IEEE 802.1D

Selective Multicasting

(Filtering Services for the Dynamic Use of Multicast Addresses)

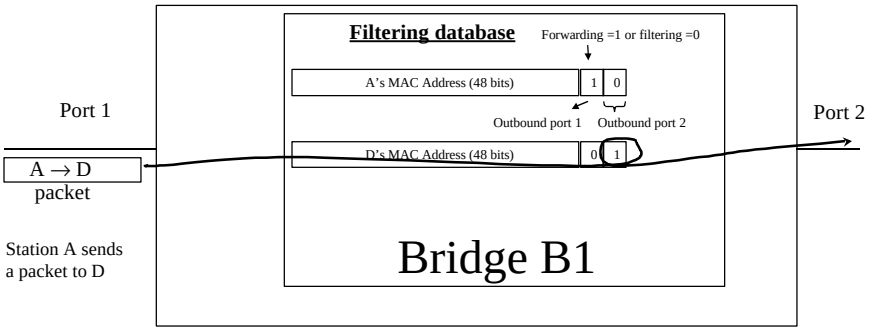
Why Selective Multicasting?

- Broadcasting of multicast traffic overloads slower links, and limits the total volume of multicast traffic
- Solution:

  Allow users to explicitly indicate their interest in receiving traffic on given multicast addresses

The Filtering Database (1)

A bridge maintains a *filtering database* which it uses to decide where to send the incoming packet.



The Filtering Database (2)

- The Filtering Database contains information in the form of filtering entries that are either:
  - Static Entries - explicitly configured by management
  - Dynamic Entries - automatically entered into the Filtering Database by the normal operation of the bridge
- Both static and dynamic entries consist of:
  - A MAC address specification
  - A Port Map which specifies the filtering state for the MAC address specification on each outbound Port

Basic Filtering Services

- A bridge must support the basic filtering services
- Basic filtering services allow the specification of the following Filtering Database entries
  - For individual MAC addresses:
    - Static entries which indicate for each outgoing port if frames destined for the specified individual MAC address should be filtered or forwarded
    - Dynamic entries which are created and updated by the learning process
  - For a specific group MAC address:
    - Static entries which indicate for each outgoing port if frames destined for the specified individual MAC address should be filtered or forwarded
    - If no static entry is present for a specific group MAC address, frames destined to the group MAC address are broadcast on all outgoing ports

Extended Filtering Services

- Extended filtering services add the following capabilities to the basic filtering services
  - For individual MAC addresses:
    - Static entries may contain a value which indicates the dynamic filtering information should be used for a specific port rather than always forward or always filter
  - For a specific group MAC address:
    - Static entries may contain a value which indicates the dynamic filtering information should be used for a specific port rather than always forward or always filter
    - Group Registration Entries - dynamic filtering entries which are created and maintained through the use of the GMRP protocol
  - Entries corresponding to All Group Addresses which do not have a specific group MAC address entry
  - Entries corresponding to All Unregistered Group Addresses which do not have a specific group MAC address entry

Default Group Filtering Behavior

- Each port may be assigned a default behavior for forwarding frames with group MAC addresses
- Three default behaviors have been defined
  - Forward All Groups: the frame is forwarded unless an explicit static filtering entry exists
  - Forward All Unregistered Groups: the frame is forwarded unless:
    - An explicit static filtering entry exists
    - An applicable Group Registration entry exists
  - Filter Unregistered Groups: the frame is filtered unless:
    - An explicit static filtering entry exists
    - An applicable Group Registration entry exists

GMRP
GARP Multicast Registration Protocol

- A mechanism that allows bridges and end stations:
  - to dynamically register (and subsequently de-register) group membership information with the MAC bridges attached to the same LAN segment
  - to disseminate that information across all bridges in the bridged LAN that support Extended Filtering Services
- Operation of GMRP relies on the services provided by the *Generic Attribute Registration Protocol* (GARP)

