                    The Nimrod Routing Architecture

Status of this Memo

Abstract

   We present a scalable internetwork routing architecture, called
   Nimrod. The Nimrod architecture is designed to accommodate a dynamic
   internetwork of arbitrary size with heterogeneous service
   requirements and restrictions and to admit incremental deployment
   throughout an internetwork. The key to Nimrod's scalability is its
   ability to represent and manipulate routing-related information at
   multiple levels of abstraction.

Table of Contents

1. Introduction

   Nimrod is a scalable routing architecture designed to accommodate a
   continually expanding and diversifying internetwork. First suggested
   by Noel Chiappa, the Nimrod architecture has undergone revision and
   refinement through the efforts of the Nimrod working group of the
   IETF. In this document, we present a detailed description of this
   architecture.

   The goals of Nimrod are as follows:

   1. To support a dynamic internetwork of arbitrary size by
      providing mechanisms to control the amount of routing information
      that must be known throughout an internetwork.

   2. To provide service-specific routing in the presence of multiple
      constraints imposed by service providers and users.

   3. To admit incremental deployment throughout an internetwork.

   We have designed the Nimrod architecture to meet these goals. The
   key features of this architecture include:

   1. Representation of internetwork connectivity and services in the
      form of maps at multiple levels of abstraction.

   2. User-controlled route generation and selection based on maps and
      traffic service requirements.

   3. User-directed packet forwarding along established paths.

   Nimrod is a general routing architecture that can be applied to
   routing both within a single routing domain and among multiple
   routing domains. As a general internetwork routing architecture
   designed to deal with increased internetwork size and diversity,
   Nimrod is equally applicable to both the TCP/IP and OSI environments.

2. Overview of Nimrod

   Before describing the Nimrod architecture in detail, we provide an
   overview. We begin with the internetworking requirements, followed
   by the routing functions, and concluding with Nimrod's scaling
   characteristics.

2.1 Constraints of the Internetworking Environment

   Internetworks are growing and evolving systems, in terms of number,
   diversity, and interconnectivity of service providers and users, and
   therefore require a routing architecture that can accommodate
   internetwork growth and evolution. A complicated mix of factors such
   as technological advances, political alliances, and service supply
   and demand economics will determine how an internetwork will change
   over time. However, correctly predicting all of these factors and
   all of their effects on an internetwork may not be possible. Thus,
   the flexibility of an internetwork routing architecture is its key to
   handling unanticipated requirements.

   In developing the Nimrod architecture, we first assembled a list of
   internetwork environmental constraints that have implications for
   routing. This list, enumerated below, includes observations about
   the present Internet; it also includes predictions about
   internetworks five to ten years in the future.

   1. The Internet will grow to include $O(10^9)$ networks.

   2. The number of internetwork users may be unbounded.

   3. The capacity of internetwork resources is steadily increasing but
      so is the demand for these resources.

   4. Routers and hosts have finite processing capacity and finite
      memory, and networks have finite transmission capacity.

   5. Internetworks comprise different types of communications media --
      including wireline, optical and wireless, terrestrial and
      satellite, shared multiaccess and point-to-point -- with different
      service characteristics in terms of throughput, delay, error and
      loss distributions, and privacy.

6. Internetwork elements -- networks, routers, hosts, and processes --
   may be mobile.

7. Service providers will specify offered services and restrictions on
   access to those services. Restrictions may be in terms of when a
   service is available, how much the service costs, which users may
   subscribe to the service and for what purposes, and how the user
   must shape its traffic in order to receive a service guarantee.

8. Users will specify traffic service requirements which may vary
   widely among sessions. These specifications may be in terms of
   requested qualities of service, the amounts they are willing to pay
   for these services, the times at which they want these services,
   and the providers they wish to use.

9. A user traffic session may include m sources and n destinations,
   where m, n > or = 1.

10. Service providers and users have a synergistic relationship. That
    is, as users develop more applications with special service
    requirements, service providers will respond with the services to
    meet these demands. Moreover, as service providers deliver more
    services, users will develop more applications that take advantage
    of these services.

11. Support for varied and special services will require more
    processing, memory, and transmission bandwidth on the part of both
    the service providers offering these services and the users
    requesting these services. Hence, many routing-related activities
    will likely be performed not by routers and hosts but rather by
    independent devices acting on their behalf to process, store, and
    distribute routing information.

12. Users requiring specialized services (e.g., high guaranteed
    throughput) will usually be willing to pay more for these services
    and to incur some delay in obtaining them.

13. Service providers are reluctant to introduce complicated protocols
    into their networks, because they are more difficult to manage.

14. Vendors are reluctant to implement complicated protocols in their
    products, because they take longer to develop.

Collectively, these constraints imply that a successful internetwork
routing architecture must support special features, such as service-
specific routing and component mobility in a large and changing
internetwork, using simple procedures that consume a minimal amount
of internetwork resources. We believe that the Nimrod architecture

meets these goals, and we justify this claim in the remainder of this
document.

2.2 The Basic Routing Functions

The basic routing functions provided by Nimrod are those provided by
any routing system, namely:

1. Collecting, assembling, and distributing the information necessary
   for route generation and selection.

2. Generating and selecting routes based on this information.

3. Establishing in routers information necessary for forwarding
   packets along the selected routes.

4. Forwarding packets along the selected routes.

The Nimrod approach to providing this routing functionality includes
map distribution according to the "link-state" paradigm, localization
of route generation and selection at traffic sources and
destinations, and specification of packet forwarding through path
establishment by the sources and destinations.

Link-state map distribution permits each service provider to have
control over the services it offers, through both distributing
restrictions in and restricting distribution of its routing
information. Restricting distribution of routing information serves
to reduce the amount of routing information maintained throughout an
internetwork and to keep certain routing information private.
However, it also leads to inconsistent routing information databases
throughout an internetwork, as not all such databases will be
complete or identical. We expect routing information database
inconsistencies to occur often in a large internetwork, regardless of
whether privacy is an issue. The reason is that we expect some
devices to be incapable of maintaining the complete set of routing
information for the internetwork. These devices will select only
some of the distributed routing information for storage in their
databases.

Route generation and selection, based on maps and traffic service
requirements, may be completely controlled by the users or, more
likely, by devices acting on their behalf and does not require global
coordination among routers. Thus these devices may generate routes
specific to the users' needs, and only those users pay the cost of
generating those routes. Locally-controlled route generation allows
incremental deployment of and experimentation with new route
generation algorithms, as these algorithms need not be the same at

each location in an internetwork.

Packet forwarding according to paths may be completely controlled by
the users or the devices acting on their behalf. These paths may be
specified in as much detail as the maps permit. Such packet
forwarding provides freedom from forwarding loops, even when routers
in a path have inconsistent routing information. The reason is that
the forwarding path is a route computed by a single device and based
on routing information maintained at a single device.

We note that the Nimrod architecture and Inter-Domain Policy Routing
(IDPR) [1] share in common link-state routing information
distribution, localized route generation and path-oriented message
forwarding. In developing the Nimrod architecture, we have drawn
upon experience gained in developing and experimenting with IDPR.

