TSVWG
Internet-Draft
Intended status: Experimental
Expires: March 14, 2015

C. Bestler, Ed.
R. Novak
Nexenta

September 10, 2014

Creation of Transactional Subset Multicast Groups
draft-bestler-transactional-subset-multicast-00

Abstract

This memo presents techniques for controlling the membership of multicast groups which are constrained to be a subset of a pre-existing multicast group, where such subset groups are only used for short duration transactions which are multicast to a subset of the larger multicast group.

Editor's Note

The proper working group for this draft has not yet been determined. Alternate working groups include PIM and INT.

Nexenta has been developing a multicast based transport/storage protocol for Object Clusters at Nexenta. This applies multicast datagrams to creation and replication of Objects such as those supported by the Amazon Simple Storage Service ("S3") protocol or the OpenStack Object Storage service ("Swift"). Creating replicas of object payload on multiple servers is an inherent part of any storage cluster, which makes multicast addressing very inviting. There are issues of congestion control and reliability to settle, but new Layer 2 capabilities such as DCB (Data Center Bridging) make this doable.

However, we found that the existing protocols for controlling multicast group membership (IGMP and MLD) are not suitable for our storage application. The Authors doubt this is unique to a single application. It should apply to many clusters that have a need to distribute transactional messages to dynamically selected subsets of a group within a cluster to multiple known recipients.

Computational clusters using MPI are also potential users of transactional multicasting. Inter-server replication in a pNFS cluster is another.

These are just examples of synchronizing cluster data where the synchronization does not replicate all of the shared data with the entire cluster. But these are merely initial hunches, working group feedback is expected to refine characterization of the applicability of transactional subset multicast groups.

This submission, and ensuing discussion of this draft and its successors will make reference to specific applications, including the Nexenta Replicast protocol for multicast replication in Nexenta's Cloud Copy-on-Write (CCOW) Object Cluster used in the NexentaEdge product. Such examples are merely for illustrative purposes. Any IETF standardization of the Replicast storage protocols would be done via the Storm or NFS groups, and would require adoption of a definition of Object Storage as a service before standardizing any specific protocol for providing Object Storage services.

At this stage in drafting message formats have not yet been set for the standardized version of the protocol. The pre-standard version was limited to a single L2 physical network, which would be an inappropriate limitation for an IETF standard. Working Group feedback on the format of these messages will be sought during the consensus building process.

Status of This Memo

Copyright Notice

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Existing standards for controlling the membership of multicast groups can be characterized as being Join-driven. These include [RFC3376],[RFC3810], [RFC4541] and [RFC4604]. Due to their inherent latency these techniques prove to be unsuitable for maintaining large sets of related multiast groups. This memo details a new method of maintaining such large sets of related multicast groups when they are all subsets of a single master reference group. This is not a restriction for most cluster-oriented applications which could use transactional multicasting.

Transactional Subset Multicasting defines techniques that extends existing control of a reference multicast group to a potentially large set of multicast addresses used with a VLAN within each local subnet that the reference multicast group reaches.

This specification makes no modifications to the forwarding of multicast packets nor to the communications between mrouters. New methods are defined to set Layer 2 multicast forwarding rules on switches within each of the relevant Layer 2 subnets.

1.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Motivation

Transactional Subset Multicast groups are maintained within each VLAN. A 'Forwarding Control Agent' is defined within each VLAN that is responsible for applying the forwarding information known for a reference multicast group to efficiently set layer 2 multicast forwarding rules within each local network.

The functionality of the Forwarding Control Agent is best understood as extending the functionality of IGMP/MLD Snooping (See [RFC4541]).

An IGMP/MLD snooper interprets IGMP (see [RFC3376]) or MLD (see [RFC3810]) messages to translate their Layer 3 objectives into Layer 2 multicast forwarding rules.

A Forwarding Control Agent interprets new messages defined in this specification for a newly defined class of transactional subset multicast groups into the same Layer 2 multicast forwarding rules. Strategies for implementing Forwarding Control Agents would include

extending IGMP/MLD snooping implementations or building the Forwarding Control Agent external to the existing L2 switch software.

The per transaction costs of using such groups are far lower than with the existing methods. The ongoing maintenance work for multicast forwarding elements is limited to the reference multicast group, it is not replicated for each of the subset transactional multicast groups.

3. An Example Application

The Replicast (see [Replicast]) usage of transactional subset multicasting involves:

- Taking a Cryptographic Hash of each chunk to be stored. This "hash id" is used with a distributed hash table to determine a conventional multicast group which will be used to negotiate placement of the chunk. This is the reference multicast group. Replicast refers to it as a "Negotiating Group".

- Multicasting a request to put the chunk to the reference multicast group. Receiving storage nodes will respond with a bid on when

they could store that chunk, or an indication that they already have that chunk stored. Each of the storage nodes is offering a provisional reservation of its input capacity for a specific time window.

- Assuming that the chunk is not already stored, selecting the best responses to make a transactional subset group. Determination of 'best' typically is driven by the earliest possible completion of the transaction, but may factor the current available storage capacity on each of the storage nodes as well.

- Form or select a "rendezvous group" which will be used to transfer the chunk. When the core network is non-blocking, the transfer will be able to proceed at close to full wire speed at the reserved time because each of the selected storage nodes has reserved its input capacity for bulk payload exclusively. A multicast message to the reference group informs both those selected and those not selected for the rendezvous transfer. Those not selected will release the provisional reservation.

