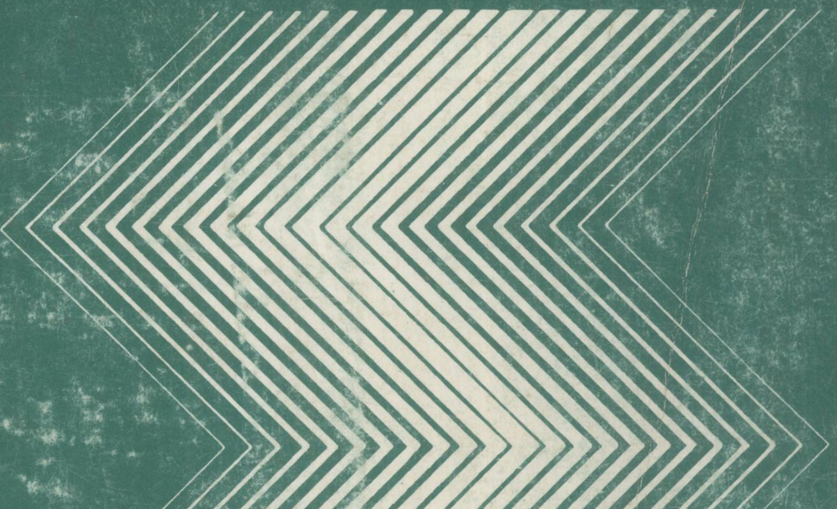
Computer Communications Review
Volume 18, Number 4
August 1988

acm PRESS

SIGCOMM '88 SYMPOSIUM

Communications Architectures & Protocols

Stanford, California
August 16–19, 1988



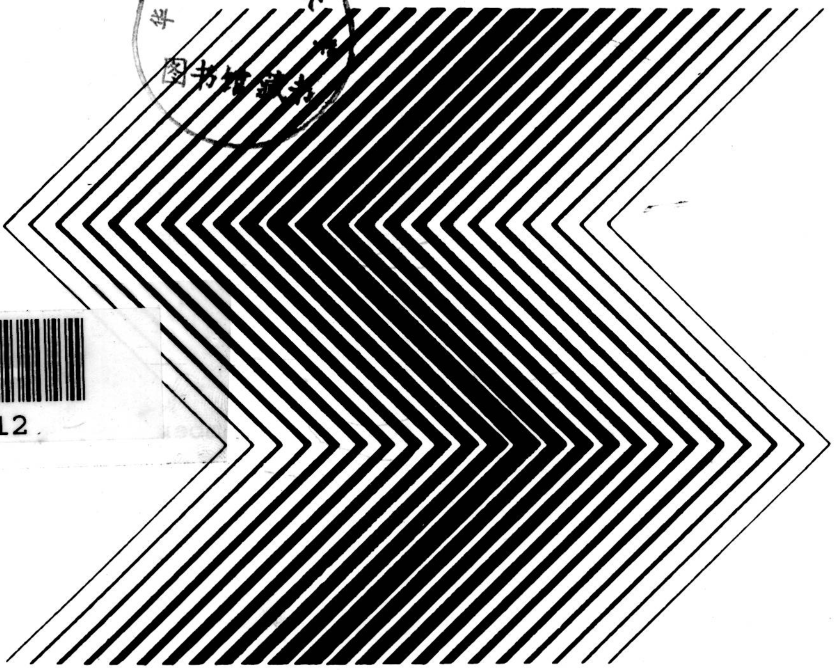
SPONSORED BY ACM SIGCOMM WITH SUPPORT GIVEN BY THE INFORMATION SCIENCES AND TECHNOLOGY CENTER OR SRI INTERNATIONAL.

acm PRESS

SIGCOMM '88 SYMPOSIUM

Communications Architectures & Protocols

Stanford, California
August 16–19, 1988



SPONSORED BY ACM SIGCOMM WITH SUPPORT GIVEN BY THE INFORMATION SCIENCES AND TECHNOLOGY CENTER OR SRI INTERNATIONAL.

The Association for Computing Machinery
11 West 42nd Street
New York, New York 10036

ISBN 0-89791-279-9

Additional copies may be ordered prepaid from:

ACM Order Department
P.O. Box 64145
Baltimore, MD 21264

Price:	
Members	........$20.00
All others	........$26.00

ACM Order Number: 533880

MESSAGE FROM THE GENERAL CHAIR

SIGCOMM symposia have had a steady increase in the submission of high-quality papers over the past few years, which has permitted their organizers to be very selective in the organization of their technical programs. As a result, these symposia are becoming *the* international conference event in computer communications; and their proceedings, published as special issues of the ACM Computer Communication Review, constitute an invaluable source of timely information for anyone interested in computer communication technology.

The technical sessions of SIGCOMM '88 reflect the growing interest in the research community on network management, high-speed networks, large-scale networks and internetworks, and new approaches to protocol engineering. After the keynote address by Dr. Donald Nielson, the symposium attendees will be able to participate in 10 paper sessions and one panel, all organized in a single track. In conjunction with the symposium, two tutorials are being presented. Mario Gerla reviews the evolution of the ISDN (narrowband and broadband) concept during the past few years. Radia Perlman discusses the interconnection of local area networks through bridges and routers.

