



Simple Connection Management Protocol

Ashley Butterworth
Apple Inc.

Intro

- SCMP is a transaction based protocol which implements one thing - connecting talker stream sources with listener stream sinks
- Uses multicast 1722 control packets
 - Doesn't require device MAC lookup
 - Allows controllers to update connection map without polling for changes
- SCMP does not attempt to verify stream ends have the same format, that is up to the enumeration/control protocol

Timeouts

- 2000ms for connect TX stream command
 - allows for MAAP allocation period
- 4500ms for connect RX stream command
 - allows for connect TX command, retry of connect TX command and some processing time
- 200ms for all other TX commands and RX state command
- 500ms for all other RX commands

Connection Modes

- Saved state loading and restore
 - Fast Connect
 - Fast Disconnect
- Controller directed connections
 - Controller Connect
 - Controller Disconnect

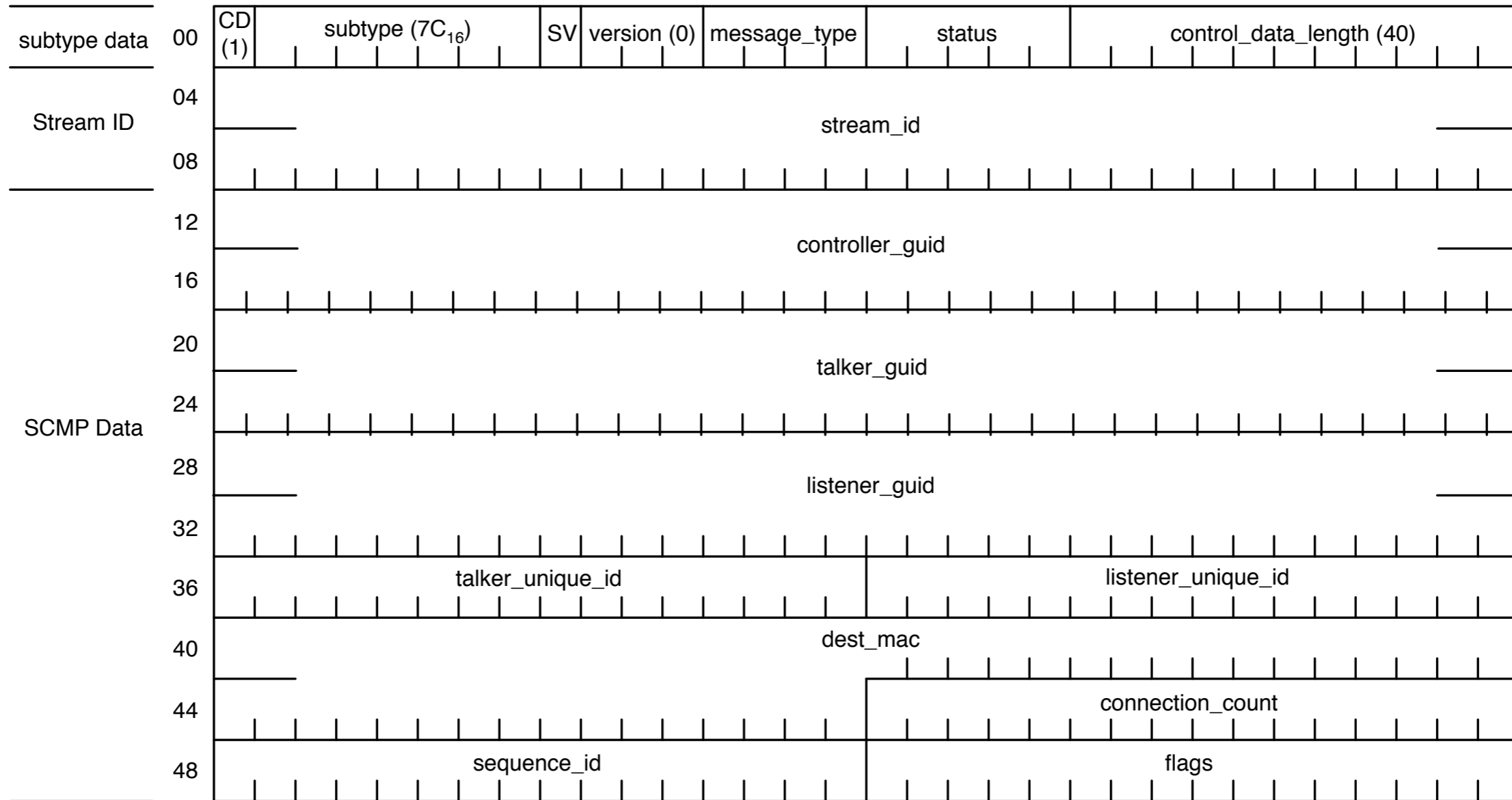
Connection Entities

- The Talker
 - The Listener
 - The Controller
-
- A device can be one or more of these, it doesn't have to implement them all

End Stations

- SCMP is an implied service of 1722.1
- Has one or more of the 3 SCMP entities as appropriate for it's functions
 - A microphone may have just a talker entity, or may have all 3
 - A control device may have just the controller

SCMPDU Format



SCMPDU Fields

- sv
- version
- message_type
- status
- stream_id
- controller_guid
- talker_guid
- listener_guid
- talker_unique_id
- listener_unique_id
- dest_mac
- connection_count
- sequence_id
- flags

Connection command flow

- Controller sends message to listener to connect/disconnect talker
 - Only in Controller Connect/Disconnect mode
- Listener sends message to talker to connect/disconnect itself
 - Sent in all modes

Listener RX Commands

- CONNECT_RX
- DISCONNECT_RX
- GET_RX_STATE

CONNECT_RX

- Establish a connection from the talker to the listener
- Command is sent by the controller
- Triggers sending a CONNECT_TX to the talker to establish the details for the connection
- On successful response from CONNECT_TX triggers MSRP listener registration

DISCONNECT_RX

- Remove listener from connection
- Command is sent by the controller
- Triggers sending a DISCONNECT_TX to the talker
- Listener uses this to remove the MSRP listener registration

GET_RX_STATE

- Get the current state of the RX side of the connection
- Command is sent by a controller
- Returns the connection status and if connected the stream ID and destination multicast MAC.

