

Isochronous resource management as submitted to the p1394.1 committee

Dr. David V. James, Sony
3300 Zanker Road, MS SJ3C1
San Jose, CA 95134-4591
Phone: 408 955-6295
FAX: 408 955-4591
Email: dvj@alum.mit.edu

October 12 1999

**This contribution is one of several, presented for independent review.
An overall contribution, which provides the context for multiple contributions,
is also provided in BR047R08.**

Isochronous resources are managed through the use of messages sent between portals.
The talker bus has a talker proxy, path bridges have pilot proxies, and the listener bus has a listener bus.
The use of these proxies, error recovery, message formats, and proxy storage are specified.
The current proposal assumes a bit map be maintained for affiliated listening nodes;
in the absence of a compelling application, this should be replaced with a count value.

2. References and definitions

NOTE—The following lists identify additional material for the current draft and therefore not intended to be complete.

2.1 References

(etc)

2.2 Technical glossary

The following terms are used in this standard:

2.2.1 listener-path portal: Defined relative to a given node and isochronous connection. If the node and the listener are on separate buses, the first portal on the path between the node and the listener. Otherwise, the delta portal on the listener's bus.

2.2.2 listener proxy: A local-bus resource, affiliated with a isochronous listener, that is responsible for establishing and maintaining the talker's isochronous connection.

2.2.3 pilot proxy: A local-bus resource, affiliated with an isochronous bridge node, that is responsible for establishing and maintaining the bridge's isochronous connection.

2.2.4 talker proxy: A local-bus resource, affiliated with a isochronous talker, that is responsible for establishing and maintaining the talker's isochronous connection.

2.2.5 talker-path portal: Defined relative to a given node and isochronous connection. If the node and the talker are on separate buses, the first portal on the path between the node and the talker. Otherwise, the delta portal on the node's bus.

2.2.6 talking proxy: Defined for a give node and isochronous connection, based on the node's location. Refers to the talker proxy or the talker-path-resident pilot proxy, if the node is on the talker bus or another bus respectively.

5. Isochronous connection management

5.1 Isochronous overview

5.1.1 Isochronous stream identifiers

An objective of the isochronous connection management is to maintain isochronous communications in the presence of semistable *nodeId* and *channel* number assignments. For this reason, managed connections are identified by the *EUI* and *plugId* of the talker. The combination of these two values is called the *streamId*.

For unmanaged connections, the talker has no *plugId*, so the pseudo-plug identifier is provided by the controller.

5.1.2 Overlaid connections

On talker may be connected to multiple listeners and portions of the route to these buses may be the same. To improve efficiency, only one isochronous channel is used on the shared portion of the connection and a use count is maintained. When the use count is larger than one, this is called an overlaid connection.

When multiple logical connections follow the same hops, these connections are “overlaid” to reduce isochronous resource requirements. An overlaid connection requires an additional up-stream counter, to avoid premature disconnections, but uses no additional isochronous channel or bandwidth.

The overlaid counter identifies the number of portals that are currently listening to the channel, and therefore is always less than 64. In a sense, the overlaid connection can “branch” on each bus and the connection count reflects the number of branches. Since each branch of an overlaid connection may itself have more branches, the total number of overlaid connections can be much larger than 64.

5.2 Isochronous proxies

To maintain the time-sensitive integrity of isochronous connections, bridge portals provide isochronous proxies that act on behalf of the connection controller. For each isochronous connection, talker and listener proxies are located on talker and listener buses; other pilot proxies are located on intermediate buses. Each type of proxy has a distinct set of responsibilities, listed below:

- 1) Listener. A listener proxy is located on the listener's delta portal. Its responsibilities include:
 - a) Bus reset recovery.
 - i) Verify. Local ROMs are checked to verify the continued presence of the listener node.
 - ii) Revise. The talker-provided channel number is copied into the listener portal's iPCR.
 - b) Bus change. Ping the talker proxy after each *bus_changed* indication. This intent of this ping is to verify the continued presence of the listener proxy.
- 2) Talker. A talker proxy is located on the talker's delta portal. Its responsibilities include:
 - a) Bus reset recovery.
 - i) Verify. Local nodes are checked to verify the continued presence of the talker.
 - ii) Reacquire. Isochronous resources are reacquired in a timely fashion.
 - iii) Revise. The talker's oPCR is updated with the current isochronous channel.
 - iv) Distribute. Listener and pilot proxies are informed of the current isochronous channel.
 - b) Disconnect. The resources are released in the absence of observed "ping" events.
 - i) Detached. A write to the talker's oPCR detaches the talker node.
 - ii) Deallocate. Internal actions release (believed to be unused) talker-proxy resources.
- 3) Pilot. A pilot proxy is located on each of the listener-to-talker portals. Its responsibilities include:
 - a) Bus reset. The pilot proxy's listener-side is involved in bus-reset recovery, as follows:
 - i) Reacquire. Isochronous resources are reacquired in a timely fashion.
 - ii) Reattach. Listener and pilot proxies are informed of the current isochronous channel.
 - b) Disconnect. In the absence of observed "ping" events, talker-proxy resources are released.

Locating the listener proxy in the delta portal, as opposed to alpha portal, reduces instabilities associated with possible alpha-portal changes (see 1.13.5).

The steps associated with connect, disconnect, bus reset, and *bus_changed* activities are illustrated in the following subclauses.

5.3 Isochronous management sequences

5.3.1 Nonoverlaid isochronous connections

5.3.1.1 Nonoverlaid connection

Establishment of an isochronous connections begins with knowledge of the isochronous payload size. This normally involves a read of the talker node's oPCR (not illustrated) or ROM. Connection management does requires this information, but its location and format are beyond the scope of this standard.

The connection formation with a message sent from the controller to the listener-bus delta portal, as illustrated in figure 1. Connection steps involve portal-to-portal messages and portal-to-IRM transactions, as listed below. Internal transfers, which never appear on a bus, are identified by a '*' marker.

