Subscription services:
Simple Reservation Protocol (SRP),
A Residential Ethernet SG presentation

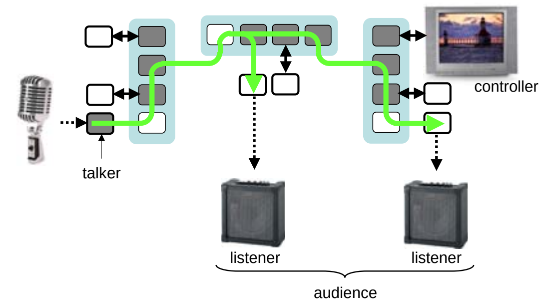
David V James JGG
Alexei Beliaev Gibson

For details, see available white papers:
dvjReCompact2005May14 – summary clauses, for quick read
dvjReComplete2005May14 – all clauses, for in-depth thought
Available at:
http://www.ieee802.org/3/re_study/material/index.htmp
dvjames.com/esync (a DVJ site, until IEEE site is updated)

To ensure bandwidth through restrictive links, subscription protocols are needed.

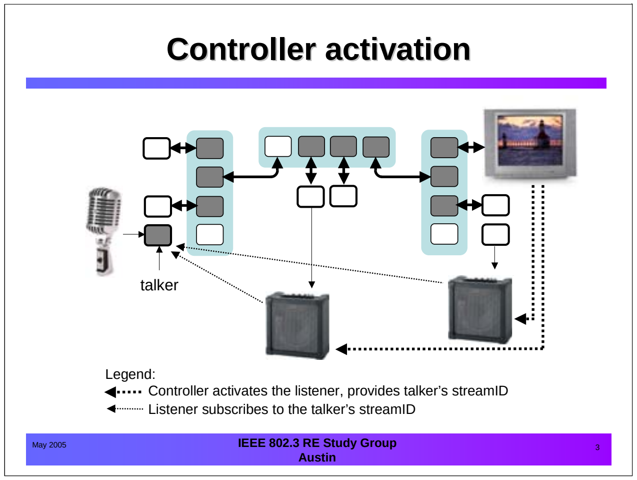
To minimize design time:
- an existing L3 protocol (RSVP) is leveraged.
- options and fancy features are discarded.

Subscription components



Who is involved in subscription?
The controller, that identifies the need and discovers talker/listener
The listener, that (when triggered) activates the path.
The talker, that provides the streaming classA frames.

When more than one listener is involved, there is an “audience”.

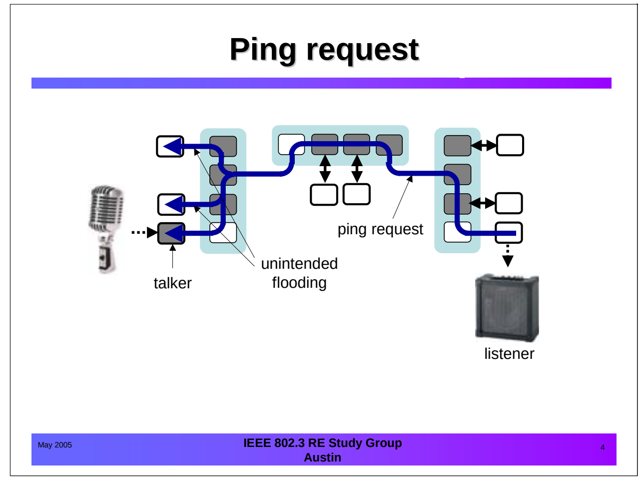


Simple listener devices may have minimal intelligence and/or human-interfaces.

Thus, a controller (that has these) is used to command a listener.