For the second time in SIGCOMM symposia, this year we held a student-paper contest to encourage student participation and reward academic excellence. Stephen Deering is the winner of the contest. His paper was refereed for acceptance to the symposium using the same criteria than for regular papers. I selected it from the list of accepted papers based on the outstanding ratings given by the referees.

I would like to acknowledge the outstanding job done by Prof. Lawrence Landweber, Program Chair, in organizing the activities of the program committee, in putting together a strong technical program, and in producing the proceedings of the symposium. We are grateful to Stanford University for hosting the SIGCOMM symposium this year, and to Dave Cheriton for sponsoring the event within Stanford.

I would also like to take this opportunity to thank the members of the technical program committee, all the authors who submitted papers, the numerous reviewers, and the panelists for their efforts in making SIGCOMM '88 a big success. Finally, I want to acknowledge Michael Frankel and Earl Craighill, both with the Information Sciences and Technology Center of SRI International, for their generous support of this undertaking.

J.J. Garcia-Luna-Aceves
General Chair, SIGCOMM '88

MESSAGE FROM THE PROGRAM CHAIR

The ACM SIGCOMM '88 Symposium provides an international forum for the presentation and discussion of recent advances on communications architectures, protocols, algorithms, and performance models. The symposium begins with a keynote session including presentation of the "best student paper" award and the keynote address by Donald Nielson. The remainder of the symposium consists of ten technical paper sessions and a panel discussion on "Internet Engineering" organized by Phil Gross.

The program reflects the diversity of the submissions. Session 1 considers local and metropolitan network design. The papers in Session 2 deal with questions related to routing. This is followed by Session 3 on operating system, network management and transport issues. The second day of the symposium begins with a discussion on lessons learned from IP internet (Session 4). Papers in Session 5 are concerned with local area network architecture and design. Session 6 deals with very high speed networking issues. Session 7 concentrates on testing, measurement and management of networks. The last day begins with Session 8 on protocol testing and design followed by Session 9 on questions arising in broadcast networks. The last paper session (Session 10) is concerned with congestion and topology control.

The proceedings include 32 articles selected from 116 submitted papers. Because of the large number of submissions and the decision to not hold parallel sessions, the acceptance rate for Sigcomm '88 is lower than that of the previous two Sigcomm symposia.

Two or three reviews were obtained for each submitted paper. I would like to thank the members of the program committee and each of the additional referees, who worked with program committee members, for their outstanding efforts in handling the very difficult task of selecting the very best among so many high-quality and diverse contributions. The committee operated under particularly severe time pressure because of the delay in the submission deadline.

I would also like to thank Jose Garcia-Luna (Conference Chair) for his invaluable assistance. Finally, I would like to thank Sheryl Pomraning of the University of Wisconsin for her fantastic support of the program committee's efforts.

Lawrence H. Landweber
Program Chair
ACM SIGCOMM '88 Symposium

SYMPOSIUM COMMITTEE

General Chair

J.J. Garcia-Luna-Aceves
SRI International
333 Ravenwood Avenue
Menlo Park, CA 94025
USA

Program Chair

Lawrence H. Landweber
Computer Sciences Department
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706
USA

PROGRAM COMMITTEE

David Anderson, *University of California at Berkeley, USA*
Vint Cerf, *Corporation for National Research Initiatives, USA*
Lyman Chapin, *Advanced Computer Environments, USA*
Roger Cheung, *Hewlett-Packard, USA*
David Farber, *University of Pennsylvania, USA*
Mario Gerla, *UCLA, USA*
Christian Huitema, *Centre de Sophia Antipolis, France*
Simon Lam, *University of Texas at Austin, USA*
David Mills, *University of Delaware, USA*
Jeff Mogul, *Digital Equipment Corporation, USA*
Jun Murai, *University of Tokyo, Japan*
Raphael Rom, *Technion, Israel*
Deepinder Sidhu, *University of Maryland, USA*
Martti Tienari, *University of Helsinki, Finland*
Steve Wolff, *National Science Foundation, USA*
Willy Zwaenepoel, *Rice University, USA*

KEYNOTE SESSION

Keynote Address: Donald Nielson, SRI International, USA

List of Referees

The work of refereeing was undertaken largely by the Program Committee. Invaluable help was provided, however, by the referees listed below.

Anderson, J.
Bannister, J.
Burnett, J.
Calvert, K.
Downing, A.
Eloranta, J.
Gerla, M
Gouda, M.
Greenberg, I.
Hagens, R.
Hall, N.
Ilnicki, S.
Jain, P.
Jain, R.
Joseph, D.
Kosinsky, W.
Lahtinen, P.
Marttinen, L.
Monteiro, J.
Mukherjee, A.
Peha, J.
Puustjärvi, J.
Rolroher, M.
Rönn, S.
Rosier, L.
Sadeniemi, M.
Schaffa, F.
Shacham, N
Song, C.
Sun, Y.
Tarpila, K.
Tirri, H.
Turunen, I.
Valencia, A.
Vernon, M.
Vihavainen, J.

