

Original

**Proceedings of the
16-17 January 1986
DARPA
Gateway Algorithms and Data Structures
Task Force**

**Prepared by:
Phillip Gross**

FIRST IETF

**The MITRE Corporation
MITRE C²I Division
Washington C²I Operations
1820 Dolley Madison Boulevard
McLean, Virginia 22102**

TABLE OF CONTENTS

Minutes of the Fourth DARPA GADS Task Force Meeting	<i>Page</i> 1
APPENDIX A Hardcopy of GADS Presentation Slides	13
APPENDIX B Papers Distributed at GADS Meeting	125

Minutes of the Fourth DARPA GADS Task Force Meeting

**Minutes of the
Fourth
DARPA Gateway Algorithms and Data Structures
Task Force Meeting**

16-17 January 1986

Prepared by

Phillip Gross
Mitre Corporation

Gateway Algorithms Task Force

Table of Contents

1. Introduction	1
2. Attendees	1
2.1 Members in Attendance (16)	2
2.2 Additional Attendees (5)	2
3. Meeting Notes	2
3.1 16 January 1986	3
3.2 17 January 1986	5
4. Addenda	6
4.1 Distributed Agenda	7
4.2 Reference Documents for this Meeting	8
4.3 Proposed Charter of the Internet Architecture Task Force (INARC)	9
4.4 Proposed Charter of the Internet Engineering Task Force (IETF)	9

1. Introduction

The fourth meeting of the DARPA Gateway Algorithms and Data Structures Task Force was held 16-17 January 1986 at M/A Com Government Systems in San Diego, California. The meeting was hosted by David Mills.

Acknowledgments: Thanks to Noel Chiappa, Zaw-Sing Su, and Carl Rokitanski, who responded to requests for information with very helpful comments. Profuse thanks to Pat Keryeski, who performed the onerous task of editing these minutes and compiling the Proceedings.

Gateway Algorithms Task Force

2. Attendees

2.1 Members in Attendance (16)

Name	Organization	Net Address
Braun, Hans-Werner	U. of Mich.	hwb@gw.umich.edu
Brescia, Mike	BBNCC	brescia@bbnccv
Callon, Ross	BBN Labs	RCALLON@BBN-UNIX
Chiappa, Noel	MIT/Proteon	jnc@mit-xx
Eldridge, Charles	Sparta	eldridge@edn-vax
Gross, Phill	MITRE	Gross@mitre
Hinden, Robert	BBNCC	hinden@bbnccv
Mathis, James	SRI	MATHIS@SRI-KL
Mills, David (Chairman)	Linkabit	Mills@USC-ISID
Nagle, John	Ford Aerospace	jbn@FORD-WDL1
Natalie, Ronald	BRL	RON@BRL
Rokitansky, Carl	DFVLR	ROKI@USC-ISID
Shacham, Nachum	SRI	Shacham@SRI-TSC
Su, Zaw-Sing	SRI	ZSu@SRI-TSC
Topolcic, Claudio	BBN Labs	topolcic@bbn-unix
Zhang, Lixia	MIT-LCS	LIXIA@MIT-XX

2.2 Additional Attendees (5)

Clark, David	MIT-LCS	dclark@mit-multics
Corrigan, Mike	DCA	corrigan@ddn1
Deering, Steve	Standford	deering@ju-pescadero
Means, Robert	M/A Com	esi@isid
St Johns, Mike	DCA (B612)	stjohns@sri-nic

Gateway Algorithms Task Force

3. Meeting Notes

3.1 16 January 1986

The Chair opened the meeting by announcing that the agenda had been substantially changed by recent events. The most important being the eminent demise of the Gateway Algorithm and Data Structures Task Force (GADS) and the formation of two new task forces in its place: the Internet Engineering Task Force (INARC) and the Internet Architecture Task Force (IETF). The INARC will focus on long term research issues and will continue to be chaired by Dave Mills. The IETF will concentrate on short term operational problems and will be chaired by Mike Corrigan. Proposed charters for these new groups are included with these minutes.

Further, the proposed joint meeting that will meet in the afternoon with the National Science Foundation (NSF) subcommittee (on interconnectivity for supercomputer networks), needed to be restricted due to space limitations. Therefore, it was proposed that Mike Corrigan chair the first session of the IETF that afternoon.

The remainder of the morning was spent listening to brief status reports and discussing various issues. The following paragraphs contain the highlights.

- 1) Hinden announced that some Butterflies would be installed by 1 March. Since a Butterfly should be able to handle up to 1000 networks, work being done on the LSI gateways (to allow the Butterflies to handle up to 300 networks) should be complete within six months. Hinden also distributed the latest Internet-on-a-chip graphic.
- 2) Nagle had been evaluating commercially available gateways and gave interesting comments on several. He also commented on the Multinet gateway, calling it a "gateway to provide isolation". His work on congestion control in gateways and a gateway database protocol will be reported in detail later in the meeting.
- 3) Mills discussed several papers on a new service enhanced model for the Internet: Autonomous Confederations and the Network Time Protocol.
- 4) Clark was very concerned with recent ISO developments. He gave his "seven year wave and trough cycle" analysis, in which three year waves of research were followed by four or more years of integration of that research into operational products. He suggested that ISO lived in the calmer seas of the trough. He distributed copies of the proposed Host-Gateway Protocol (or, in ISO parlance, End System to Intermediate System Routing Protocol) and planned to discuss it in detail on the following day. He advocated the switching of ISO Internet Protocol (IP) datagrams in the Internet gateways. This led Mills to suggest that a proposal for mapping Internet addressing onto the ISO scheme was needed. Callon volunteered to present a possible arrangement on the following day.

In the afternoon (while the Chair and several members attended the NSF Gateway Subcommittee) Corrigan chaired, what amounted to as, the initial IETF meeting.

Although there were numerous topics of immediate operational concern (Subnets, routing in the host IP layer, EGP, and switching ISO datagrams were all mentioned in an opening discussion), Corrigan focused

Gateway Algorithms Task Force

the discussion on the following areas:

IETF Areas of Concern -

- o Protocol Development and Stabilization,
- o Protocol Conformance,
- o An Implementors Support Organization,
- o Internet Performance Measurements,
- o ISO Conversion.

The remainder of the afternoon consisted primarily of an organizational brainstorming session (of IETF Areas of Concern) by members who produced the following three groups of topics:

Protocol Development and Stabilization -

- 1) Immediate Concerns (three months - one year):
 - o EGP Improvements,
 - o EGP Table Control,
 - o Specification of Host IP Requirements including:
 - Multi-Homed Hosts
 - Subnets
 - o IP Implementation Guidelines for Congestion Avoidance,
 - o TCP Specification Update,
 - o Host Interface Specification.
- 2) Intermediate Concerns (one year - three years):
 - o Improved Internet Performance (one order of magnitude),
 - o EGP Replacement,
 - o Gateway Load Sharing,
 - o Internet Access Control and Authentication (liaise with Privacy TF),
 - o Protocol Requirements for Transportable Hosts,
 - o Name/Address Service,
 - o Name/Address Convergence with ISO.
- 3) Longer Term Concerns (three years - seven years):
 - o Improved Internet Performance (two to three orders of magnitude),
 - o Large Scale Internet Routing including:
 - Partitioned Network Support
 - Multi-Path Routing
 - Type-Of-Service Routing
 - Mobile Hosts
 - o Real Congestion Control,
 - o Logical Internet Addressing,
 - o IP Multi-Cast Addressing.

The most pressing topics of immediate concern listed above fall into two broad categories: EGP modifications and IP implementation guidance. It is proposed that these topics become the focus of the next IETF meeting, which has been scheduled for 8-9 April 1986 at the Ballistic Research Laboratory (BRL) in Aberdeen, Maryland.

A more detailed version of these notes has been distributed with the agenda of the 8-9 April meeting to members of both new task forces.

Gateway Algorithms Task Force

3.2 17 January 1986

The second day of the meeting was composed primarily of technical presentations.

Eldridge gave a status report of Sparta's ongoing work for DCA. The five principle tasks are:

- Design an area routing algorithm,
- Develop gateway functional requirements,
- Describe architecture of the next generation packet switch,
- Identify improved network feedback to hosts, and
- Protocol certification support.

He then presented an *Application of Multi-Objective Optimization to Networking* by C. Eldridge. Shacham was able to provide additional references for the work.

Nagle presented his "Gateway Database Protocol", which he developed for the Multinet Survivable Internet Routing Program. In this work, he distinguishes between the routing and distributed database problems, which together make up Internet routing. He presented several interesting innovations, one of which was that his protocol runs above a reliable transport service. He distributed a paper which documented the protocol.

Roki presented the main points of his paper, *Clusters of Networks - Application to Public Data Networks (PDN)*. His proposal would allow Internet hosts with PDN connectivity to route to other PDN hosts directly (even to those on different Internet networks) without using an Internet gateway. Traffic between such Internet/PDN hosts would be preferentially routed through the PDN. Roki's scheme involves associating a set of Internet networks to a "cluster of networks" and then using a "cluster-mask", analogous to the subnet address mask scheme, for routing decisions.

Mills elaborated on two papers that he distributed since the last meeting. They were his "wiretap" routing algorithm, developed during work on the amateur packet radio network, and *A New Enhanced-Service Model for the Internet*. Mills was particularly interested in drawing parallels between his work, Roki's clustering scheme, and Su's work on gateway "affiliations".

Nagle presented his "fair queuing" scheme, in which gateways maintain separate queues for each sending host. In this way, hosts with pathological implementations can not usurp more than their fair share of the gateway's resources. This invoked spirited and interested discussion. Zhang pointed out that this was a subtle change in architecture away from a pure datagram network. Callon reminded everyone that he had written a paper advocating a connection oriented Internet Protocol several years ago.

Deering presented his work, *Host Groups: A Multicast Extension for Datagram Internetworks*. He persuasively argued in favor of multicasting and gave arguments against broadcasting schemes. He hoped that the Task Force could:

- provide some critical comments on the proposal,
- consider multicast in design of next generation protocols (e.g., routing),
- discourage proliferation of broadcast based protocols, like ARP.

Gateway Algorithms Task Force

Clark discussed the proposed ISO Host-Gateway Protocol. He was concerned with several aspects of the protocol, such as its restriction to specific network topologies. This re-opened a wider discussion on ISO issues, in which Mills again suggested that Internet gateways should switch ISO datagrams. Callon presented his suggestion for "ARPA-Internet Use of OSI NSAP Addressing". Mills suggested that this proposal be documented as a Request for Comments (RFC).

Hardcopies of slides and/or position papers are available for each of the above presentations. They are compiled with these minutes for distribution.

Gateway Algorithms Task Force

4. Addenda

4.1 Distributed Agenda

As distributed by the Chair prior to the meeting:

Thursday, 16 January

0900-0930	Welcome and admonishment
0930-1030	Old business and action items
1030-1200	Status reports Cook: Multinet Gateway Hinden and Seamonson: Butterfly Gateway Natalie and Chiappa: other gateways Mathis and Su: reconstitution demonstrations Mills: time-synchronization protocols and experiments New players: CNUCE Italy (Erina Ferro), U. Michigan (Hans-Werner Braun), NBS (Steve Ritzman) Guest players: DDN PM (Mike St. Johns), Linkabit (ESI crew)
1200-1300	Lunch
1300-1700	Joint meeting with NSF Supercomputer Gateway Committee Clark: tutorial on DoD Internet architecture Mills: tutorial on Internet gateway systems and issues

Friday, 17 January

0900-1200	Documented presentations Eldridge: gateway studies and issues (see sparta.doc) Nagle: an open architecture for routing (document to be supplied) Mills: new internet models (see newmod.doc) Clark: the ISO view on ICMP (document to be supplied) Rokitanski: cluster of networks (see roki.msg)
1200-1300	Lunch
1300-1700	Discussion Mills and Su: autonomous systems and confederations (see updated confed.doc) Nagle and Zhang: congestion-control issues and gateway design (see RFC960) Callon, Hinden and Brescia: issues on the conversion of the Internet gateway system to switch ISOgrams, especially address mappings Ritzman and Gross: issues on gateway architecture and routing standards Clark, Shacham, Cohen, and Mills: action items for future research

Gateway Algorithms Task Force

4.2 Reference Documents for this Meeting

Important files on dcn9 in /usr/ftp/pub/gads:

gads1.msg gads2.msg gads3.msg	Mailbags of messages since inception of GADS.
gadsm.msg	Minutes of previous GADS meetings.
jbn1.msg	Note on congestion-control mechanisms for gateways, by John Nagle.
roki.msg	An opus on addressing issues in public data nets, by C-H (Roki) Rokitanski.
sparta.doc	An opus on gateway issues by our Spartan friends.
egp1.msg	Exchange of messages on standards issues and EGP.
egp2.msg	Exchange of messages on other EGP issues.
rfc904.txt	Current revision of the EGP specification document. Unchanged since last posting before the last meeting.
rfc958.doc	Current revision of Mill's NTP specification document. Revised and expanded since last posting, before the last meeting as the file TIMPRO.TXT on usc-isid.arpa. Note that the other files on time-synchronization algorithms and experiments have since appeared as RFC956 and RFC957.
wirtap.doc	Current revision of Mill's document on "wiretap" algorithms, originally written for another readership, but containing an interesting multiple-path routing algorithm.
newmod.doc	Current revision of Mill's document proposing a new engineering model for the Internet, in RFC format.
confed.doc	Extensively updated revision of Mill's document on Autonomous Confederations, in RFC format.

See also:

hardcopy *Zakon, S., An architecture for routing in the ISO connectionless Internet*, ACM Computer Communications Review, October/November 1985, pages 10-39.

RFC956, RFC957, and RFC958 on time synchronization, RFC970 on gateway congestion, RFC966 on multicasting/host groups, RFC963/RFC964 on problems with the IP/TCP specs.

Gateway Algorithms Task Force

4.3 Proposed Charter of the Internet Architecture Task Force (INARC)

The mission of this task force is to explore and extend the architectures and engineering models for internet systems, in general, and the DoD Internet, in particular. The goal of the effort is to provide a sound infrastructure for new services and applications being developed by other task forces, in particular the End-to-End and Applications task forces. Primary emphasis is placed on research issues leading to near-term prototype testing and evaluation in the context of these new services and applications; however, strong emphasis is also placed on general internet research issues and in collaborating with other task forces on these issues.

The products of this task force are expected to be in the form of technical memoranda and other documents useful in the advanced planning and evaluation cycle (as well as briefings as appropriate). The task force will also serve as a source of advice and coordination on network experiments and performance evaluation, as well as to serve as an advisor on advanced planning for the operational agencies and user groups.

4.4 Proposed Charter of the Internet Engineering Task Force (IETF)

The mission of this task force is to identify and resolve engineering issues in the near-term planning and operation of the DoD Internet. The goal of the effort is to improve and expand the service for operational users, including the gateway system and various networks operated (on behalf of all users such as Arpanet and Milnet). Primary emphasis is placed on growth forecast, problem identification, and solution specification. Since solutions are expected to be effected by contractors, emphasis is also placed on advice to contractors and review of performance. Strong emphasis is also placed on near-term planning for growth in system size and improvement in performance.

The products of this task force are expected to be in the form of technical memoranda and other documents useful to the operational agencies and their contractors. It is expected that much of the agenda of this task force will be created by these agencies and the users. However, this task force is not intended as a forum for discussion of policy issues on administration or procurement.

APPENDIX A

Hardcopy of GADS Presentation Slides

<i>Author</i>	<i>Title</i>
C. Eldridge	Application of Multi-Objective Optimization to Networking
J. Nagle	A New Internet Routing Protocol
C. H. Rokitansky	Cluster of Networks
D. Mills	The Wiretap Algorithm
D. Mills	Network Time Protocol (NTP)
J. Nagle	Congestion in the Internet Doing Something About It
D. Cheriton, S. Deering	Host Groups: A Multicast Extension for Datagram Internetworks
R. Callon	Arpa-Internet Use of OSI NSAP Addressing
B. Hinden	Type of Service Routing (not presented at meeting)

Application of Multi-Objective Optimization to Networking

Motivations

A new theory emerging from classical Operations Research approaches

Hope to illuminate problems, find solutions in (inter)networking

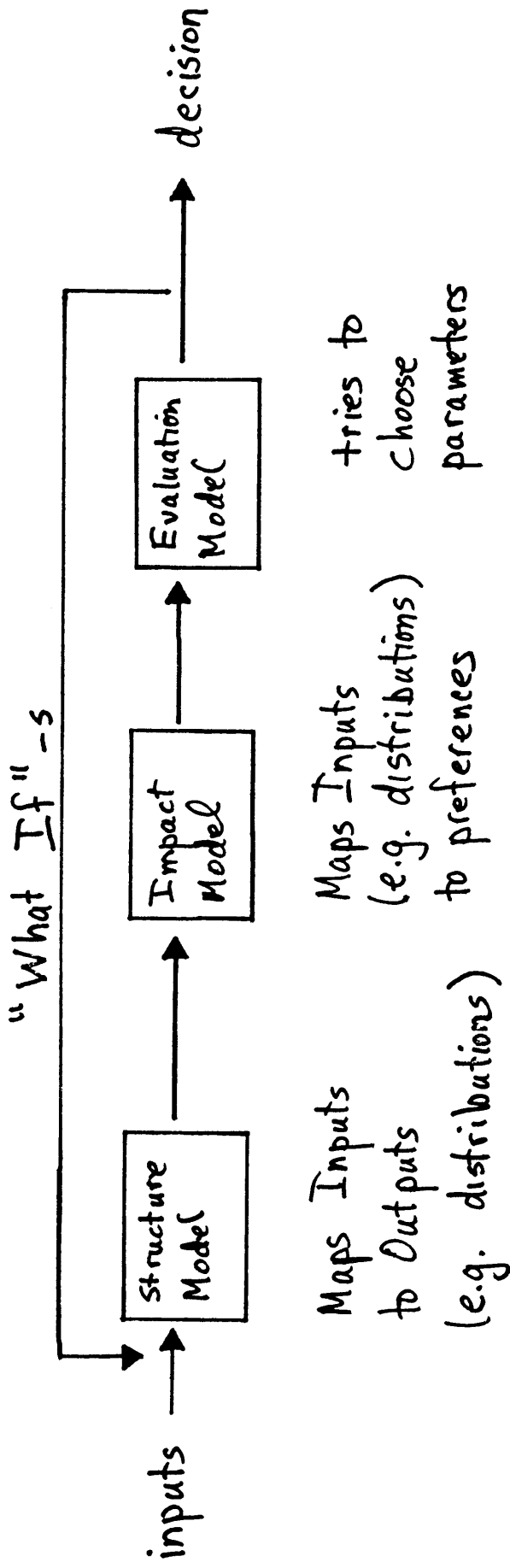
Conclusions

New theory has developed a framework, but

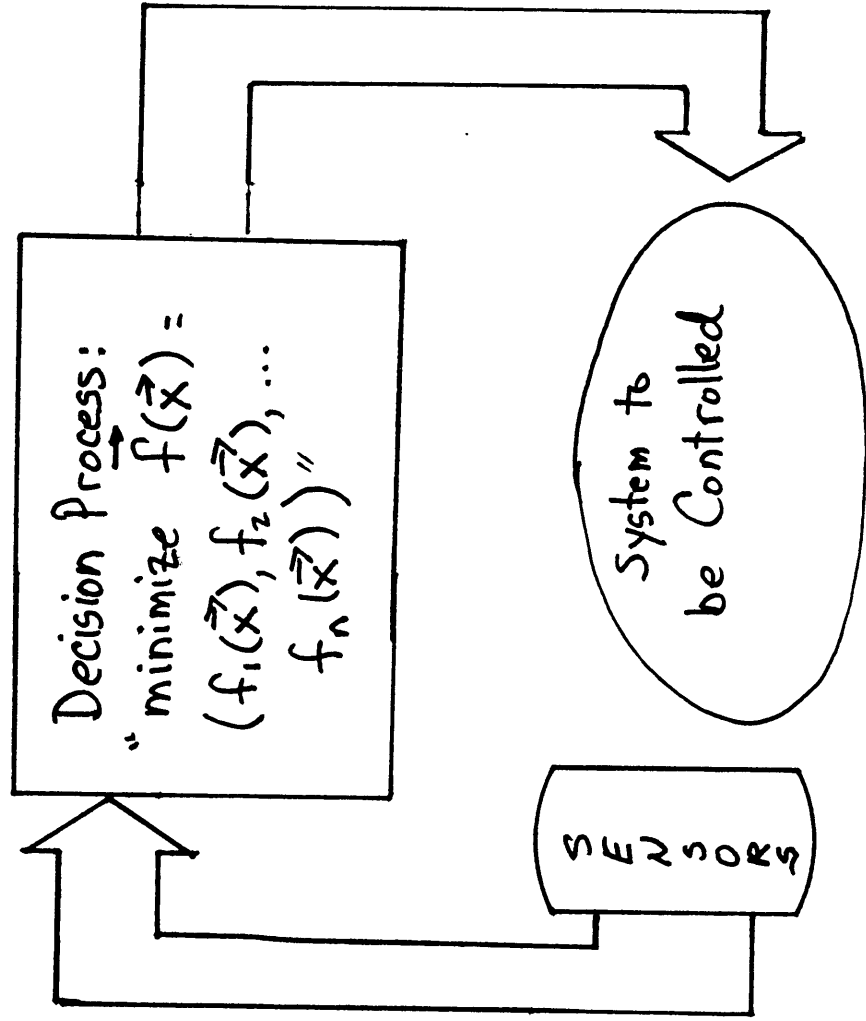
We'll still explore via implementations and simulations.

Reference: Y. Sawargi, H. Nakayama and T. Tanino, Theory of Multiobjective Optimization (Academic Press). Mathematics in Science and Engineering, Vol. 176.

Model of a Decision Making Process



Second View of Decision Process



$\{f_i(\vec{x})\}$

Objective Functions

Parameters:
typically $\vec{x} \in \mathbb{R}^n$;
may be subject
to constraints, e.g.,
 $\{g_i(\vec{x}) \leq 0\}$

(Subject to Stochastic Behaviors, Observation Errors)

(Inter network) Correlates

Controllable Parameters

Executable Code

Decision Parameters,
e.g. thresholds,

precedence assignments

Decision Algorithms,
e.g. for routing

Constraints

.....
System Limits due to
finiteness

Uncontrollable Variables

.....
Traffic Demands
Failures

Objective Functions

.....
Delay, Throughput, Queue
Sizes, Availability

What Does MOO Theory Tell Us?

1. $\vec{f}(\vec{x})$ induces/needs "domination structure"
via preference order " $>$ "

define set of efficient elements $\vec{f} \in F$:

$$E(F, >) = \{ \hat{\vec{f}} : \nexists \vec{f} \in F : \hat{\vec{f}} > \vec{f} \}$$

Corresponding set for \vec{x} : $\{ \hat{\vec{x}} \in X : \nexists \vec{x} \in X : \hat{\vec{f}}(\vec{x}) > \vec{f}(\vec{x}) \}$

can have $\{ \vec{x} : \vec{f}(\vec{x}) = \vec{f}_0 \}$ point-to-set maps

2. Existence : efficient element \vec{f} exists
if domination structure never cyclic

MOO Theory Lessons

Stability: requires that each point not in $E(F, >)$ be less preferable to some point in $E(F, >)$ -- that is "covered" by the preference relation

Connectedness: If $>$ has linear properties, then $E(F, >)$ is connected

MCO Theory Lessons

Stability: will the system behave if control parameters are perturbed?

- require continuity of point-to-set maps -- defined in MCO theory via existence and inclusions of limits

review!

- results are given for parameterization of both \vec{x} , γ !

this imposes a distance metric over all parameters + symbols

Impact Model

Structural Model does/need not produce deterministic results; instead we obtain parameters of distributions.

Particularly true in internetworking, where structural model is queueing system.

Decision-maker must choose among risky alternatives: HOW? Via a suitable utility theory.

Example: Lottery A = [3000:1.00],
Lottery B=[4000,0:0.80, 0.20]; Most prefer Lottery A. Yet, if Lottery C = [3000,0:0.25, 0.75] while Lottery D = [4000,0:0.20,0.80], most prefer D.

Impact Model

von Neumann–Morgenstern utility theory
is classical starting point; is based
on expected value.

other factors enter in, especially risk
aversion;

Internetworking correlates include
probability distributions of delays,
throughput, frequency of packet loss;

Internetworking's Impact Model is
Application–Dependent

Evaluation Model

Clarification: Task is to find values of parameters, not undertake a judgement.

Assume we have a comprehensive preference basis.

In numerical spaces, we search along gradients, apply dynamic programming and other techniques, thanks to distance measures.

In symbolic space we search for "good" parameter combinations, but we need "heuristics"; suggests "AI" approaches.

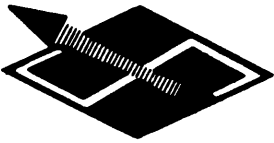
SO WHAT?

Internetworking's "structure model" is very complex; interdependencies in time and space abound; comparable to macroeconomic models;

Models (of the Internet and other systems) usually oversimplify anyway; gain from trying to apply MOO theory is uncertain;

In particular, optimization techniques depend heavily upon parameterization into Euclidean space, rendering controls into "knobs" and "dials."

Internetworking likely to continue as empirical science: design, build, simulate, experiment, analyze, uncover principles.



SPARTA'S WORK FOR DCA:

1. DESIGN AN AREA ROUTING ALGORITHM
2. DEVELOP GATEWAY FUNCTIONAL REQUIREMENTS
3. DESCRIBE ARCHITECTURE OF NEXT GENERATION PACKET SWITCH
4. IDENTIFY IMPROVED NETWORK FEEDBACK TO HOST
5. PROTOCOL CERTIFICATION SUPPORT

DESIGN AN "AREA" ROUTING ALGORITHM

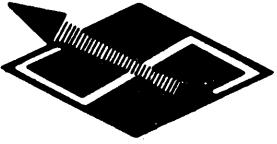
MOTIVATION:

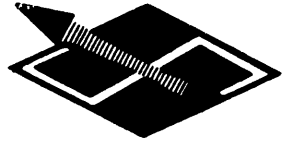
SOLVE SOME CONTROL FLOW PROBLEMS
ASSOCIATED WITH LARGE, FLAT STRUCTURE

- VOLUME INCREASE WITH N^2
- PATH LENGTH DELAYS

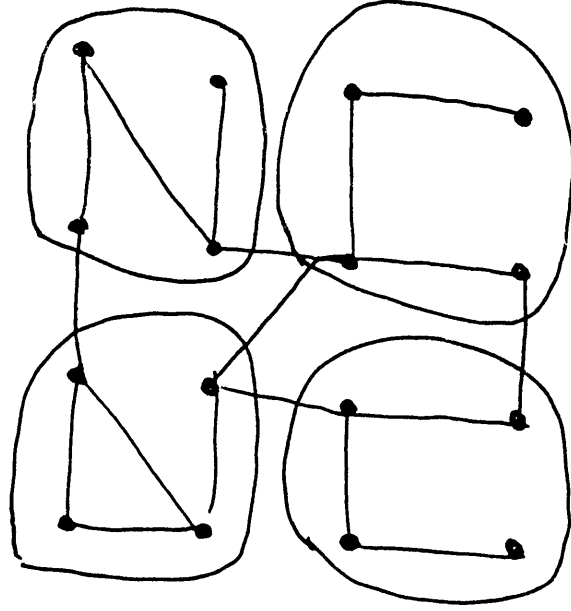
STATUS:

- BACKGROUND STUDIES
- HAND CALCULATIONS

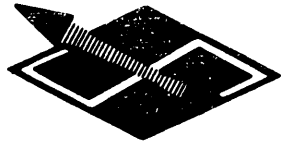




AREA ROUTING CONCEPT



- RETAIN RICH LINK TOPOLOGY
- REGARD AREA AS "MULTIPOINT" DEVICE
- SOLVE ROUTING AT TWO LEVELS:
 - WITHIN AREA
 - AMONG AREAS
- PROTOCOLS
- ADDRESS ORGANIZATION PRINCIPLES, ESPECIALLY FOR AREA DEFINITION



EVALUATION OF AREA ROUTING

- CALCULATE VOLUME OF CONTROL TRAFFIC,
COMPARE TO FLAT ROUTING
- SIMULATE LOADING OF NETWORK VIA ROUTING
ALGORITHMS ; COMPARE "AREA" TO "FLAT"
- ASSESS STRENGTHS, WEAKNESSES w/ RESPECT TO
SURVIVABILITY, ROBUSTNESS, MANAGEABILITY

A New Internet Routing Protocol

John Nagle

Ford Aerospace
and Communications Corporation

EGP has got to go

- Nobody likes EGP, it's just been available.
- It was never intended as a real routing protocol.

GGP has reached its limits

- We're nearing table size limits now.
- GGP generates N^2 traffic, at non-trivial levels.
- Any "core" gateway can kill the GGP system, and not all "core" gateways are in secured facilities. And they can't be, or the ARPANET won't work.

Survivable Internet Routing Program

- "If there's a way to get there, find it and use it."
- May route into and through other nets and internets.
- Must be robust in face of disruption, accidental or deliberate.

Gateway Database Protocol

- Designed for SIRP program, but of more general utility.
- Still in preliminary form, offered here for comments.
- A candidate as an EGP and GGP replacement.

Basic features of GDP

- An open architecture for passing around routing data.
- Everybody gets a full map of the net.
- Robust in face of bad data.
- Fully event-driven.
- Allows for mutual mistrust.
- Some nodes may trust certain nodes more than others.
- Allows for multiple routing algorithms in the same internet.
- Allows for multiple protocols in the same internet.

Architecture

- Every node has a few neighbors that it talks to on a continuing basis, just like EGP, GGP, etc.
- Nodes establish transport connections to peers to exchange routing data.
- GDP thus requires a transport protocol underneath. This gets checksums, sequencing, 3-way handshakes, timers, acknowledges, etc. out of the routing protocol. Simplifies the whole thing enormously.
- Any transport protocol will do, but TCP is recommended in IP nets and TP4 in ISO nets.
- The protocol basically defines a way of synchronizing a replicated distributed database, independent of the contents of the database.

The database

- The database consists of items of the form (owning node, attribute, value). Every database item is owned by a specific node and only that node can change its value.
- When an item changes, the new value is distributed throughout the network, by a new variant on flooding.
- Database items have been defined for routing data. Others can be added later.

Database synchronization

- This is a brief summary; see the protocol spec for the exact rules.
- The basic idea is that updates are propagated by flooding. But the mechanism has been designed to survive bad updates, phony updates, and too many updates.

Robustness mechanisms

- Bad updates about your own node's state will be accepted. But no link is up unless both ends say it is, so you can't claim links you don't have to divert traffic to you.
- Sending out bad updates about nodes that are up will cause trouble. But eventually the phony update reaches the real owner, which denies it with an update of its own. This will correct any transient error.
- Sending out bad updates about nodes that are unreachable is harmless; the data is not used for routing and any bad data will be corrected when the node becomes reachable.

Extra robustness for critical nets

- A firewall mechanism is provided, using a concept called "administrative distance", to allow sections of the network to avoid even temporary corruption of their internal routing data. This replaces the old "core network" concept with a more powerful mechanism, one which allows proper MILNET/ARPANET isolation.
- Sending out bad updates repeatedly at a considerable rate will cause trouble only if the source of the bad update is nearby in the administrative distance sense. If it is nearby in this sense, (which normally means under the same administration), there is serious trouble. But alarms will go off; the real owner node will notice that something very bad is happening and will try to tell network control. Network control can then cut the offending node out of the net. The network will then restabilize and purge itself of the bad routing data.

Economy of routing traffic

- The protocol is fully event-driven, except for a keep-alive probe. The robustness mechanisms make this safe. (We use the keep-alive probe to validate the databases, just on general principles of not trusting anything).
- We don't forget about unreachable nodes unless we need the table space or they are unreachable for a long time. Thus, we only have to flood the net with the brief note that a link is up when a whole network becomes reachable after a short outage.
- This mechanism is powerful enough that we have calculated that a 1000-node network over 9600 baud lines, with one line outage per link per five minutes, will only use about 20% of the net bandwidth for routing information.

Conclusion

- We have a new approach. So far it looks good. Please take the protocol spec home, read it, and find its weaknesses.
- How about an implementation on top of 4.3BSD for starters?

'Cluster of Networks'

C. H. Rokitansky

DFVLR Oberpfaffenhofen

Jan 1986

Cluster of Networks - Concept:

- Several INTERNET networks form a 'cluster of net'
- Use of a 'cluster-mask'
- Application to Public Data Networks (PDN)

Wide Area Networks (WANs):

- internal structure of the WAN is of interest even outside it.
- direct connections between hosts on a WAN are possible

Demands for 'Wide Area Networks'

- Subdivision (if any) of a WAN (several "entry gateways") should be taken into account in external routing decisions.
- Internal routing decisions: all hosts on a WAN should appear to be reachable "locally" (directly)

Proposed Solution:

- Assignment of different Internet network numbers to subdivisions of a WAN
- WAN \rightarrow "Cluster of Networks"
- Use of a "cluster-mask" for the specification of the "cluster" and for internal routing decisions

INTERNET - Address:

$\langle \text{INTERNET - address} \rangle ::= \langle \text{network-number} \rangle \langle \text{rest field} \rangle$

$\langle \text{network-number} \rangle ::= \langle \text{cluster-number} \rangle \langle \text{cluster-net-n} \rangle$

Cluster - Mask:

255. 0. 0. 0.
11111111 00000000 00000000 00000000

$\langle \text{cluster-number} \rangle \langle \text{cluster-net-number} \rangle \langle \text{rest field} \rangle$

$\langle \text{network-number} \rangle \langle \text{rest field} \rangle$

- ICMP Address Mask Request
- ICMP Address Mask Reply

Advantages:

- The internal structure of a "cluster" (several INTERNET networks) is visible outside the cluster. (Important for exterior routing)
- The fact that a "cluster of networks" has been formed is invisible outside the cluster. (→ No exterior changes)
- All hosts (gateways) within the same cluster appear to be reachable directly ("locally") (Important for interior routing)
- No (or only minor) changes to host software that supports subnets
- ICMP Address Mask Request and -Reply
- ICMP Redirect messages can be used between gateways and hosts on different INTERNET networks, but in the same cluster.

Disadvantage:

- Specific INTERNET network numbers must be reserved for "clusters of networks"

However: Out of a maximum number of

126	class A networks:	17	network numbers	(13 %)
16. 382	B	77	"	(0.5 %)
2. 097. 150	C	3.551	"	(0.2 %)

Public Data Networks (PDN) - Characteristics:

- Wide Area Network
- Complex of national public data networks
- International virtual circuits
- Different costs for international and national virtual circuits
- Costs depend on data volume and length of time of connection
- no broadcasting

Routing through PDN :

- Routing decisions should be done with regard to the structure of the PDN (subdivision into different national networks)
- Routing via an "international" VAN gateway only if routing via a "national" VAN gateway is impossible
- Routing between PDN hosts through PDN

Proposed Solution:

- INTERNET class B network numbers (with identical bits in the first (high-order) 8-bit field of the INTERNET address) are assigned to national public data networks.
- The national public data networks are assembled to form a cluster of networks ("PDN-Cluster")
- Use of a "Cluster-mark", thus all hosts within the "PDN-Cluster" appear to be reachable "locally"
- If necessary, VAN gateways are exchanging (modified) EGP messages on an "event driven" basis (i.e. No periodic updates (!))
- Mapping between the INTERNET address and X.121 address of PDN hosts is done by an "X.121 Address Server/Resolution Protocol"

PDN-Cluster:

$\langle \text{network-number} \rangle ::= \langle \text{cluster-number} \rangle \langle \text{cluster-net-number} \rangle$

16 bits 8 bits 8 bits

256 cluster-nets

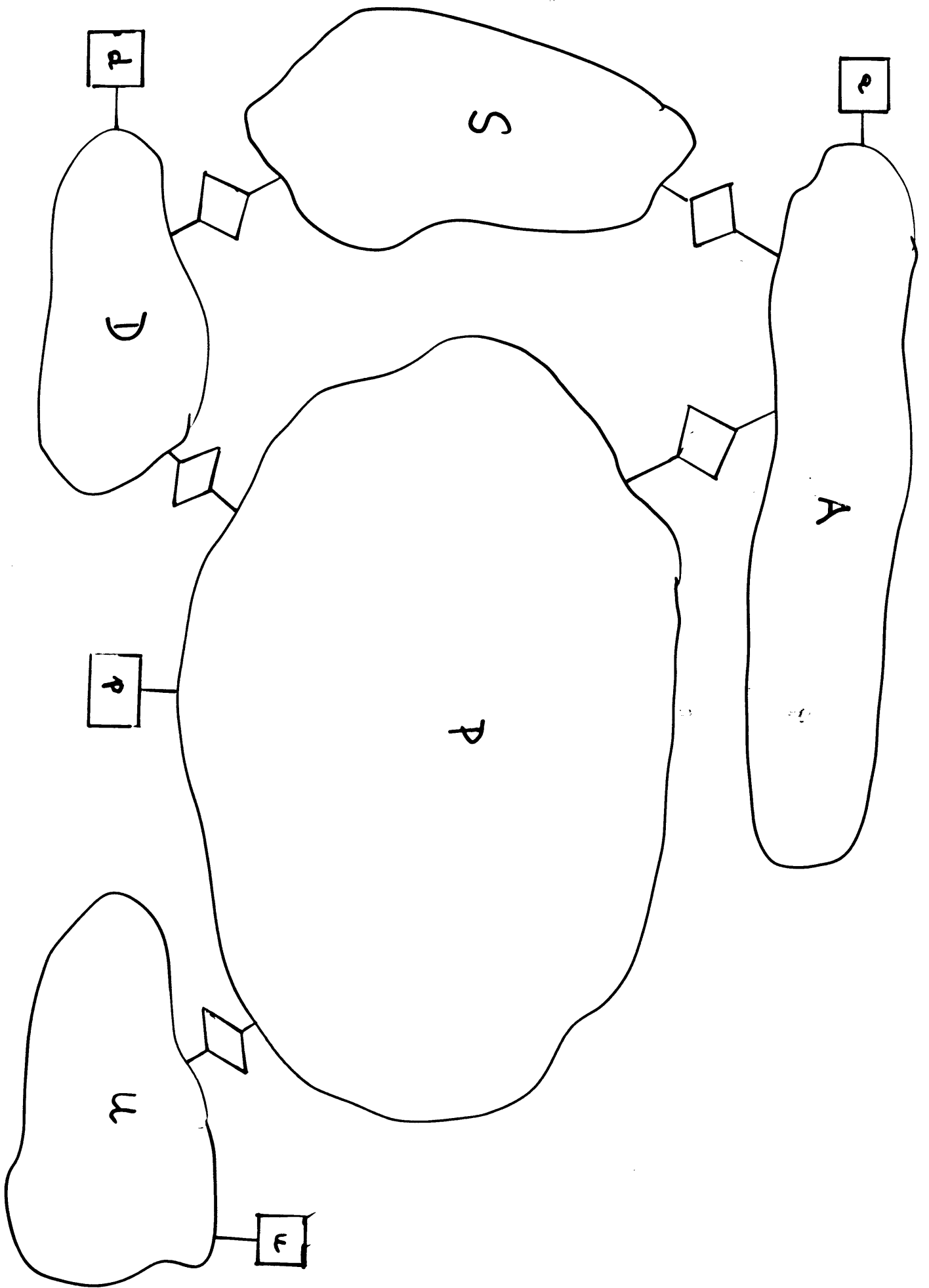
Organization of clusters and the reservation of INTERNET network numbers could start with the highest, not yet assigned network numbers of each class.

