ETSI TS 101 493-4 V1.1.1 (2001-07)

Technical Specification

Broadband Radio Access Networks (BRAN);
HIPERLAN Type 2;
Packet based Convergence Layer;
Part 4: IEEE 1394 Bridge Specific Functions sub-layer for restricted topology

Reference

DTS/BRAN-0024004-4

Keywords

broadband, radio, HIPERLAN, access

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at http://www.etsi.org/tb/status/

If you find errors in the present document, send your comment to:
editor@etsi.fr

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Project Broadband Radio Access Networks (BRAN).

The present document is part 4 of a multi-part deliverable covering the Packet based Convergence Layer of HIPERLAN/2, as identified below:

Part 1: "Common Part";

Part 2: "Ethernet Service Specific Convergence Sublayer (SSCS)";

Part 3: "IEEE 1394 Service Specific Convergence Sublayer (SSCS)";

Part 4: "IEEE 1394 Bridge Specific Functions sub-layer for restricted topology";

Part 5: "IEEE 1394 Bridge Specific Functions sub-layer for unrestricted topology".

Part 1, Common Part [4], describes the functionality for adapting variable length packets/frames to the fixed size used in the Data Link Control (DLC) layer. Further parts, each defining a Service Specific Convergence Sublayer (SSCS), describe the functionality required to support a certain protocol, e.g. IEEE 1394 or Ethernet. The 1394 SSCSs all use the services of the Common Part [4] and the DLC [5]. It is envisioned that several, independent, service specific parts will be defined in the future as market requirements develop.

1 Scope

The present document is applicable to HIgh PErformance Radio Local Area Network Type 2 (HIPERLAN/2) equipment supporting interworking with 1394 buses. It defines the functionality required for interworking HIPERLAN/2 with IEEE 1394 buses and defines how to transfer IEEE 1394 packets over the radio interface.

The present document specifies the 1394 leaf bus bridge functions required to transfer IEEE 1394 traffic between IEEE 1394 devices over HIPERLAN/2 wireless bridge devices. It does not address the requirements and technical characteristics for wired network interfaces at the HIPERLAN/2 device.

The present document uses the services provided by the Packet based convergence layer part 1 (common part, TS 101 493-1 [4]), part 3 (IEEE 1394 Service Specific Convergence Sub-layer, TS 101 493-3 [3]), and the data link control layer of HIPERLAN/2 (TS 101 761-1 [5]).

The present document does not address the requirements and technical characteristics for conformance testing. These are covered by separate documents.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] IEEE Std 1394-1995: "IEEE Standard for a High Performance Serial Bus".

[2] IEEE Std 1394a-2000: "IEEE Standard for a High Performance Serial Bus-Amendment 1".

[3] ETSI TS 101 493-3: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Packet based convergence layer; Part 3: IEEE 1394 Service Specific Convergence Sublayer (SSCS)".

[4] ETSI TS 101 493-1: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Packet based Convergence Layer; Part 1: Common Part".

[5] ETSI TS 101 761-1: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions".

[6] ETSI Technical Working Procedures ("http://www.etsi.org/directives/Directives.htm").

[7] IEEE Std 1394.1: "Draft Standard for High Performance Serial Bus Bridges".

NOTE: IEEE P1394.1 is an IEEE authorized standard project that was not approved by the IEEE-SA Standards Board at the time this document was published. The proposed draft may change as a result of comments received during the ballot process and its approval as a standard. Upon approval, the draft will be published as an IEEE standard. IEEE drafts and approved standards are available from the Institute of Electrical and Electronics Engineers, Inc. (http://www.ieee.org/).

[8] IEC 61883: "Consumer audio/video equipment - Digital interface".

[9] ISO/IEC 13213 (1994) (ANSI/IEEE Std 1212, 1994): "Information technology - Microprocessor systems - Control and Status Registers (CSR) Architecture for microcomputer buses".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

1394 SSCS: Refers to a wireless bus as specified by ETSI TS 101 493-3 [3].

adjustable cycle master: cycle master that is capable of adjusting cycle periods by processing cycle adjustment packets, as described in IEEE Std 1394.1

alpha portal: singular portal on a bus that leads to the prime portal

NOTE 1: There is no alpha portal on the bus where the prime portal is on.

bridge: Serial Bus node capable of connecting two or more buses into a Serial Bus net

NOTE 2: A Serial Bus bridge implements two portals and forwards asynchronous and isochronous packets according to routing rules.

bus_ID: 10-bit identifier that uniquely identifies a bus within a Serial Bus net

bus: group of Serial Bus nodes interconnected by the same PHY medium and mutually addressable by packets with a *destination_bus_ID* field of $3FF_{16}$

channel: 6-bit number that uniquely identifies an isochronous or asynchronous stream on a local bus

CSR architecture: Control and Status Registers (CSR) Architecture for microcomputer buses, as defined by the ISO/IEC 13213 (1994) (ANSI/IEEE Std 1212, 1994 edition)

cycle master: singular node on a bus that generates the periodic cycle start packet 8 000 times a second

entry portal: bridge portal is called an entry portal when it eavesdrops and assumes responsibility for the disposition of a bridge-bound request, response or stream subaction

NOTE 3: Common actions are to forward the subaction to the bridge's other portal (which acts as the exit portal) or to echo the subaction on the local bus, but the entry portal may also create and transmit a response packet if errors are detected.

exit portal: bridge portal is called an exit portal when it transmits bus-bound request, response or stream subactions forwarded by the entry portal

NOTE 4: The same portal may be both the entry and exit portal for a subaction; this is the case when a transaction request or response is echoed to the local bus.

global node ID: 16-bit number that uniquely identifies a node in a net. It consists of a 10-bit bus ID and a 6 bit virtual ID

IEC 61883: Refers to IEC 61883 [8].

IEEE 1394: Refers to the IEEE Std 1394-1995 High Performance Serial Bus standard [1] as amended by IEEE Std 1394a-2000 [2], and supplemented with IEC 61883 [8].

IEEE 1394.1: Refers to the IEEE Std 1394.1 High Performance Serial Bus Bridges draft standard [7].

HIPERLAN/2: HIgh PErformance Radio Local Area Network Type 2, a short-range wireless LAN providing broadband local access standardized by ETSI Project BRAN

HL2 1394 bridge layer: Refers to the HIPERLAN 2 IEEE 1394 bridge functions sub-layer, as defined in the present document.

HL2 Bus: virtual 1394 bus that is realized on a HL2 wireless network

isochronous period: period that begins when a cycle start packet is sent and ends when a subaction gap is detected

NOTE 5: During an isochronous period, only isochronous subactions may occur. An isochronous period begins, on average, every 125 μs.

isochronous resource manager: node that is the depository of stream resource information on a bus

NOTE 6: On a 1394 bus, its roles are specified in IEEE 1394. On a HL2 bus, its roles are specified in the 1394 SSCS [3].

isochronous subaction: within the isochronous period, either a concatenated packet or a packet and the gap that precedes it

listener: application at a node that receives a stream packet

local node: Serial Bus node is local with respect to another node if they are both connected to the same bus

NOTE 7: This is true whether the bus does not yet have a unique *bus_ID* and is addressable only as the local bus, 0x3FF, or if he bus has been enumerated and assigned a *bus_ID*.

local node ID: 16-bit number that identifies a node on a local bus

NOTE 8: It consists of the 10-bit local bus ID, i.e., $3FF_{16}$, and the 6-bit physical ID.

net: collection of Serial Buses, joined by Serial Bus bridges

NOTE 9: Each bus within the net is uniquely identified by its *bus_ID*.

net cycle master: singular node in a net that acts as the origin of the clock synchronization throughout the net

node: device that may be addressed independently of other nodes

NOTE 10: A minimal node consists of only a PHY without an enabled link. If the link and other layers are present and enabled, they are considered part of the node.

physical ID: 6-bit number assigned to each node by the self-identification process that follows a bus reset (see IEEE 1394)

portal: part of a Serial Bus bridge that resides on a local bus and uniquely addressable in a net

NOTE 11: Each portal presents a full set of Serial Bus CSR's, as defined in IEEE 1394 and in this document, to the connected bus. They may be multiple PHY ports for each portal.

prime portal: singular portal within a net that manages assignment of bus IDs and their distribution

remote node: Serial Bus node is remote with respect to another node if the nodes are connected to buses that have different *bus_ID*' s or if one or more Serial Bus bridges lie on the path between the two nodes

unrestricted 1394 bridge: Serial Bus bridge capable of supporting any net topology up to 1 022 buses

virtual ID: 6-bit number that is assigned by a bridge portal to each node present on the portal's local bus

NOTE 12: Unlike physical IDs, virtual IDs are stable across bus resets. All the bridge portals on a bus share the same mapping from 6-bit physical ID to virtual ID.

wireless 1394: Refers to a HL2 Bus as specified in [3].

wireless bridge: bridge which a least one of its portals resides on a wireless network

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

BRAN	Broadband Radio Access Networks (Project)
CIP	Common Isochronous Packet
CL	Convergence Layer
CSR	Control and Status Register
ETSI	European Telecommunications Standards Institute
HL2	HIPERLAN/2
IEEE	Institute of Electrical and Electronics Engineers
IRM	Isochronous Resource Manager
self-ID	Self Identify
SSCS	Service Specific Convergence Sub-layer
SPH	Source Packet Header

4 Overview

4.1 The HL2 1394 Bridge layer

The HL2 1394 bridge layer is the part 4 of the Packet based Convergence Layer. It resides on top of the 1394 SSCS, as shown in figure 1.