Table of Contents

Session 2: Local/Metropolitan Area Internets

Chair: D. Anderson, Unviversity of California, Berkeley, USA

Session 3: Routing

Chair: L. Chapin, Data General Corporation, USA

Session 4: Transport Level and Operating System Issues

Chair: S. Lam, University of Texas at Austin, USA

Session 5: Lessons of the Internet

Chair: J. Mogul, Digital Equipment Corporation, USA

Session 6: Local Area Network Architecture

Chair: R. Cheung, Hewlett-Packard, USA

Session 7: Very High Speed Networking

Chair: D. Farber, University of Pennsylvania, USA

Session 8. Measurement and Management

Chair: V. Cerf, Corporation for National Research Initiatives, USA

Topological Analysis of Local-Area Internetworks

Glenn M. Trewitt
Stanford University

Abstract

It has become common to connect local-area networks together to form high-bandwidth internetworks. The topology of such an internetwork — how the component networks and gateways are interconnected — is an important factor in determining the reliability of the internet. We present several techniques for analyzing an internetwork's topology. These techniques are based on a novel mapping of network components onto a bipartite graph.

We use these techniques to analyze the topology of the Stanford University Network and the impact of topological changes as it has grown to its current size of 60 networks.

—

1. Introduction

The topology of an internetwork — how the component networks and gateways are interconnected — is an important factor in determining how well it will perform when there are failures in its parts. Analysis of the topology can point out critical components or paths, suggesting modifications to improve overall reliability. Proposed changes can also be investigated before implementation, providing a useful planning tool.

The topology is also a basis for comparison of internetworks. Comparing the same internet at different times demonstrates the impact of systemic changes. By comparing different internetworks, implementors can learn from experience gained elsewhere.

We are primarily concerned with local-area internetworks, because they often have a richer, more dynamic topology than relatively static wide-area internetworks [1]. The methods outlined here apply equally well to wide-area internets, however. For any given internetwork topology, it is possible to do an analysis by hand that produces the same results we describe here [5, pp. 8-9]. Because local-area internet topologies change relatively frequently (see section 6), such manual analyses would be very tedious and time-consuming to do on a regular basis.

Our goal is to replace the expert intuition required for *ad hoc* analysis with precise definitions of the important characteristics of an internetwork topology. Working from these definitions, we have built a set of tools to do topological analysis and coupled them with automated network monitoring tools that extract the topology from the running network. This automation has been increasingly important as the Stanford internet has grown to the point where no one person knows all of its details [1, pp. 151-156].

We represent an internetwork topology as a bipartite graph. This model represents the most significant components of an internet — the network segments, gateways, and interfaces between them — directly in the graph. The model provides an accurate model for studying the effects of faults.

One of the most interesting characteristics of an internetwork's topology is its resistance to partitioning by failures. We refer to this characteristic as *robustness*. Robustness is usually achieved by adding *redundant* components to an internet, usually both gateways and communications links. We first define several basic topological measures and use them to quantify the amount of redundancy in a topology. We then define techniques for measuring robustness that indicate how resistant a topology is to being partitioned by a failure.

Finally, these techniques are used to analyze changes made to the Stanford University Network over a period of 21 months. We find that there have been significant changes in the robustness, and that some very small topological changes have had a large effect on the robustness of the internetwork.

1.1. Scope of This Paper

The analytical tools described here operate on abstract models of internetwork topologies. Real topologies are usually constrained by technical limits, such as limits on the size of a network or number of interfaces in a gateway, by practical limits, such as restrictions on how cables may be run between or within buildings, and economic considerations. As with any tool, common sense must be applied as well.

There is currently considerable debate about the relative merits of gateways (level-3 routers) versus bridges (level-2 routers) [2, 6, 7]. For the purposes of these analyses, we view both as packet switches and ignore the distinction.

In either case, it is assumed that the routing algorithms used by the gateways or bridges can detect changes and recover in a timely fashion (*e.g.* before connections time out). This assumption is *false* for some implementations of some algorithms and, for those systems, failures that would otherwise go unnoticed may be felt as a temporary loss of connectivity, severe enough to disrupt some connections [3].

2. Modeling the Topology

An internetwork's topology can be modeled as an undirected graph consisting of two kinds of nodes, **gateways** and **networks**, connected by edges. The edges correspond to interfaces connecting a gateway to a network. This type of graph, with two classes of nodes and edges only between different types, is known as a *bipartite* graph.

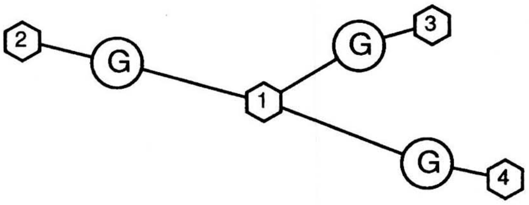


Figure 2-1: Simple Topology Schematic

Figure 2-1 is an example of a simple topology, where the hexagons represent networks and the circles represent gateways. There are several interesting properties of this graph:

- Because an edge represents a connection from a gateway to a network, the nodes encountered along any path through the graph will alternate between gateways and networks.
- Because gateways connect two or more networks, all leaf nodes in the graph will be networks, not gateways.

This model captures a great deal of useful information. Each element of the graph corresponds to a particular component in the real internet. In most technologies, the failure of a component (interface, network, or gateway) corresponds very closely to the removal of the affected piece of the graph. So, this model is useful for understanding the consequences of a failure in the internet and can be used to predict how resistant a topology is to being partitioned by failures.

