

Network Working Group
Internet Draft
Expiration Date: April 2004

Keyur Patel
Cisco Systems
Radia Perlman
Sun Microsystems
Dino Farinacci
Greg Shepherd
Procket Networks
Marshall Eubanks
Multicast Technologies

Automatic Multicast Access Protocol

[draft-keyur-amap-00.txt](#)

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

2. Abstract

This document describes a new multicast signaling protocol for solving the "last-mile connectivity" problem for multicast. This protocol is called Automatic Multicast Access Protocol (AMAP). It is intended to facilitate an endnode that resides in a portion of the Internet infrastructure without multicast support, or whose OS does not support IGMPv3 (SSM stype groups).

3. Introduction

An impediment to getting end-to-end multicast connectivity is that,

currently, ISPs seldom enable multicast routing within an entire AS. This makes end-to-end multicast content provision difficult, particularly with respect to Last Mile Multicast.

This document proposes a new multicast signaling protocol known as Automatic Multicast Access Protocol (AMAP) which allows endnodes residing in portions of the Internet infrastructure without any multicast support to join multicast groups. AMAP allows endnodes which run multicast applications to get the efficiencies of a multicast enabled infrastructure without being directly attached to such an infrastructure. This is achieved using a tunneling mechanism. As the multicast infrastructure deployment moves towards the edge of the network, the tunnel diameter is reduced to a point where it is either minimised or no longer needed.

AMAP could be used for both SSM as well as ASM style multicast. For endnodes residing in a portion of the Internet without any support for multicast, this mechanism could be used for ASM groups by using a Replicator address in place of an explicit root address. A

Patel, Perlman, Farinacci, Shepherd, Eubanks

[Page 1]

Internet Draft

[draft-keyur-amap-00.txt](#)

October 2003

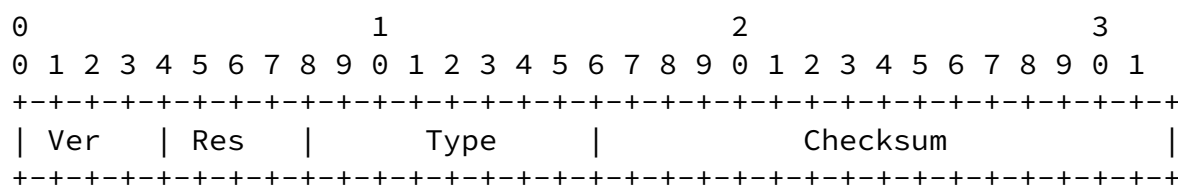
Replicator address is an address of a router which serves as a tunnel endpoint for all/set of receivers within an AS. A well known anycast address, which is a valid unicast address assigned to multiple routers to get "closest replicator capability" defined by IANA, can be used as a Replicator address within an AS.

Any router that does not support this protocol will simply unicast forward the messages according to the rules for forwarding a packet with the Router Alert set. The first multicast capable router that supports AMAP will notice these messages and may form a tunnel with the endnode that generated the message.

[4.](#) AMAP Messages

This section defines different message formats for AMAP. AMAP messages are sent unicast to a particular source or an endnode, or towards a Replicator address. AMAP messages are sent as IP protocol type = "UDP" with the port number TBD by IANA.

An implementation of AMAP must support all the message types described below. Any unrecognized messages should be logged with a fatal warning. Each AMAP message has a fixed size header. The layout of this header is shown below:



AMAP Ver

AMAP Version number is 1.

Reserved

Set to zero on transmission. Ignored upon receipt.

Type

Types for specific AMAP messages. AMAP Types are:

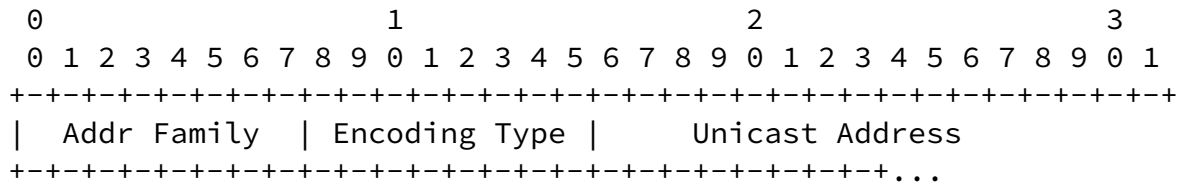
Message Type	Source	Destination
0 = Query Message	Router	Unicast to Endnodes
1 = Join Message	Endnode	Unicast to Source/ Replicator address
2 = Join-Ack Message	Router	Unicast to Endnodes
3 = Auth Ack	Router/Endnode	Unicast to Source/Endnodes
4 = Prune Message	Router/Endnode	Unicast to Source/Endnodes
5 = Prune-Ack Message	Router/Endnode	Unicast to Source/Endnodes
6 = Data Message	Router	Unicast to Endnodes
7 = Notification Message	Router/Endnode	Unicast to Source/Endnodes

Checksum

The checksum is a standard IP checksum, i.e. the 16-bit one's complement of the one's complement sum of the entire AMAP message. For computing the checksum, the checksum field is zeroed.

For IPv6, the checksum also includes the IPv6 "pseudo-header", as specified in [RFC 2460, section 8.1](#) [5]. This "pseudo-header" is prepended to the AMAP header for the purposes of calculating the checksum. The "Upper-Layer Packet Length" in the pseudo-header is set to the length of the AMAP message. The Next Header value used in the pseudo-header is 103. If the packet's length is not an integral number of 16-bit words, the packet is padded with a byte of zero before performing the checksum.

Encoded format for Unicast Source and Group address is shown below:



Addr Family

The AMAP address family of the 'Unicast Address' field of this address.

Values of 0-127 are as assigned by the IANA for Internet Address Families in [8]. Values 128-250 are reserved to be assigned by the IANA for PIM-specific Address Families. Values 251 though 255 are designated for private use. As there is no assignment authority for this space, collisions should be expected.

