CONGRUENT SOFTWARE, INC.
98 Colorado Avenue
Berkeley, CA 94707
(510) 527-3926
(510) 527-3856 FAX

FROM: Peter Johansson

TO: IEEE P1394a Ballot Response Committee

DATE: February 3, 1999

RE: Coordination requests from IETF IP1394 and P1394.1 working groups

As a consequence of discussions late in 1998 in these working groups (in which I was also an active participant), the Chairs have requested that some work items be transferred to the P1394a draft standard. Specifically:

- that the definition of the network protocol manager (NPM) in the Internet-Draft Draft-IETF-IP1394-IPv4-13 be generalized as the broadcast channel manager (BCM) and be standardized in P1394a instead of the IETF; and
- that a new format for asynchronous stream packets be defined to include *source_ID* in order to permit work in both the IETF and P1394.1 working groups to build upon a common standard.

That work has been incorporated into the draft of P1394a under review by the BRC, but there is also a desire by both the IETF and P1394.1 working groups to critically review the effort.

The rest of this document is an excerpt from the relevant portions of the P1394a draft in preparation for recirculation ballot.

8.1 Asynchronous stream packet format

The format of an asynchronous stream packet is identical to that of an isochronous stream packet, as specified by clause 6.2.3.1 of IEEE Std 1394-1995, and illustrated by figure 8-1.

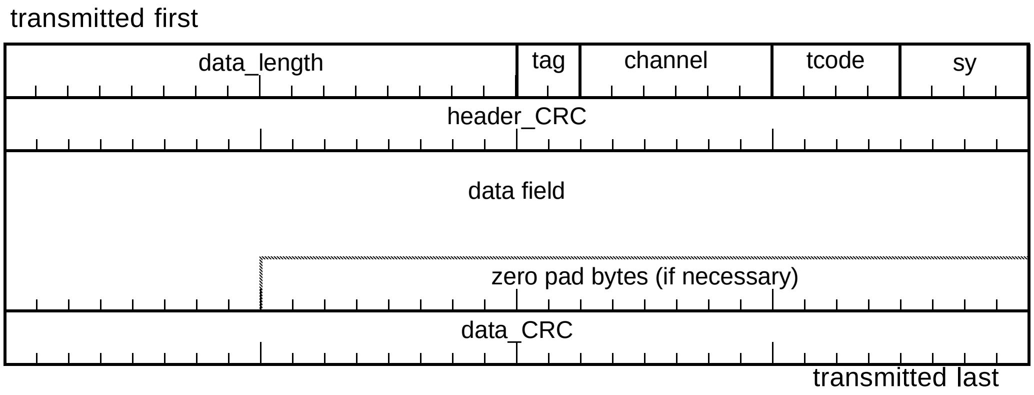


Figure 8-1 — Asynchronous stream packet format

The fields of an asynchronous stream packet shall conform to requirements of this standard and those specified in clause 6.2.4 of IEEE Std 1394-1995.

The ***data_length*** field shall specify the length in bytes of the data field in the asynchronous stream packet The number of bytes in the data field is determined by the transmission speed of the packet and shall not exceed the maximums specified by table 8-1 (which replaces table 6-4 in clause 6.2.2.3 of IEEE Std 1394-1995).

Table 8-1—Maximum data payload for asynchronous primary packets

Data rate	Maximum payload (bytes)	Comment
S25	128	TTL backplane
S50	256	BTL and ECL backplane
S100	512	Cable base rate
S200	1024	
S400	2048	
S800	4096	
S1600	8192	
S3200	16384	

The ***tag*** field shall have a value of zero, unformatted data, or one, global asynchronous stream packet (GASP) format; other values of *tag* are reserved for future standardization. When *tag* is zero the content and format of the data field are unspecified. Otherwise, when *tag* is one the format of the asynchronous stream packet is specified by clause 8.2.