- 1) Dispatch. The controller directs a dispatch message to the listener's delta portal.
*Allocate. The listener's delta portal allocates a listener proxy resource.
- 2) Acquire. The pilot proxy allocates bus-local isochronous resources.
- 3) Handoff. The pilot proxy directs the message towards the talker's delta portal.
*Allocate. The next talker-path portal allocates a pilot proxy resource.
- 4) Acquire. The pilot proxy acquires the necessary bus-local isochronous resources.
- 5) Handoff. The pilot proxy directs the connect message to the talker's delta portal.
*Allocate. The talker's delta portal allocates a talker proxy resource.
- 6) Acquire. The talker proxy allocates the necessary bus-local isochronous resources.
- 7) Attach. The talker proxy communicates writes the channel number into the talker's oPCR.

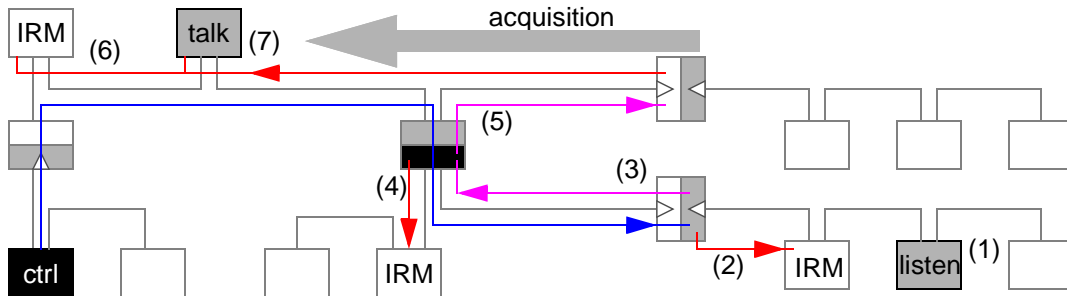


Figure 1—Nonoverlaid connection acquisition (glossary see 2.4.2)

Connection confirmation flows in the opposite direction, as illustrated in figure 2. Confirmation messages commit speculatively allocated resources and communicate channel numbers, as listed below.

- 8) Commit. The talker-bus delta portal directs a commit messages towards the listener's delta portal.
- 9) Commit. The pilot proxy redirects the message towards the listener's delta portal.
- 10) Attach. The listener's proxy writes the channel number to the listener's iPCR.
- 11) Reply. The listener's proxy directs a connection-complete reply to the controller.

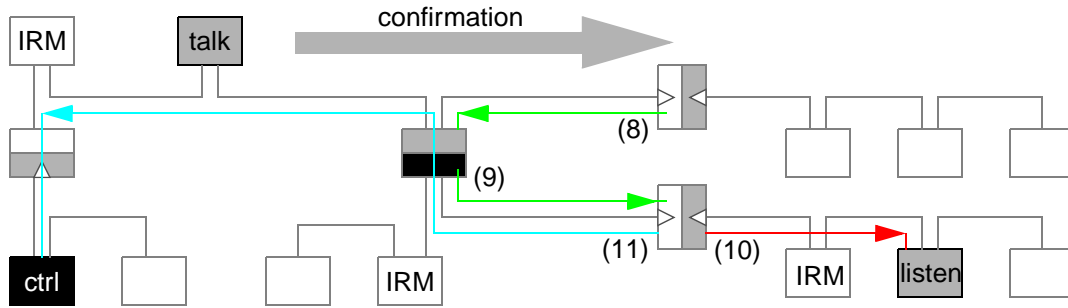


Figure 2—Nonoverlaid connection confirmation (glossary see 2.4.2)

5.3.1.2 Nonoverlaid connection state

The isochronous connection steps establish a listener proxy in the listener's delta portal, pilot proxies in intermediate bridges, and a talker proxy in the talker's delta portal, as illustrated in figure 3

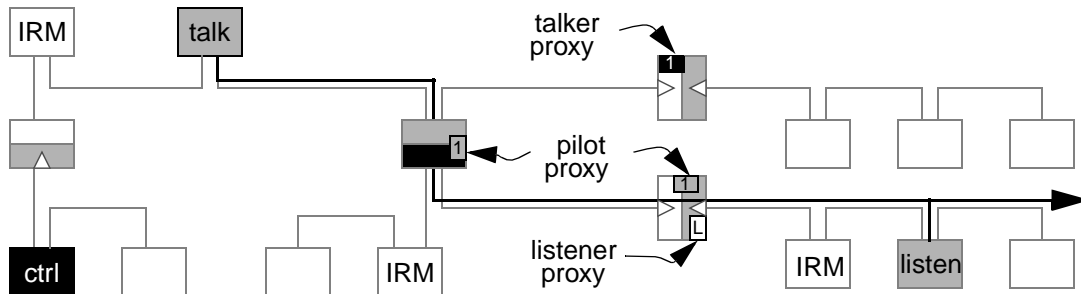


Figure 3—Nonoverlaid connection proxies

5.3.1.3 Nonoverlaid disconnection

Isochronous disconnection also starts with the listener and involves a sequence of portal-to-portal messages and portal-to-IRM transactions, as illustrated in figure 4. Note that the disconnect reply is returned from the talker-bus alpha portal, rather than the listener's talker-path portal.

- 1) Dispatch. The controller directs a disconnect message to the listener's delta portal.
*Detach. The listener proxy detaches the listener, by writing its iPCR.
- 2) *Release. The pilot proxy releases the connections's allocated resource.
- 3) Handoff. The pilot proxy redirects the message towards the talker-proxy portal.
- 4) Release. The pilot proxy releases previously acquired isochronous resources.
- 5) Handoff. The pilot proxy redirects the message towards the talker's delta portal.
- 6) Detach. The talker proxy detaches the talker, by directing a write to its oPCR.
- 7) *Release. The talker proxy releases previously acquired isochronous resources.
*Handoff. The talker proxy redirects the message towards the talker's delta portal.
*Release. The talker's delta portal releases its talker proxy resources.
- 8) Reply. The talker's delta portal directs a disconnect-complete reply to the controller.