Encoding Type

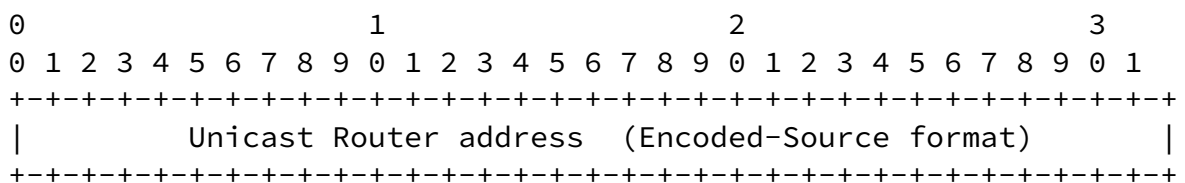
The type of encoding used within a specific Address Family. The value '0' is reserved for this field, and represents the native encoding of the Address Family.

Unicast Address

The unicast address as represented by the given Address Family and Encoding Type.

4.1 AMAP Router Query message format

Query messages are sent by AMAP routers to its tunnel endpoints for periodic refresh of their multicast state.



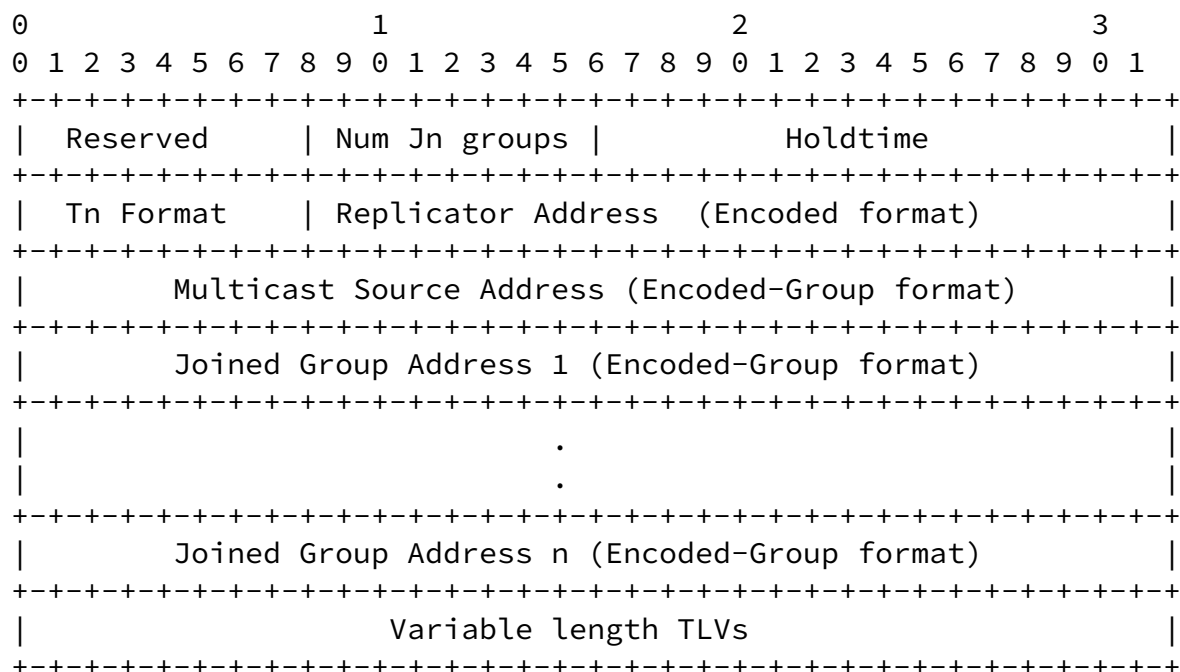
Unicast Router Address

Unicast address of an AMAP router generating Query messages. For format description refer to Encoded-Unicast address.

4.2 AMAP Host Join message format

Host Join messages are sent to assist building and maintenance of tunnel states for (S/*, G) entries. Host Join messages are sent by endnodes towards a Source address/Replicator address. For all (S, G) Host Join messages, the Multicast Source Address will have a non-zero unicast address field. For all (*, G) Host Join messages, the Multicast Source Address will have a zero unicast address field. AMAP implementations are suggested to send separate Host Join messages for

(S, G) and (*, G) entries.



Reserved

Transmitted as zero, ignored upon receipt.

Number of Join Groups

The number of multicast group sets contained in the message.

Holdtime

The amount of time a receiver must keep the tunnel state alive in seconds. If the Holdtime is set to '0xffff', the receiver of this message should hold the state until canceled by the Prune message, or timed out according to local policy. This may be used with dial-on-demand links, to avoid keeping the link up with the periodic Join or Prune messages.

Tunnel Type

A specific type of tunnel that an endnode wants to use with an AMAP Router: 0 (IP-in-IP), 1 (GRE type 0x800 carrying IPv4 as a payload).

Replicator address

Address assigned to an AMAP router. When used in anycast-mode, it is an address assigned to multiple AMAP routers to get "closest replicator

capability". For format description refer to Encoded-Unicast address.

Multicast Source address

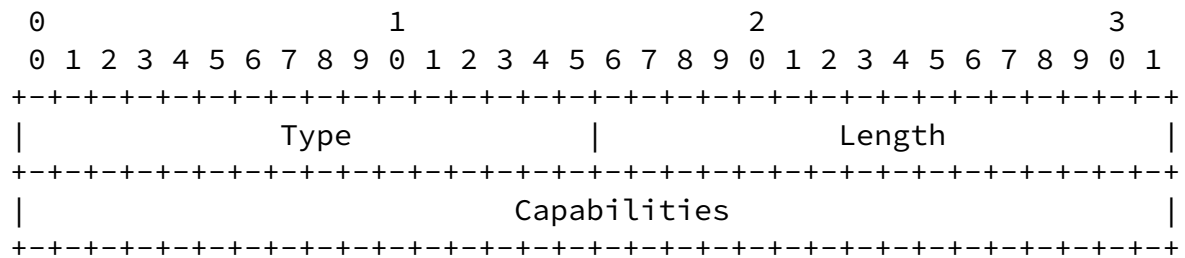
Sender address that the endnode is interested in receiving data from.
For format description refer to Encoded-Unicast address.

