Dynamic Host Routing for Production Use of Developmental Networks

Joe Touch and Ted Faber

touch@isi.edu, faber@isi.edu
USC / Information Sciences Institute

Abstract[1]

Many high-speed networks are in a development stage, compared to established technologies like Ethernet. Use of these networks for production service provides substantial benefits to the developers, e.g., for large-scale testing. However, most users are not tolerant of pre-production network outages, even in exchange for the higher bandwidth these networks afford. This paper describes ISI's experience in deploying such a network, Myricom's Myrinet, for production service. A configuration is described that encourages participation in advanced deployment, and provides dynamic backup to a conventional, production-quality network. Each machine individually and transparently reverts to a stable backup network in the event that the prototype network fails. The system is composed of existing tools, and requires no operating system modifications. Alternate configurations are discussed, and recommendations presented to support such deployment more easily in the future.

1: Introduction

In the course of networking research, it is common for a prototype system to reach a state where it is sufficiently stable that only a large scale installation under real workloads will reveal more errors in the system or information about its scaling properties. However, users accustomed to a stable production system are reluctant to switch to a potentially error-prone prototype system. This paper describes a network utilizing dynamic routing that allows transparent use of a prototype network, and resorts to a stable production network in the event of errors. The system makes no operating system modifications, and supports both unicast and multicast traffic; it has been in use at the University of Southern California/Information Sciences Institute (ISI) for more than a year.

The ATOMIC-2 research group at ISI is exploring the issues associated with introducing the ATOMIC LAN, a 640 Mbps LAN invented at ISI, into a distributed computing environment [12]. The research focuses on optimizing existing software to use the increased bandwidth offered by this new technology. The phrase "ATOMIC LAN" denotes the general technology; the particular implementation in use at ISI is manufactured by Myricom, under the name "Myrinet."[1] Although there is much to learn from laboratory experiments and measurements, a large component of testing a high speed LAN, like ATOMIC, is subjecting it to real users. Users produce traffic patterns that are difficult to replicate in the laboratory, and expose unexpected errors in system software.

A goal of the ATOMIC-2 project was to acquire such a user base, by installing the ATOMIC LAN as the default network for the entire Computer Networks Division at ISI. The division consists of other researchers who rely on a production network service. Even this sophisticated user group required reassurance that their own work would not be disrupted at the expense of this experiment.

Gaining and maintaining user confidence requires a system that is simple, reliable, and transparent. Our solution favors using the advanced network when available, as aggressively as possible. The resulting network configuration tries to behave no worse in the event of an ATOMIC net failure than the existing system does in the case of a rebooting file server, to which current users are already accustomed. The configuration also makes minimal changes to the system interface; specifically, no host names change, and the user is not responsible for any more network configuration effort than in the conventional ISI environment.

The remainder of this paper describes the system that meets these requirements in detail. Section 2 makes the case for eager deployment of pre-production networks,

Section 3 provides context and background on the general dual-homing issue and dynamic routing in general, Section 4 presents our solution, Section 5 presents alternatives and discusses benefits and detriments of each, Section 6 evaluates the utility of our solution, and Section 7 summarizes our conclusions.

2: The case for early deployment of pre-production networks

There are several benefits to using a prototype (pre-production) network for production service. In general, as networks range from production to experimental, their bandwidth increases but their reliability decreases (Figure 1).
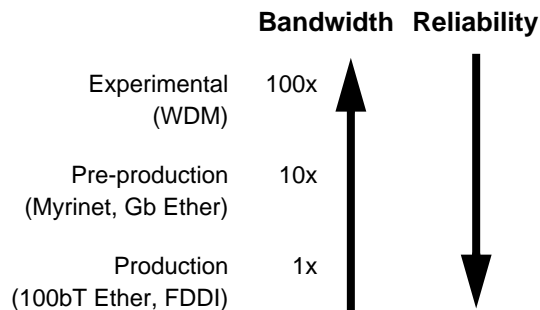
Bandwidth Reliability

Experimental (WDM)	100x
Pre-production (Myrinet, Gb Ether)	10x
Production (100bT Ether, FDDI)	1x

FIGURE 1. Comparative features of phases of networking development

By installing pre-production networks as a production service, experimenters are able to discover issues in scale and reliability that only real users can evoke. There are problems that have surfaced in the ISI ATOMIC LAN installation that have occurred only because it is the largest and most diverse user community using the ATOMIC LAN in an Internet Protocol (IP) network.