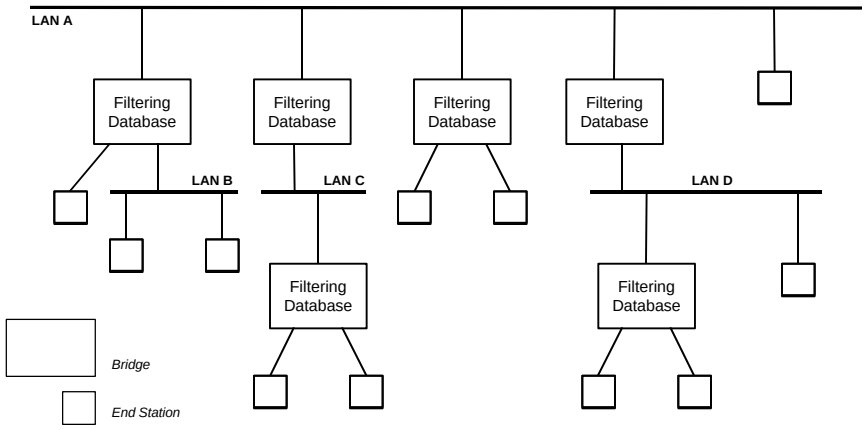
Result of Group Membership Information Registration and Propagation

- Frames sent to a particular group can be received on all LAN segments to which registered GMRP participants are attached
- Bridges filter frames on ports which have not had group registration entries created by GMRP
  - frames are not transmitted on LAN segments which neither have registered GMRP participants, nor are in the path through the *active topology* between the sources of the frames and the registered members

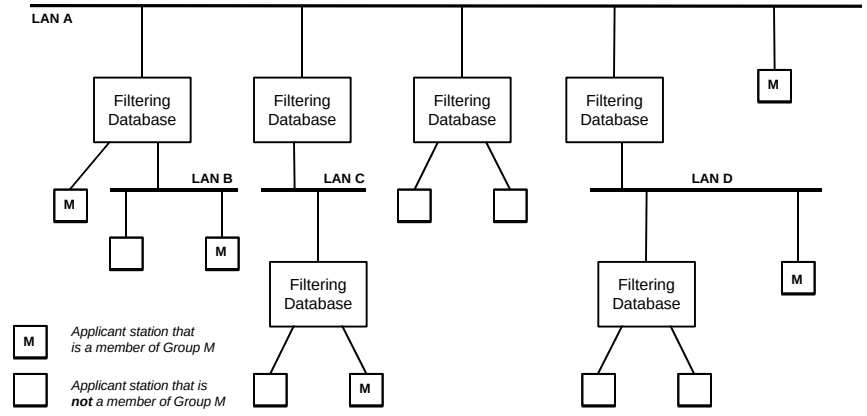
Open Host Group Concept

- Any GMRP participants that wish to receive frames transmitted to a particular group or groups register their intention to do so by requesting membership to the group(s) concerned
- Any MAC service user that wishes to send frames to a particular group can do so *from any point of attachment* in the bridged LAN
- MAC service users that are sources of MAC frames for the group do not have to register as members of the group themselves unless they also wish to receive frames sent to the group by other sources