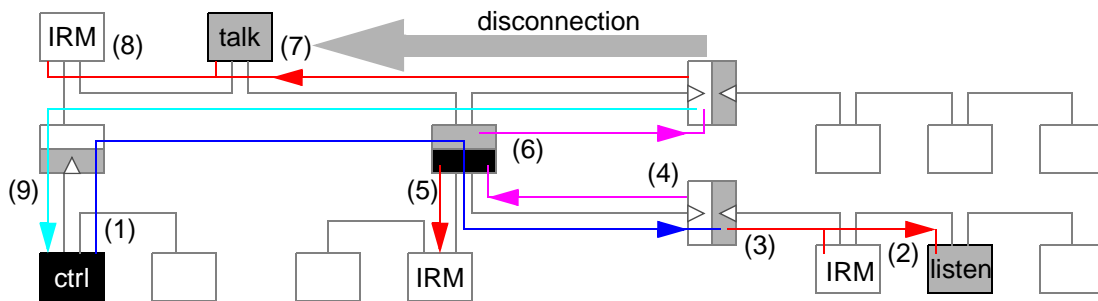


Figure 4—Nonoverlaid disconnection (glossary see 2.4.2)

5.3.2 Overlaid connections

5.3.2.1 Overlaid connection

A fully overlaid connection completes quickly, sharing the previously acquired isochronous resources, as illustrated in figure 5 and listed below.

- 1) Dispatch. The controller dispatches a disconnect message to the listener's delta portal.
 *Allocate. A listener proxy resource is allocated in the listener's delta portal.
- 2) Redirect. The listener proxy redirects the connect message towards the talker.
 *Acquire. The pilot proxy detects the overlaid connection and sets an *active_connections* bit.
- 3) Commit. The pilot proxy returns a commit message towards the listener proxy.
- 4) Acquire. The listener proxy attaches the listener node, by directing a writing to its iPCR.
- 5) Reply. The listener proxy directs a connection-complete reply for the controller.

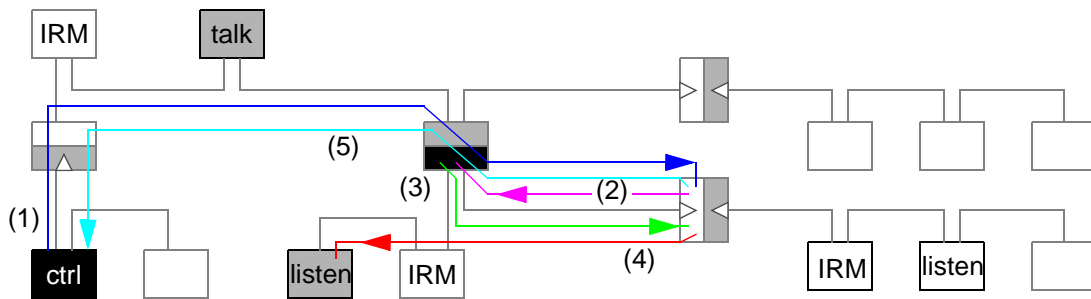


Figure 5—Overlaid connection (glossary see 2.4.2)

5.3.2.2 Overlaid connection state

The isochronous connection establishes a listener proxy and sets an *active_connections* bit in the talker-path pilot proxy, as illustrated in figure 6.

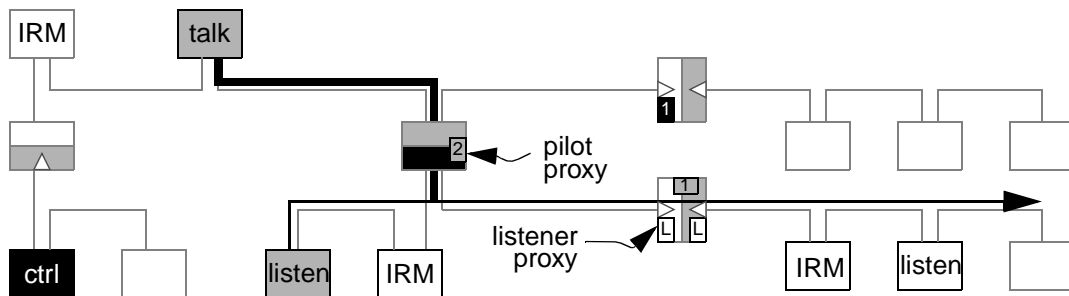


Figure 6—Overlaid connection state

5.3.2.3 Overlaid disconnection

Fully overlaid isochronous disconnection and connection steps involve a sequence of portal-to-portal messages and portal-to-IRM transactions, illustrated in figure 7 and listed below:

- 1) Dispatch. The controller dispatches a disconnect message to the listener proxy.
- 2) Detach. The listener proxy detaches the listener, by writing to its iPCR.
*Deallocate. The listener's delta portal deallocates the listener proxy resources.
- 3) Redirect. The listener's delta portal redirects the message to its talker-path pilot proxy.
*Release. The pilot proxy clears the appropriate *active_connections* bit.
(other bits are set, so the local-bus isochronous resources are retained).
- 4) Reply. The listener's talker-path portal directs disconnection-complete reply to the controller.