2.3 Scalability Features

Nimrod must provide service-specific routing in arbitrarily large
internetworks and hence must employ mechanisms that help to contain
the amount of internetwork resources consumed by the routing
functions. We provide a brief synopsis of such mechanisms below,
noting that arbitrary use of these mechanisms does not guarantee a
scalable routing architecture. Instead, these mechanisms must be
used wisely, in order enable a routing architecture to scale with
internetwork growth.

2.3.1 Clustering and Abstraction

The Nimrod architecture is capable of representing an internetwork as
clusters of entities at multiple levels of abstraction. Clustering
reduces the number of entities visible to routing. Abstraction
reduces the amount of information required to characterize an entity
visible to routing.

Clustering begins by aggregating internetwork elements such as hosts,
routers, and networks according to some predetermined criteria.
These elements may be clustered according to relationships among
them, such as "managed by the same authority", or so as to satisfy
some objective function, such as "minimize the expected amount of
forwarding information stored at each router". Nimrod does not
mandate a particular cluster formation algorithm.

New clusters may be formed by clustering together existing clusters.
Repeated clustering of entities produces a hierarchy of clusters with
a unique universal cluster that contains all others. The same
clustering algorithm need not be applied at each level in the
hierarchy.

All elements within a cluster must satisfy at least one relation,
namely connectivity. That is, if all elements within a cluster are
operational, then any two of them must be connected by at least one
route that lies entirely within that cluster. This condition
prohibits the formation of certain types of separated clusters, such
as the following. Suppose that a company has two branches located at
opposite ends of a country and that these two branches must
communicate over a public network not owned by the company. Then the
two branches cannot be members of the same cluster, unless that
cluster also includes the public network connecting them.

Once the clusters are formed, their connectivity and service
information is abstracted to reduce the representation of cluster
characteristics. Example abstraction procedures include elimination
of services provided by a small fraction of the elements in the
cluster or expression of services in terms of average values. Nimrod
does not mandate a particular abstraction algorithm. The same
abstraction algorithm need not be applied to each cluster, and
multiple abstraction algorithms may be applied to a single cluster.

A particular combination of clustering and abstraction algorithms
applied to an internetwork results in an organization related to but
distinct from the physical organization of the component hosts,
routers, and networks. When a clustering is superimposed over the
physical internetwork elements, the cluster boundaries may not
necessarily coincide with host, router, or network boundaries.
Nimrod performs its routing functions with respect to the hierarchy
of entities resulting from clustering and abstraction, not with
respect to the physical realization of the internetwork. In fact,
Nimrod need not even be aware of the physical elements of an
internetwork.

2.3.2 Restricting Information Distribution

The Nimrod architecture supports restricted distribution of routing
information, both to reduce resource consumption associated with such
distribution and to permit information hiding. Each cluster
determines the portions of its routing information to distribute and
the set of entities to which to distribute this information.
Moreover, recipients of routing information are selective in which
information they retain. Some examples are as follows. Each cluster
might automatically advertise its routing information to its siblings
(i.e., those clusters with a common parent cluster). In response to
requests, a cluster might advertise information about specific
portions of the cluster or information that applies only to specific
users. A cluster might only retain routing information from clusters
that provide universal access to their services.

2.3.3 Local Selection of Feasible Routes

   Generating routes that satisfy multiple constraints is usually an
   NP-complete problem and hence a computationally intensive procedure.
   With Nimrod, only those entities that require routes with special
   constraints need assume the computational load associated with
   generation and selection of such routes. Moreover, the Nimrod
   architecture allows individual entities to choose their own route
   generation and selection algorithms and hence the amount of resources
   to devote to these functions.

2.3.4 Caching

   The Nimrod architecture encourages caching of acquired routing
   information in order to reduce the amount of resources consumed and
   delay incurred in obtaining the information in the future. The set
   of routes generated as a by-product of generating a particular route
   is an example of routing information that is amenable to caching;
   future requests for any of these routes may be satisfied directly
   from the route cache. However, as with any caching scheme, the
   cached information may become stale and its use may result in poor
   quality routes. Hence, the routing information's expected duration
   of usefulness must be considered when determining whether to cache
   the information and for how long.

2.3.5 Limiting Forwarding Information

   The Nimrod architecture supports two separate approaches for
   containing the amount of forwarding information that must be
   maintained per router. The first approach is to multiplex, over a
   single path (or tree, for multicast), multiple traffic flows with
   similar service requirements. The second approach is to install and
   retain forwarding information only for active traffic flows.

   With Nimrod, the service providers and users share responsibility for
   the amount of forwarding information in an internetwork. Users have
   control over the establishment of paths, and service providers have
   control over the maintenance of paths. This approach is different
   from that of the current Internet, where forwarding information is
   established in routers independent of demand for this information.

3. Architecture

   Nimrod is a hierarchical, map-based routing architecture that has
   been designed to support a wide range of user requirements and to
   scale to very large dynamic internets. Given a traffic stream's
   description and requirements (both quality of service requirements
   and usage-restriction requirements), Nimrod's main function is to

manage in a scalable fashion how much information about the
internetwork is required to choose a route for that stream, in other
words, to manage the trade-off between amount of information about
the internetwork and the quality of the computed route. Nimrod is
implemented as a set of protocols and distributed databases. The
following sections describe the basic architectural concepts used in
Nimrod. The protocols and databases are specified in other
documents.

3.1 Endpoints

The basic entity in Nimrod is the endpoint. An endpoint represents a
user of the internetwork layer: for example, a transport connection.
Each endpoint has at least one endpoint identifier (EID). Any given
EID corresponds to a single endpoint. EIDs are globally unique,
relatively short "computer-friendly" bit strings---for example, small
multiples of 64 bits. EIDs have no topological significance
whatsoever. For ease of management, EIDs might be organized
hierarchically, but this is not required.

   BEGIN COMMENT

      In practice, EIDs will probably have a second form, which we can
      call the endpoint label (EL). ELs are ASCII strings of unlimited
      length, structured to be used as keys in a distributed database
      (much like DNS names). Information about an endpoint---for
      example, how to reach it---can be obtained by querying this
      distributed database using the endpoint's label as key.

   END COMMENT

3.2 Nodes and Adjacencies

A node represents a region of the physical network. The region of
the network represented by a node can be as large or as small as
desired: a node can represent a continent or a process running inside
a host. Moreover, as explained in section 4, a region of the network
can simultaneously be represented by more than one node.

An adjacency consists of an ordered pair of nodes. An adjacency
indicates that traffic can flow from the first node to the second.

3.3 Maps

The basic data structure used for routing is the map. A map
expresses the available connectivity between different points of an
internetwork. Different maps can represent the same region of a
physical network at different levels of detail.

A map is a graph composed of nodes and adjacencies. Properties of
nodes are contained in attributes associated with them. Adjacencies
have no attributes. Nimrod defines languages to specify attributes
and to describe maps.