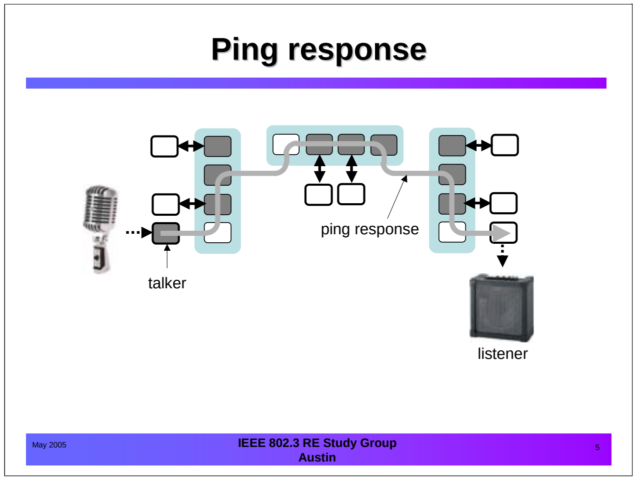
The controller could be physically located in the talker or listener.

However controller-to-listener and controller-to-talker communications are:
- beyond the scope of this presentation!
- beyond the scope of an 802 standard (most likely).



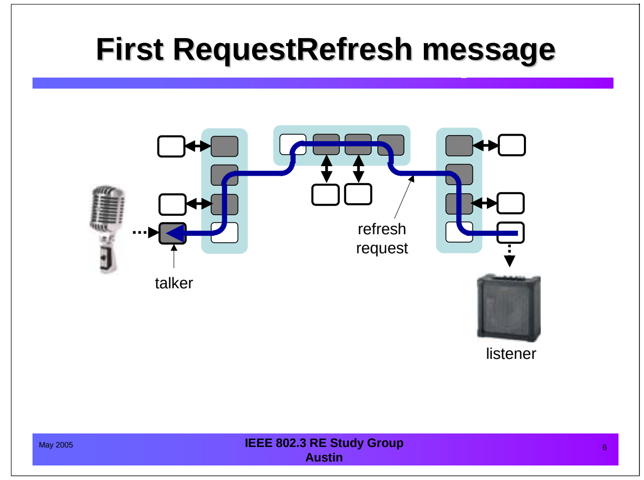
Subscription messages allocate resources along their path,
so false flooded paths are undesirable.

SO, a ping is used to initialize routing before subscription starts.



When the ping response returns, the talker-route is known to in-path bridges.

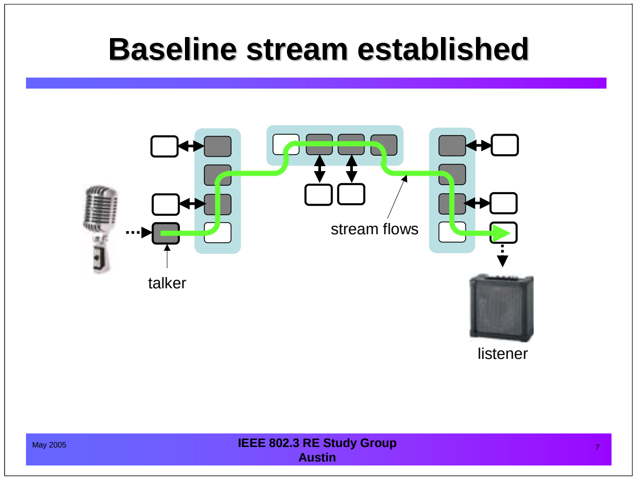
No special ping definitions are thought to be needed; existing definitions are OK.



We assume the conversation is established from the listeners, not the talker.

This allows other listeners to easily join into the conversation (shown later).

The RequestRefresh frames propagate from the listener to the talker, allocating bandwidth on a hop-by-hop basis.