Many real internet components are *not* shown in this model. The model only shows networks and the gateways interconnecting them. Hosts are not visible, unless they also serve as gateways. Other components, such as repeaters, cables, cable splices, etc. are not shown explicitly. Instead, they are just a part of some component that is shown. For example, a "network" in the graph might include several pieces of cable, some repeaters, and splices. This simplification is safe, as long as a failure in one real component results in the failure of the entire component in the topological model.

So far, this model only describes internetworks composed of gateways and networks. Bridges and network segments can be included by replacing "composite networks" with their component bridges and network segments. Bridges and gateways are equivalent from a topological point of view, as are networks and network segments. Figure 2-2 shows the example topology, assuming that network 1 is composed of two network segments and one bridge. The network segments are represented by rectangles and the bridge is represented by an oval.

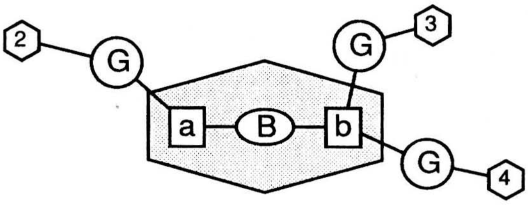


Figure 2-2: Replacing Network 1 with Component Parts

It is also possible to imagine a "composite gateway", where two gateway halves are connected by a single link. An example of this is a fiber-optic Ethernet repeater, which can connect two Ethernet network segments together using an optical fiber. Currently, such devices are strictly two-terminal — they can only connect between two networks. Since a failure in any one of the components (a repeater-half or the link) results in the apparent failure of the entire composite, there is no reason to represent the underlying structure in the topological model.

Once the composite components have been expanded, as shown in figure 2-2, all of the gateways and bridges are treated the same, as are all of the networks and their component segments. There are still only two classes of nodes, as far as the topological analysis is concerned.

3. Measures of a Topology

There are many different ways to measure topologies for comparison. Numeric *metrics* are especially useful. The most obvious metrics are the bulk sizes:

Networks
The total number of component networks. This a good measure of the size of the internet, since it bears some relation to the number of hosts that may be connected to the internet, as well as the geographical span of the internet.

Gateways
The total number of gateways in the internetwork.

Interfaces
The total number of interfaces, represented in the graph

as the number of edges between gateways and networks.

We will refer to these metrics as $N(\tau)$, $G(\tau)$, and $I(\tau)$, where τ is a topology. When it is understood what network is being referred to, these will be abbreviated as N, G, and I. By themselves, these provide a general idea of the overall size of the internet.

Some related metrics are the ratios of the above quantities:

Interfaces/Gateway

K_G The ratio of the number of edges in the graph to the number of gateways. This is the average number of interfaces per gateway. The lower bound on this ratio is two, since a gateway must have at least two interfaces.

Interfaces/Network

K_N The ratio of the number of edges in the graph to the number of networks. This is the average number of gateways connected to each network. For the pathological case of one network and no gateways, this ratio doesn't exist. Other than that, the lower bound is one, since each network must be connected to a gateway to communicate with the rest of the internet.

Based upon our observations, this ratio is usually close to one. This is because the main goal of an internetwork is to attach leaf networks, each of which is usually connected by only one gateway. Each leaf network contributes one interface to the total, pushing the ratio towards one.

K_G represents the average spreading factor at gateways, while K_N represents the average spreading factor at networks. For both of these, larger ratios for similar internetworks indicate increased redundancy. But they do not give a complete picture. It is possible to create a topology with arbitrarily large values for *either* ratio, without increasing redundancy (*i.e.* the internetwork can still have a tree structure).

Attempting to increase *both* K_G and K_N, however, will increase redundancy, because the added edges will tend to make the graph fold back in on itself. We make a more precise characterization of the behavior of these ratios in the next section.

4. Measuring Redundancy

With the above topological model of an internetwork, any internet is represented by a fully-connected, undirected graph, with no duplicate edges. An internet with no redundancy will also be acyclic — there will be exactly one path between any two nodes. Such a graph is a tree, although there is no distinguished root node. For any tree, there is a simple relation between the number of nodes and the number of edges: $Nodes = Edges + 1$. In terms of our metrics, this is $N + G = I + 1$.

If the number of interfaces in a non-redundant internet is fixed, then each choice of G implies a particular N. Since the ratios K_G and K_N are defined in terms of these three numbers, there must be a fixed relation between the two ratios, for a particular value of I. This relation can be used to identify the region that (K_G, K_N) can lie in, which turns out to be quite small. The derivation of this relation follows:

$$G = I - N + 1$$

$$K_G = \frac{I}{G} = \frac{I}{I-N+1} = \frac{I}{I - \frac{I}{K_N} + 1}$$

$$\frac{I}{I+1} = \frac{K_G K_N}{K_G + K_N}$$

For a constant number of interfaces in an internet, this equation prescribes an inverse relation between K_G and K_N. Figure 4-1 shows this curve for I equal to 2, 4, 8, and 16.

