUC		1.75	1.13		2.88
SDSC		0.48	0.44		0.92
Ether		2.20	2.67		4.87
Totals		6.07	6.31		
				%SITE	12.38
SDSC					
NCAR		0.38	0.54		0.92
UIUC		0.64	0.48		1.13
Ether		0.61	0.83		1.44
Totals		1.64	1.84		
				%SITE	3.48
UIUC					
NCAR		1.07	1.74		2.81
SDSC		0.47	0.67		1.14
PSC		2.71	2.21		4.92
Ether		2.21	1.75		3.96
Totals		6.46	6.38		
				%SITE	12.84

Site PacketSummary

PSC		input	%device	output	%device	subtotal	%site
UIUC		1727396	44.41	2161901	55.59	3889297	21.88
JvNC		2604878	43.43	3392731	56.57	5997609	33.74
DQ0		4397293	55.72	3494374	44.28	7891667	44.39
Subtotal		8729567		9049006			
		%site	49.10	%site	50.90		
Total	17778573	%Grand	26.80				

Cornell	input	%device	output	%device	subtotal	%site
NCAR	1345601	54.59	1119433	45.41	2465034	22.57
JvNC	2235879	63.18	1303076	36.82	3538955	32.41
SURA	1158001	49.47	1182804	50.53	2340805	21.43
DQ0	666412	25.87	1909389	74.13	2575801	23.59
Subtotal	5405893		5514702			

Total	10920595	%Grand	16.46

JvNC	input	%device	output	%device	subtotal	%site
Cornell	1264426	35.79	2268175	64.21	3532601	18.99
PSC	3348239	55.97	2633637	44.03	5981876	32.15
DQ0	4413051	48.55	4676357	51.45	9089408	48.86
Subtotal	9025716		9578169			
	%site	48.52	%site	51.48		

Total	18603885	%Grand	28.04

NCAR	input	%device	output	%device	subtotal	%site
Cornell	1084014	43.94	1382763	56.06	2466777	30.02
UIUC	1161484	60.86	747026	39.14	1908510	23.23
SDSC	320522	52.56	289274	47.44	609796	7.42
DQ0	1460392	45.20	1770286	54.80	3230678	39.32
Subtotal	4026412		4189349			
	%site	49.01	%site	50.99		

Total	8215761	%Grand	12.38

SDSC	input	%device	output	%device	subtotal	%site
NCAR	251964	41.43	356193	58.57	608157	26.34
UIUC	427737	57.26	319252	42.74	746989	32.35
DQ0	405240	42.49	548437	57.51	953677	41.31
Subtotal	1084941		1223882			
	%site	46.99	%site	53.01		

Total	2308823	%Grand	3.48

UIUC	input	%device	output	%device	subtotal	%site
NCAR	710990	38.08	1156290	61.92	1867280	21.93
SDSC	313645	41.37	444478	58.63	758123	8.90
PSC	1794996	55.02	1467197	44.98	3262193	38.31
DQ0	1467197	55.81	1161551	44.19	2628748	30.87
Subtotal	4286828		4229516			
	%site	50.34	%site	49.66		

Total	8516344	%Grand	12.84

Suggestions and comments welcomed.
For more information, contact:

Craig Callinan
craig@tcgould.tn.cornell.edu

Mark Oros
oros@tcgould.tn.cornell.edu

Doug Elias
elias@tcgould.tn. ...

NISC
Cornell Theory Center
(607) 255-8686

```
Process type: 000027  options: 040000
Subnet: DMV  status: 377  hello: 16  timeout: 120
Foreign address: [128.116.64.3]  max size: 576  bias: 0
Input packets    1003162 Output packets  597828
  bad format     0       ICMP msgs       334
  bad checksum   0       Input errors    75
  returned       56      Output errors   426
  dropped        496     No buffer       0
  HELLO msgs     17005   Preempted       0
```

PREEMPT BURST ACTIVITY

Reported To ↓ Reported From ==>

SUM BURST B_DATE B_TIME #SAMPLE	CMU 128. 182. 1. 2	CRN 128. 84. 238. 200	JvNC 128. 121. 50. 20	NCAR 128. 116. 64. 3	SDSC 192. 12. 207. 1	UIUC 128. 174. 5. 14
CMU 128.182. 1.2	0 0 128		10228 2997 15-Jul-87 19:11:01 128			954 357 17-Jul-87 17:10:43 128
CRN 128.84. 238.200		0 0 128	238 119 15-Jul-87 21:10:49 128	172 102 17-Jul-87 17:10:24 128		
JvNC 128.121 50.20	5956 1336 17-Jul-87 17:10:32 126	1510 459 15-Jul-87 21:10:58 125	122 80 15-Jul-87 19:10:54 126			
NCAR 128.116 64.3		922 189 17-Jul-87 18:11:08 128		0 0 128	170 138 15-Jul-87 12:11:01 128	1421 492 13-Jul-87 22:11:01 128
SDSC 192.12 207.1				0 0 128	0 0 128	161 161 15-Jul-87 12:11:08 128
UIUC 128.174 5.14	1660 210 19-Jul-87 16:10:30 128			68 59 17-Jul-87 17:10:52 129	411 108 13-Jul-87 22:10:54 128	0 0 128

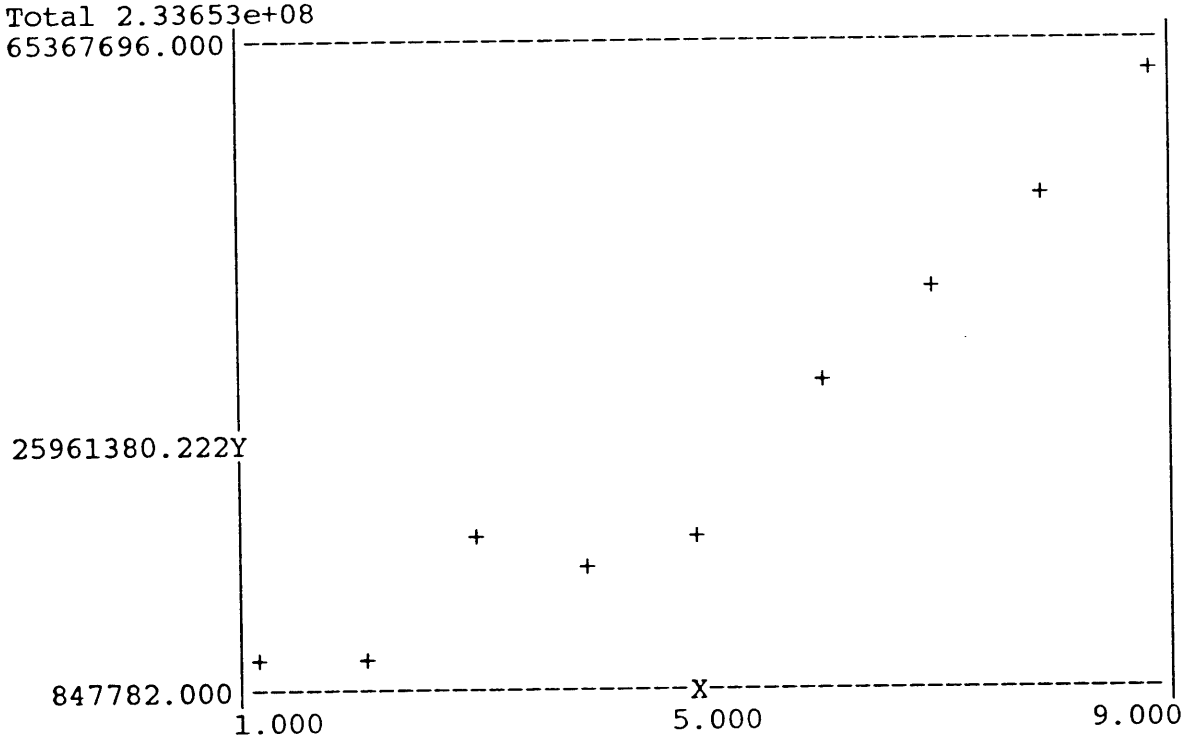
uiuc.dm2:
Summed Data and BURSTS for 13-Jul-87 @ 01:10:44 thru 19-Jul-87 @ 23:10:32

128 Samples, collected at 1hr intervals:

Variable	Sum	BURST	Date	Time
Input	1822649	73347	17-Jul-87	17:10:52
Output	1461647	55839	17-Jul-87	17:10:52
Badformat	22	2	14-Jul-87	07:11:04
Icmp	108	24	14-Jul-87	17:11:09
Badchecksum	0			
Inputerrs	220	20	15-Jul-87	19:11:10
Returned	0			
Outerrs	445	64	15-Jul-87	19:11:10
Dropped	146	26	14-Jul-87	17:11:09
Nobuffer	0			
Hello	27653	17	14-Jul-87	13:11:02
Preempt	1660	210	19-Jul-87	16:10:30

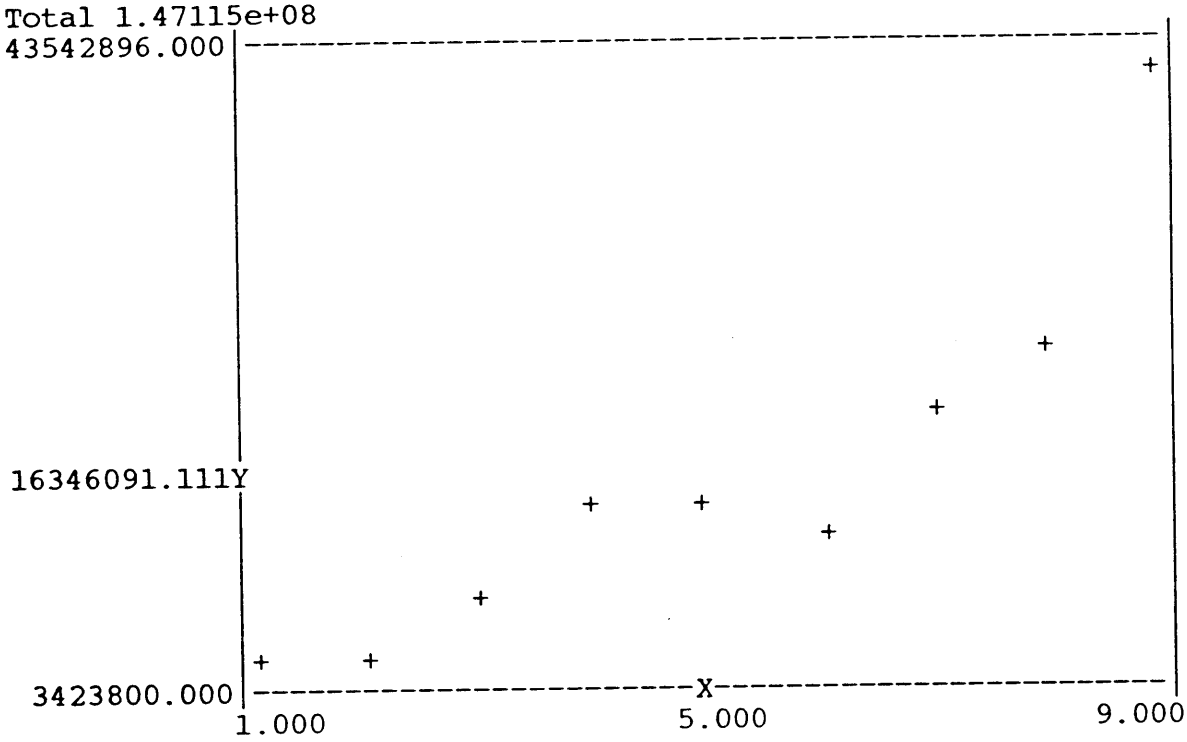
Oct 847782
Nov 1.89024e+06
Dec 1.61383e+07
Jan 1.22095e+07
Feb 1.5766e+07
Mar 3.07781e+07
Apr 4.1053e+07
May 4.96018e+07
Jun 6.53677e+07
Total 2.33653e+08

CMU MONTHLY TOTALS



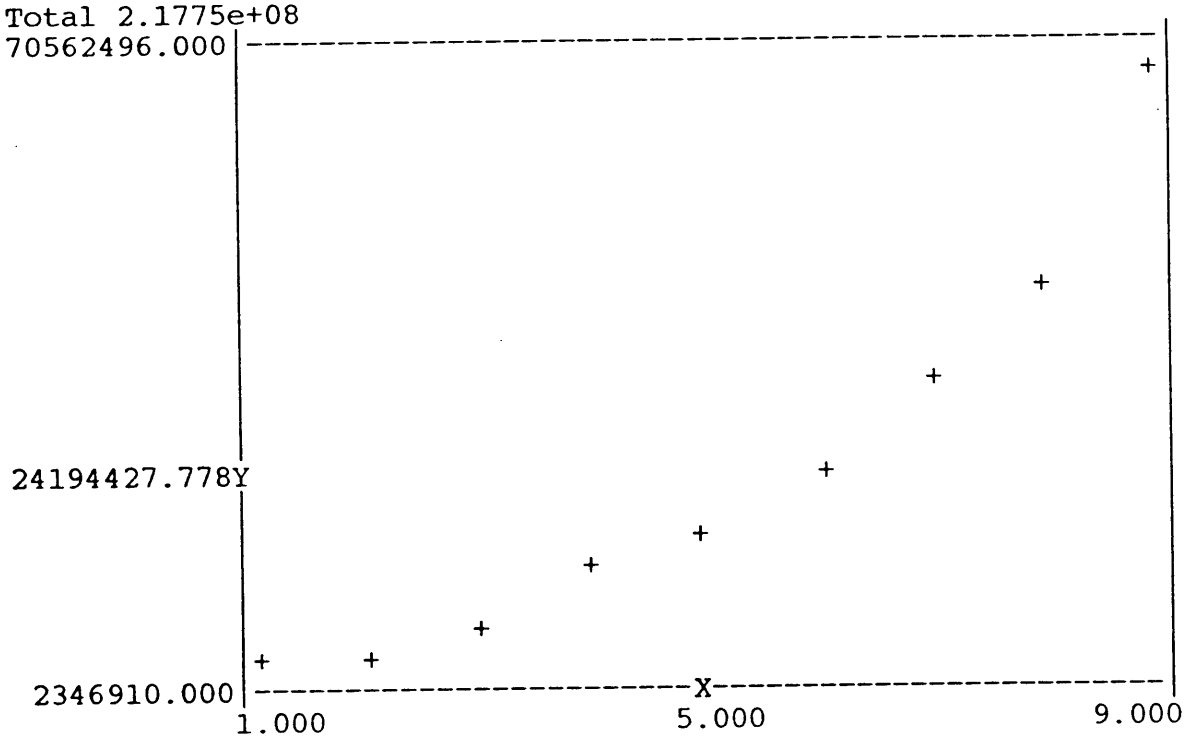
Oct 4.98829e+06
Nov 3.4238e+06
Dec 8.60273e+06
Jan 1.35748e+07
Feb 1.36196e+07
Mar 1.33001e+07
Apr 2.1215e+07
May 2.48476e+07
Jun 4.35429e+07
Total 1.47115e+08

CORNELL MONTHLY TOTALS



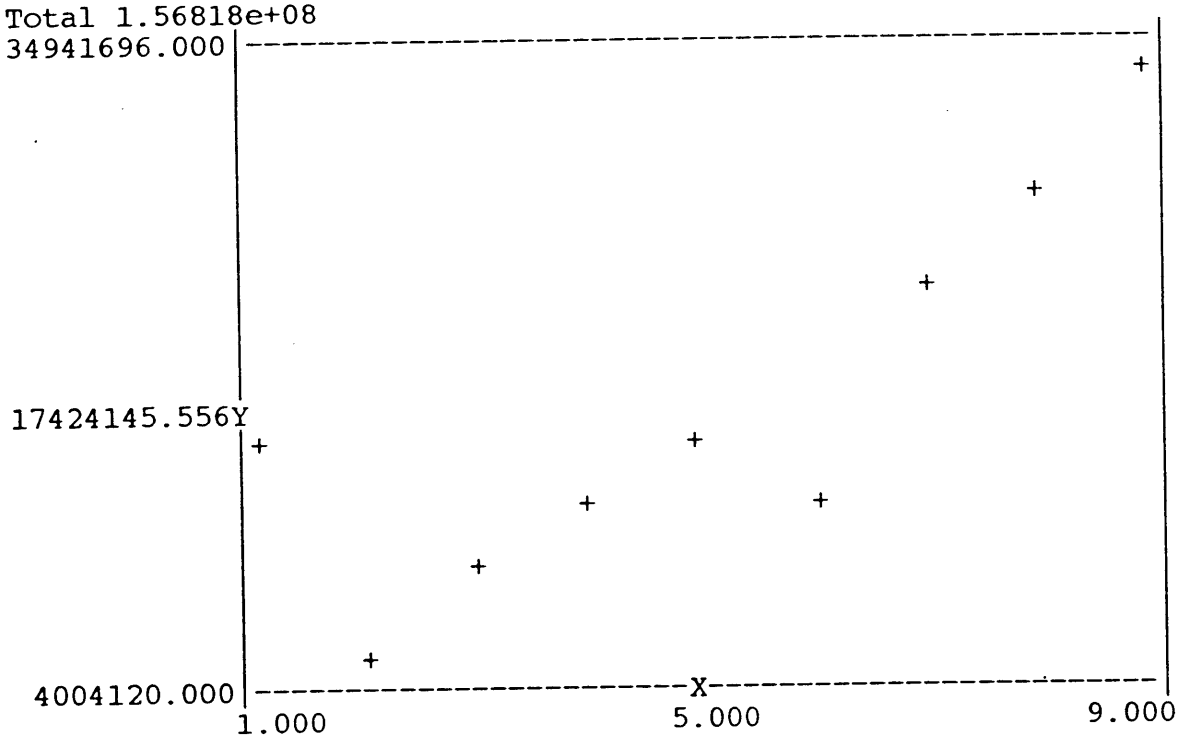
Oct 2.34691e+06
Nov 2.98007e+06
Dec 8.39647e+06
Jan 1.28987e+07
Feb 1.63958e+07
Mar 2.44316e+07
Apr 3.58616e+07
May 4.38762e+07
Jun 7.05625e+07
Total 2.1775e+08

JVNC MONTHLY TOTALS



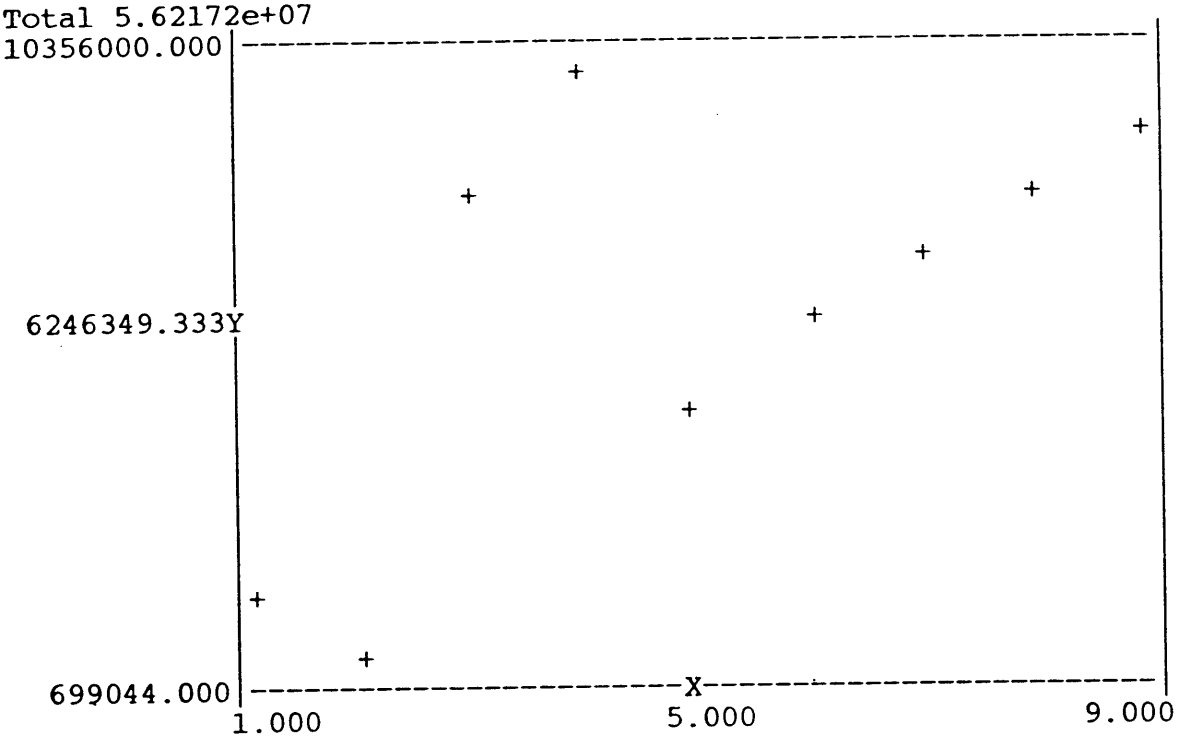
Oct 1.61628e+07
Nov 4.00412e+06
Dec 9.03769e+06
Jan 1.30512e+07
Feb 1.50921e+07
Mar 1.2542e+07
Apr 2.38634e+07
May 2.81223e+07
Jun 3.49417e+07
Total 1.56818e+08

NCAR MONTHLY TOTALS

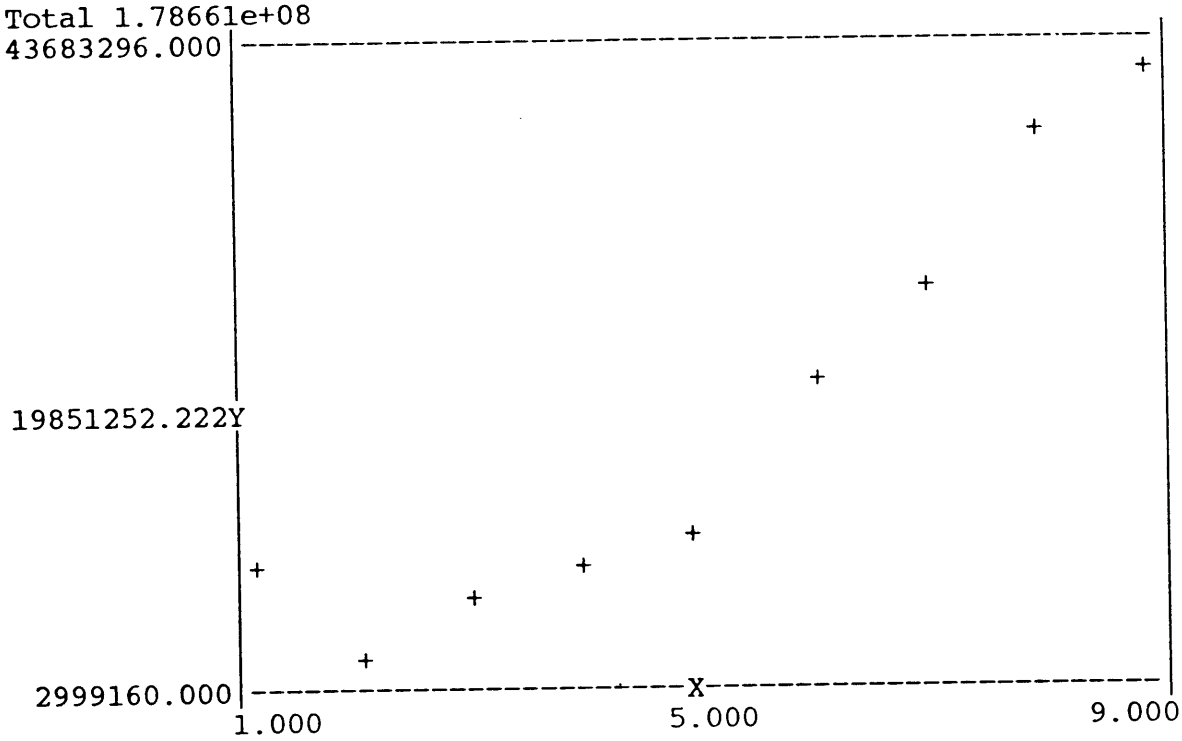


Nov 699044
Dec 8.19279e+06
Jan 1.0356e+07
Feb 4.93509e+06
Mar 6.06346e+06
Apr 7.01406e+06
May 8.08992e+06
Jun 9.02354e+06
Total 5.62172e+07