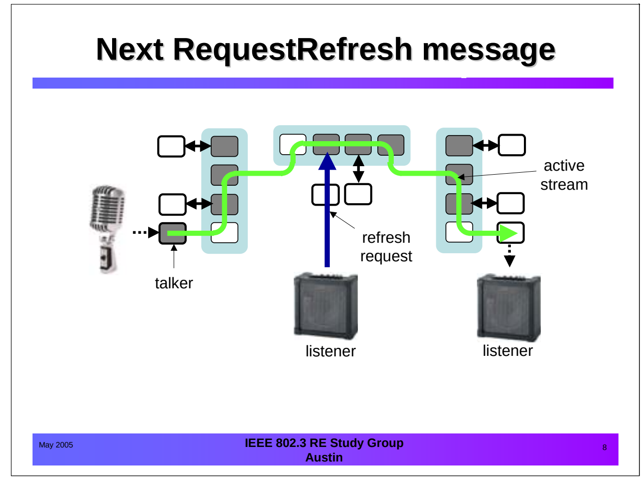


Unlike RSVP, there is no explicit response frame that returns.

Instead, the presence of a classA stream provides an implied acknowledge.

An explicit NACK (in the form of a ResponseError message) is defined, to handle cases of insufficient bandwidth, unlearned addresses, etc.

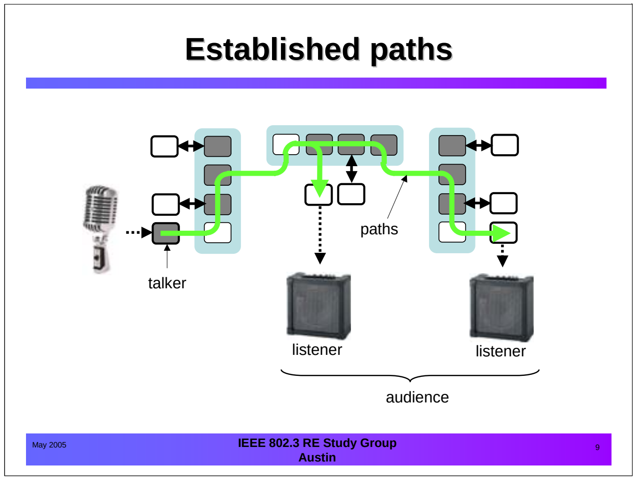
The ResponseError messages and conditions are discussed later.



Another request can come from a distinct listener.

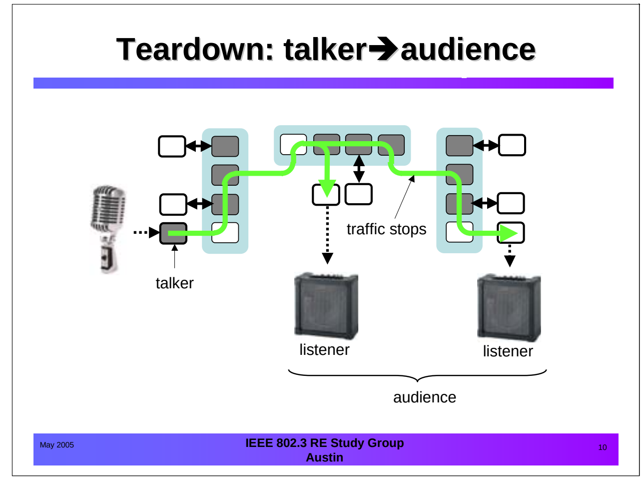
In this illustration, we assume the listener is another speaker, listening to a shared stereo stream.

The request “joins” the previously established conversation and never reaches the talker (only the bridge’s active-stream context needs to changes).



The success ‘ACK’ is the presence of the desired classA stream.

An extended lack of NACK or ACK conditions can result from a frame loss.
This error condition is handled be refresh retransmissions, discussed later.



What happens if the talker disappears or the stream can’t continue?

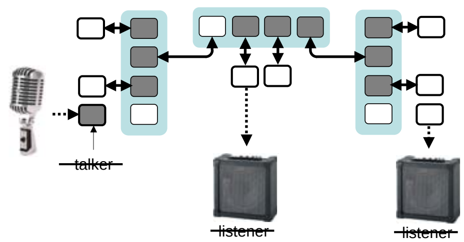
An implicit teardown message is multicast to the audience.

The multicast requires no knowledge of the listeners at the talker.

Routing of the teardown follows the established multicast path(s).