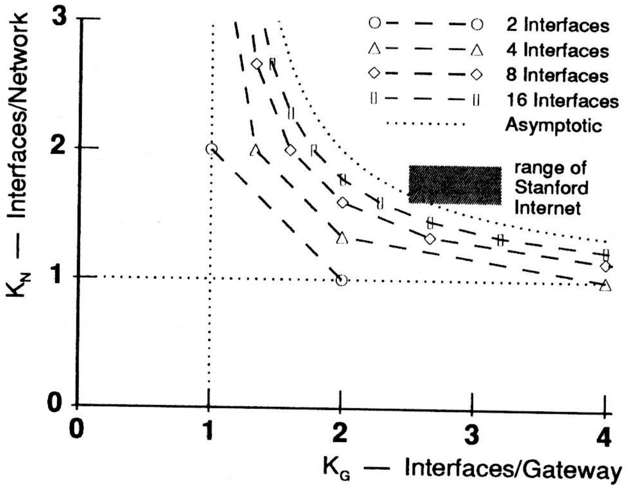


Figure 4-1: K_N vs. K_G for Varying Interfaces

Each of the dashed curves corresponds an internet with a fixed number of interfaces. Different possible combinations of networks and gateways (adding up to $I+1$) are represented by each point. For real internets, K_G will be at least 2, corresponding to the unshaded portion of the graph. For example, for $I=4$, it is possible to have a network with (G,N) combinations of (4,1) (not visible), (3,2), (2,3), and (1,4). The first two, however, represent internets with $K_G < 2$. That part of the graph will not be considered any further.

The dotted line is the limit of this family of curves as the number of interfaces goes to infinity ($\frac{I}{I+1} \rightarrow 1$). Measurements of (K_G, K_N) for the Stanford internetwork all lie inside the small gray rectangle, which is enlarged in Figure 6-1.

For internetworks with no redundant paths, (K_G, K_N) will lie exactly on the curve corresponding to the number of interfaces in the internet. If some interfaces are missing (a partitioned internet), the point will lie below the curve. If there are additional interfaces beyond the minimum required for full connectivity, the point will lie above the curve. The greater the redundancy, the farther from the curve the point will be.

To measure the redundancy, we define the function $Q(\tau)$ to be:

$$Q(\tau) = \frac{K_G K_N}{K_G + K_N} \cdot \frac{I+1}{I}$$

Values of Q equal to 1 lie on the curve, indicating no redundancy, and values greater than 1 lie above the curve, with larger values of Q indicating more redundancy. The relative values of K_G and K_N indicate whether internetwork "fan-out" happens mostly at gateways or networks, and is indicated clearly by the location of (K_G, K_N) on the graph.

Q can be computed directly in terms of I, G, and N, by substituting the definitions for K_G and K_N:

$$Q(\tau) = \frac{\frac{I}{G}\cdot\frac{I}{N}}{\frac{I}{G}+\frac{I}{N}}\cdot\frac{I+1}{I} = \frac{\frac{I^2}{GN}}{\frac{IG+IN}{GN}}\cdot\frac{I+1}{I}$$

$$= \frac{I+1}{G+N} \approx \frac{I}{G+N-1}$$

This formula provides another interpretation of $Q(\tau)$: it is approximately the number of interfaces present in an internetwork divided by the number required to minimally connect all of the gateways and networks in the internet. Therefore, $Q(\tau)-1$ is the percentage of extra interfaces present in the internet.

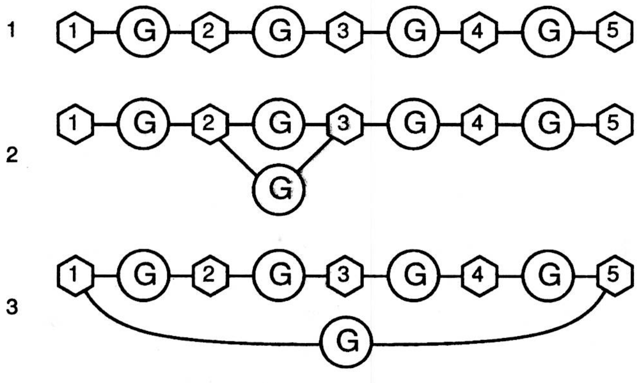


Figure 4-2: Two Possible Improvements

$Q(\tau)$ captures some information about the redundancy present in an internetwork, and is very simple to compute, but it doesn't unambiguously indicate how well the internet will function when a failure occurs. Consider the case of the internetwork shown in figure 4-2(1) and the two possible improvements in (2) and (3). Both consist of adding one gateway and two interfaces to the original internet. Both produce the same improvement in $Q(\tau)$, raising it from 1 to 1.1. However, adding the gateway between two nearby networks only protects against the failure of two gateways or their interfaces. Adding the gateway so that the internet becomes a ring, however, protects against any single failure.

To measure differences in robustness, it is necessary to examine the topology in detail. The next section discusses methods for measuring robustness, which are based upon how an internet is partitioned by failures.

5. Measuring Robustness

To find a way to measure the robustness of a topology, we must first define exactly what we want to measure. The primary purpose of an internetwork is to allow individual networks (and therefore the hosts, users, and processes on them) to communicate. The underlying measure of robustness, then, is the amount of disruption caused by failures. *e.g.* When part of an internet fails, and no one notices, then there is "good" robustness. When some other part fails, and two-thirds of the connections in progress fail, there is "poor" robustness.