The ***channel*** field shall identify the stream; the channel identified shall be allocated from the isochronous resource manager CHANNELS_AVAILABLE register.

NOTE—Unlike isochronous stream packets, which may continue to be transmitted for up to one second subsequent to a bus reset without channel reallocation, asynchronous stream packets may not be transmitted until their channel number(s) are reallocated.

The ***tcode*** field shall have a value of A_{16}. The new name for this transaction code value is stream packet; the context in which the packet is sent determines whether it is an asynchronous or isochronous stream packet.

The usage of any fields not specified above remains as described by IEEE Std 1394-1995.

8.2 Global asynchronous stream packet (GASP) format

Motivated by work on IEEE Project 1394.1, Serial Bus to Serial Bus Bridges, this standard defines an asynchronous stream packet format suitable for transport across a bridge from one Serial Bus to another.

The format of a global asynchronous stream packet is an extension of that specified by clause 8.1 which utilizes the first two quadlets of the packet's data payload. The GASP format is illustrated by figure 8-2.

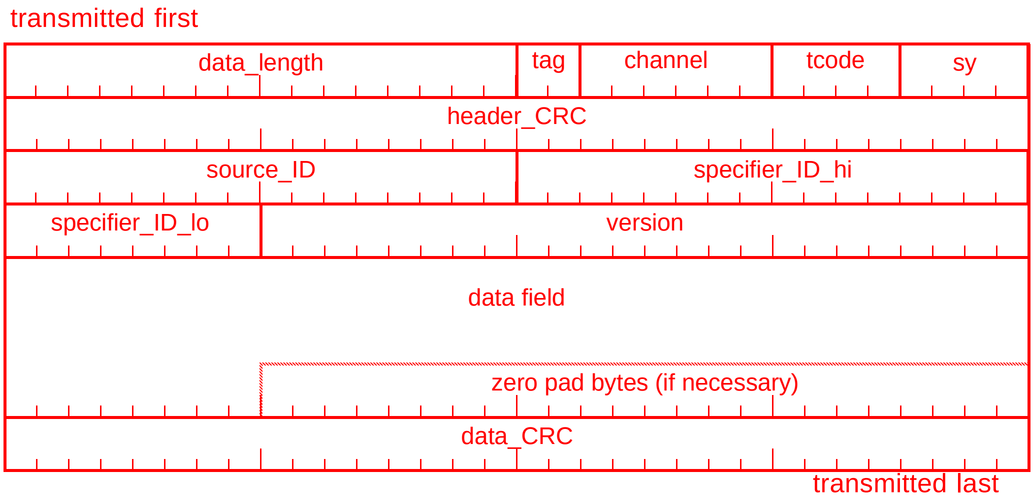


Figure 8-2 — Global asynchronous stream packet (GASP) format

Except as specified below, the definition and usage of the fields in figure 8-2 is contained within clause 8.1.

The ***tag*** field shall have a value of one.

The ***source_ID*** field shall specify the node ID of the sending node and shall be equal to the most significant 16 bits of the sender's NODE_IDS register.

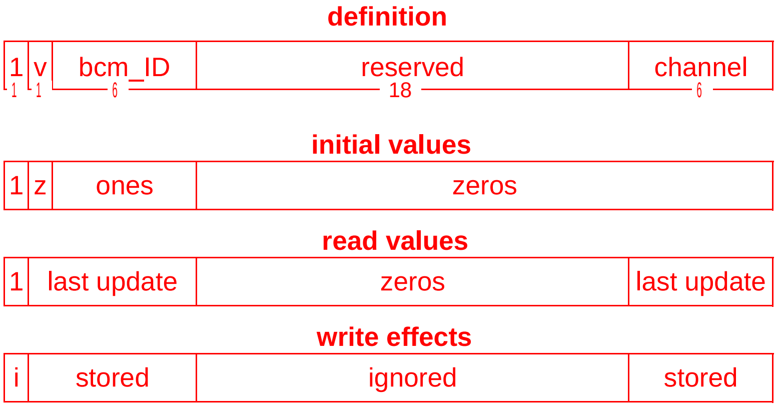
The ***specifier_ID*** field shall contain a 24-bit organizationally unique identifier (OUI) assigned by the IEEE Registration Authority Committee (RAC). The owner of the OUI (company, accredited standards organization or industry group) shall be responsible to define the meaning and usage of the remainder of the data payload in the stream packet.