Rather than defining additional messages, the implicit teardown if the absence of classA traffic flow, when continued for an extended (TBD) interval.

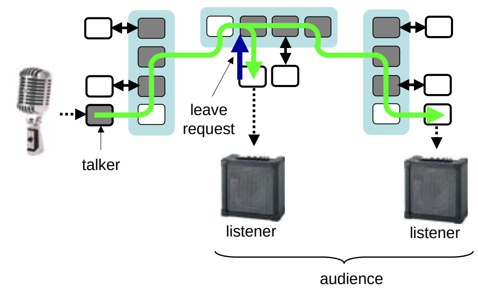
Teardown completed



After teardown completes, there are no talkers, listeners.

Nor are there any committed resources within the bridges.

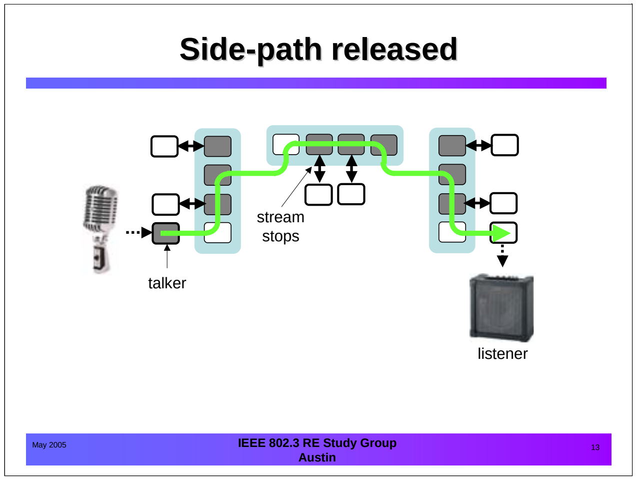
Release: listener➔talker



A release can also be initiated by one or more listeners,
by generating a RequestLeave message.

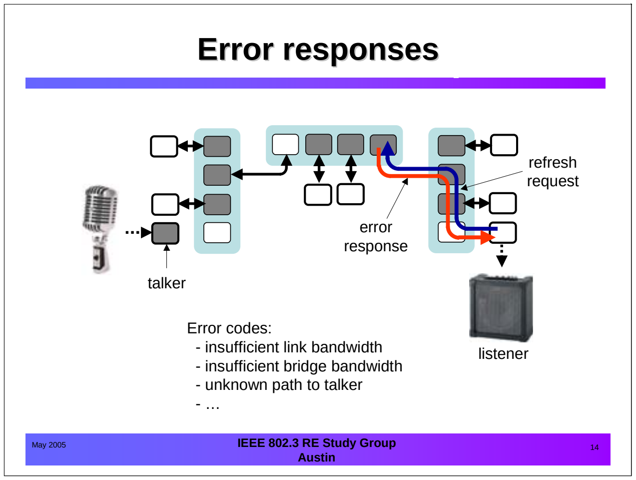
Each listener is handled independently.

If one or more listeners remain, then streaming classA transmissions continue.



In this example, only one of the listeners has left and the other listener remains connected.

Note that the second listener and the talker are both unaffected by the released side-path connection; their operation is unaffected.

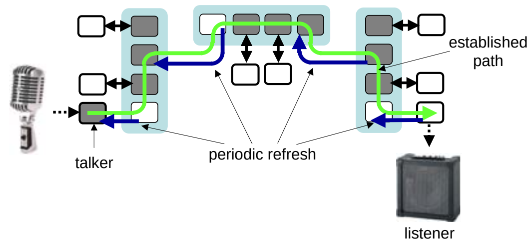


What happens if a subscription-request error is detected?

An error message is returned.

Error messages are not confirmed, since periodic refresh (discussed later) handles the case of lost error messages.

Heartbeats: periodic listener refresh



What happens if the STP changes the route?