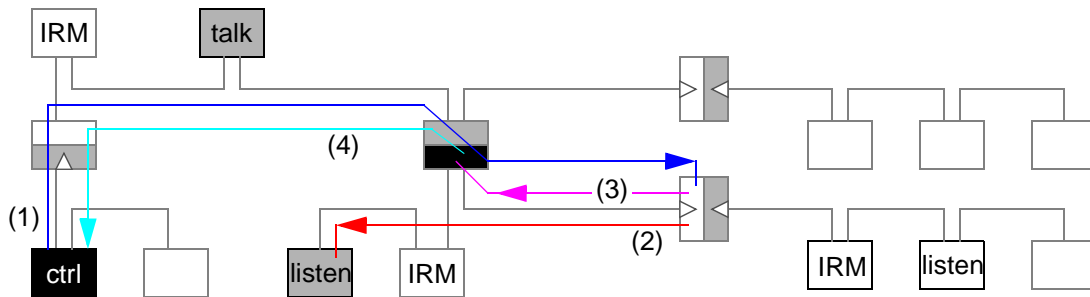


Figure 7—Overlaid disconnection (glossary see 2.4.2)

5.3.3 Partially overlaid connections

5.3.3.1 Partially overlaid connection

A partially overlaid connection involves resource allocations on the nonoverlaid segments, as illustrated in figure 8 using the steps listed below:

- 1) Dispatch. The controller sends a connect message to the listener's delta portal.
*Allocate. The listener's delta portal allocates listener-proxy resources.
*Handoff. The listener proxy transfers the message to its talker-path portal.
*Allocate. The talker-path portal allocates pilot-proxy resources.
- 2) Acquire. The listener's talker-path portal acquires the necessary listener-bus isochronous resources.
*Handoff. The listener proxy transfers the message to the talker proxy.
*Acquire. The talker proxy detects the overlaid connection and sets an *active_connections* bit..

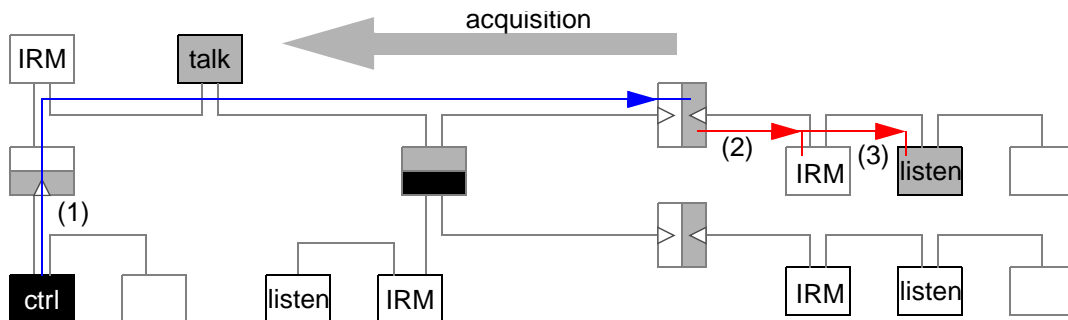


Figure 8—Partial-overlay acquisition (glossary see 2.4.2)

A partial-overlay connection completes with commit messages between the talker and listener proxies, as illustrated in figure 9 and listed below:

- 3) *Commit. The talker proxy directs a commit message to the listener proxy.
Attach. The listener proxy attaches the listener, by writing to the listener's iPCR.
- 4) Reply. The listener proxy generates a connection-complete reply for the controller.

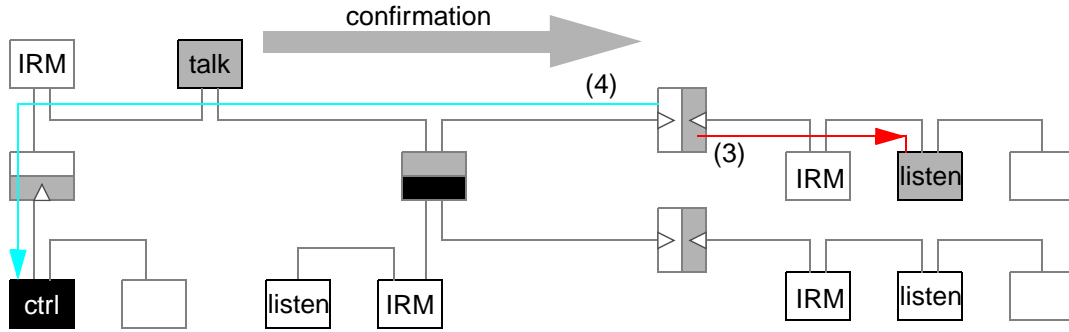


Figure 9—Partial-overlay confirmation (glossary see 2.4.2)

5.3.3.2 Partially overlaid connection state

The isochronous connection steps establish a listener proxy, a pilot proxy, and sets an *active_connections* bit in the talker proxy, as illustrated in figure 10.

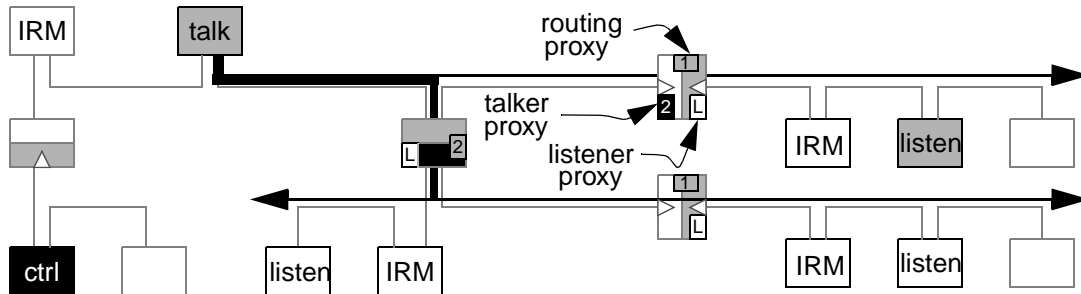


Figure 10—Isochronous connection results

5.3.3.3 Partially overlaid disconnection

A partially overlaid isochronous disconnection involve a similar sequence of portal-to-portal messages and portal-to-IRM transactions (see figure 11), but no confirmation phase is required, as illustrated in figure 11 and described in the steps below:

- 1) Dispatch. The controller directs a disconnect message to the listener proxy.
*Redirect. The listener proxy detaches the listener by writing to its iPCR.
*Handoff. The listener proxy redirects the message to the talker-path pilot proxy.
- 2) Detach. The pilot proxy releases overlaid resources, by directing transactions to the IRM.
*Deallocate. The pilot's portal deallocates the pilot-proxy resource.
*Redirect. The pilot's portal redirects the message to the talker proxy.
- 3) Detach. The talker proxy releases overlaid resources, by clearing an *active_connections* bit.
*Deallocate. The talker's delta portal deallocates the talker-proxy resources.
- 4) Reply. The talker's delta portal produces a disconnect-complete reply for the controller.