	Require	Prefer
No reboot	✓	
No OS changes	✓	
User controllable	✓	
Outage time low (mins)	✓	
Save current connections		✓?
Keep host IP number		✓?
Use net layer protocols		✓?

FIGURE 2. Summary of requirements (✓ = require, ✓? = prefer)

Users at ISI generally favor new technologies, but only if their work does not suffer excessively, so a reasonable solution must mask the reliability weakness of these new networks. It is generally acceptable for connectivity to be briefly interrupted while the system switches from the prototype network to the backup if no work is lost. System reboots should be avoided, as should loss of existing connections, because these can easily result in lost work for users. Network outages due to switchover should be isomorphic to file server stalls, which are unfortunately already tolerated as brief weekly occurrences on typical distributed installations. It is also beneficial if users have control over their own connectivity. These constraints are listed in Figure 2.

2.1: The ATOMIC LAN

The ATOMIC LAN is a 640 Mbps network composed of point-to-point full-duplex links, interconnected by simple switch boxes [12]. The links use byte-wide parallel transmission over twisted-pair copper wires, with additional signalling wires. The switches provide a small amount of buffering (several bytes) and otherwise rely on source-routed cut-through switching in hardware. The ATOMIC LAN was developed at ISI in conjunction with CalTech, and the technology has been spun off to an independent commercial venture called Myricom [1]. Myricom markets their implementation of the ATOMIC LAN under the name 'Myrinet'; we use ATOMIC LAN to refer to the technology in general, and Myrinet to refer to Myricom's particular design.

The ATOMIC LAN has properties that distinguish it from conventional LAN systems. It relies on a concatenation of point-to-point links for LAN paths, and so is susceptible to intermediate link or switch failures. Wiring distance limitations mandate a distributed installation, including in-ceiling switches with separate power. The links are composed of a pair of electrically distinct simplex media, and partial failure is common. This can result in asymmetric failures, *e.g.*, a host that can receive messages, but not respond to them.

The Myrinet supports broadcast via serial unicast at the source host interface. Multicast management protocols cannot assume global ordering of broadcast messages. In addition, partitioning of the net results in localized islands, rather than single isolated hosts.

3: Dual-homing vs. dynamic routing

Installing a pre-production LAN for production use is facilitated by the support of a backup production LAN. When the new LAN fails, the backup LAN provides connectivity (Figure 3). The main difficulty with this configu-

ration is that it requires dual-homed hosts, *i.e.*, hosts with two network attachments. The problem therein lies with the notion of an IP address.
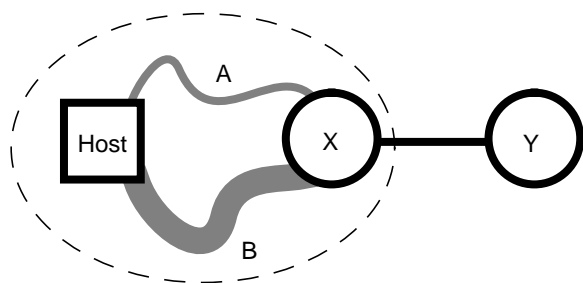

FIGURE 3. Dual-homing of a host

Traditionally, hosts have a single IP address, that of their sole network connection [2]. Hosts with multiple connections may use a single IP address or multiple IP addresses (Figure 4). A single host IP address is bound dynamically (via the Address Resolution Protocol, ARP [11]) to one of multiple interfaces; this is known as link-layer multiplexing [2]. A host alternately may use multiple IP addresses, bound to each interface; this is known as multi-homing. Although link-layer multiplexing is a somewhat less clean solution than multi-homing, it is nonetheless both common practice and documented in Internet standards [2].

The difference between the two is whether the host is known by a single IP address or multiple IP addresses, and whether the dynamic switchover occurs at a link or network layer. Link layer dynamic methods are based on variant uses of the ARP protocol, discussed in Section 5.
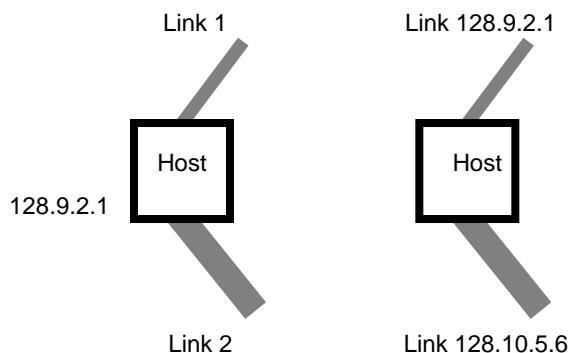