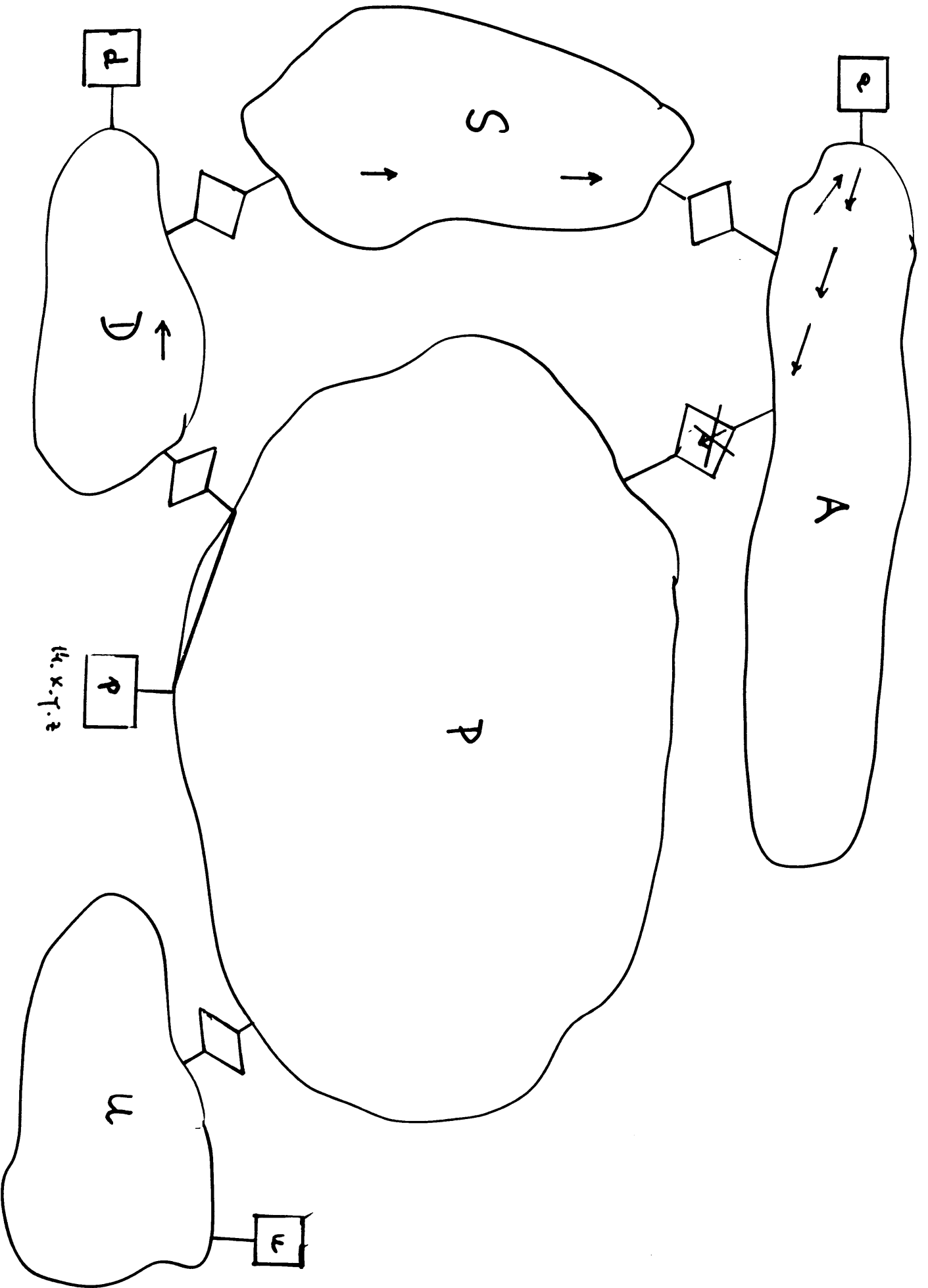
For the „PDN-Cluster“ the cluster-number

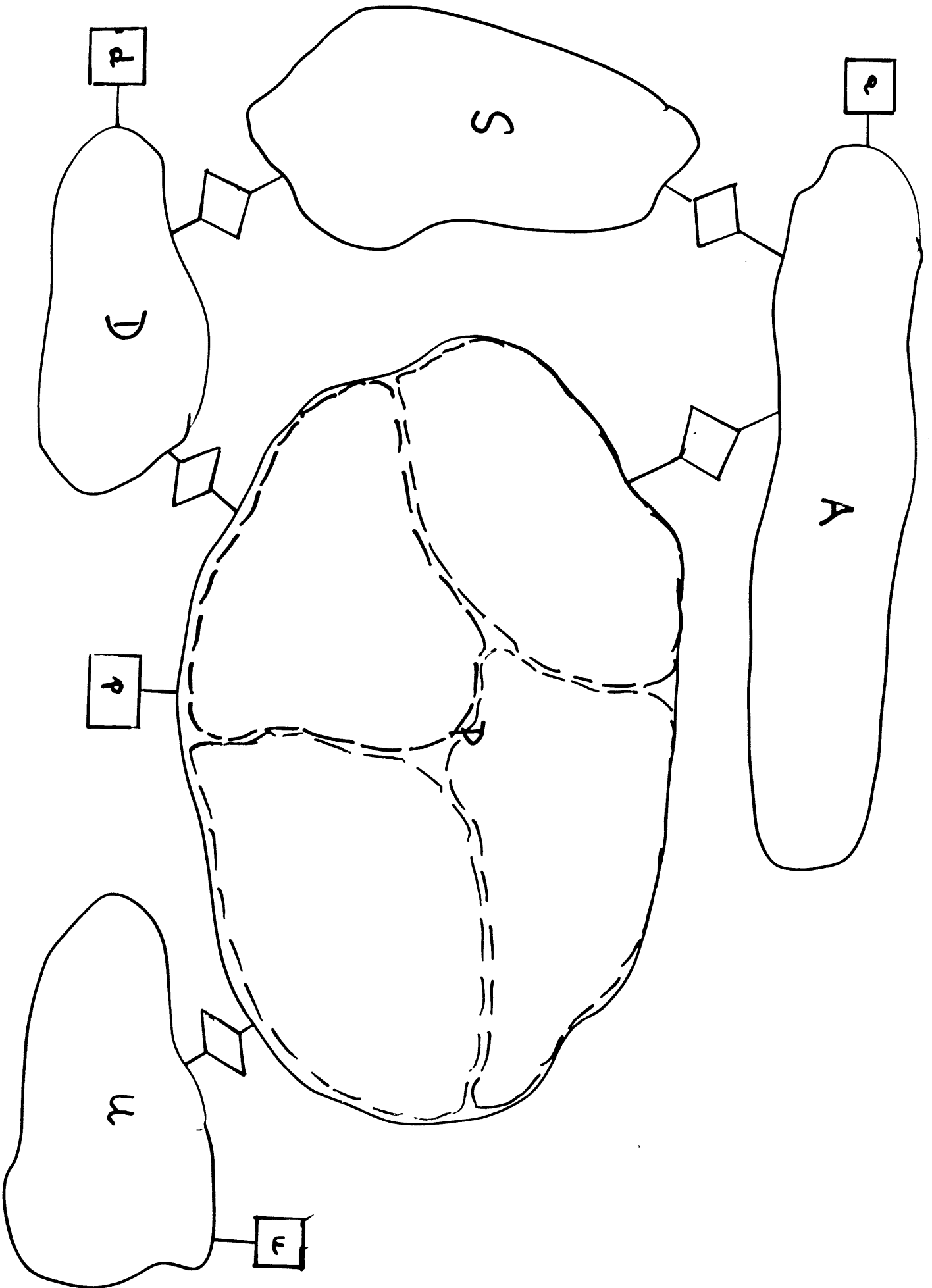
[191. nnn. rrr. rrr] is proposed

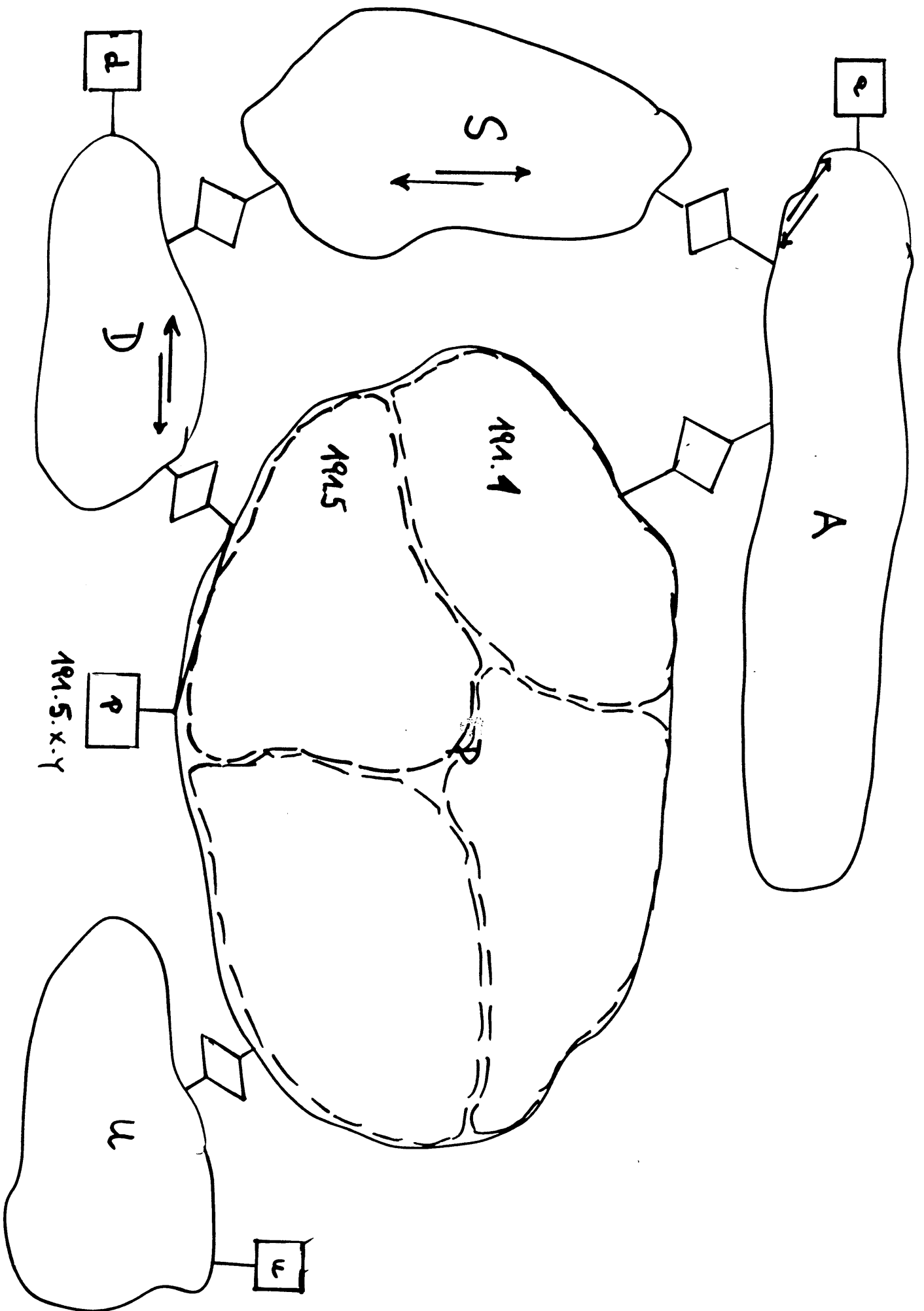
PDN-cluster-networks:

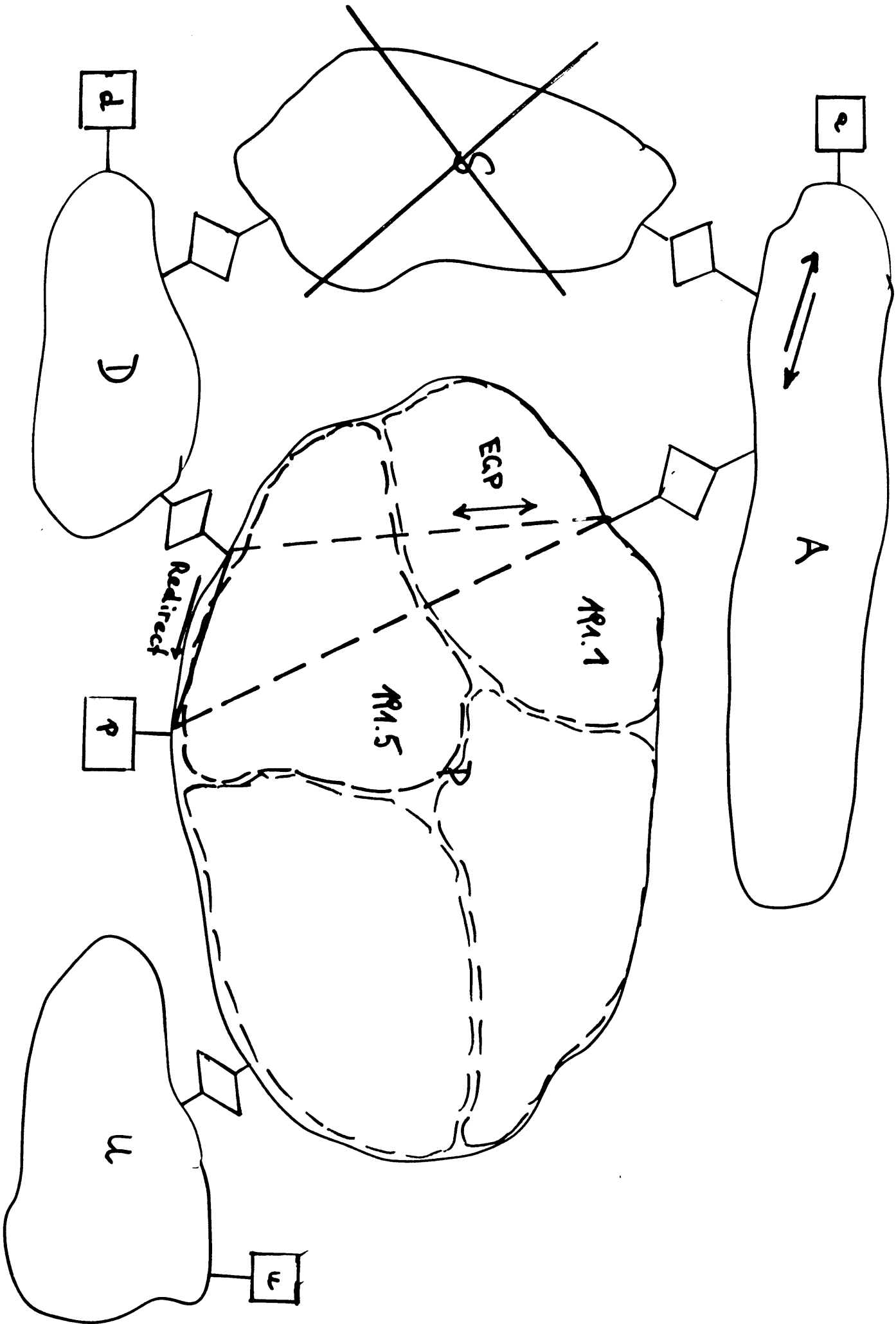
<u>DNIC</u>	<u>Public Data Network</u>	<u>INTERNET network num</u>
3110	TELENET (USA)	191. 1
2342	IPSS (U.U.)	191. 2
2405	TELEPAK (Sweden)	191. 3
2041	DATANET (Netherlands)	191. 4
2624	DATEx-? (West Germany)	191. 5











Summary

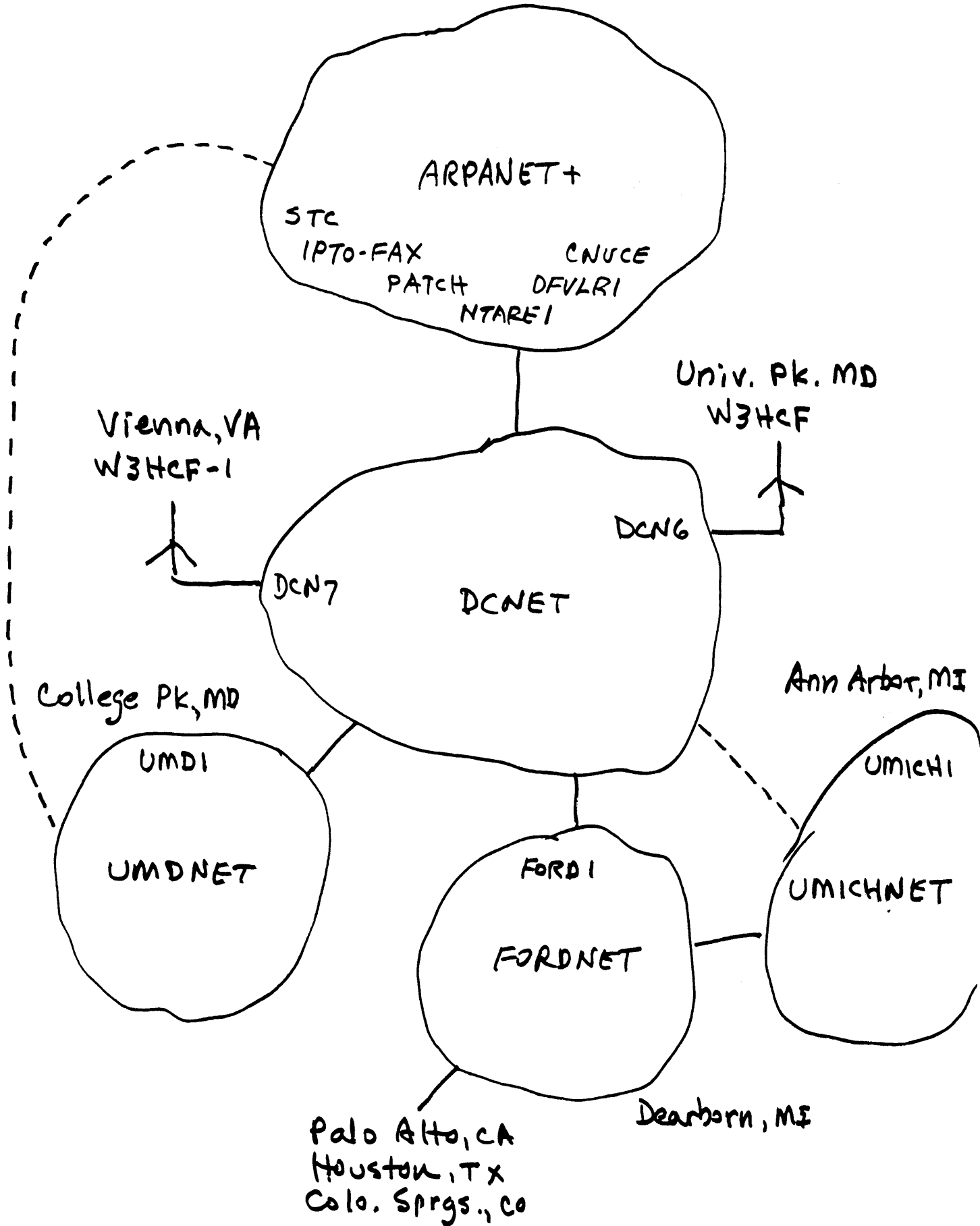
Advantages of the clustering scheme

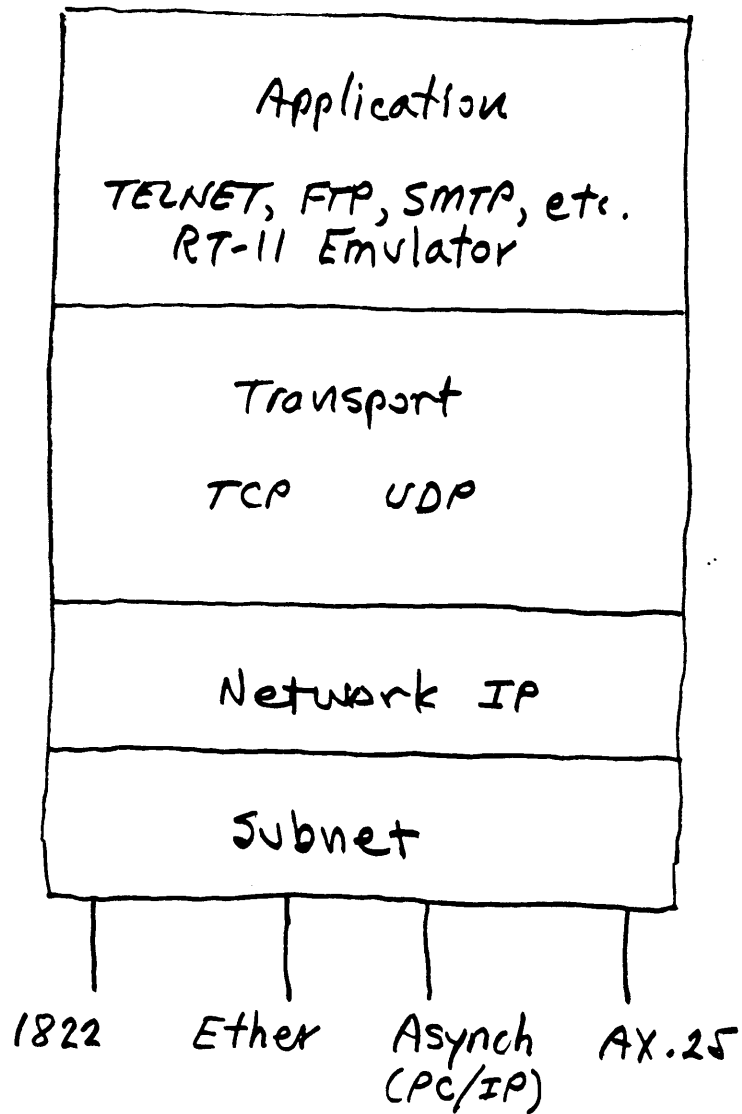
- internal structure of a Wide Area Network is visible even outside of it
- all hosts within the same cluster appear to be reachable locally (directly) if a cluster-mask is used for internal routing decisions
- ICMP Redirect messages can be sent between hosts on the same cluster
- ICMP Address Mask Request - Reply

- Runs in real time
- Uses "wiretapped" monitor info
- Computes all possible routes
- Evaluates routes by several criteria
- Can route around congestion and failed nodes
- Suitable for virtual-circuit or datagrams
- Compatible with existing procedures

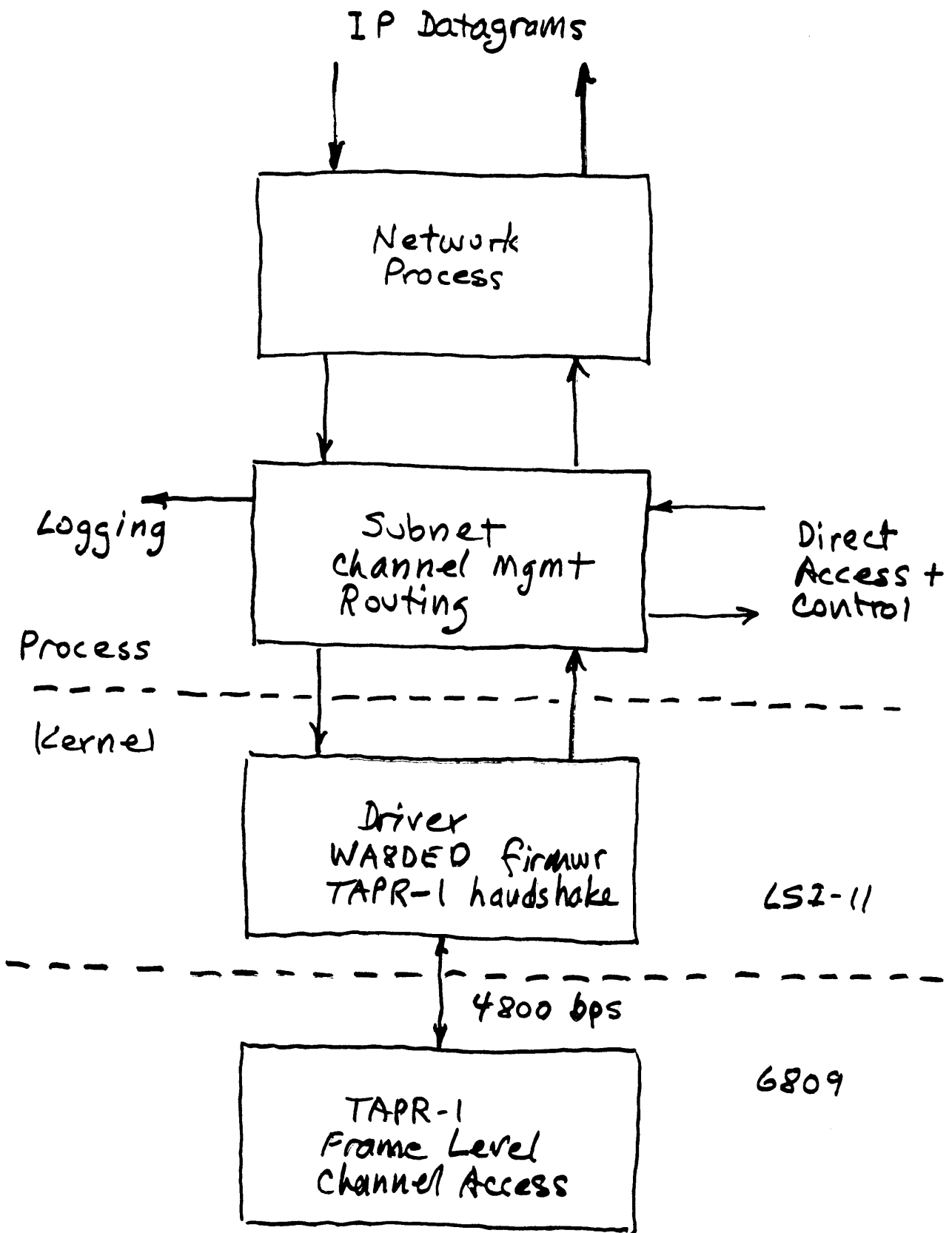
The Wiretap Algorithm

FUZZYLAND





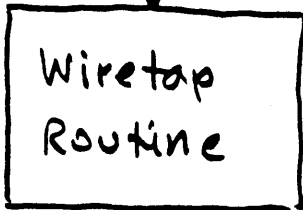
Fuzzball Architecture



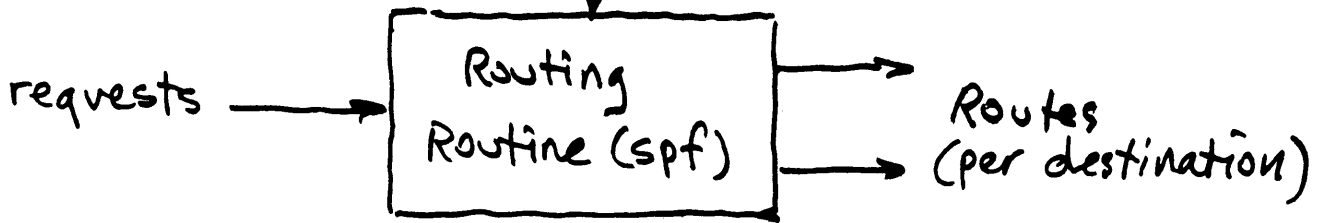
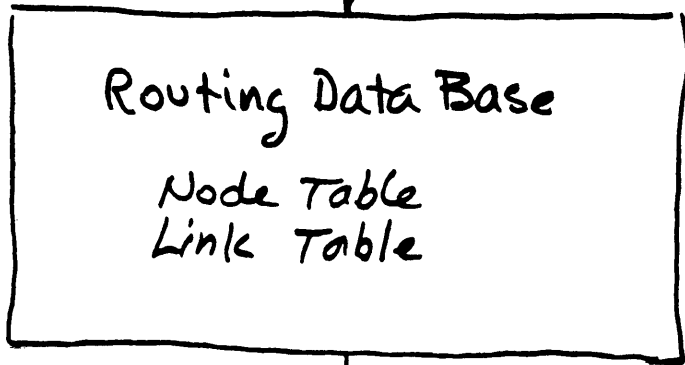
Functional Organization

Received Monitor Headers

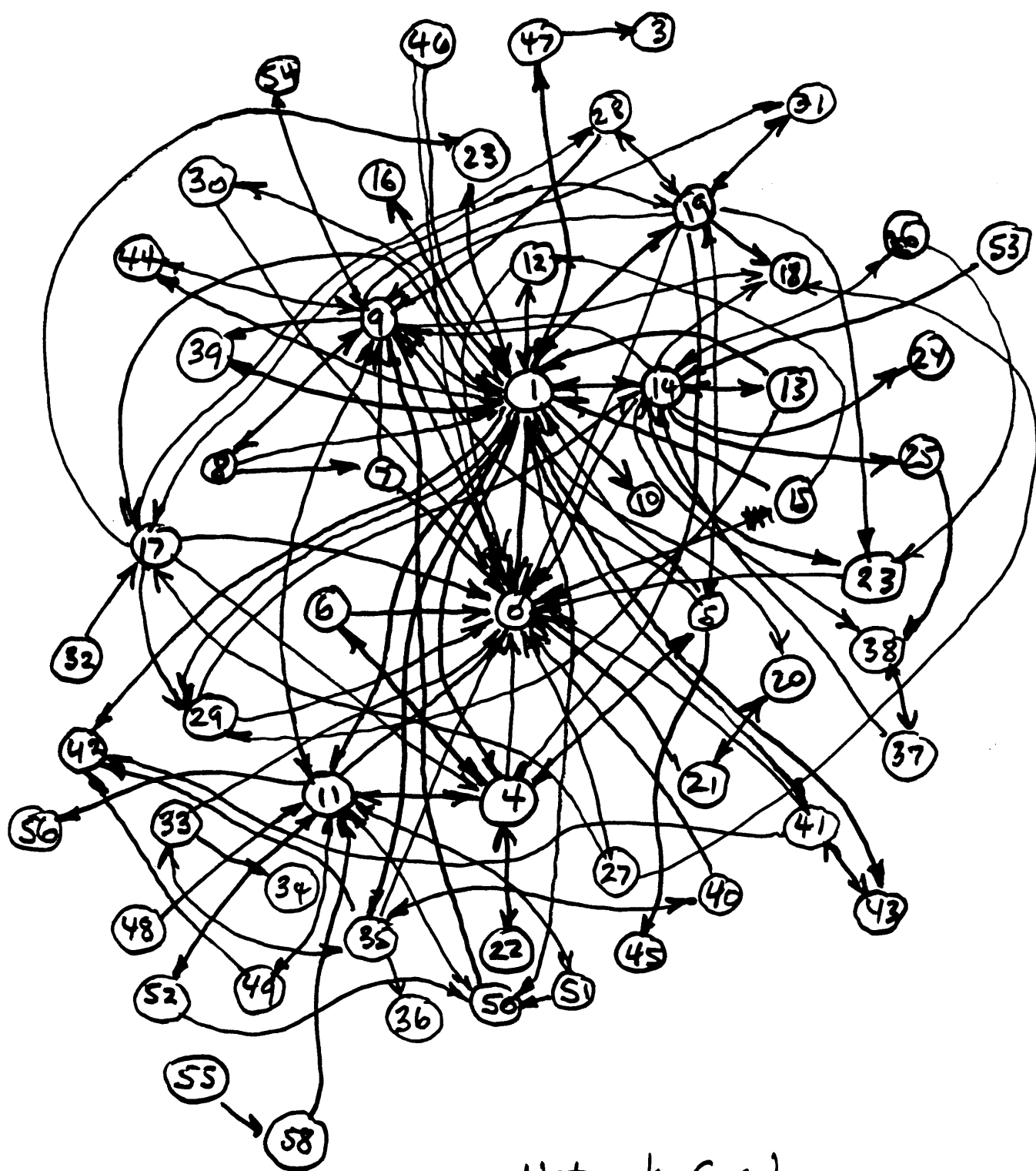
"fm W3HCF to W4HCP via WB4JF1-5
..."



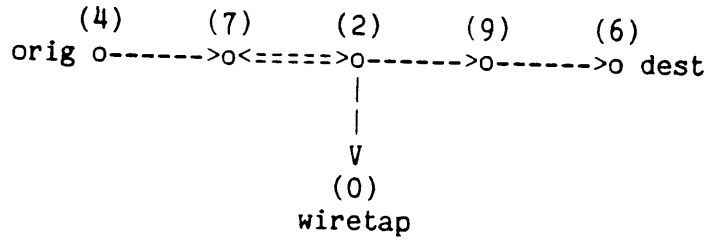
Updates



Routing Functions



Network Graph

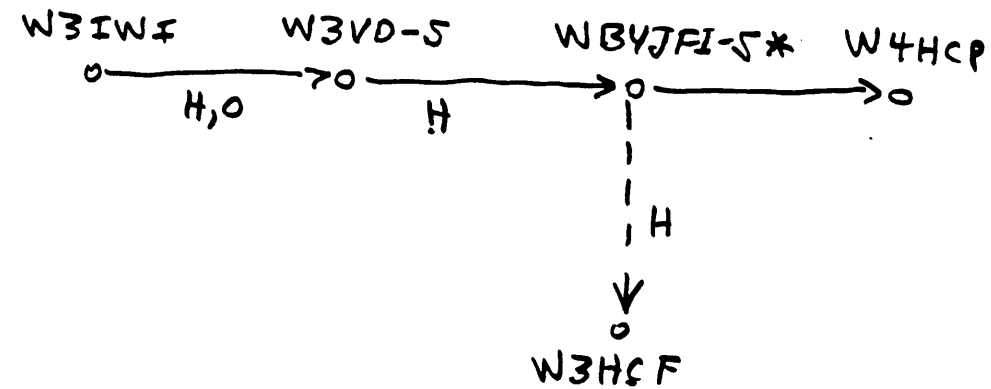
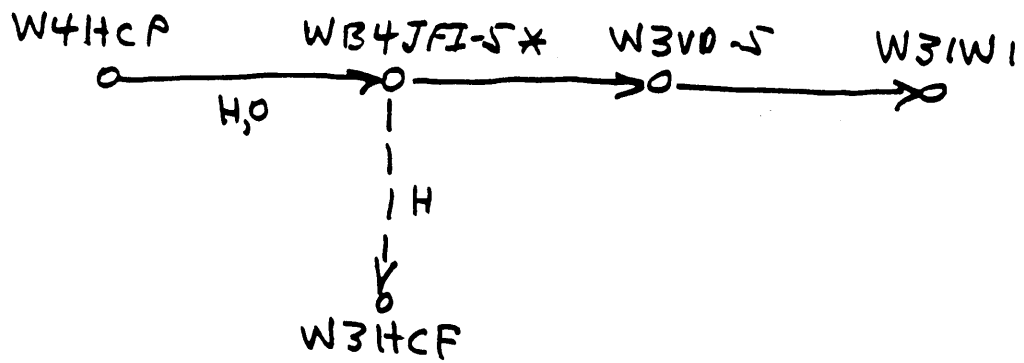


Factor	Weight	Name	How Determined
f0	30	hop	1 for each link
f1	15	unverified	1 if not heard either direction
f2	5	non-reciprocal	1 if not heard both directions
f3	5	unsynchronized	1 if no I or S frame heard

Table 1. Link Factors

Factor	Weight	Name	How Determined
f4	5	complexity	1 for each incident link
f5	5	congestion	(see text)

Table 2. Node Factors



W4HCP → WB4JFI-5 H₂O

WB4JFI-5 → W3VD-5 H, R

W3VD-5 → W3IWI H, O

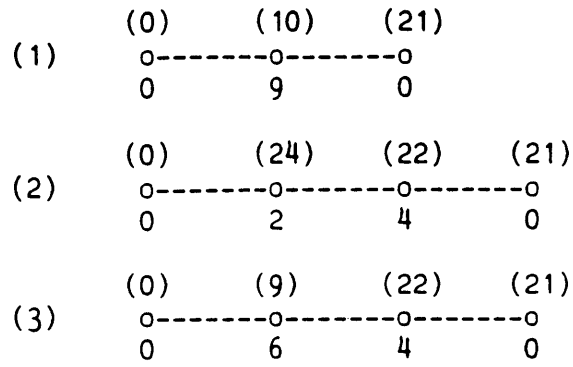
WB4JFI-5 → W3HCF H

NID	Callsign	IP Address	Flags	Links	Last Rec	Wgt	Route
0	W3HCF	[128.4.1.1]	000	14	00:00:00	0	1
1	WB4JFI-5	[128.4.1.2]	006	15	16:37:56	40	
2	W4HCP	[128.4.1.3]	000	0	00:00:00	255	
3	WD5DBC	[128.4.1.4]	000	0	00:00:00	255	
4	DPTRID	[0.0.0.0]	000	1	00:00:00	155	5 1
5	K4KMC	[0.0.0.0]	007	0	14:46:39	40	
6	WD4BAV	[0.0.0.0]	000	1	00:00:00	115	5 7
7	K4ARO-1	[0.0.0.0]	006	1	14:46:39	75	5
8	WB2RVX	[0.0.0.0]	007	3	16:25:42	85	18
9	W3IWI	[0.0.0.0]	007	6	16:37:44	40	
10	WB4APR-6	[0.0.0.0]	007	9	16:25:45	40	
11	KB5ZU	[0.0.0.0]	000	1	00:00:00	170	1
12	WB6RQN	[0.0.0.0]	003	0	16:33:17	40	
13	BEACON	[0.0.0.0]	000	3	00:00:00	80	16
14	KA4USE-1	[0.0.0.0]	007	8	15:57:59	40	
15	MAIL	[0.0.0.0]	000	1	00:00:00	125	10
16	WA4TSC	[0.0.0.0]	003	0	15:21:45	40	
17	CQ	[0.0.0.0]	000	1	00:00:00	80	5
18	KS3Q	[0.0.0.0]	007	2	16:25:47	40	
19	WB2MNF	[0.0.0.0]	006	2	15:05:05	120	10
20	KC2TN	[0.0.0.0]	007	3	15:05:05	85	18
21	AK3P	[0.0.0.0]	007	1	14:00:07	130	24 22
22	AK3P-5	[0.0.0.0]	006	4	14:00:07	80	24
23	KC3BN	[0.0.0.0]	007	2	05:42:41	80	24
24	WA3KYG-6	[0.0.0.0]	007	2	05:42:41	40	
25	KA4USE	[0.0.0.0]	003	0	15:57:57	115	14
26	TEST	[0.0.0.0]	000	1	00:00:00	110	9
27	K4NGC	[0.0.0.0]	007	0	15:14:51	40	
28	KA3KIW	[0.0.0.0]	007	1	11:39:26	85	29
29	KA3DBK	[0.0.0.0]	007	2	16:21:08	40	
30	K3SLV	[0.0.0.0]	007	1	13:17:19	40	
31	W3HCE	[0.0.0.0]	000	3	00:00:00	80	30
32	W3VH	[0.0.0.0]	007	0	12:49:21	40	
33	KE4TZ	[0.0.0.0]	003	1	13:11:27	90	29
34	WA4QNO	[0.0.0.0]	000	1	00:00:00	165	5 7 35
35	K4UMI-5	[0.0.0.0]	002	1	14:43:26	120	5 7
36	WB4FJI-5	[0.0.0.0]	002	1	14:45:41	80	27
37	WA4SZK	[0.0.0.0]	000	1	00:00:00	210	5 7 38 39
38	K4LKQ-1	[0.0.0.0]	002	1	14:46:39	120	5 7
39	W4ULH-1	[0.0.0.0]	002	1	14:46:39	165	5 7 38
40	WB4FQR-4	[0.0.0.0]	006	1	15:05:25	75	27
41	N4SN	[0.0.0.0]	007	0	15:47:25	145	1
42	KX3C	[0.0.0.0]	002	2	16:21:08	40	

Figure 1. Candidate Node Table

From	To	Flags	Age	From	To	Flags	Age
1	0	002	3	1	4	002	3
5	0	002	104	5	7	007	104
7	6	006	255	10	0	002	19
8	10	207	15	10	9	207	43
9	1	207	4	1	11	006	41
12	1	003	8	1	14	206	8
14	13	003	8	14	0	002	40
1	10	002	4	10	15	002	4
10	13	002	57	12	0	002	237
16	0	002	72	16	13	003	72
5	17	003	255	18	0	002	15
18	10	207	15	10	20	006	87
20	19	207	87	18	9	003	255
19	10	006	87	21	22	207	146
22	10	206	146	10	21	004	255
24	0	002	255	23	22	007	255
22	24	206	255	24	23	207	255
23	9	006	255	9	22	006	146
25	14	203	40	18	1	003	15
9	26	002	255	9	8	006	43
27	1	207	78	27	0	002	79
29	0	002	19	28	29	007	255
29	1	207	62	1	28	006	255
30	0	002	185	30	31	007	185
32	0	002	211	32	1	207	211
29	18	207	72	33	29	003	191
29	14	202	191	14	33	002	196
18	20	203	157	18	8	203	158
9	0	002	152	5	10	003	109
10	31	002	109	5	1	003	109
5	31	003	108	5	30	003	108
7	35	002	107	35	34	002	107
27	36	003	104	36	9	002	104
27	14	207	81	14	9	006	40
7	38	002	104	38	39	002	104
39	37	002	104	27	40	007	87
40	1	206	83	41	1	207	49
29	42	207	19	42	0	002	19

Figure 2. Candidate Link Table



From	To	f0	f1	f2	f3	f4	Incr	Total
22	21	30	0	0	0	0	30	30
10	21	30	15	5	0	0	50	50
10	22	30	0	0	0	20	50	80
23	22	30	0	5	0	20	55	85
24	22	30	0	0	0	20	50	80
9	22	30	0	5	0	20	55	85
0	10	30	0	5	5	45	85	135
8	10	30	0	0	0	45	75	125
9	10	30	0	0	0	45	75	125
1	10	30	0	5	5	45	85	135
15	10	30	0	5	5	45	85	135
13	10	30	0	5	5	45	85	135
18	10	30	0	0	0	45	75	125
20	10	30	0	5	0	45	80	130
19	10	30	0	5	0	45	80	130
5	10	30	0	5	5	45	85	135
31	10	30	0	5	5	45	85	135
9	23	30	0	5	0	10	45	110
24	23	30	0	0	0	10	40	95
0	24	30	0	5	5	10	50	130
1	9	30	0	0	0	30	60	145
18	9	30	0	5	5	30	70	155
26	9	30	0	5	5	30	70	155
8	9	30	0	0	5	30	65	150
0	9	30	0	5	5	30	70	155
36	9	30	0	5	5	30	70	155
14	9	30	0	0	5	30	70	155