SDSC MONTHLY TOTALS

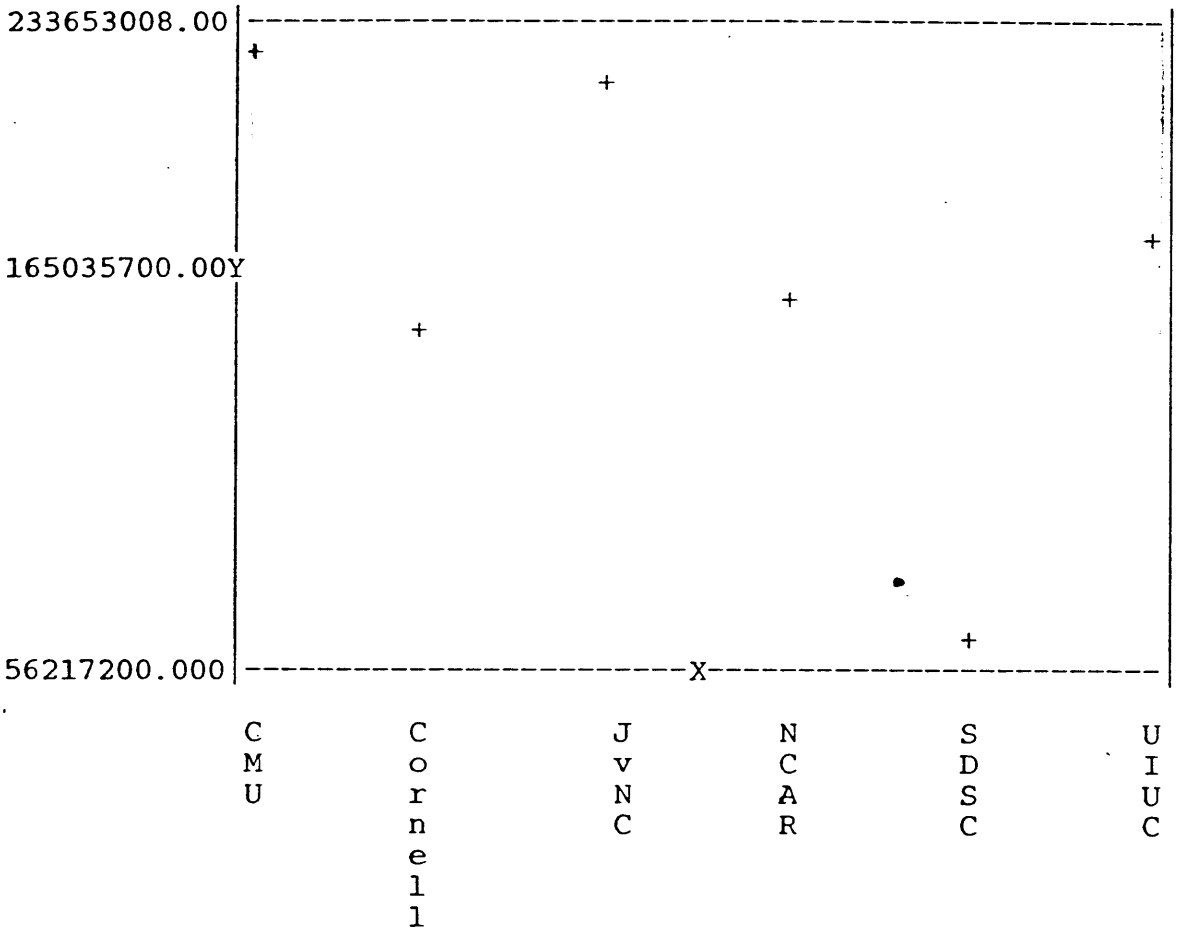


Oct 1.07025e+07
Nov 2.99916e+06
Dec 7.89881e+06
Jan 1.00113e+07
Feb 1.30762e+07
Mar 2.27405e+07
Apr 2.88073e+07
May 3.87422e+07
Jun 4.36833e+07
Total 1.78661e+08



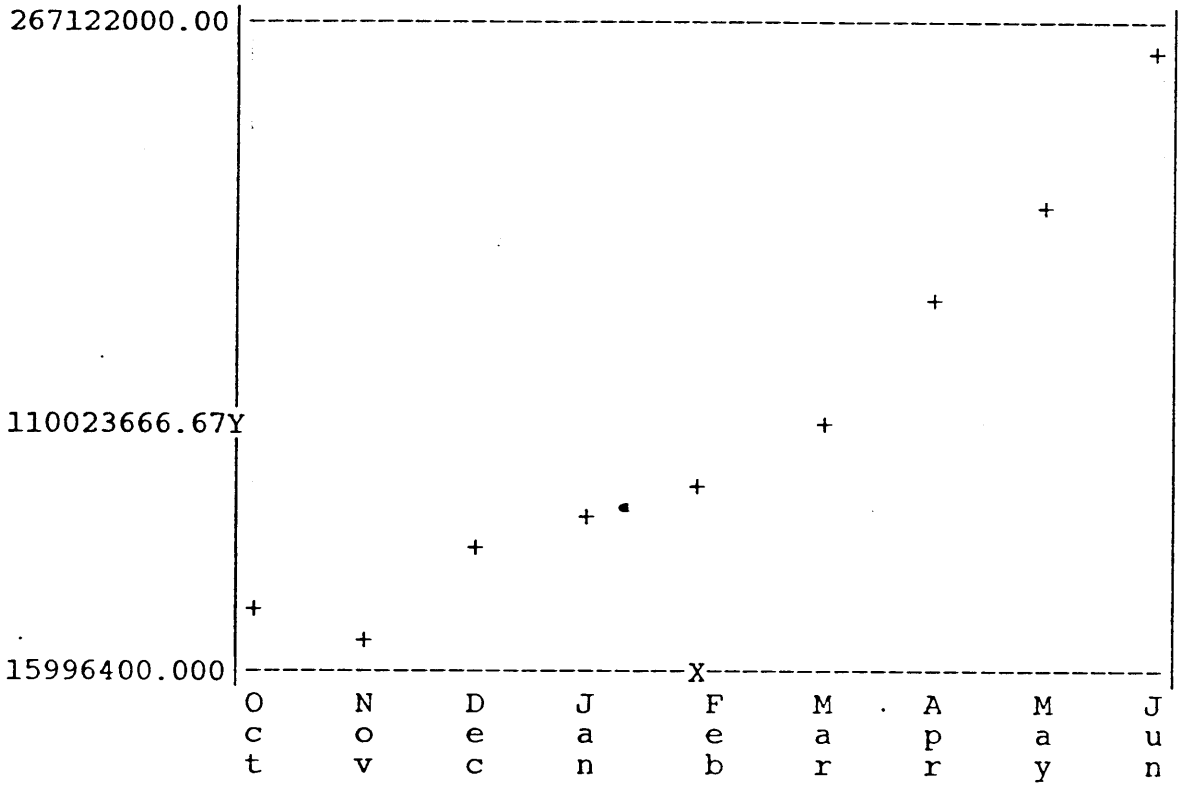
cmu 2.33653e+08
cornell 1.47115e+08
jvnc 2.1775e+08
ncar 1.56818e+08
sdsc 5.62172e+07
uiuc 1.78661e+08
Overall Total 9.90214e+08

SITE TOTALS OVER ALL MONTHS



Oct 30891522
Nov 15996434
Dec 58266790
Jan 72101500
Feb 78884790
Mar 109855760
Apr 157814360
May 193280020
Jun 267121640
Total 990214200

MONTH TOTALS OVER ALL SITES



SGMP Schoffstall (RPI)

Discussion Headings

1 Motivation

2 Time Line (Past)

3 Seeking an Engineering Balance

4 "Services"

5 Potential Uses

6 Implementations Status

7 What we've Learned

8 Possible Futures

Motivation

- Pressing network "management" needs
- Concern with other "standards", RFC's, and efforts
- Desire for implementation and interoperability experience

Time Line (Post)

- initial informal discussions before "monterrey"
- substantial discussion during "monterrey"
- authors meetings during IETF @BBN
- Set up of mailing list and distribution of first electronic documentation
- 2ND authors meeting @Proteon
- implementations begin
- 3rd authors meeting @RPI

Seeking an Engineering Balance

- simplicity
- make demands of NOC not gateway
- incorporate a few "new-to-Internet" items like:
    1) ASN.1 [only integer and octet string]
    2) architectural support for authentication [# > 1]
- unreliable transport sufficient for monitoring [use UDP]
- sensitivity to gateway implementations [Ex: size, performance ...]
- need for rapid interoperable deployment
- fit into INTERNET "Networking Style"
- need MORE than two implementation efforts (seperate)

as much as humanly possible. make RFC explicit enough to create a new interoperable implementation

have some implementation variability

Easily extend protocol support

Services

- limited number of trap/event messages
  - boot
  - link failure
  - authentication failure
  - EGP neighbor loss
- GET variables
  - routing table (and how learned)
  - version
  - interface information
    - type, "speed", packets, bytes
  - some EGP specifics
  - routing interchange info
  - vendor specific TOO!
  - counters
- Authentication
  - NULL

Potential Uses

- statistics gathering
- source of routing information
- topological mapping of networks
- network state monitoring
- monitor hosts
- could be used as input or "concept prototype" for more ambitious efforts

Implementations Status

- 2 gateway/IS implementations
- 2 NOC/ES implementations
- Proteon p4200 implementation working
- working monitoring tools for ULTRIX and MSDOS @ UTK with interface to presentation graphics
- RPI UNIX implementation is 2 working days behind
- GATED integration @ Cornell by Mark Fedor
- RPI again

C code available in PD and can be commercially exploited at no cost

What We've Learned

- subsets of ASN.1 are NOT impossib
  to parse, implement, interoperate
- ASN.1 constructs that pertain to
  multiple protocol layers are
  difficult
- easily extensible protocols are
  easier to specify and standardize
- Padlipsky was right.

~~Lessons~~ What we've learned

~~Implementation~~ ~~Implementation Education~~

subsets of

- ASN.1 not impossible to implement ~~and~~ interoperate and parse
- ASN.1 constructs that pertain to multiple-protocol layers are difficult
- easily extensible protocols are easier to specify and standardize
- ~~if you TAN the swamp with the alligators you move faster than if you ~~walk~~ on the shore~~
- Podlipsky was right. 

Possible Futures

- RFC xxxx (real soon now) not necessarily the end
- more variables
  management (SETS)
  real authentication
  more discussion at next IETF

possible

Futures?

- ~~exploit~~ RFC XXXX not necessarily the end
  - more variables
  - management (ie SETS)
  - real authentication
- more discussion (esp deployment/questions?) @ next IETF?

→ NOT Protean protocol

→ When there is a real standard deployed and interoperable consider conversion

A Plea from Vendors Crocker (TWG)

Some Pleas

from a

Vendor

Dave Crocker
The Wollongong Group
dcrocker@twg.arpa

1. Protocol Feature Checklist

Help RFP writers

Reality vs. THE SPEC

For each protocol

List all features, not just options

Telnet + Official Protocols is a start

A Simple Table

2. Implementation Details

Some text for the feature table

Common choices

SRTT	(limitation)
Silly Windows	(limitation)
RFC 822	host = IP address?

3. Local Network Login Security

Full encryption ⇒ hardware

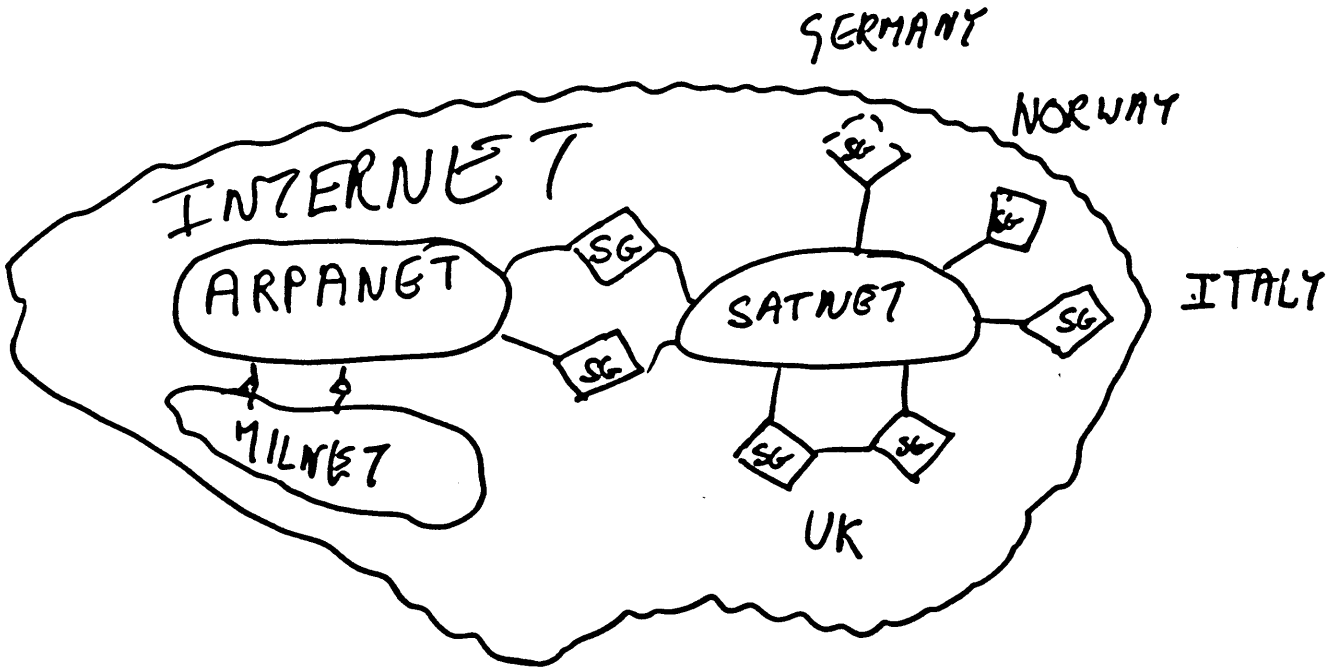
(LAN ⇒ PC monitor traffic)

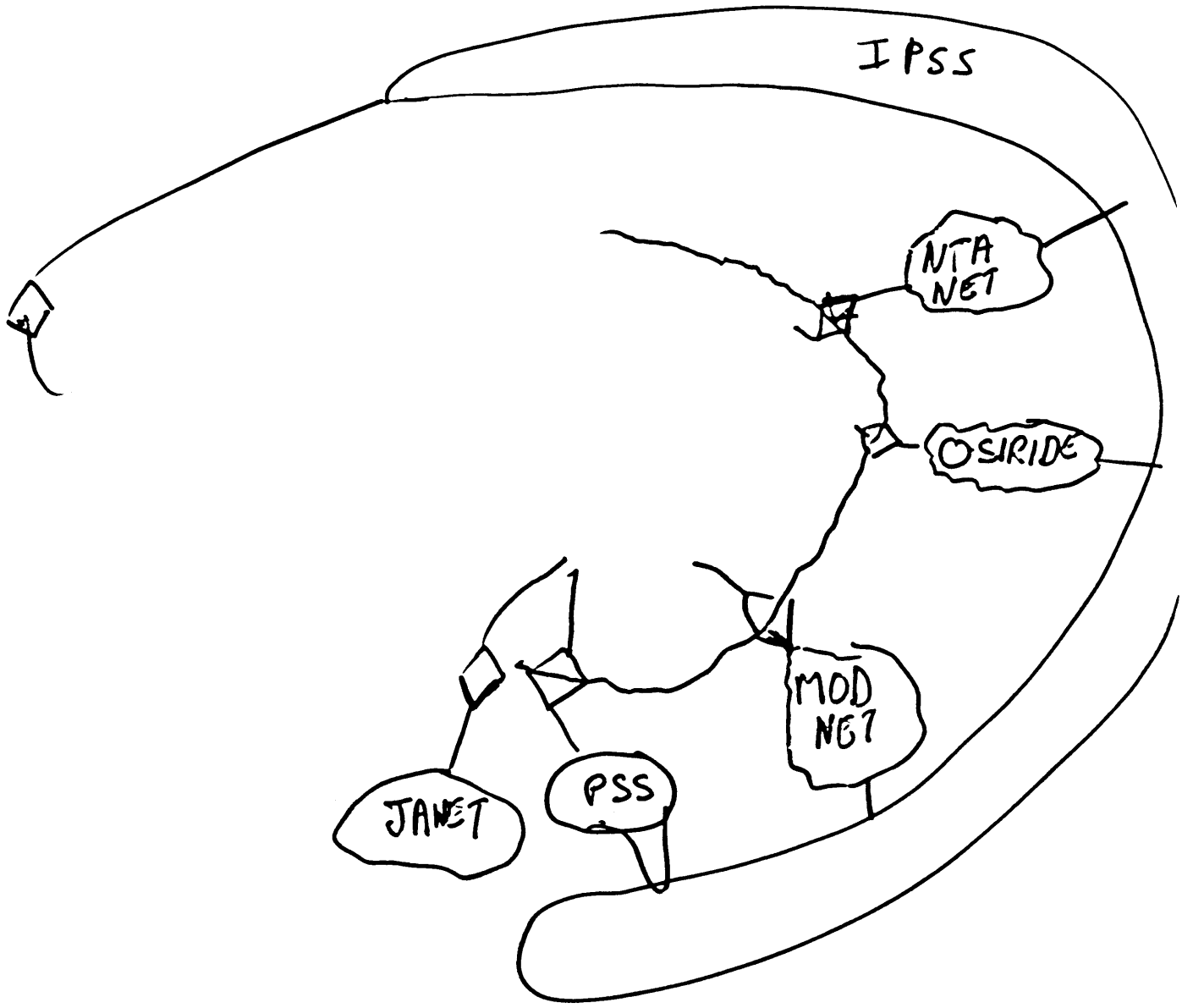
At least protect passwords!

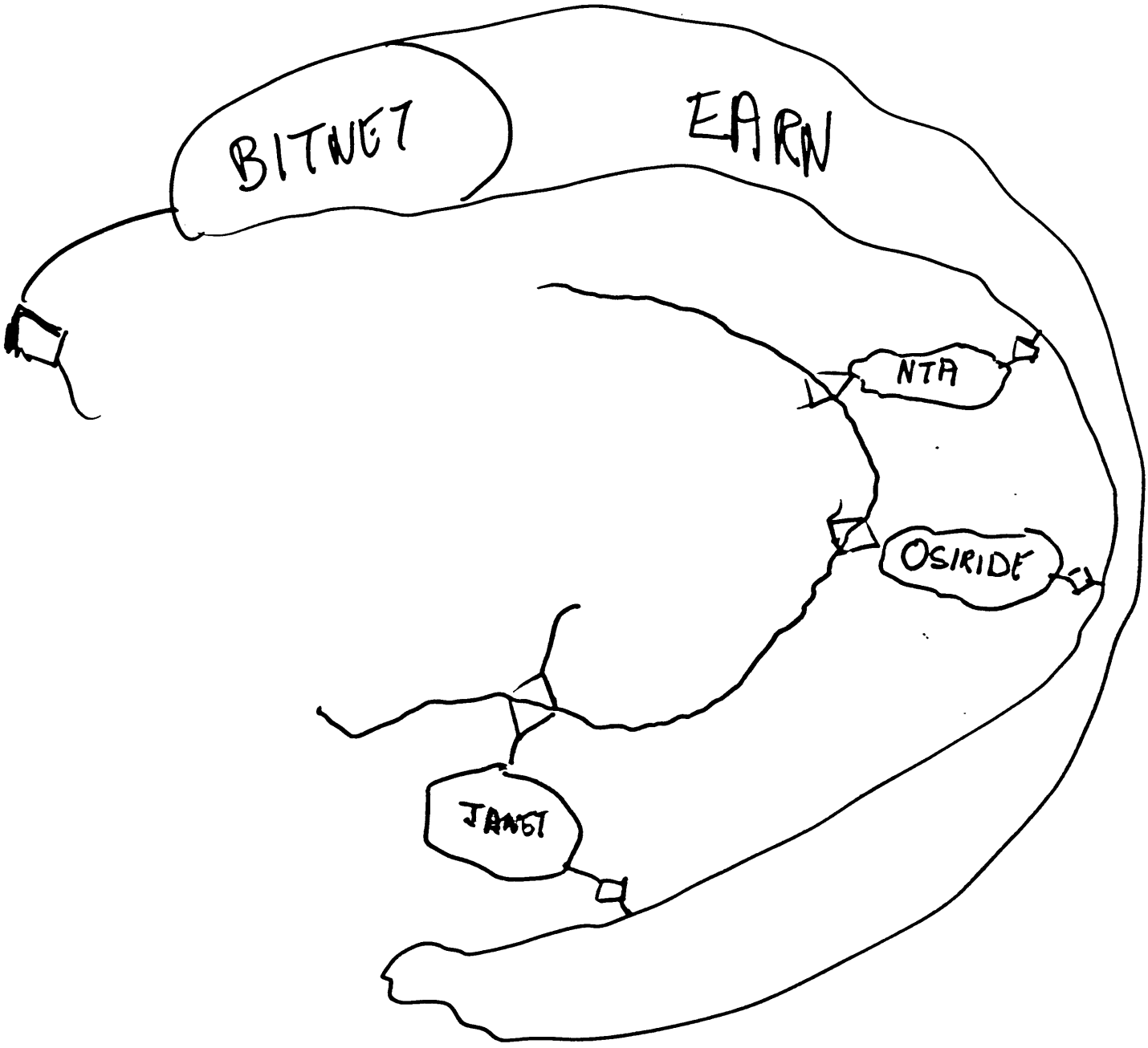
Telnet Option: Username/Password

Challenge/Response?

