Network Working Group
Request for Comments: 1222

H-W. Braun
San Diego Supercomputer Center
Y. Rekhter
IBM T.J. Watson Research Center
May 1991

Advancing the NSFNET Routing Architecture

Status of this Memo

This RFC suggests improvements in the NSFNET routing architecture to accommodate a more flexible interface to the Backbone clients. This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

Introduction

This memo describes the history of NSFNET Backbone routing and outlines two suggested phases for further evolution of the Backbone's routing interface. The intent is to provide a more flexible interface for NSFNET Backbone service subscribers, by providing an attachment option that is simpler and lower-cost than the current one.

Acknowledgements

The authors would like to thank Scott Brim (Cornell University), Bilal Chinoy (Merit), Elise Gerich (Merit), Paul Love (SDSC), Steve Wolff (NSF), Bob Braden (ISI), and Joyce K. Reynolds (ISI) for their review and constructive comments.

1. NSFNET Phase 1 Routing Architecture

In the first phase of the NSFNET Backbone, a 56Kbps infrastructure utilized routers based on Fuzzball software [2]. The Phase 1 Backbone used the Hello Protocol for interior routing. At the periphery of the Backbone, the client networks were typically connected by using a gatedaemon ("gated") interface to translate between the Backbone's Hello Protocol and the interior gateway protocol (IGP) of the mid-level network.

Mid-level networks primarily used the Routing Information Protocol (RIP) [3] for their IGP. The gatedaemon system acted as an interface between the Hello and RIP environments. The overall appearance was that the Backbone, mid-level networks, and the campus networks formed a single routing system in which information was freely exchanged.

Network metrics were translated among the three network levels (backbone, mid-level networks, and campuses).

With the development of the gatedaemon, sites were able to introduce filtering based on IP network numbers. This process was controlled by the staff at each individual site.

Once specific network routes were learned, the infrastructure forwarded metric changes throughout the interconnected network. The end-result was that a metric fluctuation on one end of the interconnected network could permeate all the way to the other end, crossing multiple network administrations. The frequency of metric fluctuations within the Backbone itself was further increased when event-driven updates (e.g., metric changes) were introduced. Later, damping of the event driven updates lessened their frequency, but the overall routing environment still appeared to be quite unstable.

Given that only limited tools and protocols were available to engineer the flow of dynamic routing information, it was fairly easy for routing loops to form. This was amplified as the topology became more fully connected without insulation of routing components from each other.

All six nodes of the Phase 1 Backbone were located at client sites, specifically NSF funded supercomputer centers.

2. NSFNET Phase 2 Routing Architecture

The routing architecture for the second phase of the NSFNET Backbone, implemented on T1 (1.5Mbps) lines, focused on the lessons learned in the first NSFNET phase. This resulted in a strong decoupling of the IGP environments of the backbone network and its attached clients [5]. Specifically, each of the administrative entities was able to use its own IGP in any way appropriate for the specific network. The interface between the backbone network and its attached client was built by means of exterior routing, initially via the Exterior Gateway Protocol (EGP) [1,4].

EGP improved provided routing isolation in two ways. First, EGP signals only up/down transitions for individual network numbers, not the fluctuations of metrics (with the exception of metric acceptance of local relevance to a single Nodal Switching System (NSS) only for inbound routing information, in the case of multiple EGP peers at a NSS). Second, it allowed engineering of the dynamic distribution of routing information. That is, primary, secondary, etc., paths can be determined, as long as dynamic externally learned routing information is available. This allows creation of a spanning tree routing

topology, satisfying the constraints of EGP.

The pre-engineering of routes is accomplished by means of a routing configuration database that is centrally controlled and created, with a subsequent distribution of individual configuration information to all the NSFNET Backbone nodes. A computer controlled central system ensures the correctness of the database prior to its distribution to the nodes.

All nodes of the 1.5Mbps NSFNET Backbone (currently fourteen) are located at client sites, such as NSF funded supercomputer centers and mid-level network attachment points.

3. T3 Phase of the NSFNET Backbone

The T3 (45Mbps) phase of the NSFNET Backbone is implemented by means of a new architectural model, in which the principal communication nodes (core nodes) are co-located with major phone company switching facilities. Those co-located nodes then form a two-dimensional networking infrastructure "cloud". Individual sites are connected via exterior nodes (E-NSS) and typically have a single T3 access line to a core node (C-NSS). That is, an exterior node is physically at the service subscriber site.

With respect to routing, this structure is invisible to client sites, as the routing interface uses the same techniques as the T1 NSFNET Backbone. The two backbones will remain independent infrastructures, overlaying each other and interconnected by exterior routing, and the T1 Backbone will eventually be phased out as a separate network.

4. A Near-term Routing Alternative

The experience with the T1/T3 NSFNET routing demonstrated clear advantages of this routing architecture in which the whole infrastructure is strongly compartmentalized. Previous experience also showed that the architecture imposes certain obligations upon the attached client networks. Among them is the requirement that a service subscriber must deploy its own routing protocol peer, participating in the IGP of the service subscriber and connected via a common subnet to the subscriber-site NSFNET node. The router and the NSFNET Backbone exchange routing information via an EGP or BGP [7] session.

The drawbacks imposed by this requirement will become more obvious with the transition to the new architecture that is employed by the T3 phase of the NSFNET Backbone. This will allow rapid expansion to many and smaller sites for which a very simple routing interface may be needed.