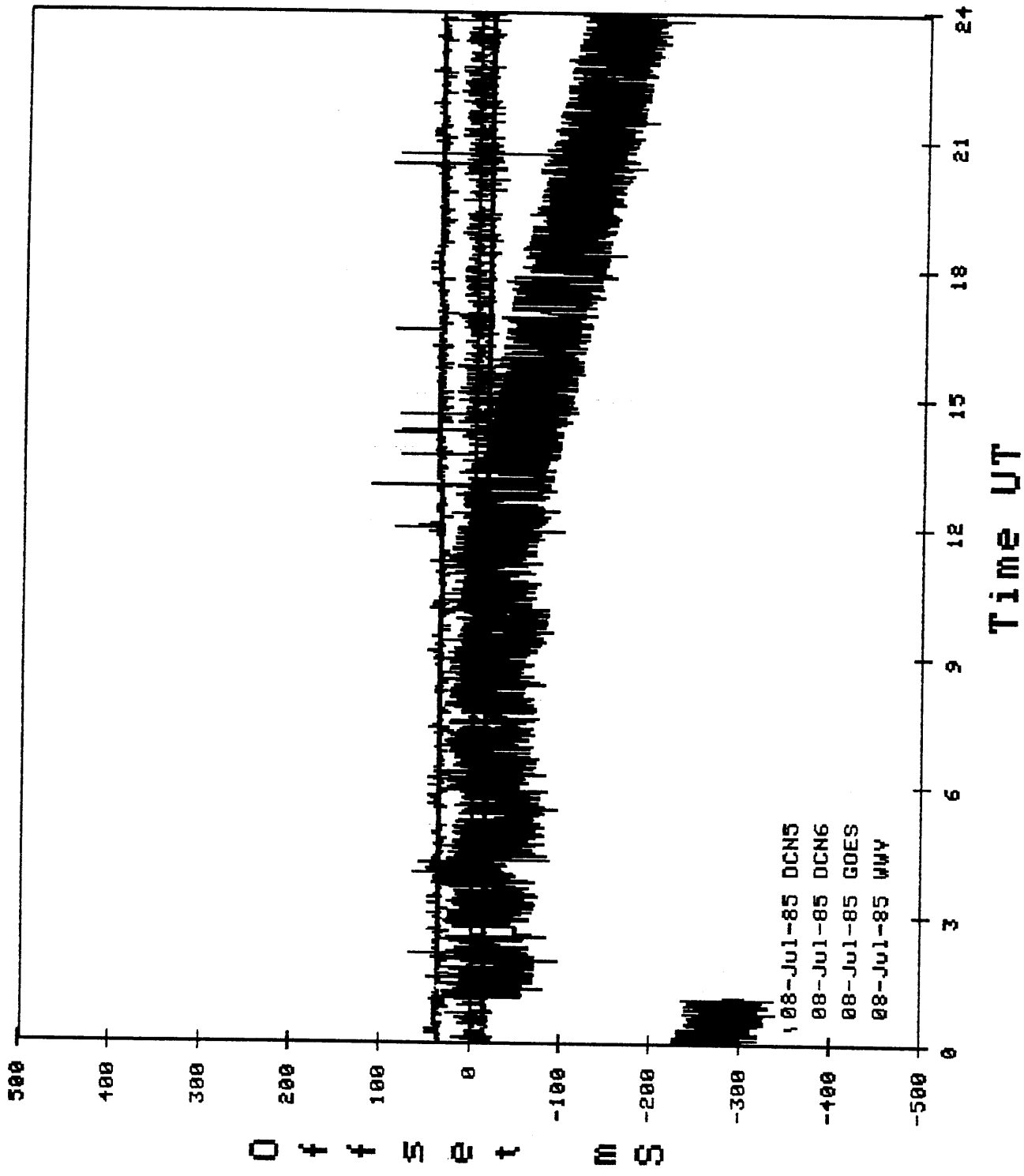
Wiretap Example

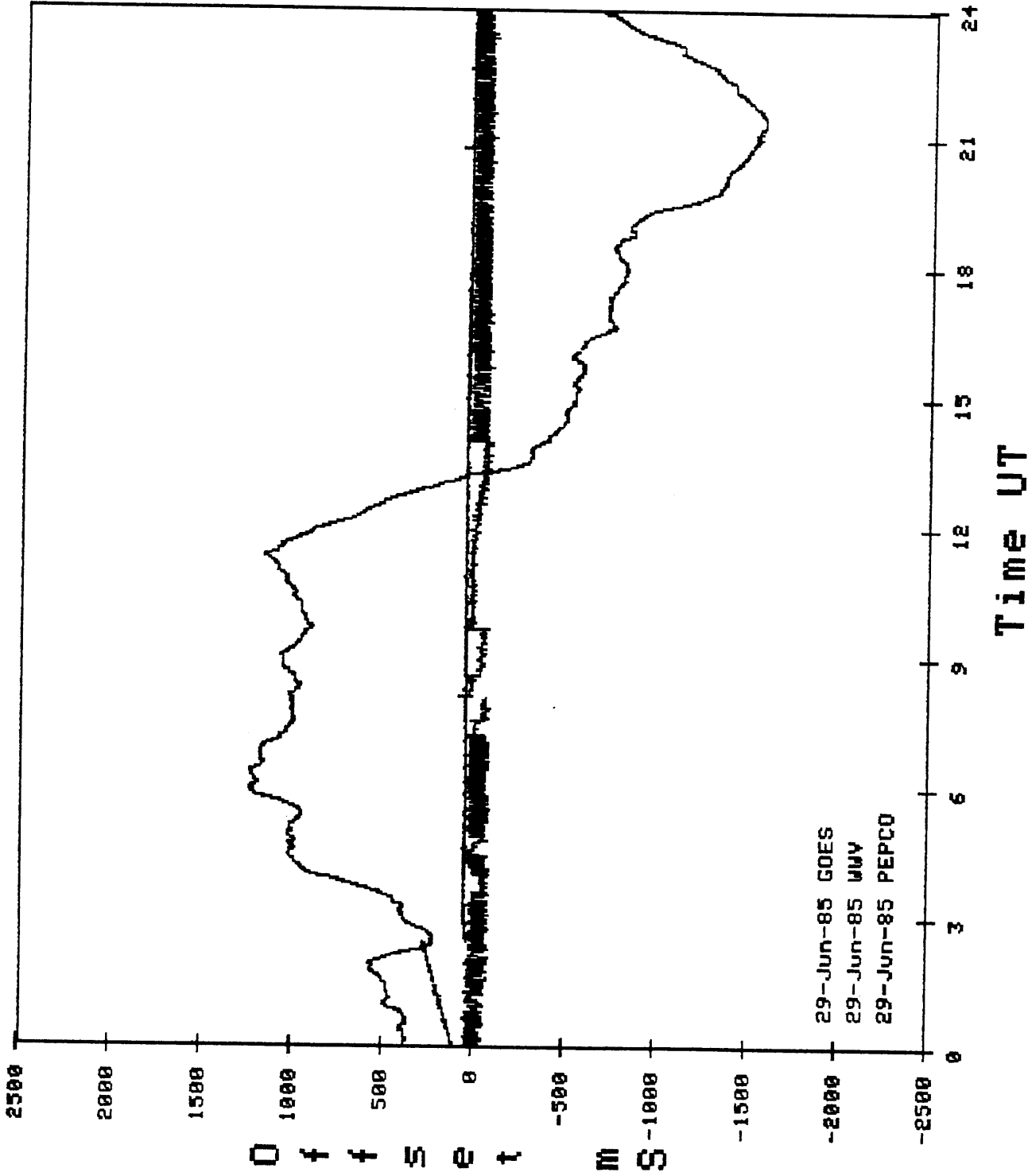
- o Data base management
- o Default routes
- o Use link-quality information (signal strength, retry, etc.)
- o Control "reasonable" reroutes
- o Control rates new routes are computed
- o Integrate with channel management
- o Develop active probes and strategies

Further Development

Network Time Protocol (NTP)

- **Synchronizes noisy, mutually-suspicious network clocks**
- **Capable of accuracies to some fraction of network delays**
- **Provides accuracy and drift estimates**
- **Based on User Datagram Protocol (UDP)**
- **Replaces RFC-868 UDP/TIME protocol**
- **Suitable for distributed-peer and broadcast configurations**

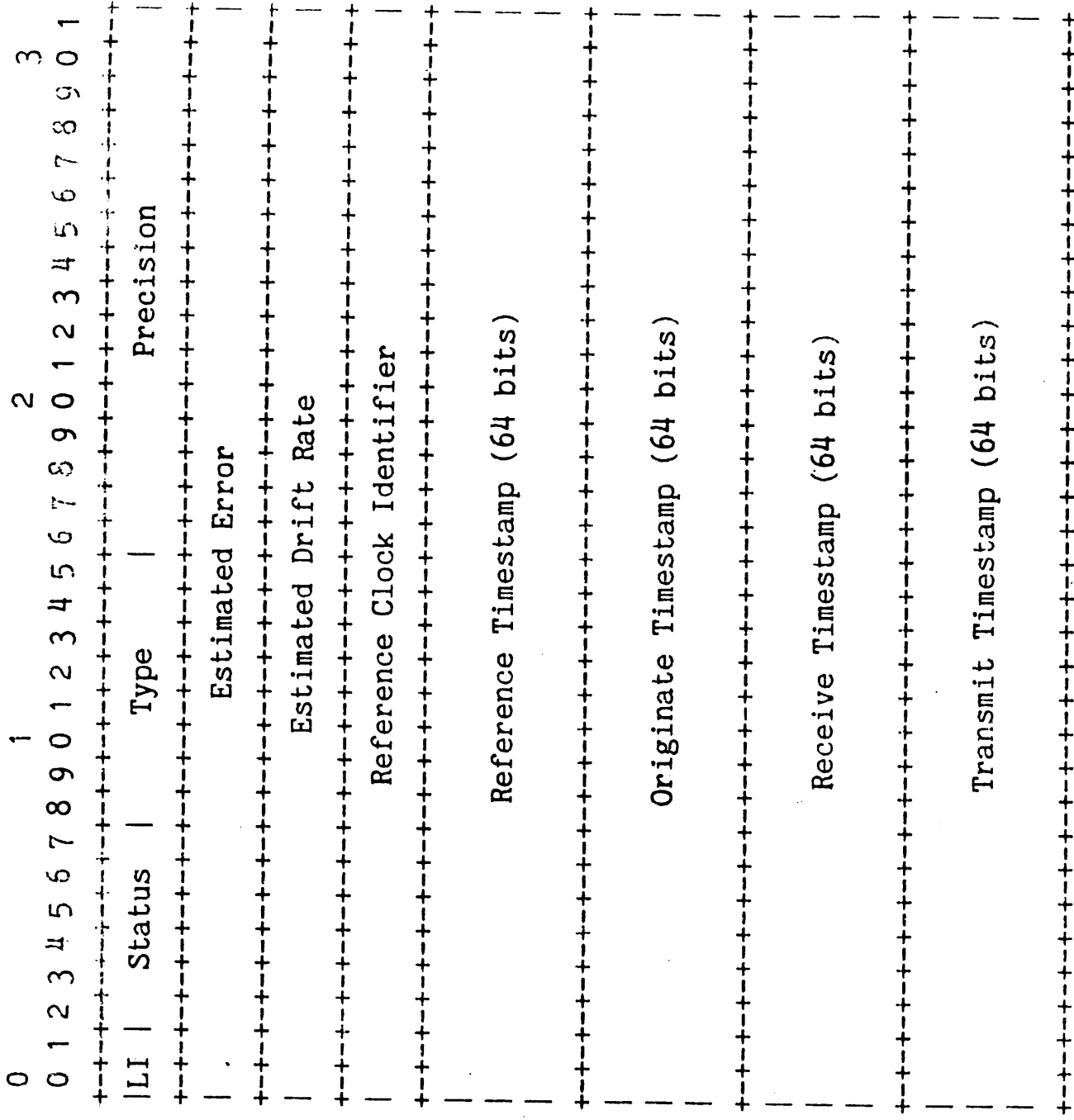




RFC-958 Protocol Features

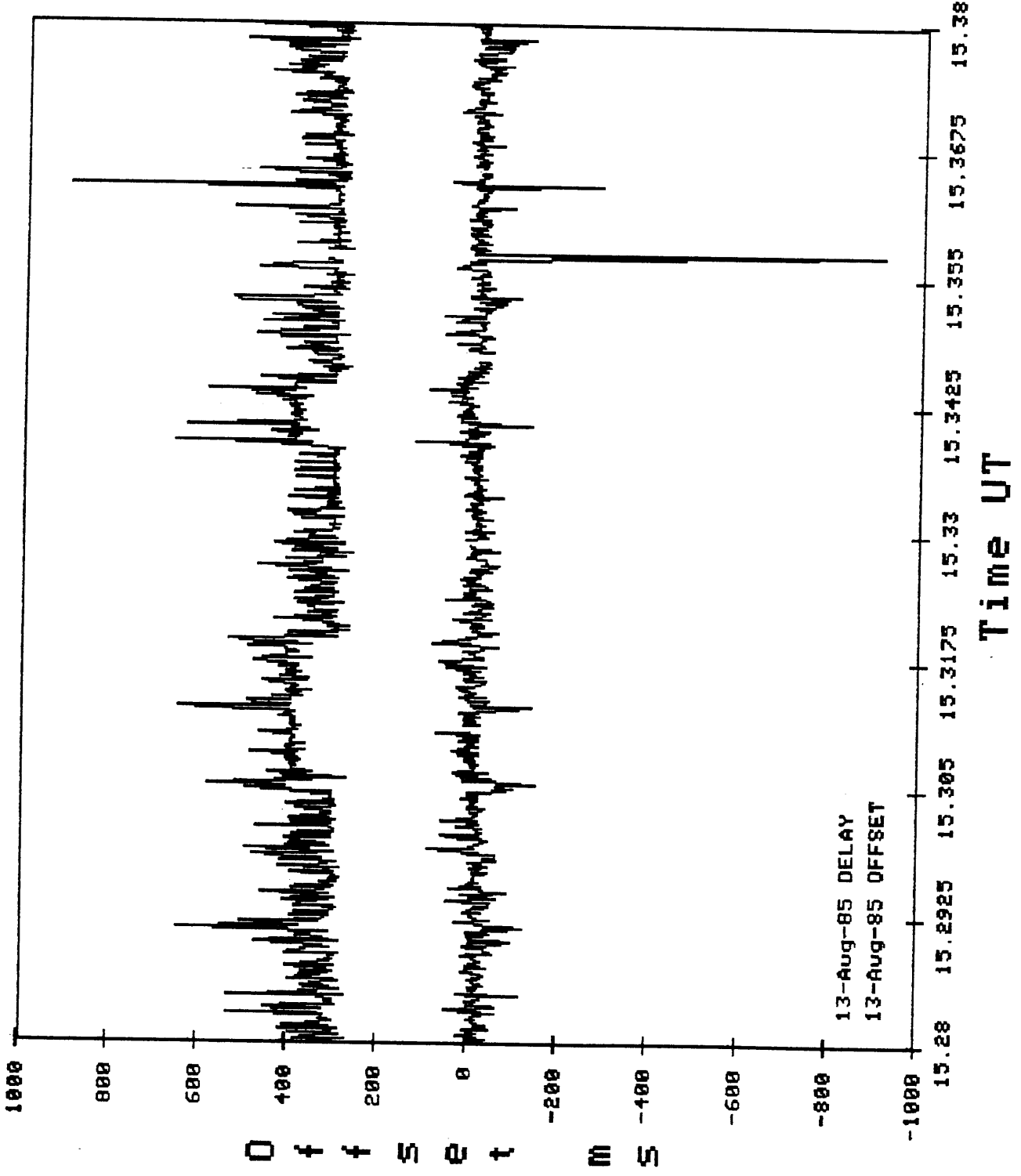
- Operates in symmetric and unsymmetric modes
 - Unsymmetric mode involves conventional user/server interactions
 - Symmetric mode intended for distributed-peer configurations
- Includes provisions for leap-seconds, loop detection and reference clock identification
- Provides precision, estimated accuracy and estimated drift rate
- Does not specify synchronization algorithm

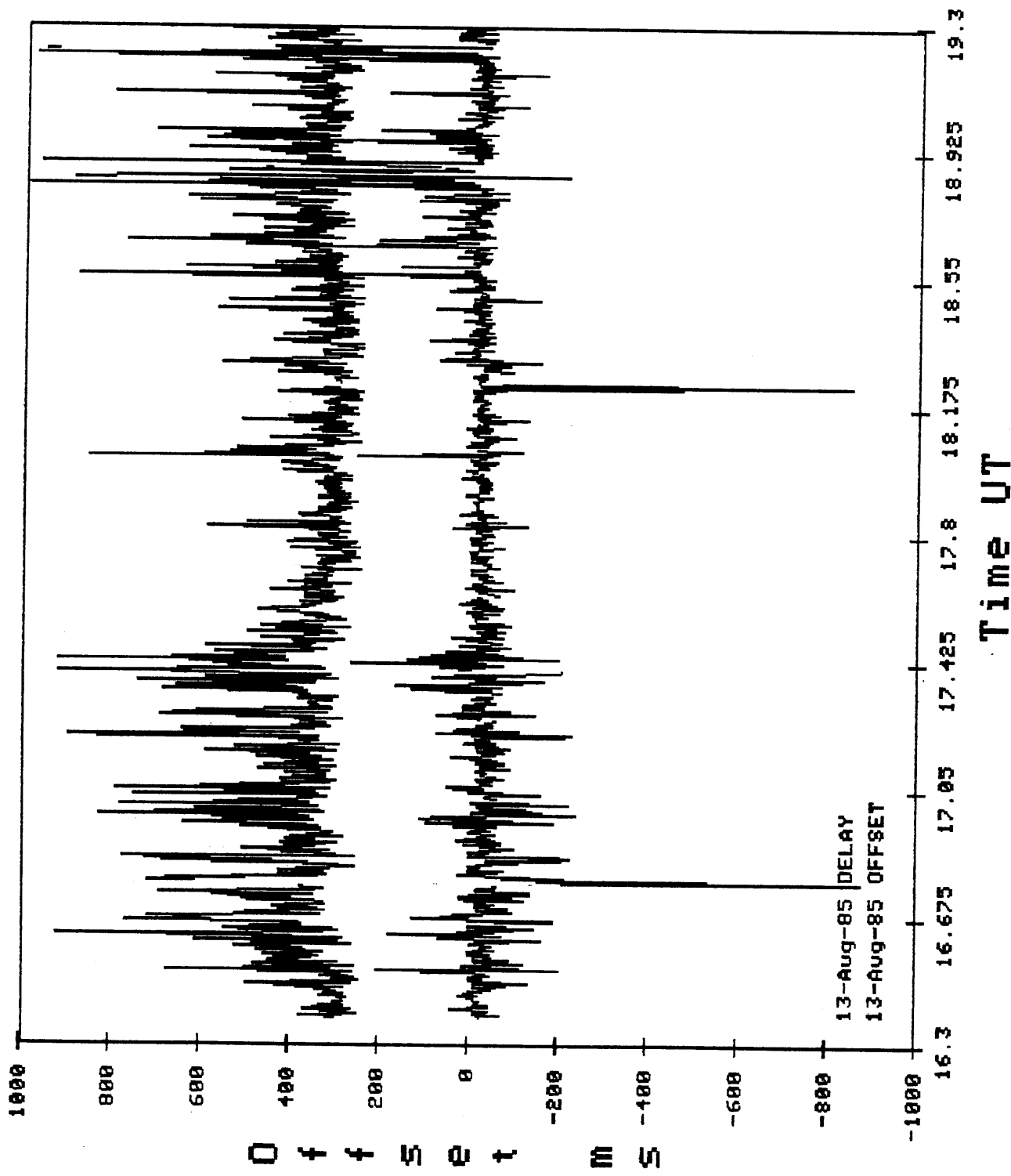
NTP Header Format



Synchronization Principles

- Assume majority of clocks are distributed about the correct time
- Assume remainder are uniformly distributed over the indication interval
- Need algorithms which can reliably synchronize to the good clocks while ignoring the bad
 - Majority-subset algorithms
 - Election algorithms
 - Clustering algorithms





LL-GW

$C(n,k)$ for n from 2 to 20

- n is the number of clocks
- k is the next largest integer in $n/2$, that is, the minimum majority
- majority subset consists of k of n offset measurements
- total number of subsets is $C(n,k)$
- choose subset with smallest variance

(n,k)	$C(n,k)$	-----	(n,k)	$C(n,k)$
(2,2)	1		(11,6)	462
(3,2)	3		(12,7)	792
(4,3)	4		(13,7)	1716
(5,3)	10		(14,8)	3003
(6,4)	15		(15,8)	6435
(7,4)	35		(16,9)	11440
(8,5)	56		(17,9)	24310
(9,5)	126		(18,10)	43758
(10,6)	210		(19,10)	92378
			(20,11)	167960

Example Clustering Algorithm

1. Start with a sample set of n observations $\{x(1), x(2), \dots, x(n)\}$
2. Compute the mean of the n observations in the sample set.
Discard the single sample $x(i)$ with value furthest from the mean, leaving $n-1$ observations in the set.
3. Continue with step 2 until only a single observation is left, at which point declare its value the maximum-likelihood estimator.

Clustering Algorithm using ICMP Timestamp Data

Size	Mean	Var	Discard
504	-3.0E+6	3.2E+14	8.6E+7
500	-3.3E+6	2.9E+14	8.6E+7
450	-1.6E+6	3.0E+13	-2.5E+7
400	29450	2.2E+11	3.6E+6
350	-3291	4.1E+9	-185934
300	3611	1.6E+9	-95445
250	2967	6.8E+8	66743
200	4047	2.3E+8	39288
150	1717	8.6E+7	21346
100	803	1.9E+7	10518
80	1123	8.4E+6	-4863
60	1119	3.1E+6	4677
50	502	1.5E+6	-2222
40	432	728856	2152
30	84	204651	-987
20	30	12810	338
15	28	2446	122
10	7	454	49
8	-2	196	24
6	-9	23	0
4	-10	5	-13
2	-8	0	-8

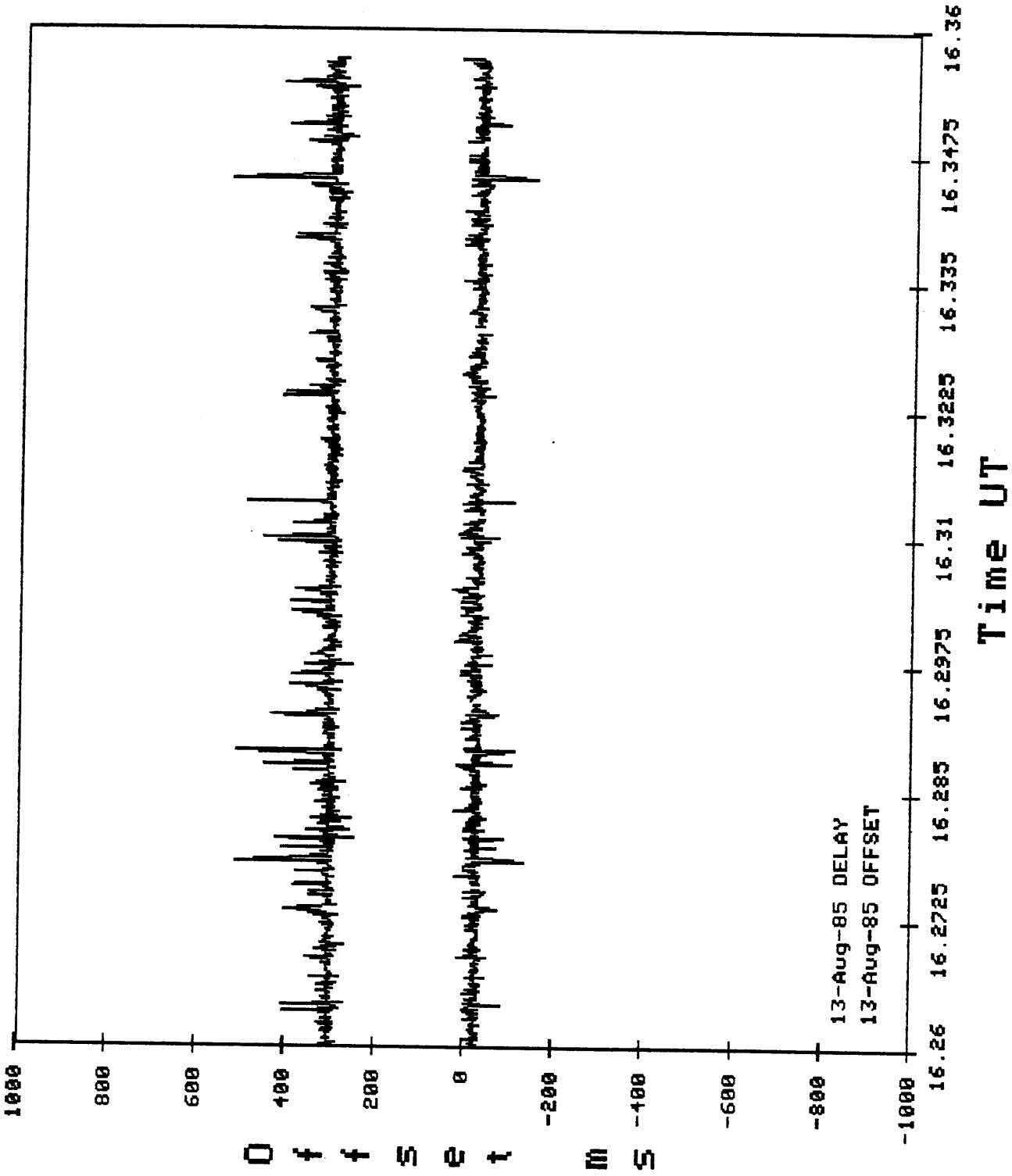
Comparison of Algorithms

	Mean	Dev	Max	Min
Raw data	566	1.8E+7	32750	-143
C(5,3)	-23	81	14	-69

LL-GW (a) Majority-Subset Algorithm

Size	Mean	Var	Discard
1000	566	1.8E+7	32750
990	242	8.5E+6	32726
983	10	1.0E+6	32722
982	-23	231	-143
980	-23	205	-109
970	-22	162	29
960	-23	128	13
940	-23	105	-51
900	-24	89	1
800	-25	49	-9
700	-26	31	-36
600	-26	21	-34
500	-27	14	-20
400	-29	7	-23
300	-30	3	-33
200	-29	1	-27
100	-29	0	-28
1	-29	0	-29

LL-GW (a) Clustering Algorithm



LL-GW

Comparison of UDP and ICMP Host Clock Offsets

Host	UDP time	ICMP time
DCN6.ARPA	0 sec	0 msec
DCN7.ARPA	0	0
DCN1.ARPA	0	-6
DCN5.ARPA	0	-7
UMD1.ARPA	0	8
UMICH1.ARPA	0	-21
FORD1.ARPA	0	31
TESLA.EE.CORNELL.EDU	0	132
SEISMO.CSS.GOV	0	174
UT-SALLY.ARPA	-1	-240
CU-ARPA.CS.CORNELL.EDU	-1	-514
UCI-ICSE.ARPA	-1	-1896
UCI-ICSC.ARPA	1	2000
DCN9.ARPA	-7	-6610
TRANTOR.ARPA	10	10232
COLUMBIA.ARPA	11	12402
GVAX.CS.CORNELL.EDU	-12	-11988
UCI-CIP5.ARPA	-15	-17450
RADC-MULTICS.ARPA	-16	-16600
SU-WHITNEY.ARPA	17	17480
UCI-ICSD.ARPA	-20	-20045
SU-COYOTE.ARPA	21	21642
MIT-MULTICS.ARPA	27	28265
BBNA.ARPA	-34	-34199
UCI-ICSA.ARPA	-37	-36804
ROCHESTER.ARPA	-42	-41542
SU-AIMVAX.ARPA	-50	-49575
UCI-CIP4.ARPA	-57	-57060
SU-SAFE.ARPA	-59	-59212
SU-PSYCH.ARPA	-59	-58421
UDEL-MICRO.ARPA	62	63214
UIUCDCSB.ARPA	63	63865
BELLCORE-CS-GW.ARPA	71	71402
USGS2-MULTICS.ARPA	76	77018
BBNG.ARPA	81	81439
UDEL-DEWEY.ARPA	89	89283
UCI-CIP3.ARPA	-102	-102148
UIUC.ARPA	105	105843
UCI-CIP2.ARPA	-185	-185250
UCI-CIP.ARPA	-576	-576386
OSLO-VAX.ARPA	3738	3739395
DEVVAX.TN.CORNELL.EDU	3657	3657026
PATCH.ARPA	-86380	20411
IPTO-FAX.ARPA	-86402	-1693
NETWOLF.ARPA	10651435	-62164450

Congestion in the Internet Doing Something About It

John Nagle

Ford Aerospace
and Communications Corporation

Good guys and bad guys

- We've been through this before, but it's still the big problem.
- A few bad guys can ruin it for everybody.
- There are still a lot of bad guys.
- I think that proportionally the bad-guy ratio is decreasing but but in absolute numbers there are more bad guys than ever before.
- We don't seem to be winning on this.

What's a bad guy?

- Bad guys are host implementations that talk too much. Usually this is due to bugs in TCP.
- Standard bug #1: retransmit timers that go off too fast.
- Standard bug #2: tinygrams
- Standard bug #3: ignoring ICMP Source Quench
- Good solutions are known for all these problems. There's no theory problem here any more; just ordinary bugs.

Just how bad is it?

- Bad implementations can easily generate an order of magnitude more traffic than necessary.
- If you are out in a 9600 baud datagram net, one bad guy can kill much of the net.

Beating on the bad guys

- Any gateway operator with good logging knows who the troublemakers are. Today this is mostly Dave Mills and myself.
- There's no effective formal mechanism for doing anything about the bad guys.
- Nagging doesn't work with the commercial vendors.
- Bad guys can pass DCA's TCP "validation".
- The TCP spec is not tight enough to fix this. "Maximum freedom for the implementor", remember? The 1984 TCP spec revision was a bust; SDC ran out of money before finishing it.
- Fixing the bugs in other people's implementations is the most effective approach, but expensive, and only feasible when you have source.

Networking despite the bad guys

- Can we make it work despite them? I think so.
- Look upon a bad guy as you would a program in a loop on an operating system. It's a resource hog, but if the resource allocation algorithms are decent, it doesn't hurt too much.
- We need smarter resource allocation in our networks.

Fair queuing

- Basic concept: equalize resource allocation amongst source hosts.
- Individual queues for each output link for each source IP address. Service queues round-robin fashion. (Implementation is not too hard. See RFC970).
- Send Source Quench whenever a queue length exceeds 1 or 2.
- If you run out of buffers, take one from the end of the longest queue.
- Host should thus adapt to have just the number of packets in transit that maximizes throughput without building up a queue in any node.

Optional additions

- Implement time-to-live countdown on the queues. Discard packets that time out.
- Discard IP datagrams instead of sending them when $TTL < \text{hops remaining to destination}$. When this is done, the queue misses its turn in the round-robin. This has the effect that the worst a host behaves, the less line time it gets, and the worst hosts get NO line time at all under overload.

Impact of fair queuing

- Nobody has implemented it yet. But implementation doesn't look too hard. See RFC970 for a way to do the queuing efficiently.
- It may go in Multinet Gateway, but that is some time off.
- We need to try it and see what happens, preferably in a gateway with substantial memory resources.
- Incidentally, more memory in the gateways will not by itself control congestion, and may make it worse, although it provides some relief from shock loads. We have some amusing experimental data obtained with a 10,000 buffer gateway.

Applicability of fair queuing

- Clearly fair queuing should help in the LAN to slow net gateways. Where a small host population generates traffic through a gateway that has a huge bandwidth drop to manage, the benefit is obvious.
- But what about interior gateways, those between long-haul nets and links, used by a large host population? We need more analysis here.
- A promising thought: what is the number of different hosts represented in the datagrams in a gateway near the interior of the network? In theory, this number only increases as the diameter of the network. Fair queuing may still be useful in interior gateways of very sizable networks. But this remains an open question.
- Fair queuing on a per-process (or per-user) basis in hosts may be useful, in equalizing service offered to each user where the output interface is slow.

Ultimate performance limits

- Can the Internet ever perform as well as the IMP system? I am beginning to think so.
- The Internet has suffered because it had no effective means of dealing with host-induced overload other than asking the hosts to exercise restraint. Now we have discovered stronger measures to take.
- The present scheme for dealing with ICMP Source Quench, combined with fair queuing, may be as powerful as the new IMP throttling mechanism.
- It may even be better. There is some argument that throttling the number of outstanding messages on a connection (as we now know to do with TCP) is better than throttling the outgoing message rate (as has been shown to be unsatisfactory where Source Quench was used to control IP-level throttles).

What about non-TCP data?

- Most UDP-based protocols are inquiry-response. Only ones with very short retransmit timers should cause real trouble.
- Fair queuing will keep them under control. But bad guys may lose.
- Someday someone will do a Sun NFS remote file system mount across the Internet. This will be interesting.

What about the ISO protocols

- In TP4, the rules require long retransmission timers; $RTT > TTL$. Good for congestion, bad for noisy nets. But CCITT's priority is to protect the network.
- In general, TP4 seems to have constants specified where TCP is adaptive.
- The tinygram fix won't work in TP4, because it is a block protocol. We will have to fall back to PAD timers in whatever replaces TELNET.
- Is there a Source Quench for ISO NP/TP4?
- It may be necessary to go with an NP-level throttle; with the long retransmission timers, this won't usually cause retransmits.
- Virtual terminal operations may be more sluggish under TP4 than under TCP.

Why not just use virtual circuits?

- It may come to that. Even the IMP system is now offering a virtual circuit interface.
- We may want to use the techniques here in gateways that connect LANs to virtual circuit nets. We then need only gateway to gateway virtual circuits, not host to host or process to process.
- The commercial packet nets have very restricted ideas about per-circuit bandwidth and packet size; they're still thinking terminal-to-host.

Conclusions

- We know enough to attack Internet congestion.
- It can be fixed piecemeal, gateway by gateway and host by host.
- The implementation isn't that tough.
- We don't have to go to virtual circuits, although we may want to.
- Let's get a test going.

**Host Groups:
A Multicast Extension for
Datagram Internetworks**

David Cheriton
Steve Deering

Stanford University

Why multicast?

- efficient multi-destination delivery
 - updating a replicated database
 - conferencing
 - parallel computing
- unknown-destination delivery
 - querying a distributed database
 - finding a network boot server
 - disseminating routing tables

Why not broadcast?

- incurs overhead on uninterested hosts
- more overhead with each new application
- unwanted listeners
- too expensive for large internetworks
- directed broadcast constrained by topology

The Host Group Model

A host group is a set of zero or more hosts.



- an address identifies a group, not a host
- static or dynamic membership



- permanent or transient groups



- special case: permanent, static group of 1



Group Management Interface

CreateGroup(restricted) → group-address, password



JoinGroup(group-address, password) → approval



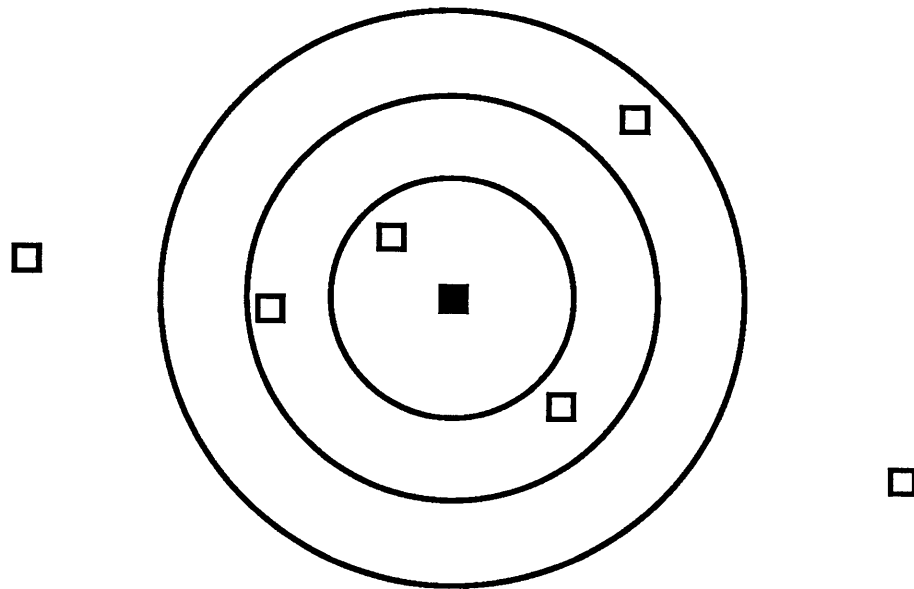
LeaveGroup(group-address) → approval



Datagram Delivery Interface

Send(data, source-address, dest-address, distance)

- deliver to all members within given distance
- refinement of hop-count or time-to-live
- expanding ring searches



Receive() —> data, source-address, dest-address

Implementation

view gateways as "communication servers"

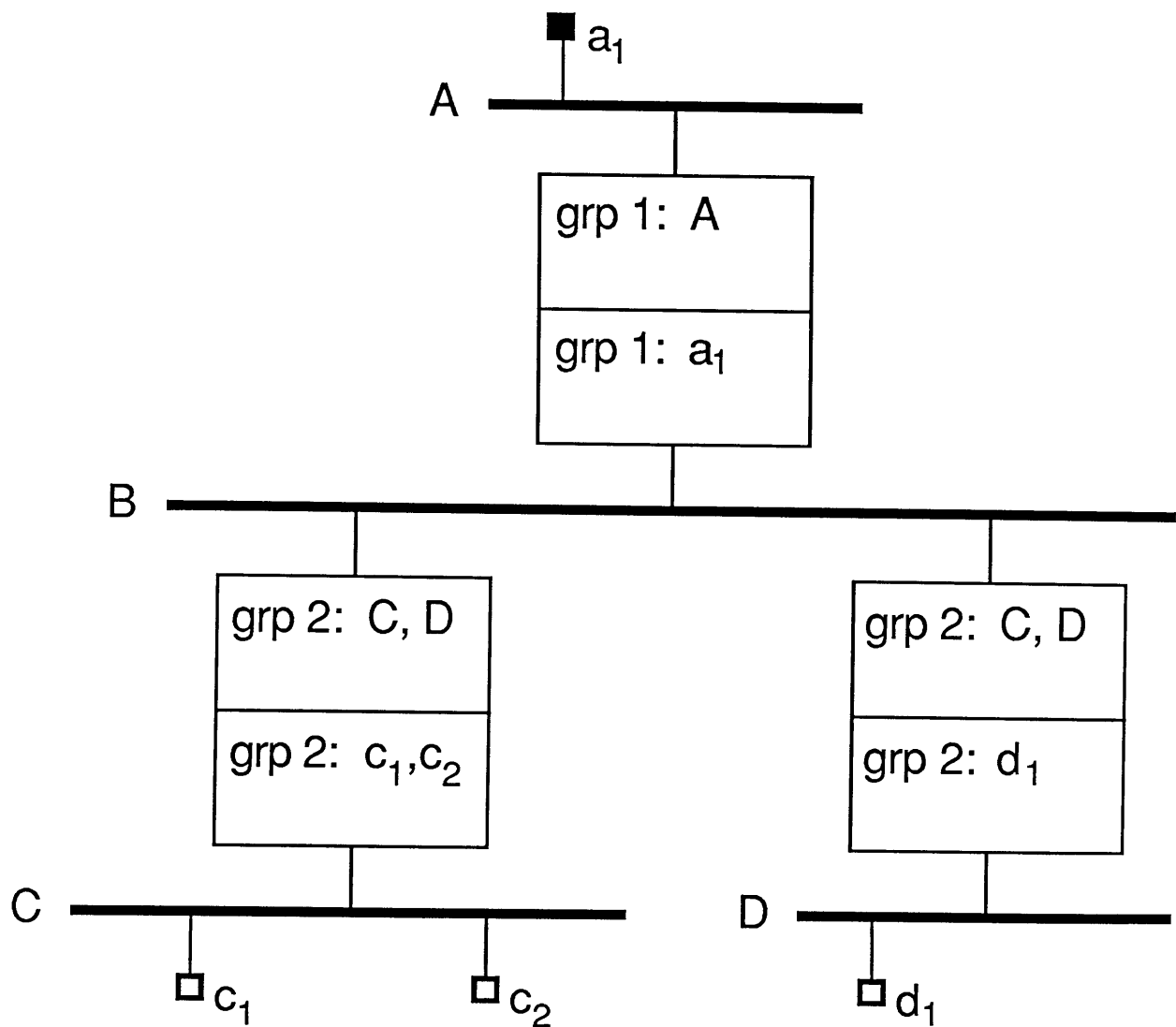
- not just transparent packet shufflers
- group management service
- multicast delivery service

general delivery strategy

- let host group define a network group
- sender delivers to gateway
- gateway delivers to network group
- networks deliver to member hosts

gateway data structures

- routing table
- network membership table
- local host membership table



master copies of network membership record

- replicated by member networks
- infrequent updates
- loose consistency constraints
- omit for permanent static groups of 1

cache copy of network membership record

- reduces table space

local host membership record

- exploit LAN multicast
- possibly cached in local hosts

handling a cache miss

- separate or piggybacked query
- multicast to gateway group
- expanding ring search
- "pruned multicast"

handling stale cache data

- detect on use
- checksum network membership record
- time out unused records

intergateway routing

- shortest-distance spanning tree
- extended reverse path forwarding (Dalal and Metcalfe)

Extensions / Refinements of IP

- host group addresses
 - “class D” addresses used for groups
 - some reserved for permanent groups
 - mapped to local multicast addresses
 - restricted to destination field?
- IGMP for creating/joining/leaving groups
- distance control — refinement of time-to-live
- minor change to ICMP Echo specification

Experiment — Multicast Agents

- “black boxes”, outside of gateways for now
 - add extra hops to delivery path
 - no access to routing information — must use wired-in knowledge
- useful for investigating:
 - internetwork multicast routing
 - internetwork group management
 - applications of internet multicasting

Some gritty details

- source route insertion/deletion for relaying
- extended ARP for Ethernet mapping
- different Ethernet packet type to avoid “destination unreachable” advisories
- delayed replies to ICMP Echo requests

What do we seek from this task force?

- critical comment on our multicast proposal and plans for experimentation
- consideration of multicast requirements in design of next-generation routing protocols
- consideration of multicast as a solution to some internet problems, e.g...
 - locating gateways
 - locating name servers
 - exchanging routing information
- discourage proliferation of broadcast-based protocols, such as ARP or BOOTP

ARPA-INTERNET USE OF OSI NSAP ADDRESSING

- WE NEED TO CHOOSE ADDRESS ENCODINGS FOR USE IN "ISO-GRAMS"
- FOLLOW ISO DRAFT STANDARD
- CAN ADDRESS ANY NSAP IN OSI
- THERE IS NO REQUIREMENT TO BE ABLE TO ROUTE TO ANY NSAP
- CONTINUE TO ROUTE MUCH AS WE DO NOW (INTERIM SOLUTION?)
- "MINIMIZE" CURRENT ADDRESSING KLUDGES
- SIMPLIFY HANDLING FOR X.25 HOSTS, ETC...

NSAP ADDRESS CHOICE

- ALWAYS USE BINARY PREFERRED ENCODING, WITH DSP BASED ON BINARY
- ENCODE "USER PROTOCOL" IN NSAP ADDRESS
- APPLY FOR BLOCK OF ADDRESSES UNDER APPROPRIATE "IDI FORMAT" (ie: CATEGORY OF ADDRESS:
 - X.121 (X.121 ADDRESS)
 - ISO DCC (COUNTRY OR G.A.) **
 - F.69 (TELEX)
 - E.163 (PSTN)
 - E.164 (ISDN)
 - ISO 6523-ICD (ISO MEMBER OR ** LIAISON)
 - LOCAL

SIMPLE ADDRESS SCHEME

FIRST OCTET:	"39" (in BCD)	} FIXED (≈ 6) OCTETS
SECOND & Third Octets:	"DCC FOR USA + FILL"	
NEXT 2 or 3 " :	ASSIGNED BY ANSI TO SPECIFY DOD	
THEN :	USE CURRENT DOD ADDRESS	
LAST OCTET:	USER PROTOCOL FIELD	

"BETTER" ADDRESS SCHEME

FIRST OCTET: "39"
SECOND & Third Octets: "DCC FOR USA + FILL"
NEXT 2 or 3 OCTETS: ASSIGNED BY ANSI TO SPECIFY DOD

} FIXED (≈ 6) OCTETS

NEXT OCTET: DEMULTIPLEXING FIELD

IF DEMUX = "0000 0001" REST = { "TYPE A NET #", LOCAL ADDRESS (variable), USER PROTOCOL (1 Octet) } (1 Octet)

IF "0000 0010" REST = { "TYPE B NET #", LOCAL ADDRESS (variable), USER PROTOCOL (1 Octet) } (2 Octets)

IF "0000 0011" REST = { "TYPE C NET #", LOCAL ADDRESS (variable), USER PROTOCOL (1 Octet) } (3 Octets)

HOSTS ATTACHED TO PDN'S

- USE X.121 IDI FORMAT
- "37"; "X.121 ADDRESS (PADDED TO 7 OCTETS)";
"USER PROTOCOL"

LAN'S ATTACHED TO PDN'S

- "37" (1 Octet)
"X.121 Address" (7 Octets)
"LAN" " " (x6 Octets)
"LSAP" (Optional, 802 ONLY, 1 Octet)
"USER PROTOCOL" (1 Octet)
- COULD OPTIONALLY USE ARP INSTEAD OF ENCODING LAN ADDRESS, DETERMINE IF IT IS THERE BY LENGTH OF ADDRESS

USE OF "LOCAL" IDI

- FIRST (≈ 6) OCTETS ARE FIXED
- HAVEN'T GOTTEN ASSIGNMENT FROM ANSI YET
- USE LOCAL FORMAT
 - FOR INTERIM
 - FOR ABBREVIATIO.