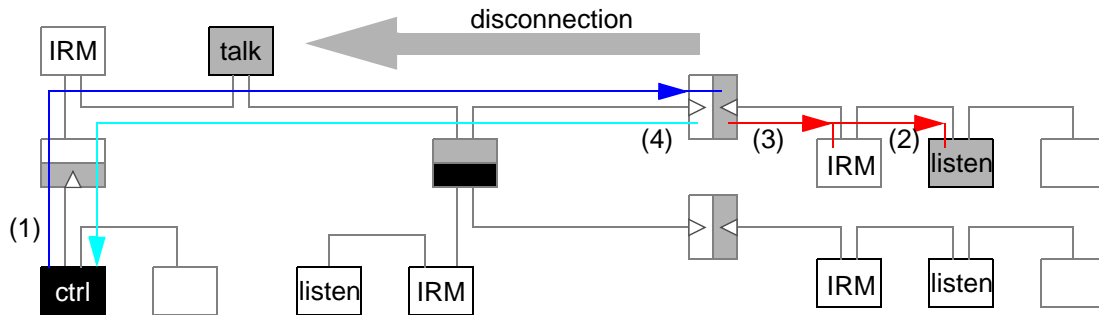


Figure 11—Partial-overlay disconnect (glossary see 2.4.2)

5.4 Bus-reset recovery

One of the reasons for invoking a bus reset is to resolve inconsistent or ambiguous bus-local isochronous resource allocations. The talker-path portal (or delta portal on the talker's bus) assumes this obligation, allowing resources to be reclaimed in a timely fashion, and (when necessary) informs other pilot/listener proxies of isochronous channel changes.

5.4.1 Talker-bus bus-reset recovery

After a bus reset, the listener proxies (in listener-bus delta or listener-path portals) is responsible for timely informing the delta portal of their presence. The talker proxy (talker-bus delta or talker-path portal) is then responsible for reacquisition of previously acquired resources. These steps are listed below and illustrated in figure 12.

- 1) Inform. The talker-bus listener/pilot proxies inform the delta-portal talker proxy of their presence.
- 2) Reclaim. The talker-bus delta portal reclaims previously acquired isochronous resources.
- 3) Reattach. The talker proxy directs a write to the talker oPCR, to reattach this connection.
*Reattach. The talker proxy directs writes to listener's iPCRs, to reattach these connections.
(these reattach writes also transport a new channel number, which may be changed by step (2)).
- 4) Restore. The talker proxy directes messages to pilot portals, to reattach these connections.

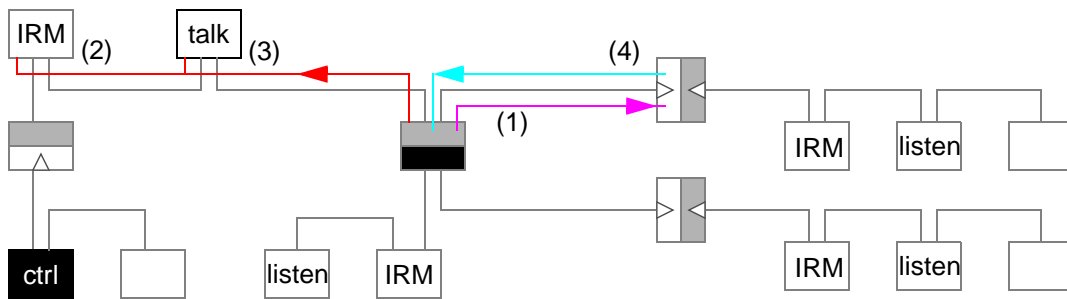


Figure 12—Talker-proxy bus-reset recovery (glossary see 2.4.2)

5.4.2 Nontalker-bus bus-reset recovery

Similar sets of bus-reset recovery operations are performed when resets occur on nontalker buses, as illustrated in figure 13.

- 1) Inform. The local-bus listener/pilot proxies inform the talker-path pilot proxy of their presence.
- 2) Reclaim. The talker-path proxy portal reclaims previously acquired isochronous resources.
- 3) Reattach. The talker proxy directs writes to listener's iPCRs, to reattach these connections. This reattach write also transports a new channel number, which can be changed by step (2).
- 4) Restore. The talker proxy directes messages to listener-path pilot proxies, to reattach them. This reattach message also transports a new channel number, which can be changed by step (2)..

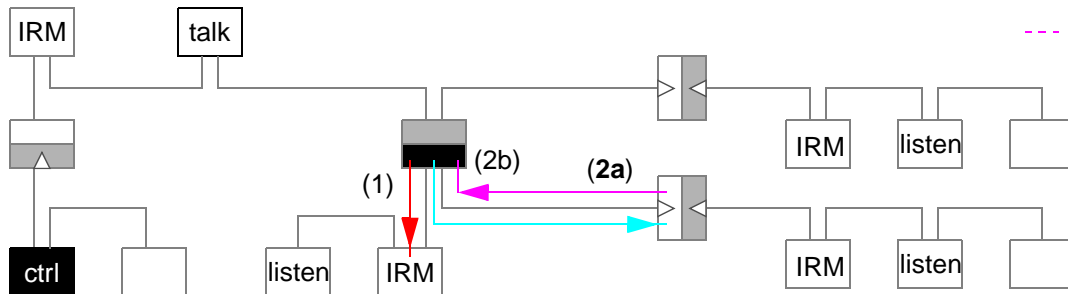


Figure 13—Pilot-proxy bus-reset recovery (glossary see 2.4.2)

5.5 Net-change recovery

One of the reasons for invoking a net refresh is to resolve inconsistent or ambiguous nonlocal isochronous resource allocations. The listener and talker proxies assume this obligation, allowing resources to be reclaimed (or lost) in a timely fashion.