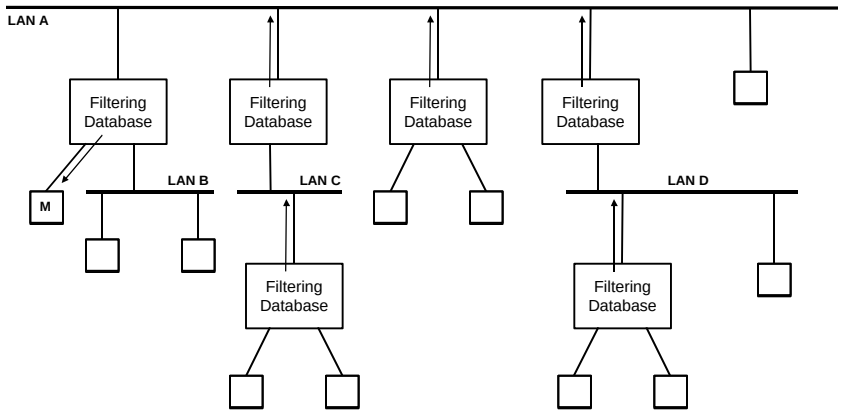
Example of an Active Topology



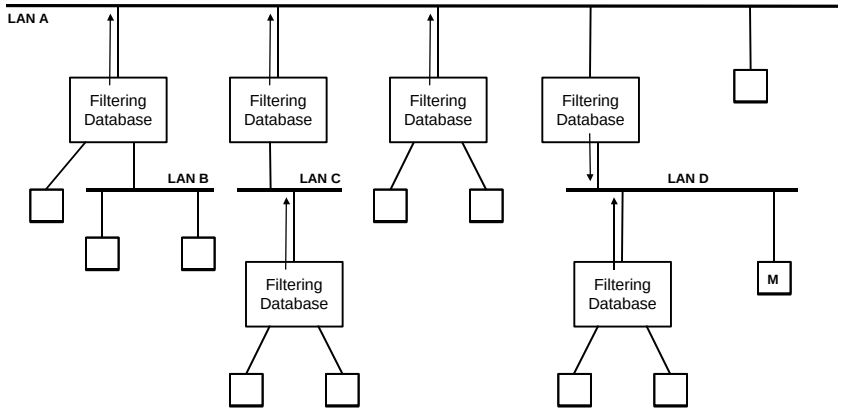
Active Topology with Group Members



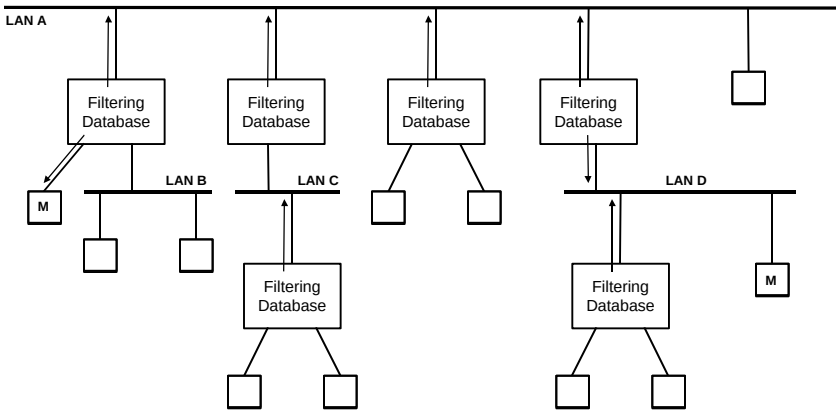
Example 1



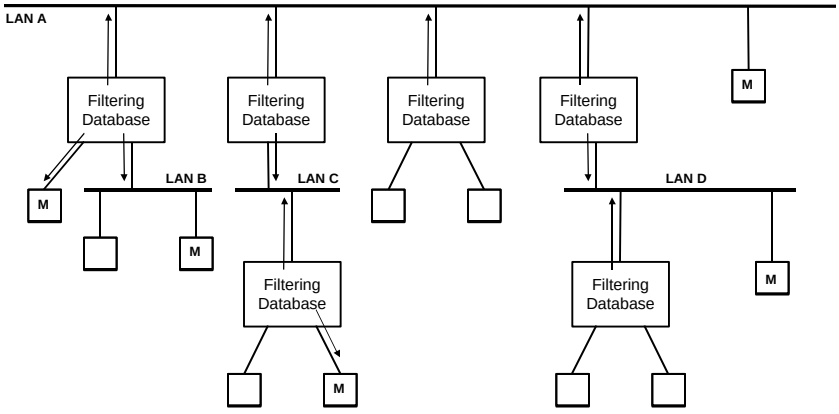
Example 2



Example 3



Group Registration Entries in Filtering Databases Resulting in a Directed Graph



Source Pruning

- End stations use Group Membership information registered via GMRP to keep track of the set of Groups for which active members exist
- End stations may suppress the transmission of frames for which there are no valid recipients
- Avoids unnecessary flooding of traffic on the LAN if there are no members that wish to receive such traffic