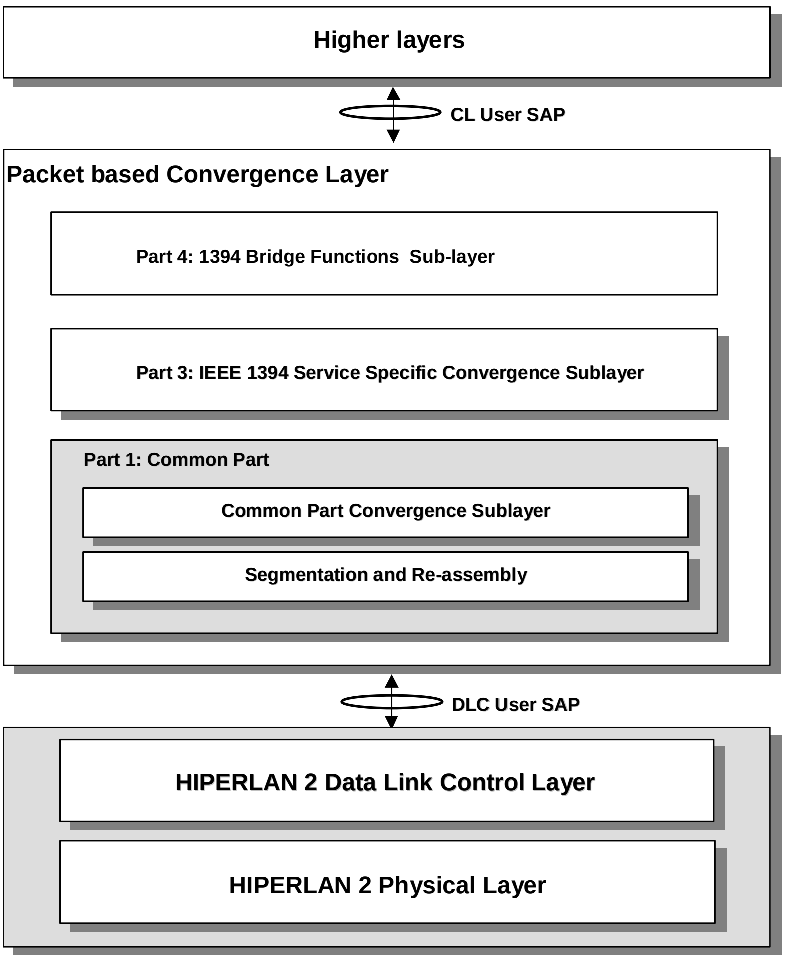

Figure 1: IEEE 1394 packet based convergence layer

The net topology when applied to HIPERLAN/2 is shown in figure 2. The HIPERLAN/2 network appears as a wired-1394 bus for all the attached nodes. This bus is called a HIPERLAN2 Bus, which is often abbreviated as HL2 Bus. Regarding the HL2 Bus, all wireless nodes are peers (there is no particular base station).

The aim of the HL2 1394 bridge layer is to interconnect a IEEE 1394 bus and a HL2 bus. The HL2 1394 bridge layer is present in wireless bridge devices, whose architecture is described in figure 3.

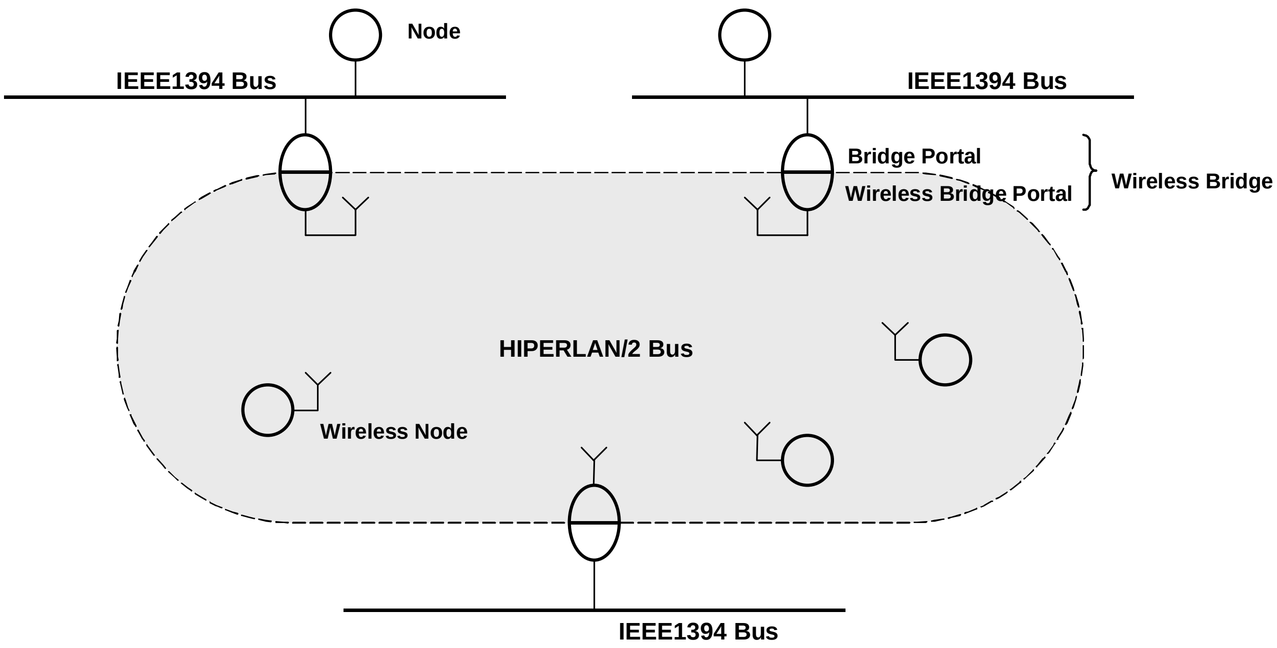

Figure 2: HL2 Bus connecting bridge and non-bridge 1394 devices

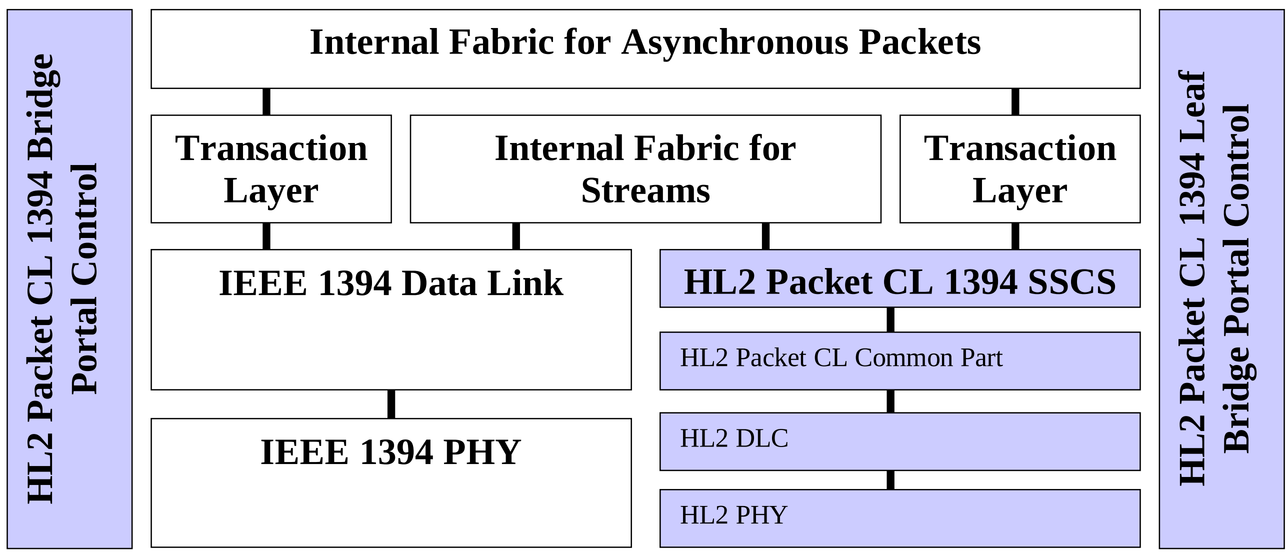

Figure 3: 1394 Wireless Bridge Model

4.2 1394 bus bridge model

The purpose of the 1394 bus bridge is to solve the following issues in a single bus network.