The International Internet Kirstein (UCL)


GERMANY
NORWAY
INTERNET
ARPANET
SG
SG
SATNET
SG
SG
SG
SG
SG
ITALY
MILNET
UK


IPSS
NTA NET
OSIRIDE
MOD NET
PSS
JANET


BITNET
EARN
NTA
OSIRIDE
JANET

IN ALL OTHERS

SINGLE RESPONSIBLE PERSON

SINGLE TROUBLE SHOOTER

(NOT REALLY IN US)

NEED TO DEAL WITH

SATNET PROBLEMS

GATEWAY PROBLEMS

HOST-HOST ROUTING/CONFORMAN

MESSAGE LOSS

PERFORMANCE

ADDRESSING

ACCESS CONTROL/ACCOUNTING

BETWEEN AUTONOMOUS SYSTEMS

EXAMPLE JANET

COLOURED BOOKS PROT

~ 1500 HOSTS

(X25 NET)

SOME LANS WITH TCP/IP

NAME REGISTRATION SCHEMES

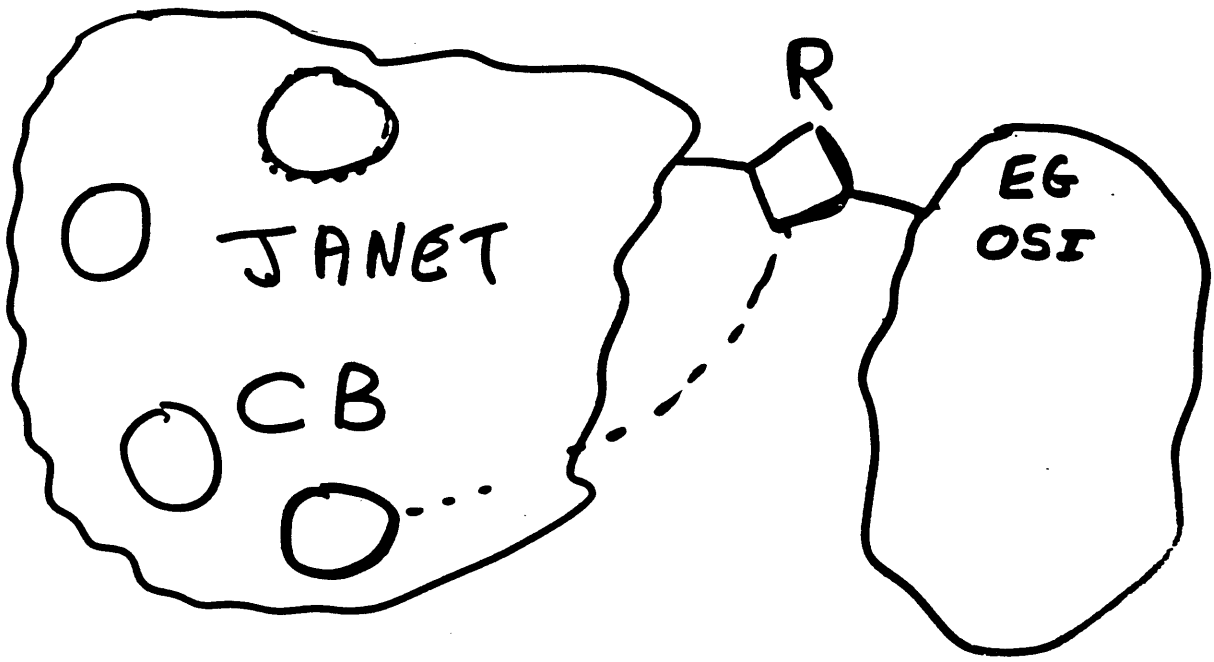
(IS DOMAIN ON INTERNET)

GATEWAYS TO IPSS /CB

GATEWAYS TO EARN

(RARE BEING DEVELOPED)

(OSI TRANSITION PLAN)



CB REAL PROTOC SET C

OSI COMING O

LOWER LEVELS ALREADY THERE

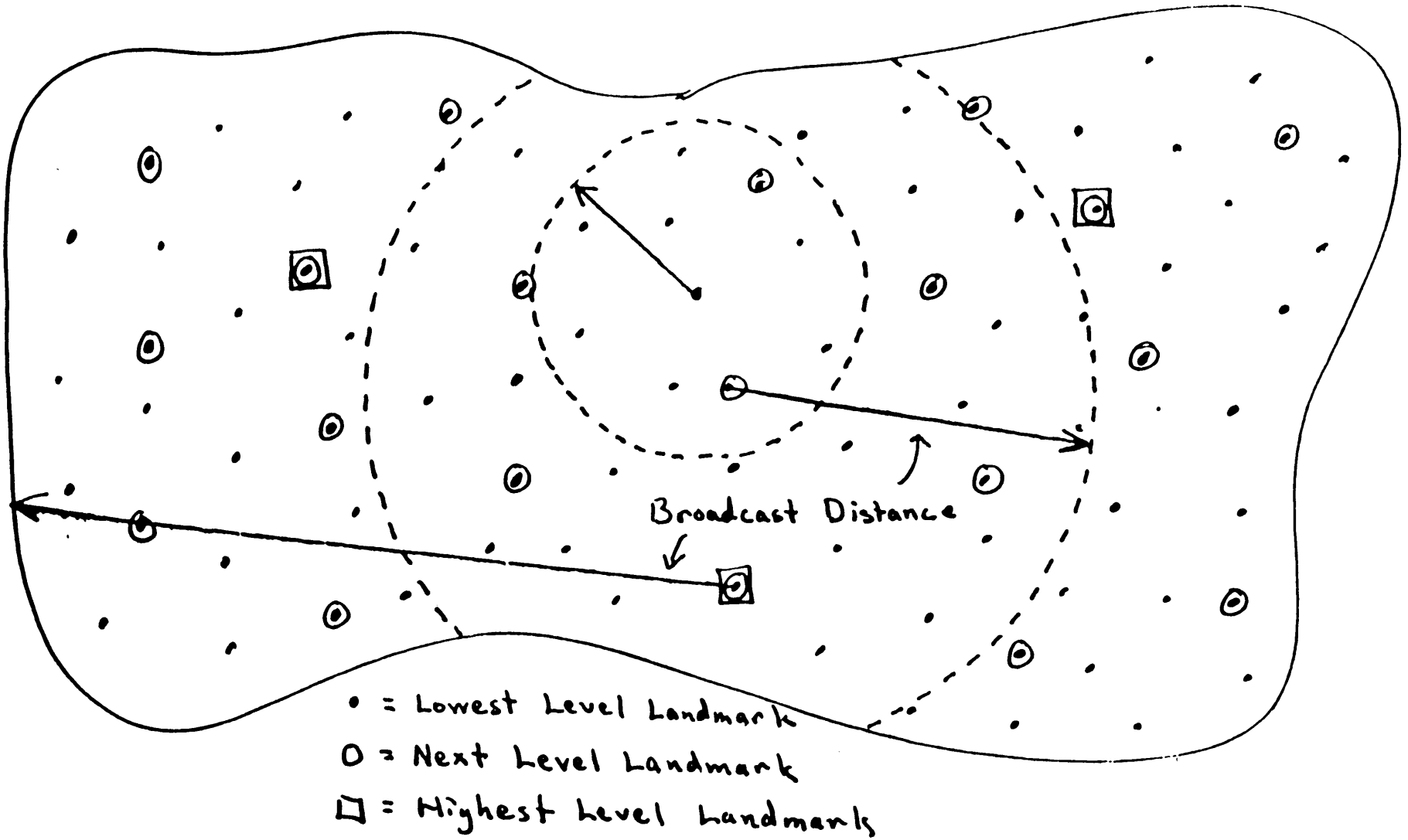
LANDMARK ROUTING

MITRE Corporation, W-31
Paul F. Tsuchiya

Contents

Contents

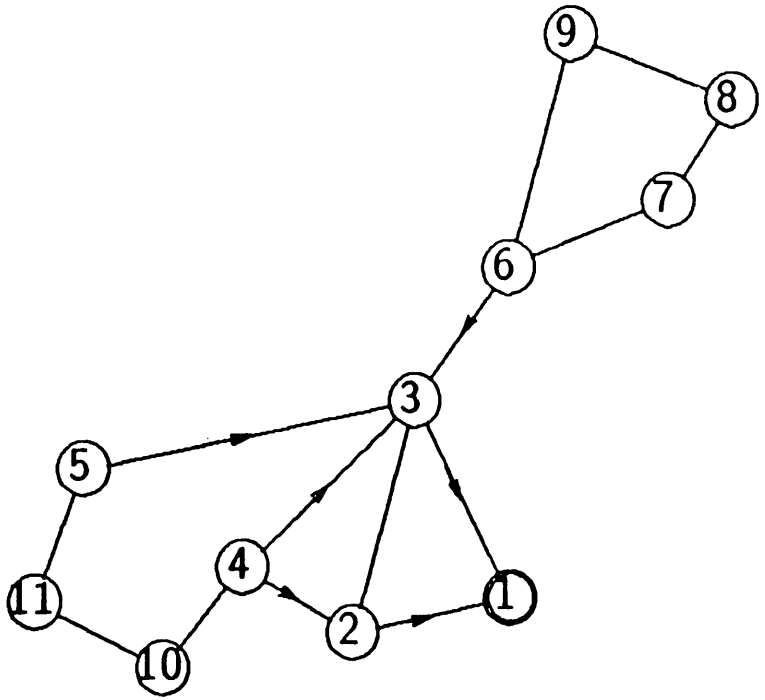
Landmark Routing Architecture



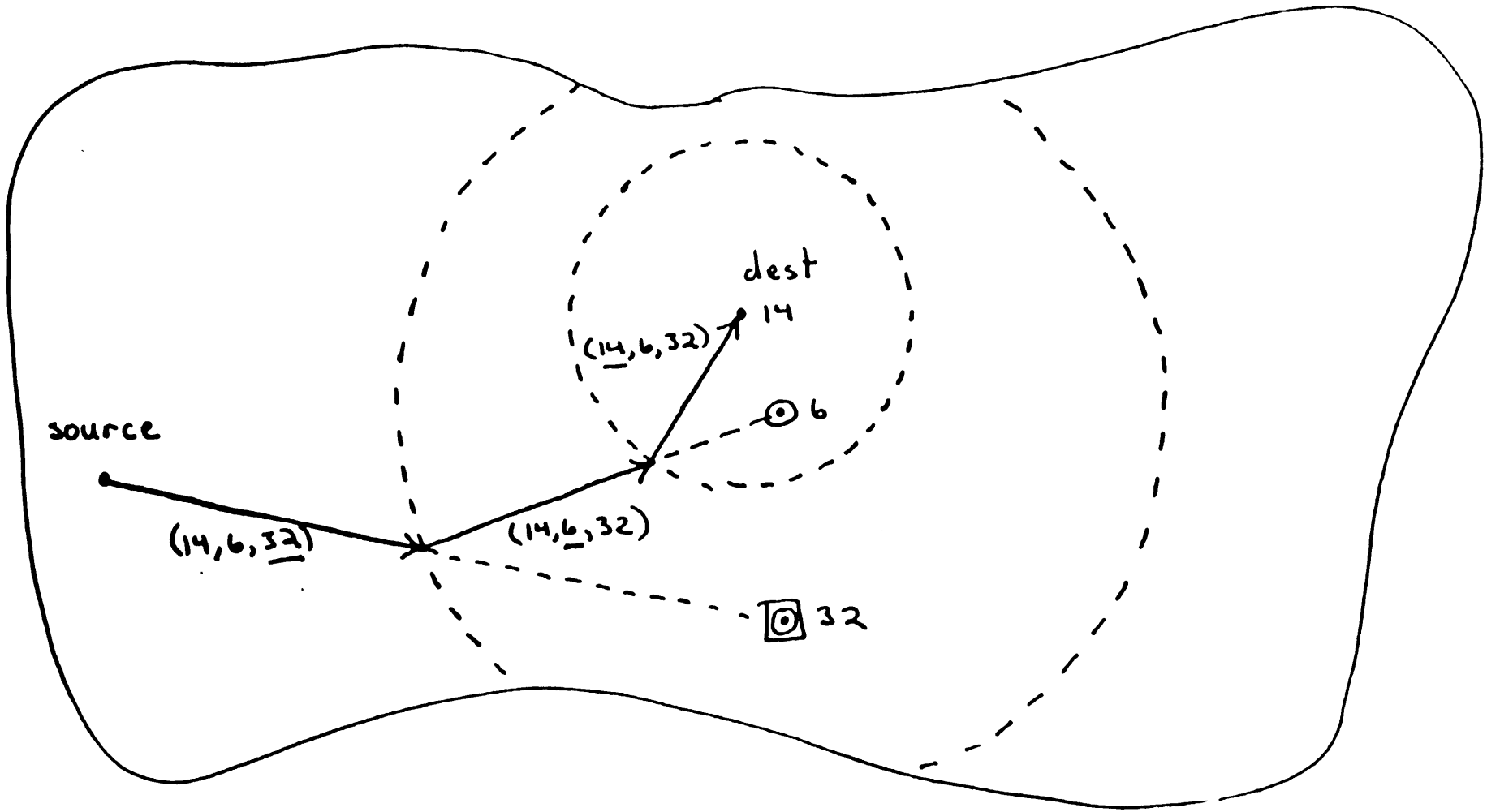
MITRE Corp.

Principal Investigator: Paul F. Tsuchiya

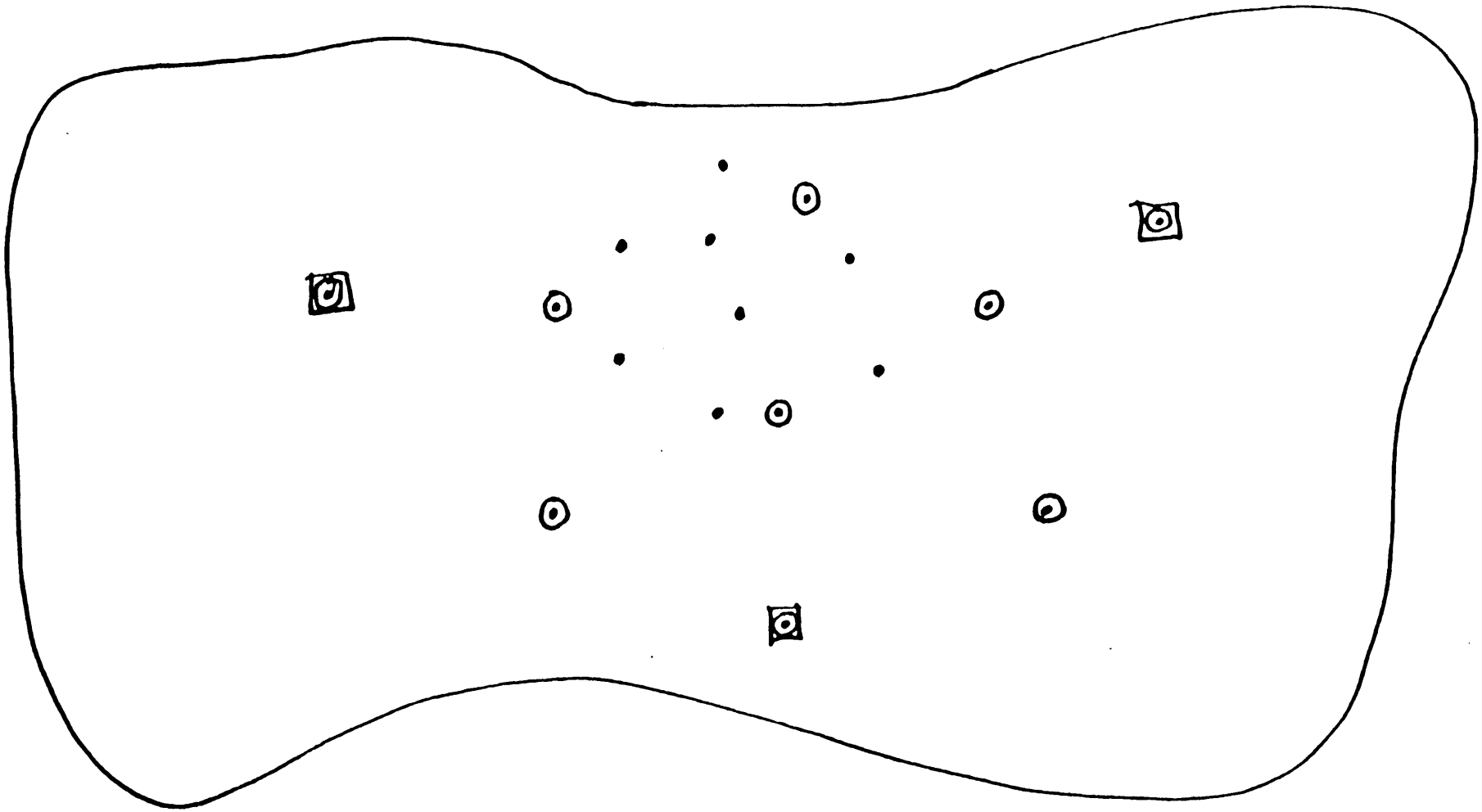
What is a Landmark?



How Routing Is Done



What Each Node Sees



Hierarchy Parameters and Characteristics

Parameters

- r is distance of Landmark broadcast
- d is distance between two nodes

Characteristics

- The distance d from a Landmark to the next higher Landmark must be less than r for the lower level Landmark ($d <= r$)
- Routing table size dependant on ratio r/d, not on specific value of r or d
- Path lengths are inversely proportional to routing table sizes

- Relationship between r and d important, not actual values of r and d

Results - Analysis Techniques

- Analyzed routing table sizes, specification of hierarchy
  - Still no analysis of path lengths
  - Specification analysis useful for later algorithm development
- Over 1000 simulations presented in MTR
  - Large body of simulations to study effects of various characteristics in statistically significant fashion
  - Additional individual simulations to further study certain characteristics
- Large Network Estimations
  - Predict performance for networks of any size

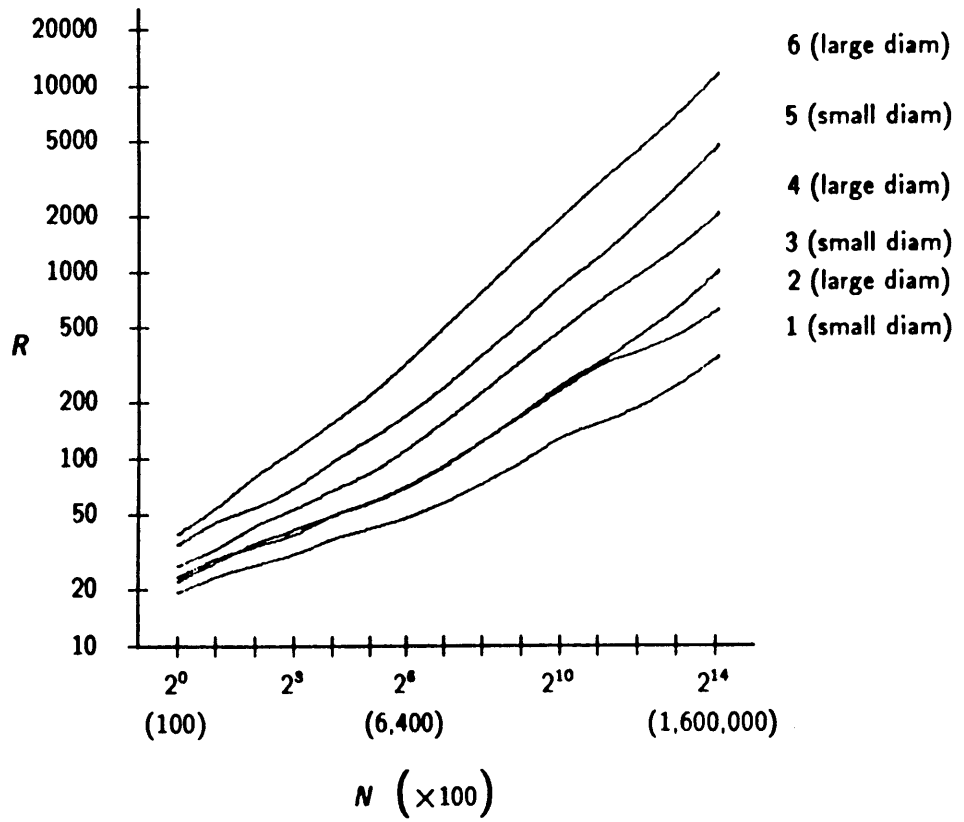
Hierarchy Characteristics

- *r* is easy to adjust (one field in broadcast message)
- *d* is hard to adjust (depends on placement on Landmarks)
  - — But doesn't matter very much because RELATIONSHIP between *r* and *d* is what is important, and that relationship can be easily adjusted by changing only *r*
- Since placement of Landmarks not particularly important, technique for creating and maintaining hierarchy can be very simple
- Simple adjustment of *r*, not choice of Landmarks, tunes hierarchy (table sizes and path lengths)

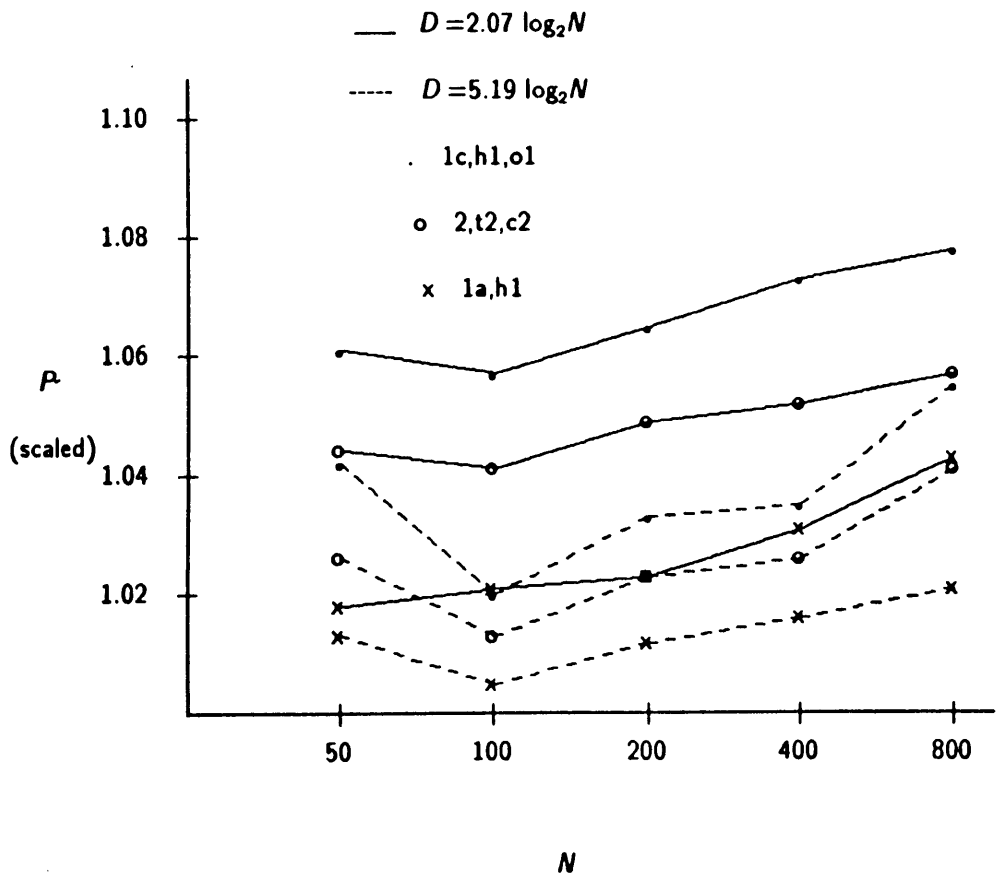
Hierarchy Characteristics

- *r* is easy to adjust (one field in broadcast message)
- *d* is hard to adjust (depends on placement on Landmarks)
  - — But doesn't matter very much because RELATIONSHIP between *r* and *d* is what is important, and that relationship can be easily adjusted by changing only *r*
- Since placement of Landmarks not particularly important, technique for creating and maintaining hierarchy can be very simple
- Simple adjustment of *r*, not choice of Landmarks, tunes hierarchy (table sizes and path lengths)

Estimation Results



Simulation Results - Path Lengths



Other Results

- Typical Routing Table Sizes approximately three times the square root of the number of nodes
- Path Lengths appear to behave similarly to area hierarchy (dependant on traffic matrix)
- Routing Table Sizes and Path Lengths worse for very small diameter networks (much smaller than the ARPANET)
- Random assignment of Landmarks nearly as good as uniform assignment
- 200 node simulations show Landmark Hierarchy Path Lengths 1/2 those of the Area Hierarchy, and Routing Table Sizes 2/3 that of the Area Hierarchy

Remaining Issues

- Hierarchy Creation and Maintainance
- Name-to-Address Binding
- Routing Technique
- Address Structure
- Administrative Boundaries

Other Results

- Typical Routing Table Sizes approximately three times the square root of the number of nodes
- Path Lengths appear to behave similarly to area hierarchy (dependant on traffic matrix)
- Routing Table Sizes and Path Lengths worse for very small diameter networks (much smaller than the ARPANET)
- Random assignment of Landmarks nearly as good as uniform assignment
- 200 node simulations show Landmark Hierarchy Path Lengths 1/2 those of the Area Hierarchy, and Routing Table Sizes 2/3 that of the Area Hierarchy

Hierarchy Maintenance: Landmark Hierarchy

- Easier because hierarchy performance not very dependent on choice of Landmarks
  - — Landmark radius determines performance
    - Radius easily adjustable
  - — Nodes can to a large extent make individual decisions
    - If too many Landmarks, decrease radius, if too few, increase radius

Hierarchy Maintenance: Area Hierarchy

- Difficult because groups of nodes (areas) must behave synchronously
  - — Groups of nodes must often agree on hierarchy state in order to make decision
    - For instance, to recognize partitioned area and elect new area leader
- Performance of area hierarchy (table sizes, path lengths) rather sensitive to choice of areas
  - — Decision for node to join one area or another rather complex
    - Can result in conflicts where two areas want the same node, or where no areas will accept a node

Hierarchy Creation and Maintenance

PROBLEM DEFINITION:

- When does a node become a Landmark?
- How far does a node broadcast (Landmark radius)?
- When does a node cease being a Landmark?

