                    MOSPF: Analysis and Experience

Status of this Memo

Abstract

   This memo documents how the MOSPF protocol satisfies the requirements
   imposed on Internet routing protocols by "Internet Engineering Task
   Force internet routing protocol standardization criteria" ([RFC
   1264]).

   Please send comments to mospf@gated.cornell.edu.

1. Summary of MOSPF features and algorithms

   MOSPF is an enhancement of OSPF V2, enabling the routing of IP
   multicast datagrams. OSPF is a link-state (unicast) routing
   protocol, providing a database describing the Autonomous System's
   topology. IP multicast is an extension of LAN multicasting to a
   TCP/IP Internet. IP Multicast permits an IP host to send a single
   datagram (called an IP multicast datagram) that will be delivered to
   multiple destinations. IP multicast datagrams are identified as
   those packets whose destinations are class D IP addresses (i.e.,
   addresses whose first byte lies in the range 224-239 inclusive).
   Each class D address defines a multicast group.

   The extensions required of an IP host to participate in IP
   multicasting are specified in "Host extensions for IP multicasting"
   ([RFC 1112]). That document defines a protocol, the Internet Group
   Management Protocol (IGMP), that enables hosts to dynamically join
   and leave multicast groups.

   MOSPF routers use the IGMP protocol to monitor multicast group
   membership on local LANs through the sending of IGMP Host Membership
   Queries and the reception of IGMP Host Membership Reports. A MOSPF
   router then distributes this group location information throughout
   the routing domain by flooding a new type of OSPF link state
   advertisement, the group-membership-LSA (type 6). This in turn
   enables the MOSPF routers to most efficiently forward a multicast

datagram to its multiple destinations: each router calculates the
path of the multicast datagram as a shortest-path tree whose root is
the datagram source, and whose terminal branches are LANs containing
group members.

A separate tree is built for each [source network, multicast
destination] combination. To ease the computational demand on the
routers, these trees are built "on demand", i.e., the first time a
datagram having a particular combination of source network and
multicast destination is received. The results of these "on demand"
tree calculations are then cached for later use by subsequent
matching datagrams.

MOSPF is meant to be used internal to a single Autonomous System.
When supporting IP multicast over the entire Internet, MOSPF would
have to be used in concert with an inter-AS multicast routing
protocol (something like DVMRP would work).

The MOSPF protocol is based on the work of Steve Deering in
[Deering]. The MOSPF specification is documented in [MOSPF].

1.1. Characteristics of the multicast datagram's path

As a multicast datagram is forwarded along its shortest-path tree,
the datagram is delivered to each member of the destination multicast
group. In MOSPF, the forwarding of the multicast datagram has the
following properties:

    o The path taken by a multicast datagram depends both on the
      datagram's source and its multicast destination. Called
      source/destination routing, this is in contrast to most unicast
      datagram forwarding algorithms (like OSPF) that route
      based solely on destination.

    o The path taken between the datagram's source and any particular
      destination group member is the least cost path available. Cost
      is expressed in terms of the OSPF link-state metric.

    o MOSPF takes advantage of any commonality of least cost paths
      to destination group members. However, when members of the
      multicast group are spread out over multiple networks, the
      multicast datagram must at times be replicated. This replication
      is performed as few times as possible (at the tree branches),
      taking maximum advantage of common path segments.

    o For a given multicast datagram, all routers calculate an
      identical shortest-path tree. This is possible since the
      shortest-path tree is rooted at the datagram source, instead

        of being rooted at the calculating router (as is done in the
        unicast case). Tie-breakers have been defined to ensure
        that, when several equal-cost paths exist, all routers agree
        on which single path to use. As a result, there is a single
        path between the datagram's source and any particular
        destination group member. This means that, unlike OSPF's
        treatment of regular (unicast) IP data traffic, there is no
        provision for equal-cost multipath.