Join Group Address 1 .. n

This list contains the Groups that the endnode is interested in receiving data from. For format description refer to Encoded-Unicast address.

Variable length TLVs

The Encoded format for TLVs is defined as follows:



Type

A 16 bit field set to 1.

Length

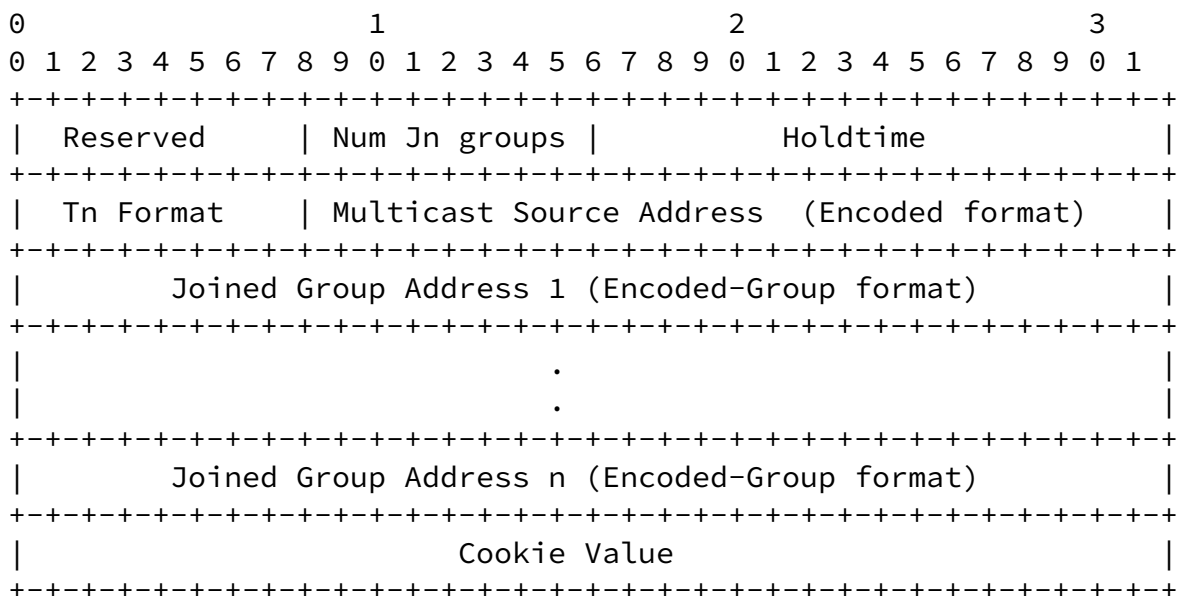
A 16 bit field that indicates the length of the value portion in bytes.

Capabilities

This comprises one or more capability values.