Maps are used by routers to generate routes. In general, it is not
required that different routers have consistent maps.

   BEGIN COMMENT

      Nimrod has been designed so that there will be no routing loops
      even when the routing databases of different routers are not
      consistent. A consistency requirement would not permit
      representing the same region of the internetwork at different
      levels of detail. Also, a routing-database consistency
      requirement would be hard to guarantee in the very large internets
      Nimrod is designed to support.

   END COMMENT

In this document we speak only of routers. By "router" we mean a
physical device that implements functions related to routing: for
example, forwarding, route calculation, path set-up. A given device
need not be capable of doing all of these to be called a router. The
protocol specification document, see [2], splits these
functionalities into specific agents.

3.3.1 Connectivity Specifications

By connectivity between two points we mean the available services and
the restrictions on their use. Connectivity specifications are among
the attributes associated with nodes. The following are informal
examples of connectivity specifications:

o "Between these two points, there exists best-effort service with no
   restrictions."

o "Between these two points, guaranteed 10 ms delay can be arranged for
   traffic streams whose data rate is below 1 Mbyte/sec and that have low
   (specified) burstiness."

o "Between these two points, best-effort service is offered, as long as
   the traffic originates in and is destined for research organizations."

3.4 Locators

A locator is a string of binary digits that identifies a location in
an internetwork. Nodes and endpoint are assigned locators.

Different nodes have necessarily different locators. A node is
assigned only one locator. Locators identify nodes and specify
where a node is in the network. Locators do *not* specify a path
to the node. An endpoint can be assigned more than one locator. In
this sense, a locator might appear in more than one location of an
internetwork.

In this document locators are written as ASCII strings that include
colons to underline node structure: for example, a:b:c. This does
not mean that the representation of locators in packets or in
databases will necessarily have something equivalent to the colons.

A given physical element of the network might help implement more
than one node---for example, a router might be part of two different
nodes. Though this physical element might therefore be associated
with more than one locator, the nodes that this physical element
implements have each only one locator.

The connectivity specifications of a node are identified by a tuple
consisting of the node's locator and an ID number.

All map information is expressed in terms of locators, and routing
selections are based on locators. EIDs are *not* used in making
routing decisions---see section 5.

3.5 Node Attributes

   The following are node attributes defined by Nimrod.

3.5.1 Adjacencies

   Adjacencies appear in maps as attributes of both the nodes in the
   adjacency. A node has two types of adjacencies associated with it:
   those that identify a neighboring node to which the original node can
   send data to; and those that identivy a neighboring node that can
   send data to the original node.

3.5.2 Internal Maps

   As part of its attributes, a node can have internal maps. A router
   can obtain a node's internal maps---or any other of the node's
   attributes, for that matter---by requesting that information from a
   representative of that node. (A router associated with that node can
   be such a representative.) A node's representative can in principle
   reply with different internal maps to different requests---for
   example, because of security concerns. This implies that different
   routers in the network might have different internal maps for the
   same node.

A node is said to own those locators that have as a prefix the
locator of the node. In a node that has an internal map, the
locators of all nodes in this internal map are prefixed by the
locator of the original node.

Given a map, a more detailed map can be obtained by substituting one
of the map's nodes by one of that node's internal maps. This process
can be continued recursively. Nimrod defines standard internal maps
that are intended to be used for specific purposes. A node's
"detailed map" gives more information about the region of the network
represented by the original node. Typically, it is closer to the
physical realization of the network than the original node. The
nodes of this map can themselves have detailed maps.

3.5.3 Transit Connectivity

For a given node, this attribute specifies the services available
between nodes adjacent to the given node. This attribute is
requested and used when a router intends to route traffic *through* a
node. Conceptually, the traffic connectivity attribute is a matrix
that is indexed by a pair of locators: the locators of adjacent
nodes. The entry indexed by such a pair contains the connectivity
specifications of the services available across the given node for
traffic entering from the first node and exiting to the second node.

The actual format of this attribute need not be a matrix. This
document does not specify the format for this attribute.

3.5.4 Inbound Connectivity

For a given node, this attribute represents connectivity from
adjacent nodes to points within the given node. This attribute is
requested and used when a router intends to route traffic to a point
within the node but does not have, and either cannot or does not want
to obtain, a detailed map of the node. The inbound connectivity
attribute identifies what connectivity specifications are available
between pairs of locators. The first element of the pair is the
locator of an adjacent node; the second is a locator owned by the
given node.

3.5.5 Outbound Connectivity

For a given node, this attribute represents connectivity from points
within the given node to adjacent nodes. This attribute identifies
what connectivity specifications are available between pairs of
locators. The first element of the pair is a locator owned by the
given node, the second is the locator of an adjacent node.

The Transit, Inbound and Outbound connectivity attributes together
wiht a list of adjacencies form the "abstract map."

4. Physical Realization

A network is modeled as being composed of physical elements: routers,
hosts, and communication links. The links can be either point-to-
point---e.g., T1 links---or multi-point---e.g., ethernets, X.25
networks, IP-only networks, etc.

The physical representation of a network can have associated with it
one or more Nimrod maps. A Nimrod map is a function not only of the
physical network, but also of the configured clustering of elements
(locator assignment) and of the configured connectivity.

Nimrod has no pre-defined "lowest level": for example, it is possible
to define and advertise a map that is physically realized inside a
CPU. In this map, a node could represent, for example, a process or a
group of processes. The user of this map need not necessarily know
or care. ("It is turtles all the way down!", in [3] page 63.)

4.1 Contiguity

Locators sharing a prefix must be assigned to a contiguous region of
a map. That is, two nodes in a map that have been assigned locators
sharing a prefix should be connected to each other via nodes that
themselves have been assigned locators with that prefix. The main
consequence of this requirement is that "you cannot take your locator
with you."

As an example of this, see figure 1, consider two providers x.net and
y.net (these designations are *not* locators but DNS names) which
appear in a Nimrod map as two nodes with locators A and B. Assume
that corporation z.com (also a DNS name) was originally connected to
x.net. Locators corresponding to elements in z.com are, in this
example, A-prefixed. Corporation z.com decides to change providers-
--severing its physical connection to x.net. The connectivity
requirement described in this section implies that, after the
provider change has taken place, elements in z.com will have been, in
this example, assigned B-prefixed locators and that it is not
possible for them to receive data destined to A-prefixed locators
through y.net.

```
                  A                       B
              +------+              +------+
              | x.net|              | y.net|
              +------+             /+------+
                                  /
                         +-----+
                         |z.com|
                         +-----+
```
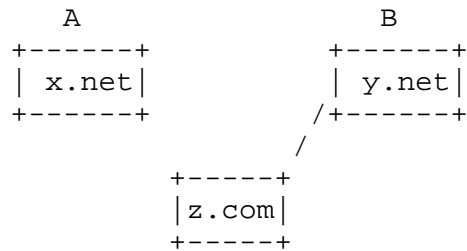
Figure 1:   Connectivity after switching providers

The contiguity requirement simplifies routing information exchange:
if it were permitted for z.com to receive A-prefixed locators through
y.net, it would be necessary that a map that contains node B include
information about the existence of a group of A-prefixed locators
inside node B. Similarly, a map including node A would have to
include information that the set of A-prefixed locators asigned to
z.com is not to be found within A. The more situations like this
happen, the more the hierarchical nature of Nimrod is subverted to
"flat routing." The contiguity requirement can also be expressed as
"EIDs are stable; locators are ephemeral."