A 1394 bus bridge provides:

- expansion of the number of the nodes supported in a net. An IEEE 1394 bus can support up to 63 nodes. A net that contains bridges can support a significantly larger number of nodes;
- increase of usable bandwidth in a net. Bandwidth is a shared resource on a single bus. Bandwidth needs to be allocated only on the route between the talker and the listener, and it does not require any resources in the other part of the net;
- improvement of stability of a net associated with bus resets. Bridge isolate local bus resets from the rest of a net;
- improvement of arbitration efficiency in a local bus by limiting the number of nodes (hops) on it.

In addition, a 1394 bridge to a HL2 bus removes physical limitations of cables, and provides mobility and ease of installation.

A 1394 bus bridge as many other bridges operates at the Data Link layer level: it filters out traffic so that only the relevant traffic is forwarded from one bus to another. This allows an efficient bandwidth usage on both sides of the bridge. Moreover a 1394 bus bridge also filters out bus reset, to avoid reset storms on the network.

A bridge portal is a node with a dedicated EUI-64 address and its own address space on the bus to which it is connected. A bridge portal is thus compliant with IEEE 1394. It therefore responds to serial bus read, write and lock transactions as described in IEEE 1394. It recognizes new primary packet fields as described in IEEE Std 1394.1 [7]. A bridge portal also forwards asynchronous and isochronous packets to its co-portal via its internal fabric according to its routing rules as described further in the present document.

The internal fabric allows bus packets transfer from one portal to its co-portal. The internal fabric also contains facilities to transmit the cycle clock from one portal to its co-portal. The details of the internal fabric are out of the scope of the present document.

4.3 1394 Leaf-bus bridge model

The present document focuses on the definition of a leaf bus bridge, which is a restricted topology bridge. Its functions are significantly simplified compared to that of an unrestricted 1394 bridge by imposing a constraint on the network topology. The HL2 bus is defined as a branch bus to which only leaf-bus bridges can be connected as shown in figure 4 (limited to the maximum of two bus hops).

Part 5 of the Packet Based Convergence Layer (see Bibliography) will address the HL2 unrestricted topology 1394 bridge in a backward compatibility manner: leaf-bus bridges function as bus bridges in a self-sufficient manner in the absence of HL2 unrestricted 1394 bridges in the network. However, in the presence of HL2 unrestricted 1394 bridges, the leaf-bus bridge performs its bridge functions in a manner that some of its bridge functions are taken over by one or more of the HL2 unrestricted 1394 bridges in the network. HL2 unrestricted 1394 bridge mechanisms are out of the scope of the present document, but the leaf-bus bridge behaviour in presence of HL2 unrestricted 1394 bridge is described in the present document.

A leaf bus is defined as a 1394 bus connected to a HL2 bus via a leaf-bus bridge as defined by the present document. Only one leaf-bus bridge is allowed to operate on a leaf bus. If several leaf-bus bridges are connected to a leaf bus, a contention resolution algorithm takes place so that only one leaf-bus bridge keeps its bridge function in operation, as described in clause 12.2. If a leaf-bus bridge and a HL2 unrestricted 1394 bridge are connected to the same 1394 bus, the leaf-bus bridge will disable its bridge functions as described in clause 12.2.

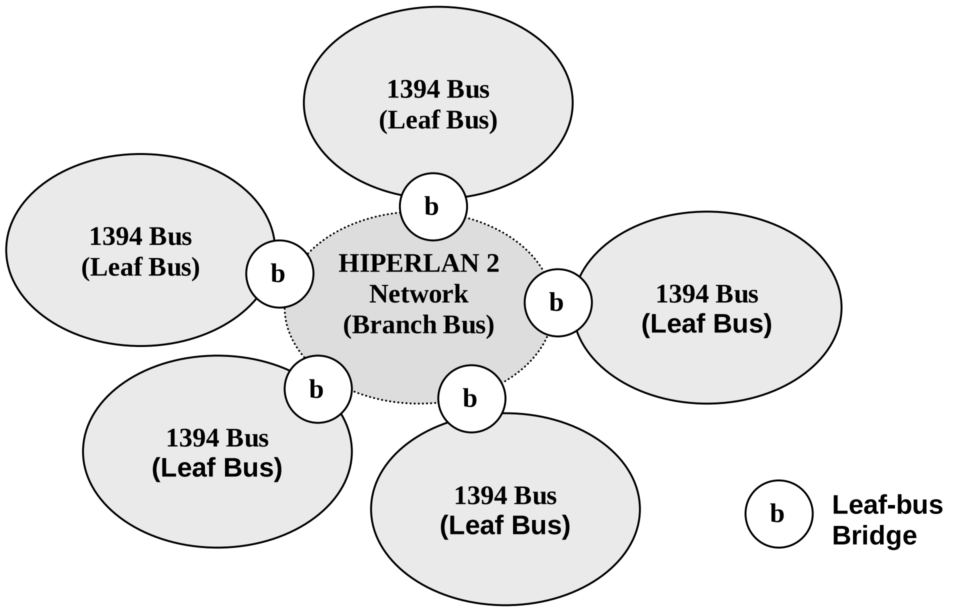

Figure 4: An example of restricted network topology

4.4 Bridged network model (informative)

Operation in a 1394 bridged network introduces some specificity regarding the operation in a single 1394 bus, which is also described in IEEE Std 1394.1 [7].

4.4.1 Global node IDs

In a 1394 bus physical IDs are allocated after each bus reset. Therefore, applications need to revalidate their EUI-64/1394 physical ID mapping after every bus reset. In a bridged environment, bus resets do not propagate across bridges. A global node ID has been introduced in IEEE Std 1394.1 [7] to minimize bus reset recovery actions. A global node ID is allocated by a bridge portal and is intended to be stable across bus resets. A device will thus keep the same global node ID whatever number of bus resets that occur on the local bus. A bus bridge forwarding a packet from the local bus (a subaction generated from the local bus) to a remote bus translates the source ID into the corresponding global node ID. A bus bridge delivering a packet from a remote bus to the local bus (i.e. a subaction packet whose destination is the local bus) translates the global node ID contained in the destination ID field into the corresponding physical node ID.

4.4.2 Bus Reset/Net Reset

A bus reset, as defined in IEEE 1394 occurs on a bus when a device is inserted or removed from the bus, or when a node is turned on or off. A bus reset may also be initiated by a device without any device insertion or removal. Bus resets are not propagated by a bridge to remote buses.

A net reset occurs when a bridge portal is inserted or removed from the network. It triggers a bus reset with specific self_id packet codes as defined in clause 5.4 that indicate that a bridge state has changed in the network. Net resets are propagated by bus bridges. When a device sees a net reset it starts a quarantine period during which no remote transaction shall take place. The quarantine period is terminated by a bridge portal. After every net reset, a device is required to revalidate the global node IDs in use.

4.4.3 Remote time-out

The split timeout on a single bus cannot be used for remote transactions. The split timeout in a bridged environment depends on the number of bridges involved in a remote transaction. Bridge aware devices can determine the *remote_timeout* in a bridged environment by sending a *bridge management message* to a remote node. In the response to this packet the requester will receive the remote time out value to be used for the remote transaction.