- At the designated time, multicast the chunk payload to the transactional subset multicast group.

- Each recipient validates the cryptographic hash of the received data, and unicasts a positive or negative acknowledgement to the sender.

- If sufficient valid copies have been positively acknowledge, the transaction is complete. Otherwise it is retried.

4. Generalized Usage of Transactional Subset Multicast Groups

Beyond a specific application, the generalized potential for dramatic savings is that transactional messaging within a cluster is a radically different use-case from traditional multicast. The set of factors that differentiates this class of applications can be examined through a series of questions:

- How is the group Selected? Section 5.2.1

- What are the endpoints that receive the messages? Section 5.2.2

- What is the duration of the group?Section 5.2.3
- Who are the potential members of the group? Section 5.2.4
- How much latency does the application tolerate? Section 5.2.5
- What must be done to maintain the group? Section 5.2.6

5. Transactional Subset Multicast Groups

5.1. Definition

A Transactions Subset Multicast Group is a multicast group which:

- Is derived from a pre-existing multicast group created by means independent of this standard. The membership of this derived group is a subset of the reference existing multicast group.
- Has a multicast group address which is part of a block allocated for transactional multicast groups.
- Will only be used for the duration of a transaction. A network failure or re-configuration during the transaction will require an upper layer retry of the transaction. Transactional Subset Multicast groups are not suitable for streaming of content. Transactional subset multicast groups may be persistent, in that the same group continues to exist and be used for a series of transactions. But each message sent to the group is part of a single short duration transaction.

5.1.1. Dynamic Specification versus Dynamic Selection

There are two basic strategies for managing the membership of subset multicast groups:

- Dynamic Specification: The selected members join a group that had been pre-selected for the transaction.

- o Dynamic Selection: A pre-existing group is selected to match the subset desired. That group is allocated for this purpose and used for the transaction.

These two strategies can also be combined to form a hybrid strategy. If there is a pre-existing group for the desired membership list it is allocated and used, otherwise an available group is allocated and re-configured to have the required membership.

5.1.2. Push vs. Join

Existing methods for managing membership of a multicast group can be characterized as Join protocols. The receivers may join the group, or subscribe to a specific source within a group, but the receivers of multicast messages control their reception of multicast messages.

This model is well suited for multimedia transmission where the sender does not necessarily know the full set of endpoints receiving its multicast content. In many cluster application the sender has determined the set of receivers. Requiring the sender to communicate with the recipients so that they can Join the group adds latency to the entire transaction.

However, there would be a serious security concern if transactional multicasting is not limited to transactional subset multicasting. Requiring that every member of a subset multicast group already be a member of a reference multicast group ensures that no new method of sending traffic is being created. Without this guarantee a denial-of-service attacker could simply push a multicast group membership listing 1000 members, then flood that multicast group. The amount of traffic delivered to the aggregate destinations would be multiplied by a factor of 1000.

Transactional subset multicasting is defined to eliminate the latency required for Join-directed multicast group membership, while avoiding creating a new attack vector for denial-of-service flooding.

5.2. Applicability

Transactional Subset Multicast Groups are applicable for applications that want to reduce overall latency by reducing the number of round-trips required for their transactions when identical content must be delivered to multiple cluster members, but the selected members are a subset of a larger group that must be dynamically selected.

Parallel processing of payload and/or storage of payload are the primary examples of such a pattern of communications.

Examples of such applications include:

- Computational Clusters, particularly those using MPI (see [MPI])
- Storage applications, including:
  - pNFS (See [RFC5661]).
  - Amazon Simple Storage Service (S3) (See [AmazonS3]).
  - OpenStack Object Storage (Swift) (See [Swift]).

Dynamic selection of subsets ultimately enables multiple concurrent transfers to occur, which would not have been possible if the message had been sent to the entire reference multicast group. Applications with relatively small payload to be multicast may find it easier to use simple multicast and slightly over-deliver the message.

5.2.1. How is the Group Selected?

In Join-directed multicasting the membership of a multicast group is controlled by the listeners joining and leaving the group. The sender does not control or even know the recipients. This matches the multicast streaming use-case very well. However it does not match a cluster that needs to distribute a transactional message to a subset of a known cluster.

The target group is also assumed to be stable for a long sequence of packets, such as streaming a video. The targeted applications direct transactions to a subset of a stable group.

One example of the need to distribute a transactional message to a subset of a known cluster is replication of data within an object cluster. A set of targets has been selected through an higher layer protocol. Joi-directed group setup here adds excessive latency to the process. The targets must be informed of their selection, they must execute IGMP joins and confirm their joining to the source

before the multicast delivery can begin. Only replication of large storage assets can tolerate this setup penalty.

A distributed computation may similarly have data that is relevant to a specific set of recipients within the cluster. Performing the distribution serially to each target over unicast point-to-point connections uses excessive bandwidth and increases the transactions' latency. It is also undesirable to incur the latency of Join-driven multicast group setup.

This specification creates two methods for a sender to form or select a multicast group for transactional purposes. With these methods no further transmissions are required from the selected targets until the full transfer is complete.