Talker TX Commands

- **CONNECT_TX**
 - Sent by the listener
- **DISCONNECT_TX**
 - Sent by the listener
- **GET_TX_STATE**
 - Sent by the listener or controller
- **GET_TX_CONNECTION**
 - Sent by the listener or controller

CONNECT_TX

- Establishes a connection from a talker to a listener
- Command is sent by the listener
- Provides the listener with the stream's ID and destination multicast MAC
- Talker uses this to trigger MSRP talker registration (if necessary) (and optionally also MAAP allocation for the stream)

DISCONNECT_TX

- Removes a listener from an established connection
- Command is sent by the listener
- Talker uses this to deregister MSRP talker (if there are no other connections)

GET_TX_STATE

- Gets the current state of the talker side of the connection
- Command is sent by a controller
- Returns the number of connected listeners, the stream ID and destination multicast address if connected

GET_TX_CONNECTION

- Gets the details of a specific connection from talker to listener
- Command is sent by a controller
- Returns the listener_guid and listener_unique_id for an indexed connection

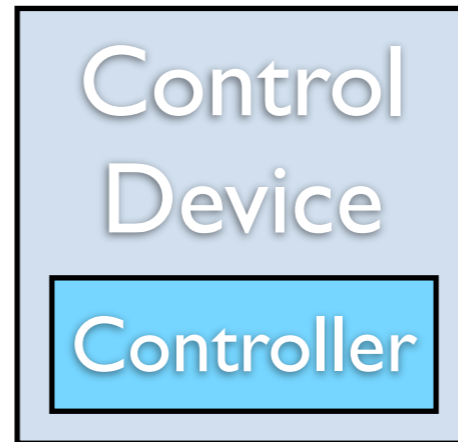
Status Codes

- SUCCESS
- LISTENER_UNKNOWN_ID
- TALKER_UNKNOWN_ID
- TALKER_DEST_MAC_FAIL
- TALKER_NO_STREAM_INDEX
- TALKER_NO_BANDWIDTH
- TALKER_EXCLUSIVE
- LISTENER_TALKER_TIMEOUT
- LISTENER_EXCLUSIVE
- STATE_UNAVAILABLE
- NOT_CONNECTED
- NO_SUCH_CONNECTION
- NOT_SUPPORTED
- UNABLE_TO_SEND_MESSAGE

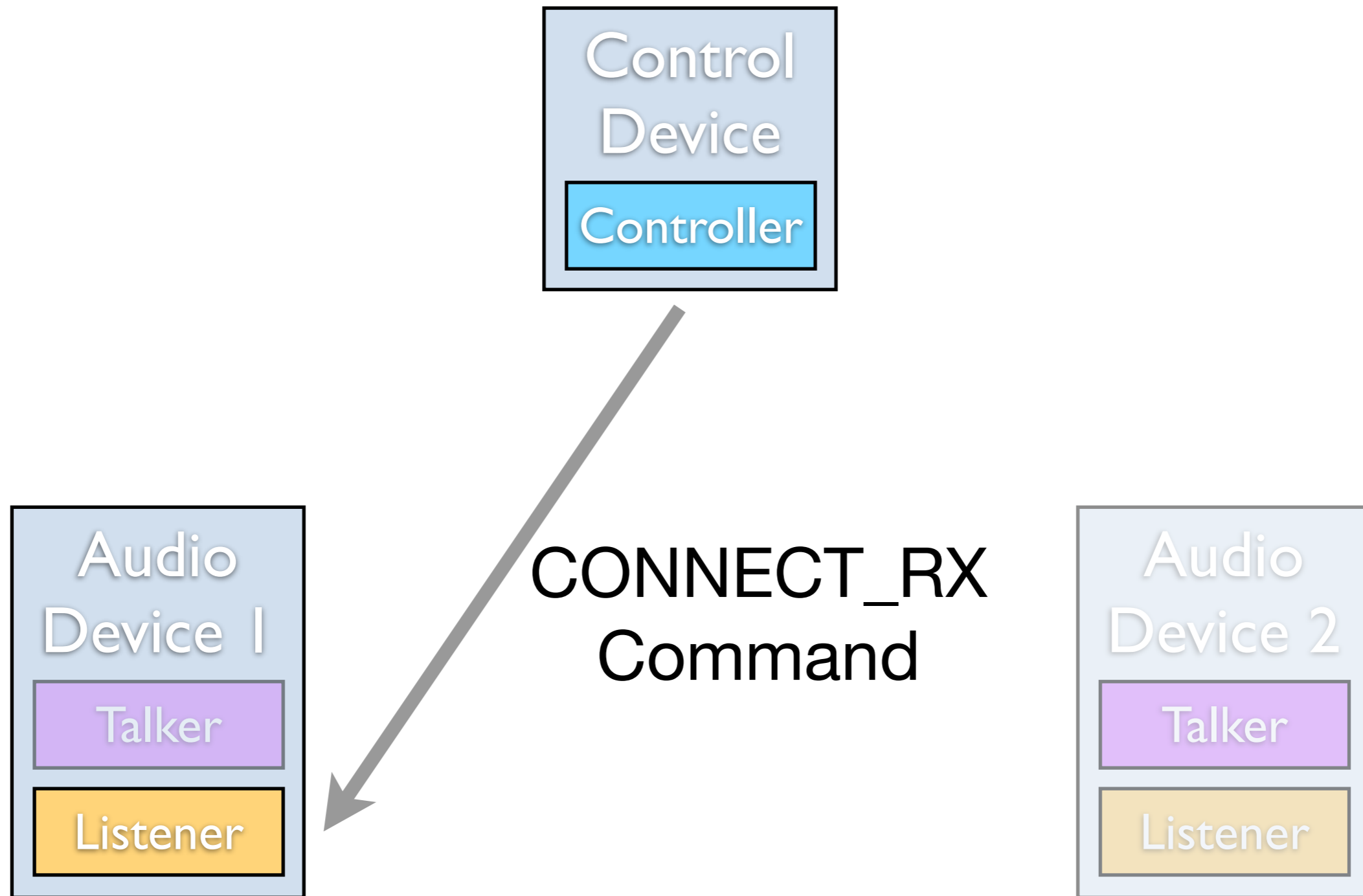
Flags

- 16-bit bitfield
- 3 defined flags
 - CLASS_B
 - FAST_CONNECT
 - SAVED_STATE
 - Talker/Listener has saved state in non-volatile memory