FIGURE 4. Link multiplexing vs. multi-homing

The issue is complicated by the use of Domain Name System (DNS) names [9] [10]. The DNS resolves names to IP addresses. When multiple IP addresses are available for a host, a list is returned, although most DNS clients recognize only the first address in the list. Lookups return-

ing multiple addresses are more commonly used for replicated hosts, so that each client consequently contacts a different host, distributing load, in round-robin fashion (Figure 5).
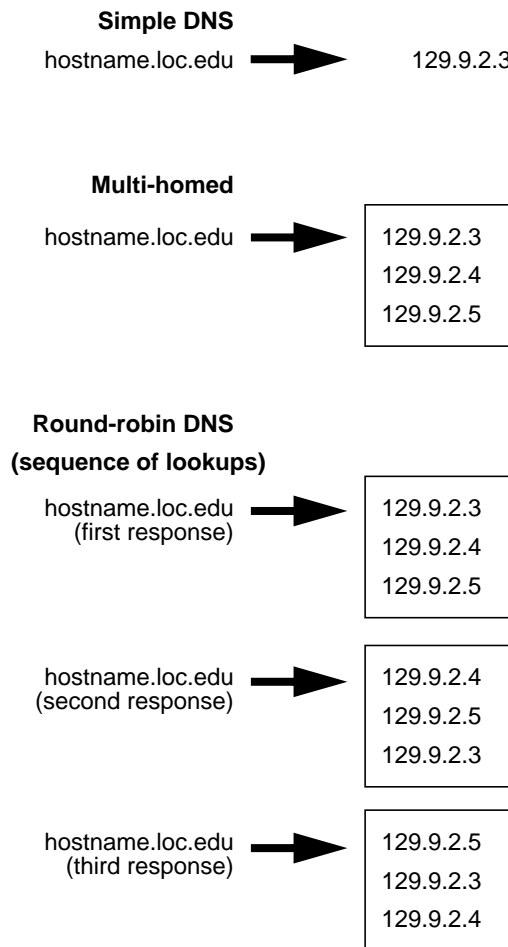

FIGURE 5. DNS variants for multihoming

In our environment, it would be useful to refer to a dual-homed host by a single name, but be able to use multiple IP addresses for that host, depending on which network is available. However, using DNS for this purpose would require disabling client-side DNS caching, which would drastically impact performance. The idea would also break connections to an IP address when it is not available. We would like to have our hosts use one IP address per interface, to support network layer dynamic switchover, but to also have a single primary IP address that is always reachable, simplifying the allocation of DNS names.

Given a proper solution to the naming issue, we also require a way to select the currently active network. Existing solutions to dynamic routing problems assume modifi-

cation of network paths, where the resulting link changes are not visible end-to-end (Figure 6). This differs from the dual-homed case, where link changes affect the endpoint addresses (Figure 3), which is why the DNS issue arises.
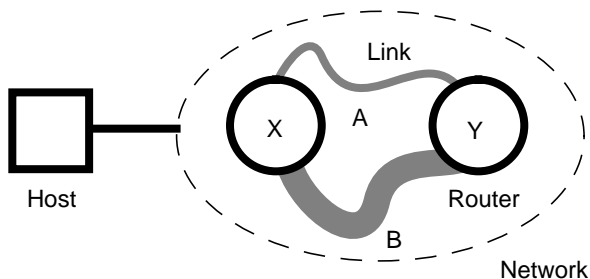

FIGURE 6. General dynamic routing

One way to use network dynamic solutions in the host case is to move the host one step further away from the network, implicitly inserting a router inside the host (Figure 7). In this solution, a host has its own IP address (H1), and each interface also has a separate IP address (H2, H3). Virtual interfaces are used to attach the host's IP address to a non-existent internal interface, and the host acts as a router among the different addresses; these interfaces are supported via multiple IP addresses per interface in recent OS's (Solaris, FreeBSD) or by OS modifications (SunOS) [6].
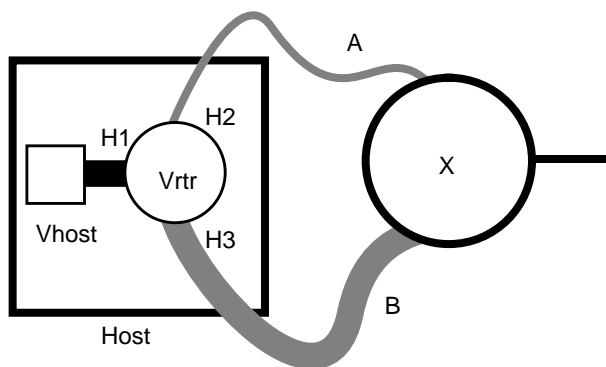