GOALS:

- Node always makes independent decisions
- Nodes are loosely coupled
  - Chain reaction of events impossible
- Minimize changes in Landmark assignments
  - Minimize number of new address bindings

Two Hierarchy Maintenance Modes

- Partitioned and Non-partitioned
  - — A partition exists when a parent cannot hear its child
    - Parent becomes black-hole for child and all of its children

Maintenance Philosophy

- — Adjust hierarchy while non-partitioned to avoid partitions
  - Non-partitioned adjustment can be controlled for minimum network perturbation
  - For instance, Landmarks can have both old and new addresses during a hierarchy admustment, to smooth out address bindings

Non-Partitioned Hierarchy Maintenance: General

- Maintain fairly large number of global Landmarks (those which broadcast network wide)
    - — Decrease number of nodes dependent on any given Landmark
    - — Experimentation shows number of global Landmarks should be roughly 30% to 50% total number of nodes in routing table

- Strive for fairly even distribution of Landmarks
    - — In terms of distance from other peer Landmarks
    - — In terms of number of children per Landmark
        - Makes address space more compact

Partitioned Hierarchy Maintenance

- Three types of partitions:
  - — Child sees parent, but parent doesn't see child (most common type)
    - Child simply increases Landmark radius to encompass parent
  - — Child no longer sees parent, but sees another potential parent
    - Child adopts new parent, gets new address
  - — Child sees no potential parents (rare type)
    - Several possible actions
    - Run election with highest level peers always a possibility

Techniques continued

3. Promote Landmark to next higher level
    - When too many siblings
    - Choose Landmark furthest from any higher level Landmarks for best spacing
    - Some siblings now become new Landmark's children
4. Demote Landmark to next lower level
    - When Landmark doesn't have enough children

Techniques for Non-Partitioned Hierarchy Maintenance

- All changes done so that old address and new address are both good for a period of time
  - — Allows smooth, gradual distribution of new address bindings

1. Neighbor of Landmark takes over Landmark job (transparently -- that is, with same Landmark Address)
   - — When neighbor has better spacing with other Landmarks
   - — When Landmark crashes unexpectantly
2. Child gets new parent
   - — When new parent closer than old parent
   - — Build hysteresis into this process to avoid flapping

Existing Ways

- Flood all network elements with all name-to-address bindings
  - — Telephone book an example
  - — Obviously inefficient for large computer networks

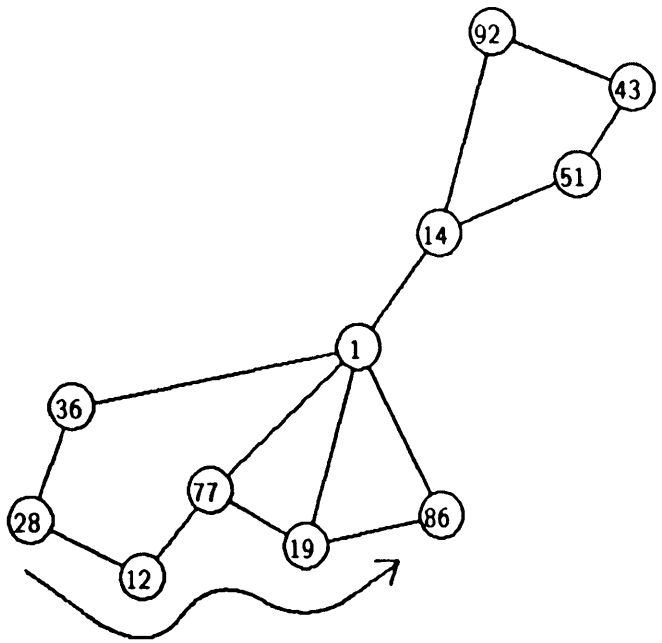
- Put some additional semantics in the name to indicate which server(s) hold the binding
  - — DARPA name structure an obvious example
  - — Results in less permanent names (name changes when server changes)
  - — Limits number of servers, less survivable
  - — Requires some apriori information (address of servers)

CATCH 22: Since addresses can change, server addresses themselves must be broadcast network wide (i.e., must bind to server before can bind to destination!)

Binding Fundamentals

- Name (unique ID) to Address (network location) mappings can change
  - — Name remains constant, but Address changes
- Source has name, but needs destination address to communicate
- Destination must supply source with its own address via some third party name-to-address server
  - — Destination and source must have some way of knowing where the name-to-address server is

Assured-Destination Routing Function Example



route from node 28 to node "79"

Normally routing function trashes messages for which it cannot locate an address

In assured-destination, routing function always delivers message, whether or not stated destination exists

In this example, routing function delivers message to "next higher address" if actual address doesn't exist

Assured Destination Binding Approach: Basic Scheme

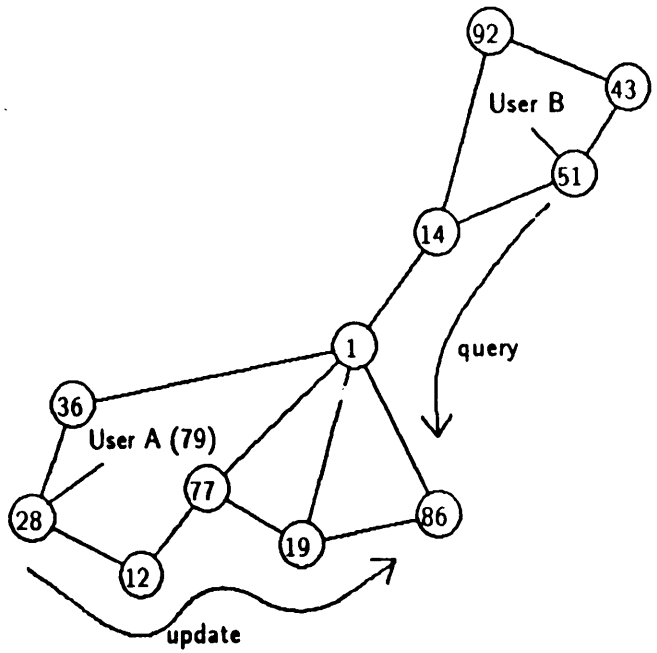
- Algorithmically derive server address directly from name using hash function
  - Translates name space into address space
  - Uniformly distributes resulting addresses
  - RESULTING ADDRESS MAY NOT CORRESPOND WITH ANY REAL ADDRESS
- Modify routing procedures so that a destination is routed to whether or not address is real
  - For example, simply route message to next numerically higher address in routing table

ADB: Variations

- Since addresses in routing table will be clustered, they must be also be hashed

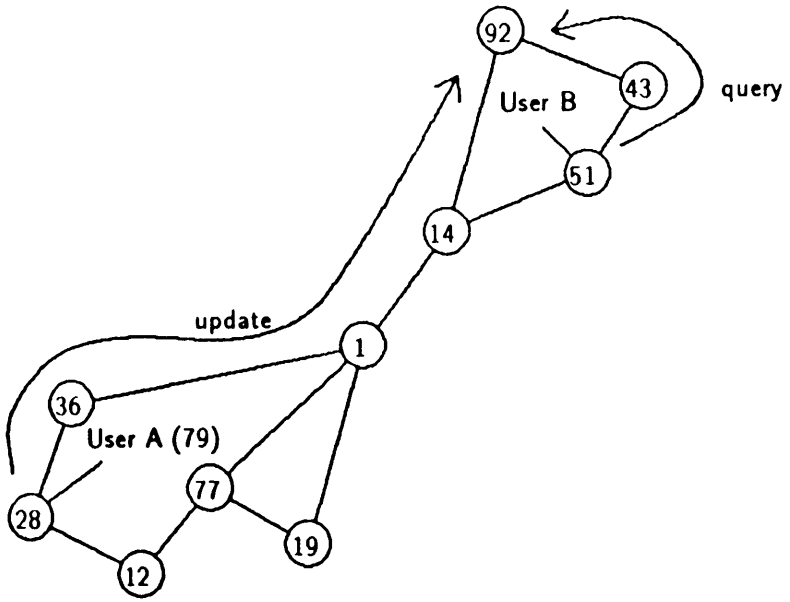
- Optimize by hashing name several times, producing multiple servers
  - — Queries then go to nearest server
  - — Increased survivability
  - — By hashing popular destinations (like the NIC) more times than unpopular destinations, binding workload is evenly distributed
  - — By hashing addresses whose machines have greater binding capacity more times than those with less capacity, binding workload is appropriately distributed
  - — By hashing name to addresses within trust zone, binding can be guaranteed to occur on trusted machines

Assured-Destination Binding Example



Node "79" is default server address for User A

Node 86 is assured-destination for Node "79", and becomes server



Node 86 crashes

Now Node 92 is assured-destination for Node "79", and becomes server

Landmark Routing Updates

- Landmark Routing must be Distance Vector (aka Bellman-Ford, Old ARPANET)
- Distance Vector routing may be timer driven or event driven
  - Old ARPANET and EGP are timer driven
    - Slow response (especially count-to-infinity bad news), very simple
  - Burroughs Integrated Adaptive Routing System (BIAS) is hybrid
    - Timer driven for good news, event driven (Jaffe-Moss algorithm, 1982) for bad news
  - BBN considered event driven Distance Vector routing, but abandoned in favor of SPF (Link State) before really studying it (BBN Report No. 3803, 1978)
    - Contains promising idea along the lines of Jaffe-Moss for solving looping problem (count-to-infinity)
- IF ENGINEERED WELL, DISTANCE VECTOR SHOULD PERFORM AS WELL AS LINK STATE

Quadruple-threat Binding Method

- When address changes, send binding to:
  - — Nodes currently communicating with
    - Existing communications not interrupted
  - — Previous address
    - Pick up in-transit messages, nodes whose addresses changed simultaneously
  - — Community-of-Interest nodes
  - — Regular hashed assured-destination servers
- Provides robustness and/or optimality and/or fast response, etc.

Addressing Considerations

- Addresses are a concatination of Landmark IDs, one for each hierarchy level
  - — Addresses must be globally unique
  - — Addresses must be short

Simplest Address Assignment (unacceptable):

  - — Make each Landmark ID same as node's name (gloabally unique)
    - Results in very long addresses
    - Sensitive to node crashes

- Solution: make Landmark IDs only locally unique

Event Driven Distance Vector

- Landmark Routing primarily event driven

- Two types of updates:
  - — Hierarchy maintenance update uses hop-count as metric
  - — Routing information update uses delay/bandwidth/whatever metric

- Updates can travel via controlled broadcast (similar to SPF broadcasts), or via more traditional "trade routing tables" method
  - — Update travel until Landmark radius hop-count reached

Typical Address Sizes

- Typically there are 2 or 3 times as many Level *i* Landmarks as there are Level *i+1* Landmarks
  - — Therefore, each Landmark will have, on the average, 2 or 3 children
  - — Even with deviation of 3 or 4 times, 4 bits (16 values) is adequate to uniquely encode all Level *i* Landmarks

Example:

- — Assume 10000 nodes, 100 highest level Landmarks, 4 children per Landmark.
- — Then there will be 5 hierarchy levels, requiring 3 bytes of address space (1 byte for highest level Landmarks, 4 half-bytes for the remaining Landmarks)

Locally Unique Landmark IDs

- Each Level *i+1* Landmark will have *x* Level *i* Landmarks closer to it than any other Level *i+1* Landmark
  - These *x* Level *i* Landmarks will choose Level *i+1* Landmark as part of their addresses (i.e., Level *i+1* Landmark becomes parent to Level *i Landmarks).*
  - *Terminology: If level i* Landmark A uses Level *i+1* Landmark B as part of its address, then B is the *parent* of A, and A is *child* of B. Two Landmarks which have same parent are *siblings.*
- If all of a Landmark's children have unique IDs, then all addresses will be unique.
  - (All highest level Landmark IDs must of course be unique)
- Since all siblings may not directly hear each other, parent must list its children's IDs in its broadcasts
  - Simple election algorithm adequate for picking Landmark IDs

Architectural Considerations

- Advantage of area hierarchy is that the areas may neatly correspond to administrative boundaries
  - By nature, Landmark Hierarchy does not recognize administrative boundaries

- Reasons for administrative boundaries:
  - Routing Autonomy within administration
  - Way of keeping internal traffic inside
  - Prevent routing 3rd party traffic (traffic whose source and destination are in other administrative areas)
  - Protection against bad external routing information

- To fix problem, we introduce Trust Zones to the Landmark Hierarchy

Trust Zones

- A Trust Zone is a group of connected routers similar to an area in the area hierarchy
  - — Members of a Trust Zone share some addressing component
    - May be either a particular Landmark, or some non-Landmark related addressing information
  - — Members of a Trust Zone establish enough Landmarks that all routing within Trust Zone relies only on Landmarks inside Trust Zone
    - Typically involves only a slight extension of some Landmark broadcasts to cover entire Trust Zone, or the addition of one higher level Landmark
  - — Trust Zones may be nested
- If name-to-address binding to occur within Trust Zone, at least one name hash must be limited those Landmarks within the Trust Zone
  - — Other name hashes must be general, for those systems outside of Trust Zone

Autonomy

- If current Autonomous System model used:
  - — Landmark Routing will be an igp
  - — Islands of Landmark Routing systems will act autonomously from each other and within the framework of the Autonomous System address structure (whatever that turns out to be)
  - — Little will have been done to improve overall DoD Internet routing
- If current Autonomous System model replaced by pervasive Landmark Hierarchy:
  - — Autonomous Systems which are NOT transitive (do not route 3rd party traffic) do not need to participate in Landmark Routing, but:
    - They will fall into Landmark address structure, and border nodes will need to participate in name-to-address binding
    - They will not be able to repair internal partitions
  - — Autonomous Systems which want to be transitive will almost certainly need to do Landmark Routing (little autonomy)

Handling Routing Updates Within Trust Zones

- Updates generated inside Trust Zone may propagate outwards, but may not come back in
    - — Border nodes must screen address of incoming updates
- If non-transitive:
    - — Externally generated updates may enter Trust Zone, but may not leave
        - Either border nodes must check address of outgoing updates, or incoming updates must be labeled and outgoing updates checked for label
- If transitive:
    - — External updates pass through unchanged
- Partial transitivity is real tricky--wouldn't recommend it as part of Landmark Routing

Problems with Current Routing Architecture

- Current Internet/Autonomous System architecture inherently non-survivable and clumsy
  - — No routing information between subnets and gateways
  - — Little routing information between different Autonomous Systems
  - — Configuration across large subnets difficult and threadbare
    - Entire exterior gateway configuration depends on a few core gateways
  - — Partitions difficult to repair without better coordination

- Given mature state of standards PROCESS, autonomy should be the exception, not the rule

Problem with Current Routing Environment

TOO FOOTLOOSE AND FANCY FREE

- — Few rules of operation
- — No inforcement of rules
- — No conformance testing

- The fanciest routing protocol in the world will not work without the above
  - — Routing is the hardest thing to conformance test, but the most important thing to conformance test
    - Because routing is n-party, ALL routers must work right for ANY of them to work right (slight exaggeration, but idea is correct)

New Proposed Architecture

- Do away with current subnetwork/gateway routing duality
- All network switching elements (IMPs, gateways, etc.) within administrative routing domain participate as equals in Landmark Hierarchy
- Across administrative routing domains, lack of trust and complexity of agreements may not allow Landmark Routing (too much auto-configuration is not possible in some cases)

EGP3

Key features:

- Version negotiation
- incremental updates
- Hello/IHY combined with Poll/Updates

Does <u>not</u> solve topology problems

Version Negotiation

- This RFC defines a new version

- Gwy A sends a request to gwy B in version K, A's highest implemented version

(i) B understands k, proceed

(ii) B understands n > k, not k
B sends error message in version n

(iii) B understands n < k:
B sends error message in version n, its highest implemented version.

- Gwy A receives no response:
Send a new request in version 2.

Pinging

- no explicit Hellos or IHYs
- Replaced with
  - Poll with data
  - Update
- Polls and Updates can be empty
  - Set offset to 0
- active/passive relations maintained

- Each entry in database has a sequence number (32 bits w/ wrap around)

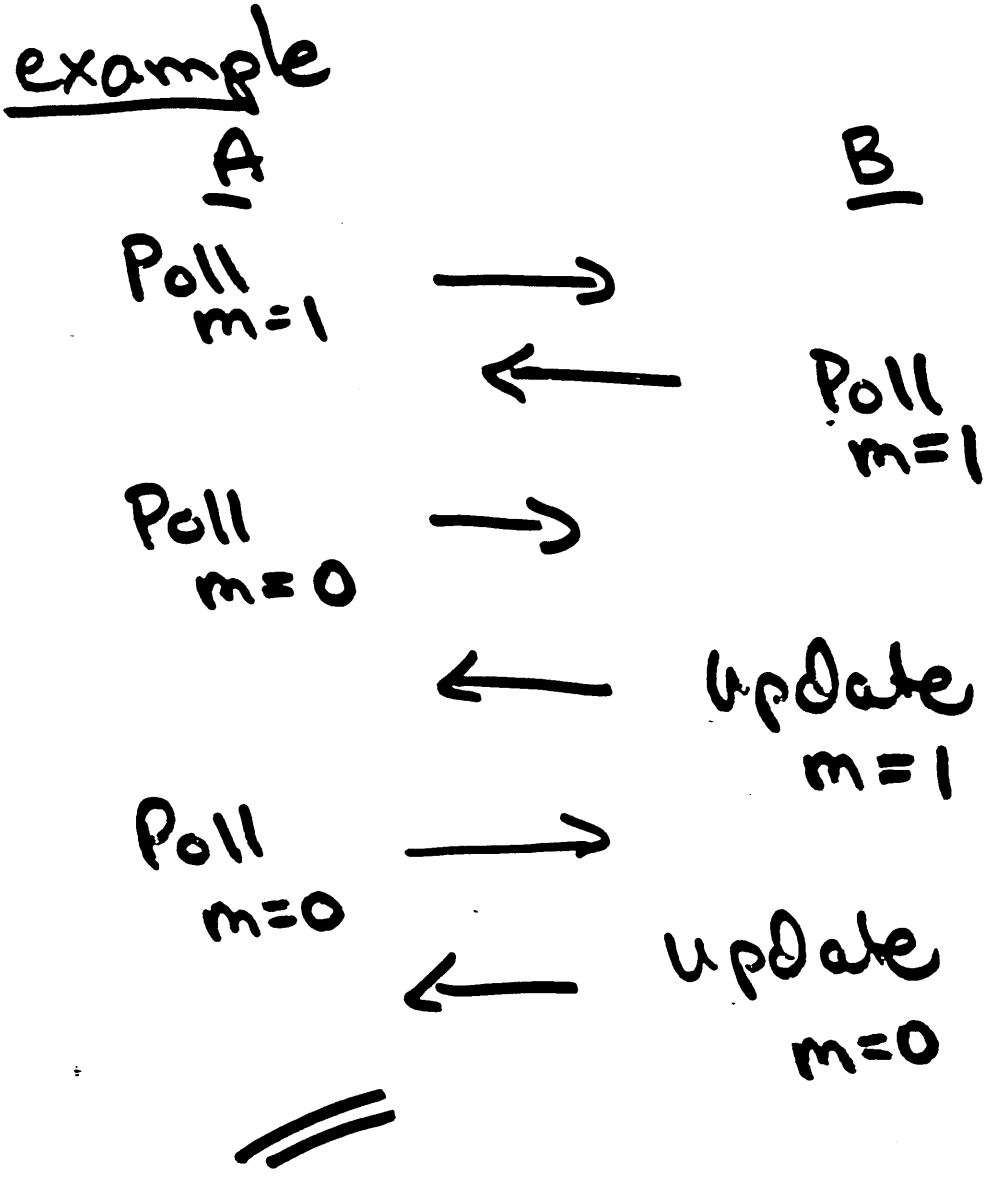
- Gateway A keeps state variables for gateway B:

  RR = seq no of last item B→A
  RS = seq no of last item A→B
  ES = last item in db when poll sequence begun

- Poll contains
  RR – send me all data after sequence no. RS
  may contain routing data as well
  RS – my data after RS
  O – offset
  last item in update has sequence no. RS+O

- If update contains more than 1 packet of data, set more bit and B will poll for more.

example



- Difference between Poll & Uplink
  Poll bit – send data
  don't send data

- At beginning of exchange, record the last sequence no. used in ES
  Exchange is over when all entries upto & incl entry ES have been sent

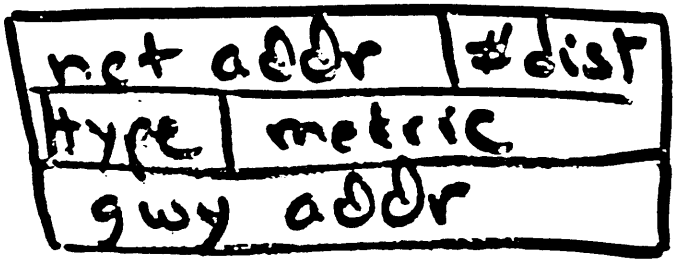
- 1 retransmission timer depends on interface

- only initial Poll counts as hello

- 4 misses = down

other decisions:

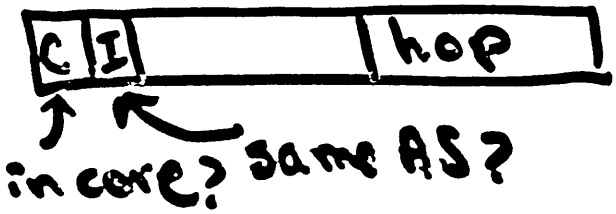
- no unsolicited updates
- no data compactification in the update



- Database refreshing
  - TTL is a function of the update
  - TTL determined during Acquire