   o While MOSPF optimizes the path to any given group member, it
     does not necessarily optimize the use of the internetwork as
     a whole. To do so, instead of calculating source-based
     shortest-path trees, something similar to a minimal spanning
     tree (containing only the group members) would need to be
     calculated. This type of minimal spanning tree is called a
     Steiner tree in the literature. For a comparison of
     shortest-path tree routing to routing using Steiner trees,
     see [Deering2] and [Bharath-Kumar].

   o When forwarding a multicast datagram, MOSPF conforms to the
     link-layer encapsulation standards for IP multicast
     datagrams as specified in "Host extensions for IP multicasting"
     ([RFC 1112]), "Transmission of IP datagrams over the
     SMDS Service" ([RFC 1209]) and "Transmission of IP and ARP
     over FDDI Networks" ([RFC 1390]). In particular, for ethernet
     and FDDI the explicit mapping between IP multicast
     addresses and data-link multicast addresses is used.

1.2. Miscellaneous features

   This section lists, in no particular order, the other miscellaneous
   features that the MOSPF protocol supports:

   o MOSPF routers can be mixed within an Autonomous System (and
     even within a single OSPF area) with non-multicast OSPF
     routers. When this is done, all routers will interoperate in
     the routing of unicasts. Unicast routing will not be
     affected by this mixing; all unicast paths will be the same
     as before the introduction of multicast. This mixing of
     multicast and non-multicast routers enables phased
     introduction of a multicast capability into an internetwork.
     However, it should be noted that some configurations of MOSPF
     and non-MOSPF routers may produce unexpected failures in
     multicast routing (see Section 6.1 of [MOSPF]).

   o MOSPF does not include the ability to tunnel multicast
     datagrams through non-multicast routers. A tunneling capability
     has proved valuable when used by the DVMRP protocol in the

MBONE. However, it is assumed that, since MOSPF is an intra-AS
protocol, multicast can be turned on in enough of the Autonomous
System's routers to achieve the required connectivity without
resorting to tunneling. The more centralized control that exists
in most Autonomous Systems, when compared to the Internet as a
whole, should make this possible.

o In addition to calculating a separate datagram path for each
  [source network, multicast destination] combination, MOSPF
  can also vary the path based on IP Type of Service (TOS). As
  with OSPF unicast routing, TOS-based multicast routing is
  optional, and routers supporting it can be freely mixed with
  those that don't.

o MOSPF supports all network types that are supported by the base
  OSPF protocol: broadcast networks, point-to-points networks and
  non-broadcast multi-access (NBMA) networks. Note however that
  IGMP is not defined on NBMA networks, so while these networks
  can support the forwarding of multicast datagrams, they cannot
  support directly connected group members.

o MOSPF supports all Autonomous System configurations that are
  supported by the base OSPF protocol. In particular, an algorithm
  for forwarding multicast datagrams between OSPF areas
  is included. Also, areas with configured virtual links can
  be used for transit multicast traffic.

o A way of forwarding multicast datagrams across Autonomous
  System boundaries has been defined. This means that a multicast
  datagram having an external source can still be forwarded
  throughout the Autonomous System. Facilities also exist for
  forwarding locally generated datagrams to Autonomous System exit
  points, from which they can be further distributed. The
  effectiveness of this support will depend upon the nature of the
  inter-AS multicast routing protocol. The one assumption that
  has been made is that the inter-AS multicast routing protocol
  will operate in an reverse path forwarding (RPF) fashion:
  namely, that multicast datagrams originating from an external
  source will enter the Autonomous System at the same place that
  unicast datagrams destined for that source will exit.

o To deal with the fact that the external unicast and multicast
  topologies will be different for some time to come, a
  way to indicate that a route is available for multicast but
  not unicast (or vice versa) has been defined. This for example
  would allow a MOSPF system to use DVMRP as its inter-AS
  multicast routing protocol, while using BGP as its inter-AS
  unicast routing protocol.

o For those physical networks that have been assigned multiple
  IP network/subnet numbers, multicast routing can be disabled
  on all but one OSPF interface to the physical network. This
  avoids unwanted replication of multicast datagrams.