FIGURE 7. Virtual interface makes dual-homing a network issue

Finally, dynamic network layer routing is based on protocols that assume that links are either point-to-point (individual) or broadcast media, e.g., RIP [7]. RIP is not guaranteed to function in the presence of asymmetric link failures, which are possible in the ATOMIC LAN. The critical issue is how badly RIP fails under these conditions, and how frequent they are.

4: Our solution

For the ATOMIC LAN we developed a variant of the virtual interface solution, in which a host has a single preferred IP address, and emulates a router internally, but no virtual interface support is required in the OS (Figure 8). Our particular configuration is shown in Figure 9.
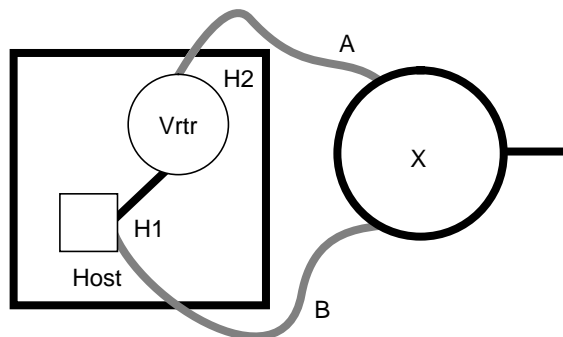

FIGURE 8. Solution (avoids virtual interfaces)

In our network, each host has three names – host, host-a, and host-s. The default names are aliases for the '-s' names (*i.e.*, X = X-s), which are IP addresses on the (slow) ethernet. The '-a' names are the interfaces on the ATOMIC LAN. Each host serves as its own router, as in Figure 8. RIP is run only on the ATOMIC LAN, and the default routes use the slow ethernet. Every host and gateway on the ATOMIC LAN runs *gated* to implement RIP and manage its local routing table [5]. The particular configurations are listed in Appendix A.
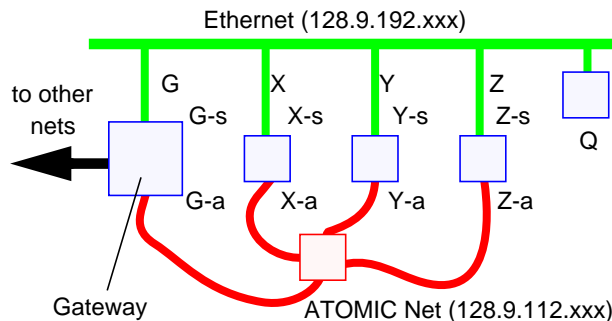
FIGURE 9. ISI's ATOMIC LAN configuration

As a result, hosts on the ATOMIC LAN and the ATOMIC gateway host each contain a default route over the ethernet to the gateway, and host routes that override the default route and use the ATOMIC LAN. *E.g.*, X has a default route to G-s, and host routes advertised in RIP for G via G-a, X via X-a, Y via Y-a, and Z via Z-a. When X attempts to send a packet to Z, that packet is sent to Z-a out X's X-a interface.

Each host periodically emits broadcast RIP packets on the ATOMIC LAN (approximately every 30 seconds); hosts receiving those RIP packets retain that host route, and routes to silent hosts are removed (silent over 3 minutes). In this way, RIP is confined to the ATOMIC LAN, not affecting hosts on the ethernet that are not dual-connected, such as laptops sharing an office ethernet connection with an ATOMIC host (Q in Figure 9). The configuration maintains maximal use of the ATOMIC LAN, and reverts hosts to the ethernet only when an ATOMIC LAN path is not available. The configuration also supports disconnected islands of ATOMIC connectivity, where the ATOMIC LAN is used within the island, and the ethernet provides backup to the rest of the LAN.

RIP does not guarantee correct operation in the presence of asymmetric failures. Our experience is that such failures are rare and easily diagnosed, but that this must be kept in mind when diagnosing routing anomalies.