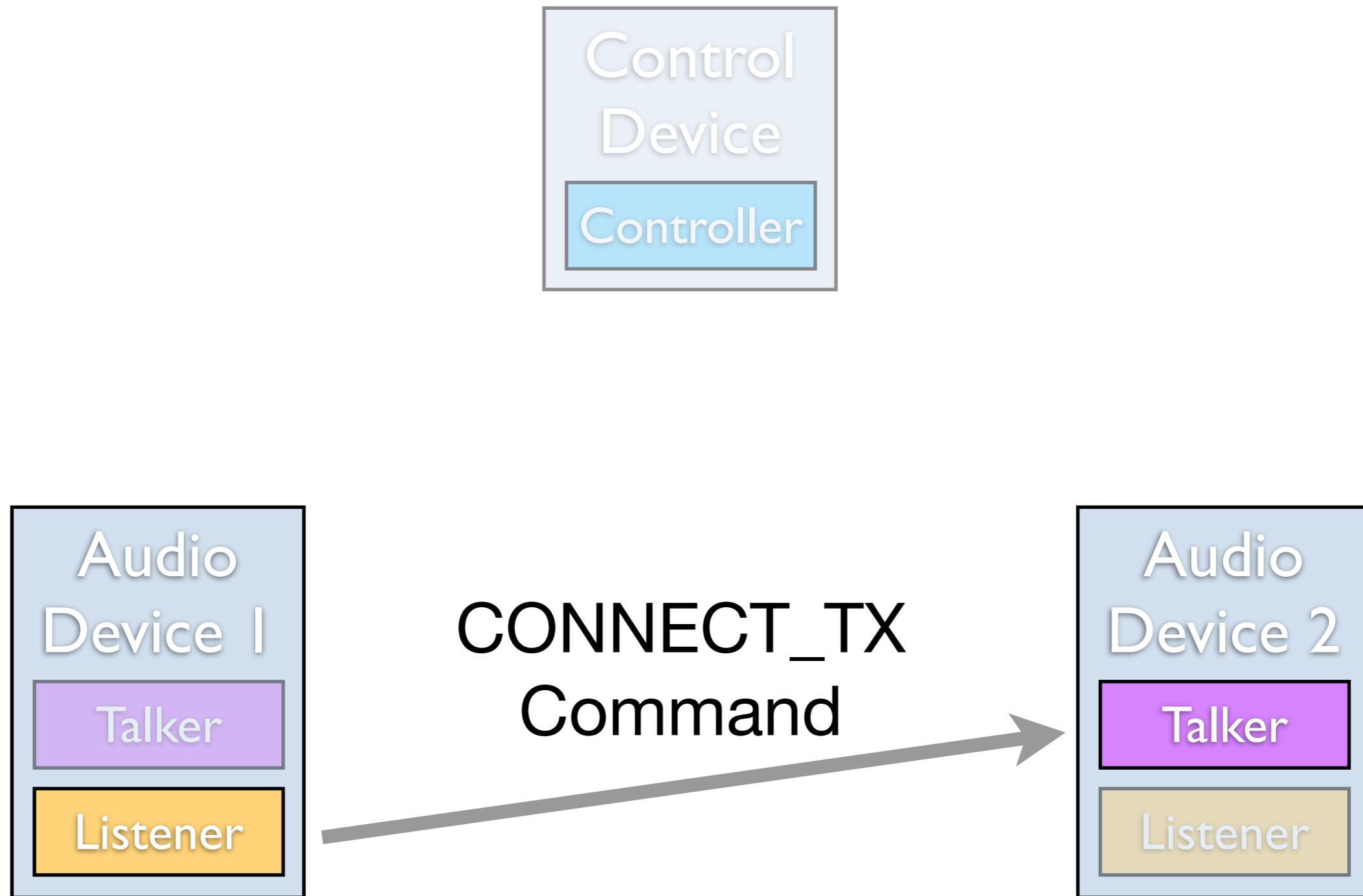
Controller Connect Message Flow



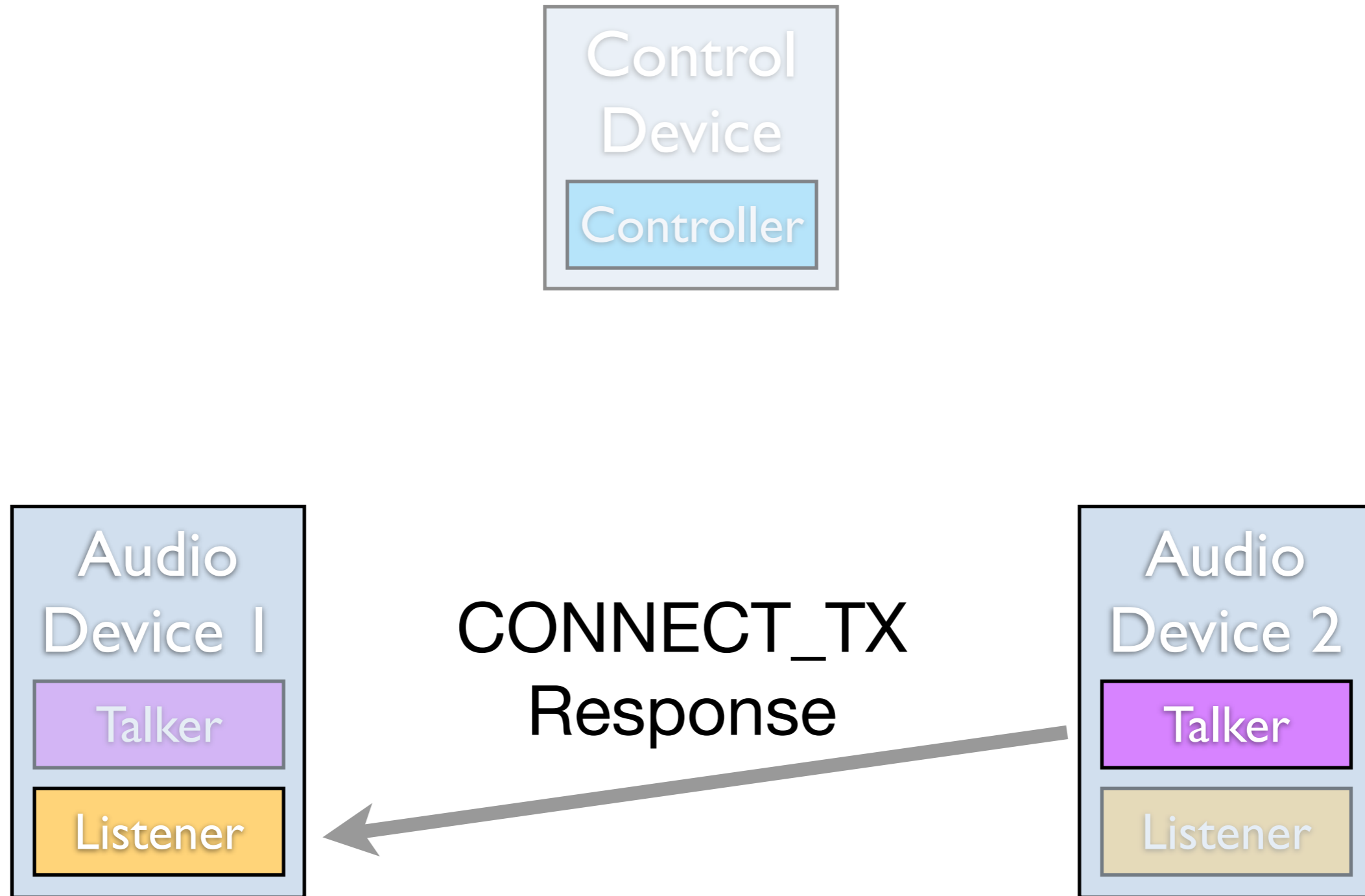
Controller Connect Message Flow