When HL2 is represented as a 1394 bus, a HL2 wireless bridge is used to wirelessly interconnect 1394 bridge aware nodes. When two bridge aware nodes exchange asynchronous packets over HL2, they cross one or two bridges. Therefore they have to use a remote timeout that takes into account the two wireless bridges as well as the HL2 transmission time.

4.4.4 Stream operations

While the transmission of asynchronous packets is mandatory for HL2 compliant bus bridges, the routing of streams over a HL2 bus is optional and vendor dependent. The number of simultaneously supported streams and the bandwidth provision of isochronous streams are also vendor dependent.

Unlike asynchronous packets, which are routed automatically based on their (destination) global node IDs, isochronous and asynchronous stream connections have to be set up by a controller. This "bridge aware" controller uses *bridge management messages* to establish, overlay or break stream connections. It is in the responsibility of a bridge portal to allocate, reallocate (in case of a local bus reset) and deallocate isochronous resources in accordance to the characteristics of the stream to be transmitted. It is essential for isochronous stream connections that the complete transmission system acts as a constant delay system. Bus bridges, which provide isochronous connections, offer mechanisms for time-stamping and buffering in order to achieve these requirements.

5 Bridge facilities

This clause describes the facilities that a Serial Bus bridge portal shall implement in order to interoperate with other bridges and bridge-aware devices. The facilities are configuration ROM entries, control and status registers or CSRs. A Serial Bus bridge portal shall be implemented as a unit architecture within a Serial Bus node.

5.1 Bridge portal configuration ROM

5.1.1 Leaf bus bridge portal configuration ROM

A bridge portal on a leaf bus shall implement a configuration ROM defined by IEEE 1394.

5.1.1.1 Bus information block

The leaf bus bridge portal bus information block shall be as specified in IEEE Std 1394.1 [7].

5.1.1.2 Bus_Dependent_Info entry

The leaf bus bridge portal root directory shall contain the Bus_Dependent_info entry as described in IEEE Std 1394.1 [7].

5.1.1.3 Bridge_Capabilities entry

30_{16}	r	streams	isochronous_delay

Figure 5: Leaf bus Bridge_Capabilities entry

The Bus_Dependent_Info directory of the leaf bus bridge portal shall contain the Bridge_Capabilities entry as defined in IEEE Std 1394.1 [7], and repeated here for convenience that specifies the bridge capabilities.

The *streams* field shall specify the number of streams that the bridge can transfer between the portals as specified in IEEE Std 1394.1 [7].

The *isochronous_delay* field shall specify the constant delay that an isochronous stream experiences when going through the bridge from the leaf bus to the HL2 bus. It is described in units of cycles (125 microseconds). If the bridge does not support isochronous streams, the *isochronous_delay* field shall report zero.

The *isochronous_delay* in the Bridge_Capabilities entry on the leaf bus bridge portal shall take the maximum HL2 isochronous transmission delay into account. This means that the wired bridge portal shall report the sum of the maximum HL2 isochronous transmission delay (49 cycles, or 6,125 ms) and an implementation dependent constant value that reflects its internal transmission delay, i.e. the time required to transfer isochronous packets from the wired IEEE 1394 bus to the 1394 SSCS [3].

5.1.1.4 HL2 Unit directory entry

The leaf bus bridge portal root directory shall contain a unit directory entry for a HL2 bus, as illustrated in figure 6.

$D1_{16}$	indirect_offset

Figure 6: Leaf bus HL2 unit directory entry

The *indirect_offset* field identifies the location of the HL2 unit directory within the configuration ROM. It shall specify the number of quadlets from the address of HL2 unit directory entry to the address of the HL2 unit directory.

The leaf bus bridge portal HL2 unit directory is illustrated in figure 7.

Length	CRC
12_{16}	specifier_ID
13_{16}	version

Figure 7: Leaf bus HL2 unit Directory

The *Length* field shall contain the length of the HL2 unit directory. The *specifier_ID* shall be set to $0180c2_{16}$, and the *version* shall be set to 000200_{16} to identify the HL2 unit directory.

The presence of the HL2 unit directory in a wired 1394 configuration ROM as described in the present document identifies a HL2 1394 bus bridge.

5.1.1.5 Bridge_Revision entry

The leaf bus bridge portal HL2 unit directory shall contain the Bridge_Revision entry, as illustrated in figure 8.

38_{16}	reserved	phase	revision

Figure 8: Leaf bus Bridge_Revision entry

The *phase* and *revision* fields shall be coded with major (*m*) and technical (*a*) version numbers respectively of the present document (*m* and *a* are defined in annex B of [6]).

For the restricted bus bridge as specified in the present document, the value of the *phase* field shall be one.

5.1.2 Branch bus bridge portal configuration ROM

Each bridge portal on the branch bus shall implement a configuration ROM as defined by the 1394 SSCS [3].

5.1.2.1 Bus information block

The branch bus configuration ROM shall contain a bus information block as defined by the 1394 SSCS [3] and repeated in figure 9 for convenience.

The definition and usage of the bus information block shall conform to the 1394 SSCS [3] with the exception of *max_rec* field that is redefined in IEEE Std 1394.1 [7]. Some new fields (that are reserved in IEEE 1394) are defined below.

<table>
<tr><td colspan="3">30₁₆ ("0")</td><td colspan="3">30₁₆ ("0")</td><td colspan="3">30₁₆ ("0")</td><td colspan="3">30₁₆ ("0")</td></tr>
<tr><td colspan="3">capabilities</td><td colspan="3">cycl_clk_acc</td><td>max_rec</td><td>r</td><td>max_ROM</td><td>generation</td><td>b</td><td>link_spd</td></tr>
<tr><td colspan="9">node_vendor_ID</td><td colspan="3">chip_ID_hi</td></tr>
<tr><td colspan="12">chip_ID_lo</td></tr>
</table>

Figure 9: Bus information block format on branch bus

The *capabilities* field (8 bits) consists of individual bit fields, as shown in figure 10, and specifies the bridge portal's interrelated facilities as defined in IEEE Std 1394.1 [7].

irmc	cmc	isc	bmc	pmc	adjustable	reserved

Figure 10: Bus information block capabilities field

5.1.2.2 HL2 Bus_dependent_Info entry

The branch bus bridge portal root directory initial entry shall be a Bus_Dependent_Info entry as illustrated in figure 11, and specified by the 1394 SSCS [3].

$C2_{16}$	indirect_offset

Figure 11: Branch bus HL2 Bus_Dependent_Info entry

The *indirect_offset* field identifies the location of the HL2 Bus_Dependent_Info directory within the HL2 configuration ROM. It shall specify the number of quadlets from the address of HL2 Bus_Dependent_Info entry to the address of the HL2 Bus_Dependent_Info directory.

The branch bus bridge portal HL2 Bus_Dependent_Info directory is illustrated in figure 12.

Length	CRC
12_{16}	specifier_ID
13_{16}	version

Figure 12: Branch bus HL2 Bus_Dependent_Info Directory

The *Length* field shall contain the length of the HL2 Bus_Dependent_Info directory. The *specifier_ID* shall be set to $0180C2_{16}$, and the *version* shall be set to 000200_{16} to identify the HL2 Bus_Dependent_Info directory as specified by the 1394 SSCS. The HL2 Bus_Dependent_Info directory shall contain the entries specified by the 1394 SSCS [3].

It shall contain the additional entries specified in the present document.

5.1.2.3 Bridge_Capabilities entry

The HL2 Bus_Dependent_Info directory of the branch bus bridge portal shall contain the Bridge_Capabilities entry as illustrated in figure 13 that specifies the bridge capabilities.

30_{16}	r	streams	isochronous_delay

Figure 13: Branch bus Bridge_Capabilities entry

The *streams* field shall specify the number of streams that the bridge can transfer between its portals, as specified in IEEE Std 1394.1 [7].

The *isochronous_delay* field shall specify the constant delay that an isochronous stream experiences when going through the bridge from the HL2 bus to the leaf bus. It is described in units of cycles (125 microseconds). The minimum value for the *isochronous_delay* field is two. If the bridge does not support isochronous streams, the *isochronous_delay* field shall report zero. The *isochronous_delay* value is a constant value that reflects its internal transmission delay when a stream is transferred from the HL2 portal to the wired IEEE 1394 portal. This value is implementation dependent.