5.5.1 Talker-bus net-refresh recovery

After receiving a *net_changed* event, all listener proxies are responsible for quickly pinging their talker proxy. These pings pass through intermediate pilot proxies, adjusting their state as well. These steps are listed below and illustrated in figure 14.

- 1a) *Declare. The bus-A listener proxy sends a message towards the talker proxy, declaring its presence.
*Reclaim. The bus-A pilot proxy updates *active_connections*, based on this declaration.
Redirect. The bus-A pilot proxy directs a message towards the talker proxy.
- 1b) Declare. The bus-B listener proxy sends a message towards the talker proxy, declaring its presence.
*Reclaim. The talker-path bus-B pilot proxy updates *active_connections*, based on this declaration.
- 1) Declare. The busB/C portal directs a message to the talker proxy, declaring its presence.
*Reclaim. The bus-A talker proxy updates *active_connections*, based on this declaration..

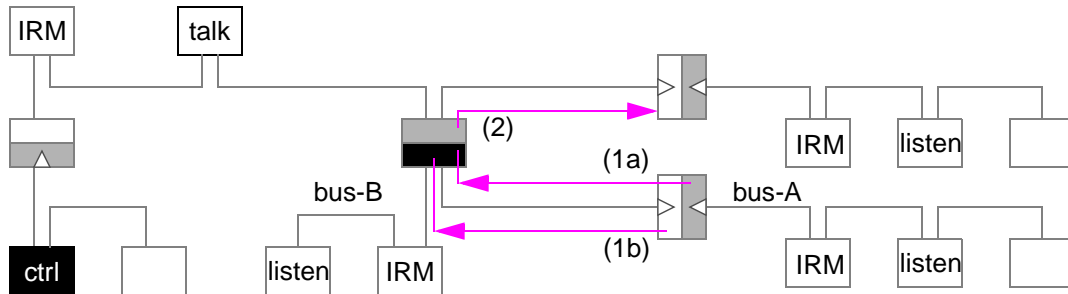


Figure 14—Talker-bus net-refresh recovery (glossary see 2.4.2)

5.5.1.1 Connection message formats

A managed connection starts with acquisition messages sent to the listener-bus controller-path portals and forwarded towards the talker. These messages have a common format, as illustrated in the left portion of figure 15. The connection confirmation returns additional parameters, as illustrated in the right portion of figure 15.

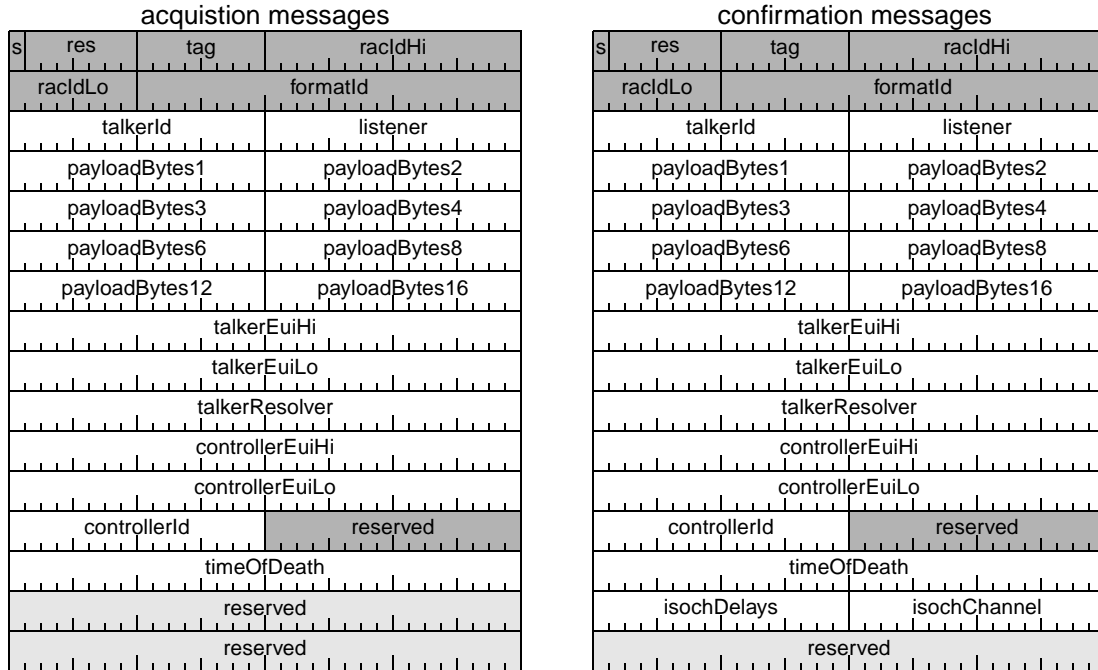


Figure 15—Acquisition and confirmation message formats

5.5.2 Large isochronous payloads

TBD—The format of an isochLost packet has not been defined.

Larger than allocated isochronous payloads are discarded (rather than forwarded) and replaced with an isochLost isochronous packet. The intent is to quickly communicate the resource inconsistencies to pilot proxies, allowing them to correct their currently inconsistent isochronous resource allocations.

5.5.3 Net refresh

TBD—Describe the following in more detail:

Portal-managed isochronous resources are released, if not re-established in T_{nr} .

The intent is to recovery from missing controllers or unrecoverable isochronous-connection message errors.

6. Bridge state

6.1 Isochronous proxy state

6.1.1 Listener proxy state

The listener proxy is colocated on the listener’s delta portal and maintains sufficient state to maintain the integrity of the listener’s attachment, as illustrated in figure 16. Long term storage of the *listenerID* is recommended but not mandated, since this field can be derived (using a bus-local ARP) before each listener-node access.