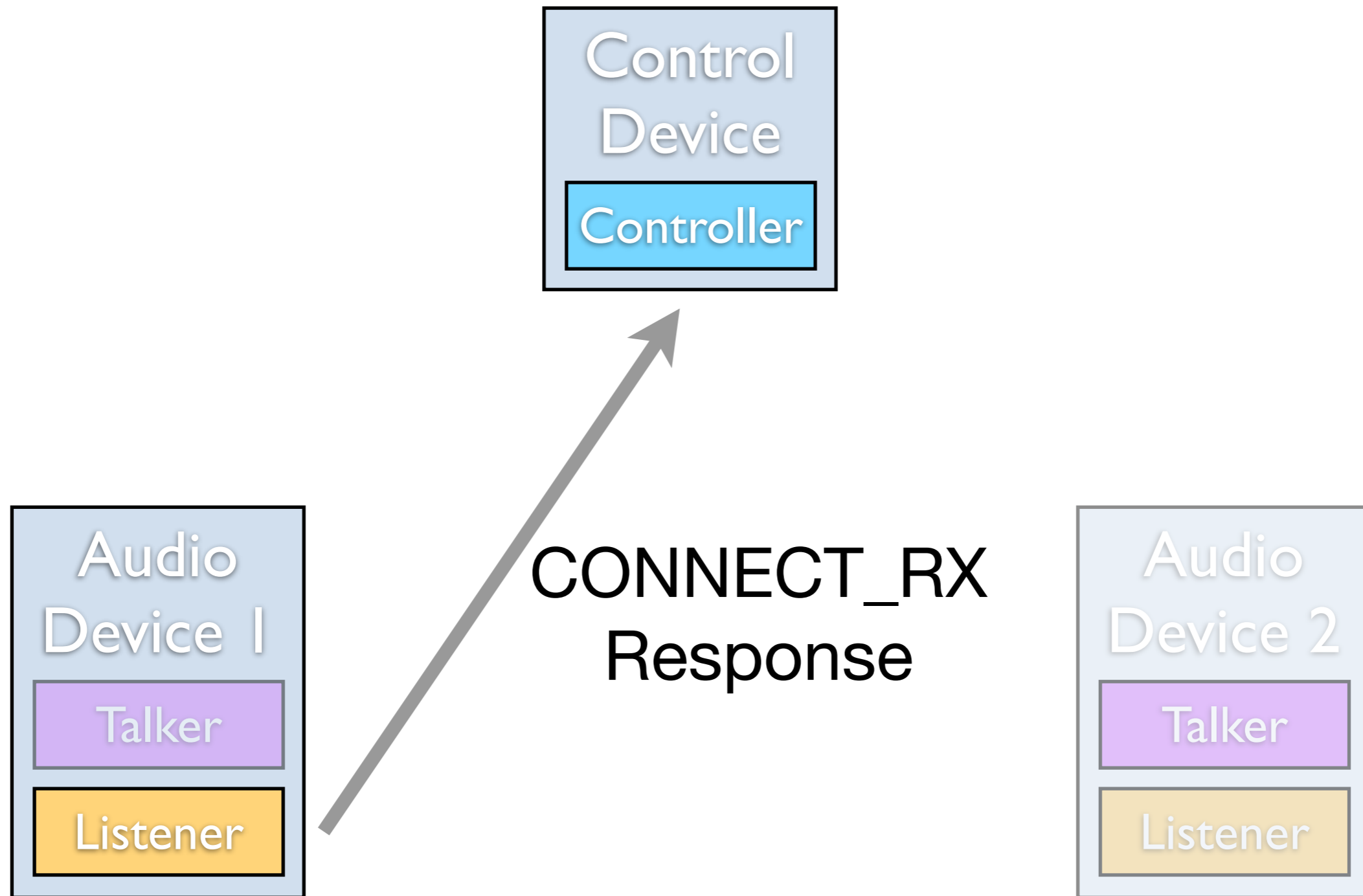
Controller Connect Message Flow



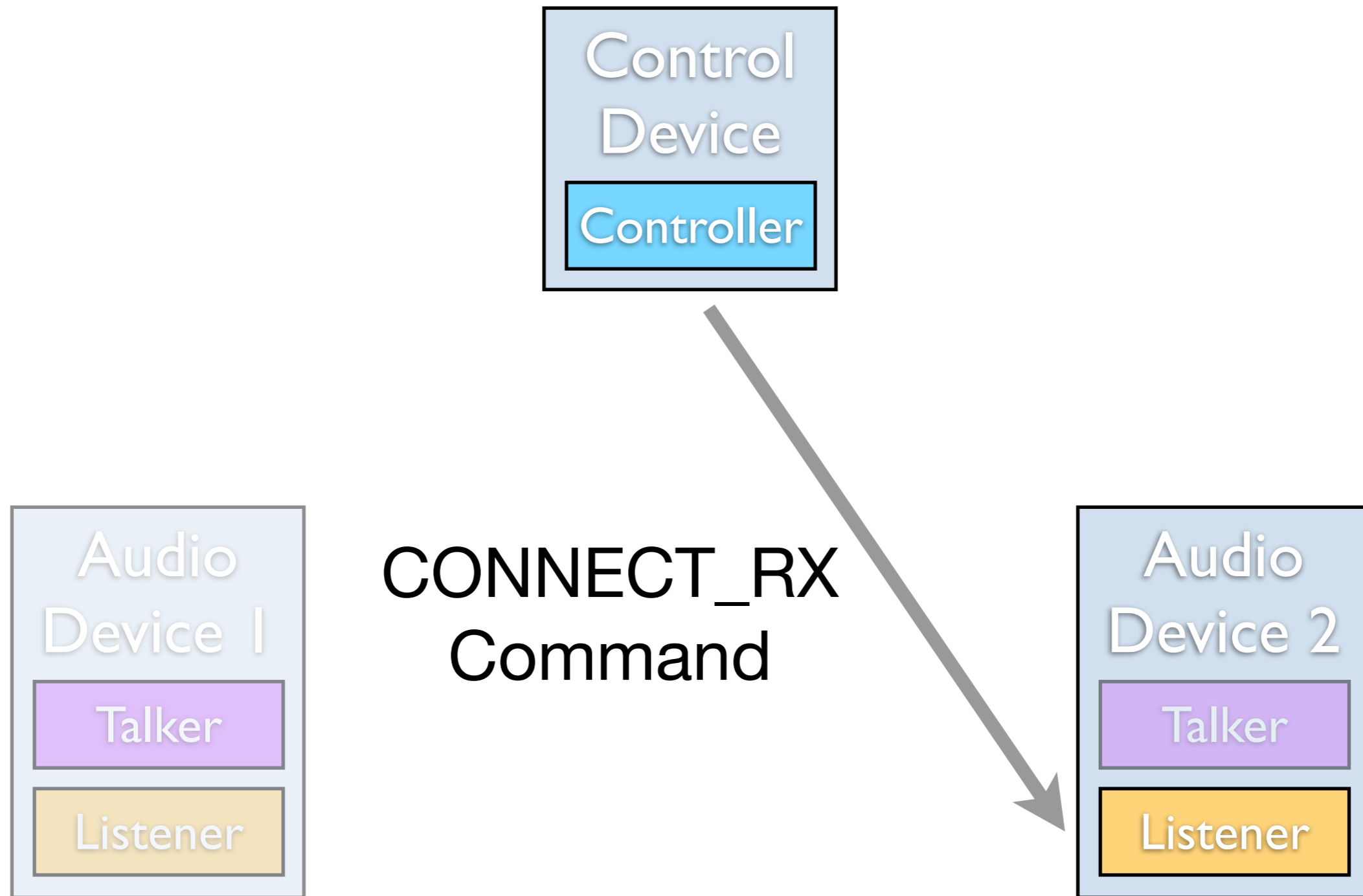
Controller Connect Message Flow