The *isochronous_delay* in the Bridge_Capabilities entry on the HL2 bus bridge portal shall not include the maximum HL2 isochronous transmission delay, i.e., 49 cycles. This means that the HL2 bridge portal shall report only an implementation dependent constant value (in cycles, minimum of two) that reflects its internal transmission delay, which is the time required to transfer a packet from the HL2 constant delay buffer to the leaf bus (see clause 11.2).

5.1.2.4 Bridge_Revision entry

The branch bus bridge portal HL2 Bus_Dependent_Info directory shall contain the Bridge_Revision entry, as illustrated in figure 14.

38_{16}	reserved	phase	revision

Figure 14: Branch bus Bridge_Revision entry

The *phase* and *revision* fields shall be coded with major (*m*) and technical (*a*) version numbers respectively of the present document (*m* and *a* are defined in annex B of [6]).

For the restricted bus bridge as specified in the present document, the value of the *phase* field shall be one.

5.2 Bridge portal control and status registers (CSRs)

The wired bridge portal's CSRs shall include registers defined by IEEE 1394. The wireless bridge portal's CSRs shall include registers defined by the 1394 SSCS [3]. Both bridge portals shall implement the NET_GENERATION and CLAN_INFO registers as defined in IEEE Std 1394.1. The implementation of the other bridge portal registers defined in IEEE Std 1394.1 [7] is optional.

5.3 Stream routing tables

5.3.1 Stream routing tables

The stream routing tables are used for management of isochronous streams and shall be implemented by a bridge portal whose *imc*, *cmc*, *isc* and *adjustable* bits in Bus information block are set to one and the *streams* field in the bridge_capability entry reports a non-zero value. The stream routing tables are an array of quadlets, and the number of entries shall be equal to the value of the *stream* field in the bus information block. The entries are addressed as STREAM_CONTROL[0] through STREAM_CONTROL[stream-1], inclusive. The both bridge portals of a bridge implement the same number of entries on the both sides, and the corresponding two entries from both portals function as a pair. The CSR offset for the STREAM_CONTROL array shall be as defined in IEEE Std 1394.1 [7].

The STREAM_CONTROL entry format for the leaf bus portal shall be as specified in IEEE Std 1394.1 [7].

The STREAM_CONTROL entry format for the HL2 bus portal shall be as described below.

st	channel	i	reserved	***payload***

Figure 15: STREAM_CONTROL entry format on the branch bus

The *st*, *channel* and *i* fields shall be as specified in IEEE Std 1394.1 [7].

The *payload* field shall indicate the maximum number of bytes to be transmitted within 2 ms HL2 frame (including the 1394 SSCS and CPCS overheads) divided by 16, as defined by the 1394 SSCS [3].

5.4 Packet formats of Self-ID packet zero

On the leaf bus self-ID packets shall be used as defined by IEEE Std 1394.1 [7].

On the branch bus, the bus reset procedure is defined in the 1394 SSCS. The self-ID process takes place in information elements as described in the 1394 SSCS. The *node_type* 2-bit field is defined by the 1394 SSCS [3]. It is a field of the CL_CONTROL/CL_SELF_ID primitives that are used to initiate and indicate a bus reset. It is also a field of the BUS_RESUME information element that is exchanged between devices during a bus reset. On the branch bus, the bridge layer shall put in the *node_type* field the *brdg* values as defined in IEEE Std 1394.1 [7], which are repeated in table 1 for convenience.

Table 1: bridge (brdg) field values

Value	Definition
00b	Non-bridge device
01b	Reserved
10b	Bridge portal (No network topology change since last bus reset)
11b	Bridge portal (The network topology has changed since last bus reset)

The HL2 bridge portals as defined in the present document shall use the value 10b or 11b, based on whether there has been a network topology change.

5.5 Cycle master adjustment packet

If the local cycle master is an adjustable cycle master and is not the leaf bus bridge portal itself, the bridge portal shall transmit a cycle master adjustment packet to the local cycle master as described in IEEE Std 1394.1. If the local cycle master is not an adjustable cycle master, the leaf bus bridge portal shall try to become the local cycle master itself IEEE Std 1394.1 [7].

5.6 Response packet

A response packet to a remote request may not be returned to its requester since a destination node may be unplugged from the destination remote bus, or the global node ID may no longer be valid. It is also possible that bus congestion may cause a serious transmission delay. In such cases, a bridge portal shall play the role of proxy for the responder identified by source_ID of the response packet.

When a bridge portal issues a response packet as a proxy to the responder, the bridge portal shall use the header fields as defined in IEEE Std 1394.1 [7].

5.7 Bridge management messages encapsulation

A bridge-aware node may send messages to bridge portals, e.g., to set up an isochronous stream. A bridge portal also may initiate or forward messages to other bridge portals. These messages shall be encapsulated in the data payload of a block write request packet addressed to a bridge portal's MESSAGE_REQUEST or MESSAGE_RESPONSE register as specified in IEEE Std 1394.1 [7].

The formats of bridge management messages are defined in IEEE Std 1394.1 [7]. A bridge shall support and process all of the stream management messages (JOIN, LEAVE, LISTEN, and STREAM_STATUS) as defined in IEEE Std 1394.1 [7]. A bridge shall support and process TIMEOUT and BUS_ID messages as defined in the present document. All other messages defined in IEEE Std 1394.1 [7] are optional.

6 Global node IDs

A global node ID is a 16 bit logical identifier used to uniquely identify a node in a net. A global node ID is composed of a 10-bit bus ID and a 6-bit virtual ID as defined in IEEE Std 1394.1. These are analogous to the bus ID and the physical ID specified by IEEE Std 1394. A global node ID is stable across bus resets.

6.1 Global node ID allocation

6.1.1 Bus ID

Bus IDs are assigned by the prime portal, as described in clause 8. Only bridge portals have a register to store the bus ID of its local bus, i.e., the CLAN_INFO register. The bus ID field of the Node ID register of each node shall not be changed from the local bus ID value of $3FF_{16}$.

6.1.2 Virtual ID

6.1.2.1 Branch bus virtual ID allocation

On the branch bus, the HL2 physical IDs are sufficiently stable (see in [3]) that they shall be used as virtual IDs.

6.1.2.2 Leaf bus virtual ID allocation (Informative)

This clause describes how virtual IDs may be allocated on a leaf bus. Alternate virtual ID allocation schemes may be implemented because no bridge co-ordination is required on the leaf bus for virtual ID mapping.

After every bus reset on the leaf bus, the leaf bus bridge browses the bus. EUI-64 values are used to uniquely identify devices (as specified in [2]). After a bus reset on the leaf bus, the leaf-bus HL2 1394 bridge layer performs the following operations:

- assigns a new virtual ID to a device that has never been connected to the leaf bus before;
- keep the same virtual ID to a device that was connected to the leaf bus before the bus reset;
- reassigns the reserved virtual ID to a device that has been connected to the leaf bus in the past;
- when a device is removed from a leaf bus, its virtual ID is set aside for the future use as long as the local bus ID remains the same, and will be reused if the same device is connected to the bus again.

When two nodes communicate directly on the same bus, they use their local node IDs in source and destination fields of the asynchronous packets. When a node sends a packet to a remote node (a node that is not connected to the same bus), it uses the remote node' s global node ID in the packet destination field and its local node ID in the packet source field. The entry bridge portal forwarding this packet translates the source local node ID into the source global node ID. The exit bridge portal connected to the destination bus translates the destination global node ID into the destination local node ID.

A node does not know its own global node ID. It just knows the global node IDs of remote devices that it wants to communicate with.

6.1.2.3 Virtual ID recycling (normative)

On either the branch bus or a leaf bus, a bridge portal shall never reassign the same virtual ID to a different device as long as the bus ID remains the same.

On a leaf bus, if the portal runs out of available virtual IDs under one bus ID, it shall change the bus ID by requesting a new bus ID from the prime portal (as described in clause 7) so that every virtual IDs on the bus can be re-allocated.

On the branch bus, the prime portal shall change the bus ID when all the physical IDs have been recycled, i.e., when the toggle bit is set in the CL_SELF_ID indication primitive of the 1394 SSCS [3].

6.2 Global node ID operation

A bridge portal shall translate global node IDs into physical node IDs for asynchronous subaction packets and GASP packets as described in IEEE Std 1394.1 [7].