[4.3](#) AMAP Router Join-Ack message format

Router Join-Ack messages are sent to acknowledge the receipt of the Host Join message by AMAP routers.




```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Reserved      |             Holdtime              |  Tn Format      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Replicator Address   (Encoded format)          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Multicast Source Address (Encoded-Group format) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Joined Group Address (Encoded-Group format)    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Reserved

Transmitted as zero, ignored upon receipt.

Holdtime

The amount of time a receiver must keep the tunnel state alive in seconds. If the Holdtime is set to '0xffff', the receiver of this message should hold the state until canceled by the Prune message, or timed out according to local policy. This may be used with dial-on-demand links, to avoid keeping the link up with the periodic Join or Prune messages.

Tn Format

A specific type of tunnel that an endnode wants to use with an AMAP Router: 0 (IP-in-IP), 1 (GRE type 0x800 carrying IPv4 as a payload).

Replicator address

Address assigned to an AMAP router. When used in anycast-mode, it is an address assigned to multiple AMAP routers to get "closest replicator capability". For format description refer to Encoded-Unicast address.

Multicast Source address

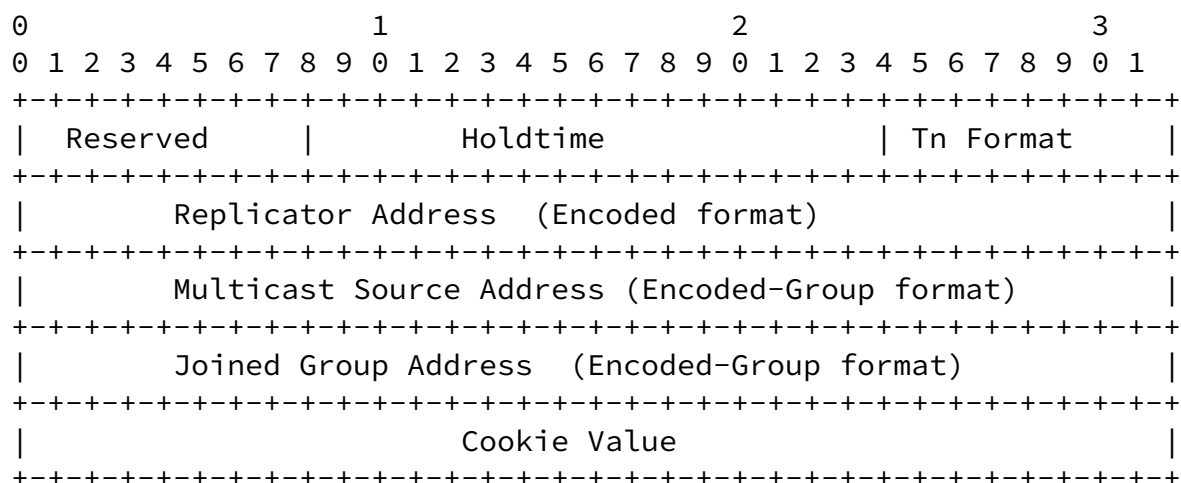
For format description refer to Encoded-Unicast address.

Prune Group Address

Group address to be pruned. For format description refer to Encoded-Unicast address.

4.5 AMAP Prune-Ack message format

Prunes-Ack messages are sent to acknowledge the receipt of the Prune messages.



Reserved

Transmitted as zero, ignored upon receipt.

Holdtime

The amount of time a receiver must keep the tunnel state alive in seconds. If the Holdtime is set to '0xffff', the receiver of this message should hold the state until canceled by the Prune message, or timed out according to local policy. This may be used with dial-on-demand links, to avoid keeping the link up with the periodic Join or Prune messages.

Tn Format

A specific type of tunnel that an endnode wants to use with an AMAP Router: 0 (IP-in-IP), 1 (GRE type 0x800 carrying IPv4 as a payload).

Replicator address

Address assigned to an AMAP router. When used in anycast-mode, it is an address assigned to multiple AMAP routers to get "closest replicator capability". For format description refer to Encoded-Unicast address.

Multicast Source address

Sender address to be pruned. For format description refer to Encoded-Unicast address.

Prune Group Address

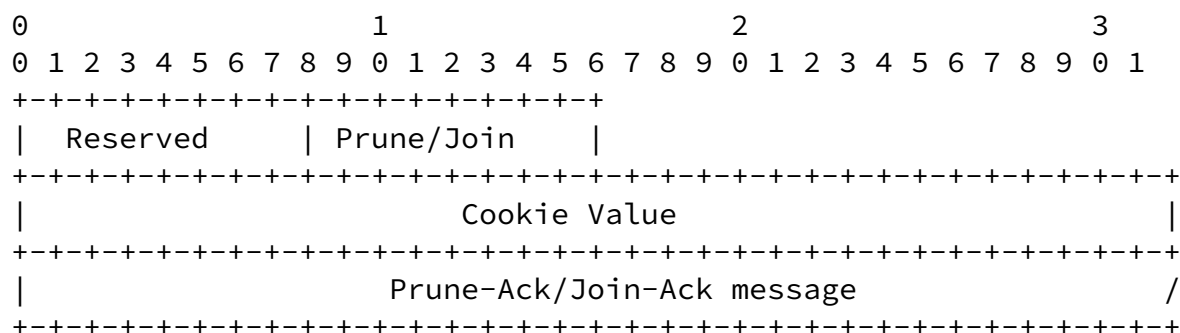
This contains the Group address to be pruned. For format description refer to Encoded-Unicast address.

Cookie Value

Sender's cookie value that is reflected back in an Auth-Ack message.

[4.6](#) AMAP Host Auth-Ack message format

Auth-Ack messages are sent to acknowledge the receipt of a Join-Ack message or a Prune-Ack message.



Reserved

Transmitted as zero, ignored upon receipt.

Prune/Join

Auth message sent in response to 0 (Join-Ack) 1 (Prune-Ack).

Cookie Value

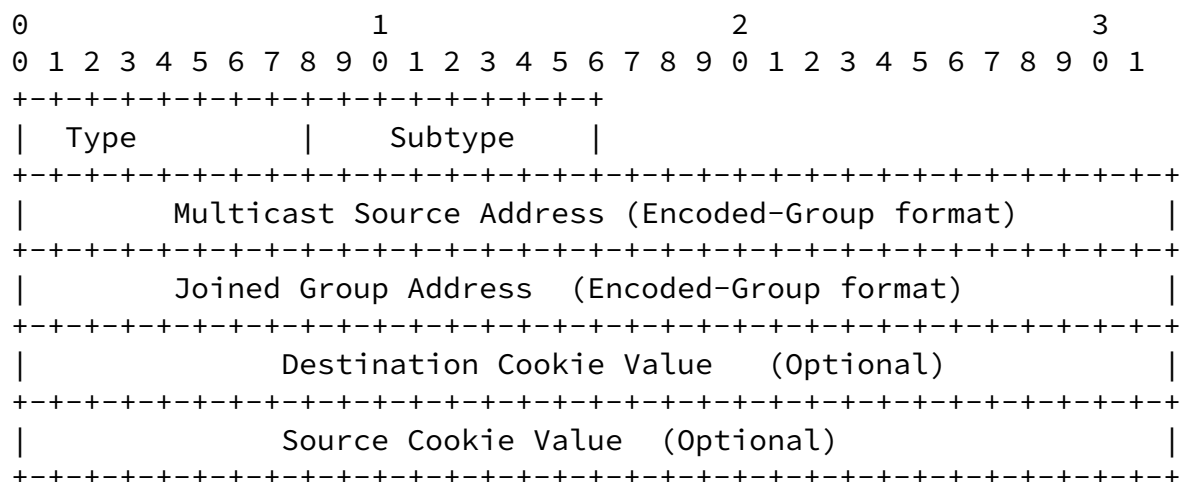
Sender's cookie value that is reflected back in Auth-Ack message.

Prune-Ack/Join-Ack message

Received AMAP Prune-Ack message/Join-Ack message triggering the Auth-Ack message.

[4.7](#) AMAP Notification message format

Notification messages are sent either by an AMAP router or an endnode initiating a tunnel connection.



Type/Subtype

AMAP Notification message are of following types:

Type:	1	Error Message
Subtype:	1	Incorrect Version Number
	2	Incorrect Cookie Value
	3	Incorrect Message Type
	4	Incorrect Address Encoding format
	5	Incorrect Message Length
	6	Unrecognised Route Entry Pruned
	7	Unrecognised Capability
	8	Incorrect Capability length
	9	Incorrect Tunnel format
	10	Incorrect Destination

Multicast Source address

Multicast Source address for which the Notification is sent.
For format description refer to Encoded-Unicast address.

Prune Group Address

Multicast Group address for which the Notification is sent.
This contains the Group address to be pruned.

Destination Cookie Value

Destination receiving Notification message's Cookie value. This field is optional

Source Cookie Value

Source generating Notification message's Cookie value. This field is optional.

All Notification messages must be logged. An implementation can choose to terminate tunnel connections for (S/*, G) route entries for which a Notification message received has cookie values successfully verified.

[5. Protocol Description for Endnodes](#)

AMAP has a separate protocol behavior for endnodes initiating the tunnel connections to join multicast (S/*, G) channels and AMAP routers terminating tunnel connections. This section describes the protocol behavior for the endnodes initiating the tunnel connections.

[5.1](#) Generating Join Messages

There are two main events that trigger generation of the Join messages for initiating (S/*, G) tunnel connections:

- * Applications residing on an endnode wanting to join a particular/set of (S/*, G) groups.
- * Reception of a Query message.

The following subsections describe actions to be taken for each of the above cases.

[5.1.1](#) Applications triggering a Join message

Whenever an endnode running multicast applications in a non-multicast network wishes to join a particular (S/*, G) channel, it signals AMAP with relevant parameters: (S/*, G) sets, a Replication address, and a configured cookie value.

AMAP schedules a Join-expire timer with a configured/(default holdtime interval)/3. AMAP creates an appropriate Join message and sends it towards a Replication address if specified, or the Source address.

[5.1.2](#) Reception of a Query message

Whenever an endnode running AMAP receives a Query message, it processes the message. It verifies the querier's router address with the stored router address in the tunnel interface information. It [re-]schedules the Join-expire timer to the join-expire interval. The suggested default value for the join-expire interval is set to the AMAP router's (Holdtime interval)/3 (received in a Join-Ack message). It also [re-]schedules the query-expire timer to an query-expire interval. The suggested default value for the query-expire interval is set to the AMAP router's (Holdtime interval)/3 (received in a Join-Ack message). It then sends a Join message for all (S/*, G) entries for which the querier acts as a tunnel endpoint.

[5.2](#) Generating Prune message

Whenever a particular (S/*, G) channel is no longer required, a Prune message is sent to the tunnel endpoint. An endnode running AMAP schedules the Prune-expire timer to the prune-expire interval. The suggested default value for the prune-expire interval is set to the AMAP router's (Holdtime interval)/3 (received in a Join-Ack message). It then sends a Prune message for a (S/*, G) channel to the AMAP router that acts as a tunnel endpoint.

[5.3](#) Generation of an Auth-Ack message