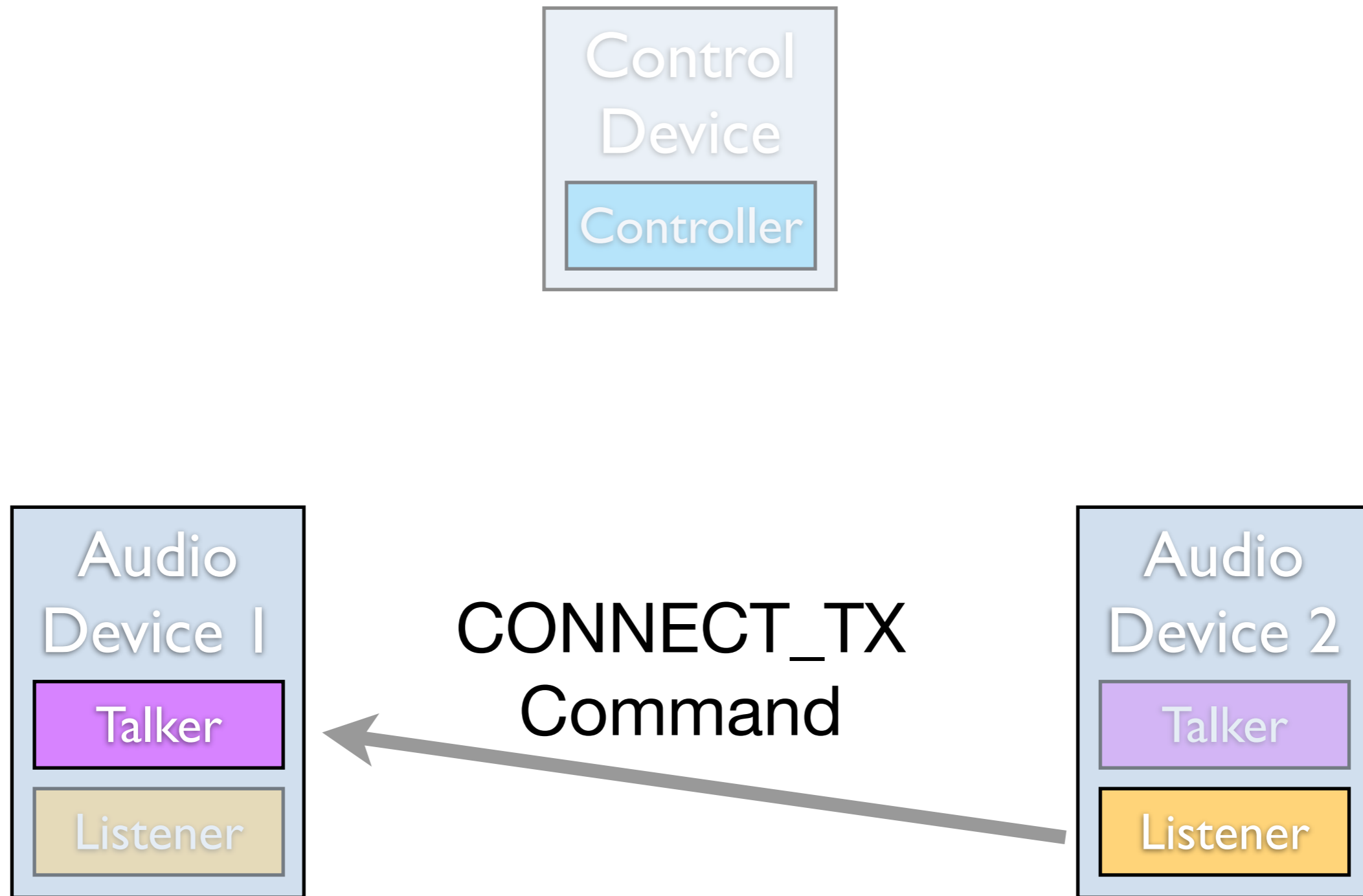
Controller Connect Message Flow



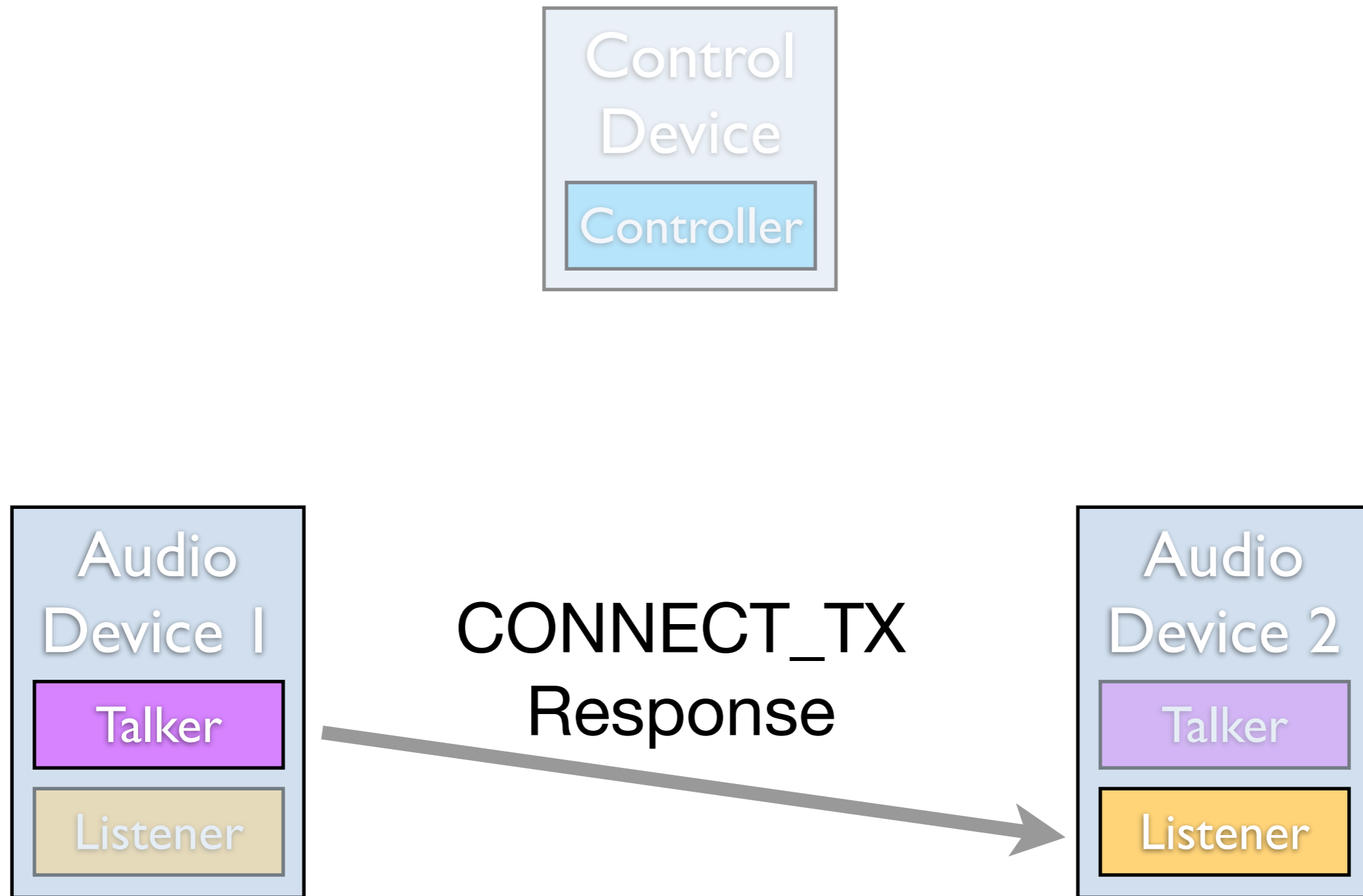
Controller Connect Message Flow