o For those networks residing on Autonomous System boundaries,
  which may be running multiple multicast routing protocols
  (or multiple copies of the same multicast routing protocol),
  MOSPF can be configured to encapsulate multicast datagrams
  with unicast (rather than multicast) link-level destinations.
  This also avoids unwanted replication of multicast datagrams.

o MOSPF provides an optimization for IP multicast's "expanding
  ring search" (sometimes called "TTL scoping") procedure. In
  an expanding ring search, an application finds the nearest
  server by sending out successive multicasts, each with a
  larger TTL. The first responding server will then be the
  closest (in terms of hops, but not necessarily in terms of
  the OSPF metric). MOSPF minimizes the network bandwidth
  consumed by an expanding ring search by refusing to forward
  multicast datagrams whose TTL is too small to ever reach a
  group member.

2. Security architecture

   All MOSPF protocol packet exchanges (excluding IGMP) are specified by
   the base OSPF protocol, and as such are authenticated. For a
   discussion of OSPF's authentication mechanism, see Appendix D of
   [OSPF].

3. MIB support

   Management support for MOSPF has been added to the base OSPF V2 MIB
   [OSPF MIB]. These additions consist of the ability to read and write
   the configuration parameters specified in Section B of [MOSPF],
   together with the ability to dump the new group-membership-LSAs.

4. Implementations

   There is currently one MOSPF implementation, written by Proteon, Inc.
   It was released for general use in April 1992. It is a full MOSPF
   implementation, with the exception of TOS-based multicast routing. It
   also does not contain an inter-AS multicast routing protocol.

   The multicast applications included with the Proteon MOSPF
   implementation include: a multicast pinger, console commands so that
   the router itself can join and leave multicast groups (and so respond
   to multicast pings), and the ability to send SNMP traps to a

multicast address. Proteon is also using IP multicast to support the
tunneling of other protocols over IP. For example, source route
bridging is tunneled over a MOSPF domain, using one IP multicast
address for explorer frames and mapping the segment/bridge found in a
specifically-routed frame's RIF field to other IP multicast
addresses. This last application has proved popular, since it
provides a lightweight transport that is resistant to reordering.

The Proteon MOSPF implementation is currently running in
approximately a dozen sites, each site consisting of 10-20 routers.

Table 1 gives a comparison between the code size of Proteon's base
OSPF implementation and its MOSPF implementation. Two dimensions of

	lines of C	bytes of 68020 object code
OSPF base	11,693	63,160
MOSPF	15,247	81,956

Table 1: Comparison of OSPF and MOSPF code sizes

size are indicated: lines of C (comments and blanks included), and
bytes of 68020 object code. In both cases, the multicast extensions
to OSPF have engendered a 30% size increase.

Note that in these sizes, the code used to configure and monitor the
implementation has been included. Also, in the MOSPF code size
figure, the IGMP implementation has been included.

5. Testing

Figure 1 shows the topology that was used for the initial debugging
of Proteon's MOSPF implementation. It consists of seven MOSPF
routers, interconnected by ethernets, token rings, FDDIs and serial
lines. The applications used to test the routing were multicast ping
and the sending of traps to a multicast address (the box labeled
"NAZ" was a network analyzer that was occasionally sending IGMP Host
Membership Reports and then continuously receiving multicast SNMP
traps). The "vat" application was also used on workstations (without
running the DVMRP "mrouted" daemon; see "Distance Vector Multicast
Routing Protocol", [RFC 1075]) which were multicasting packet voice
across the MOSPF domain.

The MOSPF features tested in this setup were:

o   Re-routing in response to topology changes.

o   Path verification after altering costs.

o   Routing multicast datagrams between areas.

o   Routing multicast datagrams to and from external addresses.

o   The various tiebreakers employed when constructing datagram
     shortest-path trees.

o   MOSPF over non-broadcast multi-access networks.