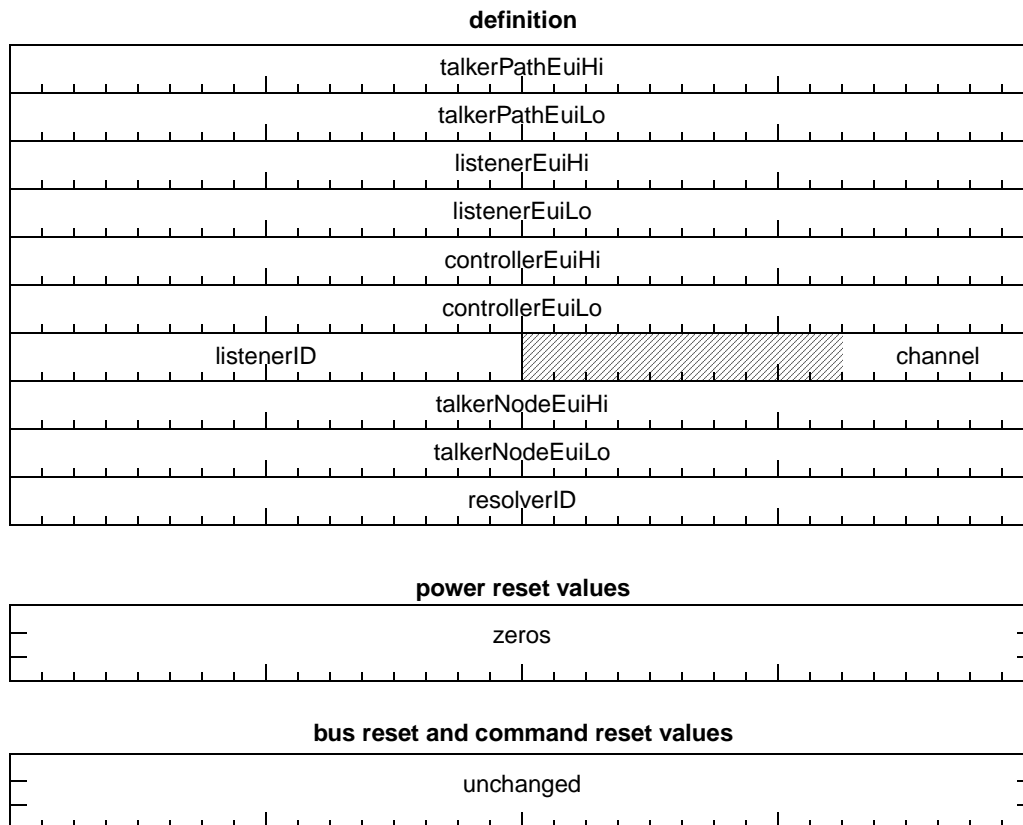


Figure 16—Listener proxy state

The 32-bit *talkerPathEuiHi* and *talkerPathEuiLo* values are concatenated to produce a 64-bit *talkerPathEui* value. When nonzero, this value shall be the talker-path portal’s EUI-64 identifier.

The 32-bit *listenerEuiHi* and *listenerEuiLo* values are concatenated to produce a 64-bit *listenerEui* value. When nonzero, this value shall be the listener node’s EUI-64 identifier.

The 32-bit *controllerEuiHi* and *controllerEuiLo* values are concatenated to produce a 64-bit *controllerEui* value. When nonzero, this value shall be the controller node’s EUI-64 identifier.

When non-null, the 16-bit *listenerID* value equals the *nodeId* of the local-bus listener node. The 6-bit *channel* number specifies the channel attached to the listener.

The 32-bit *talkerNodeEuiHi* and *talkerEuiLo* values are concatenated to produce a 64-bit *talkerNodeEui* value. The 32-bit *resolverID* value has one of two forms. For managed connections, this is the talker’s

oPCR. For unmanaged connections, this is a unique-to-the-talker stream identifier. The concatenation of *talkerEui* and *resolverID* values forms a net-unique 96-bit stream identifier.

6.1.2 Talker proxy state

NOTE—The talker proxy does not identify controllers; the information is efficiently colocated in the listener proxy.

The listener proxy is colocated on the talker’s delta portal and maintains sufficient state to maintain the integrity of the talker’s attachment, as illustrated in figure 17. Long term storage of the *talker* is recommended but not mandated, since this field can be derived (using a bus-local ARP) before each talker-node access.

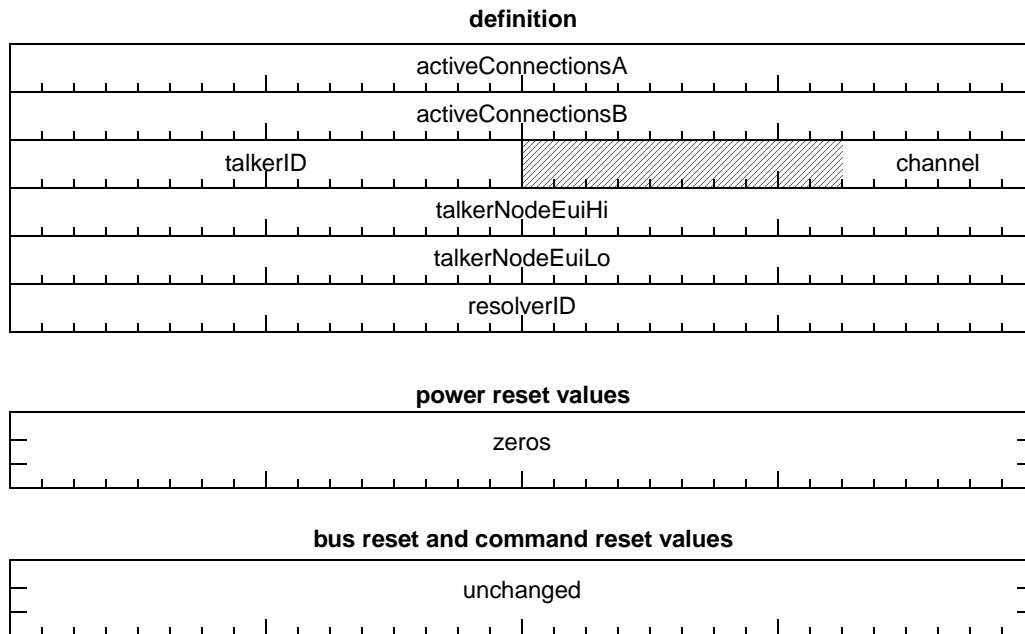


Figure 17—Talker proxy state

The 32-bit *activeConnectionsA* and *activeConnectionsB* values are concatenated to produce a 64-bit *activeConnections* value. These bits identify which of the 63 possible local-bus listener/pilot proxies are actively attached. The primary use of these bits is to determine when the last listening proxies disappears, allowing the release of this talker-proxy resource.