"NETWORK ENTITY TITLES"

- FOR USE IN SOURCE ROUTING & RETURN ROUTE
- SAME AS NSAP, BUT DROP LAST OCTET
- CONTEXT IS ALWAYS CLEAR

Type of Service Routing

Problem: How to use Wideband Network

Solution: Use two types of Service

Delay (low)
Throughput (High)

1) Characterize Service Provided by each Type of Network

	Delay	Throughput
Arpanet/Milnet	$\gamma(N)$	N
Ethernet	γ	γ
Ring	γ	γ
PR	γ	N
Satnet	$\gamma(N)$	N
Wideband	N	γ

2) Perform Two SPF Routing Calculations
Delay
Throughput (min Hop)

3) Route based on Delay or Throughput
Selected in IP Header

Also: Define Loose/Strict T.O.S. Bit

MORE TOS

- Different Internet Protocols

IP

ST

ISO/IP

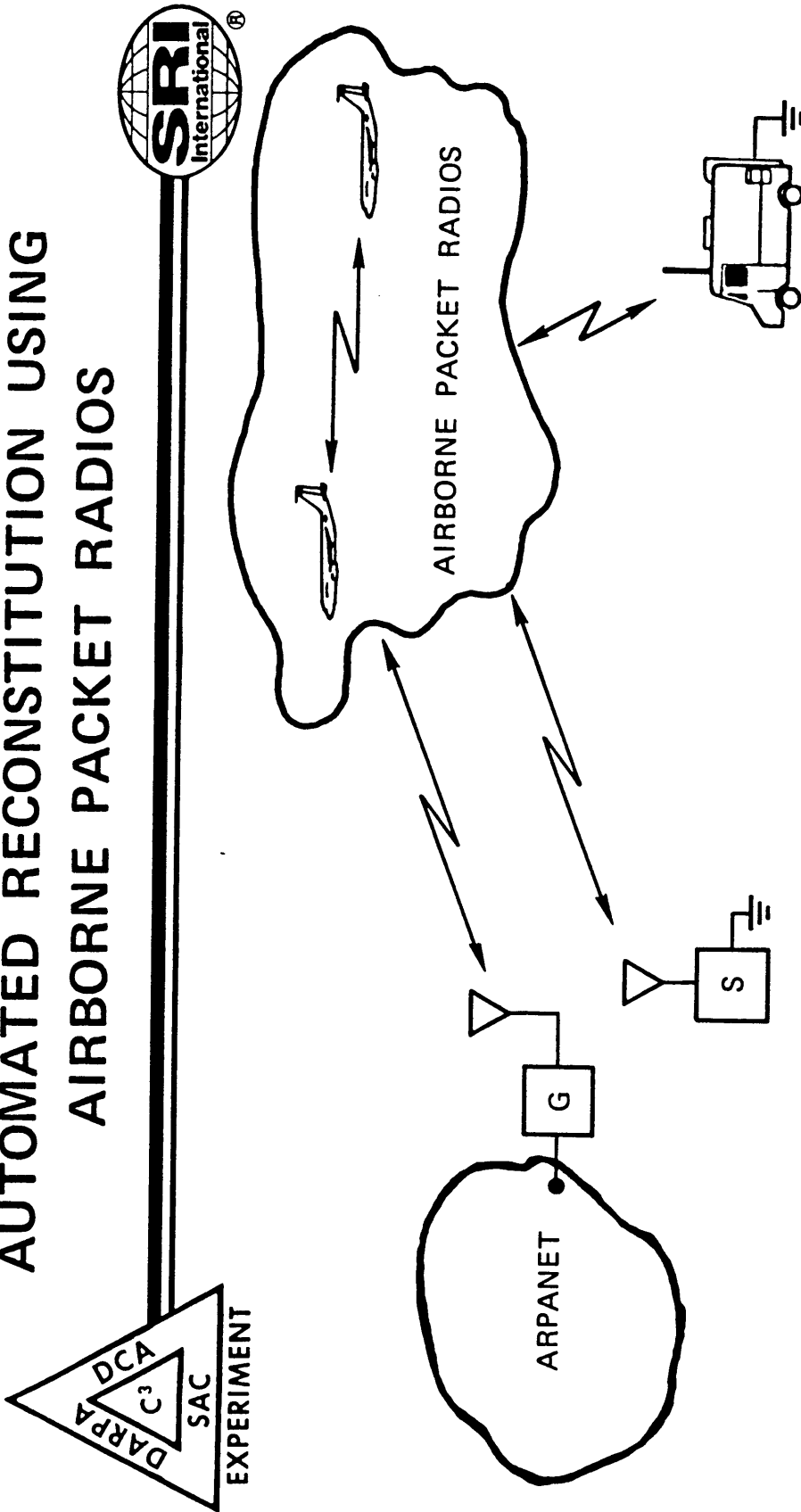
APPENDIX B

Papers Distributed at GADS Meeting

<i>Distributed By:</i>	<i>Paper</i>
J. Nagle	<i>Gateway Database Protocol</i>
Mathis	<i>Automated Reconstitution Using Airborne Packet Radios</i>
A. W. Brown	<i>Merit: Michigan's Universities' Computer Network</i>
Misc.	- <i>Milnet Name Domain Transition Plan</i> - <i>Proposed DDN Bulletin Regarding EGP Table Space</i> - <i>Internet MAP</i>

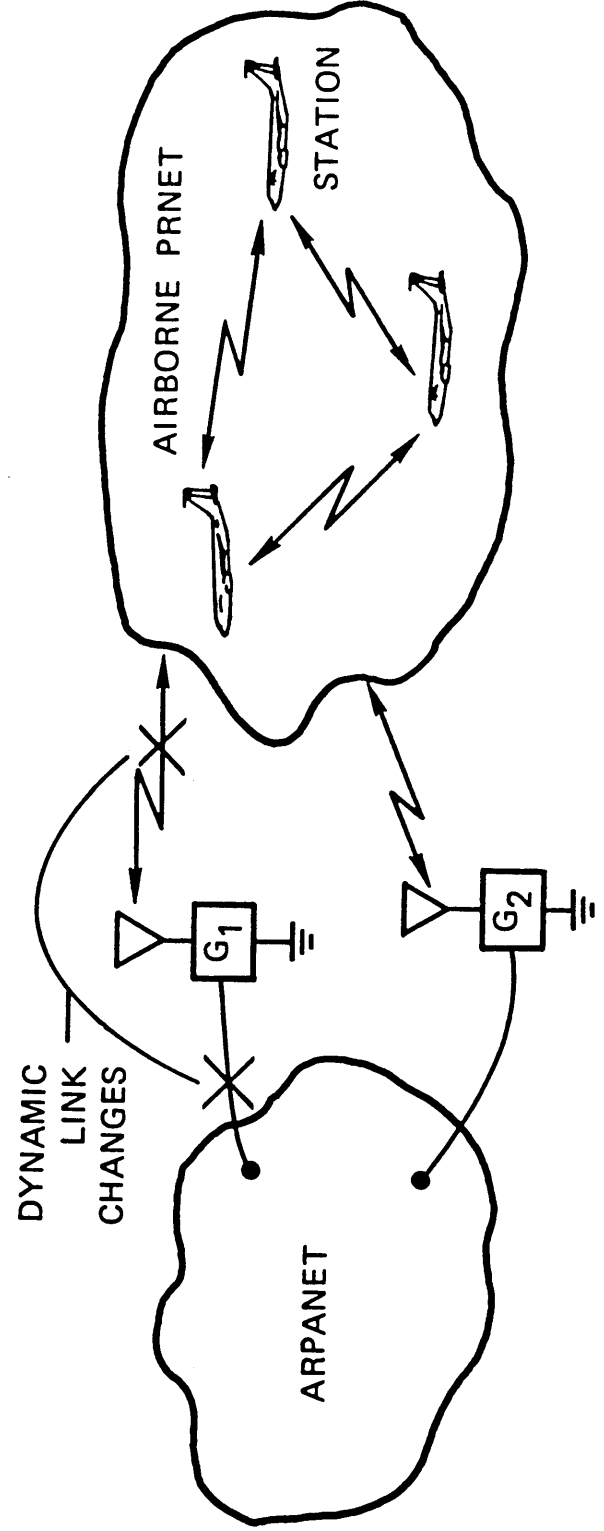
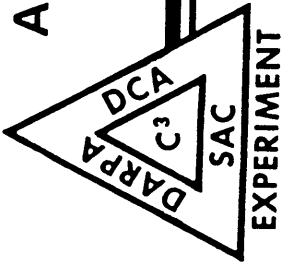
***Note: See Reading List In Minutes For Other Papers Important To This Meeting**

AUTOMATED RECONSTITUTION USING AIRBORNE PACKET RADIOS



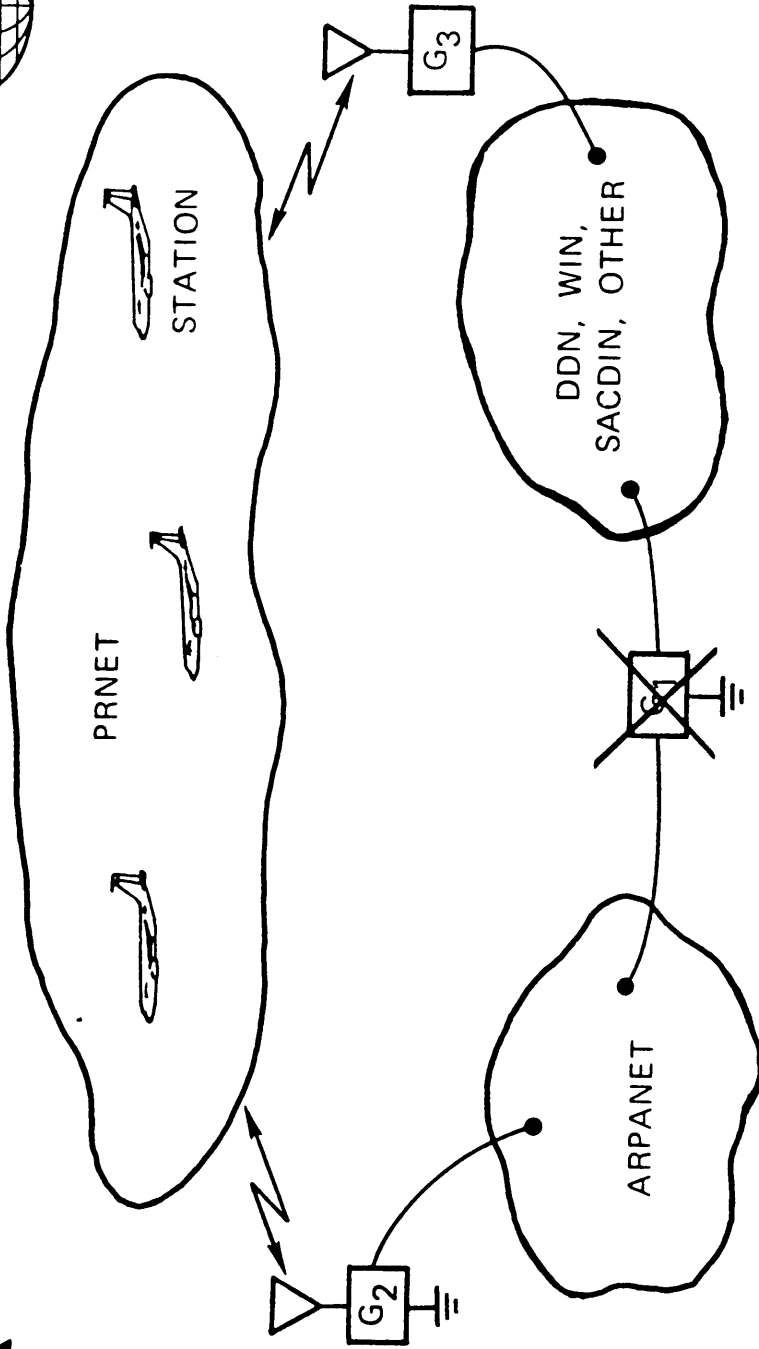
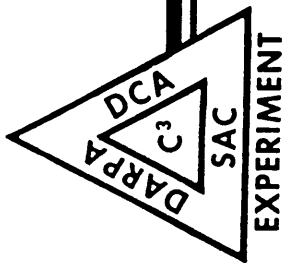
- SINGLE GATEWAY AND SINGLE STATION (GROUND BASED)
- AUTOMATED RECONSTITUTION OF RADIO NETWORK
- DEMONSTRATED DURING 1981

AUTOMATED RECONSTITUTION USING AIRBORNE PACKET RADIO NETWORK (PRNET)



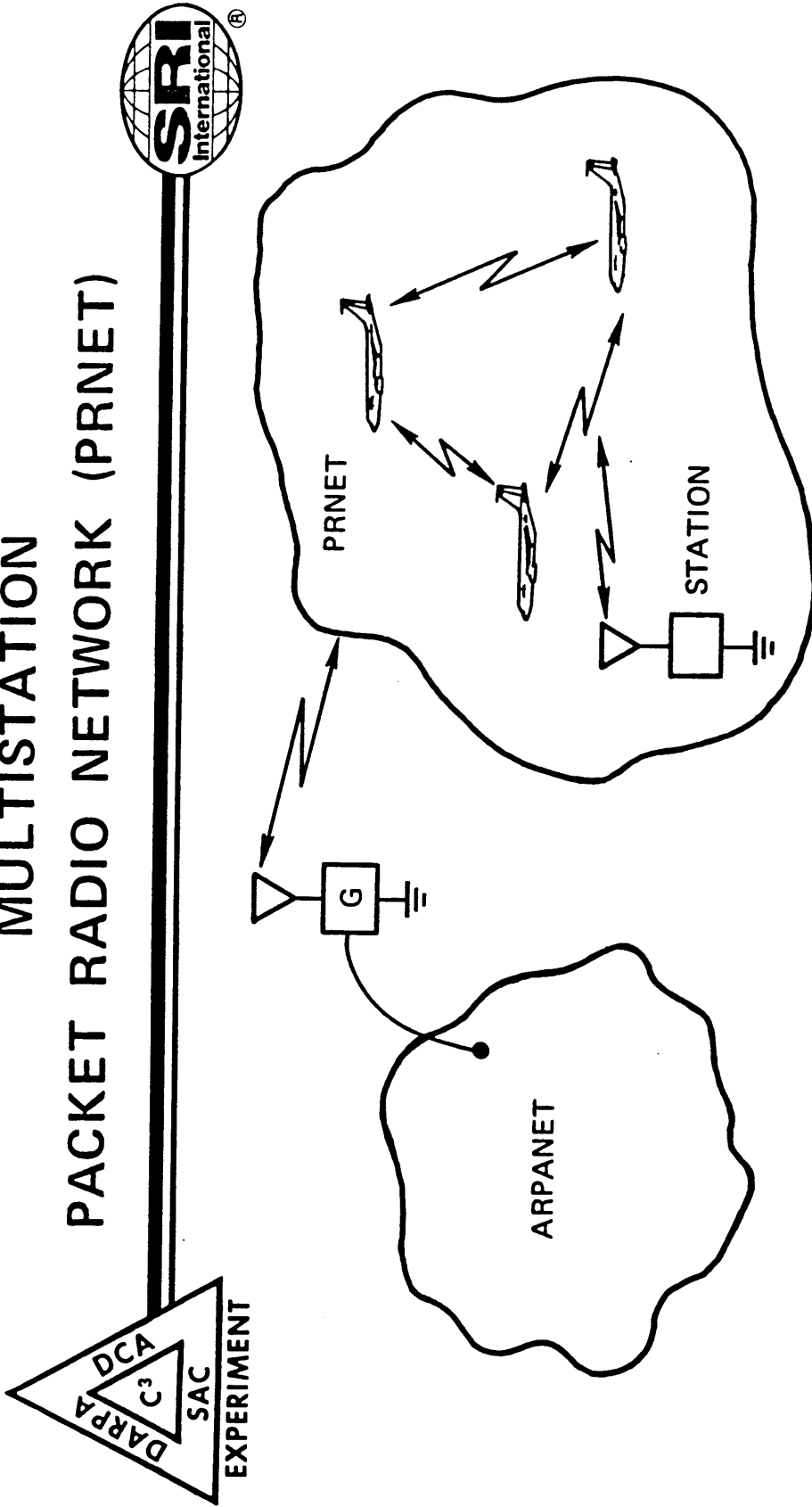
- MULTIPLE GATEWAYS AND AIRBORNE PRNET STATION
- ISSUES
 - MOBILE PRNET STATION
 - NEW IP IMPLEMENTATION FOR TIUs
 - NEW IP IMPLEMENTATION FOR ARPANET HOSTS

INTERNET RECONSTITUTION



- DYNAMIC RECONSTITUTION OF INTERNET UPON GATEWAY FAILURE
- ISSUE
- REQUIRES COMPLETE IP AND ICMP IMPLEMENTATION FOR ALL HOSTS/GATEWAYS INVOLVED

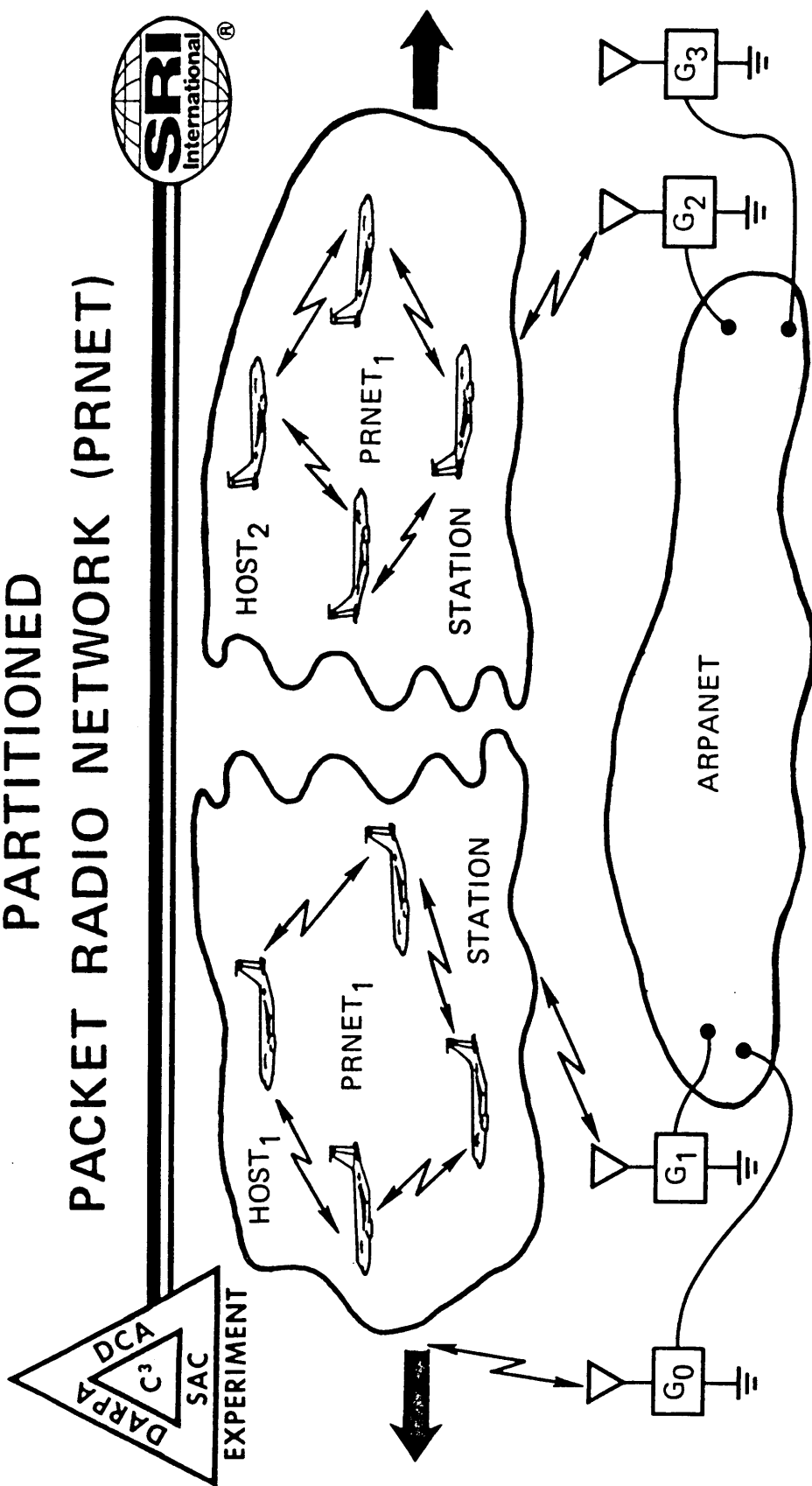
MULTISTATION PACKET RADIO NETWORK (PRNET)



- ONE AIRBORNE, ONE GROUND-BASED STATION
- CAP 6 PROTOCOL IN AN AIRBORNE NETWORK

EXHIBIT E-4

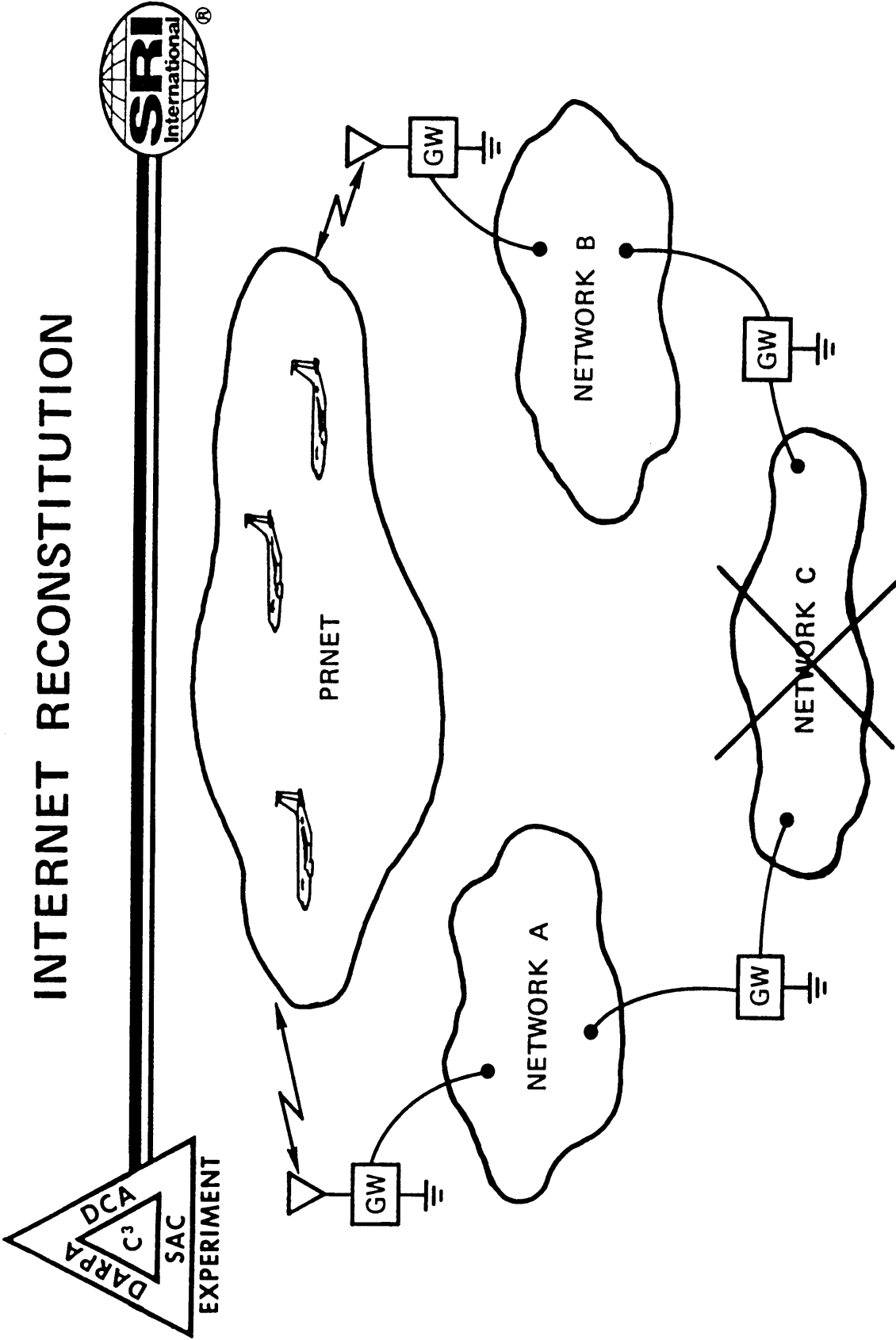
PARTITIONED PACKET RADIO NETWORK (PRNET)



- PRNET HOST₁ TALKS TO PRNET HOST₂ DIRECTLY UNTIL PARTITION OCCURS
- PRNET HOST₁ TALKS TO PRNET HOST₂ THROUGH ARPANET AFTER PARTITION
- ISSUES
 - HOW DO NETS DISCOVER THEY ARE PARTITIONED?
 - HOW DOES INTERNET DISCOVER NETS ARE PARTITIONED?
 - HOW DOES INTERNET COPE WITH A DYNAMIC/PARTITIONING OF NETS?

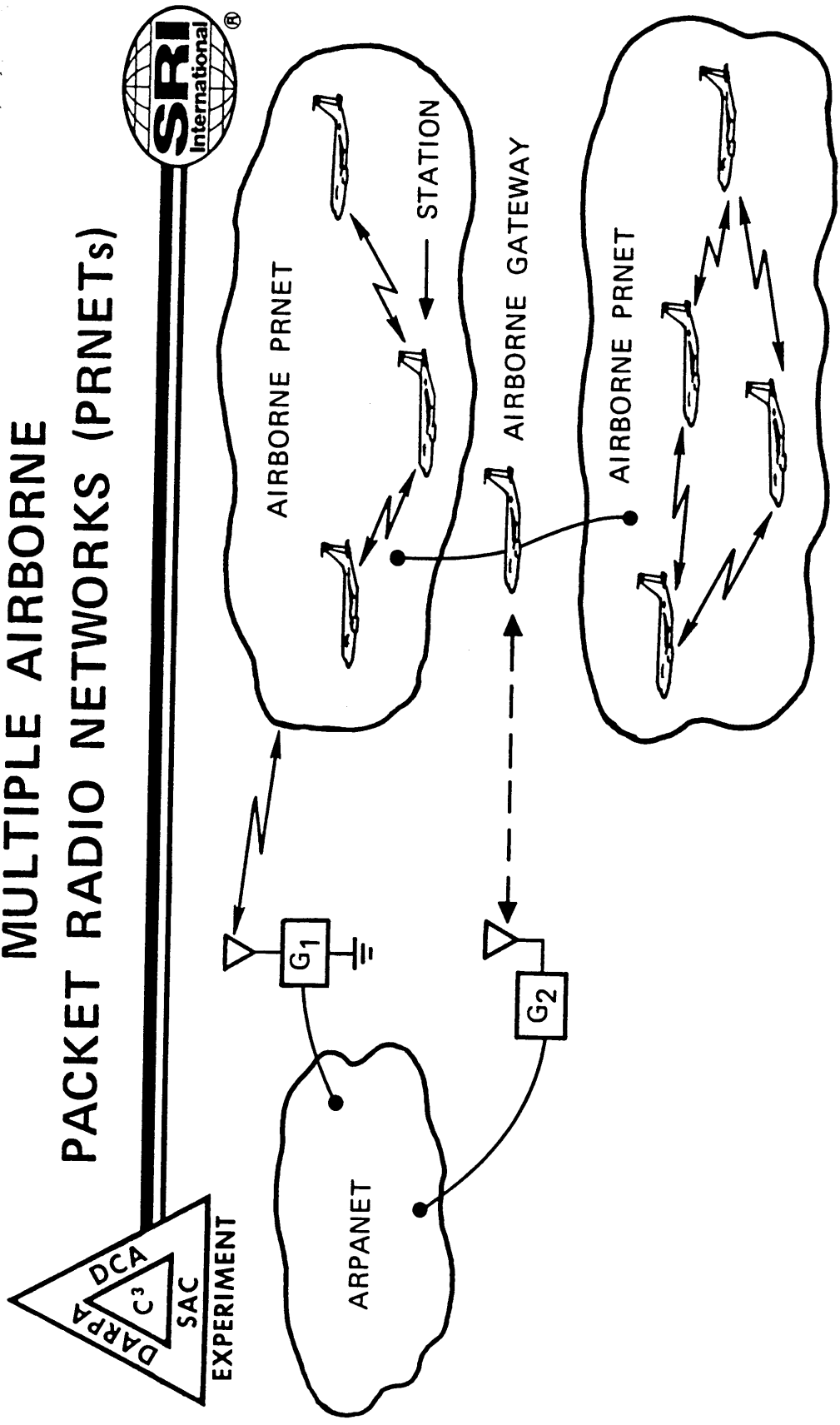
EXHIBIT E-5

INTERNET RECONSTITUTION



- DYNAMIC RECONSTITUTION OF INTERNET AFTER NETWORK C IS DESTROYED
 - POSSIBLE WITH CURRENT INTERNET TECHNOLOGY
- EXHIBIT E-6

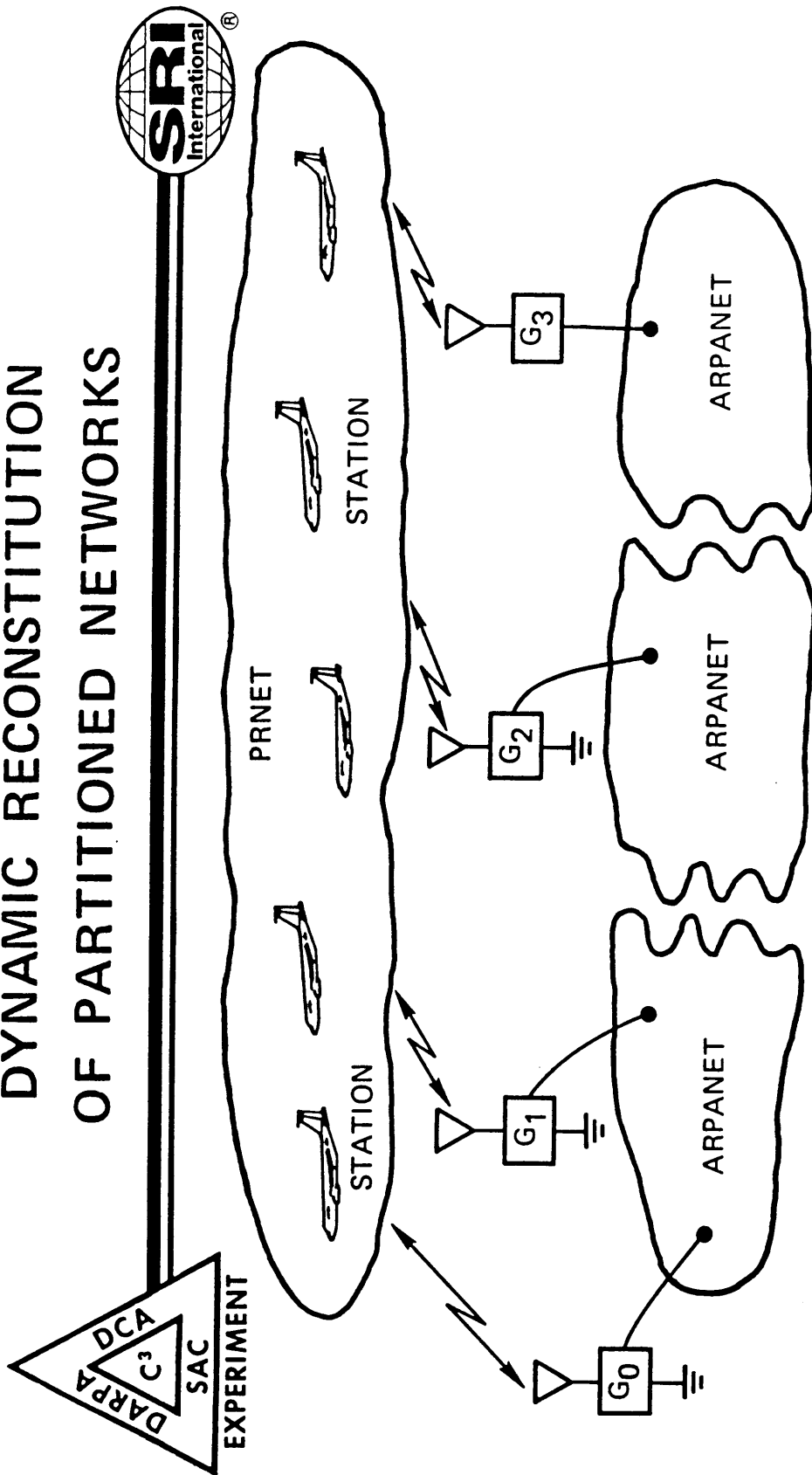
MULTIPLE AIRBORNE PACKET RADIO NETWORKS (PRNETs)



- IMPLEMENTATION OF AIRBORNE GATEWAY
- INTER-PRNET
- FROM AIRBORNE NETS TO LAND-BASED NETS

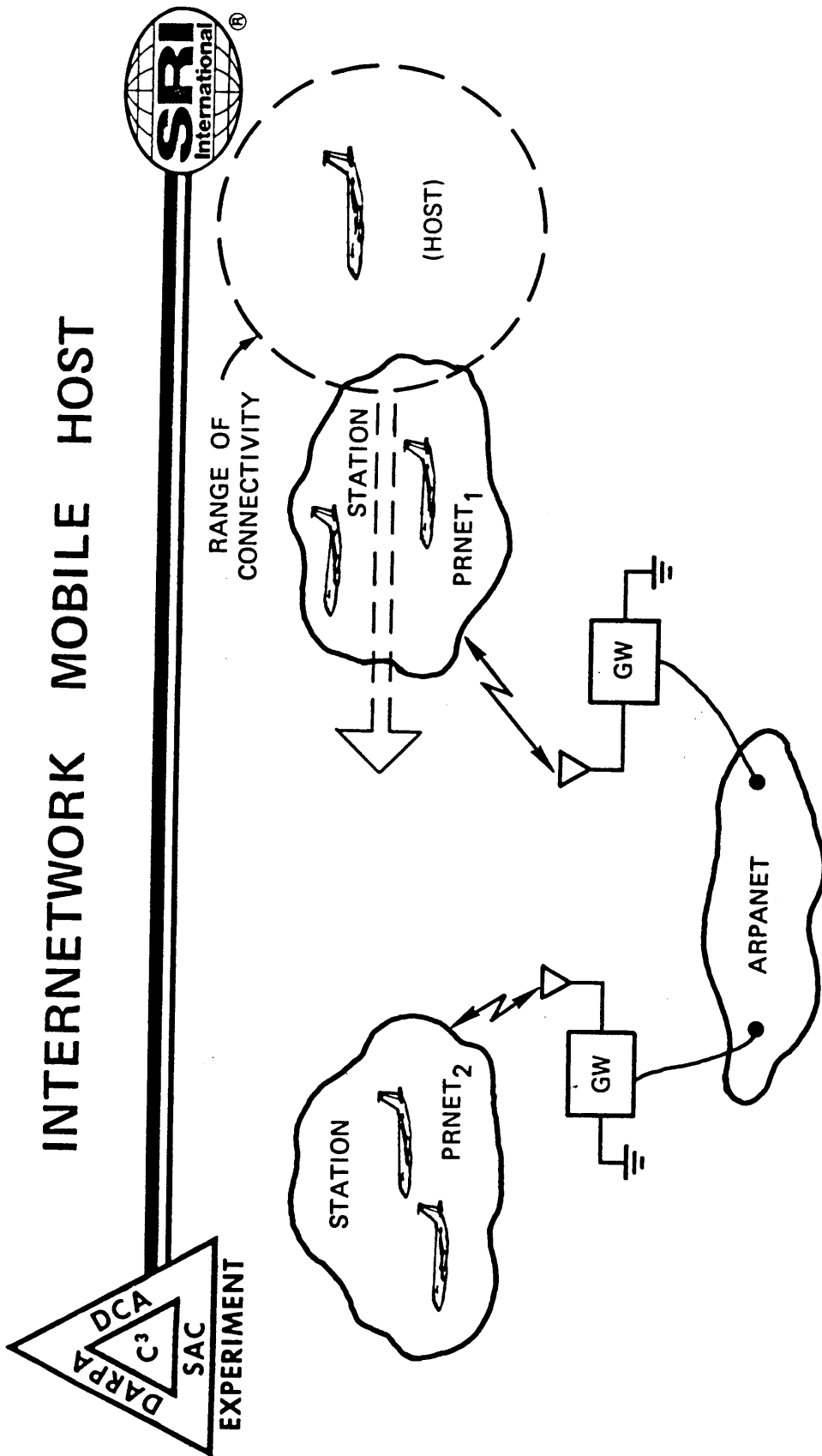
EXHIBIT E-7

DYNAMIC RECONSTITUTION OF PARTITIONED NETWORKS



- DYNAMIC RECONSTITUTION THROUGH PRNET
 - ISSUES
 - HOW CAN GATEWAYS DISCOVER AND COPE WITH PARTITIONED NET?
 - HOW DOES INTERNET FIND OUT IN WHICH PARTITION A HOST IS?
 - HOW DOES A HOST OBTAIN NEW ADDRESS FOR HOST IN PARTITIONS THAT OCCUR DYNAMICALLY?
- EXHIBIT E-8

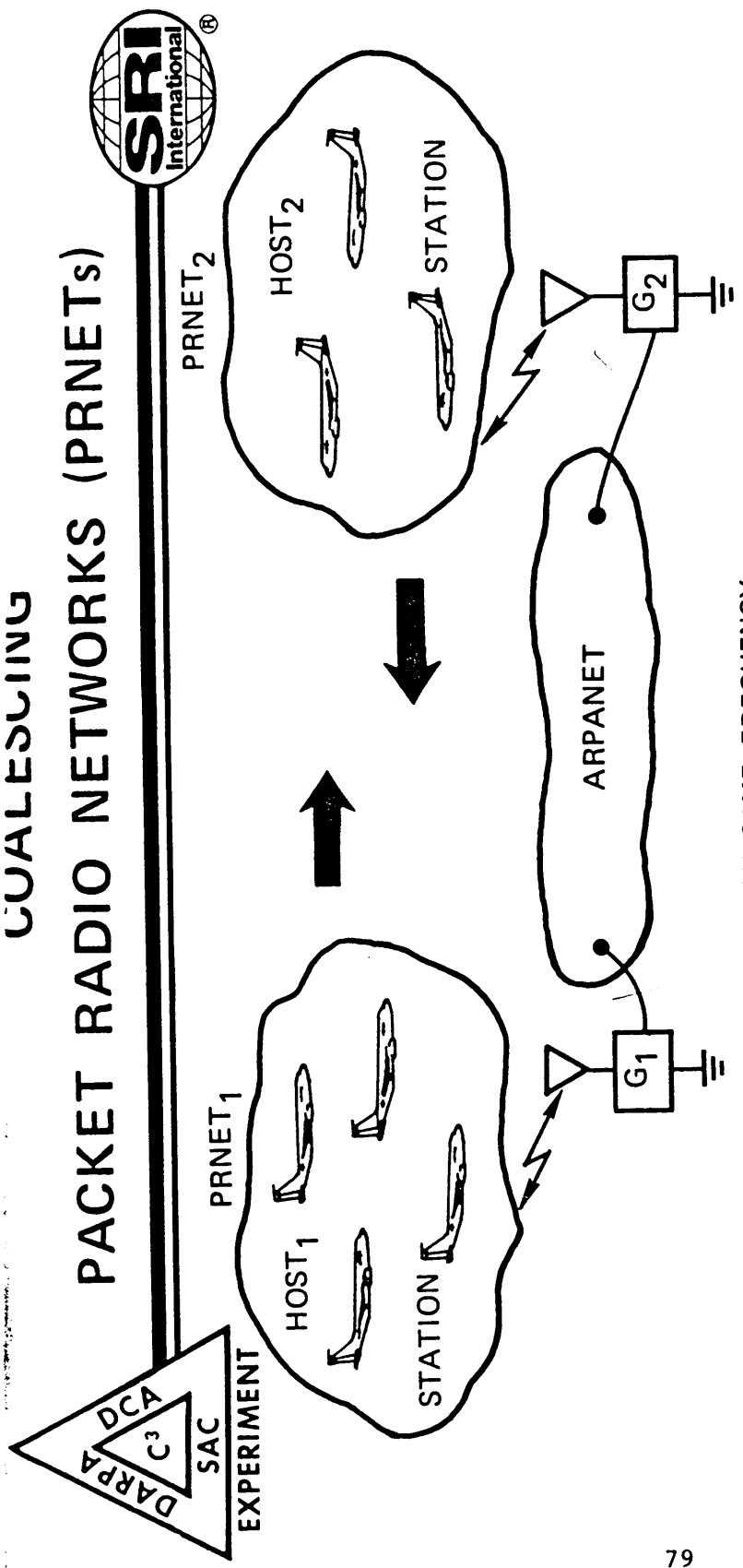
INTERNETWORK MOBILE HOST



- HOST MOVES FROM PRNET₁ TO PRNET₂
 - NEED TO DO WITH NO TCP CONNECTION LOSS
- IMPLEMENTATION ISSUES:
 - SHOULD HOST'S INTERNET ADDRESS CHANGE DYNAMICALLY? (WHO SHOULD BE RESPONSIBLE FOR MANAGEMENT CHANGES?)
 - HOW DOES GATEWAY (OR?) LEARN THAT HOSTS MOVE FROM ONE NETWORK TO ANOTHER?