o   Interoperability of MOSPF and non-multicast OSPF routers.

```
                                        +---+
    +-----------------------------|RT1|
    |                                   +---+
    |           +---------+             |
    |           |         |             |
    |  +---+     +---+     +---+         |
    |  |RT5|---------|RT2|   |NAZ|       |
    |  +---+     +----+---+   +---+       |
    |           |         |     |         |
    |           |     +-----------------------+
    |           |     |                  |      +
    |           |     |                  |      |
    |           |     |                  |      |  +---+
    |    +-----------+     +             |      |--|RT7|
    |    |         |       |             |      |  +---+
    |    +---+     |       |         +---+      |
    |    |RT4|--------|----------|RT3|----|
    |    +---+        |         +---+      |
    |                 |                    |
    |          +      +                    |
    |          |      +---+                |
    +--------------|----------|RT6|-----------|
               |      +---+                |
               +                           +
```
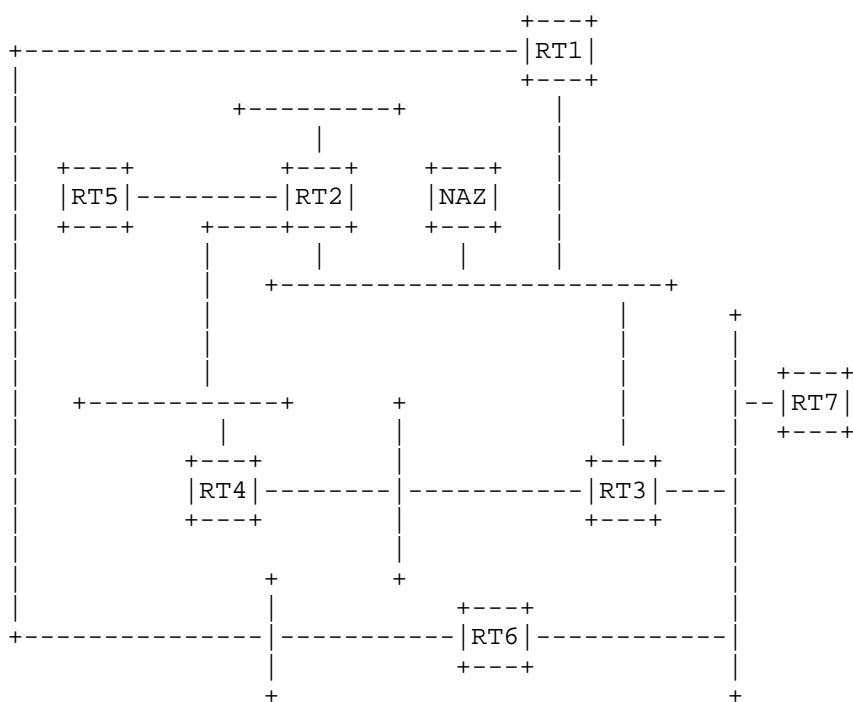
                Figure 1: Initial MOSPF test setup

Due to the commercial tunneling applications developed by Proteon
that use IP multicast, MOSPF has been deployed in a number of
operational but non-Internet-connected sites. MOSPF has been also
deployed in some Internet-connected sites (e.g., OARnet) for testing
purposes. The desire of these sites is to use MOSPF to attach to the
"mbone". However, an implementation of both MOSPF and DVMRP in the
same box is needed; without this one way communication has been
achieved (sort of like lecture mode in vat) by configuring multicast
static routes in the MOSPF implementation. The problem is that there
is no current way to inject the MOSPF source information into DVMRP.

The MOSPF features that have not yet been tested are:

o   The interaction between MOSPF and virtual links.

o   Interaction between MOSPF and other multicast routing protocols
     (e.g., DVMRP).

o   TOS-based routing in MOSPF.

6. A brief analysis of MOSPF scaling

MOSPF uses the Dijkstra algorithm to calculate the path of a
multicast datagram through any given OSPF area. This calculation
encompasses all the transit nodes (routers and networks) in the area;
its cost scales as $O(N*log(N))$ where N is the number of transit nodes
(same as the cost of the OSPF unicast intra-area routing
calculation). This is the cost of a single path calculation; however,
MOSPF calculates a separate path for each [source network, multicast
destination, TOS] tuple. This is potentially a lot of Dijkstra
calculations.