Our solution achieves all of the previously described necessary requirements (Figure 2), and most of the preferred requirements (Figure 10). It is unable to maintain TCP connections, because SunOS labels packets with the IP number of the outgoing interface. When the preferred outgoing interface changes, the label would need to change to maintain routing; because they cannot be changed mid-connection, the connection is preserved only when established on the ethernet interface first. If the connection begins on the Myrinet and the path changes to use the ethernet, a return path using the Myrinet is not guaranteed, and the connection can fail. This can be solved by running RIP on the ethernet as well as on the Myrinet, but we chose not to expose non-experiment machines (on the ethernet) to RIP traffic unless necessary.

5: Alternative solutions

There are several other ways to provide switchover to a backup LAN for dual-connected hosts. Here we evaluate alternative solutions with respect to our requirements.

Users could manually reconfigure their hosts when a network outage is noticed. This requires user action, as well as a manual reboot. For most users, this is an unacceptable solution, because of the amount of desktop state lost during a reboot. Although users in our division are sophisticated on the research issues of networking, most do not want to know the details of the daily operation of their workstation, sufficient to manage such a reboot.

The entire network could be remotely controlled, such that all hosts are switched between the development or backup networks as a group. Given that there are 53 network connections in our division, requiring all-or-nothing use of the developmental LAN would likely result in effectively avoiding that LAN altogether.

Scripted solutions were considered, *e.g.*, using *ping* to discover connectivity and reconfigure the route table as required. At first glance these appear simple and potentially more reliable than full-blown dynamic routing protocol daemons. However, it is likely that any script solution is liable to rediscover the flaws of early dynamic routing software, *e.g.*, broadcast storms of control messages. Further, it is difficult to make a scripted solution perform atomic (indivisible) updates of the routing table. The routing table can thus become inconsistent, resulting in unrecoverable host failures.

Link protocol solutions were also considered. ISI has traditionally favored network-based, IP-level solutions. However, a prevalent alternative is to use proxy ARP, where one interface response on behalf of another to an ARP request [3]. Proxy ARP can be used to emulate dynamic routing [4]. However, ARP entries are refreshed locally whenever referenced, which results in misrouting when a link address 'changes' before the ARP entry becomes stale (IP destination is mapped to a different link address). Recently proposed UN-ARP[8] provides a way to flush such stale entries, at the expense of lost connectivity until the ARP entry is restored. UN-ARP is experimental, and would require OS upgrades to deploy at ISI. Recall that OS upgrades were not desired. Our division also commonly uses UDP-based applications that refresh the local ARP caches frequently, *e.g.,* video conferencing.

	Goal	Ours	Manual	Remote	Script	Link
No reboot	✓	✓	X	✓	✓	✓
No OS changes	✓	✓	✓	✓	✓	✓
User controllable	✓	✓	✓	X	✓	X
Outage time low (mins)	✓	✓	X	X	X	X
Save current connections	✓?	X	X	X	X	X
Keep host IP number	✓?	✓	X	✓	✓	✓
Use net layer protocols	✓?	✓	X	X	X	X

FIGURE 10. Effectiveness of solutions
(✓ = require/provide, ✓? = prefer,
X = does not provide)

A summary of the features of these solutions and how they compare is shown in Figure 10. Most other solutions do not have low outage times None maintain current connections; this would likely require a modification to the kernel to avoid the 'outgoing packet label' problem described earlier. The figure does not indicate the complexity of the various solutions because it was not an explicit requirement, although it was a major factor in selecting our solution over the others.

6: Evaluation

The ISI Computer Networks Division has used this configuration of the ATOMIC LAN as its preferred network for all IP connectivity for over a year. There are currently 41 hosts, including both SunOS and Solaris Sun SPARCs. We are currently working to port the Myrinet drivers to the FreeBSD operating system, further augmenting the installation. There have been various hardware failures (switches, host interfaces) during this period, as well as a network-wide failure of the Myrinet, due to a bug in an upgraded driver. None of these interruptions affected connectivity for users, as the backup ethernet performed as expected.

We found that the psychological impact of user autonomy was critical to acceptance. When the ATOMIC LAN was initially installed, every outage, regardless of how minor, was attributed to the new technology. In the vast majority of cases the outage was due to network file server stalls on a network not on the ATOMIC LAN. These server outages were tolerated prior to the new installation, but afterwards were deemed unacceptable, mainly because users could not distinguish between LAN failures and server failures. One important feature of the dynamic system design was to provide users with a test they could perform in their own office, *i.e.*, unplug the Myrinet cable, to determine whether the new technology was responsible for the outage. Given this autonomy, most users are willing to remain on the new network.