In this section, we present a framework for analyzing an internetwork's robustness. The analysis identifies the weak points in an internet and provides an overall robustness measure that may be used to compare different internetworks. We define two practical metrics based on this framework and show ways that the computational demands of the analysis may be reduced. Finally, we extend the framework to handle double faults.

The first step is to quantify the effects of a failure. We use the term "failure" to mean one or more faults in the internetwork. Each fault is represented in the topology by the removal of a single node or edge. The techniques presented here are most easily applied to single faults, but the underlying concepts are also applicable to multiple-fault failures.

A *failure metric* is a function that describes how well the internet survives a particular failure. For example, one failure metric might measure the percentage of networks able to communicate with each other. It might have a value of one if all networks could communicate, and zero of none could communicate.

The individual failure metrics can be combined by taking a weighted sum over some set of failures. The resulting *robustness metric* gives an overall measure of the ability of the internetwork to continue to function after a failure. For the above example, the robustness metric will be close to one if most networks can continue to communicate after a failure, and close to zero if failures tend to disconnect a large fraction of the networks.

There are three choices to be made when defining a robustness metric using the above model:

Failure Metric

f_δ The measure of the effect of the failure δ. The choice of f determines what attributes of network performance are classified as important. If, for example, f only measures the percentage of terminal server traffic disrupted by a failure, then other important effects may be ignored. This is an extreme example, but many different metrics are possible, and if there is any difference between the resulting values for the same failure, δ, then they *must* measure different things.

Weights

w_δ Represents the contribution of the corresponding failure's metric value to the overall robustness metric. It might be selected to correspond to the probability of the failure, the expected time to repair, the number of phone calls likely to be triggered by the failure, etc.

Failure Set
$F = \{ \delta_1, \delta_2, \delta_3, \ldots \}$

The set of failures considered when computing the robustness metric. This choice may be constrained by limitations of the data available or the cost of computing f_δ over the entire set. If devastating failures are left out, or large classes of irrelevant failures are included, then the robustness metric will not provide an accurate picture. Inclusion in F is related to the choice of w_δ, since exclusion from F corresponds to $w_\delta=0$.

The choice of these three components makes a statement about what is considered important in the internetwork — how the "correct operation" of the internet is to be judged.

In terms of the above components, a general formulation of a robustness metric is:

$$\textit{Robustness Metric} = \frac{\sum_{\delta \in F} w_\delta \cdot f_\delta}{\sum_{\delta \in F} w_\delta}$$

There are many possible weightings for robustness metrics, depending upon what is considered important. Here are a few factors that the weighting might incorporate:

- Consider what percentage of the traffic in the internet is disrupted by a failure. "Traffic" may be measured in terms of bytes, packets, or connections.
- Weight the failures by their probability of occurrence.
- Weight the failures by their probable duration.
- Weight the disrupted traffic by its importance, however that might be defined.

A potentially serious problem with these metrics is the amount of data that may be needed to compute them. At a minimum, the topology of the internet must be known to judge the effects of a failure. Traffic volume may be available and would probably be fairly accurate. Failure probabilities are, at best, difficult to obtain with any accuracy, and, at worst, are very subjective. Any rating of the importance of traffic is completely subjective.

We will consider a robustness measure based on one failure metric, the *Partition Fraction*. It is based strictly upon the topology, and weights failures by the number of networks still able to communicate with each other in the presence of a failure.[1] We will also discuss ways to reduce the size of the failure set, without adversely affecting the validity of the robustness metric.

5.1. Partition Fraction

The first failure metric is the *partition fraction*, $PF(\tau,\delta)$, the fraction of network pairs in τ that can still communicate with each other after a particular failure, δ, has occurred. It can be computed as follows: For an internet with N networks, each network can communicate with $N-1$ other networks, for $\frac{N(N-1)}{2}$ pairs (counting each pair once rather than twice). In addition, each network can also communicate with itself. The number of communicating network pairs is therefore

$$\frac{N(N-1)}{2} + N = \frac{N(N+1)}{2}.$$

When an internet is partitioned, the number of communicating network pairs is the sum of the numbers for the partitions. So, for a given failure, all that needs to be known to compute this metric are the sizes of the partitions produced by the failure. For the failure δ, let the number of networks in each fragment be $S_{\delta i}$. The partition fraction for this failure is

$$PF(\delta) = \frac{\sum_i S_{\delta i}(S_{\delta i}+1)}{N(N+1)}$$

The sum of the $S_{\delta i}$ is not necessarily equal to N, since the failure may be in one of the networks, in which case that network will not be included in any of the fragments. Therefore, $PF(\tau,\delta)$ will be slightly less than one if δ included a failed network, even if τ was not partitioned.

As an example, we will compute PF for all possible single failures of the example topology given in figure 2-1. The topology is symmetric around network 1. Therefore, there are five classes of single failures to be analyzed:

- The hub network, #1.
- Each of the leaf networks, #2, 3, or 4.
- Each of the gateways.
- An interface connecting a gateway to network 1.
- An interface connecting a gateway to a leaf network.

$PF(\delta)$ is computed by first determining the set of $S_{\delta i}$ for the failure, and then applying the formula above. Table 5-1 gives values for the $S_{\delta i}$ and the resulting $PF(\delta)$ for the example topology, τ, and for a similar topology, τ', where each gateway connects two leaf networks instead of just one.