COALESCING

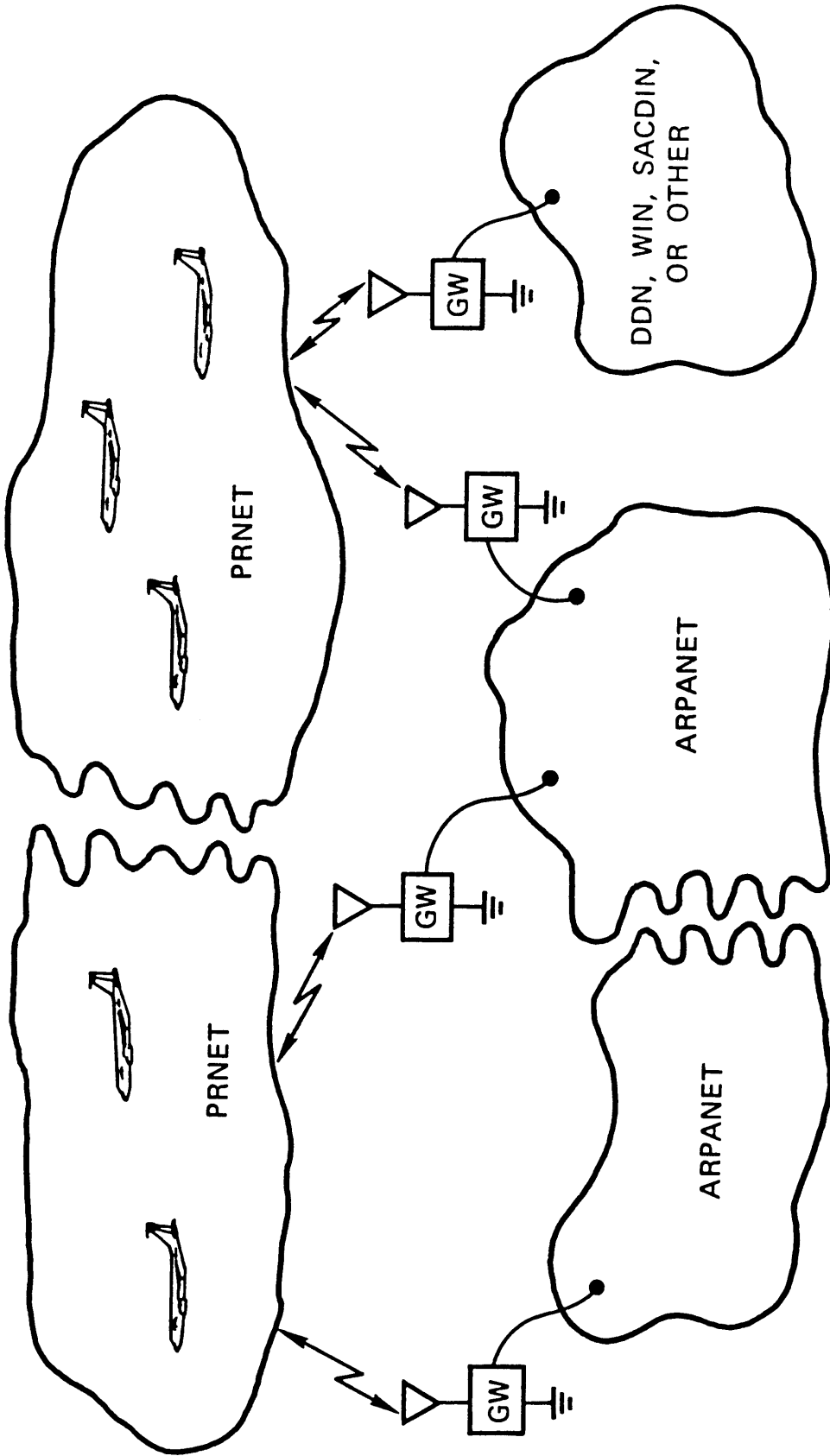
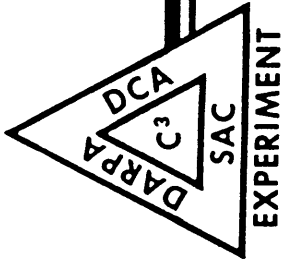
PACKET RADIO NETWORKS (PRNETs)



- TWO PRNETS ON SAME FREQUENCY
 - HOST₁ AND HOST₂ CONNECTED VIA INTERNET
- PRNETS MERGE
 - HOSTS SHOULD BE CONNECTED INTRANET
- ISSUES
 - SHOULD TWO PRNETS WITH DIFFERING NET NUMBERS BUT SAME FREQUENCY BE ALLOWED TO COEXIST?
 - SHOULD PRNET HELP IN DETERMINING HOST LOCATION?
 - HOW SHOULD NETWORK NUMBERS BE RESOLVED?

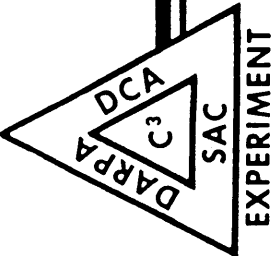
EXHIBIT E-10

RECONSTITUTION OF MULTIPLE PARTITIONED NETWORKS



- COMPLEX RECONSTITUTION ROUTES
EXHIBIT E-11

MULTIPLE PARTITIONED NETWORKS



- THE GENERAL PROBLEM TO BE SOLVED FOR SURVIVABLE, SELF-RECONSTITUTING COMMUNICATIONS

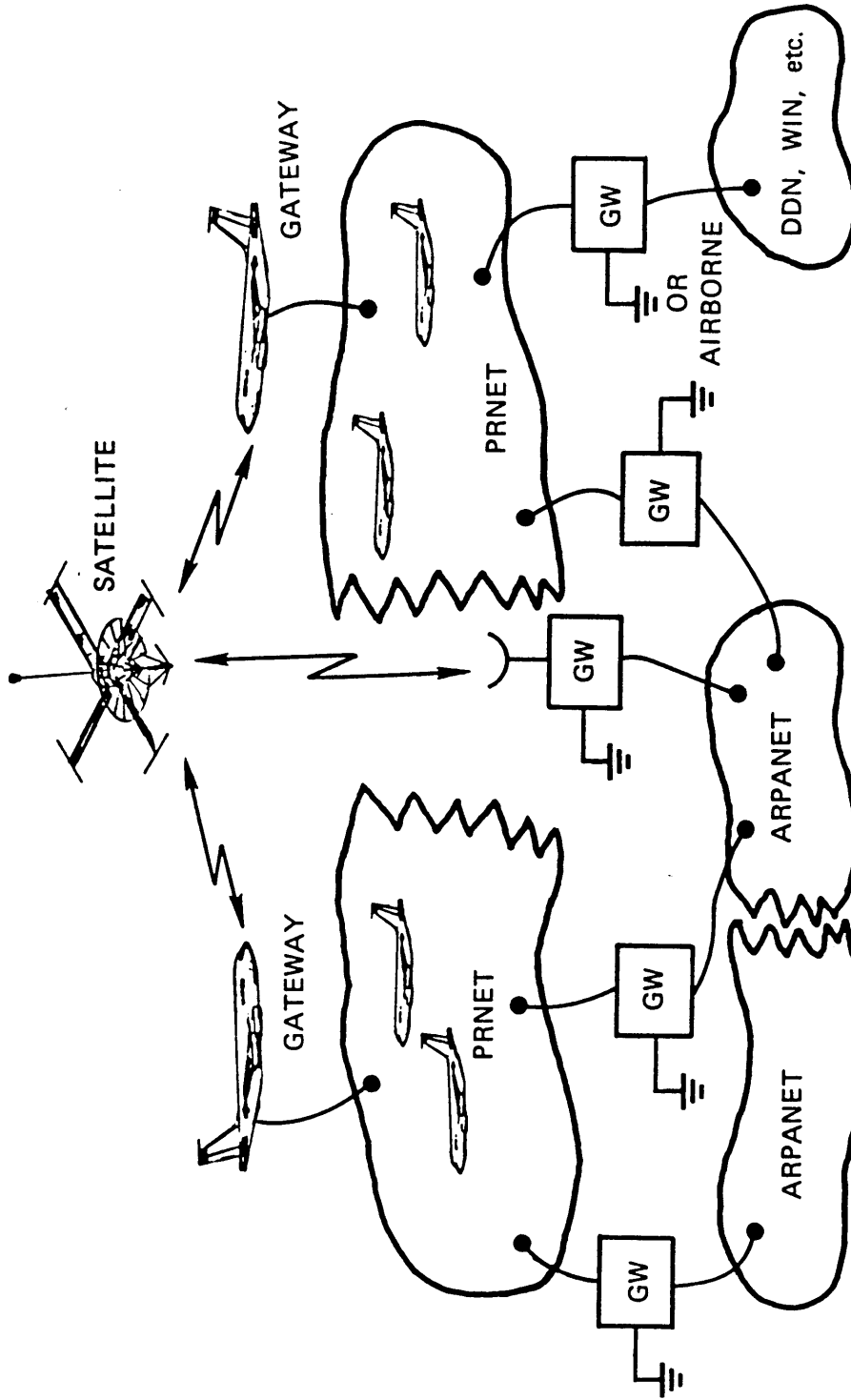


EXHIBIT E-12

**Merit:
Michigan's
Universities'
Computer
Network**

by

Eric M. Aupperle

January 1986

Preface

The Merit Computer Network Project began late in 1969 with the objective of linking several of Michigan's public university computing centers together in a resource sharing data communications network. Merit first provided operational service in 1972 and continued development of new services and capabilities over the ensuing years. Merit operated exclusively as an interuniversity network throughout the 1970's. In the 1980's Merit's networking technology was selected first by The University of Michigan and subsequently by Wayne State and Western Michigan Universities to serve as elements of these universities's internal data networks.

Within the University of Michigan this network is known as UMnet. UMnet serves all three of Michigan's campuses, Ann Arbor, Dearborn and Flint. Much of Merit's recent expansion results from the UMnet component. This was made possible by merging the U-M's Computing Center Data Communications staff with Merit's staff. This marriage produced the rapid developments in both network related hardware and software during the last four year period.

Wayne State and Western Michigan Universities respectively adopted the names WSUnet and WMUnet for intrauniversity implementations which include Merit's technology. Both buy their Merit network hardware from the University of Michigan's Computing Center where the resources exist to fabricate and assemble this equipment. WSU is implementing a WSUnet access ring around the city of Detroit to serve its suburban students and faculty. WMU provides service to its extension students with its Grand Rapids node.

Within this report the name Merit is commonly used to reference network components even though sometimes UMnet, WMUnet or WSUnet could alternatively be mentioned. This is done to simplify the narrative. It is important to recognize that while Merit/UMnet/WMUnet/WSUnet is an integrated network; its inter and intra university manifestations are separately funded and administered.

The following pages show an outline map of Michigan detailing the intercity network links connecting Merit's major switching nodes, links to other networks and remote to Michigan sites, and Michigan access sites. Merit's member universities are:

Michigan State University

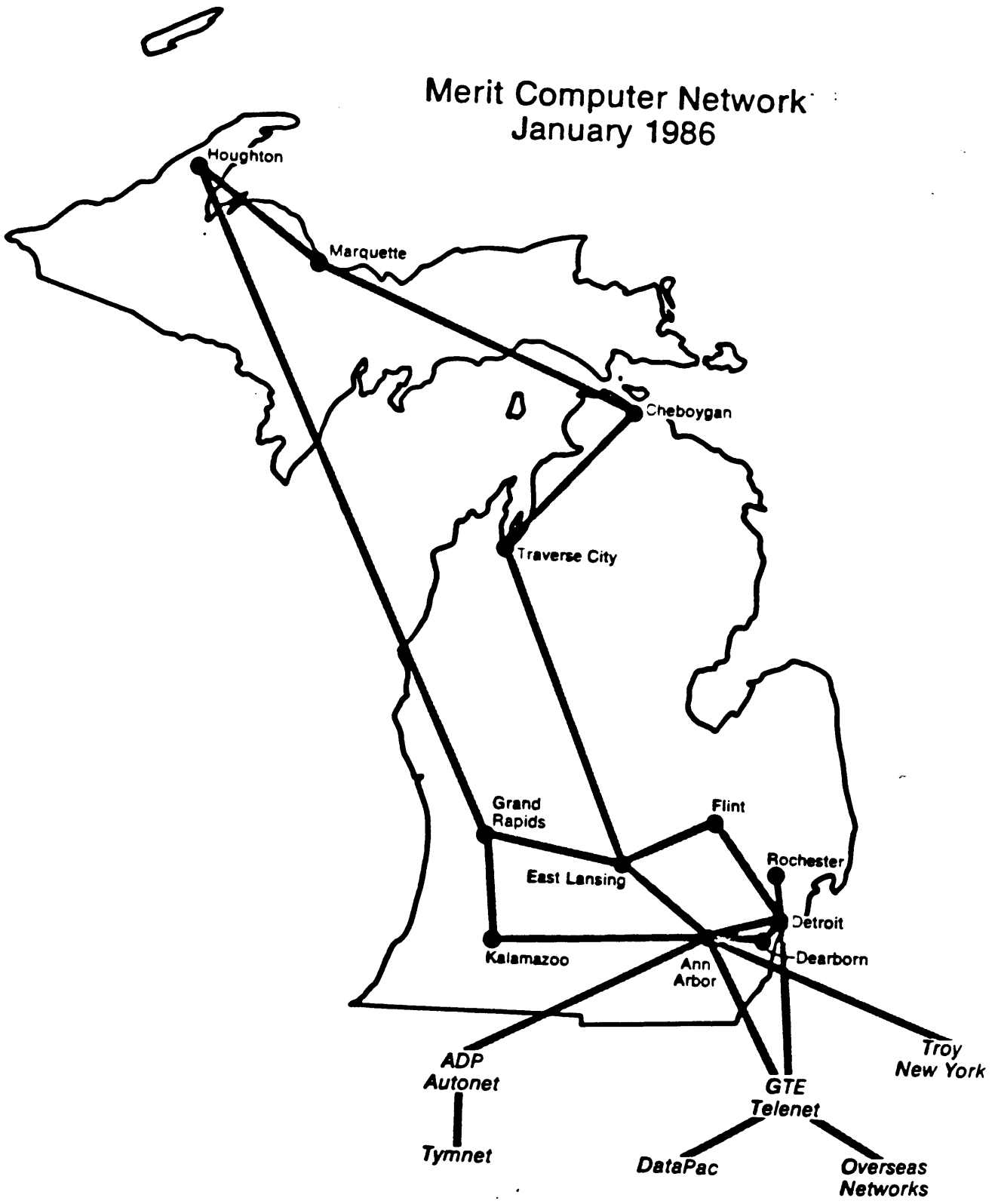
Oakland University

University of Michigan

Western Michigan University

Wayne State University

Merit Computer Network January 1986



Cities with Access Numbers to Merit and Affiliated Networks

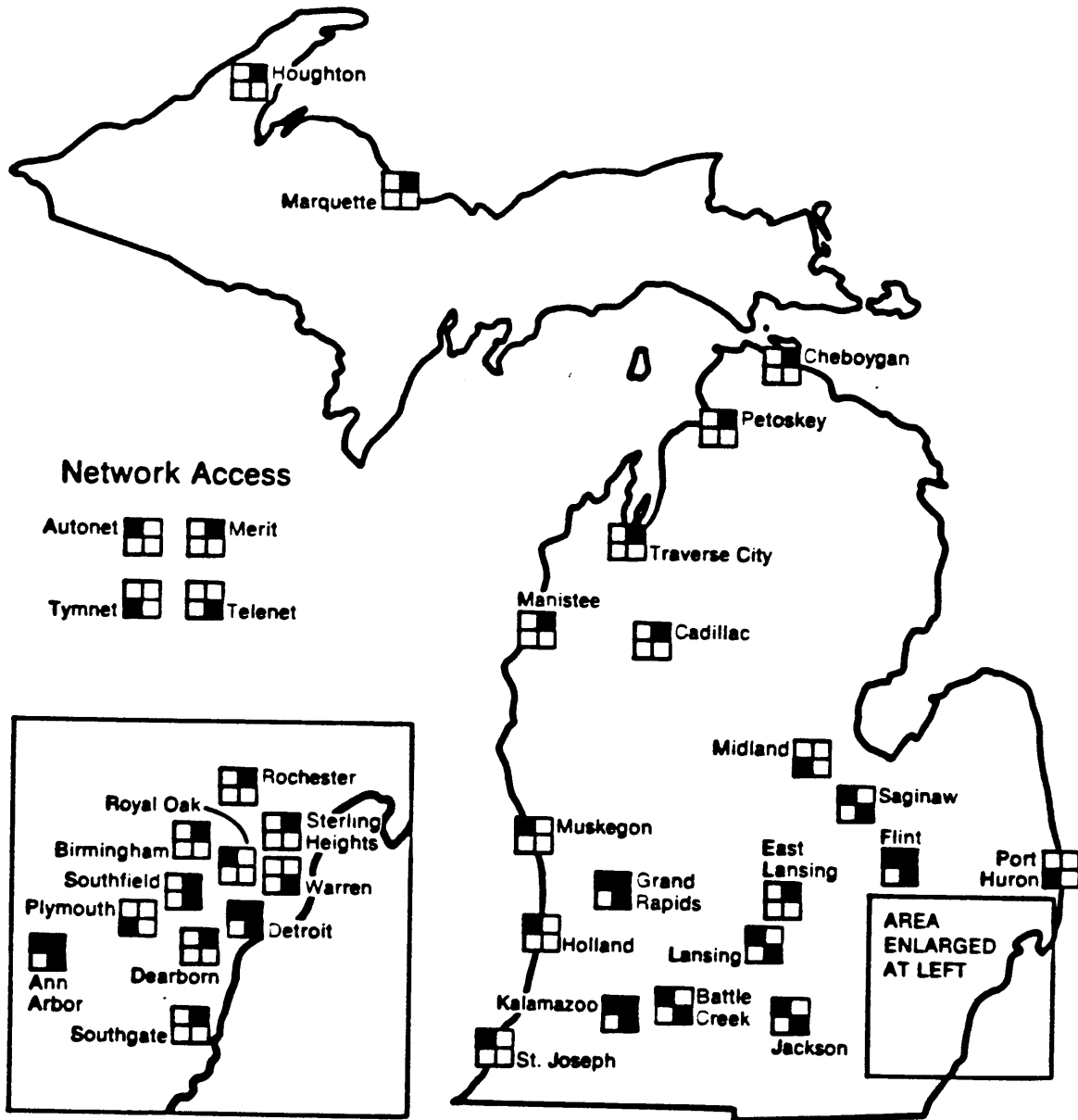


Table of Contents

Introduction	1
Merit's Hardware	11
PCP System Description	11
SCP System Description	15
Internodal Communication Lines	16
Merit's Software	18
Virtual Connections	18
User Hosts	19
Server Hosts	21
Other Hosts	22
System Software	23
Appendix	26
Host Tables	26
PCP/SCP Connection Diagrams	30

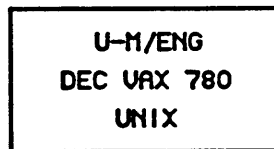
List of Figures

Fig. 1 Simplified Merit Configuration Diagram	3
Fig. 2 PCP/SCP Hierarchchy	5
Fig. 3 SCP Interconnection Services	7
Fig. 4 The ARPAnet Gateway's Interconnection	8
Fig. 5 The Planned USAN Network	9
Fig. 6 The SDSC Consortium	10
Fig. 7 PCP Block Diagram	11
Fig. 8 MM16 Block Diagram	13
Fig. 9 SCP Block Diagram	15
Fig. 10 Virtual Connection Illustration	18
Fig. 11 MCP and Related Support	20
Fig. 12 Some Host/Node System Software Components	23
Fig. 13 CCOS Software Block Diagram	25

Introduction

This report describes Merit's implementation and the current configuration of the network. Hopefully the reader will have a better understanding of such things as PCPs, SCPs, hosts, Hermes, X.25 and many other network related terms and concepts after reading this tutorial. It begins with an overview of the current system diagram and uses this to introduce several concepts. From these beginnings, various details and other topics emerge.

In part, the network exists to interconnect terminals or workstations to hosts and to interconnect hosts and workstations with each other. Hosts are computing systems which provide such services as alternative programming languages, text processors, various editors, a file system and data base systems. Usually a host is specified by its hardware and operating system; for example, a DEC VAX 780 running UNIX or an Amdahl 5860 running MTS. In the configuration diagram, hosts appear as boxes. The first line of each box identifies a host's general location, the second its hardware and the third its operating system as shown in the following example.



Many of the hosts attach to the network's Primary Communication Processors, commonly identified as PCPs. PCPs are Merit's switching nodes and are described in greater detail in the next section. The configuration diagram identifies them with the following symbol. The two letters in the second line represent the PCP's network name, e.g., EL is the PCP at Michigan State University located in East Lansing.



Hosts are attached to network nodes in four ways. Two of these are by a high speed, parallel channel interface, i.e., similar to the way disks or magnetic tape drives connect to computers or by a serial X.25 communications link, e.g., over a

dedicated telephone line. The former requires a PCP to be located near its host, usually within a few meters. The latter has no distance limits, is less costly but slower. All of the large hosts operated by the Merit university computing centers use a channel interface. Most minicomputer hosts use an X.25 link. The other two ways to attach hosts will be explained soon.

Since the network exists to interconnect workstations and hosts, the PCPs must be interconnected. Telephone circuits rented from AT&T, Michigan Bell or our own twisted pair wires provide this service. Within the U-M's Ann Arbor campus some of the links operate on coaxial cables to transmit the network's data more rapidly. Later fiber optic tubes and microwave links between Ann Arbor, Flint and Dearborn may be used for the same purpose too.

Figure 1 is a simplified diagram of the current Merit configuration. It's simplified in the sense that it omits showing how most terminals and workstations are connected and in some other details too. Even so, this figure reveals a great deal about the network's backbone and some of its hosts. It shows the network linking sixteen hosts through eighteen PCPs and serving Ann Arbor, Cheboygan, Dearborn, Detroit, East Lansing, Flint, Grand Rapids, Houghton, Kalamazoo, Marquette, and Traverse City. Later we will learn other cities and hosts also are served by the network. All the identified hosts may be accessed from these Michigan cities directly through Merit.

Observe that this configuration diagram uses line widths and shadings to show the connection between a host and its associated PCP, and for the inter-PCP links. The wide solid lines signify channel-attached hosts. The X.25 attached hosts use wide patterned lines while the inter-PCP links appear as narrow solid lines.

Another feature of this diagram is the presence of the GTE Telenet and ADP Autonet networks. Our network interconnects with both these nationwide commercial systems. Merit dually links with GTE Telenet through Ann Arbor and Detroit based PCPs and connects with ADP's Autonet on a different Ann Arbor PCP. These commercial networks afford access to Merit and its hosts from all around our country or beyond, and workstations on Merit may access hosts on either of these systems or yet other hosts on networks linked with them in an expanding worldwide computer communications system.

In addition to hosts, other networks and Merit's own PCPs, Figure 1 shows two Apollo rings. Apollos are powerful workstations with excellent graphics facilities. These workstations function most effectively when several are interconnected in a ring, in a baseband local area network. The U-M's College of Engineering provides its students and faculty with two such rings, one on the North Campus and the other on the Central Campus. Each of these rings uses an X.25 connection to link with Merit. MSU's Computer Laboratory installed a Contel coaxial cable network to serve its users; this also connects to Merit with an X.25 link. Soon other local area networks, LANs, will interconnect with Merit too.

One final point to make about this diagram is the PCP naming convention. Names like FL for Flint, KZ for Kalamazoo, and MQ for Marquette seem obvious. So is AN for Ann Arbor. They are either the first or only PCPs in these cities. Ann Arbor has several newer ones; they require names too for the network's data routing to work properly. The AB, AD and AE names stem from the U-M's Data Concentrators which these PCPs replaced. An AA PCP exists too; it currently acts as a network software testing system. The one remaining Data Concentrator will become AC after its conversion to a PCP. It follows that Wayne State University's newer PCP's be named DA and DB. CN's name derives from the CIPRNET DEC VAX cluster it serves. This leaves only U-M Dearborn's OH PCP name for the reader to speculate about.

Now that hosts, PCPs and other networks are clearly in mind what about the terminals and workstations? Some connect to PCPs but most attach to Secondary Communication Processors, the SCPs. SCPs are smaller versions of PCPs and are primarily used to connect clusters of terminals or workstations, e.g., personal computers, to Merit. SCPs may also be used to support serial printers, provide local X.25 ports, attach hosts through asynchronous ports, and link LANs. These concepts will be clearer after Figure 2 is explained.

Figure 2 complements Figure 1 by showing the hierarchical relationship of the network's one hundred plus SCPs with the PCP backbone. Actually each SCP has an individual link to its PCP but liberties were taken here to minimize these details. Figure 2 shows the network's other hosts and equipment, e.g., printers, serviced by the SCPs too. Note some hosts and the Apollo rings are connected both to PCPs and SCPs. By mentally superimposing

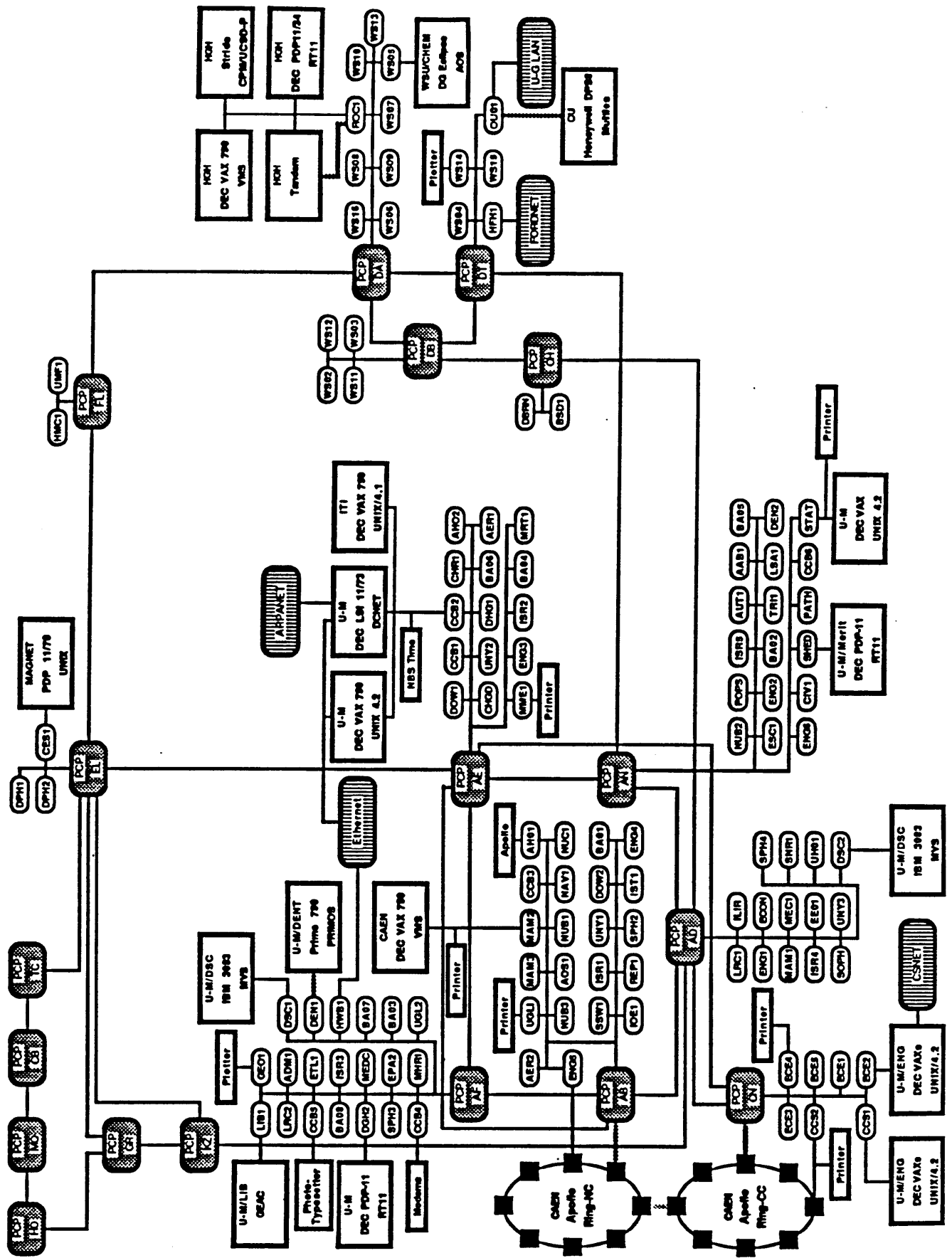


Fig. 2 The PCP/SCP Hierarchy

Figures 1 and 2 one may form a picture of the entire network.

Secondary Communication Processors are physically smaller than PCPs, use less powerful computers and cost less. Each SCP connects to a PCP through a serial communication link of the same type used between PCPs. As the PCPs, the SCPs need names in order for the network to correctly route data traffic. SCPs are given four character names like UNY1 and ENG4. Usually these names reflect either the SCP's location or its owner.

Each SCP may support up to eighty-eight terminals or workstations at data rates as high as 19.2 kbps. Few SCPs are fully configured; more typically each has between twenty and thirty terminals attached. Today the network has over 120 operational SCPs. The majority of the SCPs reside in Ann Arbor and form the dominant part of UMnet as do the SCPs in Flint and Dearborn. The other concentration of SCPs occurs in Detroit. Wayne State University owns most of these units as part of its emerging WSUnet. Recently units of the State's government have purchased SCPs too.

While SCPs primarily support directly attached terminals or workstations, an SCP port can also attach to a serial printer and have output routed to it from elsewhere in the network. Several printers already are attached to SCPs as indicated in Figure 2. Figure 2 also shows many hosts attached to various SCPs. This represents the third way of connecting a host with the network; a method known as asynchronous host support. This method connects several of an SCP's terminal ports to the similar input ports of a host. The several SCP ports assigned to an asynchronously attached host are treated as a group by the network and appear as one host name, e.g., DSC or UMLIB. Whenever a user tries to open a connection to such a host, the local SCP selects any free port in this group for it. This method of host attachment is very easy for hosts and hence is quite popular even though it is inefficient and slow relative to the other two methods. The network already supports 38 hosts through such interfaces as detailed in Figure 2. Figure 3 illustrates the full range of SCP services, including the fourth way to attach hosts by an Ethernet LAN.

Some of Merit's external network connections were described earlier but there are others of growing importance. WSU's Computer Services Center provides access to BITNET through its IBM 3081

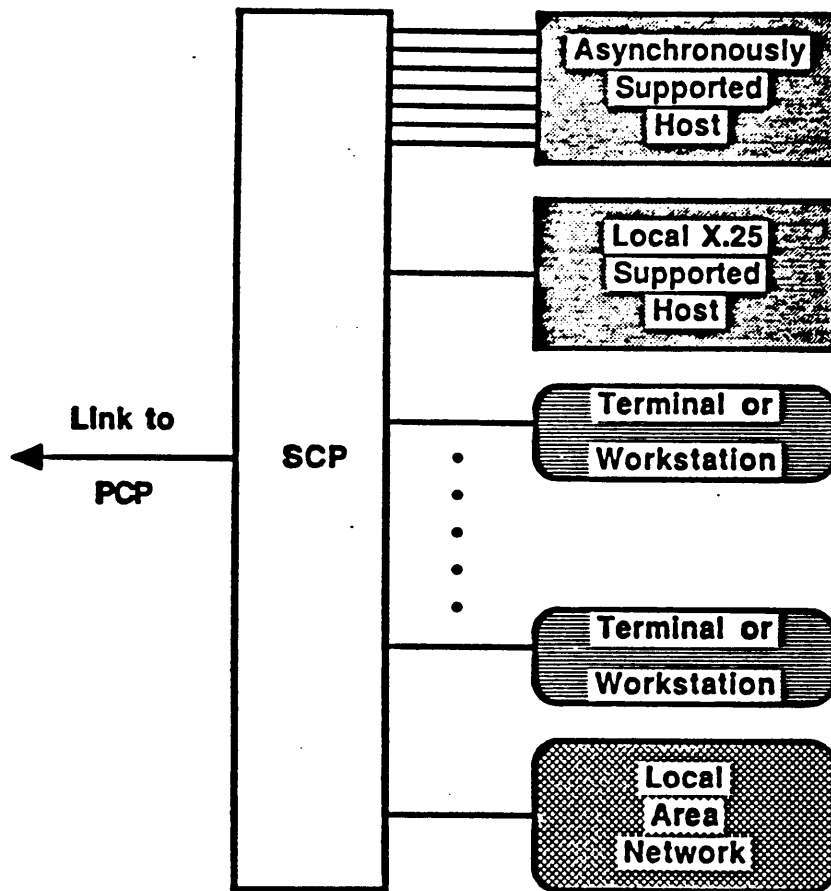


Fig. 3 SCP Interconnection Service

host, see Figure 1. The U-M's Electrical Engineering and Computer Science department operates a CSNET link from their DEC VAX cluster as Figure 2 indicates. Both these networks are of national importance within the university community.

A venerable, important, and famous network is the ARPAnet operated by the U.S. government. Merit links with it through a gateway processor jointly financed by the U-M's College of Engineering and the U-M Computing Center. The gateway consists of a DEC PDP 11/73 system running DCnet software from Linkabit. This gateway is accessible both as an asynchronous host on Merit and

through its Ethernet interface as shown in Figure 4. Currently a 9.6 kbps link connects the gateway to a similar system at Linkabit's office in Vienna, Virginia and from there a direct ARPA IMP (an IMP is like a Merit PCP) connection over a 56 kbps circuit completes this path.

The further significance of the Ethernet shown in Figure 4 is that it will soon serve as an important element in Merit's NSFnet connections. Satellite links to the USAN experiment and San Diego Supercomputer Center are expected early in 1986. Figures 5 and 6 give additional details.

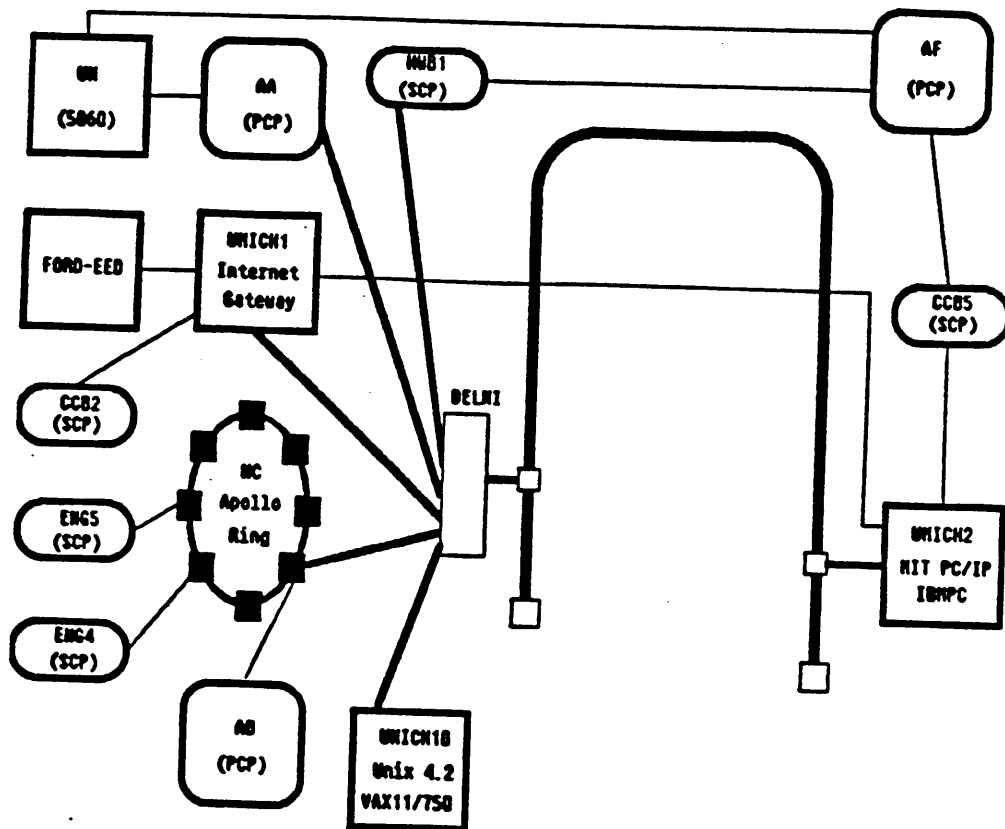
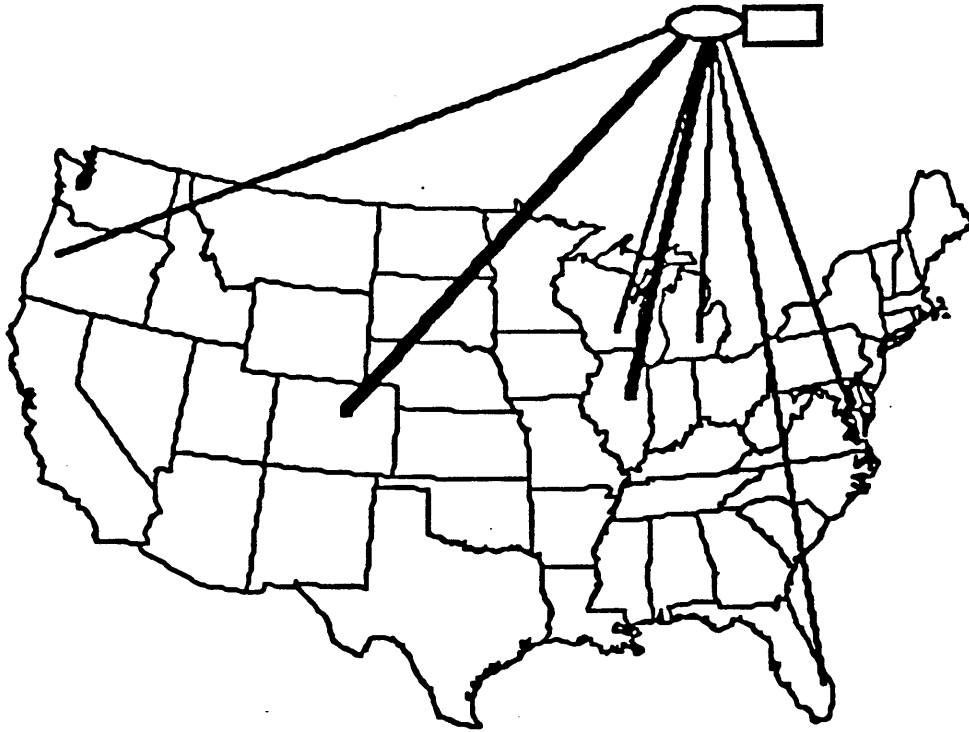


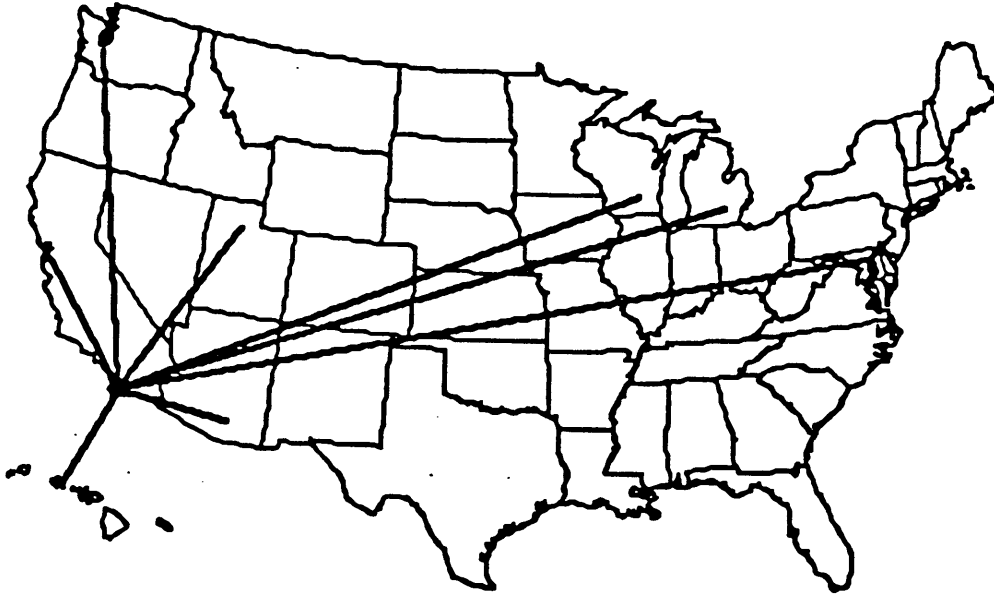
Fig. 4 The ARPAnet Gateway's Interconnection

This concludes the overview. The next section discusses the network's hardware in more detail and following that is a description of the network's software from both a user's and a system's viewpoint. This report ends with an Appendix diagramming each PCP's links and contains a listing of all the network's hosts.



NCAR, Boulder, Colorado
Oregon State University, Corvallis, Oregon
University of Illinois, Urbana, Illinois
University of Maryland, College Park, Maryland
University of Miami, Miami, Florida
University of Michigan, Ann Arbor, Michigan
University of Wisconsin, Madison, Wisconsin

Fig. 5 The Planned USAN Network



- Agouron Institute, La Jolla, California
- California Institute of Technology, Pasadena, California
- National Optical Astronomy Observatories, Tucson, Arizona
- Research Institute of Scripps Clinic, La Jolla, California
- Salk Institute for Biological Studies, San Diego, California
- San Diego State University, San Diego, California
- Scripps Institute of Oceanography, La Jolla, California
- Southwest Fisheries Center, La Jolla, California
- Stanford University, Stanford, California
- University of California -- Berkeley, Berkeley, California
- University of California -- Los Angeles, Los Angeles, California
- University of California -- San Diego, La Jolla, California
- University of California -- San Francisco, San Francisco, California
- University of Hawaii, Honolulu, Hawaii
- University of Maryland, College Park, Maryland
- University of Michigan, Ann Arbor, Michigan
- University of Utah, Salt Lake City, Utah
- University of Washington, Seattle, Washington
- University of Wisconsin, Madison, Wisconsin

Fig. 6 The SDSC Consortium

Merit's Hardware

The network's hardware primarily consists of PCPs, SCPs and the communication channels which interlink these nodes. This section describes the PCP and SCP architecture and identifies the names of their key components. An overview of the interconnecting communication channels in current use appears too.