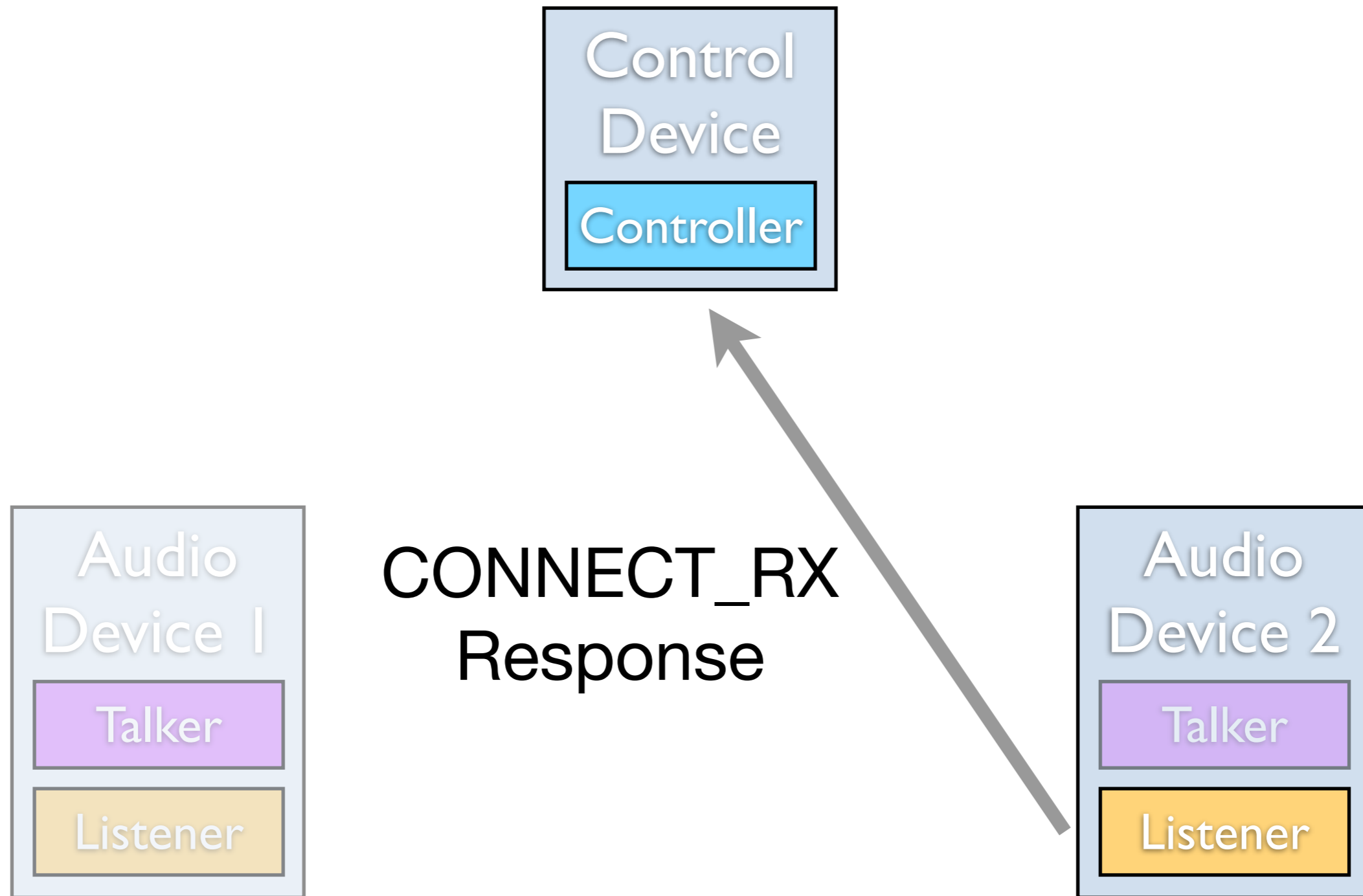
Controller Connect Message Flow



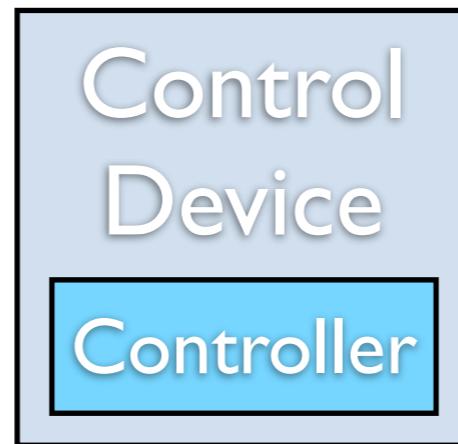
Controller Connect Message Flow