- metrics

type 0

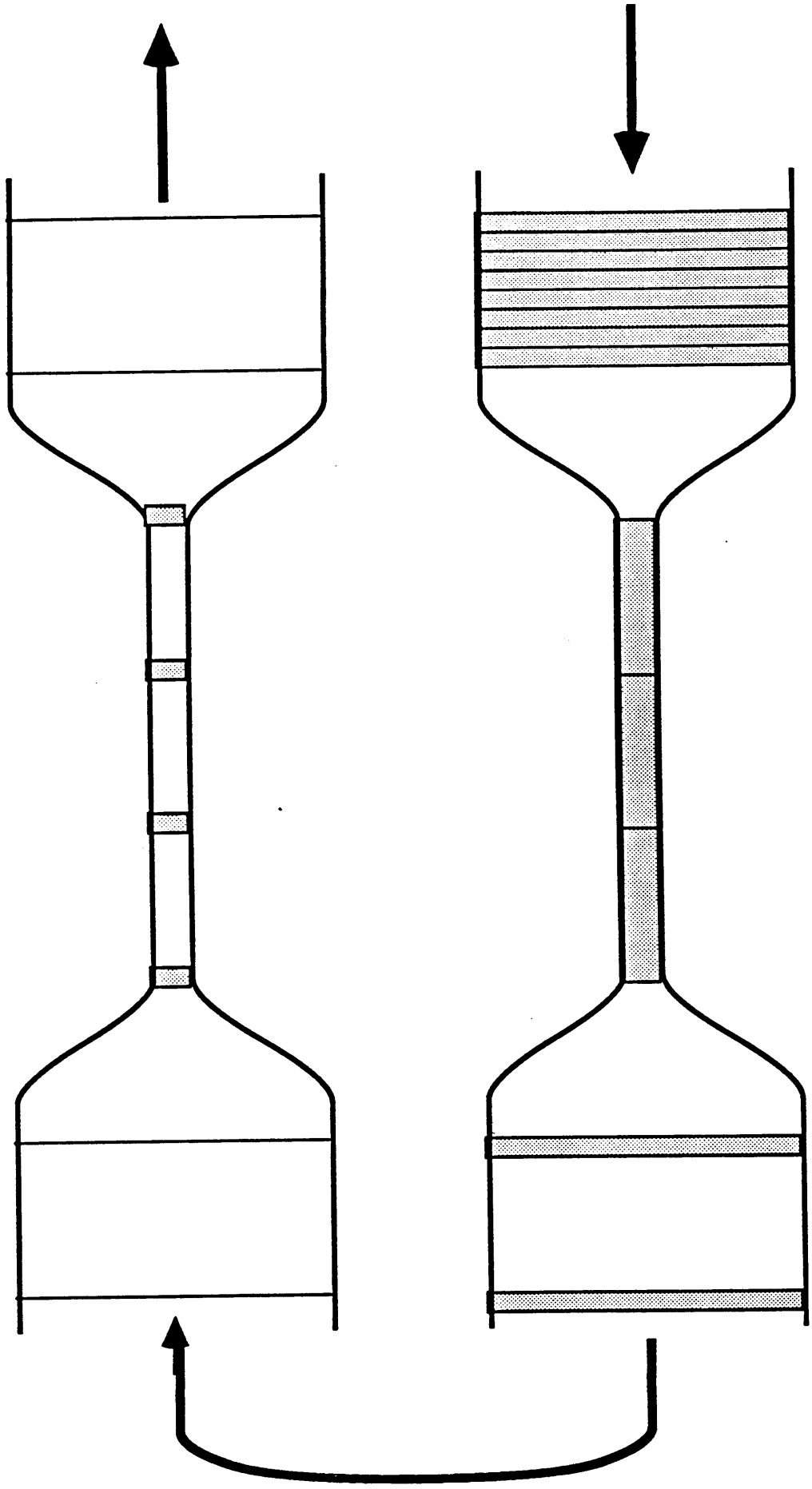


type 1

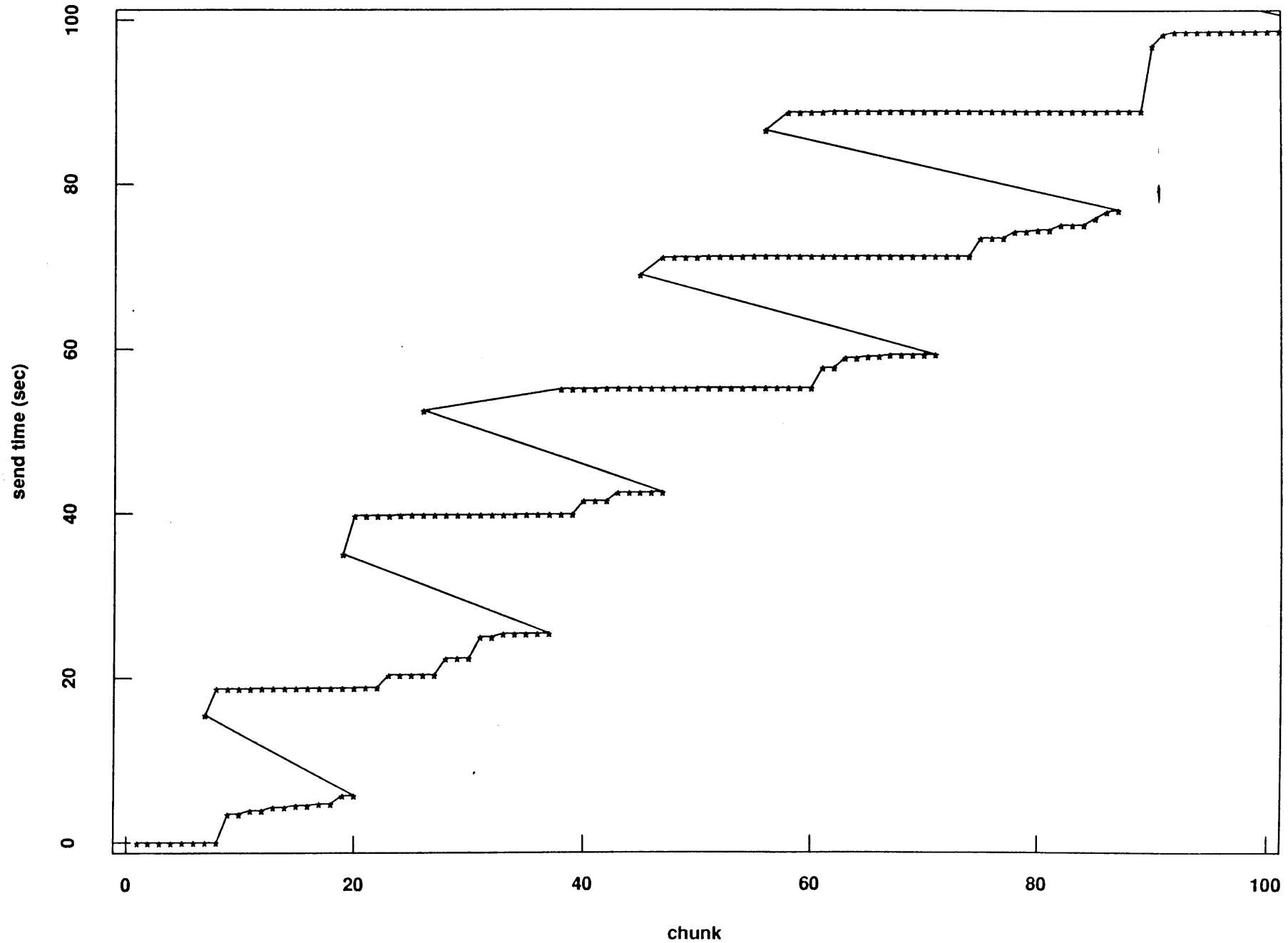
- min. time before reacquisition

ReceiveCease	TTL
Reboot	1 poll interval
1 hour time-out	4 poll intervals
SendCease	1 poll interval

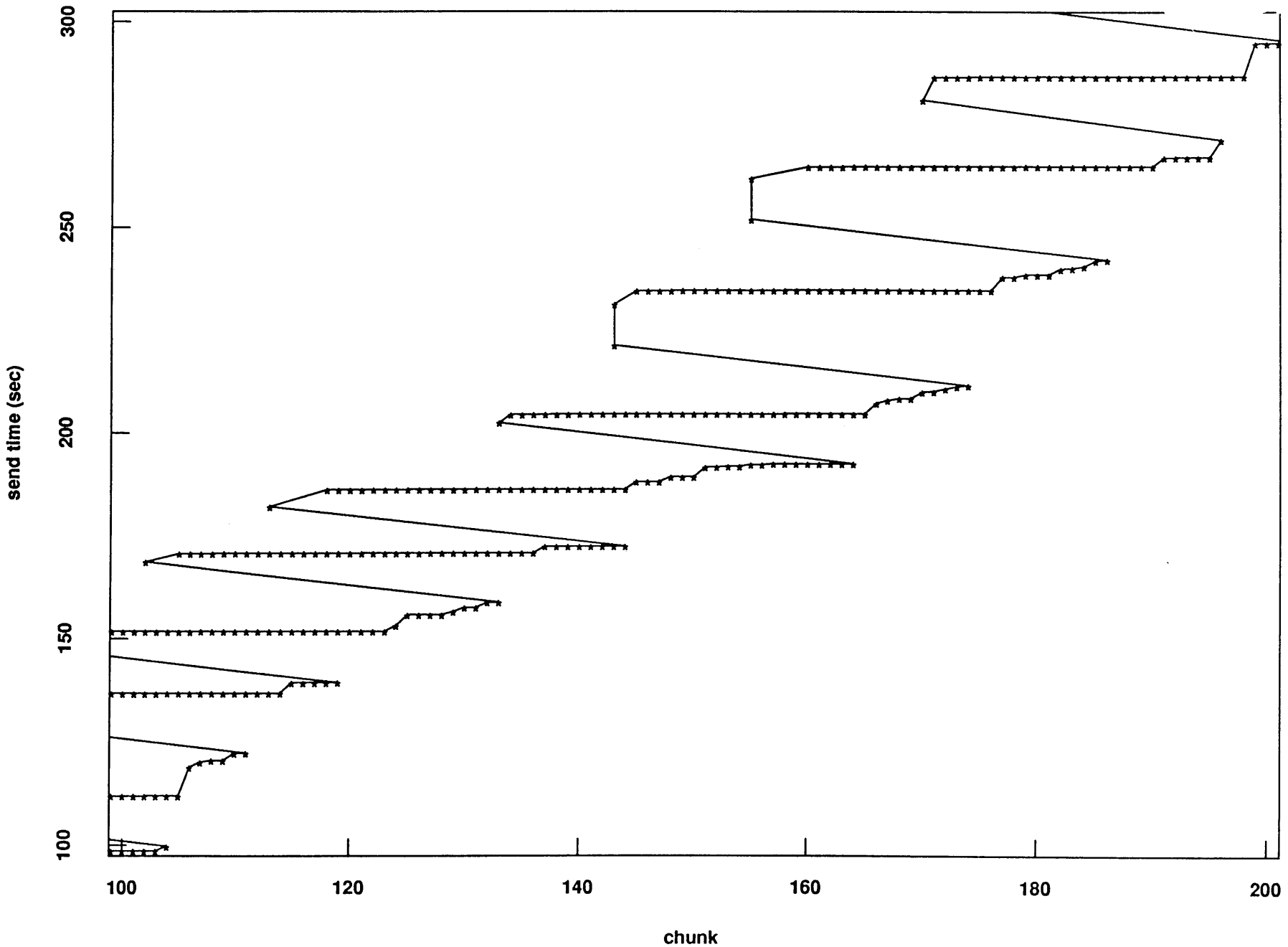
Round Trip Delay Estimation Jacobson (LBL)



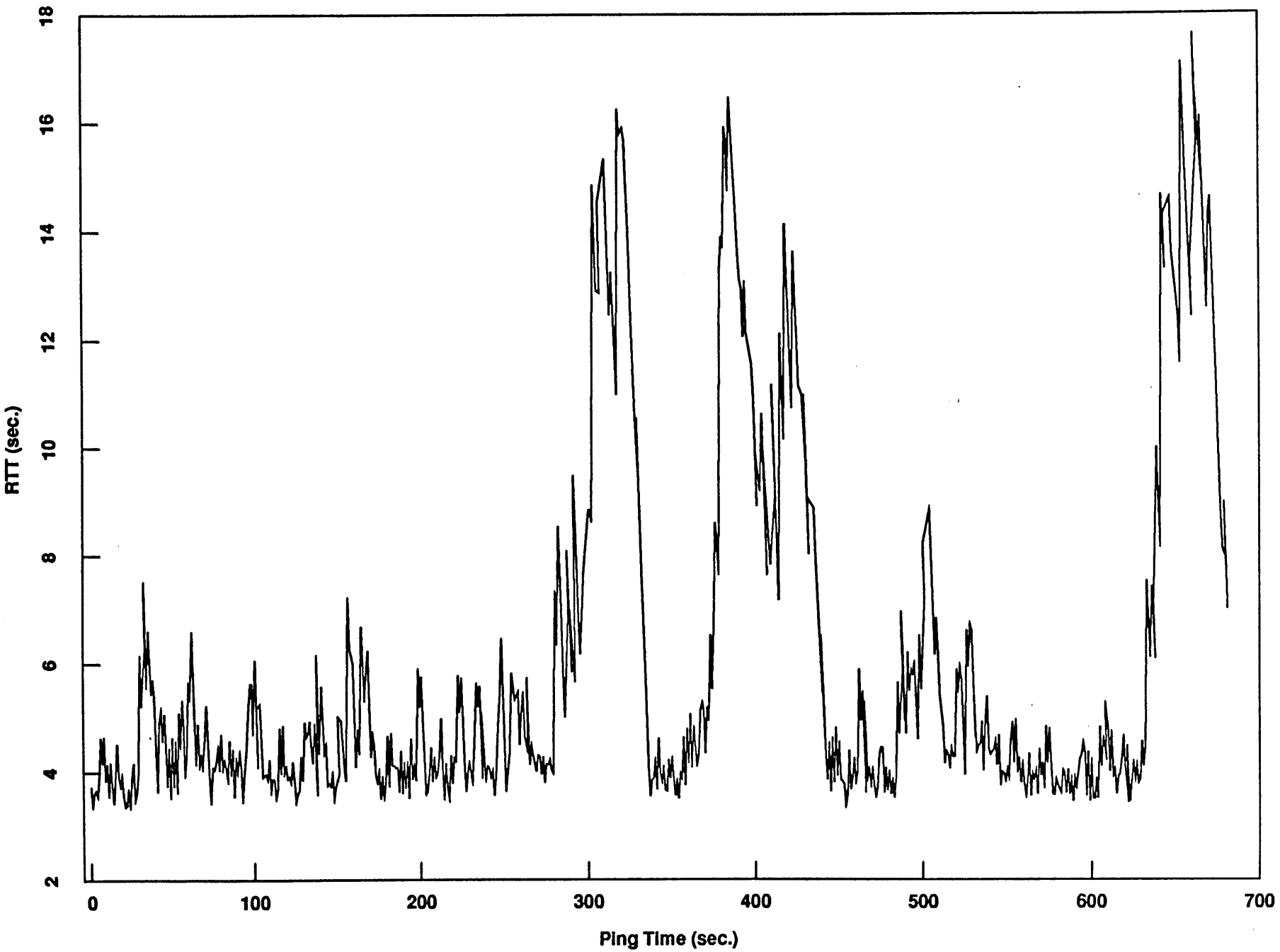
Original 4.3 Behavior, 16KB Window

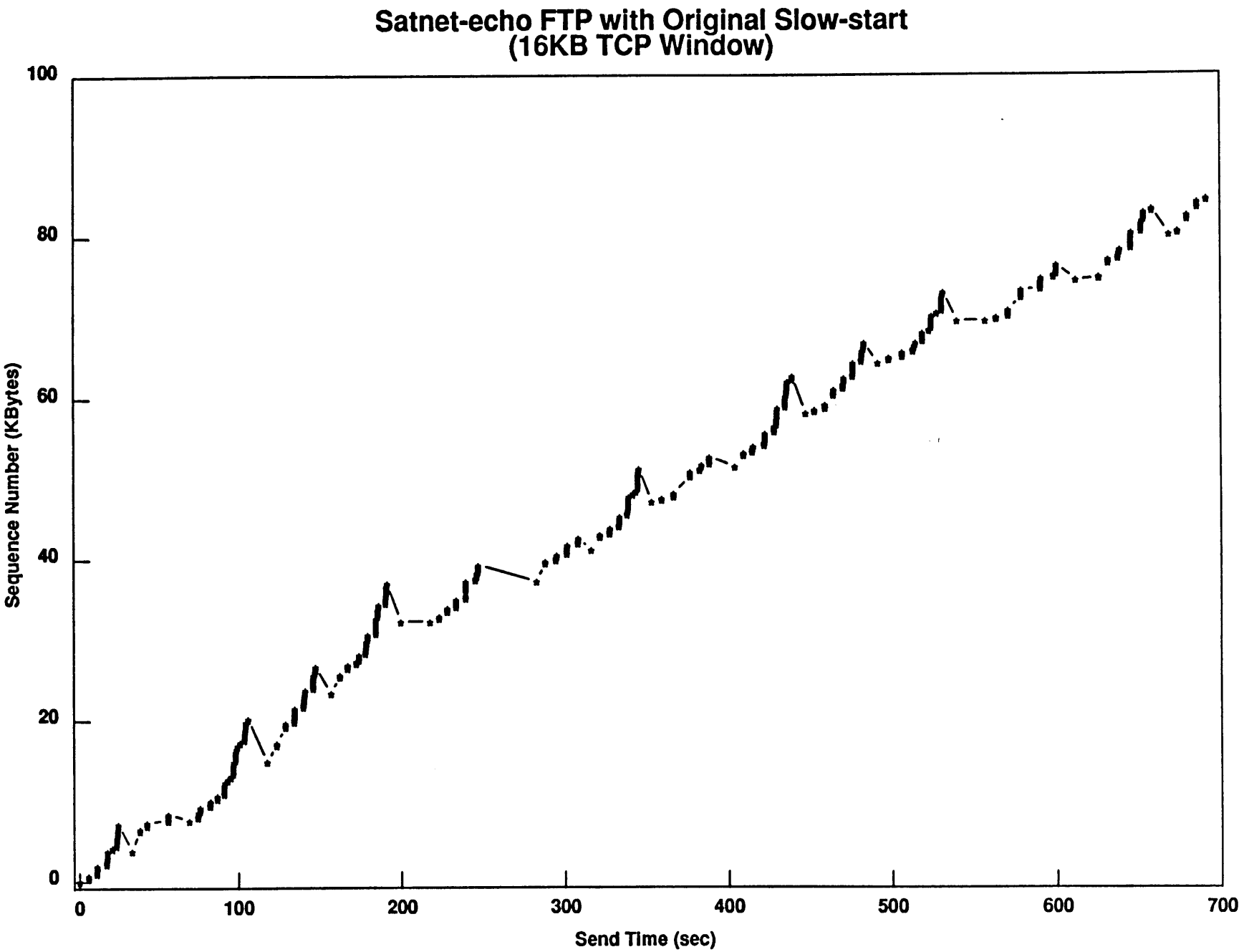


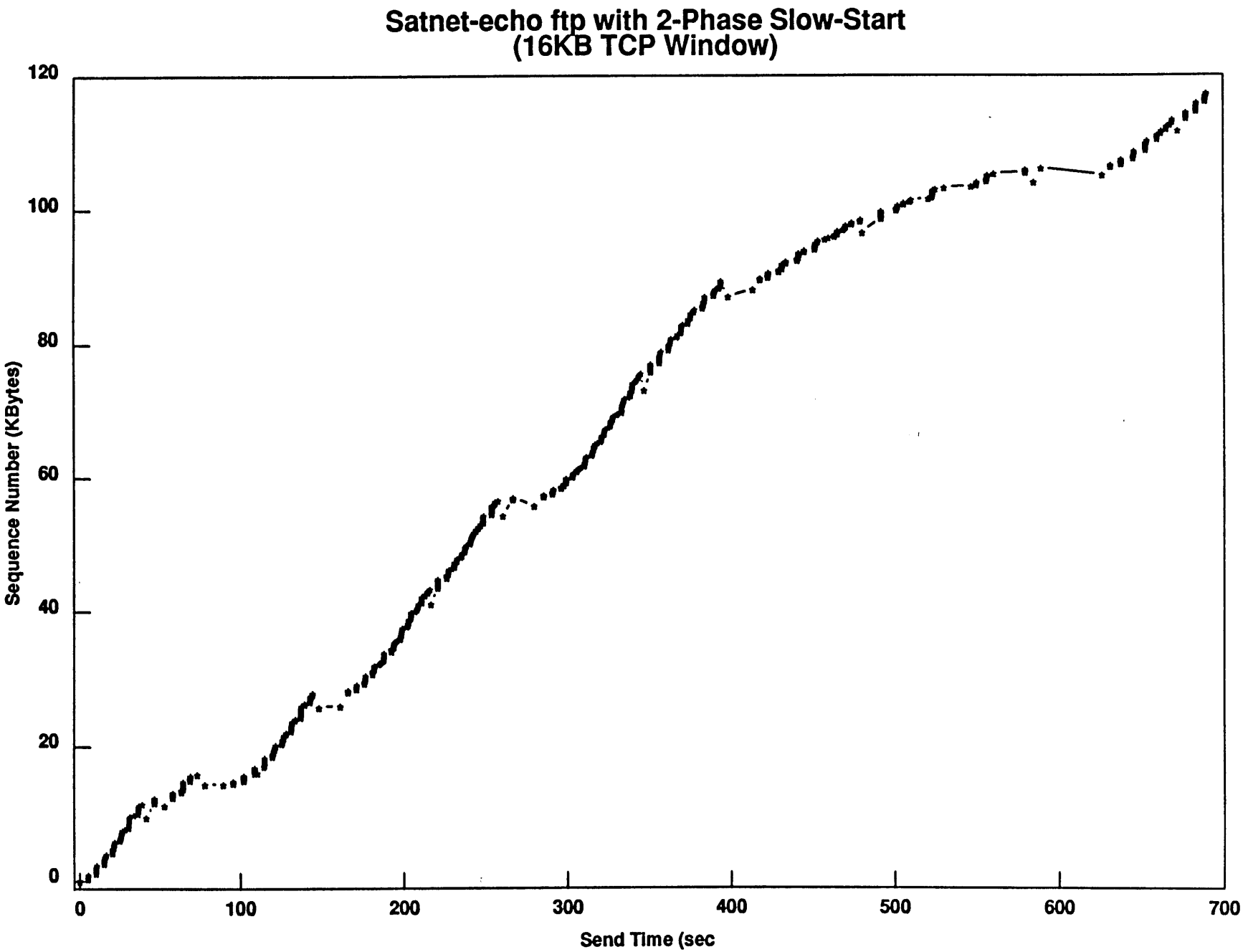
Original 4.3 Behavior, 16KB Window



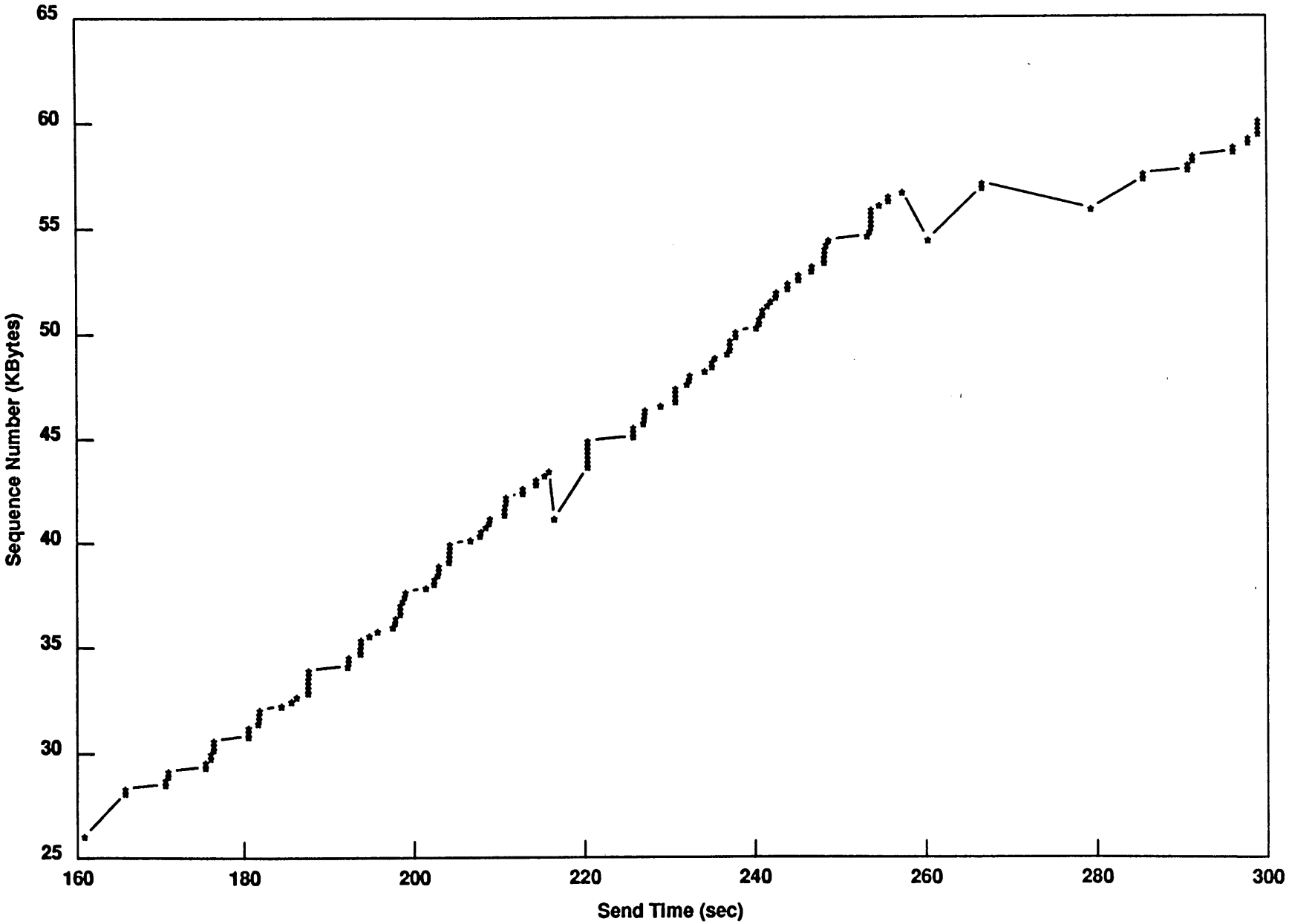
VJ (2)


Satnet-echo Ping Behavior
RTT (sec.)
2
4
6
8
10
12
14
16
18
0
100
200
300
400
500
600
700
Ping Time (sec.)