Default Group Filtering Behavior

- There are three different group services:
  - Forward All Groups
  - Forward Unregistered Groups
  - Filter Unregistered Groups

Use of Forward All Groups

- To ensure that regions of the Bridged LAN that contain legacy devices can receive all multicast frames
  - Can use static configuration to explicitly disallow certain multicast traffic
- To allow successful operation of devices that require promiscuous reception
  - Routers
  - Network monitors

Use of Forward Unregistered Groups

- Group addresses which do not have dynamic filtering database entries are forwarded
- Group addresses which have dynamic filtering entries are filtered or forwarded based on the dynamic entry
- Useful in circumstances where GMRP-aware devices distinguish between legacy multicast addresses for which they do not register and “new” multicast addresses for which they do register
- Must ensure GMRP-aware end stations do not register for legacy multicast addresses

Use of Filter Unregistered Groups

- Group addresses which do not have dynamic filtering database entries are filtered
- Group addresses which have dynamic filtering entries are filtered or forwarded based on the dynamic entry
- Intended for operation with GMRP-aware end stations only

Type of Information Registered by GMRP

a) group membership information
  - indicates that one or more GMRP participant that are members of a particular group (or groups) exist
  - carries the group MAC address(es) associated with the group(s)
  - results in the creation or updating of *group registration entries* in the filtering database to indicate the port(s) on which members of the group(s) have been registered

b) group service requirement information
  - indicates that one or more GMRP participants require *Forward All Groups* or *Forward Unregistered Groups* to be the default group filtering behavior (applied to a packet whose group address is not in the filtering database).

The Extended Filtering Database

- The Filtering Database contains information in the form of filtering entries that are either:
  - Static Entries - explicitly configured by management
  - Dynamic Entries - automatically entered into the Filtering Database by the normal operation of the bridge
  - Group Registration Entries - created modified and deleted by the operation of GMRP

GARP

- Objective:
  - registration and dissemination of information of any generic attribute over a bridged LAN
  - end stations can issue/revoke declarations for the attribute values
- Attributes are opaque to GARP
  - it is up to the GARP Application to define and interpret the generic attribute

Control Messages used in GARP

Two principal control messages

- *Join Message*:
  - Sent by a user to register an attribute
- *Leave Message*
  - Sent by a user to de-register an attribute

GARP: Design Principles (1)

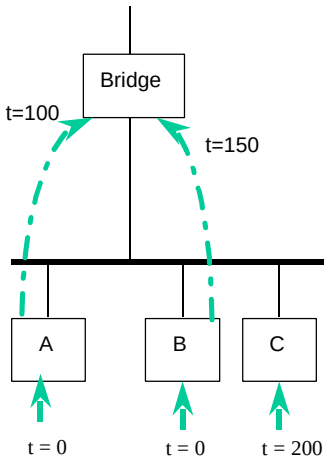
- Fully-distributed protocol
- Simple (no explicit information about members)
- Resilient against loss of a single control packet
  - A participant that wishes to make a declaration sends two Join messages
- Scalable
  - Transmission of GARP control messages by participants is randomized over time
  - An applicant that sees a Join message for the same attribute it intends to register considers it as if it were one of its own

GARP: Design Principles (2)

- Soft state
  - no acks, no confirmation, no information about registered users
  - registrations have to be refreshed continuously:

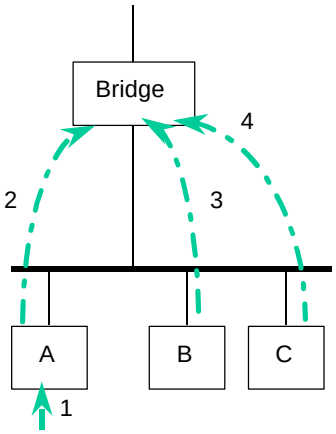
    Every 10 s. the bridge issues a *Leave All* message "threatening" to deregister all groups. Users still interested in keeping the registration alive send Join messages.
- Resilient to failure of GARP participants
  - This is in case the GARP participants fail to see some GARP messages such as Join...
- Operates in homogeneous and heterogeneous bridged LANs