The meaning and usage of the ***version*** field shall be defined by the owner of specifier_ID.

9.16 ~~NETWORK_CHANNELS~~ BROADCAST_CHANNEL register

~~The CSR architecture reserves a range of addresses in initial register space for bus-dependent uses; clause 8.3.2.3 in IEEE Std 1394-1995 defines registers within that range for use by Serial Bus. This supplement defines a new register, NETWORK_CHANNELS, within that address space whose format and usage is to be specified by the Internet Engineering Task Force (IETF).~~ This register is optional and if implemented shall be a quadlet register located at offset 234_{16} within initial register space ~~FFFF F000 0234$_{16}$ within initial node space~~. Any node that implements the BROADCAST_CHANNEL register shall be isochronous resource manager-capable. All isochronous resource manager-capable nodes should implement the BROADCAST_CHANNELS register.

The BROADCAST_CHANNEL register permits the broadcast channel manager to communicate the channel number assigned for asynchronous stream broadcast to other nodes on Serial Bus. This register also provides a mechanism used by the election protocol for the broadcast channel manager. The BROADCAST_CHANNEL register shall support quadlet read and write requests, only. The definition is given by figure 9-5 below.



definition

1	v	bcm_ID	reserved	channel
1	1	6	18	6

initial values

1	z	ones	zeros	

read values

1	last update		zeros	last update

write effects

i	stored		ignored	stored

Figure 9-5—BROADCAST_CHANNEL format

The most significant bit (a constant one) differentiates the presence of the BROADCAST_CHANNEL register from the value (all zeros) that may be returned when this register's address is read at node(s) that do not implement the register.

NOTE—Nodes compliant with this supplement return an address error response when unimplemented addresses are accessed—but some implementations are known to complete such requests with *resp_OK* and response data of zeros.

The ***valid*** bit (abbreviated as *v* above), when set to one, indicates that the *channel* field contains meaningful information. Nodes shall not transmit stream packets that specify this channel while the *valid* bit is zero.

The ***bcm_ID*** field identifies the physical ID of the broadcast channel manager. When *bcm_ID* is equal to $3F_{16}$ the physical ID of the broadcast channel manager is not specified; otherwise it shall be initialized (by the broadcast channel manager) to the 6-bit physical ID assigned to the broadcast channel manager during the self-identification process.

The ***channel*** field shall be initialized by the broadcast channel manager (see clause 9.21) to identify the channel number shared all nodes for asynchronous stream broadcast.

9.21 Broadcast channel manager

The presence of a broadcast channel manager permits all nodes on a local Serial Bus to share a single channel for asynchronous stream broadcast. The broadcast channel manager shall be responsible to allocate a channel number from the CHANNELS_AVAILABLE register and communicate the channel number to other interested nodes on the same bus.

Subsequent to a Serial Bus reset a single broadcast channel manager shall be determined by a distributed algorithm executed by all the broadcast channel manager-capable nodes. The algorithm is straightforward: the broadcast channel manager-capable node with the largest 6-bit physical ID shall be the broadcast channel manager. The steps in the algorithm are as follows:

a) A broadcast channel manager-capable node shall also be a contender for the role of isochronous resource manager. The *C* (contender) and *L* (link active) bits in its self-ID packet shall be set to one;

b) Subsequent to a bus reset, isochronous resource manager contention takes place during the self-identification process specified by IEEE Std 1394-1995;