7 Bus ID assignment & prime portal selection

In the absence of unrestricted bridges on the HL2 bus, the bridge portal on the HL2 bus that contains the IRM (as defined in the 1394 SSCS [3]) becomes the prime portal. On the HL2 bus, if the IRM is not a bridge portal, then the portal with the lowest physical ID becomes the prime portal. The prime portal is the single depository of the bus IDs throughout the net and is capable of assigning a unique bus ID to each bus in the net. The prime portal shall set the alpha bit in the CLAN_INFO register.

If there is one or more unrestricted bridges on the HL2 bus, a restricted bridge portal shall wait for the end of a quarantine period before sending BUS_ID messages to the prime portal. At the end of a quarantine period, one of the unrestricted portals has set its alpha bit in the CLAN _INFO register.

When a bridge that does not contain the prime portal wants a new bus ID for its leaf bus, it shall send a BUS_ID message with its leaf bus portal's EUI-64 to the alpha portal's MESSAGE_REQUEST register with the q bit set to one.

The *bus_ID* field may contain a preferred bus ID. If there is no preferred bus ID, the *bus_ID* field shall contain $3FF_{16}$.

When a bridge receives a BUS_ID message with its q bit set to zero and the *bus_ID* field contains a value other than $3FF_{16}$, the bridge shall check the EUI-64 in the message. If the EUI-64 matches one of its portals' EUI-64, it shall update the *bus_ID* field in the CLAN _INFO register in the corresponding portal.

If the *bus_ID* field in the BUS_ID message received is $3FF_{16}$, the requested bus ID allocation has failed.

NOTE: There may be cases when a bridge receives a BUS_ID response message without sending a BUS_ID request message. This may happen when unrestricted bridges are connected to the branch bus.

The prime portal that receives a BUS_ID request message for a leaf bus shall return the BUS_ID response with an allocated bus ID to the MESSAGE_REQUEST register of the requester. When the prime portal wants to change the bus ID of the branch bus, it shall send a BUS_ID response message to the MESSAGE_REQUEST register of all HL2 bus portals.

If a bridge portal that is not the prime portal receives a BUS_ID request message, the node shall return a BUS_ID response with an appropriate error code, i.e., FAIL.

After a net reset, every portal on the HL2 bus shall check whether the prime portal is the same node. If it has changed, the bus ID of every bus shall be revalidated.

8 Remote-timeout operations on HL2

In a bridged environment, due to longer transmission delays, the local bus *split_timeout* mechanism cannot be used for split transactions that are intended to pass bus bridges. Instead of the *split_timeout*, a *remote_timeout* is defined.

To determine a *remote_timeout* in a bridged environment, a bridge-aware device shall send a TIMEOUT bridge management message to the destination global node ID as specified in IEEE Std 1394.1. The processing of the TIMEOUT message shall be as specified in IEEE Std 1394.1 [7] with the exception of the *remote_timeout* field (described below).

INTERNAL_REQ_TIME is defined as the request processing time of the internal fabric including the maximum retry period.

INTERNAL_RESP_TIME is defined as the response processing time of the internal fabric including the maximum retry period.

The *HL2_MAX_TX_TIME* is defined as the maximum transmission time over the HL2 bus. This *HL2_MAX_TX_TIME* is bounded by the time of life mechanism specified in the 1394 SSCS. The *HL2_MAX_TX_TIME* shall be half of the HL2 split_timeout (100ms) as recommended by the 1394 SSCS, annex C.

When a bridge receives a TIMEOUT request message, it shall behave as follows:

If the *destination_id.bus_id* field and the *source_id.bus_id* field point to the same bus, then the TIMEOUT request message is not forwarded to any portal. The bridge shall generate a response message as described in IEEE Std 1394.1 [7].

Otherwise, if one of the *destination_id.bus_id* or *source_id.bus_id* fields point to the HL2 bus, then the bridge shall add *INTERNAL_REQ_TIME* plus the *HL2_MAX_TX_TIME* to the *remote_timeout* field (i.e., the *remote_timeout_seconds* and *remote_timeout_cycles* fields).

If **none** of the *destination_id.bus_id* field and the *source_id.bus_id* fields point to the HL2 bus, then the bridge shall add *INTERNAL_REQ_TIME* plus one **half** of the *HL2_MAX_TX_TIME* to the *remote_timeout* field (in fact *remote_timeout_seconds* and *remote_timeout_cycles* fields).

NOTE 1: When the HL2 bus is neither the destination bus nor the source bus, then two wireless bridges are involved in the communication.

If the bridge does not contain the exit portal (i.e. the bridge is not attached to the bus of the destination node of the TIMEOUT message), it shall forward the message to the next bridge of the path. Otherwise, it shall add the local bus SPLIT_TIMEOUT to the received *remote_timeout* field. It shall then generate a TIMEOUT response message back to the requester as specified in IEEE Std 1394.1 [7]. The *remote_timeout_seconds* and *remote_timeout_cycles* fields shall be copied from the TIMEOUT request message.

Respectively, when a bridge receives a TIMEOUT response message, it shall behave as follows:

If one of the *destination_id.bus_id or source_id.bus_id* fields point to the HL2 bus, then the bridge shall add *INTERNAL_RESP_ TIME* plus the *HL2_MAX_TX_TIME* to the *remote_timeout* field (in fact *remote_timeout_seconds* and *remote_timeout_cycles* fields).

If **none** of the *destination_id.bus_id* field and the *source_id.bus_id* fields point to the HL2 bus, then the bridge shall add *INTERNAL_RESP_TIME* plus one **half** of the *HL2_MAX_TX_TIME* to the *remote_timeout* field (in fact *remote_timeout_seconds* and *remote_timeout_cycles* fields).

NOTE 2: When the HL2 bus is neither the destination bus nor the source bus, then two wireless bridges are involved in the communication.

When sending asynchronous subactions on the 1394 SSCS [3], the bridge layer shall use the HL2_MAX_TX_TIME as *time_of_life* in the CL_ACTION primitive.

9 Clock synchronization

The net cycle master is defined as the single source of the clock synchronization throughout the net, and located on the same bus as the prime portal.

If there is no unrestricted bridge portal on the HL2 bus, the net cycle master shall be the cycle master of the HL2 bus. If there is one or more unrestricted bridge portals on the HL2 bus, one of the unrestricted bridge portals becomes the cycle master on the branch bus.

Since the net cycle master is always either on the HL2 bus or across one of the unrestricted bridges on the HL2 bus, if any, the direction of clock synchronization through a restricted bridge is always from the branch (HL2) bus to the leaf (wired) bus.

The phase synchronization with zero phase difference shall be implemented for clock synchronization as described in IEEE Std 1394.1. This means that the *cycle_offset* fields of the two adjacent buses are identical across a bridge. Other two fields in the CTR register, i.e., *second_count* and *cycle_count* fields may have independent values across a bridge. However, the *cycle_count* fields of the two buses shall change simultaneously.

If the leaf bus has a local adjustable cycle master other than the bridge portal itself, the portal shall send cycle adjustment packets to the local cycle master so that the local cycle master is synchronized to the cycle master on the HL2 bus.

10 Transaction routing and operations

The behaviour of bridge portals involved in remote transaction routing during normal operations (outside a quarantine period) is described in this clause.

10.1 Leaf bus (wired bus)

The routing rules for a leaf bus will be described in the following clauses.

10.1.1 Packet reception on leaf bus

An initial entry portal on a leaf bus shall process a successfully received asynchronous packet and forward it according to the rules defined in table 2. Unsuccessfully received packets shall be processed according to IEEE Std 1394.

Table 2: Routing rules for initial entry portal on source leaf bus

Condition		Operation
destination_bus_ID	**Packet Source**	
Neither equal to the bus_ID of the portal nor $3FF_{16}$ (global node ID of a remote node)	Bridge-aware node	The portal shall return *ack_complete* to a response packet, and *ack_pending* to a request packet. The packet shall be forwarded to the co-portal with the source ID translation as described in IEEE Std 1394.1 [7].
	Non-bridge-aware node	The portal shall return *ack_complete* to both a response packet and write request packet. The packet shall be forwarded to the co-portal after the source ID translation as described in IEEE Std 1394.1. The portal shall reject any request subactions except write request with resp_address_err and an extended response code of ext_legacy as described in IEEE Std 1394.1 [7].
Equal to the bus_ID of the portal (not $3FF_{16}$) (global node ID of a local node)		If the packet is addressed to the portal itself, it shall be processed according to IEEE Std 1394. Otherwise, the portal shall return *ack_complete* to a response packet and *ack_pending to a request packet*. The portal then shall echo the request packet to the local bus with the source ID and destination ID translation as described in IEEE Std 1394.1 [7].
3FF16 (local node ID of a local node)		If the packet is addressed to the portal itself, it shall be processed according to IEEE Std 1394. Otherwise, it shall be ignored.

