Mixed P1394a and P1394b Bus Issues

Dave LaFollette

4/13/98

Overview

This paper contains a collection of issues and thoughts to consider to enable P1394a and P1394b nodes to operate together. Most of these issues have not been resolved, but should be taken into consideration as the BOSS proposal is further developed.

Definitions

legacy	any combination of 1394-1995 and P1394a
slow traffic	traffic at 400Mbit/S or less which may be seen by legacy nodes
fast traffic	traffic at 800Mbit/S or more which is not seen by legacy nodes
border nodes	nodes with beta and legacy connections

Subaction Gaps

A legacy (1394-1995 or P1394a) bus must have subaction gaps to guarantee all nodes send their asynchronous arbitration requests. The legacy domain must see all slow isochronous traffic before a subaction gap. The legacy domain enters the asynchronous phase when a subaction gap is detected.

A mixed bus must wait for a subaction gap after a slow asynchronous packet unless that packet was the end of a subaction. This is the same situation as ack-accelerated arbitration in P1394a. The beta domain must wait if a legacy node is finishing the subaction.

It may be possible to synthesize gaps on the legacy bus during fast traffic. Subaction gaps would have to be punctuated by null packets to prevent unintentional arbitration reset gaps. Some beta packets may be too short to translate into a null packet on the legacy domain. This is another reason border nodes might generate idle time and null packets on the legacy bus. The null packets would not directly correspond to packets on the beta bus. If this type of solution is used, beta nodes will have to account for the presence of miscellaneous null packets in the legacy domains. Slow traffic following fast traffic may have to be timed to allow these null packets to complete before the slow traffic can be propagated into the legacy domain. This might be accomplished by lengthening the data prefix on the slow traffic.

Arbitration Reset Gaps

Legacy domains must see an arbitration reset gap to recognize the next fairness interval. The only way to know the legacy bus has sent all its current interval requests is to see an arbitration reset gap. Again, gaps may be synthesized during fast traffic.

It may be easier to let legacy and beta domains have separate fairness intervals. Priority could be given to legacy traffic, but the more efficient beta nodes might be able to move through their own fairness intervals more quickly. Each legacy domain might have its own fairness interval. Care must be taken not to starve any nodes. Priorities could change as fairness intervals occur in either domain.

Requests

Because of the difficulty in gathering legacy requests, legacy requests should probably be given priority over beta requests. This strategy allows the legacy traffic to complete as quickly as possible.

Border nodes could make early legacy requests on behalf of their legacy domains. This simplistic approach guarantees the legacy nodes are serviced, but may make it more difficult to generate the legacy gaps during fast traffic. Beta nodes could proceed as normal whenever there were no legacy requests.

Legacy requests should be marked distinctly from beta requests. A beta only bus should operate completely in BOSS mode even if some devices are sending slow traffic. This may be the case in a beta only bus using some slow connections for long distance.

Legacy Parent

Beta nodes can try to become root. However, there may be multiple beta domains separated by legacy nodes. In this case, some beta nodes must have legacy parents. It might still be possible for beta nodes to use BOSS arbitration when they win the bus by asserting data prefix to the parent. It would be difficult to get much performance gain in any beta domain with a legacy parent.

Beta nodes could assert data prefix to the parent only as long as they have no packets to go into the legacy domain. Any slow traffic might need to pass through legacy nodes even if it is intended for another beta node in another beta domain.

The beta nodes would have to release the bus to allow the cycle start packet. They would also have to observe arbitration reset and subaction gaps, as they could not synthesize idle time in the legacy domain.

Gap Count

Beta nodes and legacy nodes have different needs for gap count. Gap count is related to the maximum round trip propagation delay across the 1394 bus. Legacy nodes use this to determine the length of gaps to adjust arbitration timing. Beta nodes using the BOSS method essentially do away with gaps. Beta nodes may still need knowledge of bus propagation times to set optimal timeouts. Alternatively, if timeouts are determined to be rare (such as the lost GRANT case), the timeout could be set very large without significantly affecting performance.

It is conceivable that different legacy domains could have individual gap counts. However, there may be reasons not to optimize gap count on mixed busses. The problem of ending the legacy isochronous phase in a bus with more than one legacy domain is an interesting example. Each border node knows its legacy domain is done sending isochronous traffic when a subaction gap occurs. However, the border node can not let the isochronous phase end until it knows there is no more isochronous traffic coming from other legacy domains. Neither border node can be the first to let the isochronous phase end. This might be resolved by using a large gap count. Then a border node can let a gap occur that is long enough to indicate the legacy domain has no more isochronous requests but short enough that the legacy nodes do not see a subaction gap. Setting an extra large gap count increases the difference between the smallest subaction gap and the largest isochronous gap. The gray zone between these two times might allow the border node to observe and prevent the impending subaction gap.