c) A broadcast channel manager-capable node that wins the contention process is the broadcast channel manager and shall proceed with g). Other broadcast channel manager-capable node(s) not selected as the isochronous resource manager (hereafter referred to as candidates) shall continue with d);

d) A candidate broadcast channel manager shall delay before it attempts to become the broadcast channel manager. The delay time shall be equal to 15 ms * (*irm_ID* - *candidate_ID*), where *irm_ID* and *candidate_ID* are the physical IDs of the isochronous resource manager and the candidate broadcast channel manager, respectively. After the delay time has elapsed, the candidate broadcast channel manager shall examine the *bcm_ID* field in its own BROADCAST_CHANNEL register; if it is not equal to $3F_{16}$, another node is the broadcast channel manager. The losing candidate shall wait for the *valid* bit of its own BROADCAST_CHANNEL register to be set to one before transmitting any stream packets that use the *channel* number specified by the register;

e) Otherwise, the candidate broadcast channel manager shall attempt to read the BROADCAST_CHANNEL register of any contenders with a larger physical ID (these nodes were identified by the *C* bit in their self-ID packets). The candidate broadcast channel manager shall read the BROADCAST_CHANNEL register in the contender with the largest physical ID and progress downward. If the register is implemented, the broadcast channel manager is determined to be a different node. The losing candidate shall ignore the contents of BROADCAST_CHANNEL returned in the read response and shall wait for the *valid* bit of its own BROADCAST_CHANNEL register to be set to one before transmitting any stream packets that use the *channel* number specified by the register;

f) If no contender with a physical ID larger than the candidate broadcast channel manager's physical ID implements the BROADCAST_CHANNEL register, the search is complete and the candidate becomes the new broadcast channel manager;

g) Once elected, the broadcast channel manager shall update the *bcm_ID* field in the all the BROADCAST_CHANNEL registers implemented by any of the nodes on the bus (including itself) with its own physical ID. This signals to other candidates that a broadcast channel manager has been elected but may not have allocated a channel. Either a broadcast write request or a series of write requests addressed to individual nodes may be used;

h) The broadcast channel manager shall attempt to allocate a channel number from the CHANNELS_AVAILABLE register (note that the broadcast channel manager may also be the isochronous resource manager). If no channel number had been allocated prior to the bus reset, the broadcast channel manager shall wait one second before it attempts to allocate a channel number. Otherwise, the broadcast channel manager shall attempt to reallocate the same channel number in use before the bus reset; if the same channel number is not available, the broadcast channel manager may allocate a different channel number. If no channel number is available, the broadcast channel manager shall take no additional action (the *valid* bit(s) in all BROADCAST_CHANNEL registers were cleared by the bus reset);

i) Otherwise, the broadcast channel manager shall update its own BROADCAST_CHANNEL register with the allocated channel number and set the *valid* bit to one. The broadcast channel manager shall then write the updated value of the entire register to all the BROADCAST_CHANNEL registers implemented by any of the nodes on the bus. Either a broadcast write request (with the most significant 10 bits of *destination_ID* equal to the most significant 10 bits of the broadcast channel manager's NODE_IDS register and the most significant 10 bits of *source_ID* equal to $3FF_{16}$) or a series of write requests addressed to individual nodes shall be used to propagate the information.

A node that implements the BROADCAST_CHANNEL register but is not elected the broadcast channel manager typically awaits a write to its BROADCAST_CHANNEL that sets the *valid* bit to one; this indicates that the channel field is valid for asynchronous stream broadcast. If some time-out elapses without this occurrence, such a node may attempt to locate the broadcast channel manager and retrieve valid information from the BROADCAST_CHANNEL register. If the *bcm_ID* field in its own BROADCAST_CHANNEL register is not equal to $3F_{16}$, the address of the broadcast channel manager is known; otherwise the node may search for the broadcast channel manager as described in e) above. In either case, it is recommended that reads of the BROADCAST_CHANNEL register not be performed within a tight loop, as this could adversely degrade performance on the local bus.