Both PCPs and SCPs incorporate Digital Equipment Corporation, DEC for short, central processing units. The PCPs use DEC minicomputers, i.e., the PDP 11/34 or PDP 11/60 processors. The SCPs are based on DEC microcomputers, now usually PDP 11/23s and PDP 11/73s. Most PCPs and SCPs contain 128k 16-bit words of memory. Both PCPs and SCPs make use of DEC's memory management hardware. Neither PCPs nor SCPs rely on disks or any other form of local permanent memory except for a small ROM used for loading, dumping and diagnostic analysis.

PCP System Description

A typical PCP consists of the following five major functional system components. A processor, e.g., a PDP 11/34, both synchronous and asynchronous line adapters, a host interface, and a timer.

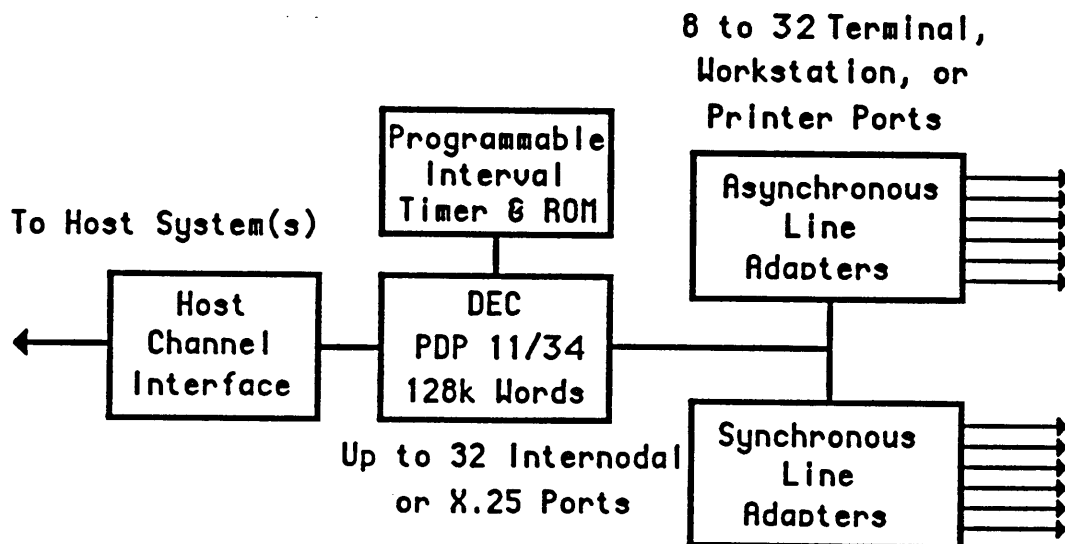


Fig. 7 PCP Block Diagram

The four devices interfaced with the processor each have special functions. The Asynchronous Line Adapters provide the communication ports to serve individual terminals or workstations. Typically these ports may operate at several different data rates to accommodate the needs of the terminal equipment. The maximum rate is either 9.6 kbps or 19.2 kbps depending on the specific hardware used, i.e., commercial DZ or DL equipment, or our own LA32 hardware. These latter 32 port asynchronous Line Adapters are considered obsolete and are being phased out of operation. The long term plan is to have most, if not all, of the asynchronous support provided by the SCPs.

Most of the PCPs' asynchronous line adapter ports are connected to modems for dial-up access to the network so most of these ports actually operate at either 300, 1200, or 2400 bps. The 300 bps ports also support 110 and 150 bps rates using an automatic baud rate selection mechanism. The Asynchronous Line Adapter equipment is the hardware used to provide Merit's Hermes terminal support. Most of Hermes's functionality is derived from software; this is explained in the next section.

The timer unit is really three independent devices, a Programmable Interval Timer, a Diagnostic Control Panel and a ROM unit. This combination device, designed and built by the network's staff, serves the following functions. As a timer it provides crystal controlled time intervals for the PCPs software needs. These needs include time-of-day calculations and the many timer functions needed to support the network's various communication protocols. The control panel allows the network's engineers and programmers to examine or alter memory and input/output interface register locations, to monitor the processor's system bus and to initiate processor interrupts for test purposes. The ROM unit stores several short programs for loading or dumping the PCP from either its host or over the network, and for diagnostic work when the PCP has crashed or is otherwise being tested.

The host channel interface allows communication of commands, status information and data between a host, e.g., an Amdahl 5860, and a PCP. The data exchange at very high rates through parallel, direct memory access transfers. Each type of host requires its own special channel interface. The interfaces used on IBM or Amdahl hosts were designed and are built by the network's staff. MSU's

channel interface to its CDC 750 is a remnant of Merit's original network hardware contract let in 1970. The WMU DEC 1099 interface and the CIPRNET VAX interfaces are commercial units, a DTE-20 and DA-11BJ respectively. Each of these devices requires its own special support software in the PCP. This software is known as the Rare code because it is not common to all PCPs.

While most PCPs feature one host interface, more than one may be supported by both the network's hardware and software. WSU's DT PCP demonstrates this case; it has two, one to the WU host and the second to the WS host. Alternatively, a PCP may not have a host interface, e.g., the FL PCP at UM-Flint. The presence of host channel interfaces exemplifies one of Merit's unique features relative to other packet-switched networks.

The synchronous line adapters, SDAs, provide the network's internodal links, the links to the SCPs, and the X.25 port links. Merit's SDAs are known as MM16s, short for Microprocessor Multiplexor 16s. The MM16 technology was jointly developed by Merit and U-M Computing Center staff. It consists of a multiplexor which interfaces a PCP's UNIBUS with up to 16 Motorola 6809 microprocessors as detailed in the following diagram.

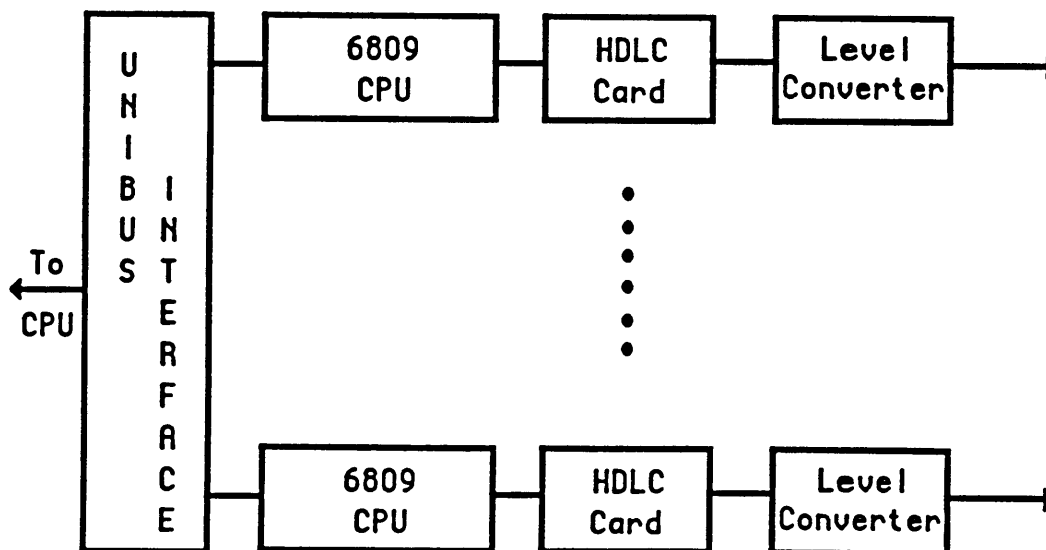


Fig. 8 MM16 Block Diagram

The MM16's multiplexor, labeled UNIBUS Interface in the diagram, serves several functions. These include providing a common address and data interface to the PDP 11's system bus for each of

the up to sixteen microprocessors, prioritizing both interrupt and direct memory access requests from the micros, and permitting the PDP 11's software to collectively or individually enable them.

Each microprocessor system, labeled as a 6809 CPU, is fabricated on its own printed circuit card. This card contains a Motorola 6809 microprocessor, a Motorola DMA controller, both RAM and ROM memory, and essential interfacing circuitry. Three of the DMA's four channels are used. One each to transfer data to and from the HDLC card and the third to transfer data and commands to and from the PDP 11's memory. The 6809's main functions are to support the HDLC chip, manage data transfers between it and the PDP 11, and provide receive data buffers in its local memory. While these may not seem very important, they relieve the PDP 11 from the drudgery of individual synchronous line control. This, in turn, allows the PDP 11's software to concentrate on higher level activities.

The acronym HDLC stands for High-level Data Link Control. This international standard link level communication protocol replaces the older Binary Synchronous protocol made famous by IBM in many newer data communication systems. In Merit elements of HDLC provide the basis for reliable communication between two node pairs. In Figure 8 the block named the HDLC card contains an integrated circuit chip which provides the primary functions required to support the HDLC protocol. This chip has independent transmitter and receiver functions and routinely operates in full-duplex mode.

The last component of the MM16 is a Level Converter card. This card converts the standard TTL integrated circuit level digital signals into those voltages or currents required by various external equipments. There are two versions of this card. The most commonly used one is an RS-232 converter which permits interconnections with the typical modems used in the network. An RS-449 converter also was developed and used in selected cases.

The MM16's modular system design allows for various applications. Its interchangeable level converter serves only as a simple example of this concept. Since providing synchronous ports for the network represents the sole operational use of the MM16s, this section omits further comments about its modularity. Lastly, the MM16 system design permits individual port data rates in excess of one megabit per second with appropriate level converters.

SCP System Description

The SCPs differ from PCPs in several important respects. They are physically the size of a small bread box rather than the PCP's nearly two meter high cabinet. They use a PDP 11/73 Q-bus based processor instead of a PDP 11/34, and SCPs primarily contain commercial hardware. A typical SCP consists of the following major components, one or more asynchronous line adapters, a DEC PDP 11/73 processor, and a synchronous line adapter.

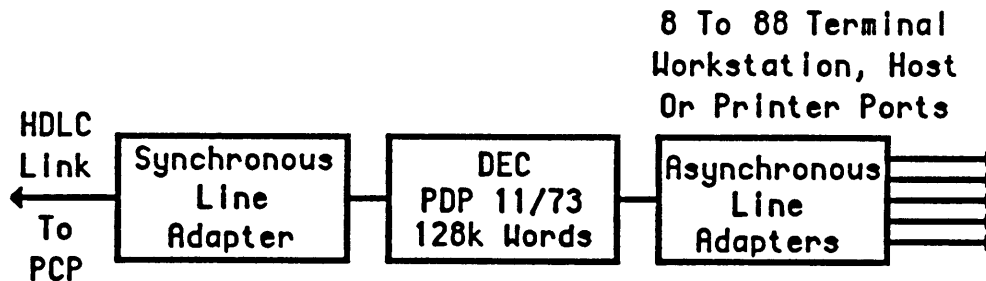


Fig. 9 SCP Block Diagram

An SCP's asynchronous line adapters serve the same functions as those described for the PCP. The primary difference is that all this hardware in SCPs is commercial DEC or MDB DZV equipment. The DEC DZV units have four ports per printed circuit card while the MDB cards contain eight ports each. By mixing these units it is possible to assemble SCPs with multiples of four ports up to a maximum of forty. All of these ports may operate at data rates up to 19.2 kbps.

The synchronous line adapter is the SCP's equivalent of the PCP's MM16. It is named a KHV after its designer, Keith Heron from the University of New Castle. Like the MM16 it supports the HDLC protocol and provides direct memory data transfers between it and the PDP 11/73. Unlike the MM16, the KHV uses no microprocessor and only supports one KHV port per unit. Each SCP uses one KHV to link to one of its PCP's MM16 ports. SCPs may be assembled with more than one KHV by sacrificing asynchronous ports to provide, for example, a local synchronous X.25 port.

The only non-commercial hardware in SCPs is the KHV card and a second one which supports the status lights on the SCP's front panel, a ROM for loading, and an operator's console. These cards, the PDP 11/73, its memory, and the DZV cards are mounted in a small cabinet which contains the necessary power supplies and a line clock. This cabinet constrains the number of asynchronous ports available in an SCP. Its line clock serves the SCP as the Programmable Interval Timer does the PCP.

Internodal Communication Lines

The final portion of this section describes the network's internodal communication links. Between cities Merit and UMnet lease telephone lines from AT&T and Michigan Bell. These companies offer analog and digital circuits at several data rates. The analog lines represent the older technology, have somewhat higher data error rates, but are less costly between some locations. Between major cities, e.g. Detroit, Flint and Lansing, the digital lines are cost effective. All of the network's analog lines operate at 9.6 kbps and use purchased modems. The digital lines terminate in Digital Service Units, DSUs, instead of modems. Some of the network's DSUs are leased but most are purchased. All of the digital lines run at 9.6 kbps except for the Ann Arbor to Detroit link which operates at 56 kbps.

The network's links with both Telenet and Autonet are 4.8 kbps analog circuits. Merit leases these lines and their modems directly from the two companies rather than from the telephone companies. The Autonet line features a dial back-up service which takes over automatically when the permanent circuit fails. The main reason for having two Telenet lines is redundancy. If either of these lines are inoperative, Telenet automatically routes new inbound connections over the functioning link.

Today local internodal links also carry their data over twisted-pair wire circuits. The universities own some of these and other cable pairs are leased from Michigan Bell. The leased ones, known as LADS, Local Area Data Service, channels, are unloaded wire pairs similar to the owned circuits. All these lines employ a different kind of analog, short distance modem and operate at 19.2 kbps, namely Gandalf 309's. Similar lines and modems link the X.25 hosts, e.g., the Prime 750 in ISR. With adjacent or nearby nodes,

as are the several PCPs in the U-M's Computing Center, twisted-pair wires couple them directly, i.e., without modems. The Non-Return-to-Zero-IBM, NRZI, capability of the MM16 hardware allows this to work without the usual modem clocking signals. Within the University of Michigan's Ann Arbor campus, some internodal 56 kbps traffic uses a coaxial cable system.

Merit's Software

While grasping the elements of the network's hardware affords a tangible appreciation for its implementation, the elegance and power of the network comes from its software. It is the software which provides the network's features and services seen by its users, e.g., the Hermes terminal support. Software also controls all the network's hardware devices, reliably routes data through its nodes, monitors its performance, manages memory in the nodes and provides other functions too numerous to detail here. This section offers a general glimpse of Merit's software. A description of the user's view precedes the network's system software overview.

Virtual Connections

While it may at first seem strange, nearly every use of the network involves a connection between a pair of hosts. This is the case whether someone uses Hermes from a terminal to access a host, accesses hosts on Merit through Telenet, copies data between two hosts or sends a job to print at another site. A simplistic view of this appears below.

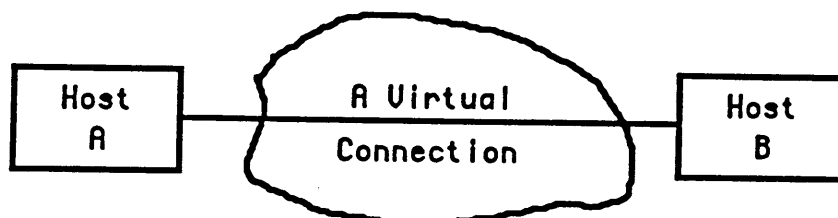


Fig. 10 Virtual Connection Illustration

Here the irregular central object represents the network or possibly even several networks. The technical name for the line connecting these two hosts is a virtual connection or a virtual circuit; it's the path over which data are exchanged between hosts through the network. In Merit, as in other packet-switched networks, a dedicated physical circuit assignment to an individual user never occurs. Rather the user's data pass through the network over physical paths shared with many other users. It is even possible for these paths to change dynamically without the user being aware of any routing switches. Hence the connection is virtual in contrast to the real circuit connections used in telephone systems.

The term packet refers to a quantity of data. For example, a packet may be all of the characters (bytes) a user enters in one line from a terminal. Another example is all of the text on one line of printer output. Within Merit the maximum packet length is 240 bytes. Merit's connections with ADP Autonet, GTE Telenet and our X.25 supported hosts use 128 byte packets to conform to the X.25 standard. On the average about 1000 packets, a kilopacket, are transferred by a typical terminal user in an hour.

User Hosts

A typical user has more interest in a virtual connection's ends than how it threads its way through the network. The most common type connects a terminal user to a serving host. Merit's terminal support software, the user's end of a connection, is named Hermes; it is a user host. Hermes receives the successive characters entered from a terminal through the network's asynchronous hardware and forms them into packets. Usually, Hermes also echos these input characters, i.e., returns each incoming character to the terminal's display or printer. As one line's input characters accumulate, Hermes allows backspacing to effect intraline editing. Once an input line is complete, typically signaled by the user pressing the return key, Hermes forwards this packet to the serving host. In response, the host often returns a packet which Hermes disassembles and then outputs one character at a time to the terminal. This entire process of Packet Assembly and Disassembly is a common characteristic of all packet-switching networks; it's generically called PAD support. Hermes, like all PADs, is a host at one end of a virtual connection.

Hermes also performs many other tasks; a complete description of its device commands appears in Merit's User Memo No. 15. Among these are: tab control; half and full duplex options; flow control using the standard X-ON and X-OFF mechanism when terminals with disks or tapes wish to transfer data into the network; display formatting, e.g., controlling lines-per-page and line width; and right margin processing. Programmable Keyboard Editing, explained in detail in Merit's User Memo No. 21, represents yet another important set of Hermes services. PKE allows a user to assign an arbitrary terminal key or keys to a specific function, for example, to have the Control-C key produce an attention interrupt or the carriage control key to signal an end-of-file. There are default settings of PKE on each of Merit's nodes. The PCPs at MSU and WMU

differ from all the other nodes and each reflects the respective keyboard editing conventions used by these two universities' major host systems. PKE's other principal value is to those intelligent terminals and personal computers which have unusual requirements when interacting with the network.

Another important and unusual Hermes service is the Michigan Communications Protocol, MCP. MCP does two things; it checks and corrects for errors in data transmitted between an intelligent terminal or workstation and the network, and it regulates data flow. This protocol was originally defined and used by the U-M's Computing Center for down-loading cross-assembled object programs on MTS to minicomputers in the early 1970s. MCP support in Hermes appeared in 1980. In a very real sense MCP represents a rudimentary form of packet-like transmission for asynchronous data traffic. Individual characters are still sent one at a time but treated as groups. Each group is checked for correct reception and the transmitting end must resend the entire group if the receiver returns a negative acknowledgement.

Today MCP is primarily used by U-M and WSU to provide IBM 3270-like services on Ontel 1503 terminals and to support several types of personal computers as intelligent terminals. Services like line replacement, windowing, i.e., vertical scrolling on the PCs and both vertical and horizontal on the Ontels, and visual editing on MTS are built on top of the reliable MCP data exchange mechanism. Another important PC service is the ability to exchange files with the MTS hosts over MCP. This is diagrammed here.

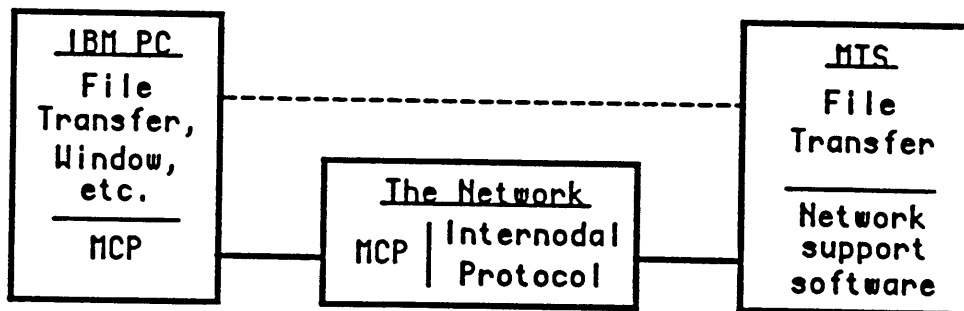


Fig. 11 MCP and Related Support

Figure 11 shows that an MCP implementation exists in both the PC and in the network. The PC terminal support functions, file transfer, windowing and others use MCP to guarantee accurate data exchanges with the network. Once the data are in Hermes they are

formed into regular network packets and routed like all other packets; there is no longer any MCP identity to them. The host's network support software receives these packets and makes them available to the full range of MTS services. Among these services is the MTS end of the file copying software. The dashed line suggests the logical link between the PC's and MTS's respective parts of the file transfer service invoked with the Telecopy command in the PC. Figure 11 also explains why the MCP based services cannot work over Autonet or Telenet since neither of these commercial networks support the MCP protocol in their PADs.

As with Merit's unique channel hardware interfacing to some hosts, the Hermes software has more functionality and capabilities than any other network's PAD support. Hermes' ten year evolution was shaped and refined in an operational network environment characterized by many demanding and differing user viewpoints. The X.3 PAD functions included in the X.25 standard specifications are a relatively small subset of those found in Hermes.

A second form of a virtual connection used extensively comes from one or the other of the two commercial network's PADs into a host on Merit, or the reverse, from Hermes to a host on either ADP Autonet or GTE Telenet. These user host initiated connections are very similar to the first kind except for the X.3 limitations alluded to previously. Merit's User Memo 15 carefully explains which device commands do and don't work through the commercial networks. Considerable effort has been expended by Merit's technical staff to make this indirect form of access as Hermes-like as possible on incoming connections. This has been made even more difficult than necessary by the failure of some foreign network administrations, those with whom Telenet interconnects, to even provide the limited, standard X.3 PAD support.

Server Hosts

The far end of a Hermes virtual connection usually terminates in a serving host, i.e., a host which offers the array of services typically associated with a time-sharing computer. Among these are an editor, a file system, various programs and data bases, and perhaps an electronic mail system. Each host appears to network users in its own unique way. For example, the MTS end of an incoming connection appears as *MSOURCE* and *MSINK* while on MSU's CDC 750 looks like the INPUT and OUTPUT files. As closely as

possible the network resembles a directly attached terminal to each host.

From several of the serving hosts, a user may open a connection to another host. This is possible from either of the MTS hosts or from WMU's TOPS-10 system. These are clearly host-to-host connections in the lay sense. Once opened, these connections allow the user to access remote resources through the local host. One common use of this connection type is to copy files between the hosts. The Merit .COPY protocol provides the basis for this and its MTS user interface documentation appears in Merit's User Memo No. 9. The WMU .COPY interface is similar. MSU does not offer this service directly but does support the .COPY protocol as a remote host.

Lastly, among the MSU, the MTS systems and WSU's MVS computer, interhost network batch and print services are possible. These too are host-to-host services and rely on the network's underlying reliable data transport services. Batch jobs may be originated at any of these hosts and routed to any of the others for execution. Any batch output or any independent print output may be returned or sent between these hosts too. It is also possible to transfer plot files between some hosts to produce remote Calcomp drawings.

Other Hosts

Other, relatively new forms of interactive access are Merit's X.25 PT, Pass Through, and X.25 OB, Out Bound, services. The former allows any X.25 attached host on Merit or any X.25 attached network to directly interconnect and transfer data. All combinations are possible, i.e., host to host, host to or from network, and network to network. In all these cases Merit acts as a transparent carrier of data. The U-M's Physics Department employs this service to communicate directly with hosts on Telenet through their Merit X.25 link. The X.25 OB service allows non X.25 attached Merit hosts, e.g., the MTS systems, to call out through Merit into X.25 attached hosts or networks. The MNET:NET program in MTS uses X.25 OB. Both these services are examples of network gateways.

At best this is but an overview of Merit's user services. Much more information appears in the series of Merit User Memos and in online help files on the major hosts. The remainder of this section provides an introduction to the network's system software.

System Software

System software manages a computer's resources, e.g., its memory and peripheral equipment, and provides basic services like a file system for the user. System software details are rarely understood or thought about by most computer users, nor should they be. This same situation prevails for network system software.

In Merit's case, as in most networks, the system software is distributed among the hosts and nodes. Differences in this distribution differentiate among networks and in this regard Merit exhibits a few unique characteristics. Figure 12 depicts the generic interface between a Merit PCP and a channel-attached host.

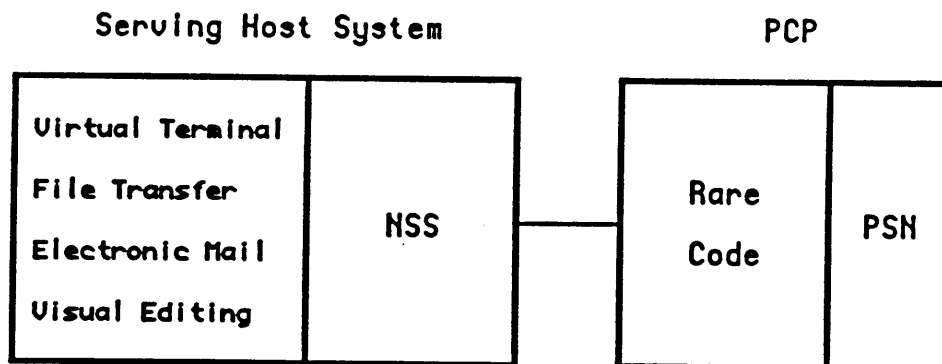


Fig. 12 Some Host/Node System Software Components

As will be explained soon, PCPs contain many system software components. One of them, the Packet-Switching Network software which routes packets through the network exists in all PCPs. In contrast with this common component is the special software needed to interface with a specific host. Each unique host system has its own network Rare Code, i.e., there are rare codes for MTS, TOPS-10, SCOPE and UNIX. An implication of this specialization is that PCPs are not all loaded with completely identical software.

The host's complement of the rare code is the Network Support Software. The NSS software and the Rare code cooperatively control the hardware interface between the PCP and its host, exchange user data for multiple users, translate the host's character codes into the network's standard, and perform many other services too. Within the host's operating system, superimposed on the the NSS, are various higher-level network software support functions. Examples of these include virtual terminal, file transfer, electronic mail exchange, and visual editing support.

Virtual terminal support refers to the ability of any host terminal to open a network connection to a remote host and use the remote host through the network as though the local terminal was directly attached to the remote host. File transfer means the ability to copy data files between hosts and electronic mail exchange represents the ability to send messages between systems. These two services require cooperating processes on the two hosts involved with the transfers, while virtual terminal support only needs an implementation in the user's local host and is an outward directed service. In contrast, visual editing is an inward directed service. It provides full-screen editing services to remote terminal users. These four examples do not constitute a full set of services but they are the common ones. Even so, not all hosts support this set.

The MTS names for the host components may help some readers understand this section better. The network Device Support Routine, more commonly simply the DSR, serves as MTS's NSS. The MTS NET command invokes the virtual terminal service and a file transfer begins with the .COPY command within NET. Interhost mail exchanges are possible using the \$MESSAGES SEND TO Smith@MIT extension. Finally, MTS supports visual editing from Ontel intelligent terminals and from many other MCP supported personal computers too.

Now consider the system software in the PCP in somewhat more detail. First there is an underlying operating system known as CCOS for Communications Computer Operating System. It manages the PCP's memory by allocating buffers on demand and recovering them when they are no longer needed. It schedules tasks, manages the interrupt stack, and provides the mechanism for swapping tasks when they are waiting for other events. In addition, CCOS provides a powerful parser and other fundamental services.

Within this framework exist the system software components shown in Figure 13. This figure portrays the next level of detail of the CCOS software system. Even so, this figure still represents only a gross overview of the intricacies of the network's system software. At the left is the Packet Switching Network portion of CCOS first identified in Figure 12. This portion reveals that these software components include Merit's Internodal Protocol, i.e., the software which supports the PCP-to-PCP and the PCP-to-SCP links. The inter PCP and the intra PCP and SCP support differs in that

SCPs only know about their master PCP while PCPs have knowledge of the entire switching network. Another important PSN function is keeping track of the network's topology, i.e., knowing which nodes and internodal links are operational. This is necessary for the PSN to properly perform packet routing.

The boundary between the PSN and the rest of the software represents the network's hosts in software form. There are several of these PSN/host interfaces. Among them are the channel supported hosts represented by the one or more Host Support Modules in Figure 13. Another is the Network Interface Module, the NIM, which provides the bulk of the Hermes host support. The other two hosts shown in Figure 13 are the Out Bound and Pass Through modules associated with Merit's comprehensive X.25 services.

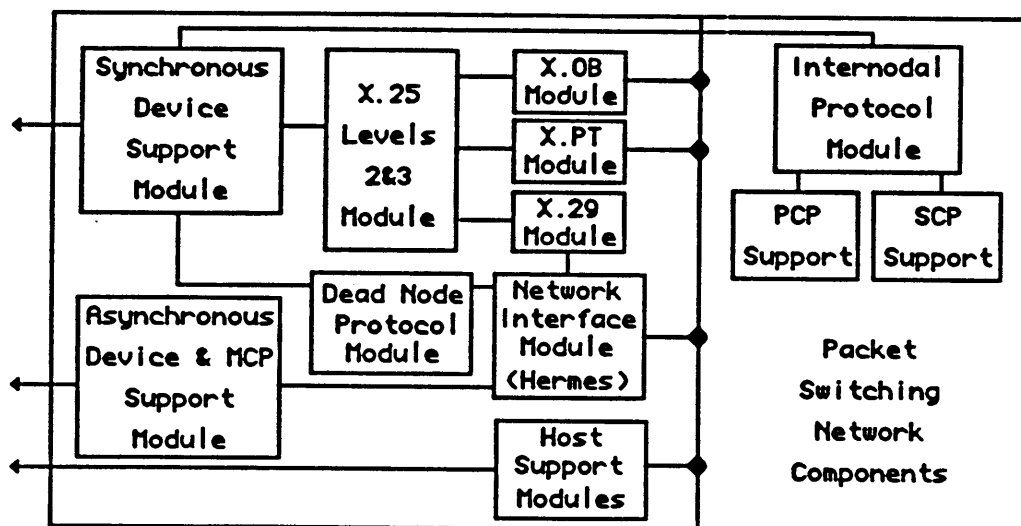


Fig. 13 CCOS Software Block Diagram

Moving to the left in Figure 13 from the host modules are the link (2nd level) and packet (3rd level) X.25 level support. This module and the X.29 module constitutes Merit's comprehensive gateway with X.25 hosts and networks. The Dead Node Protocol module is used to load or dump nodes. Lastly, the Asynchronous and Synchronous modules at the extreme left represent the set of specialize modules tailored to the specific hardware elements described in the last section.

This concludes the overview of Merit's technology. Interested readers are referred to the various Merit User Memos and technical papers for further information.

Appendix

This appendix contains supplemental information about the network's configuration. The following table lists each host and identifies how it is attached. The column labelled Allowed Access refers to whether a host can only open connections to the network, Out Only, only receive them, In Only, or both, Bidirectional.

The series of diagrams after the table shows how the network's nodes are interconnected through the MM16s. In these diagrams, one or two for each of the network's PCPs, the various shapes and shadings signify classes of objects, e.g., SCPs appear as elongated, unshaded ovals and the variously attached hosts as darkly shaded rectangles. Each PCP's network name appears in the big rectangle left of center as do its MM16 port numbers. Note, the inter-PCP links are the partially shaded, rectangles. In these inter-PCP boxes the number represents the MM16 port number in the other, named PCP.

Host Table

<u>Host Owner</u>	<u>CPU Hardware</u>	<u>Operating System</u>	<u>Network Name</u>	<u>Allowed Access</u>	<u>Network PCP(s)</u>
<u>Channel Attached Hosts</u>					
MSU CL	CYBER 750	SCOPE	MS	Bidirectional	EL
U-M CC	Amdahl 5860	MTS	UM	Bidirectional	AB,AD AE,AF AN
U-M CC	Amdahl 470/V8	MTS	UB	Bidirectional	AB,AF AN
WMU ACC	DEC 1099	TOPS-10	WM	Bidirectional	KZ
WSU CSC	Amdahl 470/V8	VM/MTS	WU	Bidirectional	DA,DB DT
WSU CSC	IBM 3081	VM/MVS	WS	Bidirectional Batch Only	DT

Host Table Continued

<u>Host Owner</u>	<u>CPU Hardware</u>	<u>Operating System</u>	<u>Network Name</u>	<u>Allowed Access</u>	<u>Network PCP(s)</u>
-------------------	---------------------	-------------------------	---------------------	-----------------------	-----------------------

X.25 Attached Hosts

CAEN	Apollo Rings	Aegis			
	North Campus		UR	Bidirectional	AB
	Central Campus		LR	Bidirectional	CN
CAEN	Harris 800	VOS	EH	Bidirectional	AB
CRC	DEC VAX 730	VMS	XB@AF	Bidirectional	AF
HGH	Tandem		XA@ROC1	Bidirectional	DA
ISR	Prime 9955	PRIMOS	SR	Out Only	AN
OU OCS	Honeywell DPS8	Multics	OU@OU01	Bidirectional	DB
RPI	IBM 3083	MTS	RP	Bidirectional	AF
U-M Dent.	Prime 750	PRIMOS	DS@DEN1	Out Only	AF
U-M Phys.	DEC VAX/780	VMS	RL	Bidirectional	AN
WMU ACC	DEC VAX/780	VMS	XA@KZ	Bidirectional	KZ
WSU CSC	IBM 3081	VM/CMS	WV	In Only	DA
(WV serves as an interactive path to three WCSC hosts)					
WSU CSC	DEC VAX/780	VMS	XB@DB	Bidirectional	DB
WSU Eng.	Harris 800	VOS	XA@DB	Bidirectional	DB
WSU Eng.	Prime 9950	PRIMOS	Xc@DB	Bidirectional	DB

X.25 Attached Networks

ADP Autonet			TP	Bidirectional	AB
GTE Telenet (Ann Arbor)			TA	Bidirectional	AN
GTE Telenet (Detroit)			TD	Bidirectional	DT
Michigan Bell Net			XA@AF	Bidirectional	AF
MSUnet	Contel LAN		XA@EL	Bidirectional	EL

Host Table Continued

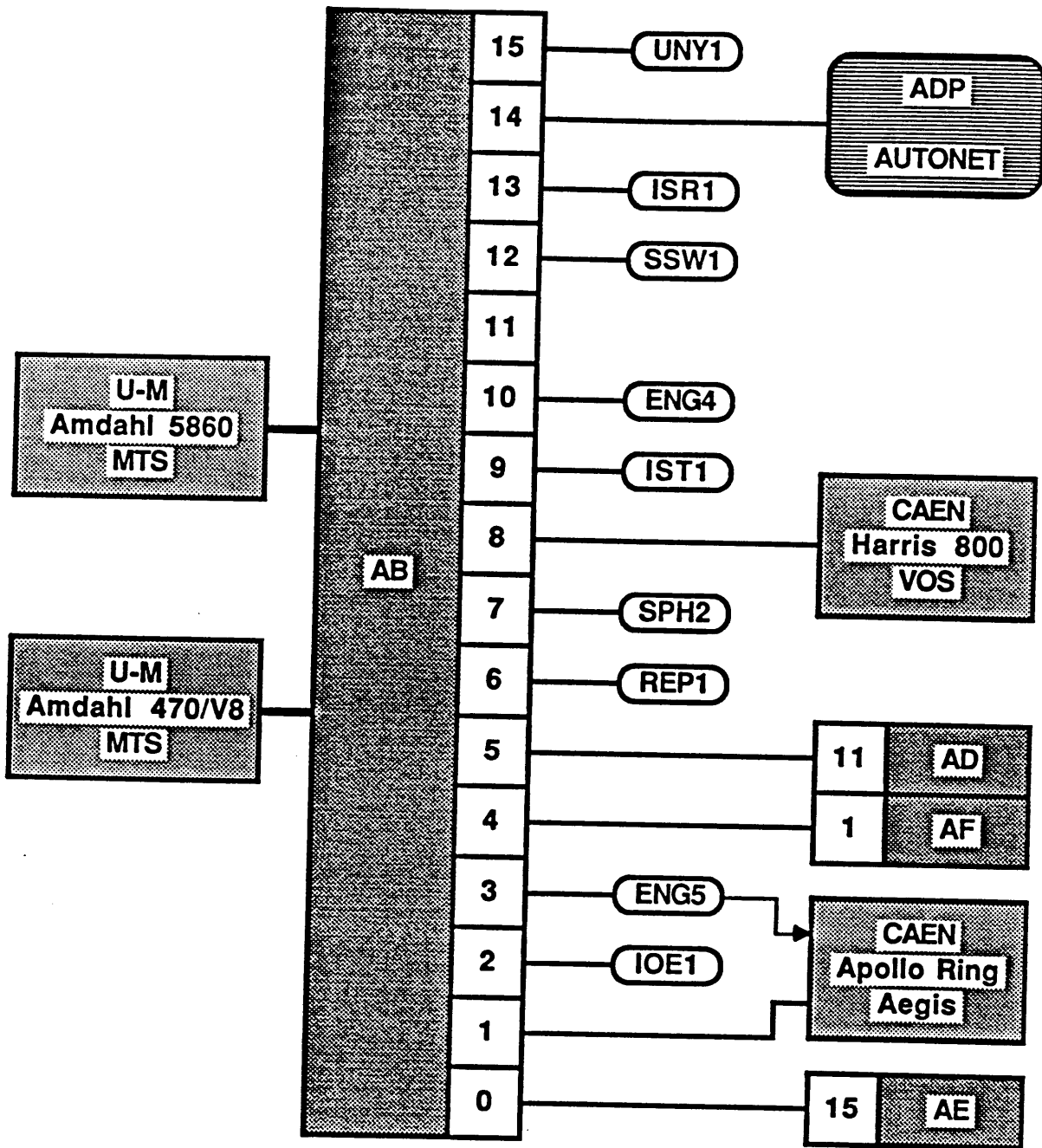
<u>Host Owner</u>	<u>CPU Hardware</u>	<u>Operating System</u>	<u>Network Name</u>	<u>Allowed Access</u>	<u>Port Count</u>	<u>Network PCP(s)</u>
<u>Asynchronously Attached Hosts</u>						
CAEN	Apollo Rings	Aegis				
	North Campus		Apollo@CCB2	Bidirect	1	AE
	North Campus		Apollo@ENG5	In Only	1	AB
	Central Campus		Apollo@CCS2	In Only	1	CN
CAEN	DEC VAX 780	VMS	MMVAX@MAM2	In Only	2	AB
CAEN	Diablo	Printer	Diablo@MAM2	In Only	1	AB
CIPRNET	DEC VAX 780	UNIX/4.2	CAVAX@ECE2	Bidirect	4	CN
	DEC VAX 780	UNIX/4.2	CVVAX@ECE2	Bidirect	3	CN
Harper/ Grace Hosp.	Tandem		RTAND@ROC1	In Only	3	DA
	DEC VAX 750	VMS	VAX@ROC1	In Only	2	DA
	Stride	CPM/UCSD	STR@ROC1	In Only	4	DA
	DEC PDP11/34	RT11	T11@ROC1	In Only	2	DA
Henry Ford Hosp.	Fordnet (a LAN)		NET@HFH1	Bidirect	4	DT
ITI	DEC VAX 750	UNIX/4.1	ITI@CCB2	Bidirect	1	AE
MSU MAG	DEC PDP11/70	UNIX	MAG1@CES1	In Only	16	EL
MTU CC	ISI	LAN	MTU@HO	Bidirect	16	HO
NWMC	DEC LA120	Printer	PRINT@TC	In Only	1	TC
OU Eng.	Ungerman-Bass	LAN	SECS@OU01	In Only	6	DB
U-M CC	DEC 11/73	DCNET	INT@CCB2	Bidirect	8	AE
U-M CC	DEC VAX 750	UNIX/4.2	CCVAX@CCB2	Bidirect	4	AE
U-M CC	DEC PDP11	RT11	PDP@SHED	In Only	1	AN
U-M CC	Dial-Out	Modems	DO300@CCB4	In Only	1	AF
U-M CC	Dial-Out	Modems	DO1200@CCB4	In Only	1	AF
U-M CC	MAPS-5	Typeset	TYPE@CCB5	In Only	1	AF
U-M CC	NBS Time		TIME@CCB2	In Only	1	AE
U-M CC	Xerox 2700	Printer	X2700@UGL1	In Only	1	AB
U-M DSC	IBM 3083	MVS	DSC1@DSC1	In Only	10	AF
U-M DSC	IBM 3083	MVS	DSC1@DSC2	In Only	10	AD
U-M DSC	IBM 3083	MVS	DSCA@DSC1	In Only	1	AF
U-M EECS	DEC LA120	Printer	LA120@CCS2	In Only	1	CN
U-M EECS	NCube		NCUBE@CCS1	In Only	6	CN
U-M EECS	Laserwriter	Printer	EPRINT@ECE4	In Only	1	CN
U-M Eng.	HP Laserjet	Printer	LASER@MME1	In Only	1	AE
U-M Geo.	Zeta	Plotter	ZETA@GEO1	In Only	1	AF
U-M HG	DEC PDP11	RT11	PDP@DHG2	Bidirect	4	AF

Host Table Continued

<u>Host</u> <u>Owner</u>	<u>CPU</u> <u>Hardware</u>	<u>Operating</u> <u>System</u>	<u>Network</u> <u>Name</u>	<u>Allowed</u> <u>Access</u>	<u>Port</u> <u>Count</u>	<u>Network</u> <u>PCP(s)</u>
-----------------------------	-------------------------------	-----------------------------------	-------------------------------	---------------------------------	-----------------------------	---------------------------------

Asynchronously Attached Hosts Continued

U-M Lib.	GEAC 1200 bps		UMLIB@LIB1	In Only	8	AF
U-M Lib.	GEAC 300 bps		LIB300@LIB1	In Only	2	AF
U-M Math R.	Apollo	Aegis	AHAP@AH01	In Only	1	AB
U-M SRL	DEC VAX	UNIX	UNIX@STAT	In Only	7	AN
U-M SRL	HP Laserjet	Printer	LASER@STAT	In Only	1	AN
WSU CSC	Calcomp	Plotter	PLTR@WS14	In Only	1	DT
WSU Chem.	DG Eclipse S-130(1)		LCN1@WS05	In Only	1	DA



