

**BR074R00 (submitted for p1394.1 committee vote):
Bridge aware node requirements**

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

December 1, 1999

**This contribution is one of several, presented for review and incorporation.
An overall contribution, which provides an overall context for this and other contributions, is presented in BR047R10.**

The requirements for a bridge-aware node are proposed, notes indicate what remains to be done.

Annex B

(informative)

Bridge-aware node requirements

NOTE—Early bus bridge implementations (see D.2) may subset bus-bridge functionality (see annexxx) so that implementations may progress at a faster pace than this standard. Although it is acceptable to replace these bridges as this standard progresses, for many it is unacceptable to revise bridge-aware node designs. This annex is an attempt to identify the bridge-aware device requirements, so these can be quickly stabilized.

B.1 Bridge aware node requirements

Asynchronous bridge-aware nodes have a small set of special requirements, listed in the remainder of this subannex. The notes that follow each listed item indicate when work remains to be done.

B.1.1 Bridge-aware controllers

Bridge aware AV/C controllers are required to handle quarantine processing on behalf of legacy devices. This implies the following responsibilities:

- 1) Detection. When a response-frame timeout occurs, the controller shall check and (if necessary) remove the device's quarantine, which is located in the device's controller-path portal (see 1.10.3).
Note—The quarantine-clear offset, tcode, and format has not been specified.
- 2) Reaction. If the device was found to be quarantined, the controller shall purge the device's EUI-to-nodeID translations before invoking additional AV/C commands.
Note—AV/C devices should support translation purges without mandate the use of bus reset.
- 3) Recovery. The controller shall recover from the lost response-frame by generating a sequence of device-dependent AV/C commands.

B.1.2 Bridge-aware device quarantines

Bridge aware devices, which are not explicitly controlled by a bridge-aware controller, are required to handle their own quarantine processing, as follows:

- 1) Detection. When acting as a requester, the bridge-aware device shall recognize a {*resp_data_error*, *ext_quarantined*} error-reporting response.
- 2) Release. After a quarantine detection, the device is responsible for removing its quarantine, which is located in the requester's responder-path portal (see 1.10.2).
Note—The quarantine-clear offset, tcode, and format has not been specified.
- 3) Reaction. After removing a new quarantine, a node is responsible for purging EUI-to-nodeID translations.
Note—The sequence number for identifying new quarantine conditions, presumably based on a sequence number, has not been specified.

B.1.3 Remote node discovery

Bridge aware devices and controllers shall be capable of extending their discovery operations to include searches of remotely located nodes, as follows:

- 1) Bus presence. The presence of buses involves reading a local portal's routing table.
Note—The location of the routing table has not been specified.
Note—The format of the routing table may depend on routing simplifications proposed in F.4.
- 2) Node presence. The presence of nodes involves a 512-byte read from a destination-bus portal.
Note—The format of the FinalRead transaction has not been finalized.
Note—The location of the EUI_PRESENCE table has not been specified.
Note—The behavior of these resources, in the presence of transient net-refresh conditions has not been specified.

B.1.4 Remote node timeouts

Bridge aware devices and controllers shall be capable of extending their split-response timeouts when accessing remotely located responders, as follows:

- 1) Local. The T_{st} (subaction timeout) value limits request-to-response time for local-node accesses. (this is already defined by p1394a).
- 2) Global. The T_{rt} (response timeout) limits the time spent waiting for the expected response.
Note—The format and location of the REMOTE_TIMEOUT register has not been finalized.
Note—The initial value for the REMOTE_TIMEOUT register, and the method for determining this value during net refresh, has not been specified.

B.1.5 Extended responder behavior

Bridge aware devices and controllers shall (if so specified) be capable of providing extended-response packets (see 1.7.2), as follows:

- 1) *offset_low*. The 16 least-significant bits of the 48-bit offset field are returned in a response.
- 2) *responder_ID*. The virtual node_ID of the actual responder is returned in the response.
Note—The need for the redundant responder_ID value has not been reviewed.
- 3) *sCode*. A distinct sCode bit shall be used to distinguish between bridge-aware extended-response and legacy-device basic-response formats.
Note—The value of the bridge-aware responder's scode value has not been finalized.