The restriction that the targeted group must be a subset of an existing multicast group is necessary to prevent a denial-of-service flooding attack. Transactional multicast groups that were not restricted to being a subset of an existing multicast group could be used to flood a large number of targets that were unprepared to process incoming multicast datagrams.

5.2.2. What are the endpoints that receive the messages?

The endpoints of the transactional messages may be higher layer entities, where each network endpoint supports multiples instances of the higher layer entities. For example, a storage application may have IP addresses associated with specific virtual drives, as opposed to an IP address associated with a server that hosted multiple virtual drives.

Having an IP address for each drive makes migrating control over that drive to a new server easier, and allows the servers to direct incoming payload to the correct drive.

5.2.3. What is the duration of the group?

Join-directed multicasting is designed primarily for the multicast streaming use-case. A group has an indefinite lifespan, and members come and go at any time during this lifespan, which might be measured in minutes, hours or days.

Transaction multicasting is designed to support applications where a transaction lasts for microseconds or milliseconds (possibly even seconds). Transactional multicasting seeks to identify a multicast group for the duration of sending a set of multicast datagrams

related to a specific transaction. Recipients either receive the entire set of datagrams or they do not. Multicast streaming

typically is transmitting error tolerant content, such as MPEG encoded material. Transaction multicasting will typically transmit data with some form of validating signature and transaction identifier that allows each recipient to confirm full reception of the transaction.

This obviously needs to be combined with applicable congestion control strategies being deployed by the upper layer protocols. The Nexenta Replicast protocol only does bulk transfers against reserved bandwidth, but there are probably as many solutions for this problem as there are applications. Replicast relies upon IEEE I802.1 Datacenter Bridging (DCB) protocols such as Priority Flow Control and Congestion Notification to provide no-drop service. The DCB protocols deal with the fine timing of congestion avoidance, but require higher layer transport or application protocols to keep the sustained traffic rates below the sustained capacity. Creating explicit reservations for bulk transfers is the main method for accomplishing this.

The relevant DCB protocols include:

- Congestion Notification:[IEEE.802.1Qau-2011]
- Enhanced Transmission Selection:[IEEE.802.1Qaz-2011]
- Priority Flow Control[IEEE.802.1Qbb-2011]

The important distinction between Replicast and conventional multicast applications is that there is no need to dynamically adjust multicast forwarding tables during the lifespan of a transaction, while IGMP and MLD are designed to allow the addition and deletion of members while a multicast group is in use. This distinction is not unique to any single storage application. Transactional replication is a common element in cluster protocol design.

The limited duration of a transactional multicast group implies that there is no need for the multicast forwarding element to rebuild its forwarding tables after it restarts. Any transaction in progress will have failed, and been retried by the higher-layer protocol.

Merely limiting the rate at which it fails and restarts is all that is required of each forwarding element.

Another implication is that there is no need for the forwarding elements to rebuild the membership list of a transactional multicast group after the forwarding element has been reset. The transactions using the forwarding element will all fail, and be retried by a higher layer transport or application protocol. Assuming that

forwarding elements do not reset multiple times a minute this will have very limited impact on overall application throughput.

The duration of a transaction is application specific, but inherently limited. A failed transaction will be retried at the application layer, so obviously it has a duration measured in seconds at the longest.

5.2.4. Who are the members of the group?

Join-directed multicasting allows any number of recipients to join or leave a group at will.

Transactional multicast requires that the group be identified as a small subset of a pre-existing multicast group.

Building forwarding rules that are a subset of forwarding rules for an existing multicast group can be done substantially faster than creating forwarding rules to arbitrary and potentially previously unknown destinations.

Some applications, including Object Clusters, benefit considering the members to be higher layer entities (such as virtual drives) rather than simply being the base IP address of the servers that host the higher layer entities. Doing so allows groups to be defined for each set of logical endpoints, not merely sets of physical endpoints. An Object Cluster, for example, could have two different groups ([A,B,C] vs [A,B,D]) even when the destinations are the same Layer 2 MAC address (i.e., C and D are hosted by the same server). This allows the server hosting both C and D to distinguish which entity is addressed using the Destination IP Address.

5.2.5. How much latency does the application tolerate?

While no application likes latency, multicast streaming is very tolerant of setup latency. If the end application is viewing or listening to media, how many msecs are required to subscribe to the group will not have a measurable impact to the end user.

For transactions in a cluster, however, every msec is delaying forward progress. The time it takes to do an IGMP join would be a significant addition to the latency of storing an object in an object cluster using a relatively fast storage technology (such as SSD, Flash or Memristor).

5.2.6. What must be done to maintain the Group?

The Join-directed multicast protocols specify methods for the required maintenance of multicast groups.mMulticast forwarders, switches or mrouters, must deal with new routes and new locations for endpoints.

The reference multicast group will still be maintained by the existing Join-directed multicast group protocols. The existing IGMP/MLD snooping procedures will keep the L2 multicasting forwarding rules updated as changes in the network topology are detected. Nothing in this specification changes the handling of the reference multicast group.

Transactional subset multicast groups are defined to be used only for short transactions, allowing them to piggy-back on the maintenance of the reference multicast group.

6. Forwarding Control Agent

The Forwarding Control Agent is responsible for translating forwarding control messages as defined in Section 7 into Layer 2 multicast forwarding for one or more subnets associated with a single physical layer 2 subnet.