According to table 2, an entry portal on leaf bus shall forward every successfully received asynchronous packet addressed to a remote node to its co-portal, even if the destination bus ID is not valid. The exception to this rule is that read and lock request transactions initiated by non-bridge aware devices are rejected with resp_address_err and an extended response code of ext_legacy as described in IEEE Std 1394.1 [7].

Invalid destination bus ID is detected by the 1394 SSCS [3] of the co-portal as described in clause 10.2.1.

10.1.2 Packet transmission on leaf bus

When a portal on a leaf bus receives a remote transaction from its co-portal, the leaf bus portal is called a terminal exit portal.

A terminal exit portal is responsible for the destination ID translation as described in IEEE Std 1394.1 [7] before forwarding the packet to the bus. A terminal exit portal expects to receive an ack_pending or an ack_complete for a request subaction. When ack_complete is received, the portal synthesizes a response packet with resp_complete and sends it back to the originator on the source bus. If an ack_pending has been received, the portal needs no further action, since the destination device will return the corresponding response packet.

When a transmission error occurs, a terminal exit portal shall synthesize a response packet as specified in IEEE Std 1394.1 [7].

10.2 Branch bus (HL2 Bus)

The branch (HL2) bus can be either source, intermediate or destination bus for a remote transaction.

10.2.1 Packet transmission on the branch bus

If the branch bus is the destination bus, the exit portal shall perform the destination ID translation as described in IEEE Std 1394.1 [7].

The bridge layer shall post packets to the 1394 SSCS using the CL_ASYNC_ACTION request primitive. The 1394 SSCS [3] reports *reply_accepted*, *reply_busy*, or *reply_missing*.

When *reply_accepted* is returned, the packet has been successfully accepted by the 1394 SSCS and will be delivered to the destination.

When *reply_busy* is returned, the 1394 SSCS could not accept the packet. The bridge layer shall retry the transmission later until it succeeds or its maximum forward time (either INTERNAL_REQ_TIME or INTERNAL_RESP_TIME as defined in clause 8) is reached. If the maximum forward time is expired before *reply_accepted* is returned by the 1394 SSCS [3], the bridge layer shall build a response packet as defined in clause 5.6 with *resp_conflict_error* response code and *ext_congestion* extended response code.

When *reply_missing* is returned, the destination was not found on the HL2 bus. This is due to either an invalid bus ID or an invalid virtual ID. In this case, the bridge layer shall build a response packet as defined in clause 5.6 with *resp_address_error* response code and *ext_invalid_global_ID* extended response code.

10.2.2 Packet reception on the branch bus

The 1394 SSCS performs packet routing within the HL2 bus as described in [3].

If the bridge layer receives a CL_ASYNC_ACTION indication primitive from the 1394 SSCS with a positive *time_of_life*, it shall forward this packet to its co-portal.

If the bridge layer receives a CL_ASYNC_ACTION indication primitive from the 1394 SSCS with zero or negative *time_of_life*, it means the packet spent too much time during its transmission over the HL2 bus and shall be discarded. If the packet was a request subaction, the bridge layer shall build a response packet as defined in clause 5.6 with *resp_conflict_error* response code and *ext_congestion* extended response code and send it back to the requester.

11 Stream management

11.1 Stream management message processing

The stream management message processing shall be as defined in IEEE Std 1394.1. The resource management on the HL2 bus shall follow the rules defined in the 1394 SSCS. The resource management on a leaf bus shall follow the rules defined in IEEE Std 1394.

It is recommended that the averaging window size (as defined in IEEE Std 1394.1 [7]) used in a stream management message be set to 2 ms to optimize the bandwidth reservation on the HL2 bus. If the averaging window size is 2 ms or smaller, a portal reserving bandwidth on the HL2 bus should use the *average_payload* value in the message to calculate the *payload* field of the ALLOCATE_SOME command defined in the IEEE 1394 SSCS [3]. Otherwise, a portal should use the *max_payload* value to calculate the *payload* field.

11.2 Time stamping and isochronous transmission

A bridge shall provide a constant delay system for every isochronous stream. A time stamping and buffering system, i.e., the HL2 constant delay buffer, is required for this purpose.

A bridge shall support the CIP time stamps as defined in IEC 61883 [8]. The following two kinds of CIP packet formats shall be supported.

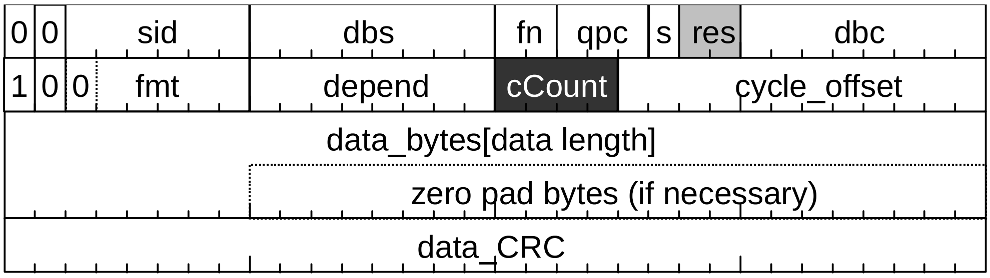

Figure 16: CIP Packet Format (without SPH)

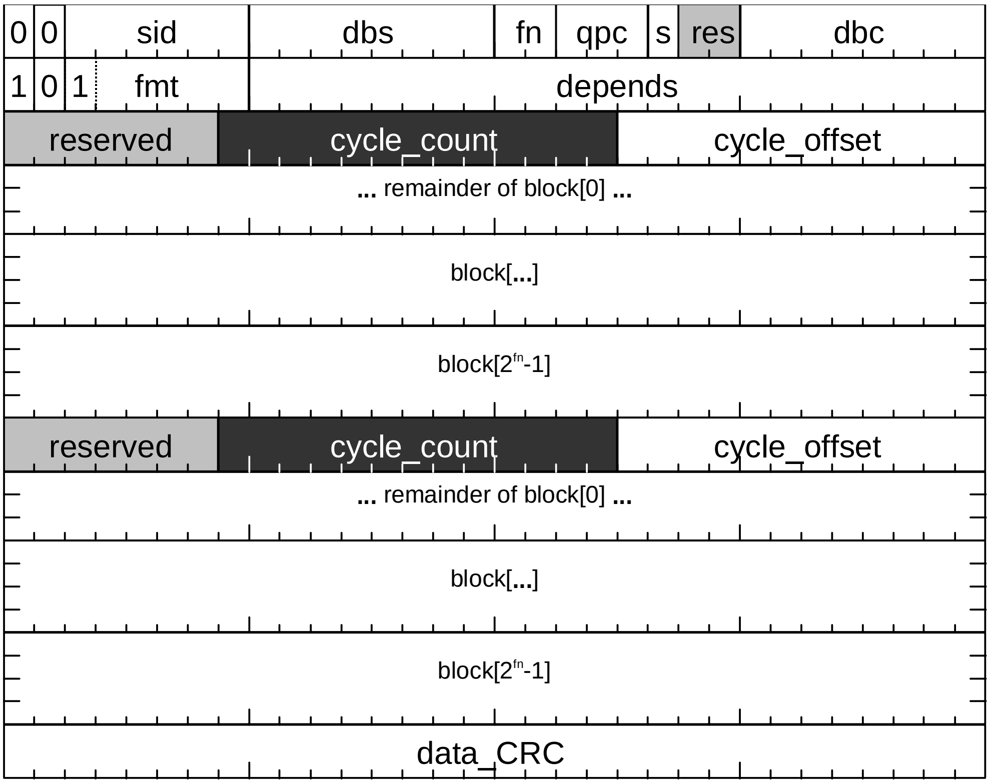

Figure 17: CIP Packet Format (with SPH)

The direction of an isochronous traffic can be either (1) from a leaf bus to the HL2 bus, or (2) from the HL2 bus to a leaf bus.