A Registration Example



• Stations A and B decide to join group G at the same time (t = 0). Both choose a timer uniformly between 0 and *JoinTime* (200 ms). If the timer happens to be set to 100 ms for A and 150 for B.
• ***At t = 100 ms***, A sends a Join message and reset the timer for the second Join message. Let's say it is set to expire at t = 225 ms.
• ***At t = 150 ms***, B has seen the first message sent by A, so B sends a Join message (it knows that another Join has been sent so it doesn't try to send another message).
• A will not send another Join message since it consider the one sent by B to be one of his. Any other station that decides to join group G, does not send any Join message.

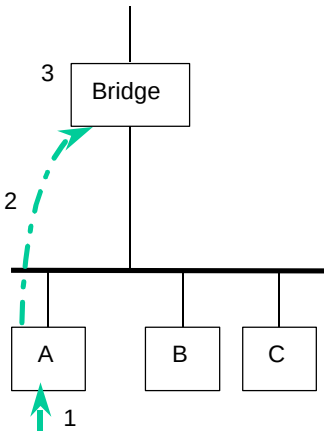
A De-registration Example (1)



1. The user at station A decides to leave group G, however B and C are still interested in receiving group G
2. After a *JoinTime*, A sends a Leave(G)
3. After another *JoinTime*, B sends a Join(G)
4. After another *JoinTime*, C sends a second Join(G). Note that the second Join can be sent by another participant

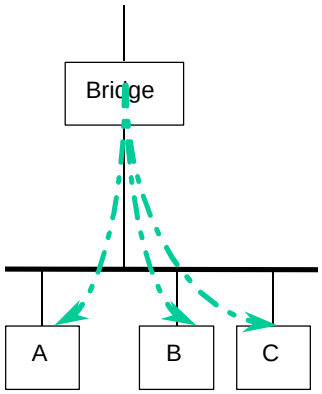
Note that the bridge has never stopped forwarding frames destined to group G

A De-registration Example (2)



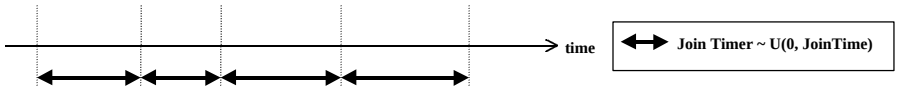
1. The user at station A decides to leave group G. B and C left the group before
2. After a *JoinTime*, A sends a Leave(G)
3. After a *LeaveTime* (this time is to let stations reregister in case they would like to) without receiving a Join the bridge deregisters group G from this port and stops relaying frames for group G

A Leave All Example



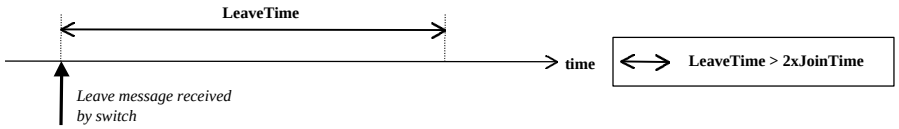
- The bridge is sending a LeaveAll message after a period of inactivity greater than the LeaveAll period.
- If stations are alive, they should join all the groups they are interested in.
- After the expiration of the Leave Timer, if no registration has been done the group will be left and the packet from this multicast address will not be forwarded anymore.

Timing Parameter: Join Timer



- GARP generates its control messages at random intervals uniformly chosen between 0 and a timing parameter *JoinTime*:
  - Allows aggregation of internal control messages
  - Prevents message implosion in response to a Leave or LeaveAll
  - *Recommended value: 200 ms*