Each Forwarding Control Agent can be though of as extending the IGMP/MLD snooping capabilities of an L2 forwarding element. It is translating the forwarding control agent messages into configuration of L2 multicast forwarding just as an IGMP/MLD snooper translates IGMP/MLD messages into configuration of Layer 2 multicast forwarding. This MAY be done external to the existing implementation, or it may be integrated with the IGMP/MLD snooper implementation.

Each Forwarding Control Agent:

- MUST Accept authenticated forwarding control agent messages controlling the creation and membership of Transactional Subset Multicast Groups within the context of a specified VLAN.
- MUST support at least one VLAN.
- MAY support multiple VLANs.
- MUST update the controlled Layer 2 forwarding element's multicast forwarding rules to reflect the subset specified for the group.
- MUST Update the controlled L2 forwarding elements multicast forwarding rules to reflect changes in the mapping of IP addresses

to L2 MAC addresses between transactions for persistent transactional suset multicast groups when informed of a prior transactional failure with a Refresh Membership message (see Figure 7).

- MAY refresh the Layer 2 multicast forwarding rules at any time.

6.1. Network Topology

Forwarding Control Agents are applicable for networks which consist of one or more local subnets which have direct links with each other.

6.2. Isolated VLANs Strategy

Transactional Subset Multicast groups define a very large number of multicast addresses which must be delivered within a closed set of IP subnets without having to dynamically co-ordinate allocation of these multicast addresses with a wider network.

This MAY be accomplished using a "Isolated VLANs Strategy" where the reference multicast group and all transactional multicast groups derived from it are used strictly inside of a single VLAN or a set of interconnected VLANs which route these multicast groups solely within this closed set.

Specifically, an implementation using the Isolated VLANs Strategy:

- MUST include only a pre-defined set of subnets,each enforced with a VLAN.
- MUST provide for routing or forwarding of all packets using the reference multicast group and all transactional subset multicast groups derived from it amongst these subnets.
- MUST NOT allow any packet using the reference multicast group or any transactional subset multicast groups derived from it to be routed to any subnet that is not part of the identified Isolated VLAN set.
- MAY/SHOULD guard the confidentiality of multicast packets routed between subnets that transit subnets that are not part of the Isolated VLAN set.

Applications MAY use the Isolated VLAN Strategy. Virtually all applications will elect to do so because allocating a very large block of adjacent multicast addresses would be very difficult. Confining usage of these addresses to a single VLAN is highly desirable.

Direct connections between the VLANs hosting Forwarding Control Agents is required because the Transactional Subset Multicast Groups are not known to any intermediate multicast routers that would implement indirect links. Co-locating Forwarding Control Agents with RBridges [[RFC6325]] MAY be a solution.

7. Forwarding Control Agent Methods

7.1. Dynamically Pushed Subset Groups

Each Pushed Subset Membership commands MUST contain the following:

- Subset Transactional Multicast Group: Group multicast address that is to have its multicast forwarding rules updated. This address must be within a block of Transactional Multicast Groups previously created using the Create Transactional Multicast Address Block command (Section 10.1).

- Target List: List of IP Addresses which are to be the targets of this group. These addresses are intended to be members of the reference group. When formulating the list, non-members MUST NOT be included. However there is no transaction lock placed upon the group, and therefor there may be changes in the group membership before the message is received. Therefore the Forwarding Control Agent MUST ignored any listed target that is not a member of the reference group.

This sets the multicast forwarding rules for pre-existing multicast forwarding address X to be the subset of the forwarding rules for existing group Y required to reach a specified member list.

This is done by communicating the same instruction (above) to each multicast forwarding network element. This can be done by unicast addressing with each of them, or by multicasting the instructions.

Each multicast forwarder will modify its multicast forwarding port set to be the union of the unicast forwarding it has for the listed members, but result must be a subset of the forwarding ports for the parent group.

For example, consider an instruction is to modify a transaction multicast group I which is a subset of multicast group J to reach addresses A,B and C.

Addresses A and B are attached directly to multicast forwarder X, while C is attached to multicast forwarder Y.

On forwarder X the forwarding rule for new group I contains:

- The forwarding port for A.

- The forwarding port for B. The forwarding port to forwarder Y (a hub link). This eventually leads to C.

While on forwarder Y the forwarding rule for the new group I will contain:

The forwarding port for forwarder X (a hub link). This eventually leads to A and B.

The forwarding port for C.

This assumes that the Forwarding Control Agent can perform a two-step translation: first from IP Address to MAC Address, and then from MAC Address to forwarding port. For typical applications of Transactional Subset Multicasting, all of the referenced IP Addresses will have been involved in recent messaging, and therefore will frequently already be cached.

Many ethernet switches already support command line and/or SNMP methods of setting these multicast forwarding rules, but it is challenging for an application to reliably apply the same changes using multiple vendor specific methods. Having a standardized method of pushing the membership of a multicast group from the sender would be desirable.

A Forwarding Control Agent MAY accept a request where the Target List is expressed as a list of destination L2 MAC addresses.

7.2. Persistent Transactional Subset Groups

There is a large group of pre-configured multicast groups which are an enumeration of the possible subsets of a master group. This will be a specific subset, such as all combinations of 3 members for multicast group X. These groups are enumerated and assuaged successive multicast addresses within a block.