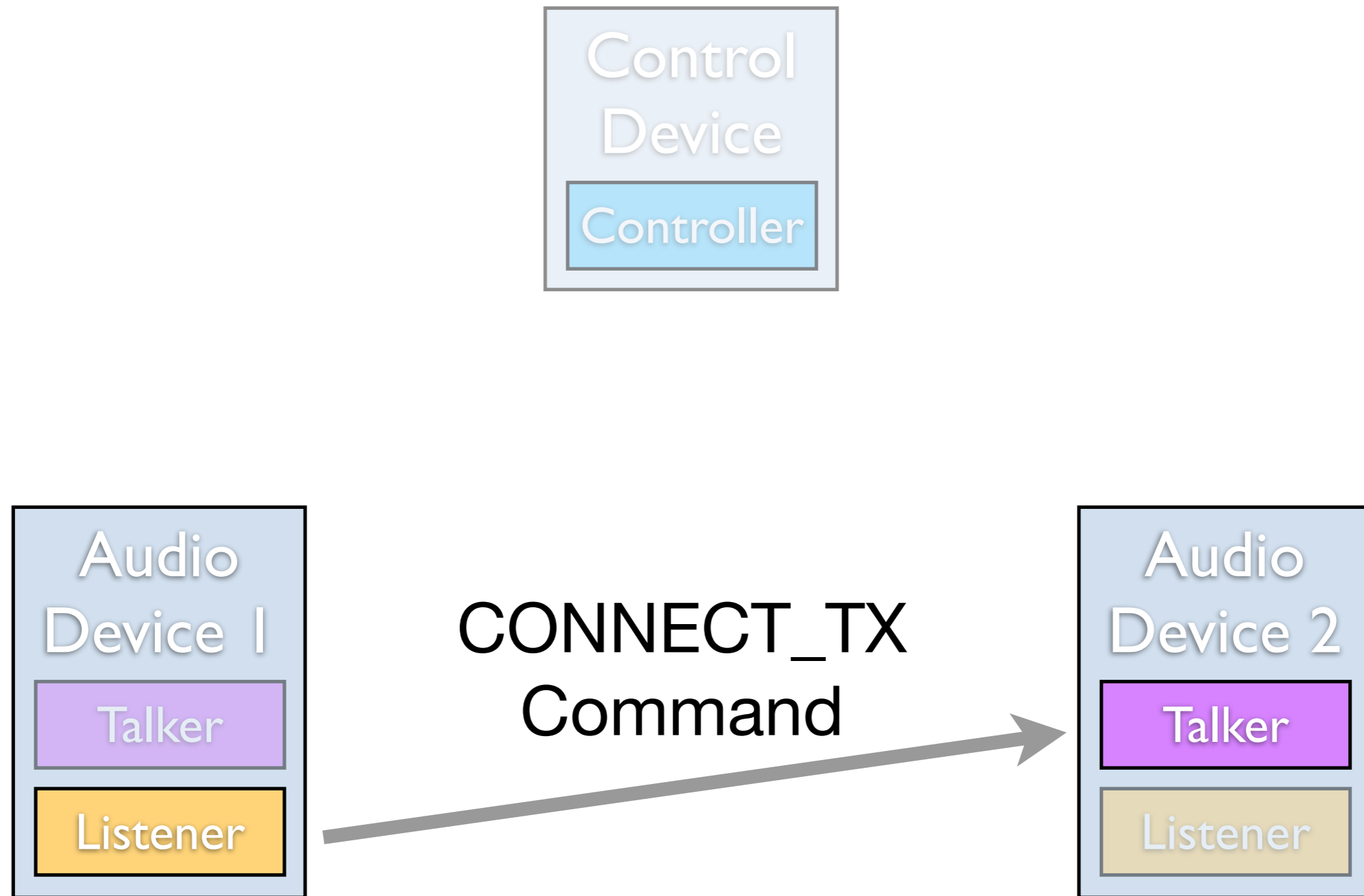
Controller Connect Message Flow



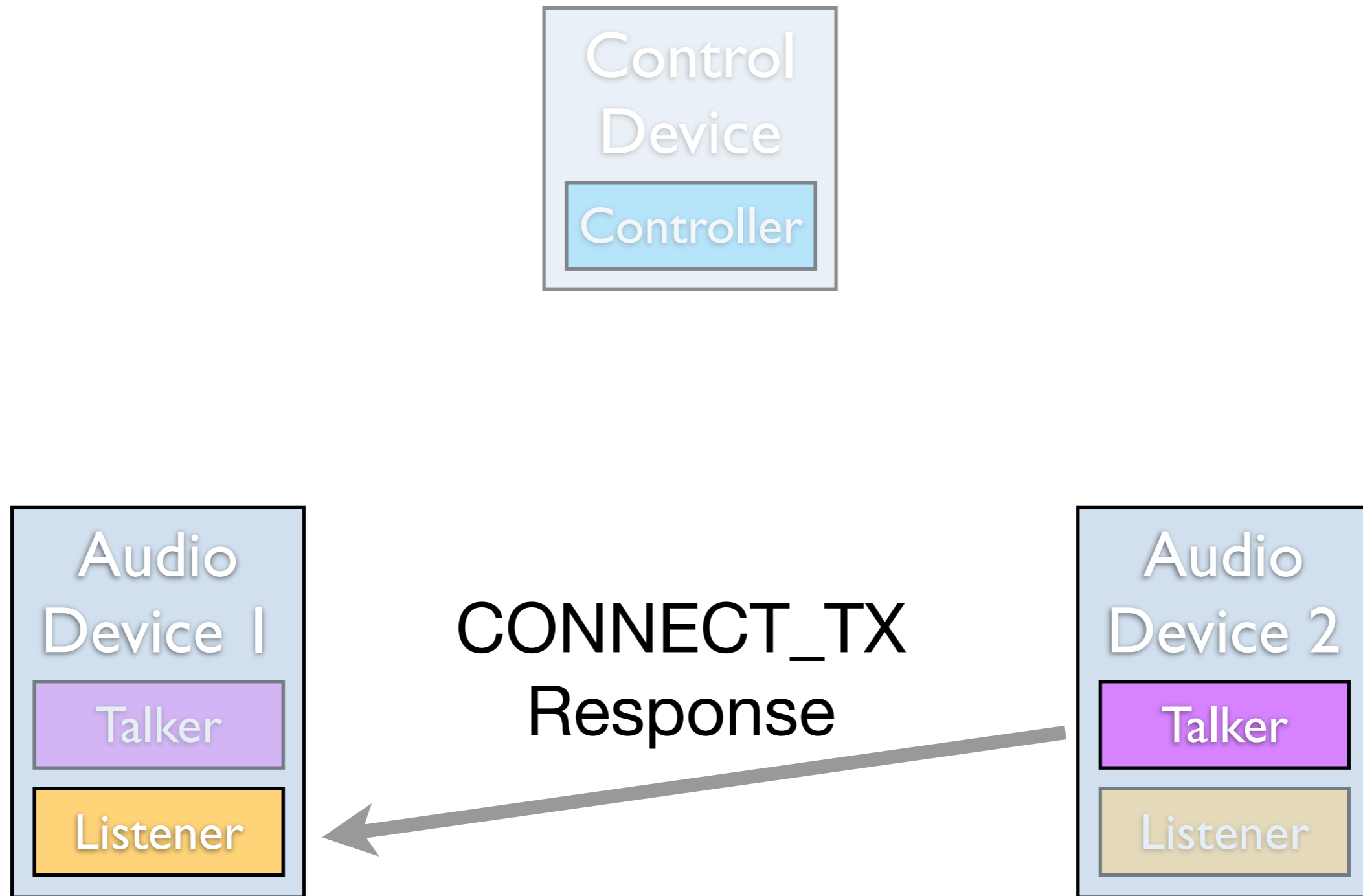
Fast Connect Message Flow