When non-null, the 16-bit *talkerId* value equals the *nodeId* of the local-bus talker node. The 6-bit *channel* number specifies the channel attached the talker.

The 32-bit *talkerNodeEuiHi*, 32-bit *talkerEuiLo*, and *resolverID* values form a net-unique 96-bit stream identifier, as defined in 6.1.2.

6.1.3 Pilot proxy state

NOTE—The pilot proxy does not identify controllers; the information is efficiently colocated in the listener proxy.

The listener proxy is colocated on the talker’s delta portal and maintains sufficient state to maintain the integrity of the talker’s attachment, as illustrated in figure 18. Long term storage of the *talkerPathID* is recommended but not mandated, since this field can be derived (using a bus-local ARP) before each talking prox access.

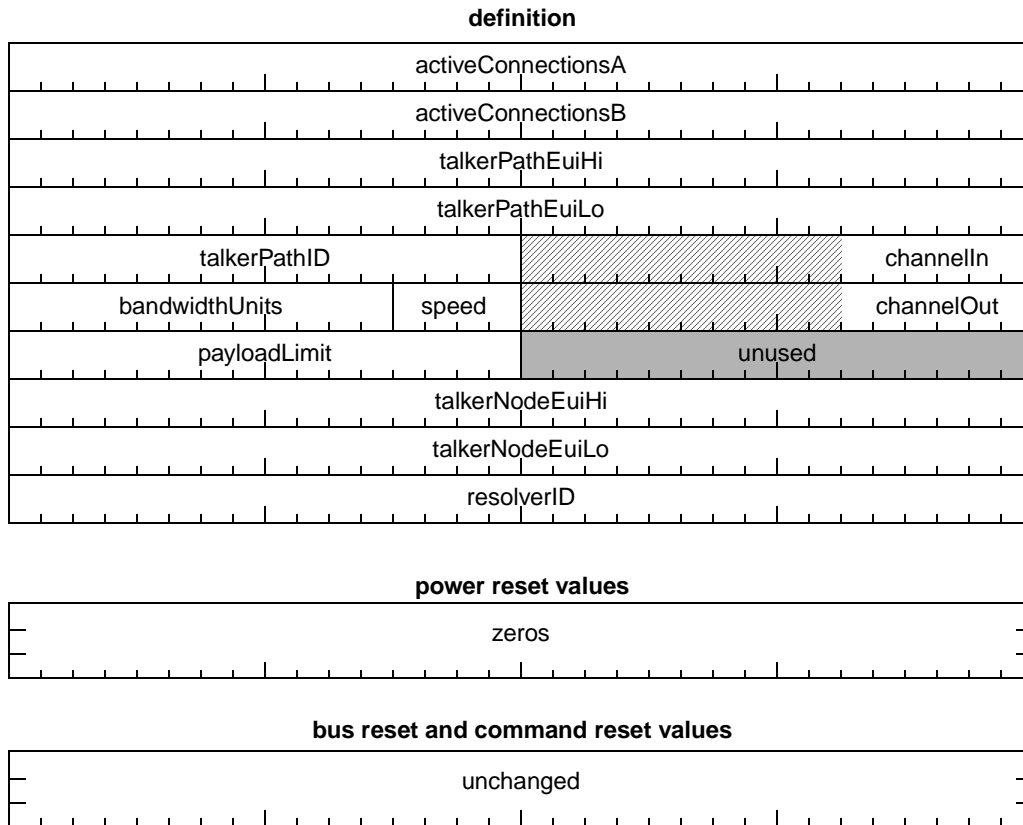


Figure 18—Pilot proxy state

The 32-bit *activeConnectionsA* and *activeConnectionsB* values are concatenated to produce a 64-bit *activeConnections* value. These bits identify which of the 63 possible local-bus listener/pilot proxies are actively attached. The primary use of these bits is to determine when the last listening proxies disappears, allowing the release of pilot-proxy resources.

The 32-bit *talkerPathHi* and *talkerPathLo* values are concatenated to produce a 64-bit *talkerPathEui* value, which identifies the talking proxy. When nonzero, this shall be the talking proxy’s EUI-64 identifier. The talking portal is defined to be the delta portal or the talker-path portal, if the proxy’s talker-side connects to a talker or nontalker bus respectively. When non-null, the 16-bit *talkerPathId* value equals the nodeId of the talking proxy.

The 6-bit *channelIn* value specifies the channel number on the talker-side portal (other Surreal interconnects may mandate the hash-shaded extension).

The 12-bit *bandwidthUnits* value specifies the number of bandwidth units acquired on the listener-side bus. The 4-bit *speed* field indicates the speed to be used when transmitting isochronous packets on the listener-side bus, as specified in table 1.

Table 1—Isochronous *speed* values

Value	Name	Description
0	S100	Forward at S100
1	S200	Forward at S200
2	S400	Forward at S400
3-15	—	Reserved

The 6-bit *channelOut* value specifies the channel number on the listener-side portal (other Surreal interconnects may mandate the hash-shaded extension).

The 16-bit *payloadLimit* field is an upper bound on the `data_length` in isochronous packets.

The 32-bit *talkerNodeEuiHi*, 32-bit *talkerEuiLo*, and *resolverID* values form a net-unique 96-bit stream identifier, as defined in 6.1.2.