The sender first obtains exclusive permission to utilize a portion of the reception capacity of each desire target, and then selects the multicast address that will reach that group.

In a straightforward enumeration of 3 members out of a group of 20, there are 20*19*18/3*2 or 1040 possible groups. Typically the higher layer protocol will have negotiated the right to send the transaction with the member prior to selecting the multicast group. In making the final selection, the actual multicast group is selected and some offered targets are declined.

Those 1040 possible groups can be enumerated in order (starting with M1, M2 and M3 and ending with M18, M19 and M20) and assigned multicast addresses from N to N+1039.

When the transaction requires reaching M4, M5 and M19, you simply select that group. Because exclusive rights to use multicasting to M4, M5 and M19 have already been obtained through the higher layer protocol the group [M4,M5,M19] is already exclusively claimed.

These 1040 groups may be set up through any of the following means:

- Traditional IGMP/MLD joining/leaving.
- Setting static forwarding rules using SNMP MIBs and/or switch-specific command line interfaces. Note that the wide-spread existence of command line interfaces to custom set multicast forwarding rules is an indicator that there are existing applications that find the exising IGMP/MLD protocols to be inadequate to fulfill their needs.
- The Dynamically Pushed Multicast Group method. See Section 7.1

8. Relationship to Existing Multicast Membership Protocols

TBD: briefly describe and cite IGMP, MLD and PIM.

Transactional Subset Multicast Groups are not a replacement for Join-based management of Multicast Groups. Rather it extends the group maintenance performed by the Join-based multicast control protocols from the reference group to any entire set of multicast addresses that are subsets of it.

This extension requires no modification to the existing data-plane multicast forwarding protocols or implementations. Transactional Subset Multicast groups may be implemented solely in the sender, receivers and the Forwarding Control Agents associated with each multicast forwarder supporting the reference group.

The maintenance work of the Join-based multicast protocols performed on the reference multicast group is leveraged to allow maintenance of a potentially large number of derived Transactional Multicast groups. This allows identification of a large number of subsets of the reference group, without requiring a matching increase in the maintenance traffic which would have been required had the derived groups been formed with a Join-based protocol.

9. Control Protocol

Note: the pre-standard protocol relies on multicasting of commands within a single secure VLAN. More general usage of these techniques will require transmitting Forwarding Control Agent instructions between subnets where they may be subject to interception and even alteration. Therefore a more secure method of delivering Forwarding Control Agent instructions is required.

The methods standardized by the KARP (Key Authentication for Router Protocols) are, in the Authors' opinion, fully applicable to this protocol. See [RFC6518]. Working Group feedback is sought as to how to expand this section, whether to split the Control Protocol to a separate document, or other methods of dealing with the control protocol.

The following requirements apply to any Control Protocol used:

- Each request MUST be uniquely identified. This identification MUST include the source IP address of the requester.
- The message MUST be authenticated.
- WG discussion is needed to reach a consensus as to whether the message contents need to be kept confidential, or whether preventing alteration is sufficient.
- The sender MUST NOT be required to transmit the command more than once other than as required for retries. For example, requiring SSH connections with each Forwarding Control Agent is not acceptable.
- Barring network errors, the message MUST be delivered to all Forwarding Control Agents that can receive the reference master group.

10. Forwarding Control Agent Methods

10.1. Create Transactional Multicast Address Block

TBD:This section will define the fields required for the command to create a block of transactional subset multicast addresses within a specific VLAN. The command defined here is delivered within a control protocol.

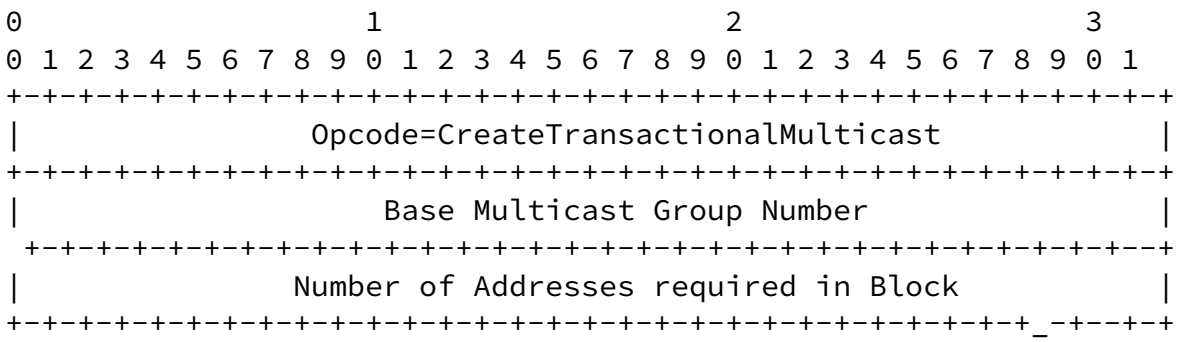


Figure 1: Create Transcaction Multicast Address Block Message

The Multicast Group Number is the 24-bit L2 Multicast MAC address. This matches both the IPV4 and IPV6 addresses which map to it. A given UDP datagram is sent using either an IPV4 or an IPV6 address, so the membership of a Multicast Group is either IPV4 endpoints or IPV6 endpoints at any given instant.