4.2 An Example

Figure 2 shows a physical network. Hosts are drawn as squares,
routers as diamonds, and communication links as lines. The network
shown has the following components: five ethernets ---EA through EE;
five routers---RA through RE; and four hosts---HA through HD. Routers
RA, RB, and RC interconnect the backbone ethernets---EB, EC and ED.
Router RD connects backbone EC to a network consisting of ethernet EA
and hosts HA and HB. Router RE interconnects backbone ED to a
network consisting of ethernet EE and hosts HC and HD. The assigned
locators appear in lower case beside the corresponding physical
entity.

Figure 3 shows a Nimrod map for that network. The nodes of the map
are represented as squares. Lines connecting nodes represent two
adjacencies in opposite directions. Different regions of the network
are represented at different detail. Backbone b1 is represented as a
single node. The region of the network with locators prefixed by "a"
is represented as a single node. The region of the network with
locators prefixed by "c" is represented in full detail.

4.3 Multiple Locator Assignment

   Physical elements can form part of, or implement, more than one node.
   In this sense it can be said that they can be assigned more than one
   locator. Consider figure 4, which shows a physical network. This
   network is composed of routers (RA, RB, RC, and RD), hosts (HA, HB,
   and HC), and communication links. Routers RA, RB, and RC are
   connected with point-to-point links. The two horizontal lines in the
   bottom of the figure represent ethernets. The figure also shows the
   locators assigned to hosts and routers.

   In figure 4, RA and RB have each been assigned one locator (a:t:r1
   and b:t:r1, respectively). RC has been assigned locators a:y:r1 and
   b:d:r1; one of these two locators shares a prefix with RA's locator,
   the other shares a prefix with RB's locator. Hosts HA and HB have
   each been assigned three locators. Host HC has been assigned one
   locator. Depending on what communication paths have been set up
   between points, different Nimrod maps result. A possible Nimrod map
   for this network is given in figure 5.

```
                                a:h1 +--+        a:h2 +--+
                                     |HA|             |HB|
                                     |  |             |  |
                                     +--+             +--+
                               a:e1   |                |
                                    --------------------- EA
                                                 |
                                  /\             /\
                                 /RB\ b1:r1     /RD\ b2:r1
                                /\  /\          \  /
                               /  \/  \          \/
               EB        b1:t:e1  /     \         |    EC
            -----------------------     ----------------------- b2:e1
               /                            \
              /                              \
            /\                                \
           /RA\ b1:r2                          \
           \  /                                 \/\
            \/                                  /RC\   b2:t:r2
              \                                 \  /
               \                                 \/
                \        --------------------------  /   ED
                 --------------------------------- b3:t:e1
                                     |
                                     |
                                     |
                                    /\
                                   /RE\ b3:t:r1
                    EE             \  /
                    ---------------- \/ ------------------ c:e1
                         |                    |
                        +--+                 +--+
                        |HC|  c:h1           |HD|   c:h2
                        |  |                 |  |
                        +--+                 +--+
```
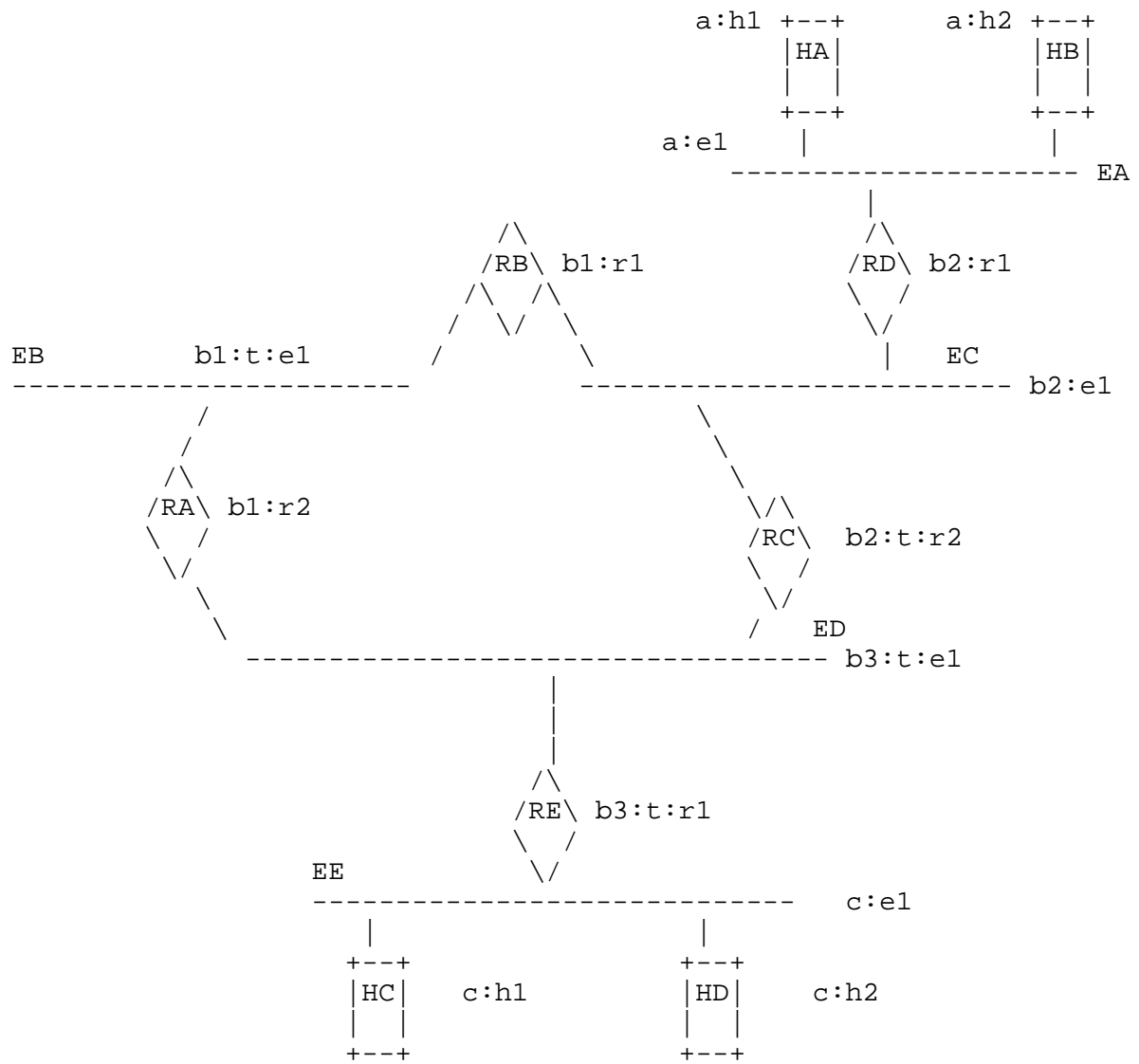
                    Figure 2: Example Physical Network

```
                                 +-----+              +-----+
        +---------+              |     |              |     |
        |         |  -------------| b2  |  ------------|  a  |
        |         |              |     |              |     |
        |   b1    |              +-----+              +-----+
        |         |                 |
        |         |                 |
        |         |                 |
        +---------+                 |
                   \                |
                    \               |
                     \              |
                      \          +--------+
                       \         |        |
                 ------- |  b3:t:e1|
                         |        |
                         +--------+
                             |
                             |
                             |
                             |
                         +-------+
                         |       |
                         |b3:t:r1|
                         |       |
                         +-------+
                             |
        +-----+          +-----+          +-----+
        |     |          |     |          |     |
        | c:h1|-------|  c:e1|-----|  c:h2|
        |     |          |     |          |     |
        +-----+          +-----+          +-----+
```