We strongly believe that separating the routing of the service subscriber from the NSFNET Backbone routing via some kind of EGP is the correct routing architecture. However, it should not be necessary to translate this architecture into a requirement for each service subscriber to install and maintain additional equipment, or for the subscriber to deal with more complicated routing environments. In other words, while maintaining that the concept of routing isolation is correct, we view the present implementation of the concept as more restrictive than necessary.

An alternative implementation of this concept may be realized by separating the requirement for an EGP/BGP session, as the mechanism for exchanging routing information between the service subscriber network and the backbone, from the actual equipment that has to be deployed and maintained to support such a requirement. The only essential requirement for routing isolation is the presence of two logical routing entities. The first logical entity participates in the service subscriber's IGP, the second logical entity participates in the NSFNET Backbone IGP, and the two logical entities exchange information with each other by means of inter-domain mechanisms. We suggest that these two logical entities could exist within a single physical entity.

In terms of implementation, this would be no different from a gatedaemon system interfacing with the previous 56Kbps NSFNET Backbone from the regional clients, except that we want to continue the strong routing and administrative control that decouple the two IGP domains. Retaining an inter-domain mechanism (e.g., BGP) to connect the two IGP domains within the single physical entity allows the use of a well defined and understood interface. At the same time, care must be taken in the implementation that the two daemons will not simultaneously interact with the system kernel in unwanted ways.

The possibility of interfacing two IGP domains within a single router has also been noted in [8]. For the NSFNET Backbone case, we propose in addition to retain strong firewalls between the IGP domains. The IGP information would need to be tagged with exterior domain information at its entry into the other IGP. It would also be important to allow distributed control of the configuration. The NSFNET Backbone organization and the provider of the attached client network are each responsible for the integrity of their own routing information.

An example implementation might be a single routing engine that executed two instances of routing daemons. In the NSFNET Backbone case, one of the daemons would participate in the service subscriber's IGP, and the other would participate in the NSFNET

Backbone IGP. These two instances could converse with each other by running EGP/BGP via a local loopback mechanism or internal IPC. In the NSFNET Backbone implementation, the NSFNET T1 E-PSP or T3 E-NSS are UNIX machines, so the local loopback interface (lo0) of the UNIX operating system may be used.

Putting both entities into the same physical machine means that the E-PSP/E-NSS would participate in the regional IGP on its exterior interface. We would still envision the Ethernet attachment to be the demarcation point for the administrative control and operational responsibility. However, the regional client could provide the configuration information for the routing daemon that interfaced to the regional IGP, allowing the regional to continue to exercise control over the introduction of routing information into its IGP.

5. Long-Term Alternatives

As technology employed by the NSFNET Backbone evolves, one may envision the demarcation line between the Backbone and the service subscribers moving in the direction of the "C-NSS cloud", so that the NSFNET IGP will be confined to the C-NSS, while the E-NSS will be a full participant in the IGP of the service subscriber.

Clearly, one of the major prerequisites for such an evolution is the ability for operational management of the physical medium connecting a C-NSS with an E-NSS by two different administrative entities (i.e., the NSFNET Backbone provider as well as the service subscriber). It will also have to be manageable enough to be comparable in ease of use to an Ethernet interface, as a well-defined demarcation point.

The evolution of the Point-to-Point Protocol, as well as a significantly enhanced capability for managing serial lines via standard network management protocols, will clearly help. This may not be the complete answer, as a variety of equipment is used on serial lines, making it difficult to isolate a hardware problem. Similar issues may arise for future demarcation interfaces to Internet infrastructure (e.g., SMDS interfaces).

In summary, there is an opportunity to simplify the management, administration, and exchange of routing information by collapsing the number of physical entities involved.

6. References

[1] Mills, D., "Exterior Gateway Protocol Formal Specification", RFC 904, BBN, April 1984.

[2] Mills, D., and H-W. Braun, "The NSFNET Backbone Network", SIGCOMM

1987, August 1987.

[3] Hedrick, C., "Routing Information Protocol", RFC 1058, Rutgers University, June 1988.

[4] Rekhter, Y., "EGP and Policy Based Routing in the New NSFNET Backbone", RFC 1092, IBM T.J. Watson Research Center, February 1989.

[5] Braun, H-W., "The NSFNET Routing Architecture", RFC 1093, Merit/NSFNET, February 1989.

[6] Braun, H-W., "Models of Policy Based Routing", RFC 1104, Merit/NSFNET, June 1989.

[7] Lougheed, K., and Y. Rekhter, "A Border Gateway Protocol (BGP)", RFC 1163, cisco Systems, IBM T.J. Watson Research Center, June 1990.

[8] Almquist, P., "Requirements for Internet IP Routers", to be published as a RFC.

7. Security Considerations

Security issues are not discussed in this memo.

8. Authors' Addresses

Hans-Werner Braun
San Diego Supercomputer Center
P.O. Box 85608
La Jolla, CA 92186-9784

Phone: (619) 534-5035
Fax: (619) 534-5113

EMail: HWB@SDSC.EDU

Yakov Rekhter
T.J. Watson Research Center
IBM Corporation
P.O. Box 218
Yorktown Heights, NY 10598

Phone: (914) 945-3896

EMail: Yakov@Watson.IBM.COM