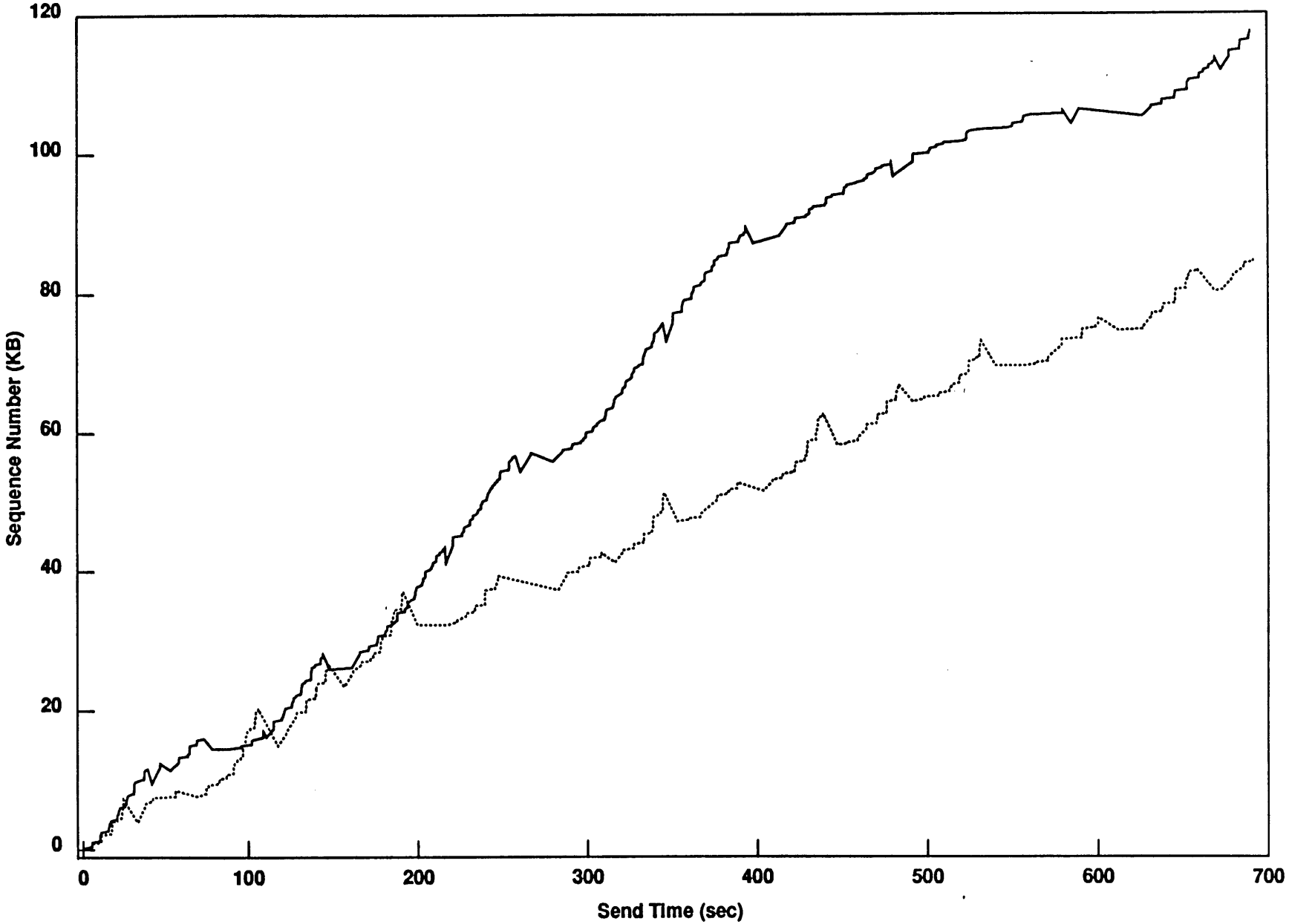

Satnet-echo FTP with Original Slow-start
(16KB TCP Window)
Sequence Number (KBytes)
0
20
40
60
80
100
0
100
200
300
400
500
600
700
Send Time (sec)

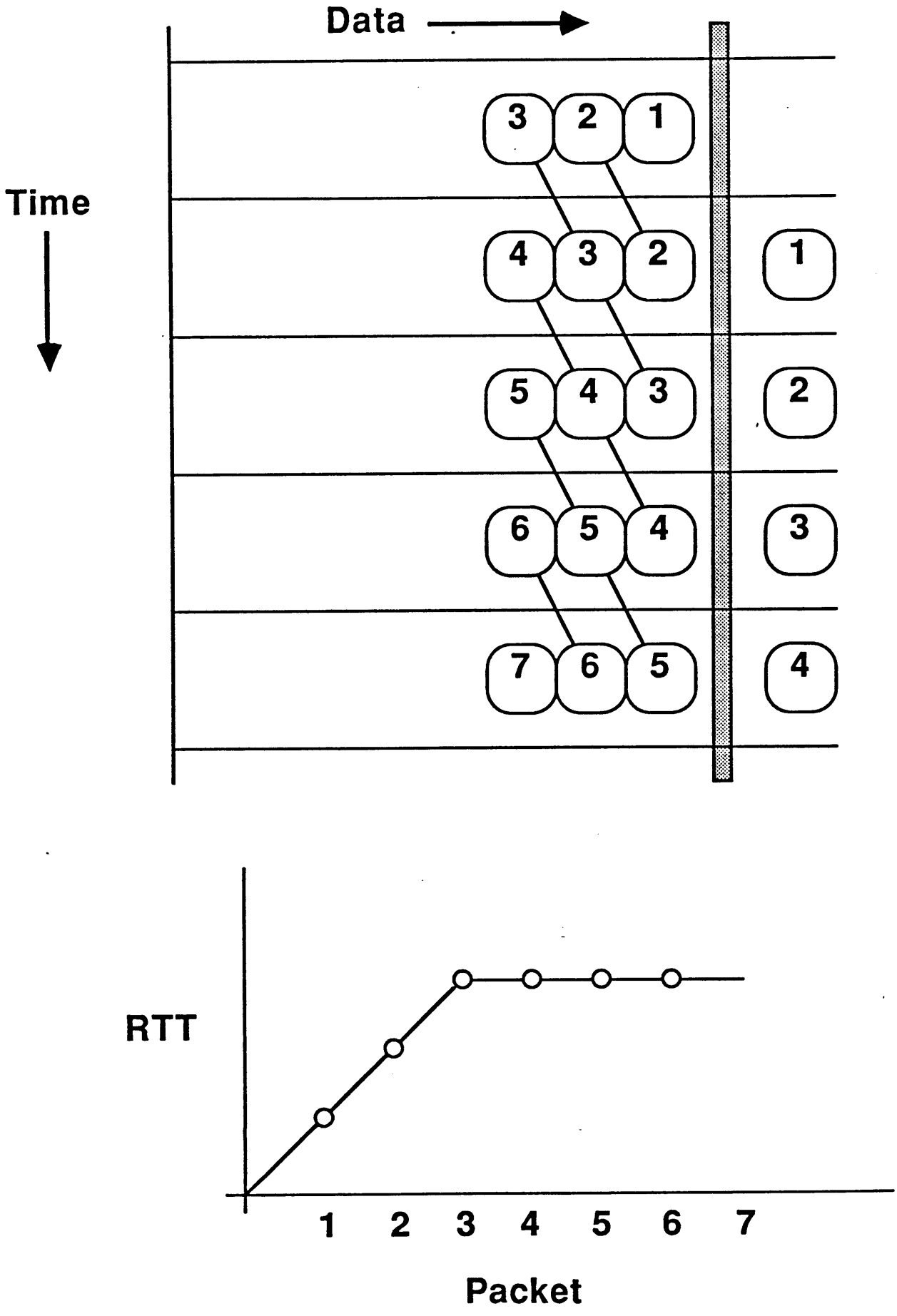

Satnet-echo ftp with 2-Phase Slow-Start
(16KB TCP Window)
Sequence Number (KBytes)
120
100
80
60
40
20
0
0
100
200
300
400
500
600
700
Send Time (sec