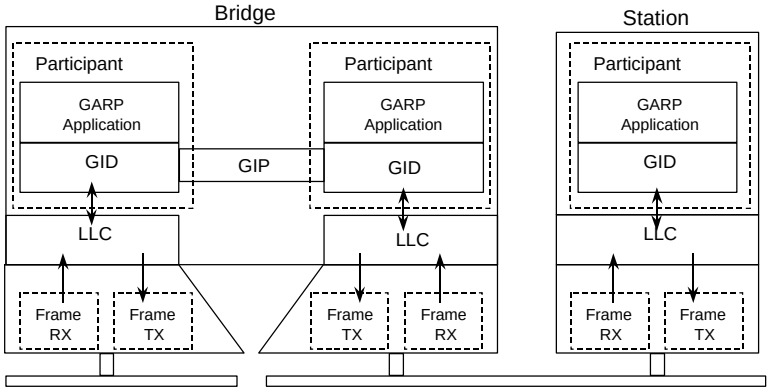
Timing Parameter: Leave Timer



- Set when a Leave is received or a LeaveAll is issued
- Delays the de-registration of an attribute so that other participants have a chance to rejoin.
- May not be arbitrarily large as it may result in excess traffic getting forwarded
- *Recommended LeaveTime value : 600 ms*

GARP Architecture

H.O. #5
Winter 2000



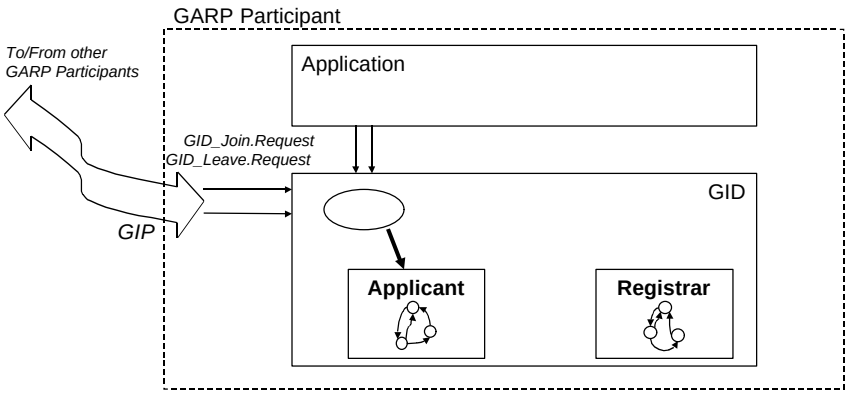
GARP Participant (1)

- A GARP "Participant" consists of:
  - A GARP Application component
  - A GARP Information Distribution (GID) component
- One participant exists per port, per application

GARP Participant (2)

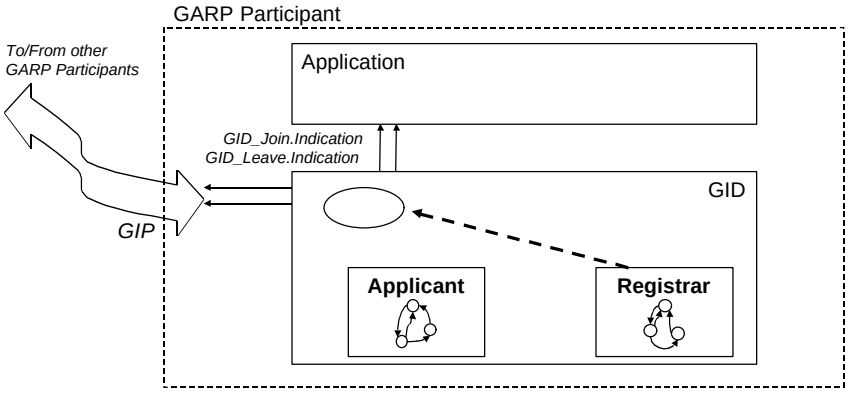
- The GARP Application:
  - Defines a set of one or more *Attribute Types*, their set of values and their semantics
  - Specifies the structure and encoding of Attribute Types and values in the GARP PDU
  - For each application, a unique *Group MAC Address* is specified and used as destination MAC address for all GARP protocol exchanges pertaining to the application
- The GID:
  - Distributes Attribute information among participants on the same segment
  - Contains a *Registrar* and an *Applicant* state machine *for each attribute*

GARP Participant (3)



Note: There is an instance of the Applicant and the Registrar per Attribute

GARP Participant (4)



