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

FROM:	Peter Johansson
TO:	T10 SBP-3 working group
DATE:	February 19, 2001
RE:	Bridge-aware targets and node handles

The pages that follow contain proposed additions to the SBP-3 working draft that incorporate some, but not all, of the functionality required for "bridge-aware" targets. One clause is (informative) introductory material to be added to the model while the other is a new management ORB used to obtain a node handle from the target.

I have not documented another change that I believe is necessary to the login ORB. It is the addition of a *bridge_aware* bit that would place the target into a bridge-aware operating mode for all activity concerned with the login. I would like to have more working group discussion before documenting this change.

Please review this material in preparation for the meeting in Dallas, March 6 – 7.

4.7a Bridge-awareness

Targets designed to operate with initiators or data buffers on remote buses (*i.e.*, not the local bus but buses accessible *via* one or more intervening bridges) are described as *bridge-aware*. In general terms, this means that their designs embody an understanding of the particular requirements of draft standard IEEE P1394.1. More specifically, the salient features of bridge-aware targets are:

- The ability to distinguish between local node IDs, whose scope is restricted to the local bus, and global node IDs that reference a remote node (or indirectly reference a local node);
- Separate split transaction time-out values for requests addressed to local nodes and those addressed to remote nodes. The remote time-out value is significantly longer than the local bus split time-out;
- The ability to generate commands addressed to remote bridges. These commands are identified both by the data payload of the packet and by the value of the *snarf* field in the packet header of block write requests;
- Comprehension of new primary packet header fields, such as *proxy_ID* and *ext_rcode*, and new response codes;
- Implementation of the NET_GENERATION register;
- Particular behaviors in response to bus reset, notably self-quarantine of remote subactions and the possible invalidation of any global node IDs cached by the device;
- Recognition of commands originated by bridge portals and intended for bridge-aware nodes.

Most of the requirements above are best understood by direct reference to draft standard IEEE P1394.1 itself. However, there are other changes in Serial Bus Protocol necessitated by some of these new behaviors. In particular, *a)* initiators and targets require a stable method to identify nodes that contain data buffers and *b)* initiators and targets may no longer use bus reset as a mutual synchronization point since they will not observe bus reset on the other's bus.

The obvious candidate for stable reference to a node is its 64-bit unique ID, EUI-64. Unfortunately, legacy SBP-2 data structures are restricted to a 16-bit field to identify a node. The solution is to differentiate between two types of information that may be contained within the 16-bit field. One type is the local node ID documented by SBP-2. The second is a *node handle*, an arbitrary value assigned by the target to represent a particular EUI-64. Because IEEE P1394.1 restricts the most significant ten bits of a local node ID to all ones, this standard is free to define a node handle to be any 16-bit value whose most significant ten bits are other than all ones.

NOTE – Although a node handle and a global node ID are similar in that their most significant ten bits are not all ones, they are not the same thing. A global node ID, when used in the *destination_ID* field of a Serial Bus request subaction, causes the subaction to be routed by bridges to the intended recipient. A node handle should never be used in *destination_ID*; its value might coincidentally be equal to a valid global node ID—but one that correlates with a different EUI-64 than is associated with the node handle.

Before an initiator may use a node handle to refer to a particular node, it asks the target for a node handle that corresponds to the node's EUI-64. The initiator may provide a global node ID as a hint to the target, but the target is responsible to discover the global node ID that corresponds to the EUI-64 supplied by the initiator. Once the target has validated a relationship between the EUI-64 and a global node ID, the target returns a node handle to the initiator. Thereafter the node handle may be used in any address pointer to address the node identified by the EUI-64.

When a target encounters a node handle in any address pointer field, it decodes the reference into a global node ID which may be used to address the desired node. The target is responsible to maintain a valid correlation between a node handle and its associated EUI-64 and global node ID. A target that encounters a remote addressing error when a global node ID is used does not immediately terminate the associated task. Instead, device discovery methods (such as those described by IEEE P1394.1) are used to rediscover the global node ID that matches the EUI-64 cached for the node handle. Only if the desired node is no longer connected to the net does the target terminate affected tasks.

The other protocol change necessitated by bridges concerns bus reset. In general, bus resets in a network of interconnected buses are a local event not propagated by bridges.[1] If a bus reset occurs on the target's bus and the initiator is on a remote bus, the event will pass unobserved by the initiator. As a consequence, a protocol behavior useful in legacy SBP-2 becomes detrimental when bridges are present: a bridge-aware target cannot afford to abort its task sets in response to bus reset. The remote initiator is ill-equipped to detect such an event; even if the target attempts to notify the initiator, it is moot whether the initiator can successfully reinitiate work at the target in the face of subsequent bus resets.