MOSPF proposes to deal with this calculation burden by calculating
datagram paths in an "on demand" fashion. That is, the path is
calculated only when receiving the first datagram in a stream. After
the calculation, the results are cached for use by later matching
datagrams. This on demand calculation alleviates the cost of the
routing calculations in two ways: 1) It spreads the routing
calculations out over time and 2) the router does fewer calculations,
since it does not even calculate the paths of datagrams whose path
will not even touch the router.

Cache entries need never be timed out, although they are removed on
topological changes. If an implementation chooses to limit the
amount of memory consumed by the cache, probably by removing selected
entries, care must be taken to ensure that cache thrashing does not
occur.

The effectiveness of this "on demand" calculation will need to be proven over time, as multicast usage and traffic patterns become more evident.

As a simple example, suppose an OSPF area consists of 200 routers. Suppose each router represents a site, and each site is participating simultaneously with three other local sites (inside the area) in a video conference. This gives 200/4 = 50 groups, and 200 separate datagram trees. Assuming each datagram tree goes through every router (which probably won't be true), each router will be doing 200 Dijkstras initially (and on internal topology changes). The time to run a 200 node Dijkstra on a 10 mips processor was estimated to be 15 milliseconds in "OSPF protocol analysis" ([RFC 1245]). So if all 200 Dijkstras need to be done at once, it will take 3 seconds total on a 10 mips processor. In contrast, assuming each video stream is 64Kb/sec, the routers will constantly forward 12Mb/sec of application data (the cost of this soon dwarfing the cost of the Dijkstras).

In this example there are also 200 group-membership-LSAs in the area; since each group membership-LSA is around 64 bytes, this adds 64*200 = 12K bytes to the OSPF link state database.

Other things to keep in mind when evaluating the cost of MOSPF's routing calculation are:

o Assuming that the guidelines of "OSPF protocol analysis" ([RFC 1245]) are followed and areas are limited to 200 nodes, the cost of the Dijkstra will not grow unbounded, but will instead be capped at the Dijkstra for 200 nodes. One need then worry about the number of Dijkstras, which is determined by the number of [datagram source, multicast destination] combinations.

o A datagram whose destination has no group members in the domain can still be forwarded through the MOSPF system. However, the Dijkstra calculation here depends only on the [datagram source, TOS], since the datagram will be forwarded along to "wild-card receivers" only. Since the number of group members in a 200 router area is probably also bounded, the possibility of unbounded calculation growth lies in the number of possible datagram sources. (However, it should be noted that some future multicast applications, such as distributed computing, may generate a large number of short-lived multicast groups).

o By collapsing routing information before importing it into the area/AS, the number of sources can be reduced dramatically. In particular, if the AS relies on a default external route, most external sources will be covered by a single Dijkstra calculation (the one using 0.0.0.0 as the source).

One other factor to be considered in MOSPF scaling is how often cache
entries need to be recalculated, as a result of a network topology
change. The rules for MOSPF cache maintenance are explained in
Section 13 of [MOSPF]. Note that the further away the topology change
happens from the calculating router, the fewer cache entries need to
be recalculated. For example, if an external route changes, many
fewer cache entries need to be purged as compared to a change in a
MOSPF domain's internal link. For scaling purposes, this is exactly
the desired behavior. Note that "OSPF protocol analysis" ([RFC 1245])
bears this out when it shows that changes in external routes (on the
order of once a minute for the networks surveyed) are much more
frequent than internal changes (between 15 and 50 minutes for the
networks surveyed).

7. Known difficulties

The following are known difficulties with the MOSPF protocol:

o When a MOSPF router itself contains multicast applications, the
  choice of its application datagrams' source addresses should be
  made with care. Due to OSPF's representation of serial lines,
  using a serial line interface address as source can lead to
  excess data traffic on the serial line. In fact, using any
  interface address as source can lead to excess traffic, owing to
  MOSPF's decision to always multicast the packet onto the source
  network as part of the forwarding process (see Section 11.3 of
  [MOSPF]). However, optimal behavior can be achieved by assigning
  the router an interface-independent address, and using this as
  the datagram source.

