Pride and Privilege in 1394 Arbitration

Richard Churchill

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 desirable to whatever extent they improve the bus' aggregate behavior in critical applications, and necessary to the extent that meaningful applications otherwise unsupportable become practical -- without unnecessary or excessive degradation of the behavior of other more or less critical applications, undue cost in hardware, or burden in software support. We are therefore seeking to improve the behavior of the bus without serious or objectionable consequential harm to the applicability of the bus to its present wide application domain.

Ideally, any solution arrived at should be logically consistent with the existing arbitration scheme, and thus not result in fundamental changes in the character or behavior of the theoretical bus, or concrete implementations thereof. Further, such changes should be transparent to existing devices.

Privileged Arbitration

This proposal is actually quite simple, being a rather direct but tuned extension of an existing arbitration mechanism of the 1394 backplane environment into the cabled environment. Thus it is logically consistent with the present arbitration mechanisms of the 1394 cabled environment, and makes use of existing structures in the 1394 environment. The 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, associated applications, and the aggregate system 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 proposal will permit 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, the frequency of long Arbitration Reset Gaps can be reduced to the extent that such active nodes are "privilege enabled," thereby improving throughput. In complex topologies with general activity by a high proportion of the nodes, the ability to enhance the bus availability for critical nodes (such as PCs and mass-storage devices) will tend to improve the behavior of critical application and nodes. Finally, the ability to disable privileged arbitration permits restoration of the more uniform and understood "fairness" of the present arbitration mechanisms.

Priority and Urgent Arbitration in the Backplane Environment

Clause 5.4.1.3 of the IEEE 1394-1995 standard describes the Urgent Arbitration mechanism of the backplane environment. The core of this clause states that a node supporting priority arbitration must have a priority count register, which is set to three (3) upon an arbitration reset gap; upon issuing or seeing (either as the recipient of such a packet or "in passing") an urgent packet the node must decrement this count, and so long as its count is greater than one the node may arbitrate to send an urgent packet.

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

It is impossible to directly apply this scheme to the cabled environment 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" higher-speed packets, and higher speed nodes will not see high-speed packets passed between nodes on the other side of any intervening speed trap slower than that packet. Therefore, nodes cannot reliably decrement their priority counts consistently across busses with nodes of disparate speeds.

Privilege in Cable Environment Arbitration

All nodes have the right to arbitrate for access to the bus for a request once per Fairness Interval, but a node that has the right to arbitrate for the bus more than once per interval has a privilege. Priority – natural priority -- is determined by the topology of a particular instance of the bus, and not by node characteristics (except, of course, for the case of a cycle master), 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 mechanism's 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. (Whether this bit is writeable from the node, the location of the bit in a register, what register to use, etc., are 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 (1), and faster_node_ident cleared to zero (0).

- The privilege_enable bit shall be "sticky," in that once cleared to zero (0) it shall not be capable of being reset to one (1) except by a bus reset.

- During the bus Self_ID process, a privilege-capable node shall examine all self-ID packets, and if a packet is identified as being 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 of an Arbitration Reset Gap, the contents of privilege_count 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 packet via privileged arbitration so long as the contents of privilege_count are greater than zero, and privilege_enable is equal to one (1).

- 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. A privileged packet is identified by a non-zero value in the priority field of the asynchronous packet header.

- Any packet sent by means of privileged arbitration shall have its priority field set to one (0x1) by the transmitting node.

- No privilege-capable node may inhibit the clearing of privilege_enable by other nodes in any manner, except as specifically permitted by other subsequent IEEE standards.

An additional consideration is that the asynchronous packet transmit queue must be unary, so that a privilege-capable node will be able to transmit all nodes it is requested to send even if the privilege_count is always exhausted before that node has an opportunity to arbitrate for a privileged packet.

There are numerous 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.

Desiderata

The reasons for some of these rules were discussed on the reflector, but the one that most 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 the following:

- It remains more nearly deterministic than a fairness budget.

- Tampering can at worst result in falling back to the existing arbitration mechanisms.
- In operation, it places no necessary additional burden on any other node, including the bus manager, for management of the capability.
- It requires no changes to non-privilege-capable PHYs.

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. (Though it is axiomatic to say that the isolation of these clusters by means of bridges is preferable, cost constraints will likely make bridge-less configuration common.)

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. Extending the right to disable privilege on any particular node to all nodes is beneficial since the device most capable of determining what bus behavior is most desirable – a PC – may not be the bus master, even when present in the bus. Also, in configurations with two or more PCs it may be necessary for individual PCs (or their users) to determine if aggregate (or individual PC) performance will be best served by disabling privilege for some or all nodes.

Bridging to the Future

One class of device that needs special consideration is bridges, both as the simple device and as the broader class including routers. In order to keep the cost of such devices as low as possible, and to obtain the best practical performance, such devices need to move data through their queues as expeditiously as practical. Such considerations may well make it necessary to define higher privilege classes, and to grant exemption from the "Thou shalt not inhibit writes/clears of the privilege enable bit," rule.