The use of node handles is essential to surmount the bus reset problem. When a target operating in bridge-aware mode observes a bus reset, it does not abort its task sets. Instead it revalidates the global node IDs in use and insures that they continue to correspond to nodes originally specified by EUI-64. The initiator need not know about this action by the target, since the initiator continues to use stable node handles to identify nodes.

NOTE – Although this new protocol feature was designed as a response to bridges, it may be very useful in the context of the local bus. A node handle may refer to a local node; a target operating in bridge-aware mode is capable of determining the corresponding local node ID after bus reset. Without task set aborts forced by bus reset, target operations may be significantly more efficient.

2.4.4.4a Get node handle ORB

When an initiator establishes a login in bridge-aware mode, it shall obtain node handle(s) to use in all address pointers signaled to the target for the duration of the login. A node handle for a particular node, identified by its EUI-64, may be obtained by a get node handle request that uses the ORB format shown below.

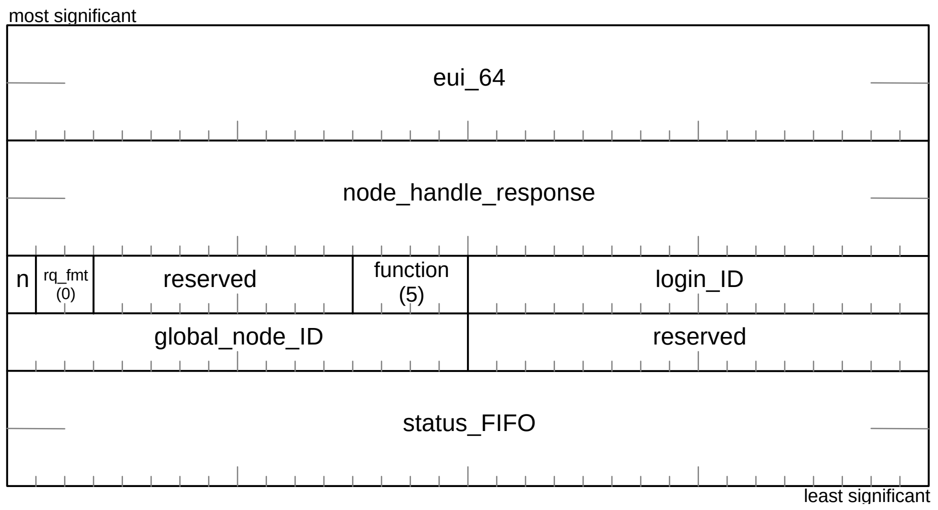

Figure 28a – Get node handle ORB

[1] Net topology changes anywhere within the net eventually cause a bus reset to occur on each bus within the net, but these bus resets are not synchronized with each other and do not carry any useful information except notification that net topology has changed.

The *eui_64* field shall contain the unique identifier of the node for which the node handle is requested and shall not be equal to the initiator's EUI-64.

The *node_handle_response* field shall specify the address of a quadlet buffer allocated for the return of the node handle. The *node_handle_response* field shall conform to the format for address pointers specified by Figure 11. The buffer shall be accessible to a Serial Bus quadlet write request.

The *notify* bit and the *rq_fmt* field are as previously defined for management ORB formats.

The *login_ID* field shall contain a login ID value obtained as the result of a successful login.

The initiator may set the value of the *global_node_ID* field to provide a hint to the target. When the value is all ones, no hint is given. Otherwise, *global_node_ID* contains a global node ID that may identify a node whose EUI-64 matches that specified by the *eui_64* field. In this case, the target, in place of an EUI-64 discovery search, may use a TIMEOUT message (as specified by draft standard IEEE P1394.1) to validate whether or not *global_node_ID* correlates with the specified EUI-64.

The *status_FIFO* field is as previously defined for management ORB formats and shall contain an address allocated for the return of status for the GET NODE HANDLE request, only. The contents of this field shall not update the status FIFO address established by the successful login that returned *login_ID*.

If the get node handle request fails, the contents of the *node_handle_response* buffer are unspecified. Otherwise, upon successful completion of a login, the target shall store the quadlet illustrated below.

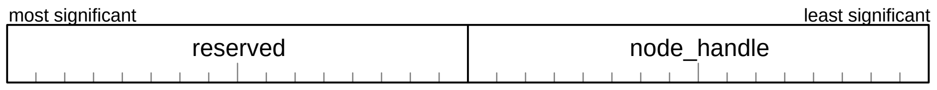

Figure 28b – Node handle response

The *node_handle* field shall contain a node handle whose value is assigned by the target. The most significant ten bits of the node handle shall not be ones. The initiator may use *node_handle* in place of a local node ID any address pointer signaled to the target in the context of the login identified by *login_ID*. For the duration of the login, the node identified by *node_handle* shall be the one specified by the *eui_64* field in the get node handle request.