Note: There is an instance of the Applicant and the Registrar per Attribute

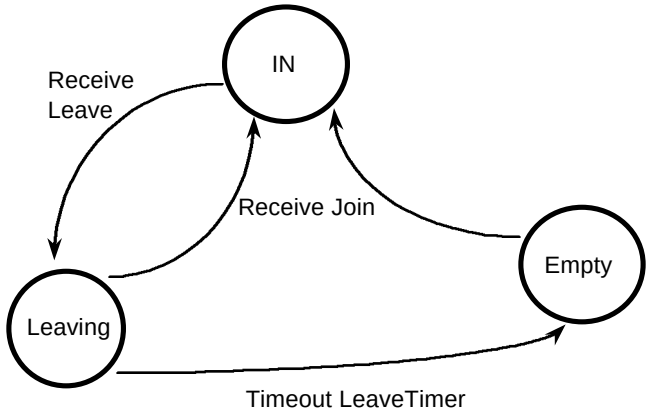
Protocol Elements in a Participant

- Registrar
  - records attribute registration declared by other participants on the same segment
  - required in bridges
  - may be implemented in stations for *source pruning*
- Applicant
  - ensures that this Participant's declarations are registered by other Participants

Registrar State Machine (1)

- IN:
  - Someone registered this attribute value on this segment
- Empty (MT):
  - All declarations for this attribute value on this segment have been withdrawn
- Leaving (LV):
  - Timing out the registration; if no (re)declaration is received before the LeaveTimer expires, the transition to the Empty state will occur

Registrar State Machine (2)



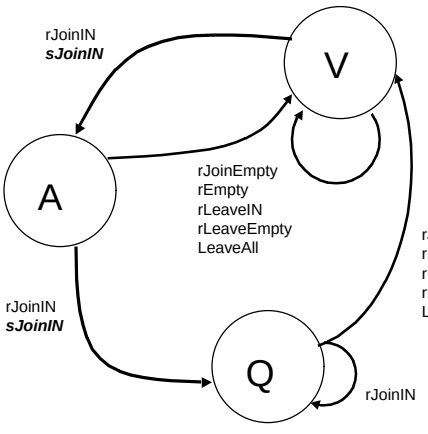
Applicant State Machines

- Three levels of complexity defined:
  - For participants that do not include a registrar (stations)
    - Simple Applicant
    - Applicant Only
  - For participants that implement a registrar (bridges and some source stations)
    - Full Applicant

Simple Applicant

- Simplest type of applicant: no registrar and only three states
- Very Anxious (V):
  - Applicant has no reason to believe that the other Registrars have registered this attribute value (No JoinIn message has been received and no Join message has been sent).
- Anxious (A):
  - If no messages have been lost, other Registrars will have registered this attribute value
    - A second Join message has still to be sent/seen
- Quiet (Q):
  - No need to send more messages

Simple Applicant State Machine



Messages are labeled by the registrar state of the sender.
The JoinIN sent when a Join timer expires is shown as ***sJoinIN***.
Other transitions happen in response to received messages (prefixed with r).
Note that even though the Participant doesn't have a Registrar, it sends *JoinIN* and *LeaveEmpty* by default.

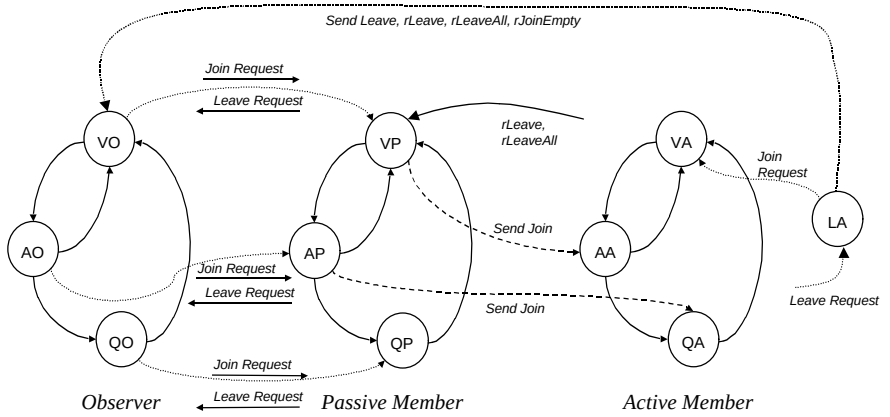
Applicant Only (1)