Auth-Ack message is sent in response to the received Join-Ack message or

a Prune-Ack message. An Auth-Ack message is sent with the cookie value received in a Join-Ack/Prune-Ack message from the AMAP router. Tunnel connection is either considered functional (if sent in response to the Join-Ack message) or deleted (if sent in response to the Prune-Ack message) once an Auth-Ack message is sent by an endnode to the AMAP router.

[5.4](#) Reception of Data message

Data messages are encapsulated in an negotiated tunnel format by AMAP routers. Whenever an endnode receives a data message, its encapsulated format is verified with the stored negotiated format. If the tunnel format differs, then a Notification message is sent and the tunnel connection is terminated. Otherwise, data packets received are decapsulated and sent to appropriate applications.

[5.5](#) Reception of Prune message

Whenever an endnode receives a Prune message for a given (S/*, G) entry from an AMAP router, it checks to verify if the sending AMAP router is acting as a tunnel endpoint. In case of an error, the Prune message is ignored and a Notification message is sent to an AMAP router. Otherwise, the PruneAck-expire timer is scheduled for the prune-expire interval and a Prune-Ack message is sent to an AMAP router. An endnode cookie is passed in the Prune-Ack message.

[5.6](#) Reception of an Auth-Ack message

An Auth-Ack message is received in response to any Prune-Ack message sent by an endnode. Whenever an endnode receives an Auth-Ack message, it processes the message. It verifies cookie values sent within an Auth-Ack message. In the event of an error, a Notification message is sent and an Auth-Ack message is ignored. Otherwise, the PruneAck-expire timer is stopped for all (S/*, G) entries mentioned in the Auth-Ack message. At this point the tunnel is disconnected and the state is removed as described in [6.7].

[5.7](#) Timer expiry

In the event of a join-expire, prune-expire, or a query-expire timeout, a Notification message is sent and the tunnel connection is disconnected.

[6](#). Protocol Description for AMAP Routers

This section describes the protocol behavior for AMAP routers. Multicast routers implementing AMAP will process Join messages according to rules defined in this specification even when the IP destination address is not assigned to the router. To allow a router to recognize AMAP messages addressed to the unicast root address, and place it on the slow path, rather than simply forwarding it towards the specified destination, Join messages will have the router alert option, and the IP protocol type="UDP" with port number TBD by IANA.

[6.1](#) Reception of a Join message

Whenever an AMAP router intercepts a Join message it verifies them with its locally configured policies. If the locally configured policies disallow the router from intercepting Join messages, then it simply forwards the message towards the destination. All the multicast routers forwarding AMAP packets towards the desired destination must use unicast RIB to resolve destination address. Otherwise, the intercepting router processes the received Join message. In the event of an error, the intercepting router should drop the received Join message and send a Notification message to an endnode. Otherwise, the intercepting router schedules a JoinAck-expire timer for join-expire interval and sends a Join-Ack message to the endnode. The suggested default value for the join-expire interval is set to the endnode's (Holdtime interval)/3 (received in a Join message). A router based cookie value is sent in the Join-Ack message.

In an event where an AMAP router processes and resolves a subset of the (S/*, G) entries received in a Join message, the AMAP router should send a Join-Ack message for only those entries which are resolved. For all unresolved (S/*, G) entries, the AMAP router forwards them towards the direction of the Source address/Replication address in the IP destination of the Join message.

[6.2](#) Reception of a Prune message

Whenever an AMAP router receives a Prune message for a given (S/*, G) entry, it checks to verify if the sending endnode is a tunnel endpoint. In case of an error the Prune message is ignored and a Notification message is sent to an endpoint. Otherwise, the PruneAck-expire timer is scheduled

for the prune-expire interval and the Prune-Ack message is sent to an endnode. The suggested default value for the Prune-expire interval is set to the endnode's (Holdtime interval)/3 (received in a Prune message). A router based cookie value is sent in the Prune-Ack message.

[6.3](#) Reception of an Auth-Ack message

Auth-Ack messages are received in response of:

- * AMAP router's Join-Ack message sent
- * AMAP router's Prune-Ack message sent

The following subsections describe actions to be taken for each of the above cases.

[6.3.1](#) Auth-Ack message in response of a Join-Ack message

Whenever an AMAP router receives an Auth-Ack message, it processes the message. It verifies cookie values sent within an Auth-Ack message. In the event of an error, a Notification message is sent and an Auth-Ack message is ignored. Otherwise, the JoinAck-expire timer is stopped for all (S/*, G) entries mentioned in the Auth-Ack message. This signals the successful completion of the tunnel signaling between an endnode initiating the tunnel connection and an AMAP router terminating the tunnel connection. At this point the tunnel is setup and the state is created as described in [6.7]. An AMAP router starts to send multicast packets over the tunnel.

[6.3.2](#) Auth-Ack message in response of a Prune-Ack message

Whenever an AMAP router receives an Auth-Ack message, it processes the message. It verifies cookie values sent within an Auth-Ack message. In the event of an error, a Notification message is sent and an Auth-Ack message is ignored. Otherwise, the PruneAck-expire timer is stopped for all (S/*, G) entries mentioned in the Auth-Ack message. At this point the tunnel is disconnected and the state is removed as described in [6.7]. An AMAP router stops sending multicast packets over the tunnel for all (S/*, G) entries mentioned in the Auth-Ack message.

[6.4](#) Generation of an Auth-Ack message

Auth-Ack message is sent in response to the received Join-Ack message or a Prune-Ack message. An Auth-Ack message is sent with the cookie value received in the Join-Ack/Prune-Ack message from the AMAP router. The tunnel connection is either considered functional (if sent in response to a Join-Ack message) or deleted (if sent in response to a Prune-Ack message) once an Auth-Ack message is sent by an endnode to the AMAP router.

[6.5](#) Generating a Prune message

AMAP routers can generate Prune messages whenever they loose upstream connectivity or due to some local policy reasons. An AMAP router schedules the Prune-expire timer to the prune-expire interval. The suggested

default value for the prune-expire interval is set to endnodes's (Holdtime interval)/3. It then sends a Prune message for (S/*, G) entry to all/a subset of endnodes listed in the oif-list of (S/*, G) entry.

[6.6](#) Generating a Query message