PCP Name: AB

PCP Location: U-M Computing Center

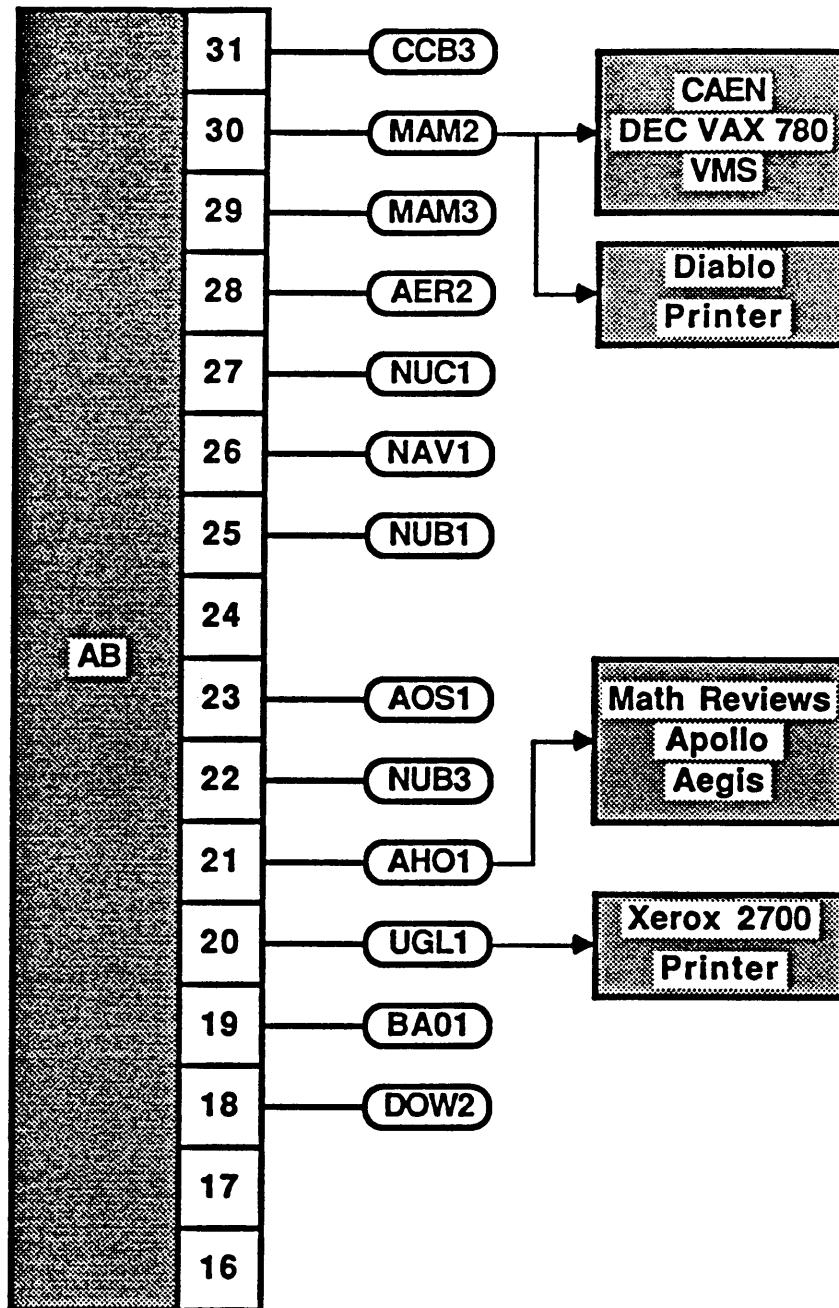
PCP Hardware: PDP 11/34, 2 MM16s, 2 IBM Block Multiplexor Host I/Fs

Hermes Ports: None

Number of SCPs: 22

Number of X.25 Ports: 3

Number of Internodal Links: 3



PCP Name: AB

PCP Location: U-M Computing Center

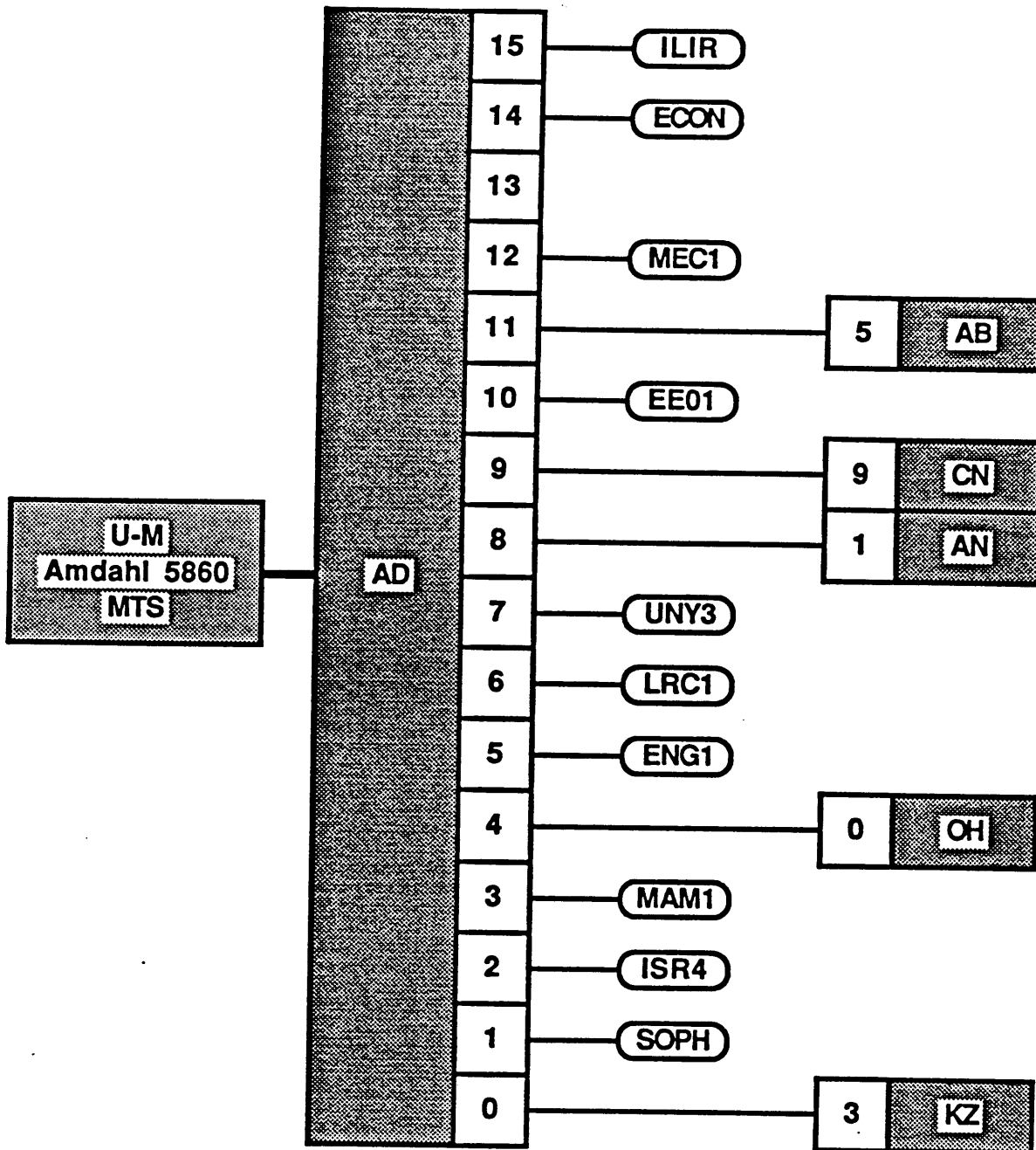
PCP Hardware: PDP 11/34, 2 MM16s, 2 IBM Block Multiplexor Host I/Fs

Hermes Ports: None

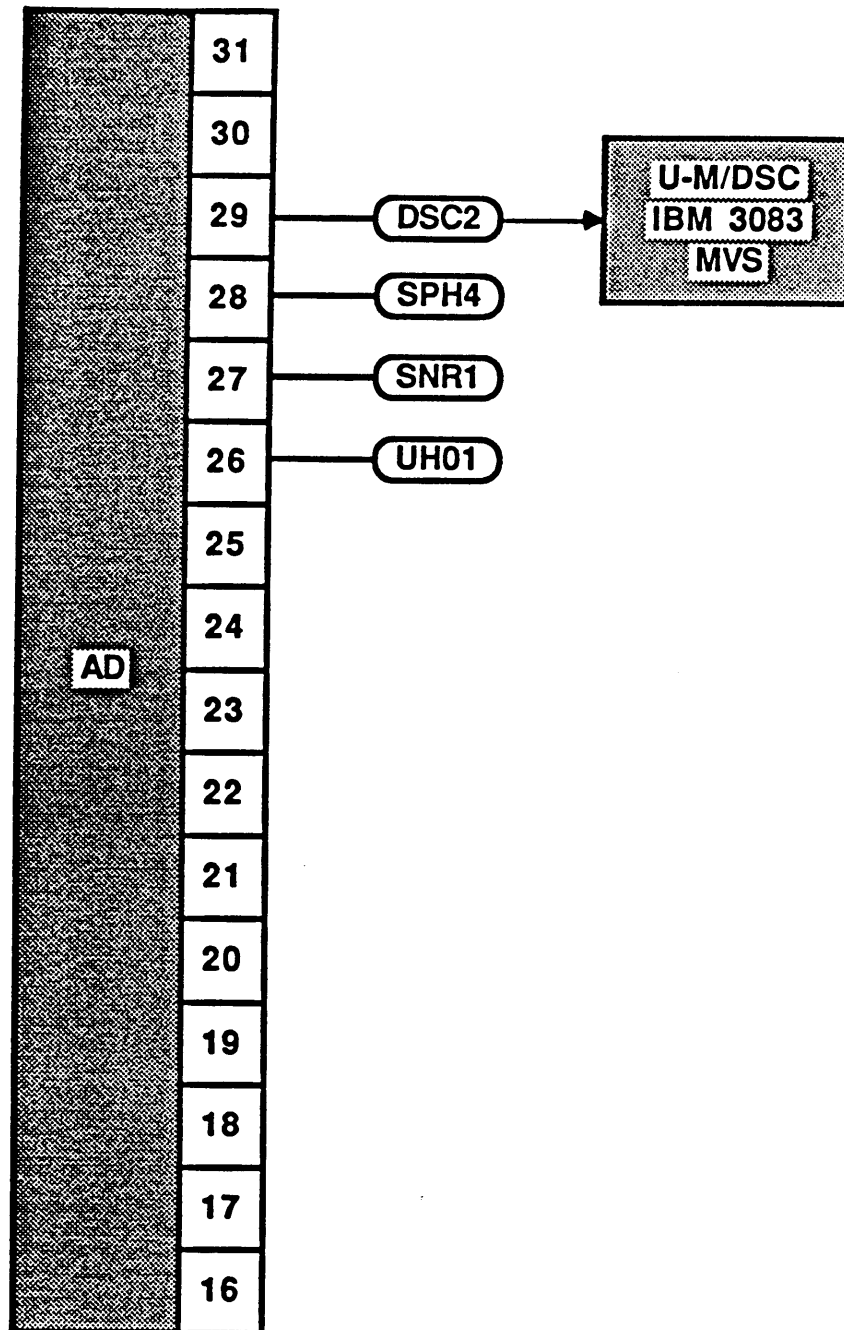
Number of SCPs: 22

Number of X.25 Ports: 3

Number of Internodal Links: 3



PCP Name: AD
 PCP Location: U-M Computing Center
 PCP Hardware: PDP 11/34, 2 MM16, 1 IBM Byte Multiplexor Host I/F
 Hermes Ports: None
 Number of SCPs: 14
 Number of X.25 Ports: None
 Number of Internodal Links: 5



PCP Name: AD

PCP Location: U-M Computing Center

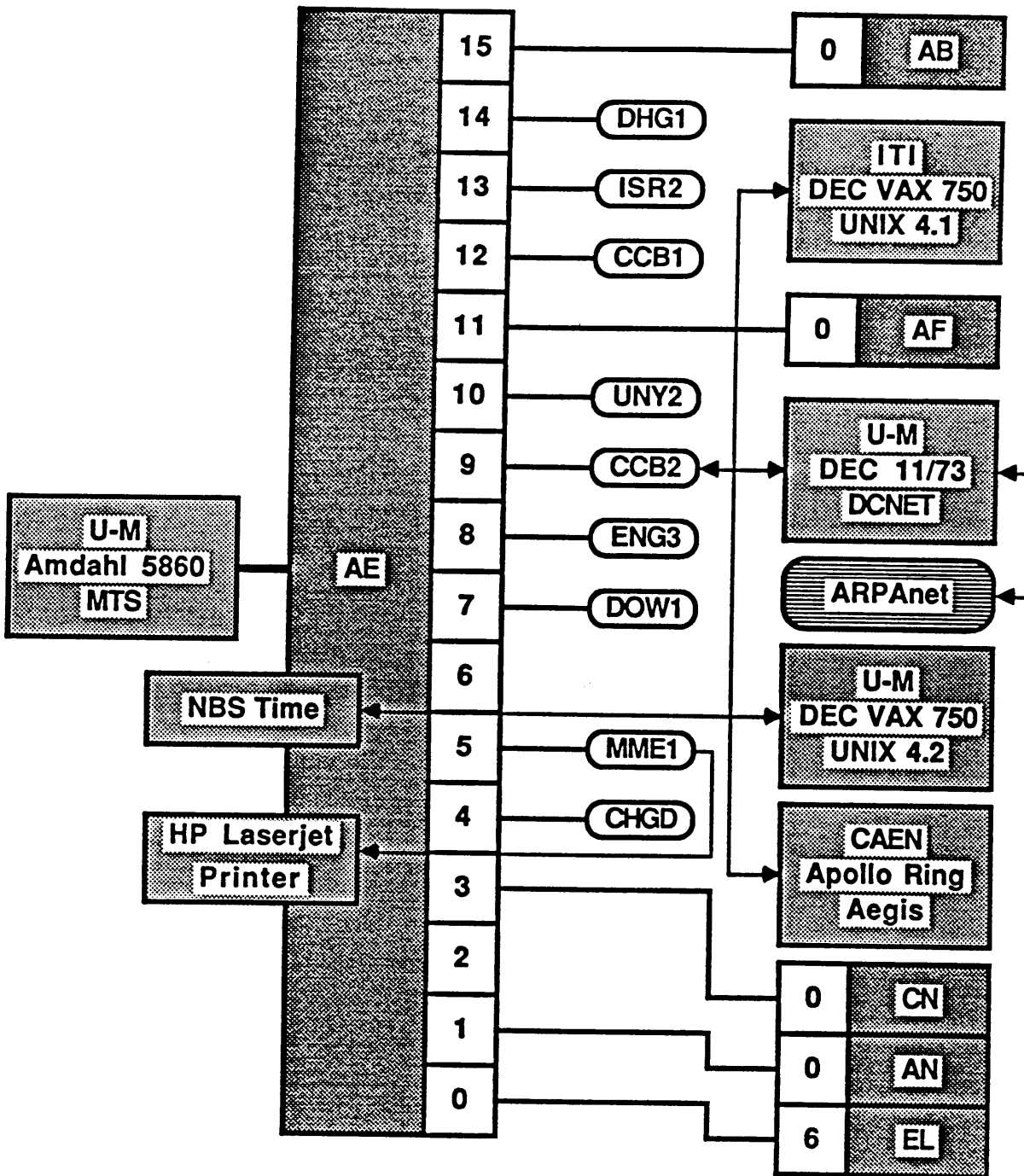
PCP Hardware: PDP 11/34, 2 MM16, 1 IBM Byte Multiplexor Host I/F

Hermes Ports: None

Number of SCPs: 14

Number of X.25 Ports: None

Number of Internodal Links: 5



PCP Name: AE

PCP Location: U-M Computing Center

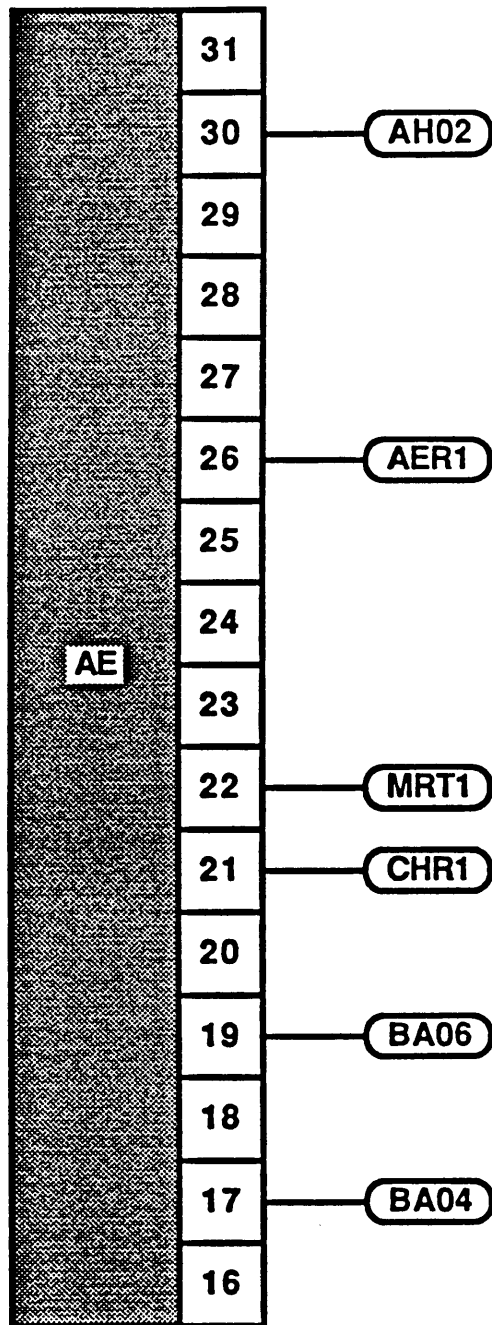
PCP Hardware: PDP 11/34, 2 MM16s, 1 IBM Byte Multiplexor Host I/F

Hermes Ports: None

Number of SCPs: 15

Number of X.25 Ports: None

Number of Internodal Links: 5



PCP Name: AE

PCP Location: U-M Computing Center

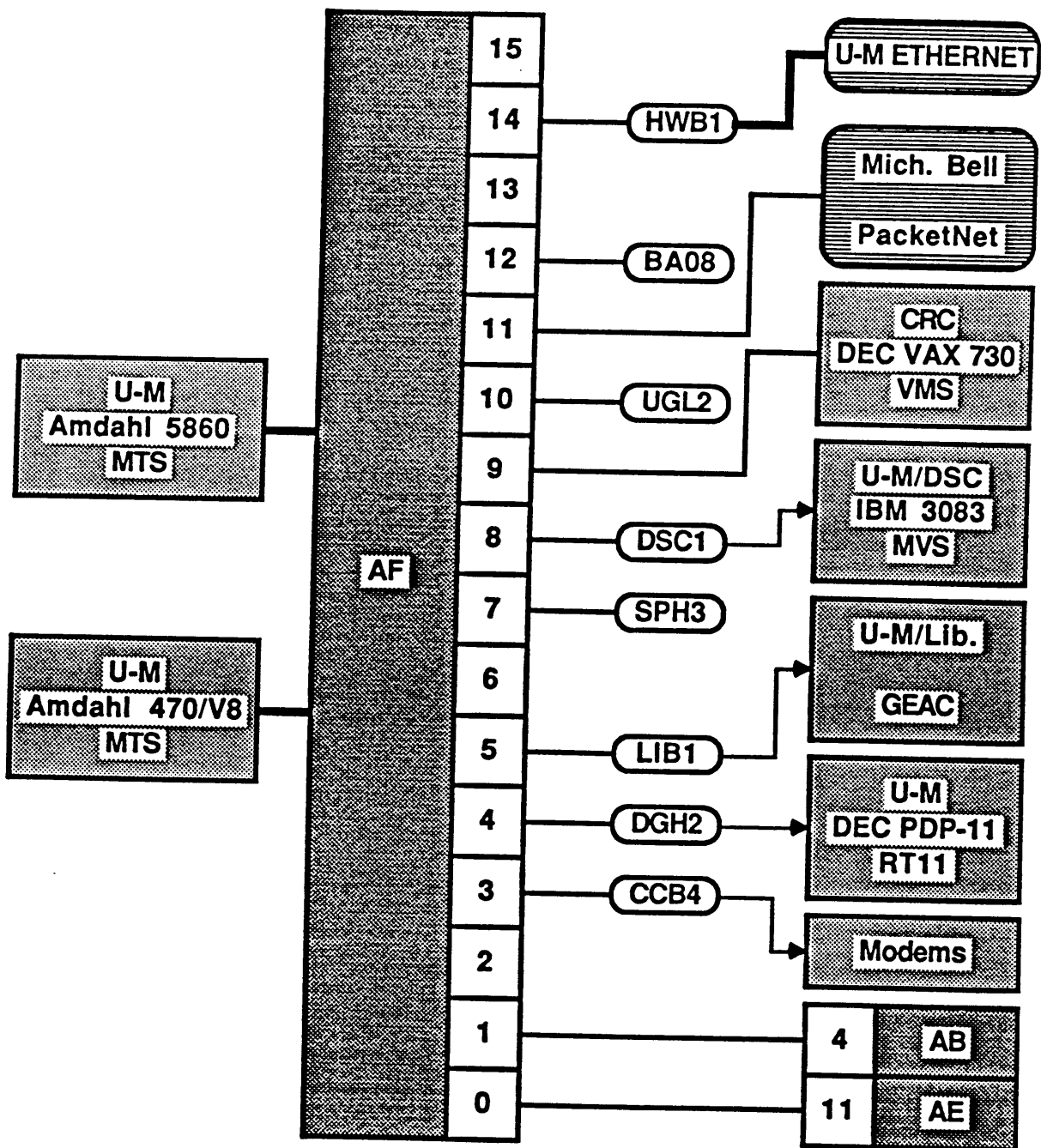
PCP Hardware: PDP 11/34, 2 MM16s, 1 IBM Byte Multiplexor Host I/F

Hermes Ports: None

Number of SCPs: 15

Number of X.25 Ports: None

Number of Internodal Links: 5



PCP Name: AF

PCP Location: U-M Computing Center

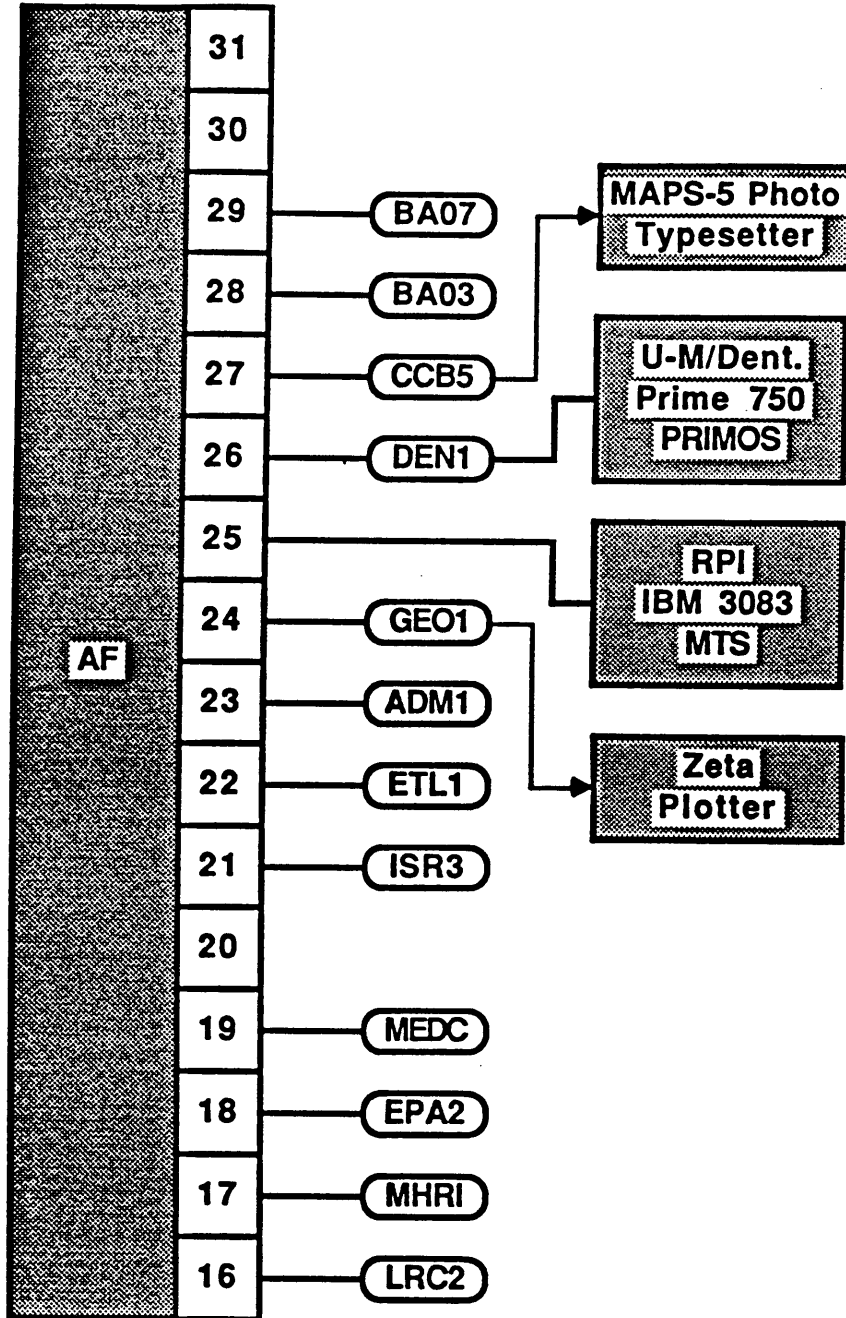
PCP Hardware: PDP 11/60, 2 MM16s, 2 IBM Block Multiplexor Host I/Fs

Hermes Ports: None

Number of SCPs: 20

Number of X.25 Ports: 3

Number of Internodal Links: 2



PCP Name: AF

PCP Location: U-M Computing Center

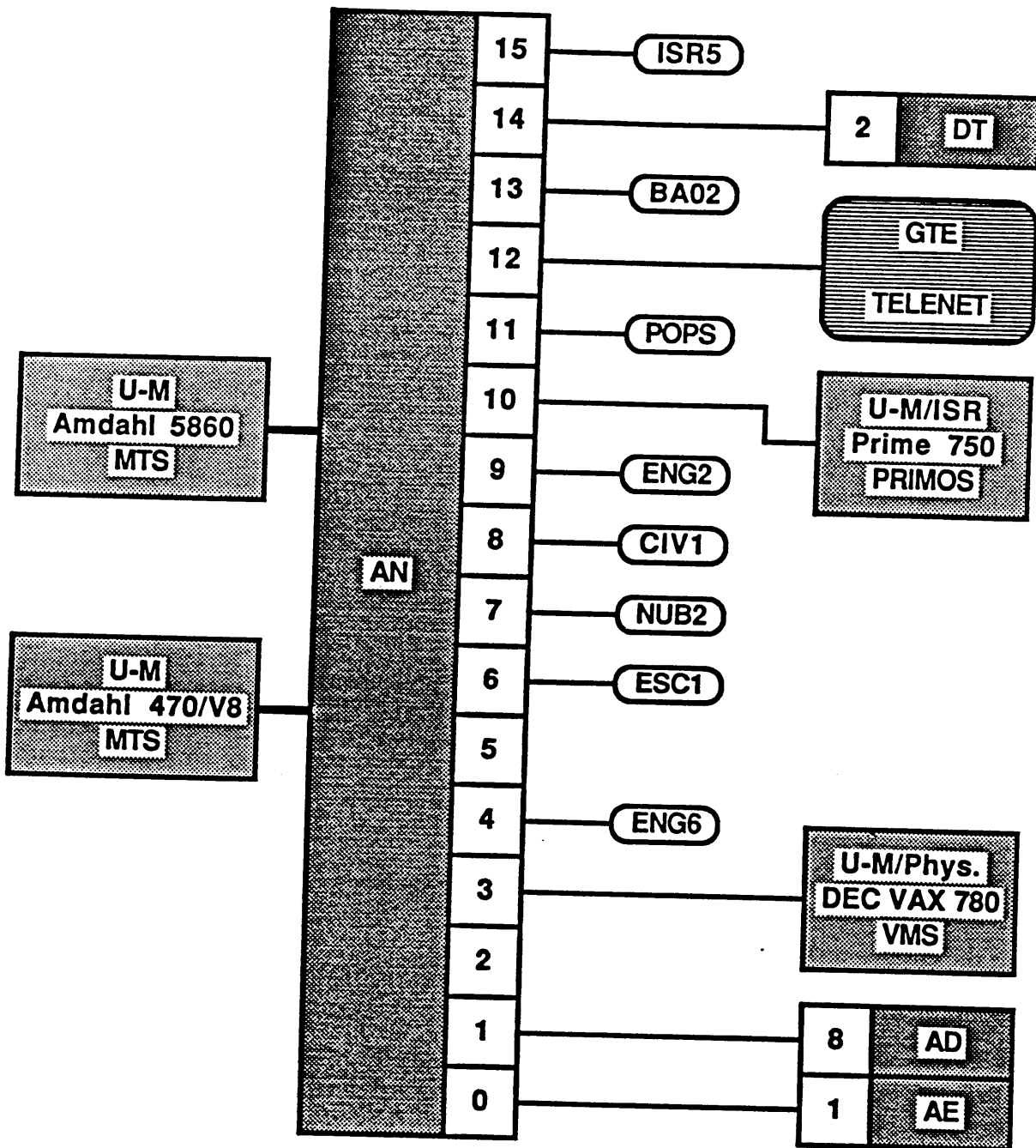
PCP Hardware: PDP 11/60, 2 MM16s, 2 IBM Block Multiplexor Host I/Fs

Hermes Ports: None

Number of SCPs: 20

Number of X.25 Ports: 3

Number of Internodal Links: 2



PCP Name: AN

PCP Location: U-M Computing Center

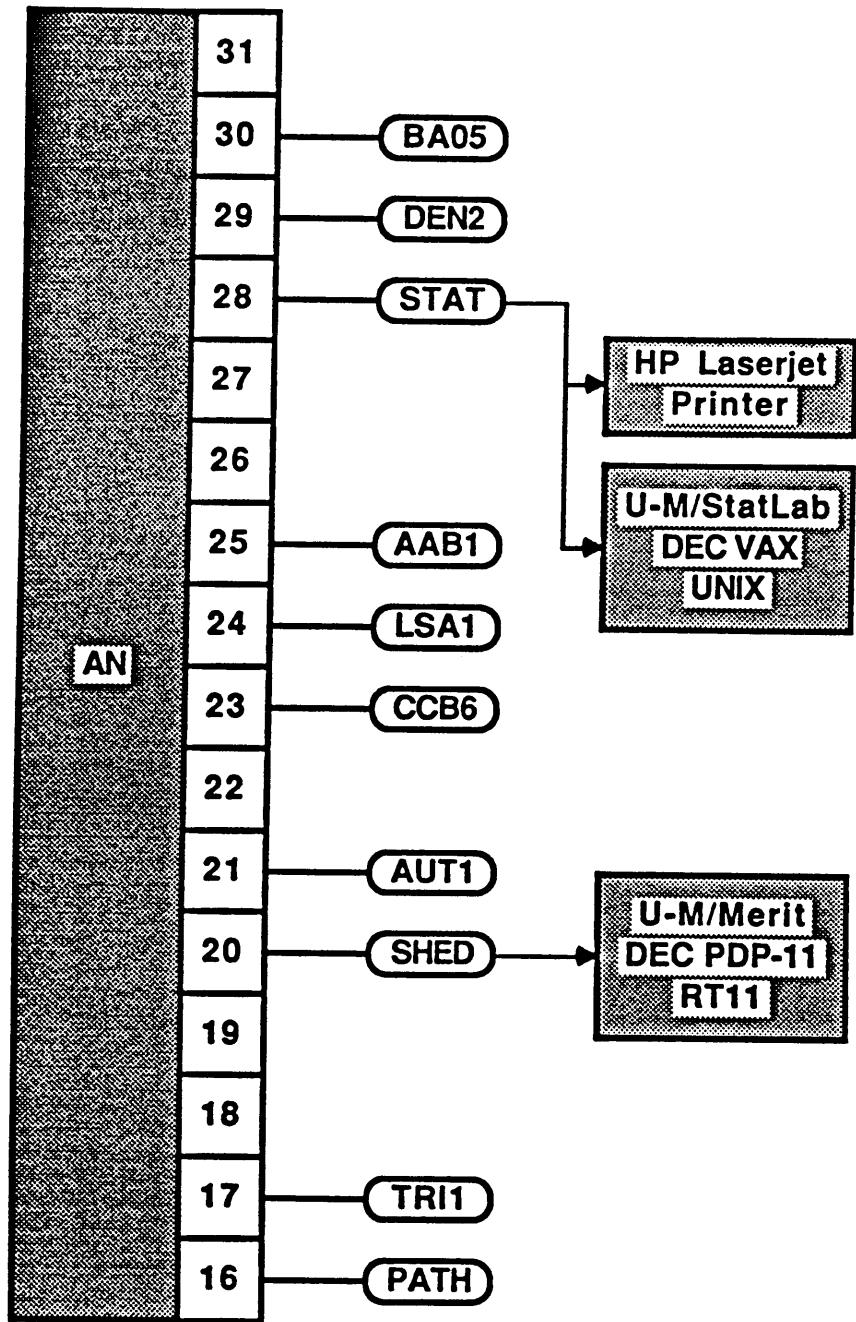
PCP Hardware: PDP 11/60, 2 MM16s, 2 IBM Block Multiplexor Host I/Fs

Hermes Ports: None

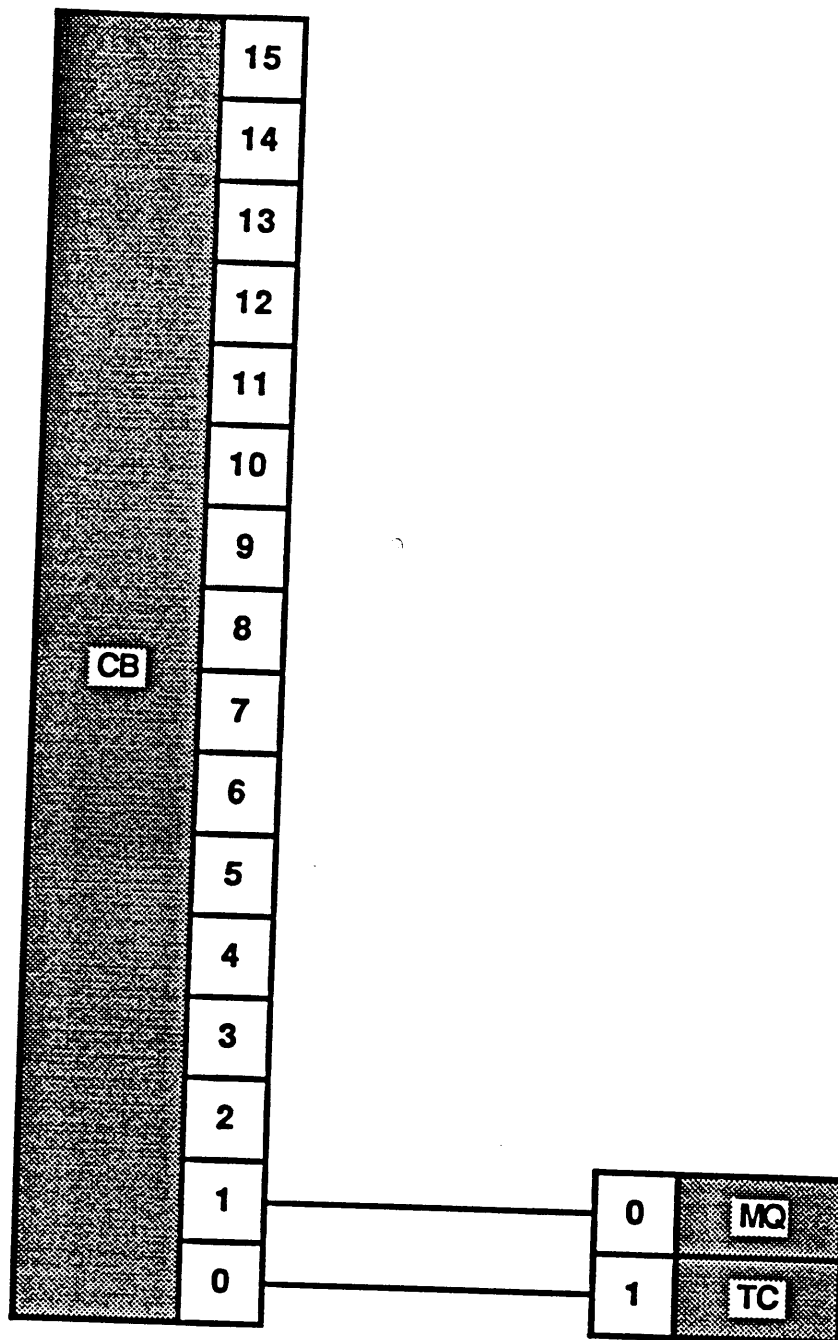
Number of SCPs: 18

Number of X.25 Ports: 3

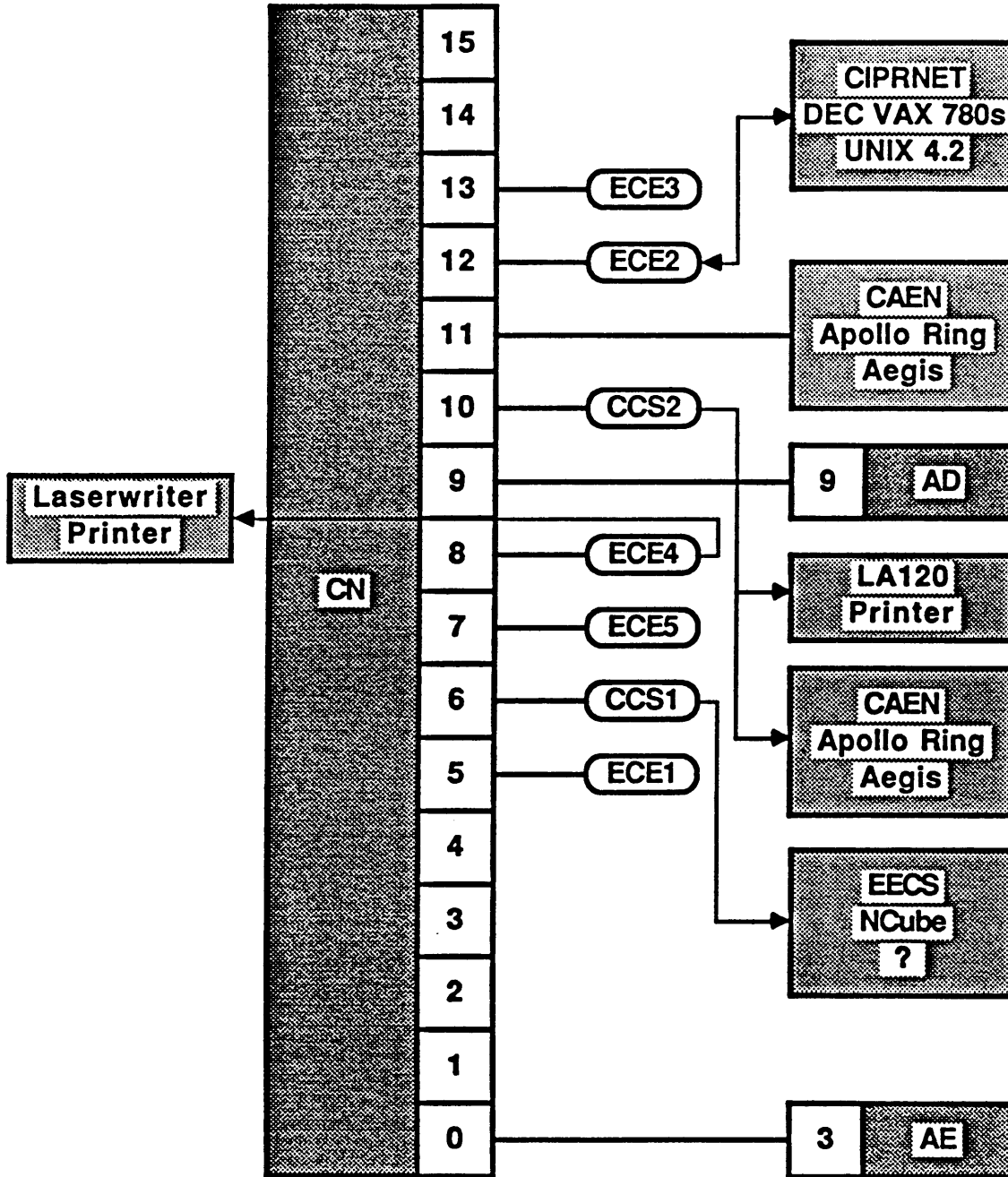
Number of Internodal Links: 3



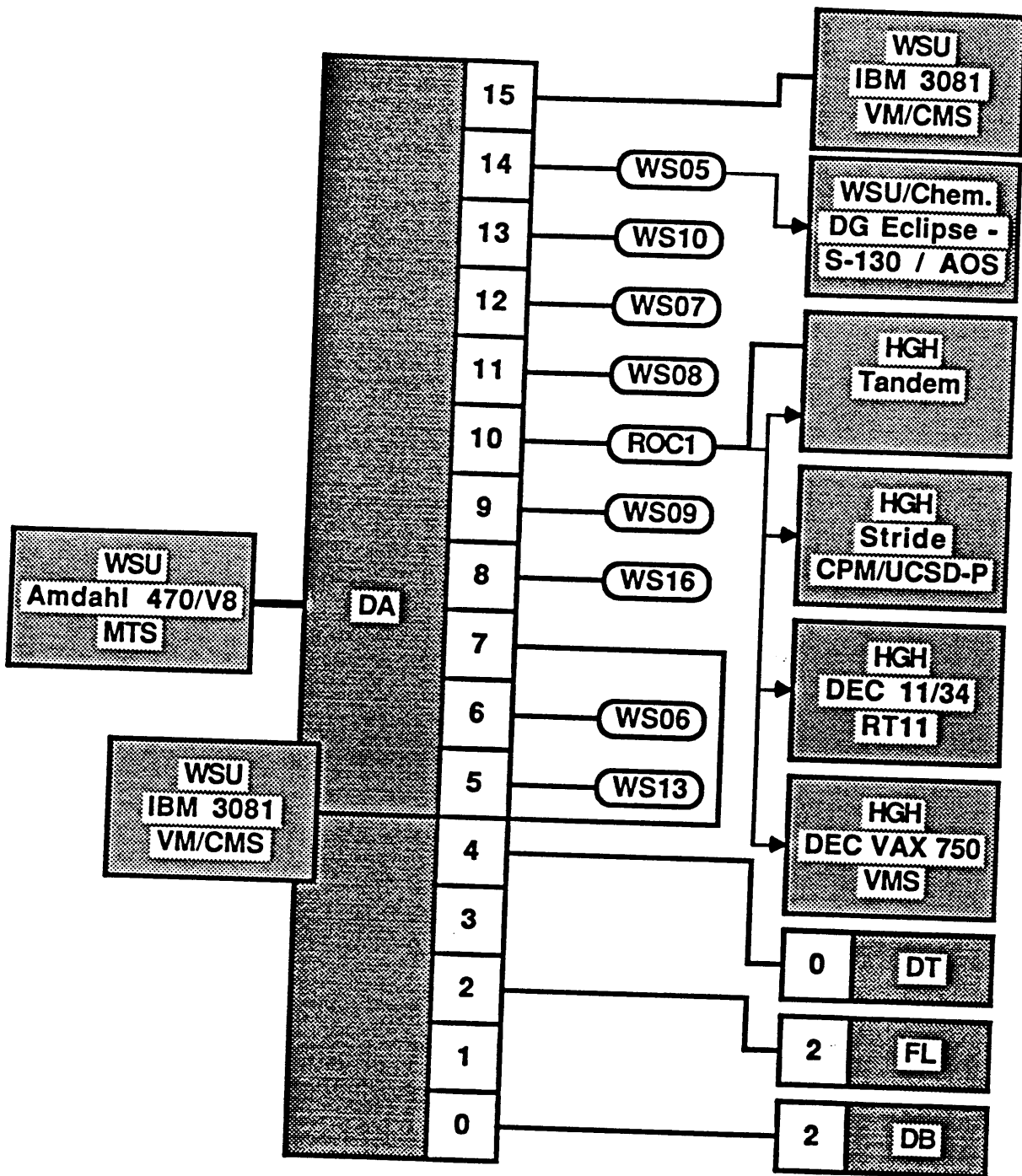
PCP Name: AN
 PCP Location: U-M Computing Center
 PCP Hardware: PDP 11/60, 2 MM16s, 2 IBM Block Multiplexor Host I/Fs
 Hermes Ports: None
 Number of SCPs: 18
 Number of X.25 Ports: 3
 Number of Internodal Links: 3