Failure δ	$S_{\delta i}(\tau)$	$PF(\tau,\delta)$	$S_{\delta i}(\tau')$	$PF(\tau',\delta)$
Network #1	{ 1, 1, 1 }	.3	{ 2, 2, 2 }	.321
Other network	{ 3 }	.6	{ 6 }	.75
Any gateway	{ 1, 3 }	.7	{ 5, 1, 1 }	.607
Interface to net #1	{ 1, 3 }	.7	{ 2, 5}	.643
Other interface	{ 1, 3 }	.7	{ 1, 6 }	.786

Table 5-1: Analysis of Topology in Figure 2-1

As we have defined it here, the partition fraction only has a boolean concept of connectivity. *i.e.* nodes are either connected or they aren't. It would be useful to extend this model to take into account the available bandwidth between two nodes. In widely dispersed local-area-internetworks,

[1] Although we have a considerable amount of traffic volume data, we haven't studied a metric that weights failures by the amount of traffic disrupted by the failure because this data only covers about eighty percent of the gateways and none of the bridges. In addition, it turns out to be difficult to infer what fraction of the traffic through a gateway or network could *not* find alternate routes when that component fails — there many be many routes in use through the failed node, and alternate paths may exist only for some of them.

communication links may vary in bandwidth by several orders of magnitude, and some faults may decrease the available bandwidth from, for example, 10 Mb/sec to 56 Kb/sec. We leave this as future work.

The partition fraction measures how well an internetwork will survive failures in its components, using a model of connectivity that has a solid practical basis and is intuitively clear. It is reasonable to compute *PF* for all possible single faults in an internet, and from that obtain an expected value over all single failures. This expected value may be used to directly compare the robustness of different internetworks. Besides providing an overall metric, the analysis can be used to identify faults that have especially serious effects. This information can be used to guide modifications to the topology.

5.2. Failure Set Trimming

The simplest choice of a failure set is to include all nodes and edges, or perhaps just all nodes, in the topology. This set includes many faults that aren't very interesting, because their effects are minor and very predictable. The resulting robustness metric will require more work to compute, and will be diluted by the effects (or, rather, the non-effects) of the uninteresting failures.

The prime candidates for exclusion from the failure set are the leaf nodes. In a local-area internetwork, these are networks, and the effect of one failing is entirely predictable: other than that network, nothing else is affected. *Near-leaf* gateways that are connected only to leaf networks plus one link to the main body of the internet have similarly predictable effects. In the internetworks that we studied, over half of the networks were leaves, and almost half of the gateways were near-leaves.

It would be possible to further extend this method, and eliminate from the failure set any trees whose root is attached to the main body of the internet. At some point, however, the pieces eliminated would be large enough that the fault effects would no longer be small enough to be ignored. Also, there may not be much additional savings: in the internets that we studied, only about ten additional nodes would have been eliminated, compared to about sixty leaves and near-leaves.

5.3. Double Faults

The general framework given above for computing robustness measures works well for single faults, but there are several problems to be overcome when failures consisting of multiple faults are considered. As we will see below, most of these problems can be overcome for double faults in internetworks of moderate size. The complications introduced by multiple faults include:

- There are O(N^k) possible *k*-multiple faults in a network with *N* possible single faults. The large size of the failure set, even for small *k*, may make computation of an overall robustness metric too expensive.
- Multiple physical faults are often linked in unexpected ways, making the problem not one of multiple independent faults, but multiple, correlated faults. The usual cause for this is shared facilities such as power or conduit space.[2]

  Because of these linkages, joint probabilities are needed to weight the failures. For such failures, it is difficult to even imagine the range of problems that would cause them, much less assign probabilities to them. The problems usually appear to be one-shot events — a backhoe cutting a bundle of cables, power failure in one building, building air conditioning failure, etc. — and the same failure rarely recurs. By comparison, probabilities for single failures are easy to obtain.

Because many failures are linked, it is important to understand what multiple faults are likely to occur in a particular internet topology, and how those fault sets will affect connectivity. It is usually impractical to take a "shotgun" approach and compute a robustness metric for all possible fault sets — most of those fault sets won't ever happen, and the ones that are likely can only be determined by looking at factors not readily visible in the topology, such as shared resources.

Because of these factors, the concept of an overall robustness metric is not very useful — it would either be too expensive to compute or would require input data that isn't available. A better approach is to search for multiple faults whose effects are interesting, and present them for manual analysis. We will consider only double faults. These techniques extend directly to higher-order faults, but get much more expensive to compute.

The number of fault pairs to be analyzed can be greatly reduced by trimming nodes out of the failure set, as described in section 5.2. In this case, a double fault is not considered if *either* of the component faults is a leaf or near-leaf. In the topologies that we studied, this reduced the number of fault pairs to be considered by a factor of 7 to 9.