AMAP routers acting as a tunnel endpoint for various (S/*, G) entries will periodically query all its interested tunnel endnodes by sending a Query message. An AMAP router sends Query messages every query-expire interval. The suggested default for the query-expire interval is set to an endnode's (Holdtime interval)/3 (received in a Join message). AMAP routers keeps track of all the tunnel destinations (regardless if the tunnel destination is joined to one or more route entries).

Patel, Perlman, Farinacci, Shepherd, Eubanks

[Page 10]

Internet Draft

[draft-keyur-amap-00.txt](#)

October 2003

Any (S/*, G) entries not refreshed by the endnodes in their Join messages sent in response to Query messages within a join-expire interval will be immediately dropped. This assists AMAP routers sending the Query messages and endpoints replying with the Join messages to detect any unwanted tunnel breakages.

The holdtime used in Join messages and Join-Ack messages determines the maximum waiting time that an AMAP router and an endnode should wait before terminating a tunnel at each end. If an endnode fails to receive a Query message for more than the Holdtime interval advertised by an AMAP router in its Join-Ack message, then it terminates and re-initiates tunnel connection. Similarly if the Join messages are not received for more than the Holdtime interval advertised in the previous Join message, then the router removes the tunnel interface from its (S/*, G) entries.

Any (S/*, G) entries created with a holdtime value of '0xffff' need not be refreshed periodically using Query messages. Such (S/*, G) entries can only be explicitly removed using Prune messages or timed out using local policies.

[6.7](#) Reception of Data messages

Data messages are either received as native multicast, or as encapsulated in a negotiated tunnel format. Whenever an AMAP router receives encapsulated data messages, its encapsulated format is verified with the stored negotiated format. If the tunnel format differs, then the Notification message is sent and tunnel connection is terminated.

If an (S/*, G) entry exists for data messages received, then the messages

are encapsulated with a negotiated tunnel format and forwarded to all the AMAP tunnel endpoints listed in the oif-lists. Otherwise, data messages are dropped and a Prune message is sent to the upstream router.

[6.8](#) Timer expiry

In the event of join-expire, prune-expire, or a query-expire timeout, a Notification message is sent and the tunnel connection is disconnected.

[6.9](#) State Creation and Deletion

AMAP routers will add the tunnel interfaces [8] in its (S/*, G) oif-lists when it receives an Auth-Ack message in response to a Join-Ack message from the endnodes. An implementation must have a configured upper bound on the number of the tunnel interfaces that it can put in oif-lists of any (S/*, G) entry.

If a tunnel interface added is the first interface in an oif-list, and if Multicast router has PIM enabled, then the PIM Protocol should be signaled for generation of a PIM Join message.

AMAP routers will delete the tunnel interfaces from their (S, G) oif-lists when:

- * It receives an Auth-Ack message in response to a Prune-Ack messages from endnodes.
- * Any of the Join-expire, or Prune-expire, timers for the tunnel interfaces expire and its state is not refreshed.
- * If a Join message (in response to a Query message) is not received within a Holdtime interval.
- * If a Query message is not received from a tunnel endpoint within a query-expire interval.
- * AMAP routers sends an Auth-Ack message in response to a Prune-Ack message received from endnodes.

[6.10](#) Signaling and Forwarding Rules:

Whenever multicast routers implementing AMAP receive Join messages from any endnodes, the following is done:

The received (S/*, G) entry is looked up in the Multicast Routing Table. If the entry does not exist, then the (S/*, G) entry is created.

- * If a Join message or a Prune message has a non-zero Source address, (i.e (S, G) and not (*, G)) then the next hop for the Source address check is done. If the next hop is a PIM neighbor, or it is directly connected (and PIM enabled) then a JoinAck-expire timer is scheduled for the join-expire interval and a Join-Ack message is sent to an endnode.

- * If the next hop is neither a PIM neighbor nor directly connected, then the packet is forwarded towards the ip destination address. If the (S/*, G) entry was created as a result of a lookup, then delete the (S/*, G) entry from the Multicast Routing Table.

- * If there is no source address (i.e (*, G)), then check for the group mapping in the RP cache. If the group mapping is found for a particular entry then schedule a JoinAck-expire timer for the join-expire interval and send a Join-Ack message to an endnode. Otherwise, forward the Join message towards the ip destination address. If the (*, G) entry was created as a result of lookup, then delete the (*, G) entry from the Multicast Routing Table.

Whenever Multicast routers implementing AMAP receive an Auth-Ack message for a Join message, the following is done:

- * Stop the JoinAck-expire timer.

- * Add the tunnel interface to the Outgoing Interface List (oif-list) of (S/*, G) entry.

- * For all (S, G) route entries, signal PIM to send a PIM Join message towards the Source address if PIM is enabled.

- * For all (*, G) route entries, signal PIM to send a PIM Join message towards the RPL address if PIM is enabled.

Whenever multicast routers implementing AMAP receive Prune messages from any endnodes, the following is done:

The received (S/*, G) entry is looked up in the Multicast Routing Table. If the entry does not exist, then a Notification message is sent for the received (S/*, G) entry.

- * If the entry is found, then a PruneAck-expire timer is scheduled for the prune-expire interval and a Prue-Ack message is sent to an endnode.

Whenever multicast routers implementing AMAP receive an Auth-Ack message for Prune messages, the following is done:

The received (S/*, G) entry is looked up in the Multicast Routing Table. If the entry does not exist, then a Notification message is sent for the received (S/*, G) entry.

- * Stop the PruneAck-expire timer.

- * Remove the tunnel interface to the Outgoing Interface List (oif-list) of (S/*, G) entry.

- * For all (S, G) route entries, signal PIM to send a PIM Prune message towards the Source address if PIM is enabled.

- * For all (*, G) route entries, signal PIM to send a PIM Prune message towards the RPL address if PIM is enabled.

Patel, Perlman, Farinacci, Shepherd, Eubanks

[Page 12]

Internet Draft

[draft-keyur-amap-00.txt](#)

October 2003

[6.11](#) Aggregation

TBD

[6.12](#) Tunnel Chaining

TBD

[7](#). AMAP Message Processing

[7.1](#) Router receiving AMAP Join message

Endnode		Router
-----		-----
Send AMAP Join message for interested (S, G) groups	----->	Receive AMAP Join message
Receive AMAP Notification message	<-----	if (error in message processing) Send Notification message with Type="Error", Subtype="Appropriate Message Error Code"

For each (S, G) received entry,

Lookup (S, G) entry.

