Pride and Privilege in 1394 Arbitration

What We Are about

It appears accepted by most that in the present arbitration scheme some devices will not be as well-served, with regard to access to the bus, as would be desirable for those devices' optimal behavior, and, by extension, of the applications supported by those devices. An example of this type of problem is that addressed by the "transaction fairness" proposal previously accepted, wherein device may arbitrate to send more than one asynchronous packet during a single Fairness Interval, so long as it issues no more than one request packet per Fairness Interval. For devices critical to overall system behavior, such changes must be viewed as necessary to whatever extent they improve the bus' aggregate behavior in critical applications, without unnecessary or excessive degradation of the behavior of other less-critical applications, or undue cost in hardware or software support.

We are therefore seeking to improve the behavior of the bus without serious consequential harm to the applicability of the bus to its present wide application domain.

Privileged Arbitration

This proposal is actually quite simple, being a rather direct extension of an existing arbitration mechanism of the 1394 backplane environment into the cabled environment. Thus it is not entirely new, and is more logically consistent with the present arbitration mechanisms of the 1394 cabled environment. Its intent is to permit one or more critical nodes to arbitrate to send asynchronous requests more than once per Fairness Interval, with the expectation that such nodes and associated applications will perform "better" than when nodes may only arbitrate once per interval for the transmission of a request.

In cases where a small number of nodes are present on the bus, and some or all are privilege-enabled, this should result in an increase in bus utilization. In cases where complex topologies yield high gap-counts, but where only a few nodes (or just one) are actively issuing requests, this can reduce the frequency of Arbitration Reset Gaps, thereby improving throughput. In complex topologies with general activity by a high proportion of the nodes, the ability to enhance the availability of the bus for critical nodes (such as PCs and HDDs) critical application performance should be improved, while the ability to disable privileged arbitration permits a return to the more uniform "fairness" of the present arbitration mechanisms, but no worse.

Priority and Urgent Arbitration in the Backplane Environment

In clause 5.4.1.3 of the IEEE 1394-1995 standard is described the Urgent Arbitration mechanism of backplane environment. The core of this clause says that a device that supports priority arbitration must have a priority count register which is set to three (3) upon an arbitration reset gap, that upon issuing or seeing (either as the recipient of the packet or not) a priority packet the node must decrement this count, and that so long as that count is greater than one the node may arbitrate to send a priority packet.

An inherent advantage of this scheme is that it is self-limiting, and a fairness interval cannot be extended any longer than by three packets, with associated gaps, acks, etc. Thus, the arbitration scheme remains deterministic. No node may alter the bus arbitration behavior, and any node "knows" what the worst case it may be expected to deal with is, in terms of delay before it may arbitrate to send a packet, and when it will be able to send that packet.

However, it is not possible to apply this scheme to the cabled environment unchanged for a number of reasons, but two in particular. First, in the cabled environment there is no means by which to assert priority in arbitration, let alone degree of priority. (There is "natural" priority in the cabled environment, but this is a quite different thing.) Second, low-speed devices cannot "see" high-speed packets, and so will not reliably decrement their priority counts consistently with higher speed nodes that can see those packets.

Privilege in Cable Environment Arbitration

A node that has the right to arbitrate for access to the bus has a privilege, but not priority, since the node's priority is determined by its natural priority in the individual configuration. Yet, it is by the use of the priority field of the packet header that we will identify packets sent via privileged access.

The proposal is as follows:

No change shall be made to the ordinary fair arbitration of the asynchronous Fairness Interval, and all nodes may arbitrate for bus access according to that mechanisms rules. Privileged arbitration is optional, and operates in addition to the existing fair arbitration. If a node is privilege-capable, such support shall be implemented according to the following rules.

- A node that is privilege-capable shall have a register called privilege_count.
- A node that is privilege-capable shall have a status bit called faster_node_ident.
- A node that is privilege-capable shall have a control bit called privilege_enable, which shall be writeable from the bus, but may also be writeable from the node if the node is the bus manager. (The location of this bit in a register, or of a register for this function, is left to others within the working group….)
- Upon a Bus Reset the contents of the privilege_count register shall be cleared to zero (0x0), privilege_enable shall be set to one (0x1), and faster_node_ident cleared to zero (0x0).
- The privilege_enable bit shall accept writes of a value of zero only from the bus. (… Or maybe not, according to the group's wisdom….)
- During the bus Self_ID process, a privilege-capable node shall examine all self-ID packets, and if a packet is identified as from a node supporting speeds higher than this privilege-capable node supports this privilege-capable node shall set faster_node_ident to one (0x1). If faster_node_ident has previously been set to one, privilege_enable shall be cleared.

- Upon identification an Arbitration Reset Gap, the contents of this register shall be set to PRIVILEGE_LIMIT. (The value of PRIVILEGE_LIMIT is left to the group to define.)
- A node may arbitrate for control of the bus to transmit a privileged packet so long as the contents of privilege_count are greater than zero, and privilege_enable is equal to one (0x1).
- The contents of the privilege_count register shall be decremented by one (to saturation at zero (0x0)) upon the transmission, receipt or identification of a privileged packet on the bus of a packet the priority field of which is non-zero by a node.
- Any packet sent by means of privileged arbitration shall have its priority field set to one (0x1) by the transmitting node.

There are several items (such as where in Config ROM indication of the node's being privilege-capable will be indicated, the value of PRIVILEGE_LIMIT, size location of the privilege_count register, etc.) that have been intentional left out, pending the judgement of those who know or care more about how such things should be done.

The reasons for some of these rules were discussed on the reflector, but the one that bears repeating is that I believe it to be reasonable to assume that a slow node in a typical bus configuration may be assumed 'a priori' to be non-critical, and thus not a candidate for privilege.

Among the advantages I see in this approach are that it remains more nearly deterministic than a fairness budget, tampering can at worst result in falling back to the existing arbitration mechanisms, and it places no necessary additional burden on any other node, including the bus manager.

Problems that may arise include the appearance of "privilege domains" due to the presence of speed traps between privileged nodes. This can be beneficial, though, in cases where the domains are separate PCs with associated mass storage devices. Further, provision of the ability to disable privileged arbitration on a per node basis allows the bus manager to remove problem domains, and disable devices that it determines to be seeking privilege unnecessarily, or to the detriment of overall system performance.