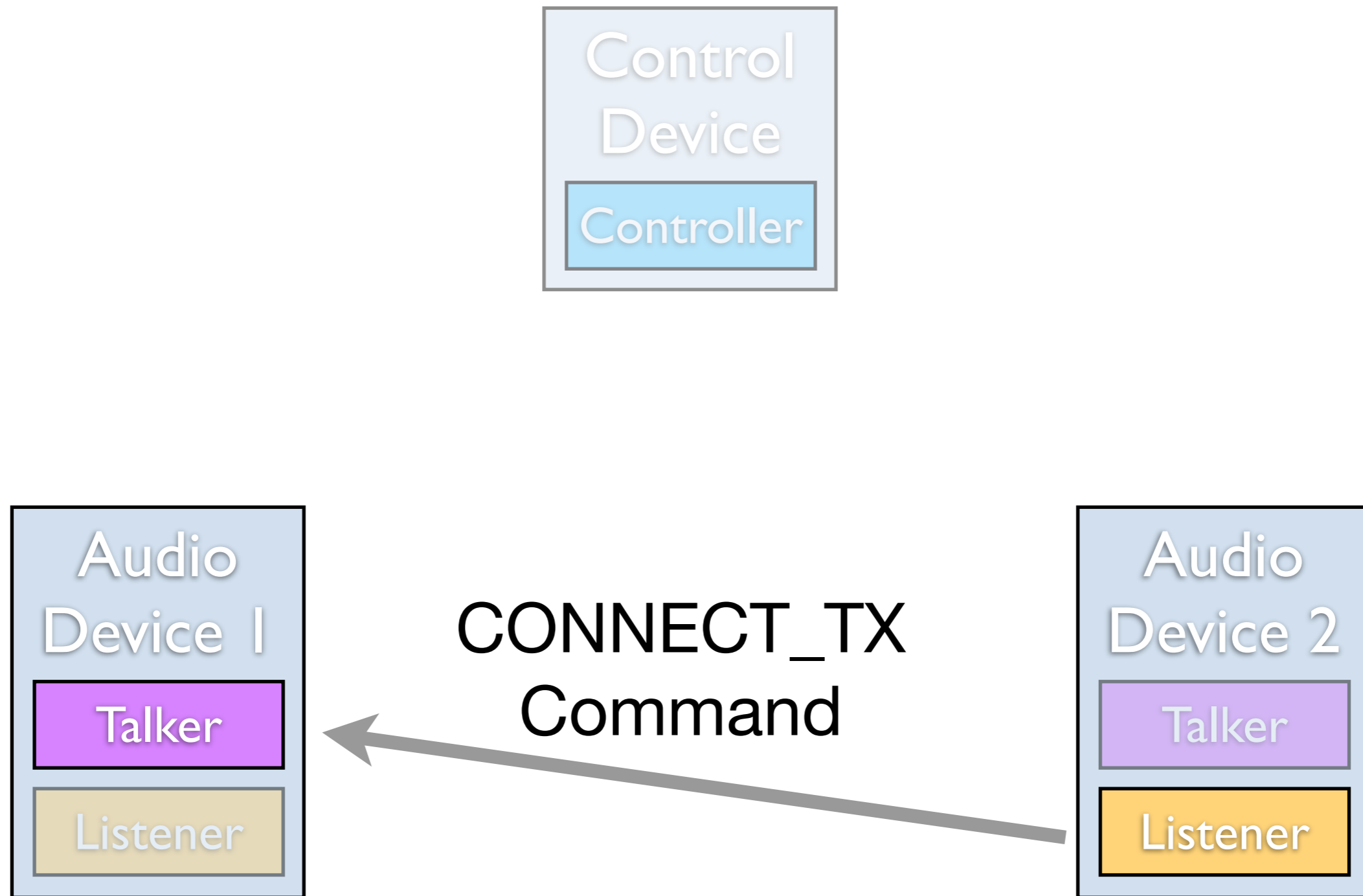
Fast Connect Message Flow



Fast Connect Message Flow



Fast Connect Message Flow



Fast Connect Message Flow