This command does not allow creating numerically scattered group of addresses. Doing so would have made the job of each Forwarding Control Agent more complex, and would be of no benefit in the recommended Isolated VLANs strategy (See Section 6.2).

note: add IANA language here

10.2. Release Transactional Multicast Address Block

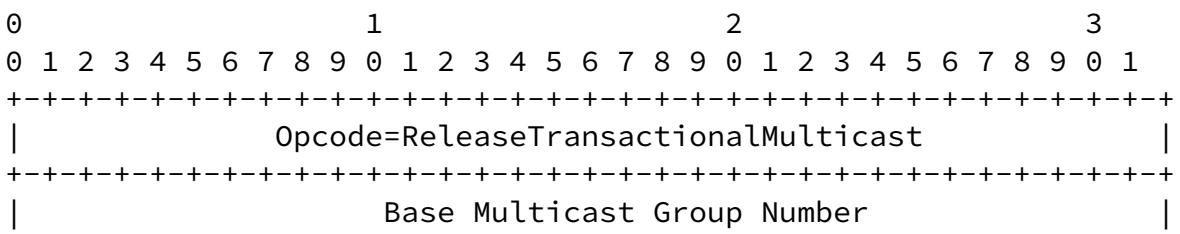


Figure 2: Release Transcactin Multicast Address Block Message

note: add IANA language here

10.3. Set Dynamic Transactional Multicast Group Membership IPV6

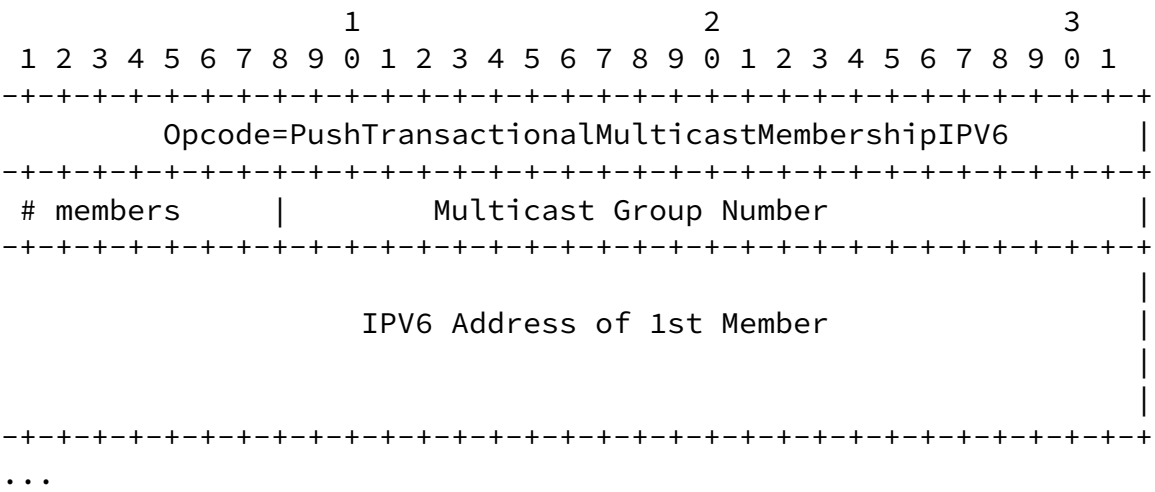
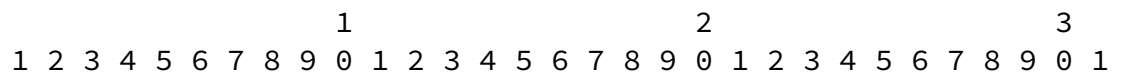


Figure 3: Set Dynamic Transactional Multicast Group Membership Message

Members: 8 bit unsigned number of IPV6 addresses that are to be the target of this specified Multicast Group Number.

note: add IANA language here

10.4. Set Dynamic Transactional Multicast Group Membership IPV4



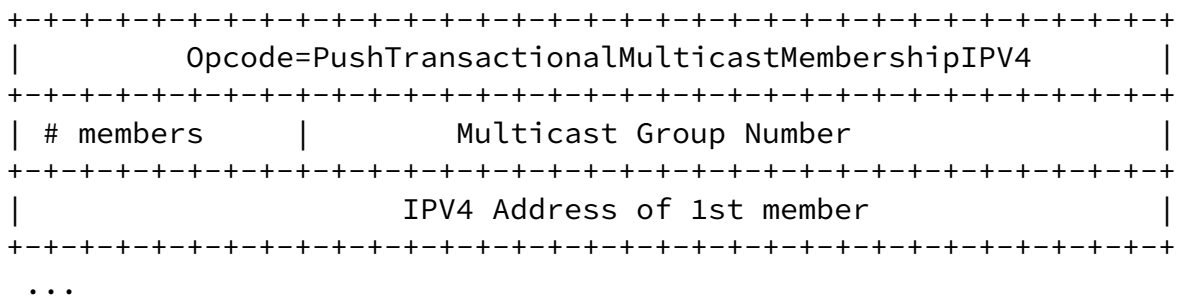


Figure 4: Set Dynamic Transactional Multicast Group Membership Message

Members: 8 bit unsigned number of IPV6 addresses that are to be the target of this specified Multicast Group Number.