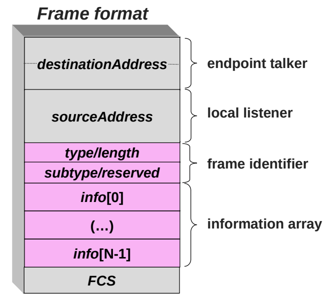
What happens if a RequestRefresh is lost?

What happens if a listener disappears?

All cases are handled using one uniform mechanism:
- The listener sends periodic refresh messages.
- A talker timeout detects the absence of these expected messages.

NOTE: Checks ensure the path is released if not needed:
- A loss of the talker releases all paths
- The loss of a redundant listener releases its side path
- The loss of the only-remaining listener releases the entire path

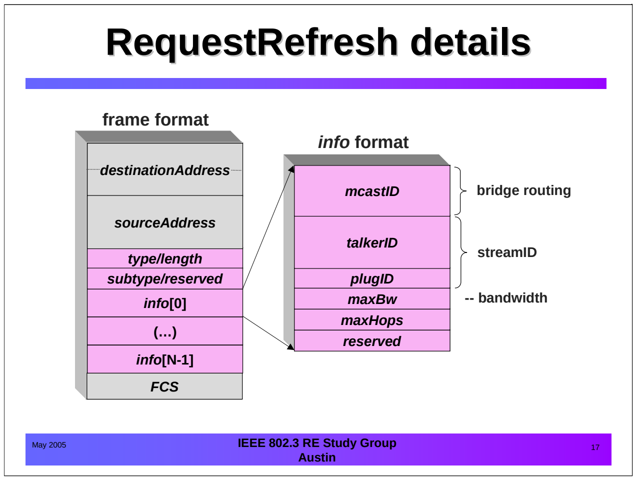
RequestRefresh format



What is contained within each frame?

Frames are sent from the listener to the endpoint talker:
- The path to the talker need not be know to the listener.
- Distinct coding allows the frame to be “hijacked” at bridge ports.

To reduce the overhead of periodic RequestMessage transmissions, multiple information blocks are included within each one (as in RSVP).



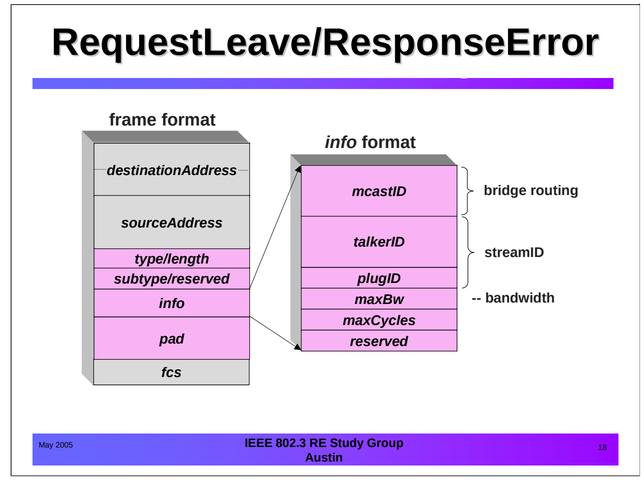
What is contained within each info block?

The routing information associated with each classA frame.
(In this example, we have assumed this is a multicast address.

The talkerID and plugID are also needed, since they are the “true” stream identifier parameters.

The maxBw (maximum bandwidth) requirements are specified.

Other information may be provided, as discussions progress…



The RequestLeave and ResponseError frame formats are very similar, but only one info block is provided.

There is no need for multiple blocks, since these are rarely sent.

NOTE: This format was based on the RSVP writeup, which encapsulated info blocks in the RequestRefresh, apparently for efficiency.

Alternatives for this L2 protocol could be:

- All info blocks are encapsulated as 1-per-frames
- All info blocks are tagged and encapsulated as N-per-frame (subject to MTU limitations)
- Like the RequestRefresh message, ResponseError and RequestLeave may have multiple info blocks.

The authors have no strong leanings, one way or the other,
so comments/recommendations are appreciated.