                       Figure 3:   Nimrod Map

```
               a:t:r1                    b:t:r1
                 +--+                      +--+
                 |RA|------------|RB|
                 +--+                      +--+
                  \                        /
                   \                      /
                    \                    /
                     \                  /
                      \                /
                       \              /
                        \            /
                         \          /
                          \        /
                           \      /
                            \    /
                             \  /
                              \
                            +--+
                            |RC|   a:y:r1
                            +--+   b:d:r1
                              |
              ---------------------------
              |             |             |
    a:y:h1   +--+          +--+          +--+   a:y:h2
    b:d:h2   |HA|          |RD| c:r1     |HB|   b:d:h1
    c:h1     +--+          +--+          +--+   c:h2
                             |
                             |
                   -------------------
                             |
                           +--+
                           |HC| c:h3
                           +--+
```
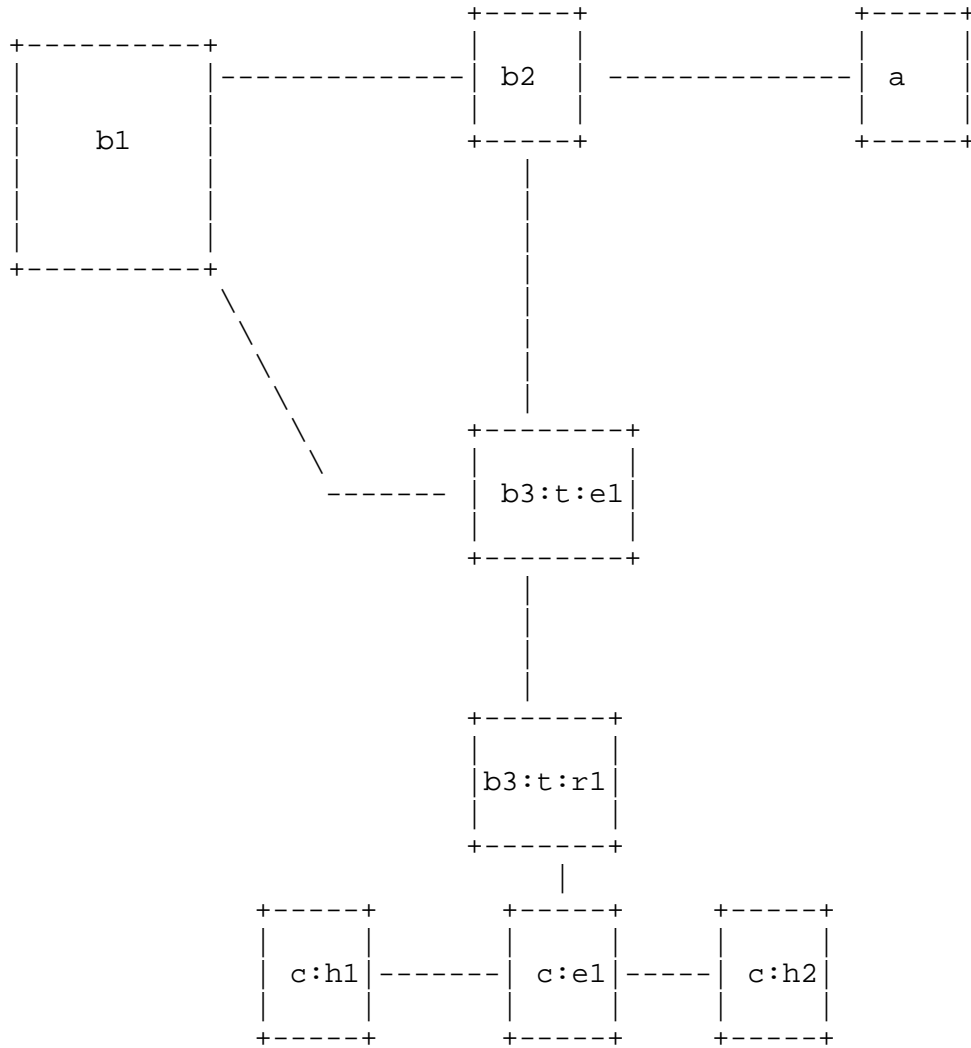
Figure 4: Multiple Locators

```
           a                           b                        c
   +------------+           +------------+           +--------------+
   |            |           |            |           |              |
   |     a:t    |           |     b:t    |           |              |
   |   +--+     |           |   +--+     |           |              |
   |   |  |-------------------|--|  |    |           |              |
   |   +--+     |           |   +--+     |           |              |
   |    |       |           |    |       |           |              |
   |   +--+     |           |   +--+     |           |              |
   |   +  +     |           |   +  +     |           |              |
   |   +--+ a:y |           |   +--+ b:d |           |              |
   |            |           |            |           |              |
   +------------+           +------------+           +--------------+
```