note: add IANA language here

10.5. Set Persistent Transactional Multicast Groups IPv6

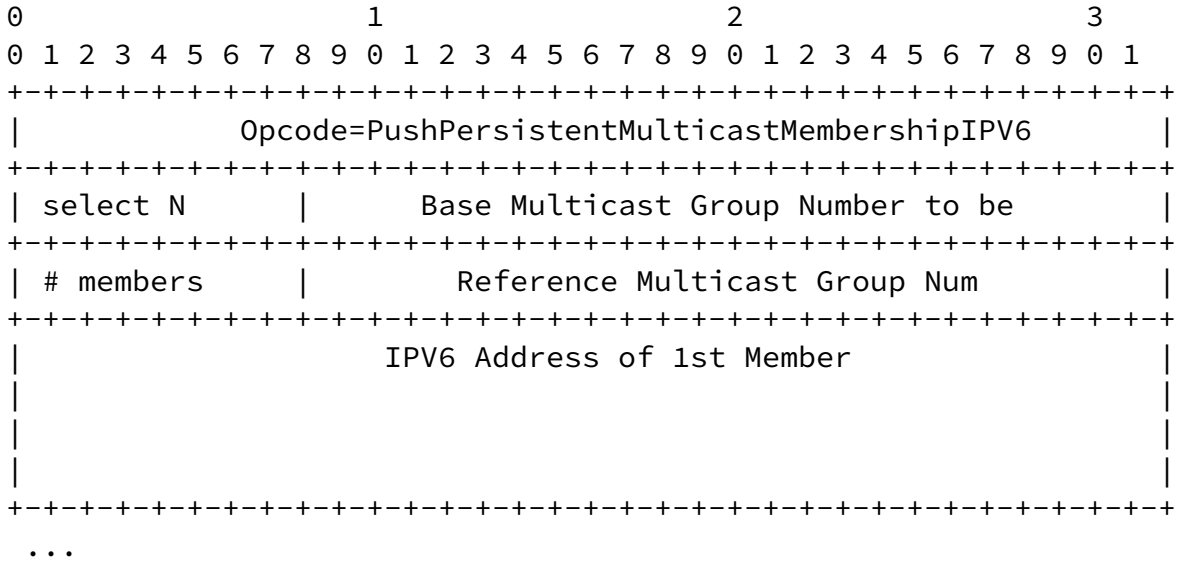


Figure 5: Set Persistent Transactional Multicast Groups Message IPV6

Members: 8 bit unsigned number of Members that are to be included in

each Transactional Subset Group set by this command.

Base Multicast Group Number to be set.

Members in the following list of IPV6 addresses. These must all be members of the Reference Multicast Group.

Reference Multicast Group Num: 24 bit L2 Multicast Group Number.

The motivation for supplying the list of IP addresses is to avoid race conditions where an IGMP or MLD join is in progress. If there were a method to refer to a specific generation of a multicast group membership then it would be possible to omit this list. Working Group suggestions are encouraged on this topic.

note: add IANA language here

10.6. Set Persistent Transactional Multicast Groups IPv4

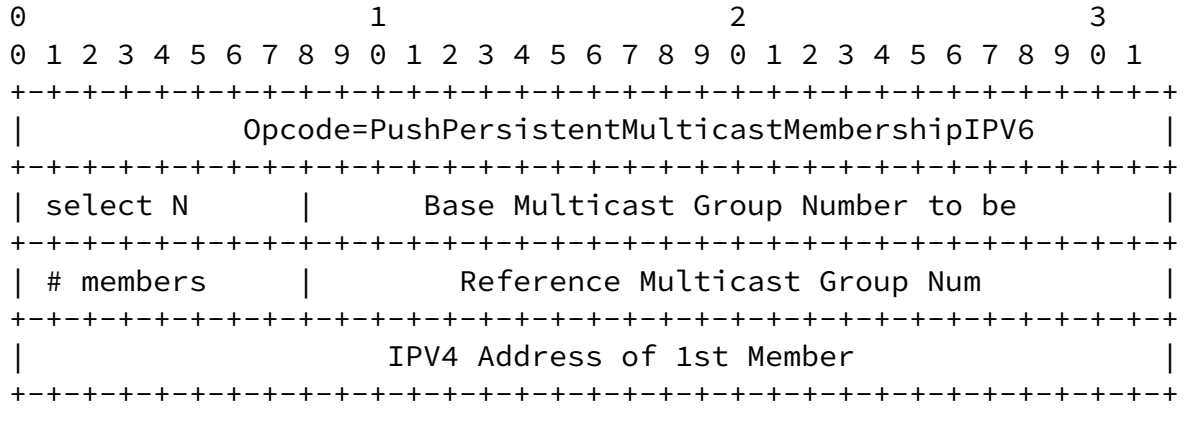


Figure 6: Set Persistent Transactional Multicast Groups Message IPv4

Members: 8 bit unsigned number of Members that are to be included in each Transactional Subset Group set by this command.

Base Multicast Group Number to be set.

Members in the following list of IPV6 addresses. These must all be members of the Reference Multicast Group.