We discovered the complexity of installing and maintaining a separate router for network isolation. The ATOMIC LAN's router is a host-based gateway which, due to link length limitations, could not be located in the computer support center. As a result, our group was responsible for maintaining this server, which involves non-trivial ongoing effort. We also discovered the complexity of maintaining additional IP addresses for each host, an artifact of the lack of convenient tools for shared DNS management.

The RIP protocols used to manage the routing are not well suited to the ATOMIC LAN. In particular, RIP assumes link symmetry, that "if I can hear you, I can speak to you." For many new link technologies, Myrinet included, this assumption fails, and can result in asymmetric routes or loss of connectivity altogether. Additionally, our deployment of RIP is fairly large, involving over 40 'routers' exchanging information.

Our configuration also supports IP multicast dynamic routing in addition to unicast. A dummy host route, for 224.0.0.0, is used to indicate the preferred interface for IP multicast associations. We include this default route in the *gated* configuration (Appendix A), and also run *mrouted* (the IP multicast routing daemon) on each host to dynamically manage the multicast routing. This is also a very large *mrouted* system, with 40 'multicast routers' sharing multicast route state.

Finally, there is a complication with using both paradigms of IP addressing – per host and per interface. When we emit a packet out the ATOMIC interface, it is labelled with the IP address of the outgoing interface. This can result in asymmetric routes or asymmetric failures when the new LAN is partitioned into islands. A better implementation would allow a host to emit packets with a desired, static IP source address, regardless of the outgoing interface.

Our experience indicates a network of dual-homed hosts configured as routers is functional and the technique is useful for providing transparent backup, given a small physical scale. We feel that this technique would benefit from virtual interfaces, and hosts that support sending packets with pre-configured outgoing IP addresses, regardless of outgoing interface.

7: Summary

The ATOMIC-2 project designed a configuration of a dynamic network layer routing system for use in a dual-homed LAN where it supports over 40 hosts. This system allows a pre-production network to be deployed in a production environment and to support the daily networking needs of an entire research division. Although there were several candidate solutions, only one, using a full dynamic routing protocol at each host, provides the desired fault tolerance and automation.

Our experience is that networking protocols can be deployed effectively on large numbers of hosts in a LAN. The configuration was non-trivial, requiring consideration of OS packet labelling, multicast protocols, and the properties of teleconferencing applications. The overall solution using RIP in gated, is sufficient, but requires care in diagnosing routing anomalies, because it is not well suited to emerging asymmetric link technologies, such as ours.

The experience would have been less cumbersome if operating systems supported virtual interfaces (not mapped to a specific interface). This includes the ability to allow packet source addresses to be determined independent of the interface they exit (avoiding the packet labelling problem). Finally, the Internet protocols should more directly support multi-homed hosts, rather than by artifact.

The authors appreciate the assistance of ISI's Information Processing Center, notably Jim Koda and Richard Nelson for comparisons to link-multiplexing solutions, and the indulgence of ISI's Computer Networks Division who have served as beta-testers for this solution, and the help of Ramesh Govindan and Bill Manning for specifics of the *gated* configuration and debugging.

8: References

[1] Boden, N., Cohen, D., *et al*, "Myrinet – A Gigabit-per-Second Local-Area Network," IEEE-Micro,Vol.15, No.1, February 1995, pp.29-36.
<http://www.myri.com/research/publications/Hot.ps>.

[2] Braden, R., editor, "Requirements for Internet Hosts – Communication Layers," Network Working Group STD-3, RFC-1122, USC/Information Sciences Institute, Oct. 1989.

[3] Braden, R., and Postel, J., "Requirements for Internet Gateways," Network Working Group RFC-1009, USC/Information Sciences Institute, June 1987.

[4] Carl-Mitchell, S., and Quarterman, J., "Using ARP to Implement Transparent Subnet Gateways," Network Working Group RFC-1027, Texas Internet Consulting, Oct. 1987.

[5] Hunt, C., "Appendix C: A gated reference," in *TCP/IP Network Administration,* O'Reilly & Associates, Inc., CA, May 1994.