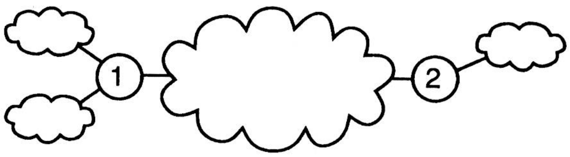


Figure 5-1: An Independent Fault Pair

[2] One of the more spectacular examples of how multiple, apparently independent, systems were disabled by a single event occurred on December 12, 1986, when a construction crew dug through a fiber optic cable running between Newark, New Jersey and White Plains, New York. As it turned out, seven Arpanet links, including all of the cross-country links into the Northeast US, went through this cable, causing a major portion of the Arpanet to be disconnected. It was not obvious, from looking at a geographic map of the sites connected by those links, that they shared a common point of failure. In fact, no one really knew where the physical communications channels went, because the Arpanet links were just leased from the telephone company. This partition lasted for 11 hours, the time it took AT&T to restore service [4].

Most possible fault pairs are *independent* — their effects do not interact. Consider figure 5-1, which shows two possible faults, labeled 1 and 2. If either fault occurs by itself, the internetwork is broken into three or two fragments, respectively. If both faults occur, the internet is broken into four fragments. The effects of the two faults are just additive. Independent fault pairs are not particularly interesting because nothing "new" has happened just because both faults occurred at the same time.

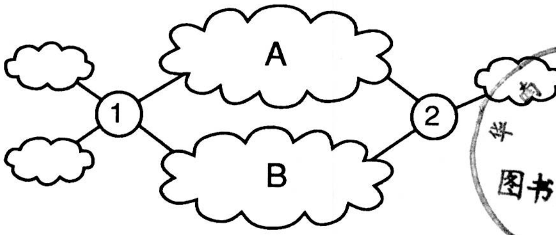


Figure 5-2: A Dependent Fault Pair

Dependent fault pairs are more interesting, however. Consider figure 5-2, which shows a similar internetwork. In this case, the effects of either fault by itself is the same as before, because the large fragments (A and B) are still connected by the non-failed node. If both faults occur together, however, the two fragments are disconnected, and the effects of the faults have been compounded.

Dependent fault pairs can be detected very easily. If two faults are independent, then the number of fragments caused by the fault pair will be one less than the sum of the number of fragments caused by each of the faults alone. If the faults are dependent, the number of fragments produced by the fault pair will be greater. In the topologies that we studied, the number of dependent fault pairs was about two orders of magnitude less than the number of possible pairs, leaving between 20 and 50 fault pairs to be inspected by hand.

Many of these fault pairs are not interesting because the two faults are not causally linked. As pointed out above, however, determining such linkages requires extensive physical knowledge and intuition. Given the relatively small number of dependent faults, manual inspection is reasonable.

6. Measurements of the Stanford Internetwork

The Stanford University Network (SUNet) was monitored from August, 1986, to April, 1988. During this time, changes in the topology were monitored and recorded. Two network maps, at the end of this paper, show the topology of the Stanford internet during this time. The map is laid out geographically, with gateways and bridges placed close to their true positions. South is approximately at the top of each figure. Each map also includes a table giving the number of gateways, bridges, and networks in the internet, as well as calculations of K_G, K_N, and Q.

During this period, 54 long- or medium-term changes were made to the topology of the internetwork. Many other short-term or transient changes were observed, but these are ignored for the purposes of these analyses. Several special-purpose or experimental gateways and networks are consistently omitted from the topology in order to provide a stable baseline for comparison.

During this period, two major changes were made:

- The *link-net*, a 10-Mbit Ethernet backbone built out of Ethernet bridges was expanded to cover most of campus. The bridges are shown as squares connected together by heavy lines. This is the "composite network" mentioned in the table printed with the maps.
- Several older experimental 3-Mbit Ethernets were taken out of service or severely pruned. These were nets 33, 37, 40, and 48.

We now analyze the impact of these changes using the techniques described earlier in this paper.

6.1. Redundancy Results

Figure 6-1 is a plot of (K_G,K_N) for the Stanford internet during this period. The three curves indicate different amounts of redundancy. Each point, marked with a '+', represents one distinct value that (K_G,K_N) took on. The state as of August, 1986 is marked with a circle, and the state as of April, 1988 is marked with a diamond.

(K_G,K_N) shifted quite a bit during the period. The most dramatic change is that K_G went down from 3.0 to 2.7, with a peak of 3.15 in between. This change is significant because the lowest value that K_G can have is 2.0. Some of this change is due to the addition of four bridges, each of which has two interfaces. The rest of the change isn't attributable to any one cause, and seems to represent a general trend toward gateways with fewer interfaces.

The redundancy factor shows a general downward trend, with a peak of 12 percent redundant connections in late 1986, ending at about 6 percent in May, 1988. This is a result of the general trend toward a backbone topology from a less structured mesh. Notice that, for any single change in the topology, the shift of (K_G,K_N) is very slight. This is to be expected, since most changes consisted of the addition or deletion of a single interface, gateway, or network. In the next section, we will show that changes in robustness, for the same small changes in topology, were very dramatic.

6.2. Partition Fraction Results

Studying the behavior of the partition fraction produces more insight into the characteristics of the internet. Figure 6-2 shows the robustness metric based on the partition fraction averaged over two single-fault failure sets: all non-leaf networks and all non-near-leaf gateways. In addition, the raw partition fraction values for the ten most disruptive single faults are shown.

The vertical axis shows $1-PF$, so "no effect" is at the bottom and "full partition" is at the top. The ten worst values of PF are shown shaded, with the most disruptive shaded light gray and the least disruptive shaded dark gray. The interpretation of this