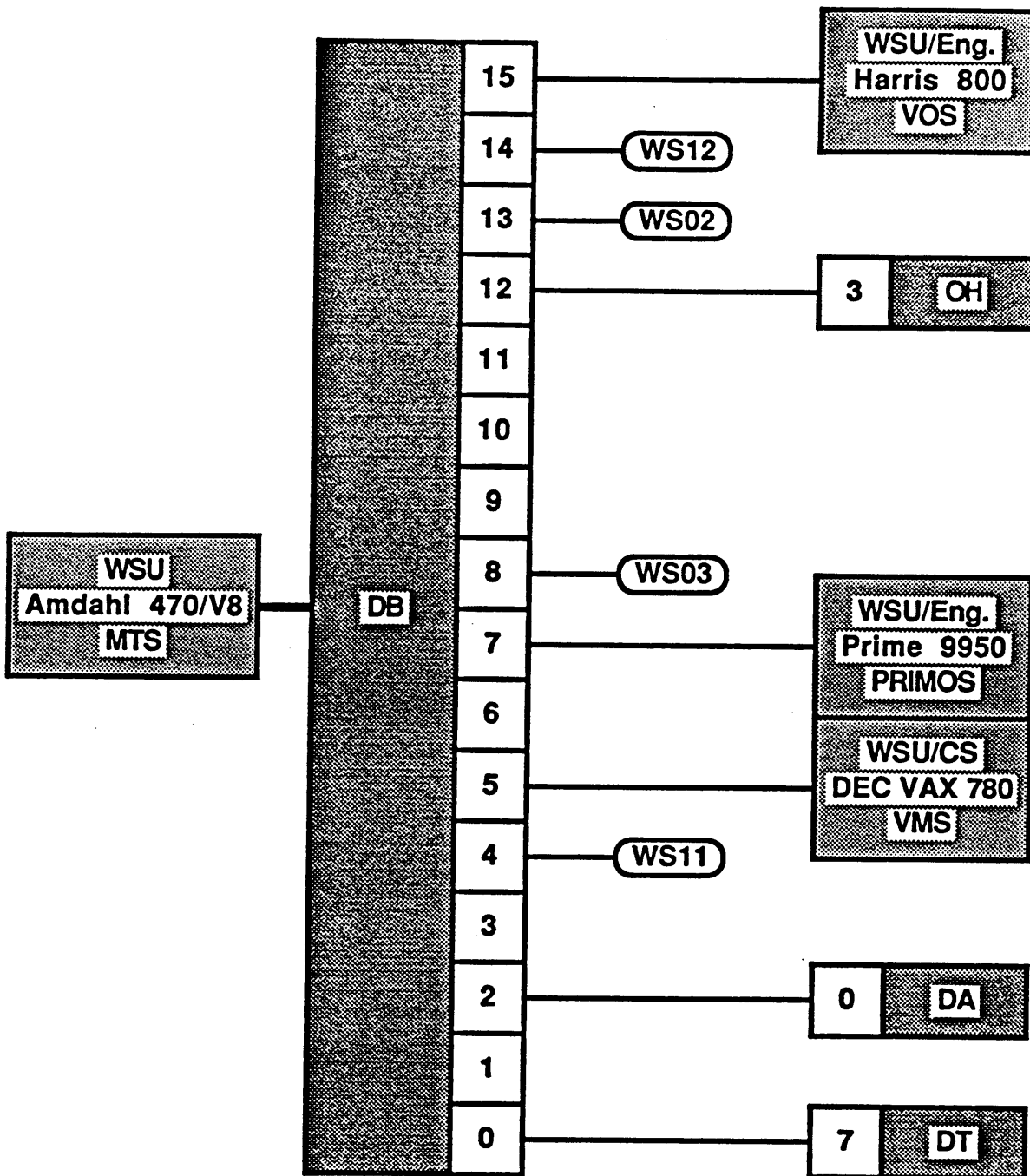
PCP Name: CB
 PCP Location: Cheboygan, Michigan
 PCP Hardware: PDP 11/73, 2 KHVs
 Hermes Ports: 8 Hardwired, 8 Dial-Up
 Number of SCPs: None
 Number of X.25 Ports: None
 Number of Internodal Links: 2



PCP Name: CN
 PCP Location: U-M's East Engineering Building
 PCP Hardware: PDP 11/60, 1 MM16
 Hermes Ports: None
 Number of SCPs: 7
 Number of X.25 Ports: 1
 Number of Internodal Links: 2



PCP Name: DA
 PCP Location: WSU Computing Center
 PCP Hardware: PDP 11/60, 1 MM16, 1 IBM Byte Multiplexor Host I/F
 Hermes Ports: None
 Number of SCPs: 9
 Number of X.25 Ports: 2
 Number of Internodal Links: 3



PCP Name: DB

PCP Location: WSU Computing Center

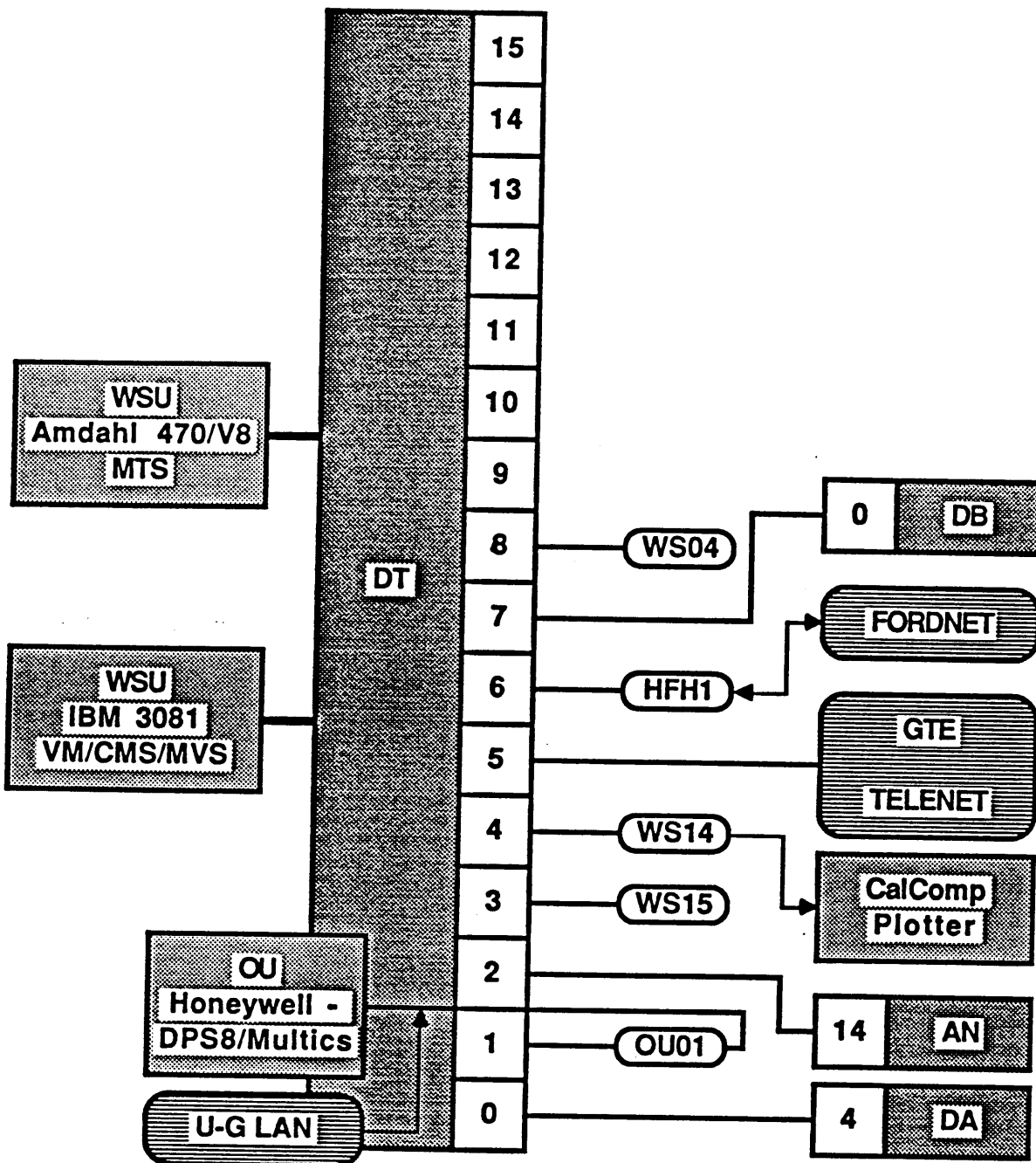
PCP Hardware: PDP 11/60, 1 MM16, 1 IBM Byte Multiplexor Host I/F

Hermes Ports: None

Number of SCPs: 4

Number of X.25 Ports: 3

Number of Internodal Links: 3



PCP Name: DT

PCP Location: WSU Computing Center

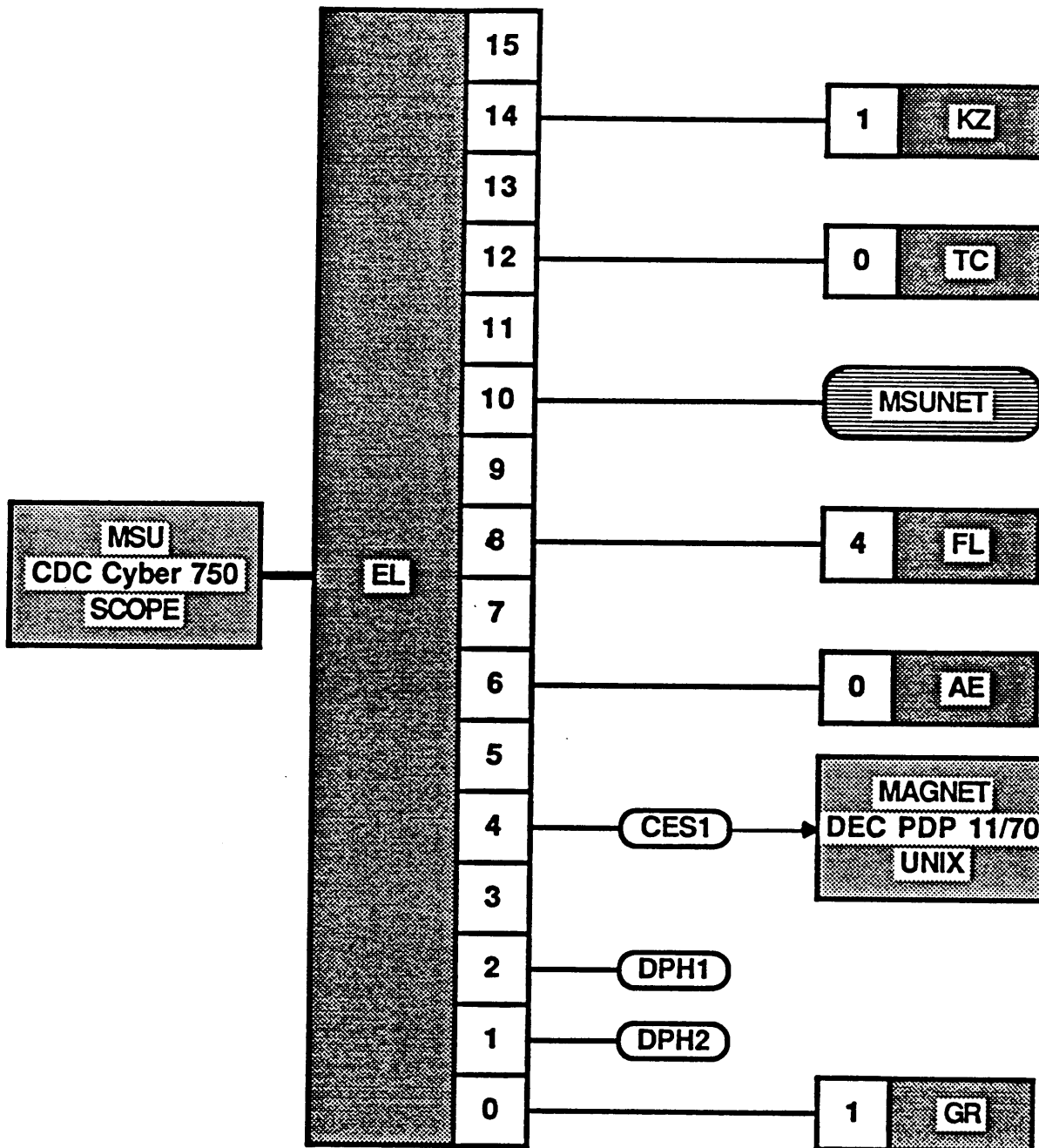
PCP Hardware: PDP 11/60, 1 MM16, 2 IBM Byte Multiplexor Host I/Fs

Hermes Ports: None

Number of SCPs: 5

Number of X.25 Ports: 1

Number of Internodal Links: 3



PCP Name: EL

PCP Location: MSU Computer Laboratory

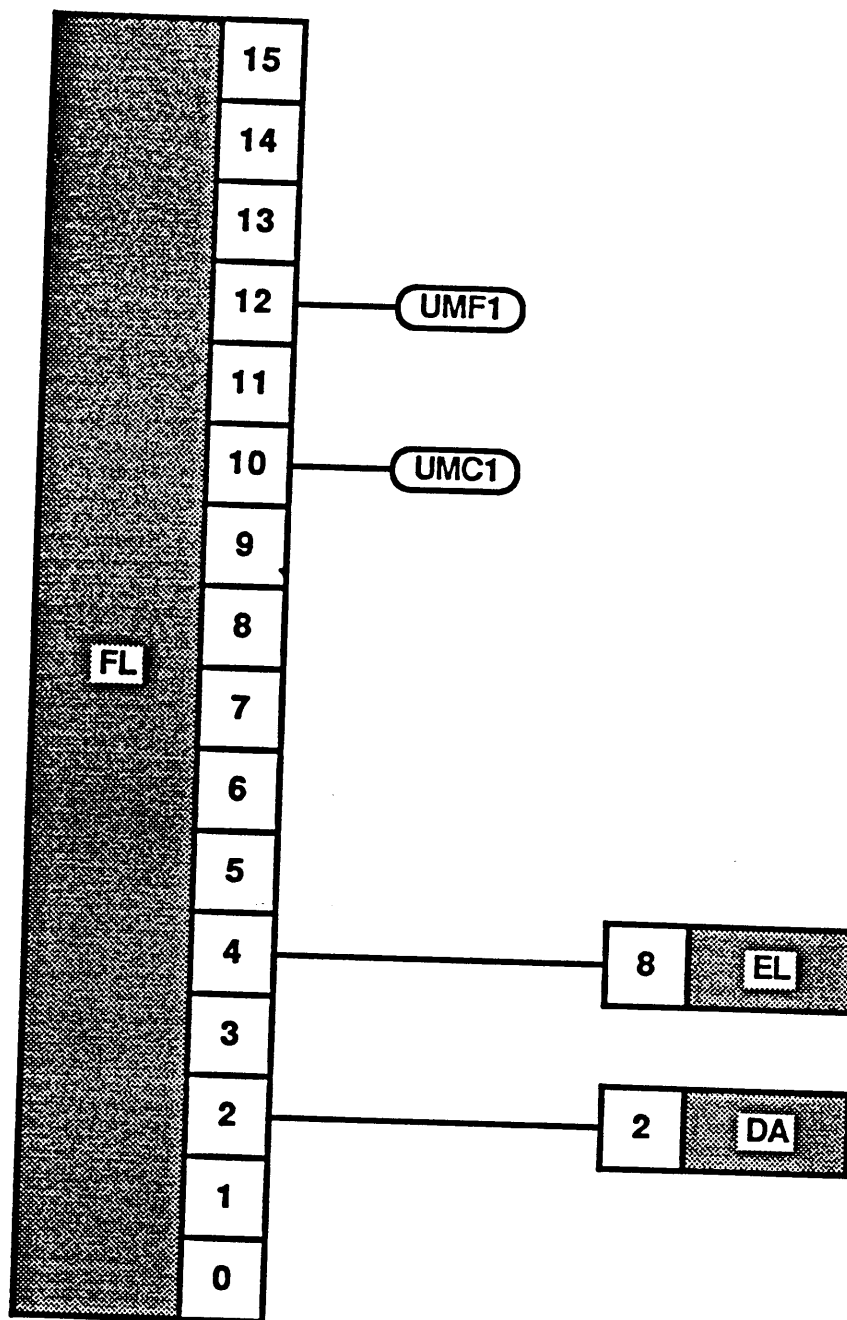
PCP Hardware: PDP 11/34, 1 MM16, 1 CDC Host I/F

Hermes Ports: 6 300 bps, 2 1200 bps and 2 Hardwired

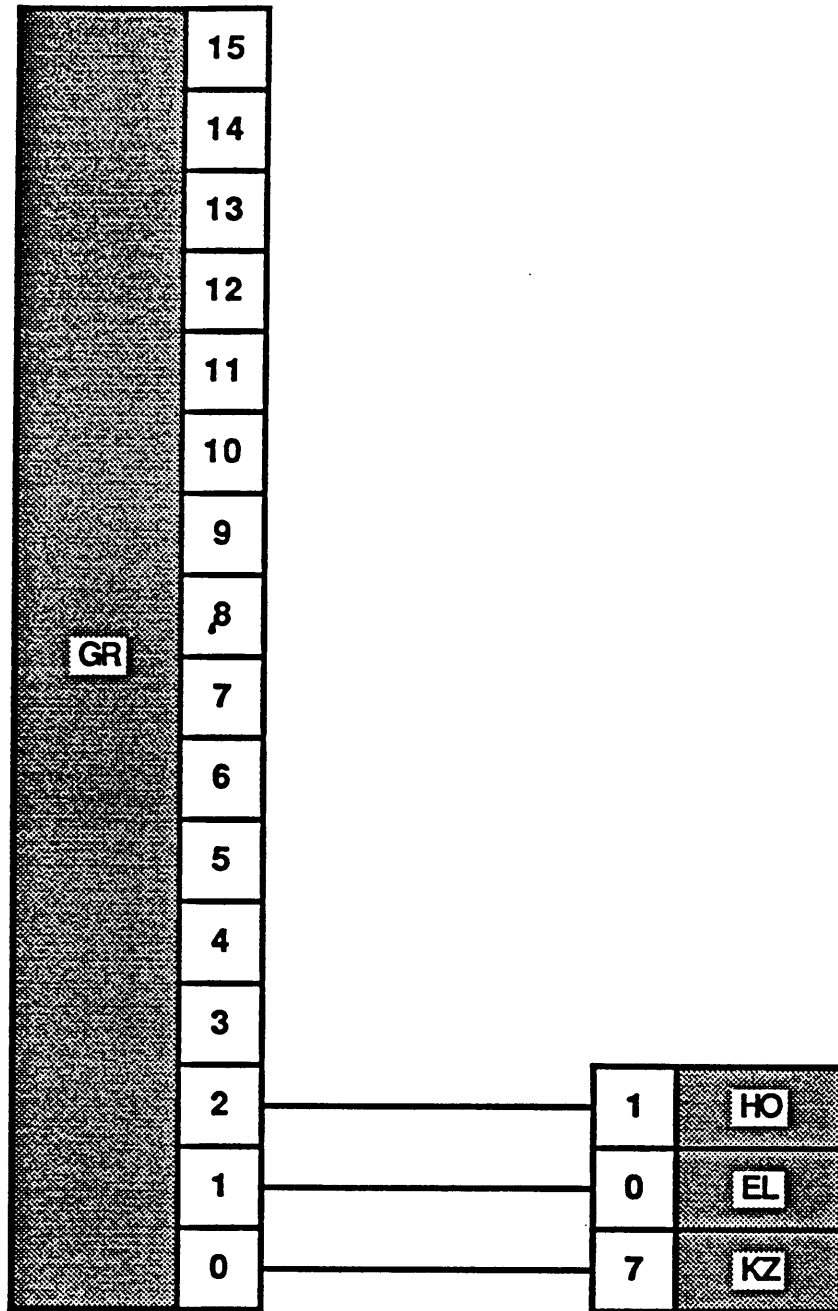
Number of SCPs: 3

Number of X.25 Ports: 1

Number of Internodal Links: 5



PCP Name: FL
 PCP Location: U-M/Flint
 PCP Hardware: PDP 11/60, 1 MM16
 Hermes Ports: 13 Hardwired, 6 1200 bps
 Number of SCPs: 2
 Number of X.25 Ports: None
 Number of Internodal Links: 2



PCP Name: GR

PCP Location: WMU Extension Center in Grand Rapids

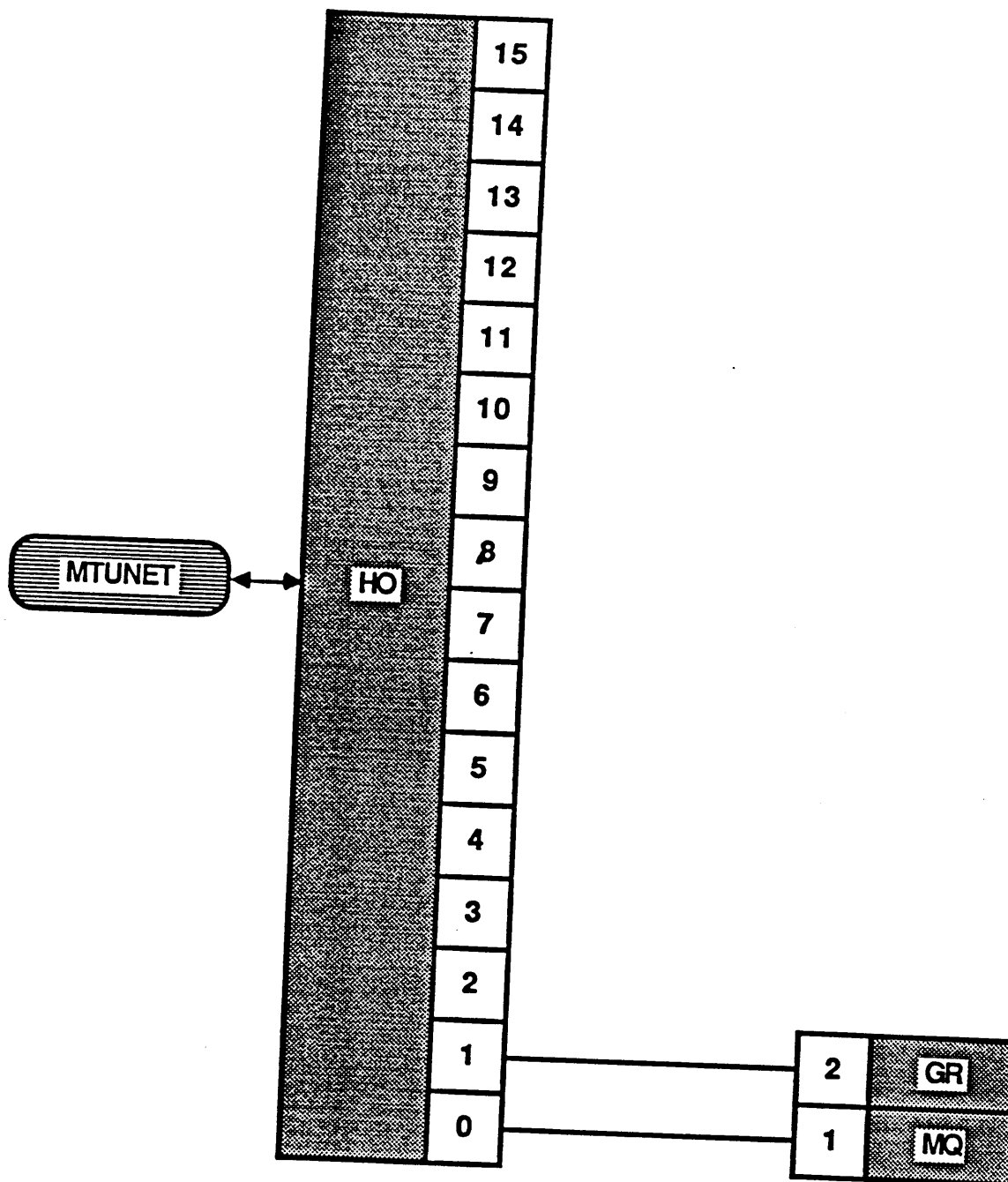
PCP Hardware: PDP 11/73, 3 KHVs

Hermes Ports: 16 Hardwired, 8 Dial-UP

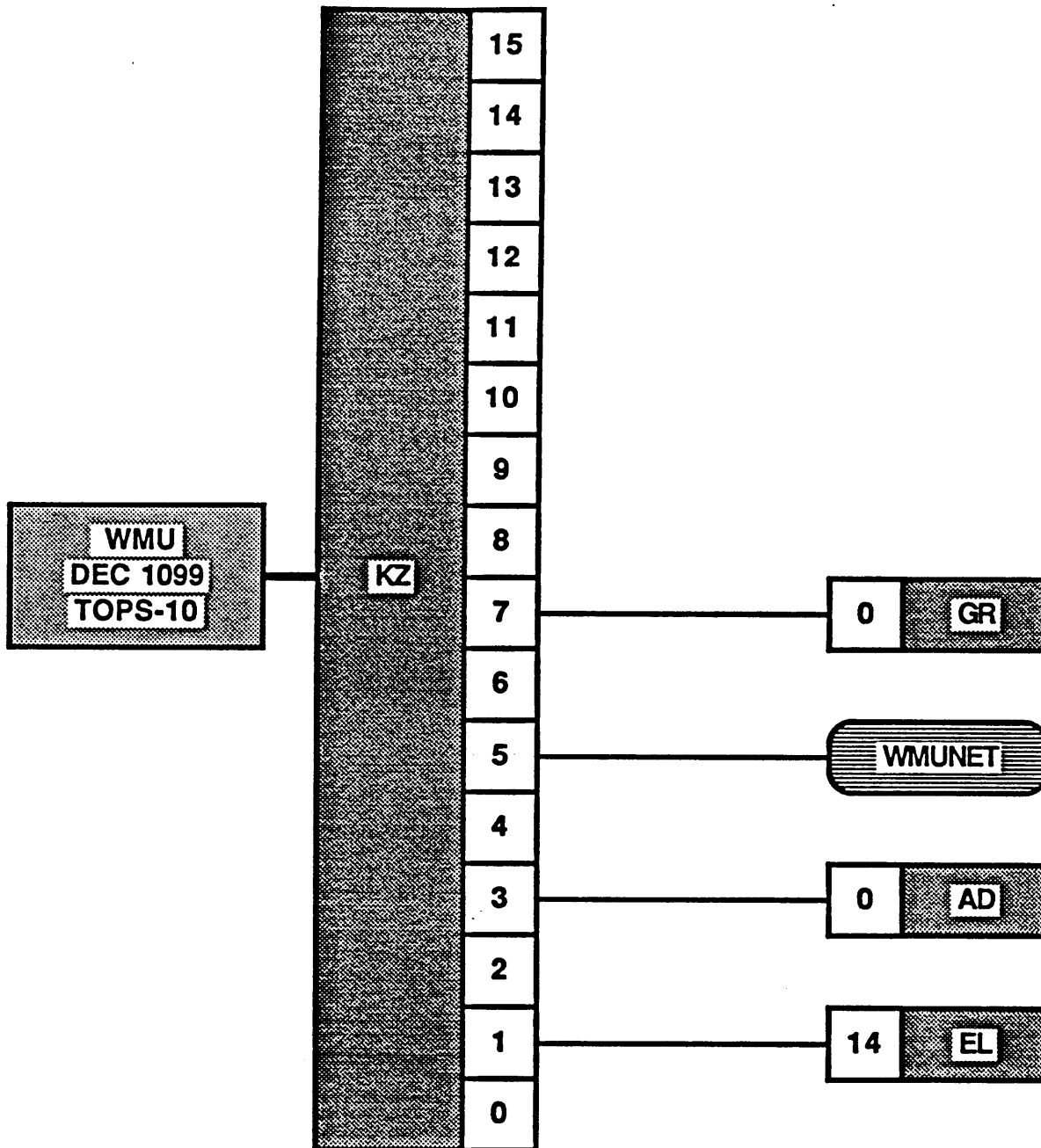
Number of SCPs: None

Number of X.25 Ports: None

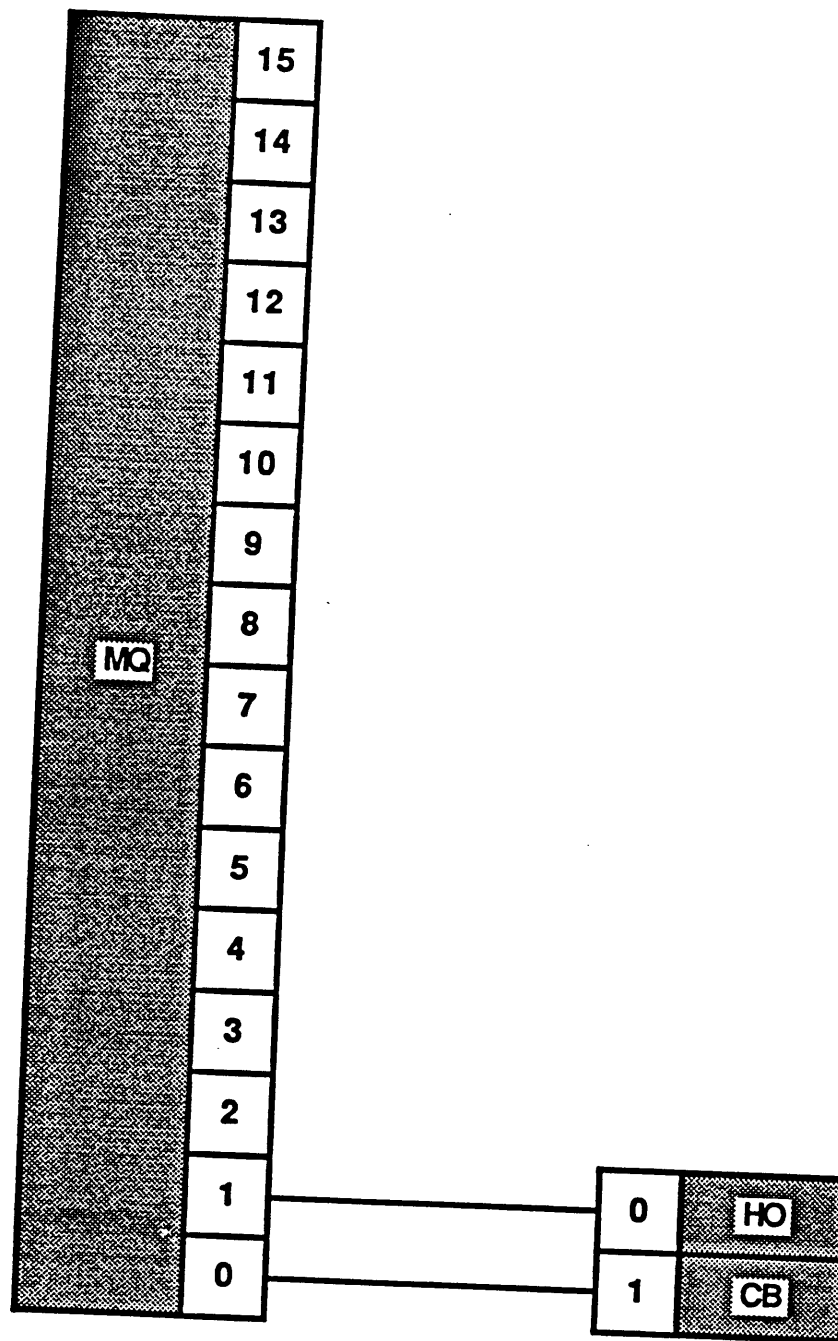
Number of Internodal Links: 3



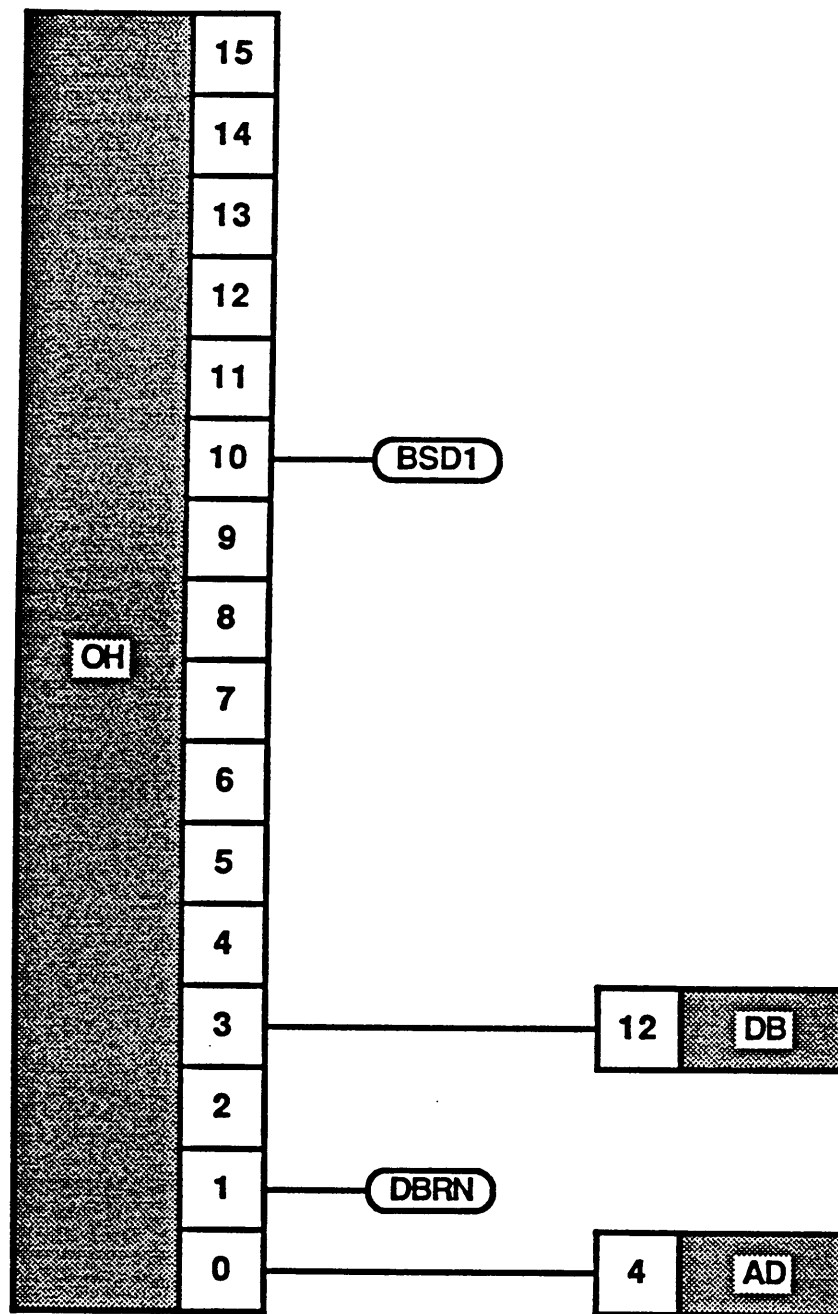
PCP Name: HO
 PCP Location: MTU Computing Center in Houghton, Michigan
 PCP Hardware: PDP 11/73, 2 KHVs
 Hermes Ports: 16 Hardwired
 Number of SCPs: None
 Number of X.25 Ports: None
 Number of Internodal Links: 2



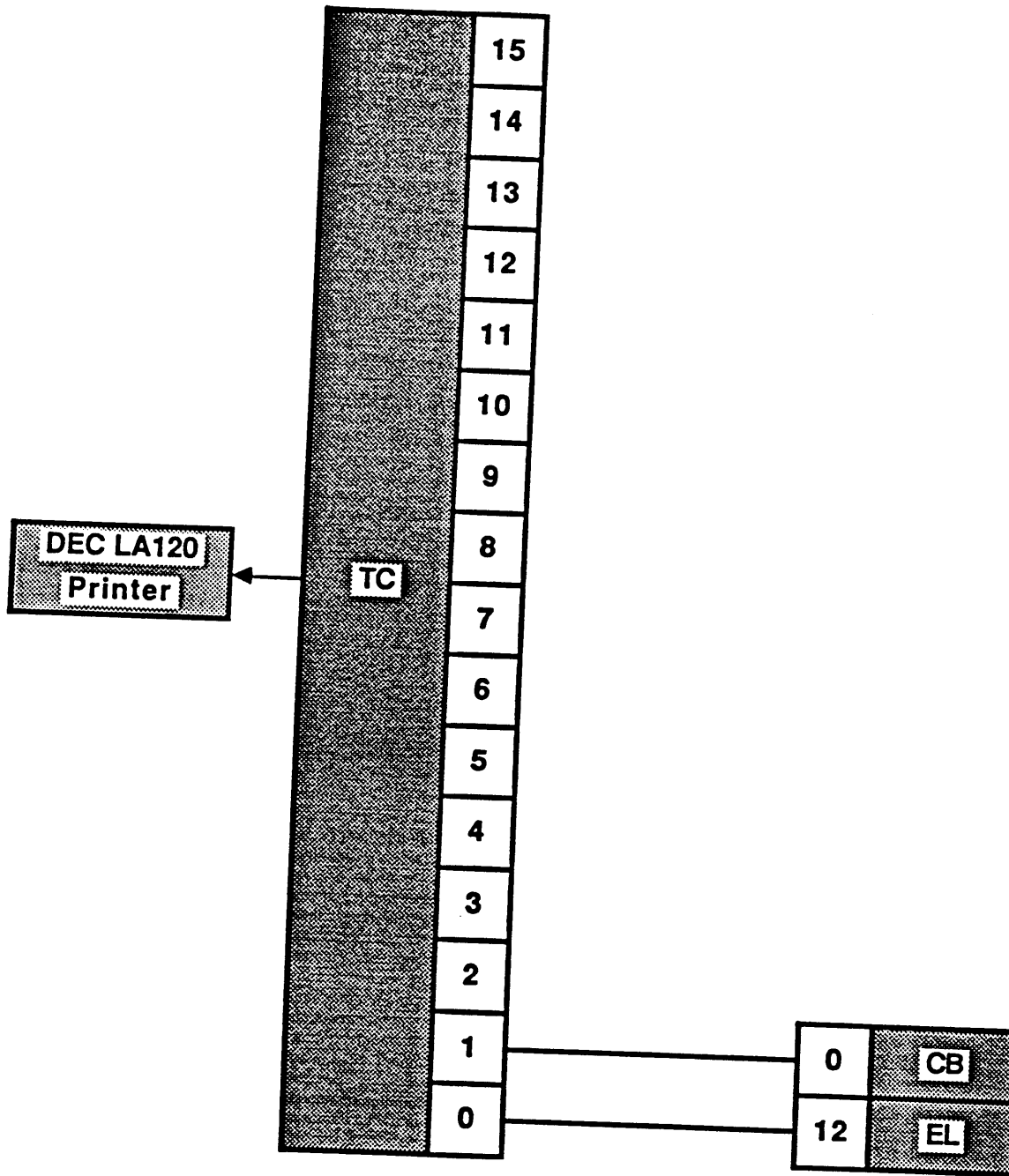
PCP Name: KZ
 PCP Location: WMU Computing Center
 PCP Hardware: PDP 11/40, 1 MM16, DEC DTE Host I/F
 Hermes Ports: 8 Hardwired, 8 1200 bps
 Number of SCPs: None
 Number of X.25 Ports: 1
 Number of Internodal Links: 3



PCP Name: MQ
 PCP Location: Marquette, Michigan
 PCP Hardware: PDP 11/73, 2 KHVs
 Hermes Ports: 24 Hardwired, 8 Dial-Up
 Number of SCPs: None
 Number of X.25 Ports: None
 Number of Internodal Links: 2



PCP Name: OH
 PCP Location: U-M/Dearborn Computing Center
 PCP Hardware: PDP 11/60, 1 MM16
 Hermes Ports: 16 Hardwired
 Number of SCPs: 2
 Number of X.25 Ports: None
 Number of Internodal Links: 2



PCP Name: TC
 PCP Location: Traverse City, Michigan
 PCP Hardware: PDP 11/73, 2 KHVs
 Hermes Ports: 16 Hardwired, 16 Dial-Up
 Number of SCPs: None
 Number of X.25 Ports: None
 Number of Internodal Links: 2