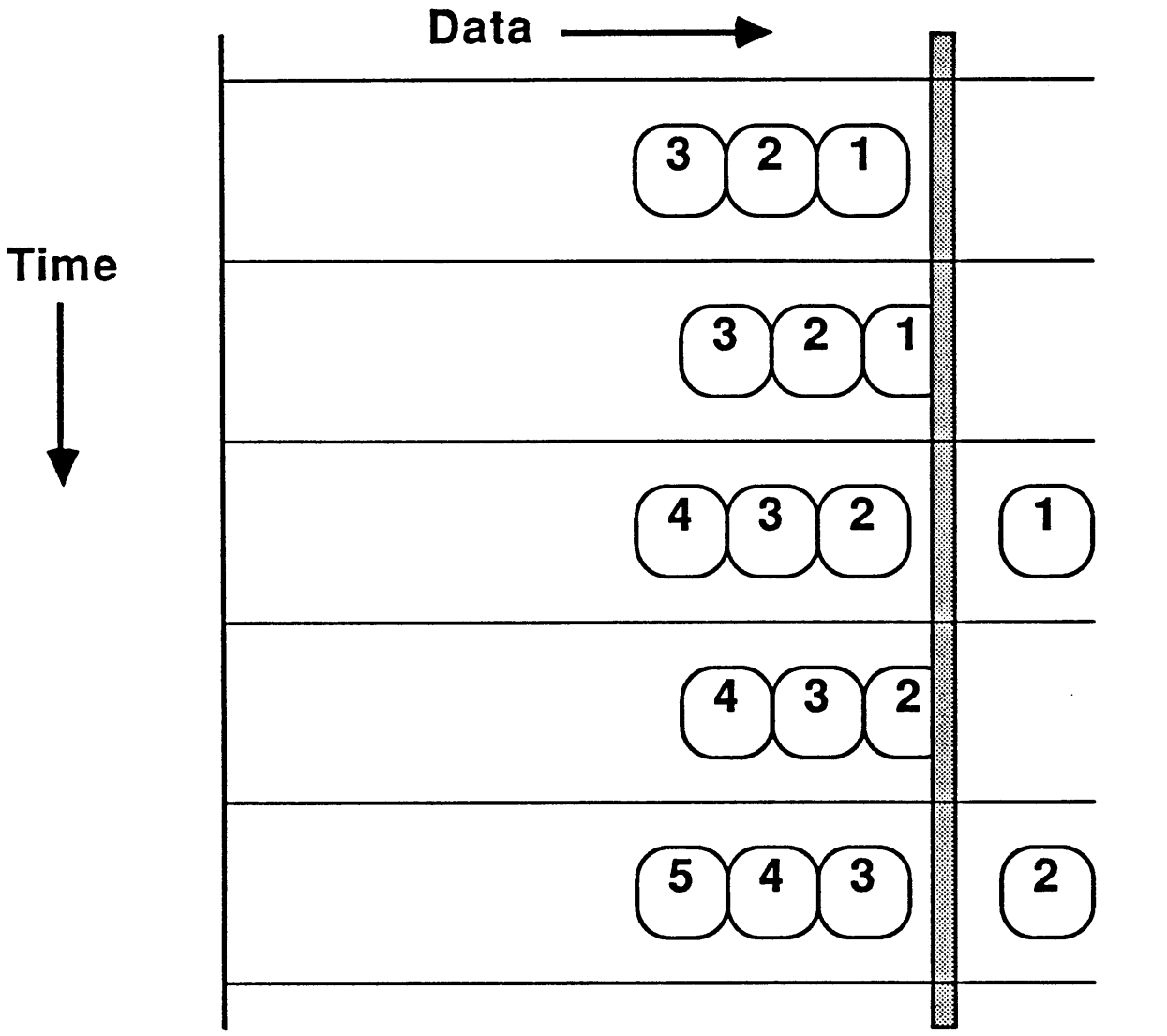
New Slow-start Detail

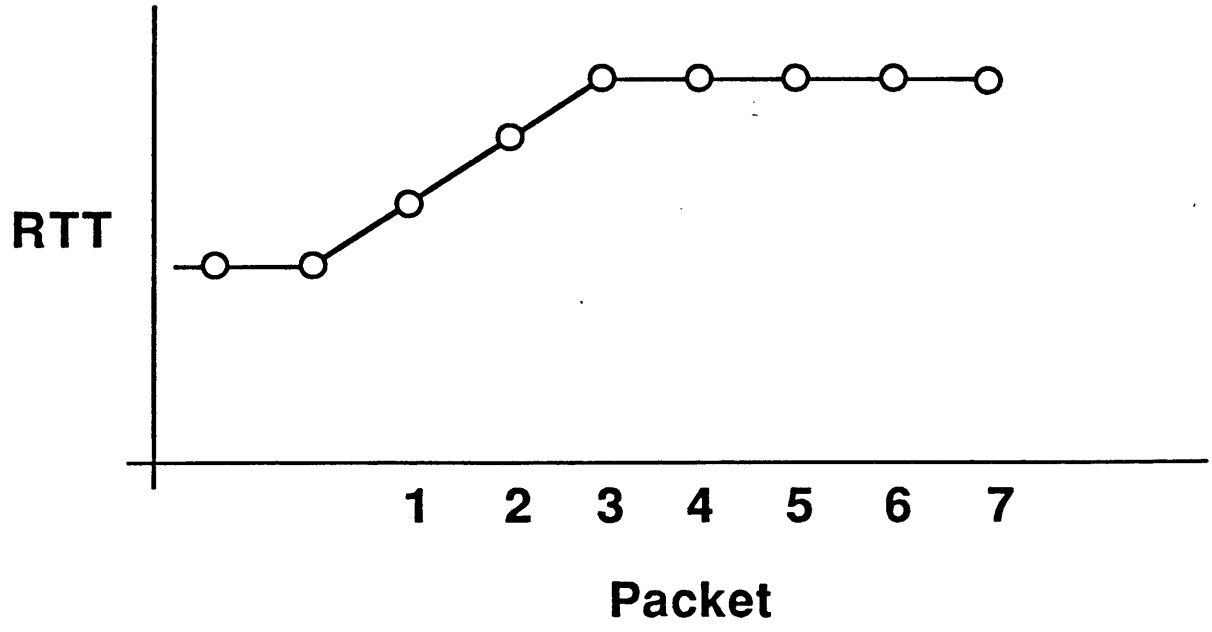


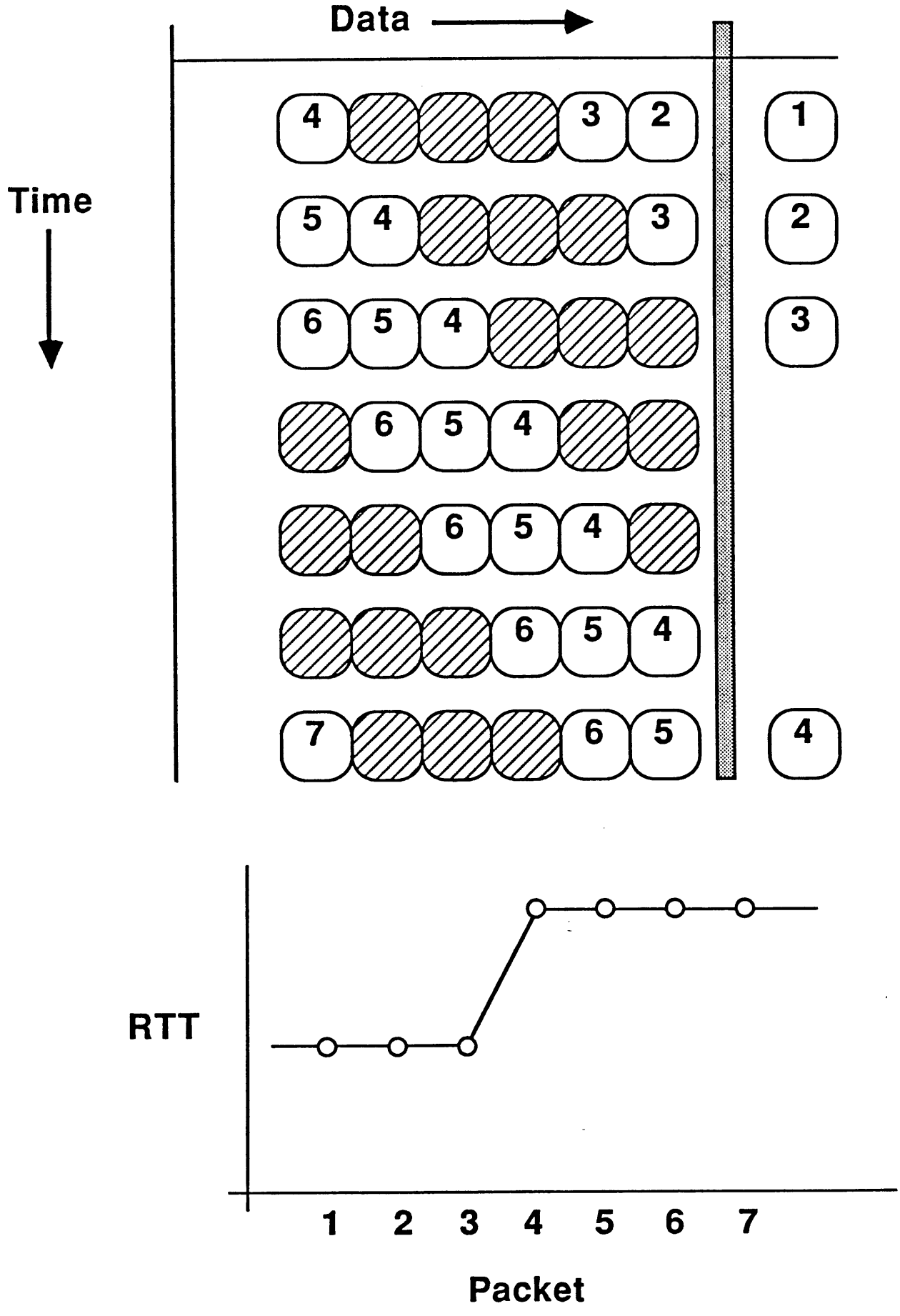
Old and New Slow-start Comparison

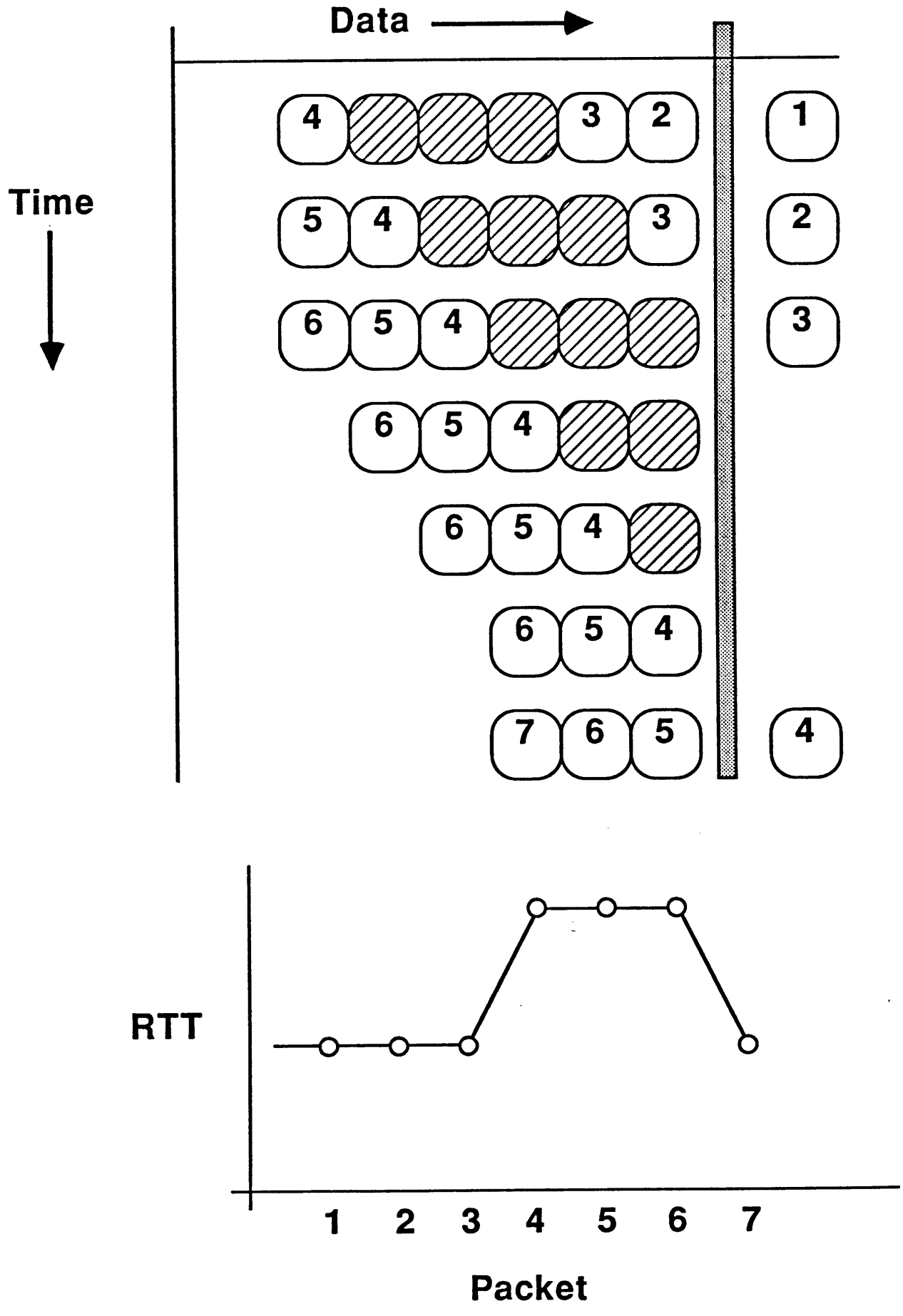


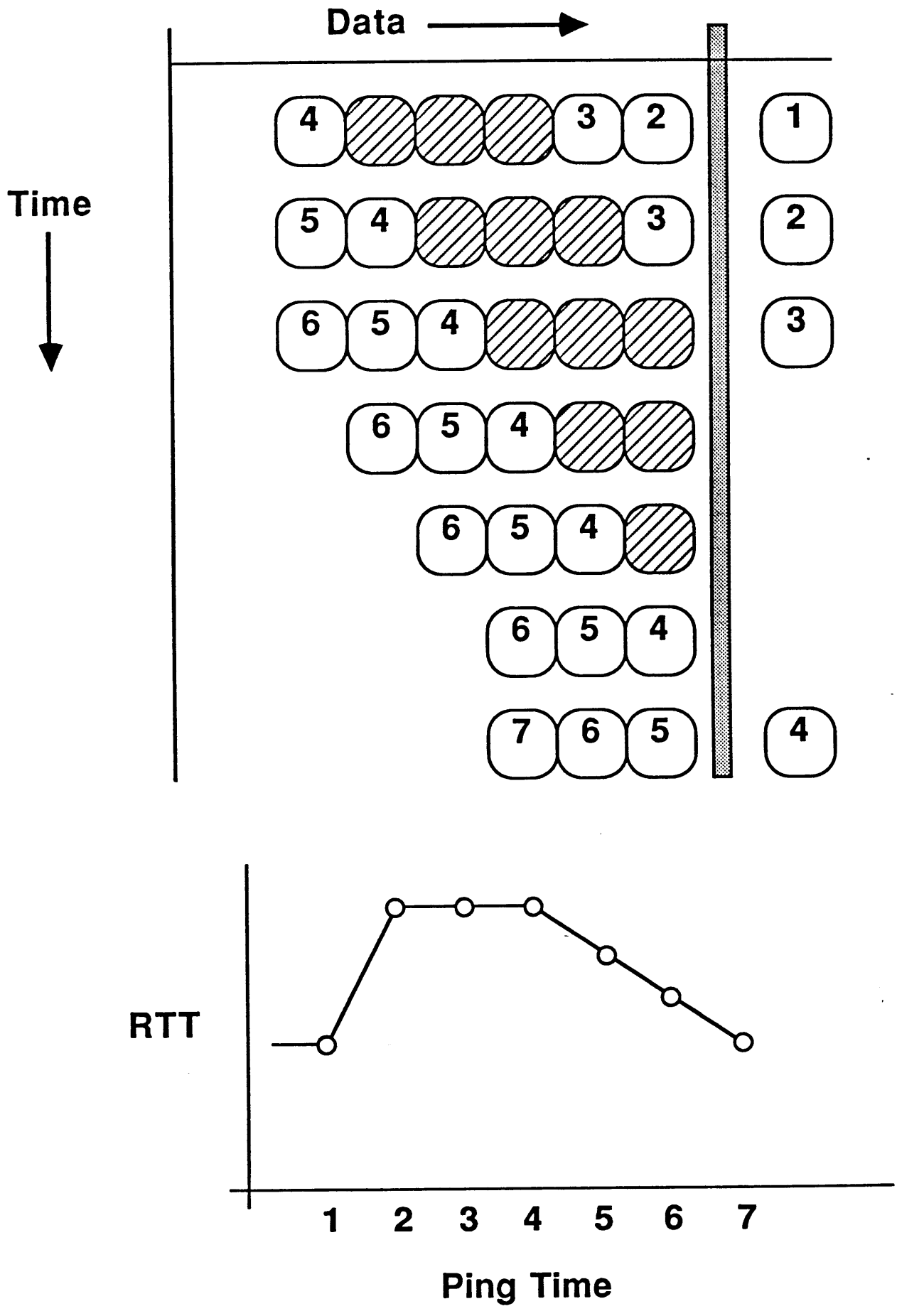

Data
Time
3 2 1
4 3 2 1
5 4 3 2
6 5 4 3
7 6 5 4
RTT
1 2 3 4 5 6 7
Packet

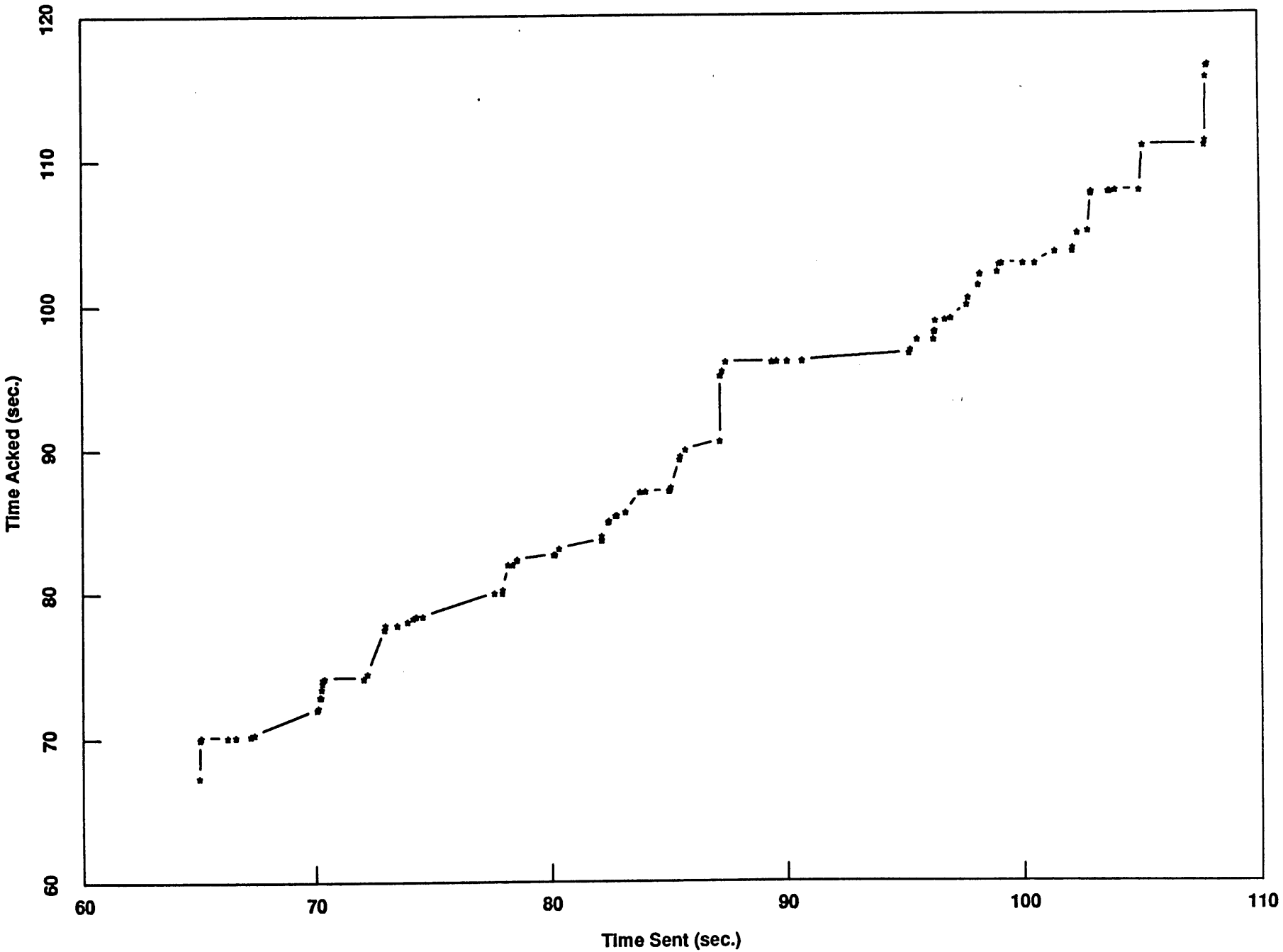

Data
Time
3 2 1
3 2 1
4 3 2 1
4 3 2
5 4 3 2

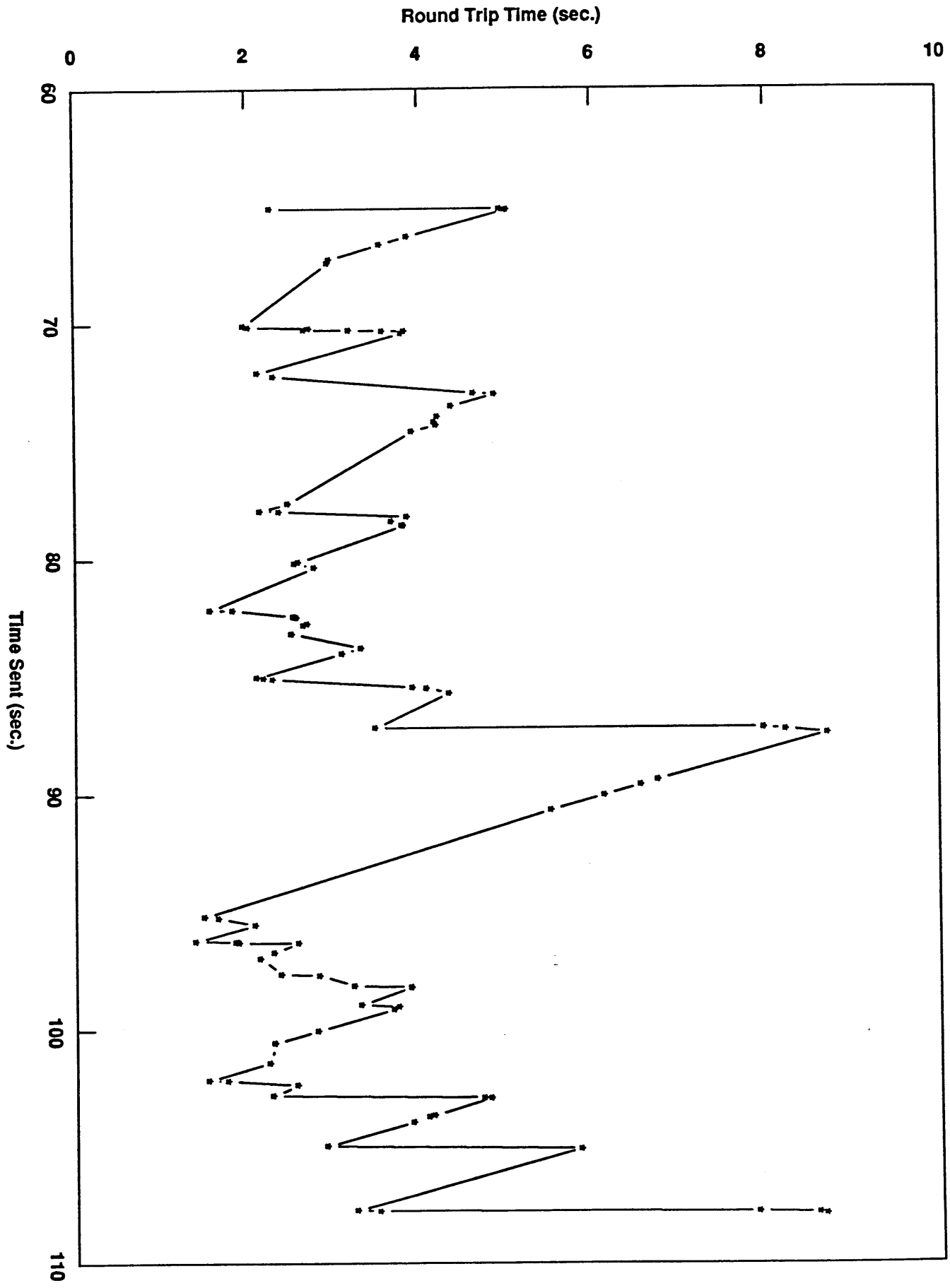

RTT
1 2 3 4 5 6 7
Packet

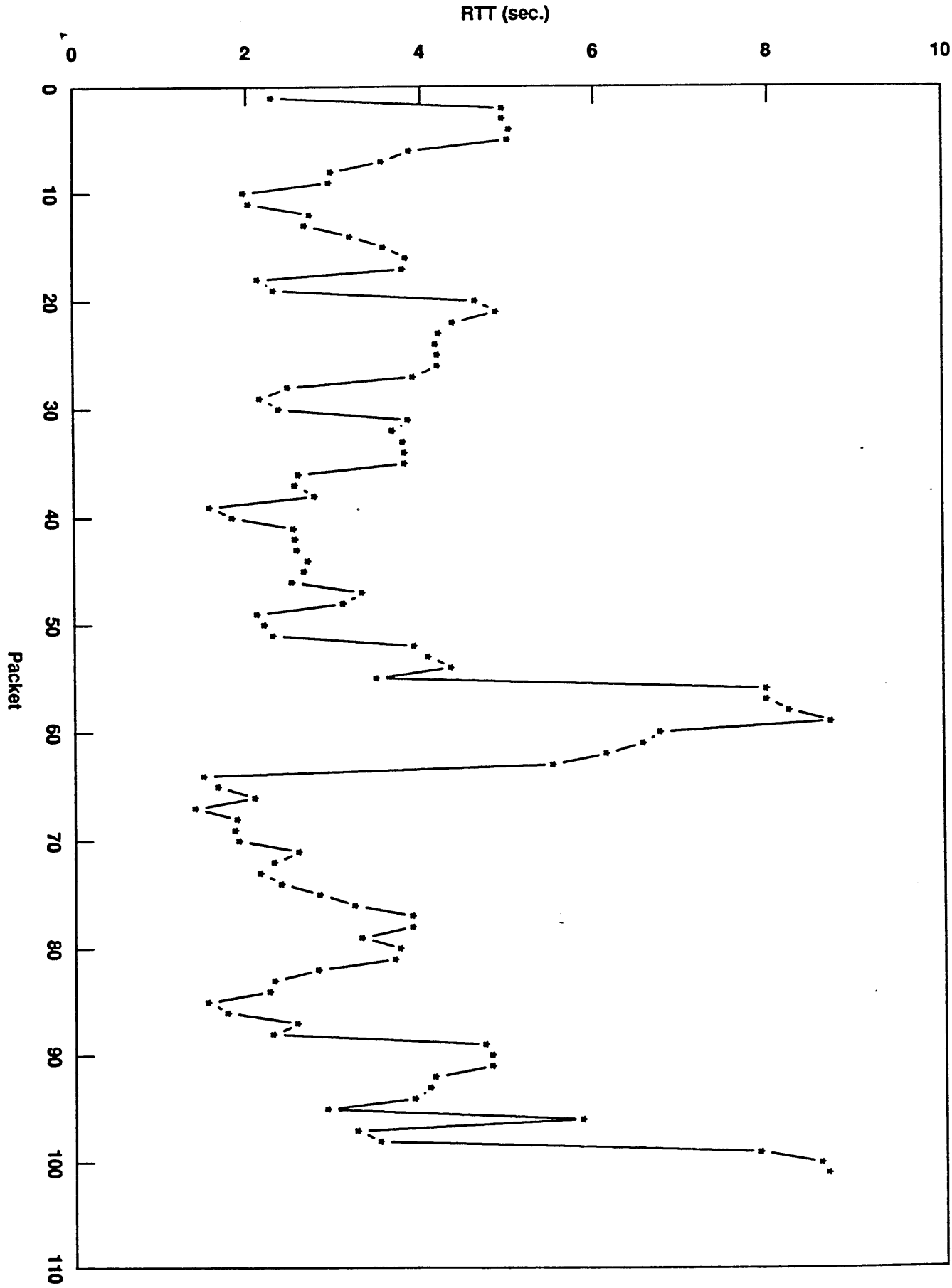

Data
Time
4
3
2
1
5
4
3
2
6
5
4
3
6
5
4
6
5
4
6
5
4
7
6
5
4
RTT
1
2
3
4
5
6
7
Packet

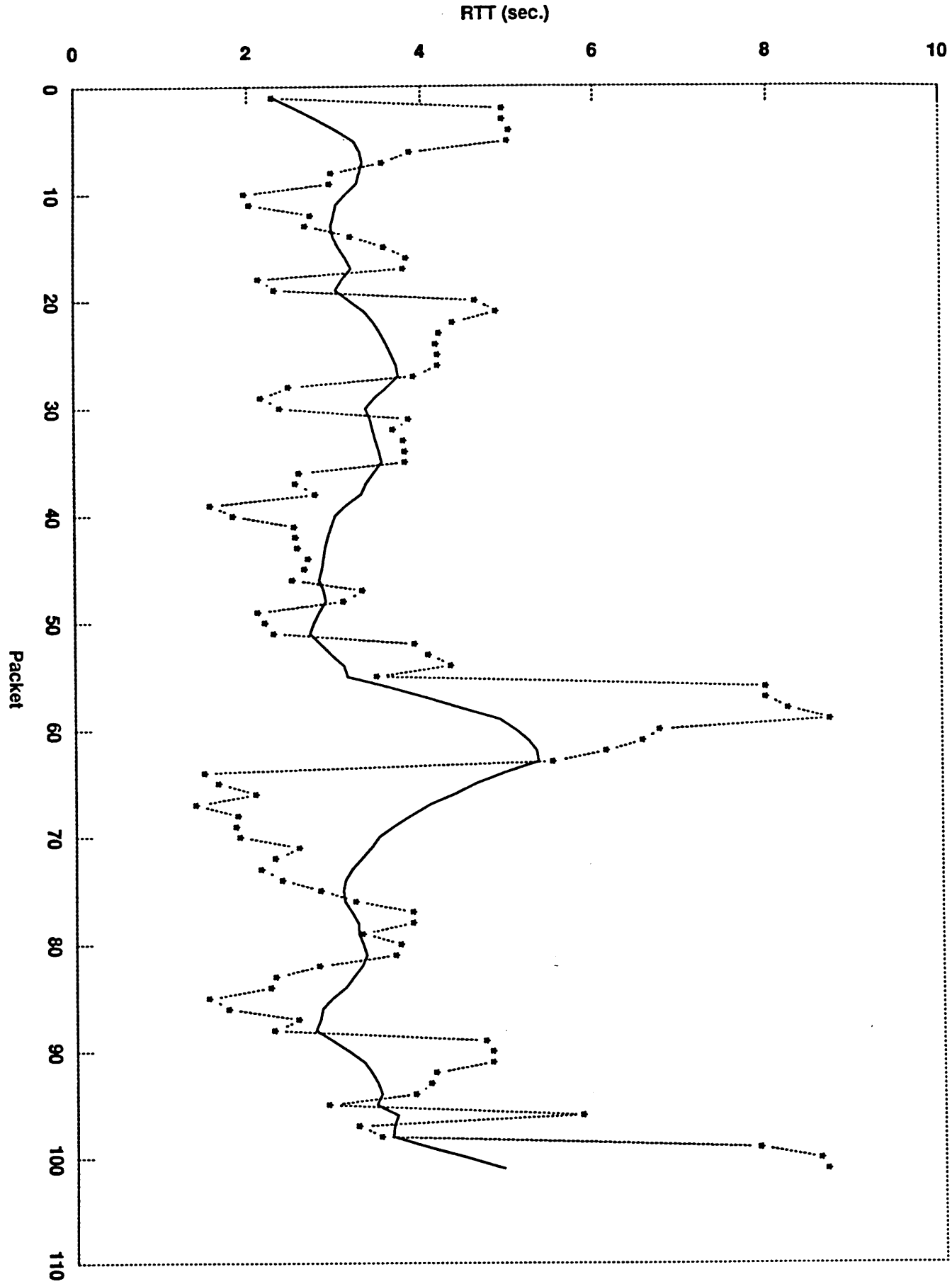

Data
Time
4
3
2
1
5
4
3
2
6
5
4
3
6
5
4
6
5
4
6
5
4
7
6
5
4
RTT
1
2
3
4
5
6
7
Packet


Data
Time
4 3 2 1
5 4 3 2
6 5 4 3
6 5 4
6 5 4
6 5 4
7 6 5 4
RTT
1 2 3 4 5 6 7
Ping Time

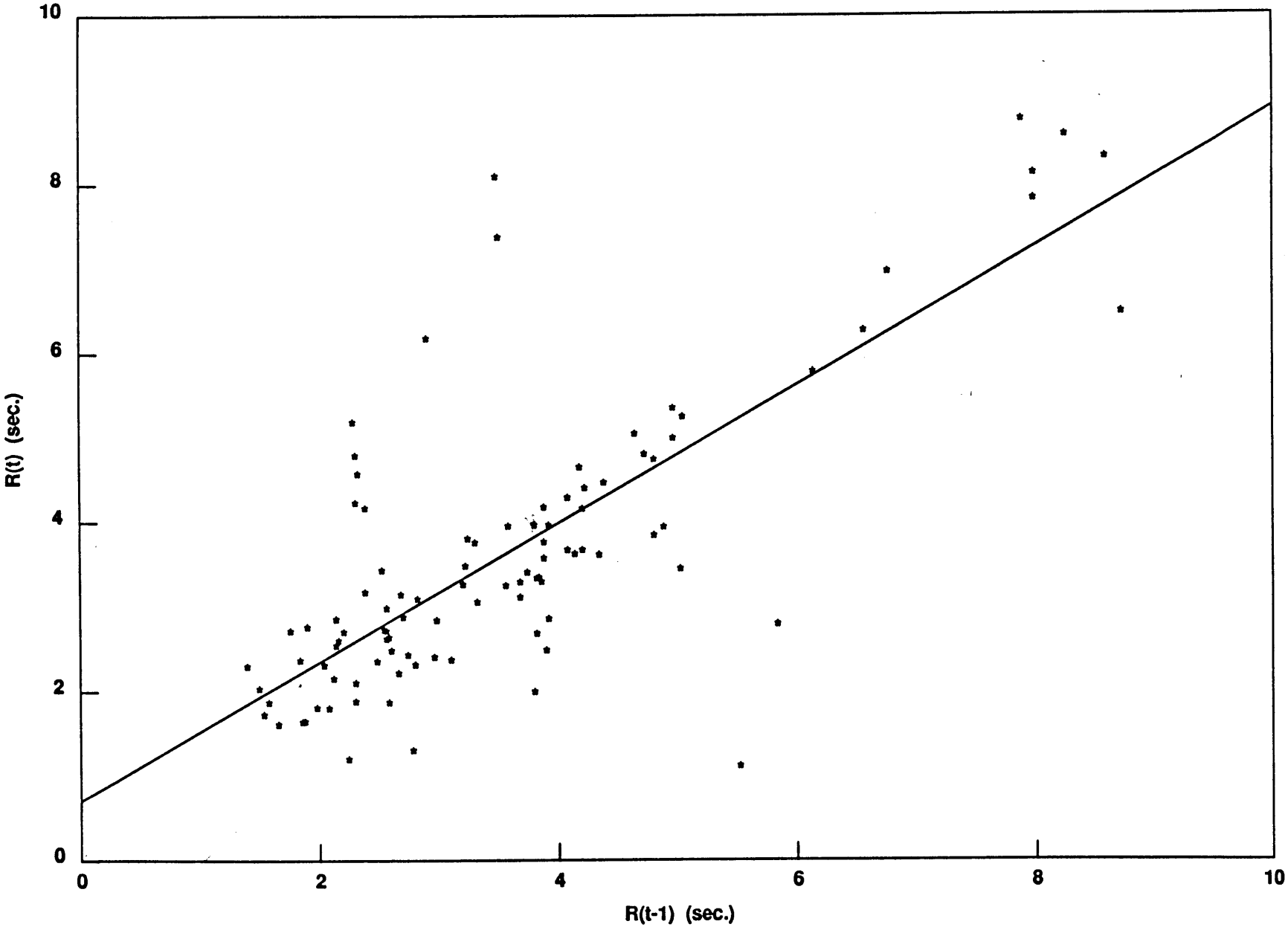

Time Acked (sec.)
120
110
100
90
80
70
60
60
70
80
90
100
110
Time Sent (sec.)


Round Trip Time (sec.)
0
2
4
6
8
10
60
70
80
90
100
110
Time Sent (sec.)