[6] Ioannidis, J., "Virtual Interface (VIF)" Columbia Univ, 1991, (Unix TAR file,
<ftp://ftp.unit.no/pub/unix/network/vif-1.11.tar.gz>.

[7] Malkin, G., "RIP Version 2 – Carrying Additional Information," Network Working Group RFC-1388, Xylogics, Inc., Jan. 1993.

[8] Malkin, G., "ARP Extension – UNARP," Network Working Group RFC-1868, Xylogics, Inc., Nov. 1995.

[9] Mockapetris, P., "Domain Names – Concepts and Facilities," Network Working Group RFC-882, USC/Information Sciences Institute, Nov. 1983.

[10] Mockapetris, P., "Domain Names – Implementation and Specification," Network Working Group RFC-883, USC/Information Sciences Institute, Nov. 1983.

[11] Plummer, D., "An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware," Network Working Group RFC-826, Symbolics, Inc., Nov. 1982.

[12] Touch, J., Faber, T., Hutton, A., *et al*, "Experiences with a Production Gigabit LAN," IEEE Gigabit Networking Workshop, Kobe, Japan, April 1997
<http://www.isi.edu/atomic2/gbn97/>

Appendix A: Gated config. files

8.1: ash (host) gated config

```
# A generic RIP configuration file.
traceoptions normal parse adv route;

rip on {
   traceoptions detail packets
        request response;
   # This allows us to
   # announce static routes
   defaultmetric 1;

   # We do RIP only over ATOMIC
   interface 128.9.112.118 ripin
        ripout version 2; # ATOMIC
   interface 128.9.192.118 noripin
        noripout; # Slow Ether
};

static {
   # Assign second address to ATOMIC
   128.9.192.118 masklen 32 interface
        128.9.112.118 noinstall;

   # Declare a static default.
   default gateway 128.9.192.72
        preference 20 retain;

   # Host route for IPmcast defaults
   224.0.0.0 masklen 32 interface
        128.9.192.118 preference 20
        retain;
};

import proto rip interface
        128.9.112.118 {
   # We accord a higher preference
   # to default heard over ATOMIC
   default preference 10;
   # We accept all other routes
   all;
};

export proto rip interface
        128.9.112.118 {
   proto static {
        # Announce slowEther host
        route over ATOMIC
        128.9.192.118 masklen 32;
   };
};
```

8.2: idn (gateway) gated config

```
traceoptions "/etc/gated.trace"
      replace size 50k files 3
      normal parse adv route;

#
# Enable rip on slowEther and myri.
#
rip on {
   traceoptions detail packets
      request response;

   # This allows us to
   # announce static routes
   defaultmetric 1;

   # We do RIP only on ATOMIC
   interface 128.9.112.72 ripin
      ripout version 2;# ATOMIC
   interface 128.9.192.72 noripin
      noripout;# Slow Ether
   interface 128.9.160.72 noripin
      noripout;# Fast Ether
};

#
# Statics
#
static {
   # Add host route to our slowEther,
   # to enable announcing it
   # over the ATOMIC interface.
   128.9.192.72 masklen 32 gateway
      128.9.112.72 noinstall;

   # optimized internal routes
   128.9.0.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.16.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.32.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.48.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.64.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.80.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.96.0 masklen 20 gateway
      128.9.160.5 retain;
   # 112 - interface already directly
      routes - atomic-net
   128.9.128.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.144.0 masklen 20 gateway
      128.9.160.5 retain;
   # 160 - interface already directly
      routes - zephyr-net
   128.9.176.0 masklen 20 gateway
      128.9.160.5 retain;
   # 192 - interface already directly
      routes - atomic-s-net
   128.9.208.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.224.0 masklen 20 gateway
      128.9.160.5 retain;
   128.9.240.0 masklen 20 gateway
      128.9.160.5 retain;

   # Add a static default,
   # so we can correctly advertise
   # ourselves as a router for
   # all outbound traffic.
   default gateway 128.9.160.249
      retain;
};

# Import filters: import all RIP
# routes on ATOMIC interface.
#
import proto rip interface
      128.9.112.72 {
   all;
};

# Export filters: export only
# our own static routes.
#
export proto rip interface
      128.9.112.72 {
   proto static {
      # Announce our host route, so
      Atomic hosts will
      # know to get to us using the
      myri net if possible
      128.9.192.72 masklen 32;
      # Announce our default route,
      so myri hosts will
      # use us to get to the zephyr
      net.
      default metric 1;
   };
};
```