```

        if (entry found) {

            schedule Join-expire Timer
            for a received interface

            if (cookie value != old cookie
                value) {

                store the received cookie
                value
            }

Receive AMAP Join-Ack <-----
message                                send AMAP Join-Ack message
                                        with router cookie value

        } else {

            create (S, G) entry

            Nexthop Lookup for (S).

            if ((S == Pim Nbr) ||
                (S == directly connected)) {

                schedule Join-expire Timer
                for a received interface

                store the received cookie value

Receive AMAP Join-Ack <-----
message                                send AMAP Join-Ack message
                                        with router cookie value

            } else if (!S &&
                group mapping found in RP Cache) {

                schedule Join-expire Timer
                for a received interface

                store the received cookie value

Receive AMAP Join-Ack <-----
message                                send AMAP Join-Ack message
                                        with router cookie value

```

```

    } else if (destination in IP (!local

```

```

interface)) {

stop the Join-expire Timer

if (multicast enabled) {

lookup in MRIB and

forward the Join towards
source.

} else {

lookup in unicast RIB and

forward the Join towards
the source.

}
} else {
Send Notification message with
Type="Error", Subtype=
"Incorrect Destination"
}
}
}

```

Receive AMAP Notification message <-----

[7.2](#) Router receiving AMAP Auth-Ack for the Join-Ack message

Endnode		Router
-----		-----
Send AMAP Auth-Ack for the Join-Ack message	----->	Receive AMAP Auth-Ack for the Join-Ack message
Receive AMAP Notification message	<-----	if (error in message processing) Send Notification message with Type="Error", Subtype= "Appropriate Message Error Code"
Receive AMAP Notification message	<-----	if (error in cookie received) Send Notification message with Type="Error", Subtype= "Incorrect Cookie Value"
		Lookup received (S, G) entry if (found) { Stop the Join-expire Timer Add the tunnel interface to outgoing list. }

```

if (PIM enabled &&
    first interface) {
    Send PIM Join message
}
} else {
    Send Notification message with
    Type="Error", Subtype=
    "Unrecognised Route Entry
    Pruned"
}

```

Receive AMAP Notification message <-----

[7.3](#) Router receiving AMAP Prune message

```

Endnode
-----
Send AMAP Prune message ----->
Router
-----
Receive AMAP Prune message

```

Patel, Perlman, Farinacci, Shepherd, Eubanks

[Page 14]

Internet Draft

[draft-keyur-amap-00.txt](#)

October 2003

```

Receive AMAP Notification message <-----
if (error in message processing)
    Send Notification message with
    Type="Error", Subtype=
    "Appropriate Message Error
    Code"

Lookup received (S, G) entry and
tunnel interface in oif-list

if (found) {
    Start the Prune-expire Timer
    for received interface

Receive AMAP Prune-Ack message <-----
    Send Prune-Ack message with
    the router cookie value