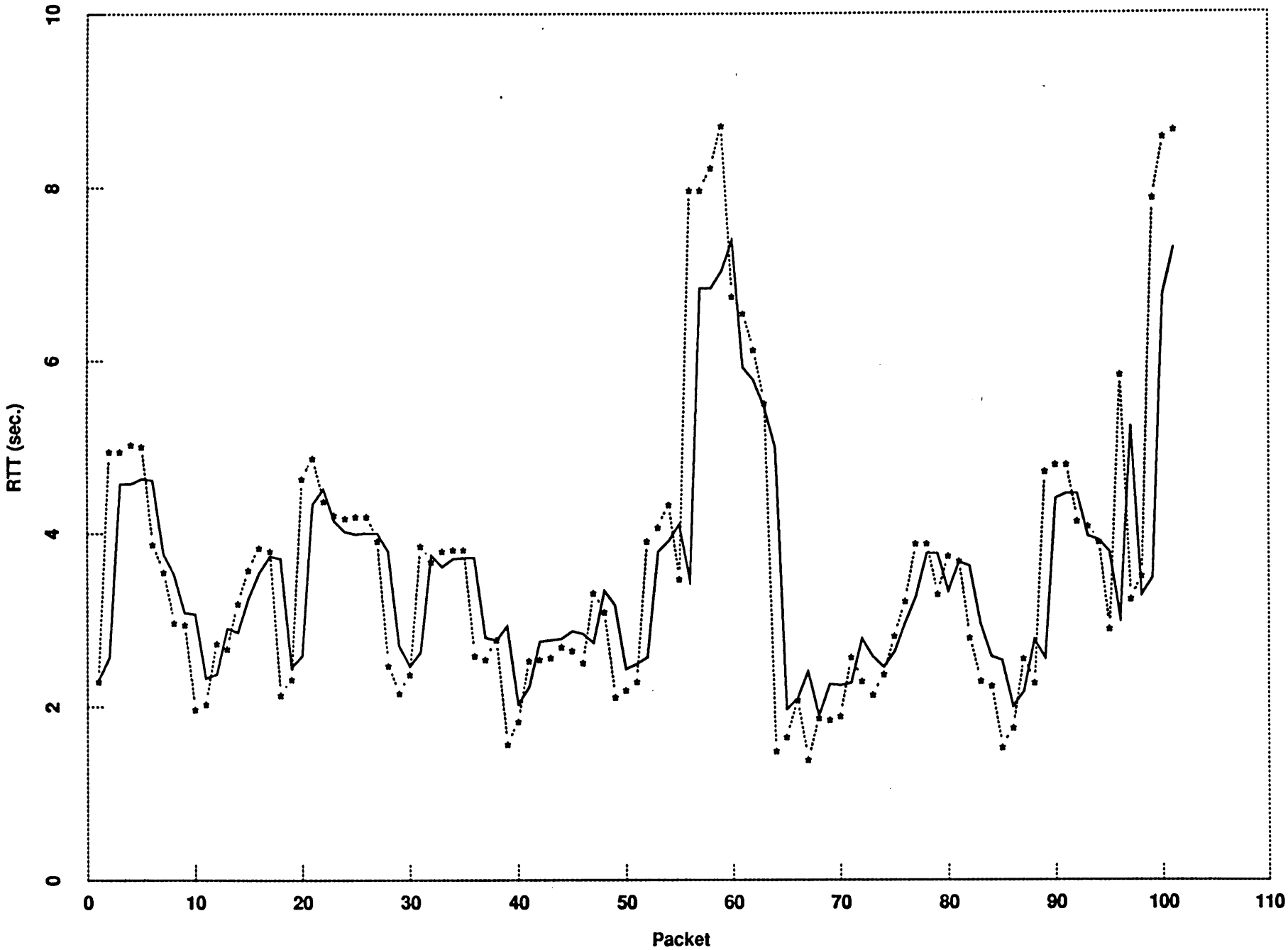

RTT (sec.)
0
2
4
6
8
10
Packet
0
10
20
30
40
50
60
70
80
90
100
110


RTT (sec.)
0
2
4
6
8
10
Packet
0
10
20
30
40
50
60
70
80
90
100
110

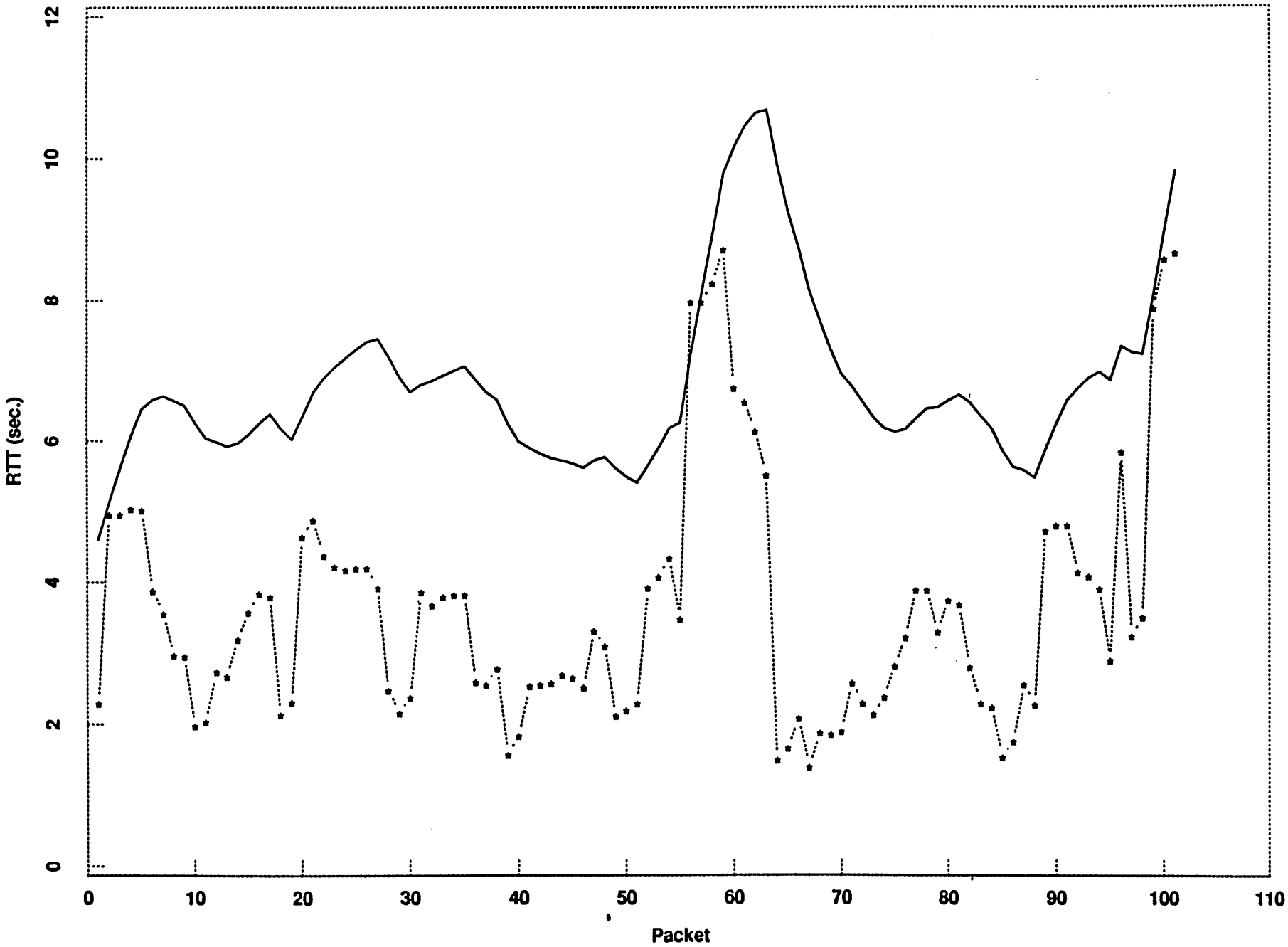
RTT Lag-1 Plot + Least Squares Fit Line



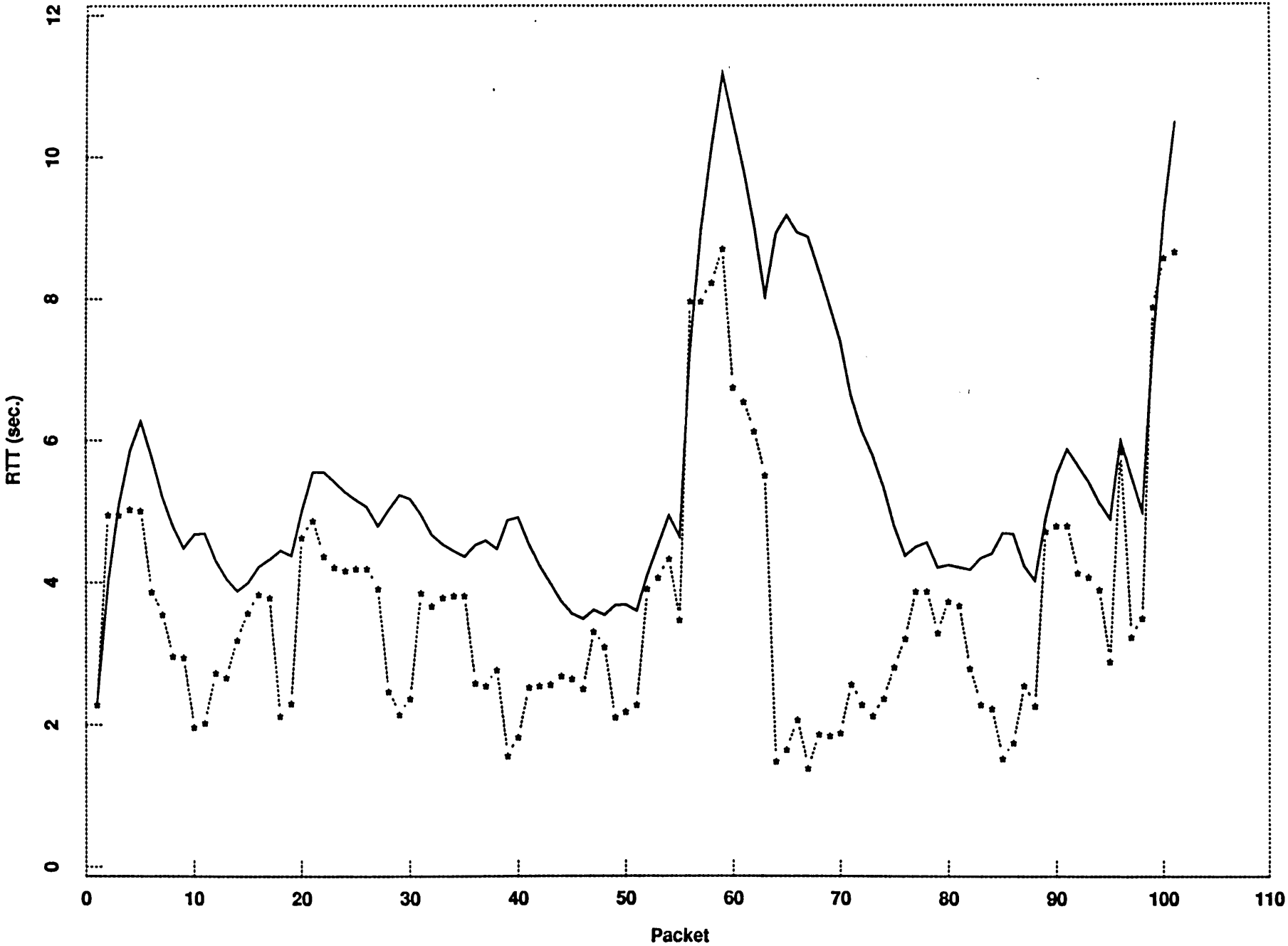
ARX(1) "fit" model



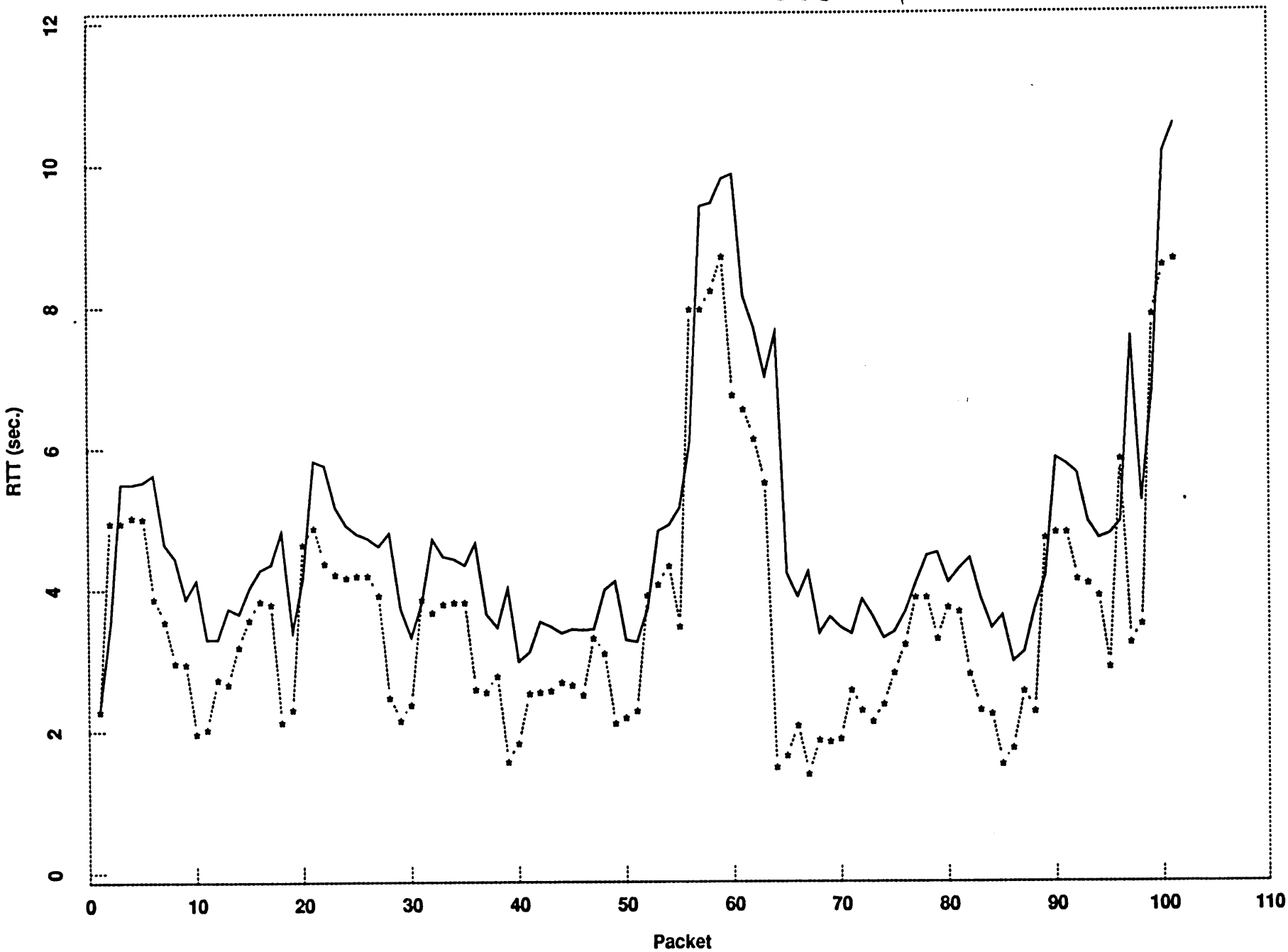
Current TCP RTO calculation



mean deviation model



fit model + prediction error



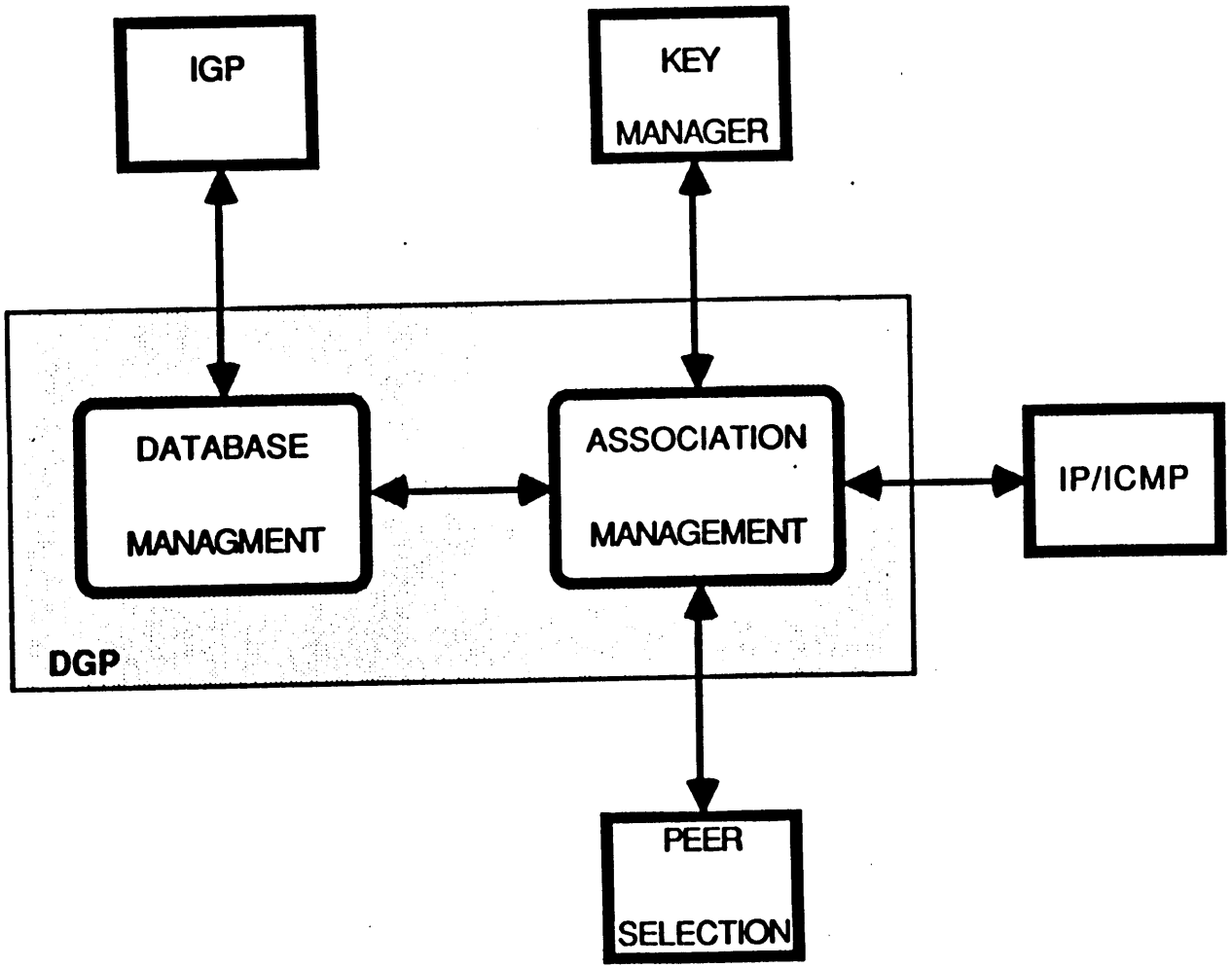
Dissimilar Gateway Protocol Little (MA/COM), Mills (UDel)

MAJOR GOALS OF PROJECT

- PROVIDE ROUTING BETWEEN A VARIETY OF AS ENVIRONMENTS
- BETTER UTILIZATION OF EXISTING CONNECTIVITY
- REMOVE THIRD-PARTY RULE
- ROUTING SUPPORT FOR:
    - MULTIPATH ROUTING
    - TYPE-OF-SERVICE
    - ROUTE RESTRICTIONS
- PEER AUTHENTICATION AND SELF PROTECTION

MACOM

DGP OVERVIEW



MACOM

INTERFACE OVERVIEW

- PEER SELECTION -
  - GATEWAY, ADDRESS, LEVEL, STATUS
- KEY MANAGER -
  - KEY REQUEST (GATEWAY, ADDRESS, LEVEL)
  - REPLY (TWO KEY AND TIMEOUT PAIRS)

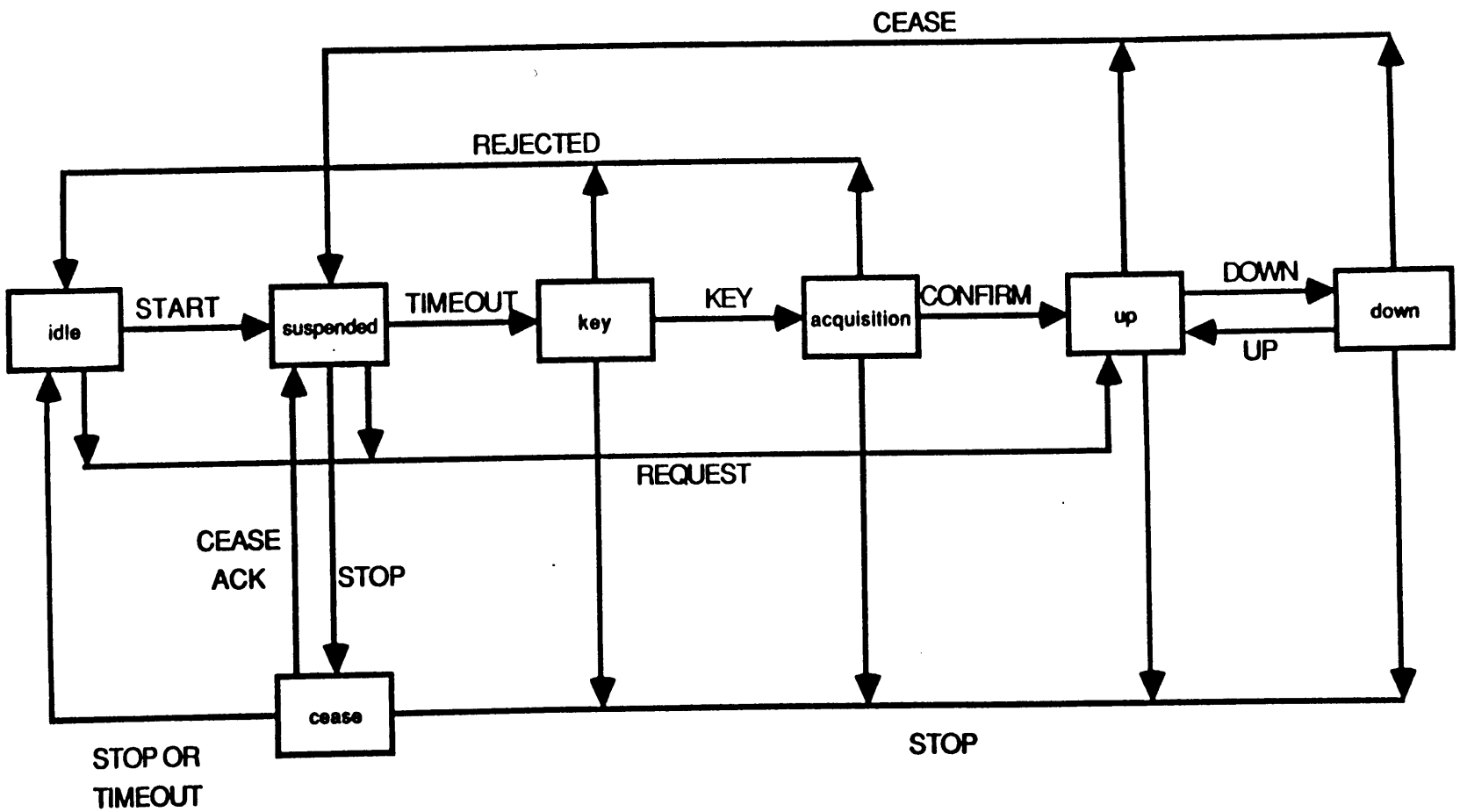
IGP INTERACTION

- FROM IGP - SYNTHESIZED INTERFACE LIST
  - GATEWAY DESCRIPTION(S)
  - NETWORKS REACHABLE (AND COST) FROM EACH GATEWAY
- TO IGP - DGP GATEWAY(S) INTERFACE LIST
  - DGP GATEWAY ADDRESSES ON IGP NETS
  - REACHABLE NETWORKS FROM DGP GATEWAY

MACOM

PEER INTERACTION

- RESOURCE REQUIREMENT NEGOTIATION
- PEER AUTHENTICATION AND SELF PROTECTION
- INTEGRATED HELLO/IHU AND UPDATE MESSAGES
- SUSPENSION OF COMMUNICATIONS



PEER ASSOCIATION STATE MACHINE

DATABASE MANAGEMENT

- UPDATES -
  - INCREMENTAL
  - PRIORITIZED
- DISSEMINATION -
  - PERIODIC UPDATE
  - RELIABLE FLOODING
- AGING -
  - EXPIRATION
  - DELETION

NYSACOM

PROTOTYPING

- PARAMETERIZATION
- DATABASE DYNAMICS
- ROUTE DETERMINATION
- FULL PEER STATE MACHINE
- PRIORITY QUEING AND TRANSFER CONTROL
- DRAWBACK - NO FORWARDING FUNCTION MODIFICATION

NYACOM

TEST ENVIRONMENT

- THREE SUN WORKSTATIONS and A FILESERVER
- MULTIPLE DGP INSTANCES PER WORKSTATION
- UP TO 15 SYSTEMS OF 100 NETWORKS EACH
- SCRIPT SCENARIO CAPABILITY FOR IGP
- NETWORK CHARACTERISTICS - SIMULATED AND REAL

NASCOM

CURRENT ISSUES BEING TIED DOWN

- SPECIFIC ROUTING ALGORITHM
- INFORMATION HIDING
- METRIC CONVERSION/ASSIMILATION

3.0 Distributed Documents

The following documents and papers were distributed at the meeting. As indicated, a number of them are drafts. The EGP document is under current revision. For copies or additional information, please contact the authors or the SRI Network Information Center.

Routing Information Protocol
Draft RFC (Hedrick)

Routing Information Protocol: Revised Metric
Draft RFC (Hedrick)

Proposal to ANSI X3S3.3 for ISO IS-IS Intra-Domain
Routing Exchange Protocol (DEC)

Proposal to ANSI X3S3.3 for ISO IS-IS Intra-Domain
Routing Exchange Protocol (UNISYS)

A Simple Gateway Monitoring Protocol
Draft RFC (Davin, Case, Fedor, Schoffstall)

Design Overview for a UNIX Version of SGMP
(Schofftall, Shikarpur, Yeong)

The Landmark Hierarchy: Description and Analysis
Draft MITRE Technical Report (Tsuchiya)

Exterior Gateway Protocol, Version 2
Draft (BBN, July 1987)