Figure 5: Nimrod Map

Nodes and adjacencies represent the *configured* clustering and
connectivity of the network. Notice that even though a:y and b:d are
defined on the same hardware, the map shows no connection between
them: this connection has not been configured. A packet given to
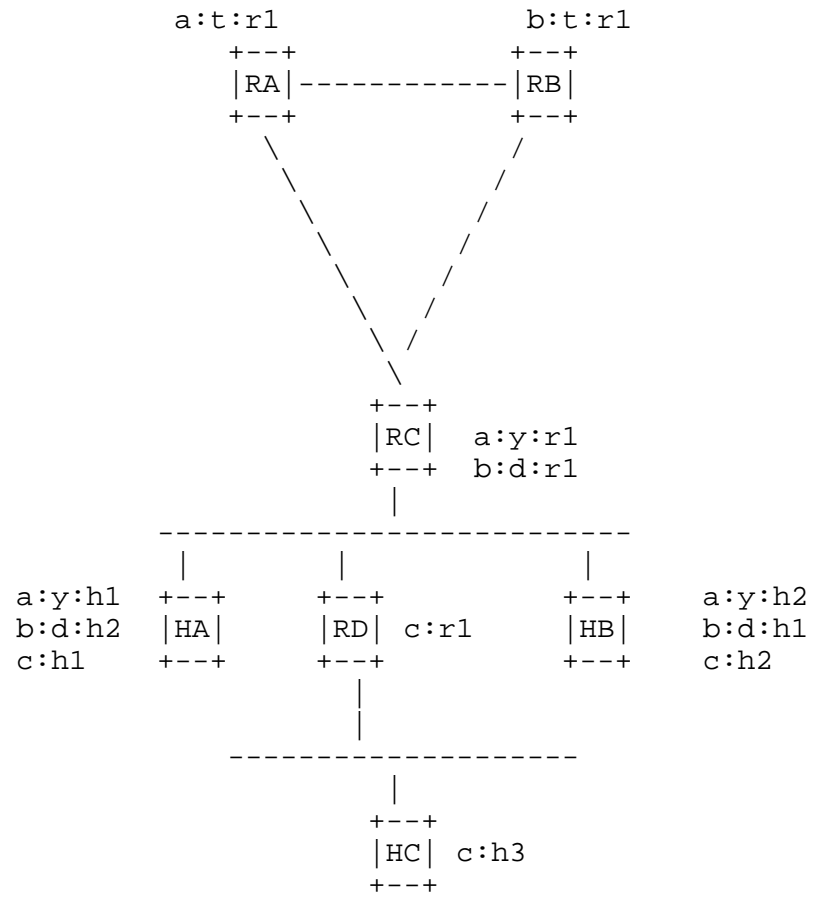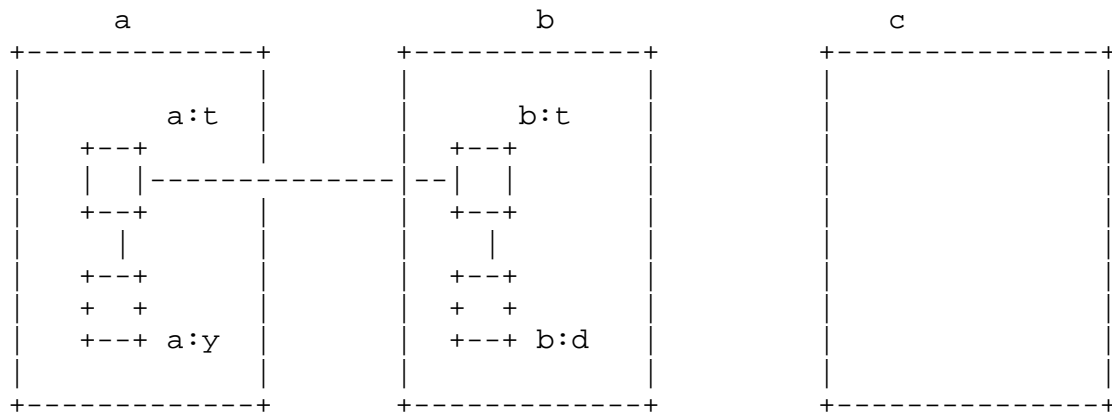node 'a' addressed to a locator prefixed with "b:d" would have to
travel from node a to node b via the arc joining them before being
directed towards its destination. Similarly, the map shows no
connection between the c node and the other two top level nodes. If
desired, these connections could be established, which would
necessitate setting up the exchange of routing information. Figure 6
shows the map when these connections have been established.

In the strict sense, Nimrod nodes do not overlap: they are distinct
entities. But, as we have seen in the previous example, a physical
element can be given more than one locator, and, in that sense,
participate in implementing more than one node. That is, two
different nodes might be defined on the same hardware. In this
sense, Nimrod nodes can be said to overlap. But to notice this
overlap one would have to know the physical-to-map correspondence.
It is not possible to know when two nodes share physical assets by
looking only at a Nimrod map.

5. Forwarding

   Nimrod supports four forwarding modes:

 1. Connectivity Specification Chain (CSC) mode: In this mode, packets
    carry a list of connectivity specifications. The packet is
    required to go through the nodes that own the connectivity
    specifications using the services specified. The nodes associated
    with the listed connectivity specifications should define a
    continuous path in the map. A more detailed description of the
    requirements of this mode is given in section 5.3.

```
   +--------+                                          +--------+
   |        |                                          |        |
   | a:t:r1 |------------------------------------------| b:t:r1 |
   |        |                                          |        |
   +--------+                                          +--------+
      |                                                    |
      |         /--------------------------------------\   |
      |         |                                      |   |
      |     +--------+        +--------+           +--------+ |
      |     |        |        |        |           |        | |
      |     | a:y:h1 --------|  c:h1   |-------------------| b:d:h1 | |
      |     |        |        |        |           |        | |
      |     +--------+        +--------+           +--------+ |
      |      |    |        |   |    |            |    |   | |
   +--------+ |   |        | +------+ +------+   |  +--------+ |
   |        | |   |        | |      | |      |   |  |        | |
   | a:y:r1 | |   |        | | c:r1 |--| c:h3 |   |  | b:d:r1 | |
   |        | |   |        | |      | |      |   |  |        | |
   +--------+ |   |        | +------+ +------+   |  +--------+ |
      |   |   |   |        |   |    |          |    |   | |
      | +--------+ |       | +--------+        | +--------+ |
      | |        | |       | |        |        | |        | |
      | | a:y:h2 |--------- |  c:h2   |-------------------| b:d:h2 | |
      | |        | |       | |        |        | |        | |
      | +--------+ |       | +--------+        | +--------+ |
      |     |      |       |    |              |    |       |
      |     |      |       |    |              |    |       |
      |     |      \-------|----|--------------|----/       |
      |     |              |    |              |            |
      \-----|--------------|----|--------------|------------/
            \--------------|----|--------------|
```
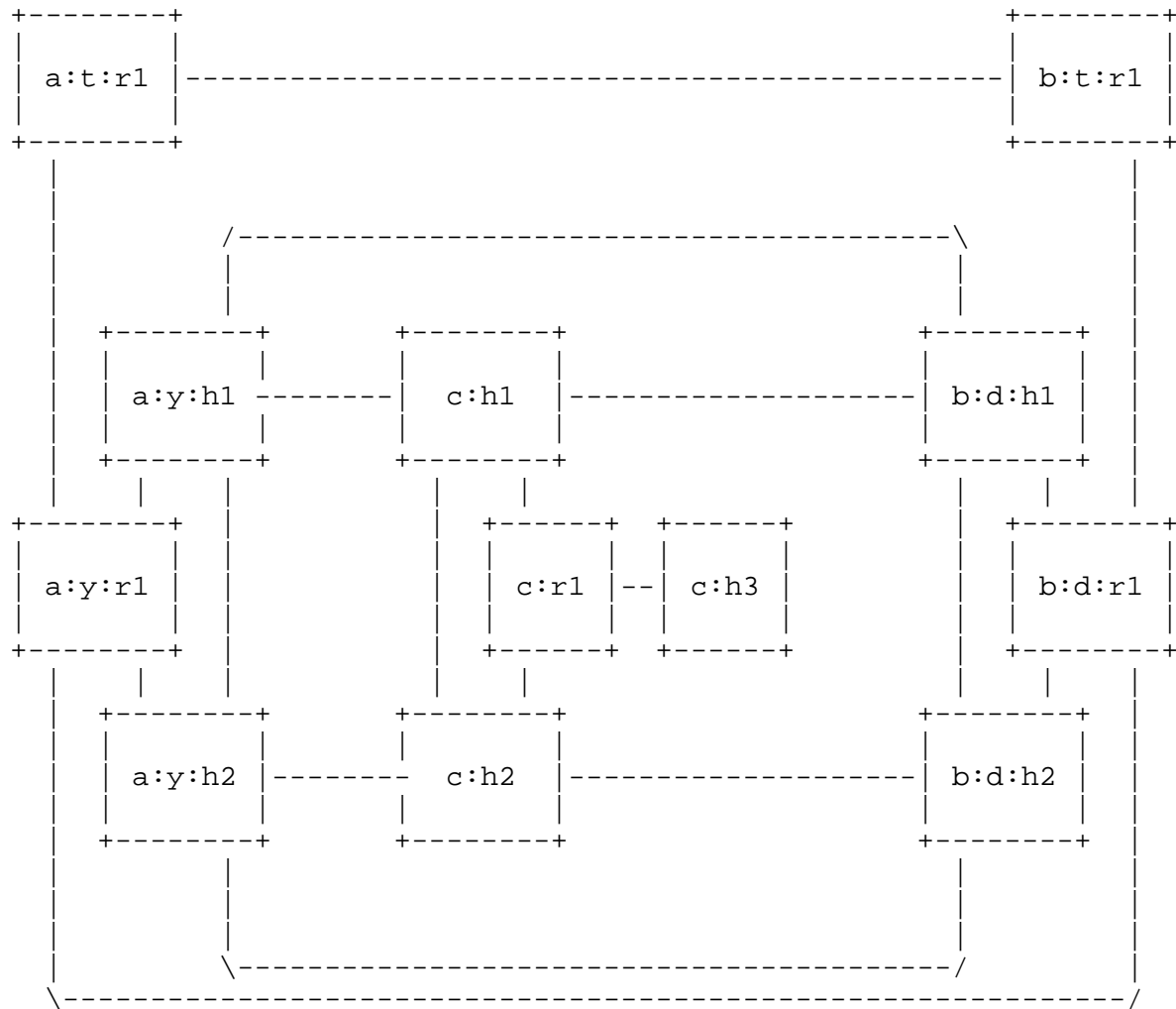
                     Figure 6: Nimrod Map II