  This concern does not apply to regular IP hosts (i.e., those
  that are not MOSPF routers).

o It is necessary to ensure, when mixing MOSPF and non-multicast
  routers on a LAN, that a MOSPF router becomes Designated Router.
  Otherwise multicast datagrams will not be forwarded on the LAN,
  nor will group membership be monitored on the LAN, nor will the
  group-membership-LSAs be flooded over the LAN. This can be an
  operational nuisance, since OSPF's Designated Router election
  algorithm is designed to discourage Designated Router transitions,
  rather than to guarantee that certain routers become
  Designated Router. However, assigning a DR Priority of 0 to all
  non-multicast routers will always guarantee that a MOSPF router
  is selected as Designated Router.

8. Future work

    In the future, it is expected that the following work will be done on
    the MOSPF protocol:

    o More analysis of multicast traffic patterns needs to be done, in
      order to see whether the MOSPF routing calculations will pose an
      undue processing burden on multicast routers. If necessary,
      further ways to ease this burden may need to be defined. One
      suggestion that has been made is to revert to reverse path
      forwarding when the router is unable to calculate the detailed
      MOSPF forwarding cache entries.

    o Experience needs to be gained with the interactions between multiple
      multicast routing algorithms (e.g., MOSPF and DVMRP).

    o Additional MIB support for the retrieval of forwarding cache
      entries, along the lines of the "IP forwarding table MIB" ([RFC
      1354]), would be useful.

9. References

    [Bharath-Kumar] Bharath-Kumar, K., and J. Jaffe, "Routing to
                    multiple destinations in Computer Networks", IEEE
                    Transactions on Communications, COM-31[3], March
                    1983.

    [Deering]   Deering, S., "Multicast Routing in Internetworks
                    and Extended LANs", SIGCOMM Summer 1988
                    Proceedings, August 1988.

    [Deering2]   Deering, S., "Multicast Routing in a Datagram
                    Internetwork", Stanford Technical Report
                    STAN-CS-92-1415, Department of Computer Science,
                    Stanford University, December 1991.

    [OSPF]   Moy, J., "OSPF Version 2", RFC 1583, Proteon,
                    Inc., March 1994.

    [OSPF MIB]   Baker F., and R. Coltun, "OSPF Version 2 Management
                    Information Base", RFC 1253, ACC, Computer Science
                    Center, August 1991.

    [MOSPF]   Moy, J., "Multicast Extensions to OSPF", RFC 1584,
                    Proteon, Inc., March 1994.

    [RFC 1075]   Waitzman, D., Partridge, C. and S. Deering,
                    "Distance Vector Multicast Routing Protocol", RFC
                    1075, BBN STC, Stanford University, November 1988.

    [RFC 1112]   Deering, S., "Host Extensions for IP Multicasting",
                    Stanford University, RFC 1112, May 1988.

    [RFC 1209]   Piscitello, D., and J. Lawrence, "Transmission of IP
                    Datagrams over the SMDS Service", RFC 1209, Bell
                    Communications Research, March 1991.

    [RFC 1245]   Moy, J., Editor, "OSPF Protocol Analysis", RFC
                    1245, Proteon, Inc., July 1991.

    [RFC 1246]   Moy, J., Editor, "Experience with the OSPF
                    Protocol", RFC 1245, Proteon, Inc., July 1991.

    [RFC 1264]   Hinden, R., "Internet Routing Protocol
                    Standardization Criteria", RFC 1264, BBN, October
                    1991.

   [RFC 1390]   Katz, D., "Transmission of IP and ARP over FDDI
                     Networks", RFC 1390, cisco Systems, Inc., January
                     1993.

   [RFC 1354]   Baker, F., "IP Forwarding Table MIB", RFC 1354,
                     ACC, July 1992.

Security Considerations

   Security issues are not discussed in this memo, tho see Section 2.

Author's Address

   John Moy
   Proteon, Inc.
   9 Technology Drive
   Westborough, MA 01581

   Phone: (508) 898-2800
   EMail: jmoy@proteon.com