} else {
    Send Notification message with
    Type="Error", Subtype=
    "Unrecognised Route Entry
    Pruned"
}

```

Receive AMAP Notification message <-----

[7.4](#) Router receiving AMAP Auth-Ack message for the Prune-Ack message

```

Endnode
-----
Router
-----

```

<pre> ----- Send AMAP Auth-Ack -----> for the Prune-Ack message Receive AMAP <----- Notification message Receive AMAP <----- Notification message </pre>	<pre> ----- Receive AMAP Auth-Ack for the Prune-Ack message if (error in message processing) Send Notification message with Type="", Subtype= "Appropriate Message Error Code" if (error in cookie received) Send Notification message with Type="Error", Subtype= "Incorrect Cookie Value" Lookup received (S, G) entry and tunnel interface in oif-list if (found) { Stop the Prune-expire Timer for a received interface Remove the tunnel interface to outgoing list. if (PIM enabled && last interface) { Send PIM Prune message Remove (S, G) entry } } else { Send Notification message with Type="Error", Subtype= "Unrecognised Route Entry Pruned" } </pre>
<pre> Receive AMAP <----- Notification message </pre>	<pre> } else { Send Notification message with Type="Error", Subtype= "Unrecognised Route Entry Pruned" } </pre>

7.5 Hosts receiving AMAP Join-Ack or Prune-Ack messages

<pre> Endnode ----- Receive AMAP Join/Prune-Ack <----- message </pre>	<pre> Router ----- Send Join/Prune-Ack message with the router cookie value </pre>
--	--

Send Auth-Ack message with Router's cookie value ----->	Receive Auth-Ack messages received in Join/Prune message
--	---

7.6 Hosts receiving AMAP Query messages

Endnode -----	Router -----
	Re-schedule Query time
	schedule Join-expire Timer for a received interface
Receive AMAP Query message	<----- Send Query message
Send AMAP Join message for interested (S, G) ----->	Receive AMAP Join messages groups

7.7 Routers sending AMAP Prune messages

Endnode -----	Router -----
	Start the Prune-expire Timer for a tunnel interface
Receive an AMAP Prune message	<----- Send an AMAP Prune message with the router cookie value
if (error in message processing) Send Notification message with -> Type="Error", Subtype= "Appropriate Message Error Code"	Receive AMAP Notification message if (cookie value matches stored cookie value) { Lookup received (S, G) entry and tunnel interface in oif-list if (found) { Stop the Prune-expire Timer for a received interface Remove the tunnel interface to outgoing list. } }
Send AMAP Prune-Ack message ----->	Receive Prune-Ack message with the host cookie value
	Lookup received (S, G) entry and


```

tunnel interface in oif-list

if (found) {
    Stop the Prune-expire Timer
    for a received interface

    Remove the tunnel interface to
    outgoing list.

} else {
    Receive AMAP      <----- Send Notification message with
    Notification message      Type="Error", Subtype=
                              "Unrecognised Route Entry
                              Pruned"
}

```

8. Tunnel Interfaces

Because not all intermediate routers support native ip multicast, AMAP requires all the oifs on which it receives Host Join or Prune messages from receivers which are not directly connected to be created as tunnel interfaces. In practice, tunnels typically use either IP-IP [[Perk96](#)] or Generic Routing Encapsulation (GRE) [[Han94a](#),[Han94b](#)], although, other encapsulation methods are acceptable.

9. Security Considerations

AMAP does not change any multicast security issues. PIM is open to many types of resource overload. For instance, a node which transmits from many bogus source addresses will cause PIM routers to join to many (S, G) pairs, eventually exhausting the state. PIM routers must protect themselves by limiting the amount of state for multicast, so that a denial of service attack on multicast will not destroy unicast.

With LAN-based IGMP join message, only nodes on the LAN can initiate a join message, so if the LAN is physically protected, and all routers along the path to the RP (or S in an (S, G) join) are trusted, it might be somewhat harder for a node to create a join state untraceably than it would with this tunnel proposal. With this tunnel proposal, the join message is sent over a multi-hop path.

As stated in the document, a router is configured with a maximum number of tunnels it is willing to accept, so an attack to deplete its tunnel resources will only affect attempts to create tunnels,

and not cause any other denial of service.

Additionally, the router to whom tunnels will be created is likely to be the entry router at an ISP, and it can consider an entire customer network to be a LAN. If suspiciously large number of tunnels are being created from that customer network, then the router can start discriminating against join messages from that customer network, when resources are being depleted.

Eventually, cryptographic authentication can be added between the joiner and the router, by having potential joiners obtain a key (or certificate) from the router before they are allowed to join, and having join messages cryptographically authenticated.

10. Acknowledgements

We express our thanks to Alex Zinin, Lorenzo Vicisano, liming Wei, Tom Pusateri, and Nidhi Bhaskar for their review and comments on the earlier versions of the draft.

11. References

- [PIM-SM] Bill Fenner, Mark Handley, Hugh Holbrook, Isidor Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)".
- [Han94a] Hanks, S., Li, T, Farinacci, D., and P. Traina, "Generic Routing Encapsulation", [RFC 1701](#), NetSmiths, Ltd., and cisco Systems, October 1994.
- [Han94b] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation over IPv4 networks", [RFC 1702](#), NetSmiths, Ltd., cisco Systems, October 1994.
- [Perk96] Perkins, C., "IP Encapsulation within IP", [RFC 2003](#), October 1996.

12. Author Information

Keyur Patel
Cisco Systems. Inc.
email: keyupate@cisco.com

Radia Perlman
Sun Microsystems. Inc.
email: Radia.Pperlman@sun.com

Dino Farinacci
Procket Networks Inc.
email: dino@procket.com

Greg Shepherd
Procket Networks Inc.
email: shep@procket.com

Marshall Eubanks
Multicast Technologies Inc.
email: tme@multicasttech.com