 2. Connectivity Specifications Sequence (CSS) mode: In this mode,
    packets carry a list of connectivity specifications. The packet
    is supposed to go sequentially through the nodes that own each one
    of the listed connectivity specifications in the order they were
    specified. The nodes need not be adjacent. This mode can be seen
    as a generalization of the CSC mode. Notice that CSCs are said to
    be a *chains* of locators, CSSs are *sequences* of locators. This
    difference emphasizes the contiguity requirement in CSCs. A
    detailed description of this mode is in section 5.6.

3. Flow mode: In this mode, the packet includes a path-id that
   indexes state that has been previously set up in routers along the
   path. Packet forwarding when flow state has been established is
   relatively simple: follow the instructions in the routers' state.
   Nimrod includes a mechanism for setting up this state. A more
   detailed description of this mode can be found in section 5.4.

4. Datagram mode: in this mode, every packet carries source and
   destination locators. This mode can be seen as a special case of
   the CSS mode. Forwarding is done following procedures as
   indicated in section 5.5.

   BEGIN COMMENT

   The obvious parallels are between CSC mode and IPV4's strict
   source route and between CSS mode and IPV4's loose source route.

   END COMMENT

   In all of these modes, the packet may also carry locators and EIDs
   for the source and destinations. In normal operation, forwarding
   does not take the EIDs into account, only the receiver does. EIDs
   may be carried for demultiplexing at the receiver, and to detect
   certain error conditions. For example, if the EID is unknown at the
   receiver, the locator and EID of the source included in the packet
   could be used to generate an error message to return to the source
   (as usual, this error message itself should probably not be allowed
   to be the cause of other error messages). Forwarding can also use
   the source locator and EID to respond to error conditions, for
   example, to indicate to the source that the state for a path-id
   cannot be found.

   Packets can be visualized as moving between nodes in a map. A packet
   indicates, implicitly or explicitly, a destination locator. In a
   packet that uses the datagram, CSC, or CSS forwarding mode, the
   destination locator is explicitly indicated . In a packet that uses
   the flow forwarding mode, the destination locator is implied by the
   path-id and the distributed state in the network (it might also be
   included explicitly). Given a map, a packet moves to the node in
   this map to which the associated destination locator belongs. If the
   destination node has a "detailed" internal map, the destination
   locator must belong to one of the nodes in this internal map
   (otherwise it is an error). The packet goes to this node (and so on,
   recursively).

5.1 Policy

   CSC and CSS mode implement policy by specifying the connectivity
   specifications associated with those nodes that the packet should
   traverse. Strictly speaking, there is no policy information included
   in the packet. That is, in principle, it is not possible to
   determine what criteria were used to select the route by looking at
   the packet. The packet only contains the results of the route
   generation process. Similarly, in a flow mode packet, policy is
   implicit in the chosen route.

   A datagram-mode packet can indicate a limited form of policy routing
   by the choice of destination and source locators. For this choice to
   exist, the source or destination endpoints must have several locators
   associated with them. This type of policy routing is capable of, for
   example, choosing providers.

5.2 Trust

   A node that chooses not to divulge its internal map can work
   internally any way its administrators decide, as long as the node
   satisfies its external characterization as given in its Nimrod map
   advertisements. Therefore, the advertised Nimrod map should be
   consistent with a node's actual capabilities. For example, consider
   the network shown in figure 7 which shows a physical network and the
   advertised Nimrod map. The physical network consists of hosts and a
   router connected together by an ethernet. This node can be sub-
   divided into component nodes by assigning locators as shown in the
   figure and advertising the map shown. The map seems to imply that it
   is possible to send packets to node a:x without these being
   observable by node a:y; however, this is actually not enforceable.