B.1.6 Packet-size constraints

Bridge aware requesters shall be capable of constraining the size of transactions that are generated to remotely located nodes, independent of the transmission speeds supported by the local bus, as described in 1.7.2 (this may be dynamically determined, by attempting larger size transactions until an error occurs).

Note—The value of the responder's "ext_truncated" scode value has not been finalized.

B.1.7 Local node addressing

The form of local-bus addressing (see xx) must be fully defined.

Note—The elimination of virtualIDs has not been reviewed or approved by the working group.

Note—The two options for addressing local nodes need to be evaluated in more detail, and one selected.

B.1.8 Auto-quarantining

When isolated from portals, the node shall discard virtual-addressed packets and cached EUI-to-virtualID translation.

B.1.9 Event recognition

Nodes should be capable of recognizing broadcast events, such as net_changed or iso_changed. Bridge-aware controllers shall be capable of recognizing broadcast iso_changed events.

Note—The format and behavior of broadcast events has not been finalized.

B.1.10 Isochronous management

Bridge-aware controllers have special requirements associated with isochronous-connection management responsibilities, as follows:

- 1) Setup messages. Connection messages shall be used to establish the connection.
Note—The format of connection messages has not been finalized.
Note—Routing of connection messages to the listener's talker-path portal has not been finalized.
- 2) Nonpersistent connections. We have previously considered persistent isochronous connections.
Note—The initial proponent of persistent isochronous connections has no longer favors this approach, but the working-group has not commented on this reflector traffic.
- 3) Bursty traffic. A credit-based system may be used to improve the data-transfer rate of bandwidth-limiting interfaces, such as wireless.
Note—Although the working group approved this idea (in concept), the details in this document have not been reviewed by the working group or the initial proponent (Philips).
- 4) Unmanaged connection. Unmanaged isochronous connections may have to be supported, as not all isochronous devices support plug-control registers.
Note—The proposal for managed isochronous connections, contained in this draft, has not been reviewed by the working group.

B.2 Extended IRM designs

Nodes that desire to be an IRM, in the presence of bus-bridge nodes, have special additional requirements and interfaces:

- 1) Message interface. A swap transactions provides resource allocation arguments and returns results.
- 2) Event trigger. A change of the IRM values is distributed to other portals, which propagate messages to the listeners, which deliver the messages to their controllers.
Note—The format and behavior of bandwidth-changed message has not been finalized.
- 3) Clock synch. The IRM node shall be capable of adjusting its clock to the reference provided by the local alpha portal.
Note—The format and behavior of clock-synchronization packets has not been finalized.

B.3 Open topics

The following topics are active and could significantly influence bus bridge design:

- 1) Net reset/refresh. Detailed definitions of net refresh operations could affect the handling of transactions (and possibly error-status codes) for in-progress transactions.
Note—The net refresh protocols have not been approved by the working group.
Note—Some working-group members are believed to prefer another net refresh protocol, but that is hard to evaluate as it has not progressed to the detailed specification phase.
- 2) Virtual IDs. Previous proposals have assumed the presence of virtual IDs and do not change busID addresses after each bus reset.
Note—The elimination of busIDs has not been approved by the working group.
Note—A constrained behavior of bridge-aware nodes, after bus reset, would have to be defined if the initially attractive virtualID concept were to be maintained.
- 3) Extended headers. Additional parameters (i.e. time-of-death) could be included in packets. This could significantly affect the design of adaptive (or adaptive-compatible) bridge-aware requesters.
Note—The concept of time-of-death adaptive delay-sensing protocols has not been reviewed by the working group.
- 4) Efficient quarantines. Efficient quarantines recovery assumes the presence of a sequence identifier, to distinctively label each quarantine action.
Note—The format and initialization behavior for this sequence number have not been defined.