In case (1), a bus bridge shall invoke the CL_ISOCH_STREAM request primitive of the 1394 SSCS for every isochronous packet. The *cycle_count* value in the SPH and/or the cCount field in the CIP header of each isochronous packet (as defined in IEC 61883 [8]) shall be replaced with a relative time stamp value. This relative time stamp shall indicate how many cycles in advance, relative to the original IEC 61883 [8] time stamp value, the packet is received by the entry portal. The *cycle_offset* field shall remain unchanged.

If the Source Packet Header is present in a CIP packet, its time stamp value (cipStd.cycle_count) shall be converted to a relative value (hl2Bus.cycle_count) by subtracting the local cycle time (*cycle_count*) of the leaf bus when the packet was received from the time stamp value (cipStd.cycle_count). The following equation defines this time stamp conversion:

hl2Bus.cycle_count = (cipStd.cycle_count - *cycle_count + 8 000*) % 8 000;

If the cCount field of a CIP header is used, its time stamp value (cipStd.cCount) shall be converted to a relative value (hl2Bus.cCount) by subtracting the local cycle time (*cycle_count*) of the leaf bus when the packet was received from the time stamp value (cipStd.cCount). The following equation defines this time stamp conversion:

hl2Bus.cCount = (cipStd.cCount - *cycle_count*) & 0xF;

NOTE 1: The data CRC of the isochronous packet shall be recalculated after the *cycle_count* value in the SPH and/or the cCount field in the CIP header has been modified.

The *time_stamp* parameter (seconds and cycles) of the CL_ISOCH_STREAM request primitive shall represent the time when an isochronous packet is posted to the 1394 SSCS on the HL2 bus cycle time domain.

In case (2), a bus bridge receiving a CL_ISOCH_STREAM indication primitive from the 1394 SSCS shall store the isochronous packet in its HL2 constant delay buffer until the presentation time, which is the sum of the *time_stamp* parameter and the maximum isochronous delay on the HL2 bus, i.e., 49 cycles, is reached.

Each packet shall be sent out on the leaf bus during the isochronous period that is indicated by the sum of the time the packet came out of the HL2 constant delay buffer and the *isochronous_delay* value in the configuration ROM of the HL2 bridge portal (see clause 5.1.2.3).

If the Source Packet Header is present in a CIP packet, its relative time stamp (hl2Bus.cycle_count) value shall be converted back to an absolute value by adding the local cycle time (*cycle_count*) of the leaf bus when the packet will be transmitted to the relative time stamp value. The following equation defines this time stamp conversion:

cipStd.cycle_count = (hl2Bus.cycle_count + *cycle_count*) % 8 000;

If the cCount field of a CIP header is used, its relative time stamp (hl2Bus.cCount) shall be converted back to an absolute value by adding the local cycle time (*cycle_count*) of the leaf bus when the packet will be transmitted to the relative time stamp value. The following equation defines this time stamp conversion:

cipStd.cCount = (hl2Bus.cCount + *cycle_count*) & 0xF;

NOTE 2: The data CRC of the isochronous packet shall be recalculated after the *cycle_count* value in the SPH and/or the cCount field in the CIP header has been modified.

12 Net update

A restricted net topology exists when only restricted bridges are connected to the HL2 bus. Only restricted net topologies are in the scope of the present document.

12.1 Net reset and quarantine

A net reset is defined as a bus reset with at least one self-ID packet having its *brdg* field set to three. A net reset shall be propagated through bridges for quarantine purposes. When a portal detects a net reset or the insertion or the removal of a bridge on its local bus, it shall propagate a net reset to its co-portal's bus.

Quarantine is the period of time during which a bridge-aware node shall not transmit asynchronous subactions to a remote node (a node whose most significant ten bits are other than $3FF_{16}$). This also applies to a bridge portal. A quarantine period commences either when a net reset is detected or when the insertion or the removal of a bridge is observed on the local bus.

A quarantine period ends when the quarantine bit in the NET_GENERATION register is cleared. Clearing quarantine means sending a broadcast write request to the NET_GENERATION register of all bridge-aware nodes (including bridge portals) on the local bus with the quarantine bit set to zero. A quarantine period shall be terminated as described below:

On the branch bus, in the presence of only restricted bridges, the prime portal shall clear the quarantine immediately after a net reset or a bus reset with a net topology change in the local bus (insertion or the removal of a bridge). In the presence of unrestricted bridges on the branch bus, one of the unrestricted bridges will clear the quarantine. The leaf bus bridge portal shall clear the quarantine on the leaf bus when and only when the quarantine has been cleared on the branch bus.

A bridge-aware node shall behave during and after a quarantine period as described in IEEE Std 1394.1 [7].

12.2 Enabling/Disabling of a restricted bridge

A restricted bridge portal has limited capabilities because it is intended to operate on a leaf bus. If it happens that several restricted bridge portals are connected to the same leaf bus, only one of them shall survive. If at least one unrestricted bridge portal is connected to the leaf bus, the restricted bridge portal shall be disabled. A disabled bridge may continue to respond normally to asynchronous subactions on its local bus, but all traffic across its internal fabric is stopped. A disabled bridge shall set the *brdg* field of the seld-ID packet to 0.

On the leaf bus side, the presence of the HL2 unit directory (see clause 5.1.1.4) in the configuration ROM identifies a HL2 bridge. The HL2 unit directory contains a *bridge_revision* entry as described in clause 5.1.1.5. If the *phase* field of the *bridge_revision* entry is set to 1, a restricted bridge is identified. If the *phase* field of the *bridge_revision* entry is set to a value greater than 1, an unrestricted bridge is identified.

The restricted bridge portal shall monitor the *brdg* field of all self-ID packets after each bus reset. Actions that shall be taken by the restricted bridge portal depend on whether the bridge is already enabled. Note that a bridge shall be disabled after a power reset.

Bridge currently disabled

If the analysis of the self-ID packets reveals that there is at least another node on the bus and all the self-ID packets have their *brdg* field set to 0, the restricted bridge may be enabled. When enabling itself, a bridge shall initiate a bus reset with the *brdg* field of the self-ID packet set to three on the both portals' buses. After a portal has been enabled, it shall set the *brdg* field of its self-ID packet to two for the following bus resets.

If the analysis of the self-ID packets reveals that at least one self-ID packet has its *brdg* field set to either two or three, the restricted bridge shall remain disabled.

Bridge currently enabled

If the analysis of the self-ID packets reveals that all the self-ID packets originated from the other nodes have their *brdg* field set to 0, the restricted bridge may remain enabled.

If the analysis of the self-ID packets reveals that there is at least one self-ID packet with its *brdg* field set to two or three, then the restricted bridge shall run the following contention procedure:

All the incoming transactions shall be suspended (i.e., no ack_complete or response packets are returned) except for read transactions.

If at least one self-ID packet with *brdg* field set to either two or three is originated from either an unrestricted HL2 bridge or a non HL2 bridge (bridge configuration ROM does not contain any HL2 unit directory), then the restricted bridge shall disable itself.

If all self-ID packets with *brdg* field set to either two or three are originated from restricted HL2 bridges then the restricted bridge with the highest physical ID may remain enabled and the suspended transactions may be resumed. All the restricted bridges without the highest physical ID shall disable themselves.

When disabling itself, a bridge shall initiate a bus reset on the both portals' buses after setting the *brdg* field of its self-ID packet to zero for the following bus resets.

13 Operation in a bridged environment

13.1 Bridge-aware devices

A bridge aware device shall behave as defined in IEEE Std 1394.1 [7].

13.2 Legacy device

A device that does not behave as a bridge-aware device as defined in IEEE Std 1394.1 [7] is called a legacy device. Legacy devices are unaware of global node IDs that refer remote nodes and node IDs can be used only within a local bus.

However, some legacy devices also may be able to communicate successfully with remote devices in a bridged environment. There are two ways for legacy devices to operate with remote nodes:

A legacy device responds to a remote request subaction originated by bridge-aware device on a remote bus.

A legacy device may originate remote write requests as described in clause 10.1.1 when it is given the destination global node ID by a bridge-aware controller.

Annex A (informative): Bibliography

ETSI TS 101 761-2: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 2: Radio Link Control (RLC) sublayer".

ETSI TS 101 761-4: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Data Link Control (DLC) Layer; Part 4: Extension for Home Environment".

ETSI TS 101 493-5: "Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Packet based Convergence Layer; Part 5: IEEE 1394 Bridge Specific Functions sub-layer for unrestricted topology".

History

Document history		
V1.1.1	July 2001	Publication