   In general, it is reasonable to ask how much trust should be put in
   the maps obtained by a router. Even when a node is "trustworthy,"
   and the information received from the node has been authenticated,
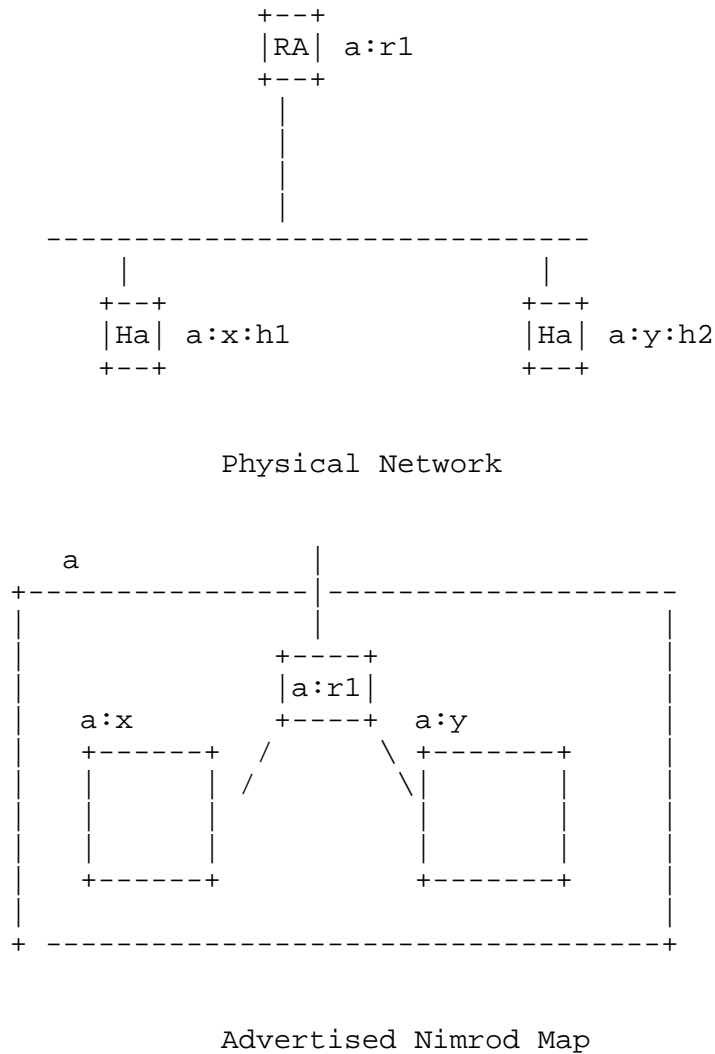   there is always the possibility of an honest mistake.

```
                       +--+
                       |RA|  a:r1
                       +--+
                          |
                          |
                          |
                          |
              -------------------------------
                  |                        |
                +--+                      +--+
                |Ha|  a:x:h1              |Ha|  a:y:h2
                +-+                       +-+


                      Physical Network


         a                    |
    +---------------|------------------
    |               |                 |
    |               |                 |
    |            +----+               |
    |            |a:r1|               |
    |   a:x      +----+   a:y         |
    |   +------+   /      \ +-------+  |
    |   |      |  /        \|       |  |
    |   |      | /          |       |  |
    |   |      |            |       |  |
    |   +------+            +-------+  |
    |                                 |
    + -------------------------------+


                 Advertised Nimrod Map




              Figure 7:   Example of Misleading Map
```

5.3 Connectivity Specification (CSC) Mode

   Routing for a CSC packet is specified by a list of connectivity
   specifications carried in the packet. These are the connectivity
   specifications that make the specified path, in the order that they
   appear along the path. These connectivity specifications are
   attributes of nodes. The route indicated by a CSC packet is specifed
   in terms of connectivity specifications rather than physical
   entities: a connectivity specification in a CSC-mode packet would

correspond to a type of service between two points of the network
without specifying the physical path.

Given two connectivity specifications that appear consecutively in
the a CSC-mode packet, there should exist an adjacency going from the
node corresponding to the first connectivity specification to the
node corresponding to the second connectivity specification. The
first connectivity specification referenced in a CSC-mode packet
should be an outbound connectivity specification; similarly, the last
connectivity specification referenced in a CSC-mode packet should be
an inbound connectivity specification; the rest should be transit
connectivity specifications.

5.4 Flow Mode

A flow mode packet includes a path-id field. This field identifies
state that has been established in intermediate routers. The packet
might also contain locators and EIDs for the source and destination.
The setup packet also includes resource requirements. Nimrod
includes protocols to set up and modify flow-related state in
intermediate routers. These protocols not only identify the
requested route, but also describe the resources requested by the
flow---e.g., bandwidth, delay, etc. The result of a set-up attempt
might be either confirmation of the set-up or notification of its
failure. The source-specified routes in flow mode setup are
specified in terms of CSSs.

5.5 Datagram Mode

A realistic routing architecture must include an optimization for
datagram traffic, by which we mean user transactions which consist of
single packets, such as a lookup in a remote translation database.
Either of the two previous modes contains unacceptable overhead if
much of the network traffic consists of such datagram transactions.
A mechanism is needed which is approximately as efficient as the
existing IPv4 "hop-by-hop" mechanism. Nimrod has such a mechanism.

The scheme can be characterized by the way it divides the state in a
datagram network between routers and the actual packets. In IPv4,
most packets currently contain only a small amount of state
associated with the forwarding process ("forwarding state")---the hop
count. Nimrod proposes that enlarging the amount of forwarding state
in packets can produce a system with useful properties. It was
partially inspired by the efficient source routing mechanism in SIP
[5], and the locator pointer mechanism in PIP [6]).

Nimrod datagram mode uses pre-set flow-mode state to support a
strictly non-looping path, but without a source-route.

5.6 Connectivity Specification Sequence Mode

   The connectivity specification sequence mode specifies a route by a
   list of connectivity specifications. There are no contiguity
   restrictions on consecutive connectivity specifications.

      BEGIN COMMENT

      The CSS and CSC modes can be seen as combination of the datagram
      and flow modes. Therefore, in a sense, the basic forwarding modes
      of Nimrod are just these last two.

      END COMMENT

6. Security Considerations

   Security issues are not addressed in this document.

7. References

   [1] Steenstrup, M., "Inter-Domain Policy Routing Protocol
       Specification: Version 1," RFC 1479, June 1993.

   [2] Steenstrup M., and R. Ramanathan, "Nimrod Functionality and
       Protocols Specification," Work in Progress, February 1996.

   [3] Wright, R., "Three Scientists and their Gods Looking for Meaning
       in an Age of Information", New York: Times Book, first ed., 1988.

   [4] Deering, S., "SIP: Simple Internet Protocol," IEEE Network, vol.
       7, May 1993.

   [5] Francis, P., "A Near-Term Architecture for Deploying Pip," IEEE
       Network, vol. 7, May 1993.

8. Authors' Addresses

   Isidro Castineyra
   BBN Systems and Technologies
   10 Moulton Street
   Cambridge, MA 02138

   Phone: (617) 873-6233
   EMail: isidro@bbn.com

   Noel Chiappa
   EMail: gnc@ginger.lcs.mit.edu

   Martha Steenstrup
   BBN Systems and Technologies
   10 Moulton Street
   Cambridge, MA 02138

   Phone: (617) 873-3192
   EMail: msteenst@bbn.com