- In order to save on Join and Leave messages when there are several Members on the same segment, stations may track the GARP message activity on the segment for a given Attribute, even when they are not currently interested in registering the Attribute.
- For this purpose, a distinction is made within an applicant between being a Member or an Observer:
  - An Observer tracks the Attribute state, but does not wish to make a declaration.
  - A Member is a Participant that is interested in maintaining a declaration for a given attribute value. There are two types of Members:
    - Passive member (P): attempts to maintain the registration of an attribute value, but hasn't itself sent a Join message to declare it
    - Active member (A): attempts to maintain the registration of an attribute value and has sent a Join message to declare it.
  - Note that the observer is always "passive"

Applicant Only (2)

- An Observer can become a Member without sending a Join message (if it is in the *Quiet* state)
- A Member that is Passive can quit without sending a Leave message
- State Machine Acronyms
  - States:
    - VO: very anxious, observer
    - AO: anxious, observer
    - QO: quiet, observer
    - VP: very anxious, passive...
    - VA: very anxious, active…
    - LA: Leaving, Active (a substate where a Leave message is generated)

Applicant Only (3)



Applicant Only (4)

- When a Participant becomes interested in a given Attribute
  - The Applicant transitions first from *Observer* to *Passive Member*.
  - If the Observer was in the *Quiet* state, the Applicant becomes *Passive and Quiet*.
  - Otherwise, it schedules a Join for the earliest transmission occasion
    - A *Passive Member* that sends a Join becomes an *Active Member*.

Applicant (1)

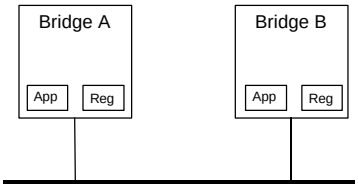
- This is the Full Applicant state machine.
  - It is used in bridges, where a Registrar is also present.
  - It is also used in stations which include a Registrar for the purpose of source pruning.
- Differences with Applicant Only state machine:
  - When sent, a Join message is labeled with the current state of the Registrar
    - When Registrar is in IN state, send *JoinIN*
    - When Registrar is not in IN state, send *JoinEmpty*
    - A *JoinEmpty* message causes the Applicants on a segment to go to the *Very Anxious* state. Its goal is to force other Members to update the sender's state by not considering this Join message valid for that purpose.

Applicant (2)

  - An Observer has an additional state (Leaving Observer, LO state) similar to the *Leaving Active (LA)* state for an Active Member, which it enters upon hearing a *Leave* in which it generates an *Empty* message, unless it hears a Join.
    - Like the *JoinEmpty* message, the *Empty* message forces the Applicants on a segment to go to the *Very Anxious* state.
    - The purpose of this is to protect against problems incurred when other participants fail to see a Leave.

Applicant (3)

- Suppose Join messages are not labeled with registrar state:
  - If A sends 2 Join messages that B hears
    - The registrar states will be MT for Bridge A and IN for Bridge B, i.e. A's registrar is in an incorrect state.
  - The problem is due to the fact that A's Join messages have no indication about its knowledge of B's wish to register the Attribute.
- Therefore, by labeling the Joins with its Registrar state, A will force B to send Joins for AT. This corrects the problem above.



Scenario

Bridges A and B are both interested in an Attribute AT, and both have their registrar in the MT state

GIP

- GIP: GARP Information Propagation
  - Propagates Attribute registration information between participants for the same application on all ports which are in the *forwarding* state in a bridge
    - A registration is propagated on a port if **any** other port in the bridge has seen a registration for the Attribute concerned.
    - A de-registration is only propagated to a port if **all** other ports are de-registered for the Attribute concerned.