Reference Multicast Group Num: 24 bit L2 Multicast Group Number.

note: add IANA language here

10.7. Refresh Persistent Transactional Multicast Group

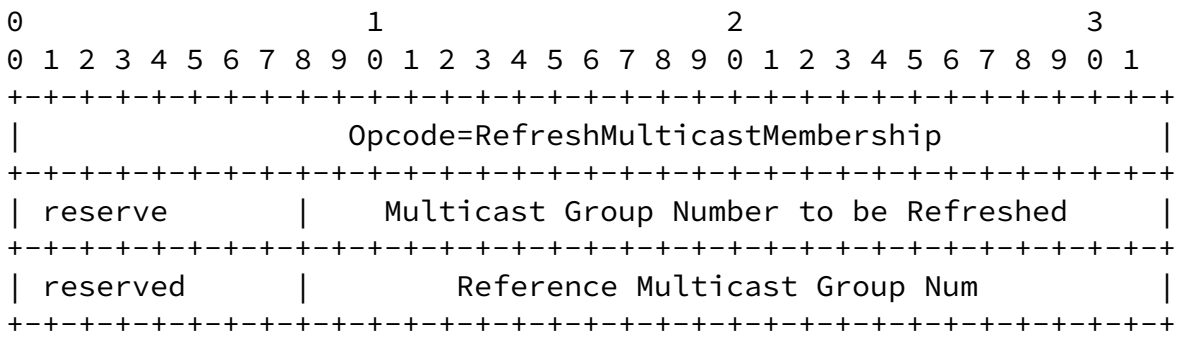


Figure 7: Refresh Persistent Transactional Multicast Groups Message

The existing Join-directed multicast group control protocols maintain delivery of a multicast group to the subscribers independent of network topology changes either at Layer 2 or layer 3. If a unicast IP datagram to a member would be delivered, then the multicast forwarding can be expected to also be current.

Transactional subset multicast groups do not require the same effort for maintenance. For a given transaction the entire set of datagrams is either delivered or it is not. There is no benefit to the application that the Forwarding Control Agent can achieve by promptly updating the L2 multicast forwarding tables after a network topology

change. The current transaction will miss at least one datagram, and therefore does not care if it misses multiple datagrams.

However, a Persistent Transactional Subset Mutlicast Group is used for a sequence of transactions targeting the same group. The upper layer protocol sender must have obtained exclusive rights to use the group for the period of time that it will be sending the transaction.

One method that it MAY use is to obtain the exclusive right to send the specific type of transaction to each of the members of the targeted group during negotiations conducted prior to use of the transactional group. For example, a reservation on inbound bandwidth may have been granted.

The Forwarding Control Agent MAY refresh its mapping from member IP addresses to L2 MAC address and then to L2 forwarding port at any time. However it MUST do so after receipt of a Refresh Transactional Subset Multicast Group for the group.

The sender of a transaction SHOULD send a Refresh Transactional Subset Multicast Group message after it fails to receive acknowledgement of an attempted transaction.

11. Security Considerations

The methods described here enable no sender to multicast messages to any destination that was not already addressable by it. Therefore no new security vulnerabilities are enabled by these techniques.

Because authentication of subset commands is kept lightweight there is an implicit trust within the application that transactional subset groups will be formed or selected in accordance with application layer expectations. The transport layer lacks sufficient information to enforce application layer expectations. If a malicious actor deliberately creates a transactional subset multicast group with an incorrect group it may adversely impact the operation of the specific upper layer application. However in no case can it be used to launch a denial of service attack on targets that have not already voluntarily joined the reference group

The protocol does not currently provide any mechanism to guard against selecting an existing but unrelated multicast group as a reference multicast group. Explicitly enabling use of an existing

multicast group to be a reference group would not solve the problem that the existing management of multicast groups is not aware of the need to explicitly forbid creation of derived multicast groups based upon a multicast group that it creates.

12. IANA Considerations

To be completed.

13. Summary

The proposal provides for two new methods to manage multicast group membership, Thee are simple techniques, but provide a cohesive cluster-wide approach to providing transactional multicasting. These techniques are better suited for transactional multicasting that the existing methods, IGMP and MLD, which are oriented to streaming use-cases.

14. References

14.1. Informative References

[Replicast] Bestler, C., "White Paper: Nexenta Replicast http://info.nexenta.com/rs/nexenta/images/Nexenta_Replicast_White_Paper.pdf", November 2013.

[MPI] MPI Forum, "Message Passing Inteface", 2012.

[AmazonS3] Amazon, "Amazon Simple Storage Service (S3) http://aws.amazon.com/s3/", 2014.

[Swift] Openstack, "OpenStack Object Service (Swift) http://docs.openstack.org/developer/swift/", 2014.

[IEEE.802.1Qau-2011] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 10: Congestion Notification", IEEE Std 802.1Qau, 2011.

[IEEE.802.1Qaz-2011] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 18: Enhanced Transmission Selection.", IEEE Std 802.1Qaz, 2011.

[IEEE.802.1Qbb-2011]
IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 17: Priority-based Flow Control.", IEEE Std 802.1Qbb, 2011.

[RFC5661] Shepler, S., Eisler, M., and D. Noveck, "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 5661, January 2010.

14.2. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.

[RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

[RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012.

Authors' Addresses

Caitlin Bestler (editor)
Nexenta Systems
455 El Camino Real
Santa Clara, CA
US

Email: caitlin.bestler@nexenta.com,cait@asomi.com

Robert Novak
Nexenta Systems
455 El Camino Real
Santa Clara, CA
US

Email: robert.novak@nexenta.com