An alternative root contention resolution method

Alistair Coles, Eric Deliot
Hewlett-Packard Laboratories
Filton Road
Bristol BS12 6QZ
U.K.

anc@hplb.hpl.hp.com
ed@hplb.hpl.hp.com

Root contention may be resolved prospectively during the start up procedure followed by 1394B nodes. During start up 1394B nodes negotiate to determine which of the two nodes will become root in the event that a contention should occur. Negotiation uses the method of exchanging bids described below. The advantages of this technique are:

Negotiation occurs only once, not at each bus reset.

When contention occurs, no time overhead is incurred.

Round trip delay incurred by multiple attempts to resolve contention can be avoided by using an event driven rather than timed process.

Bidding method

During 1394B Beta mode start up, connected 1394B ports exchange bids. A bid is an indication that in the event of a root contention occurring after bus reset, a node wishes to become either root or the child of the root. Thus two types of bids are defined. The two nodes issue a bid and then wait to receive a bid from the peer node. If the received bid is different from the last issued bid, then the contention has been resolved (or rather, the negotiation as to how future contentions will be resolved has completed). If the received and issued bids are the same, a new bid is issued.

Beta mode start up is defined as all the signaling between 1394B nodes up to the point that Reset and Tree ID begin. During this start up phase 1394B node exchange control signals that allow synchronization and negotiation of the highest mutually acceptable operating speed. It is proposed that the bidding also occurs during this phase (i.e. before the first Bus reset), concurrent with the speed negotiation.

Bids would take the following form:

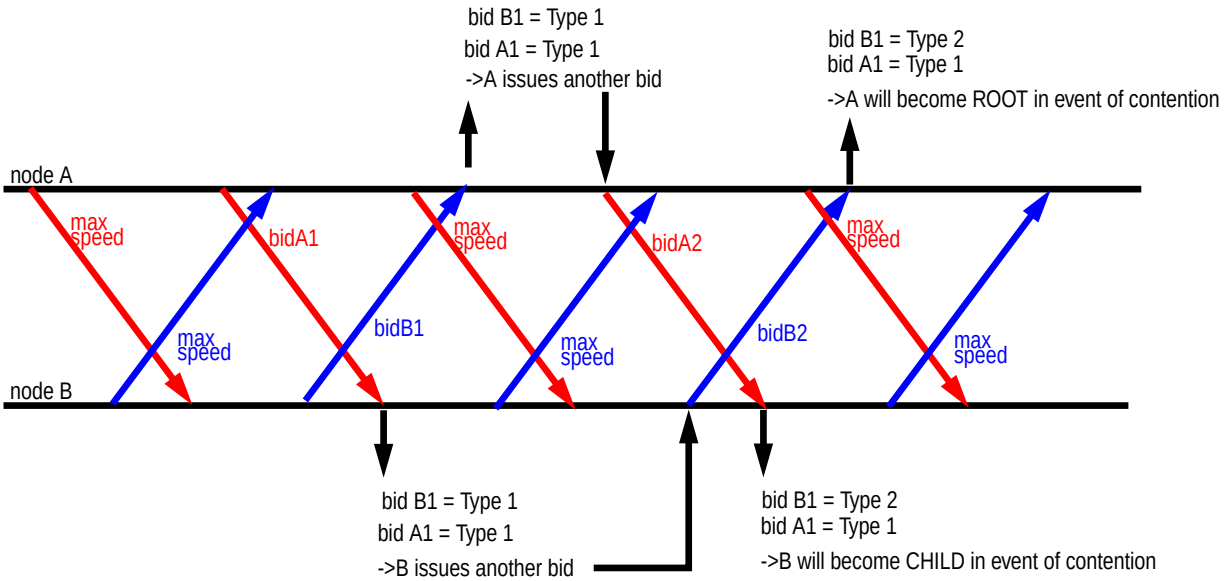
> Type 1: [ESCAPE, CHILD_NOTIFY] meaning “Issuing node will become root in event of contention”
>
> Type 2: [ESCAPE, PARENT_NOTIFY] meaning “Issuing node will become child of root in event of contention.

ESCAPE, PARENT_NOTIFY and CHILD_NOTIFY each indicate one occurrence of a 10 bit code character representing those control states. Bids are therefore 20 coded bits long (equivalent to two byte times). ESCAPE is a special control state defined for the modified 8B10B scheme.

Note: Although the method proposed here uses the properties of the modified 8B10B coding scheme, this coding scheme does not rely upon this method of root contention and works with other recent root contention proposals.

Example: two bids are exchanged before contention is resolved. After bidding is complete (and speed negotiation is complete), nodes either change to higher speed or move to Reset and Tree

ID.

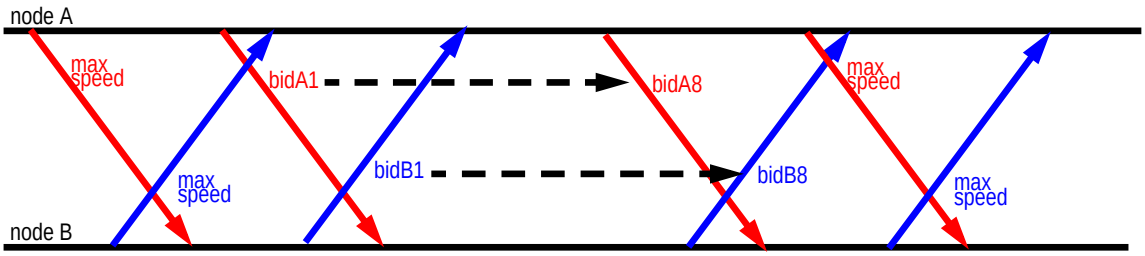


Elimination of round trip delay

The bidding method can be enhanced by having each node issue a burst of say 8 bids sequentially. Each node then compares consecutive received bids with issued bids until a complementary pair of bids occurs. At this point the negotiation is complete, and subsequent bids are ignored.

In this way, the round trip delay incurred whenever an exchange of bids fails to resolve the contention can be eliminated in all cases except those where each node generates 8 identical bids (small probability). In these cases a new set of 8 bids is issued by each node.

Example: Bursts of eight bids are exchanged. Contention is resolved by the fourth pair of bids. After bidding is complete (and speed negotiation is complete), nodes either change to higher speed or move to Reset and Tree ID.



i	bid Ai	bid Bi	outcome
1	Type 2	Type 2	no result
2	Type 1	Type 1	no result
3	Type 1	Type 1	no result
4	Type 2	Type 1	A = child, B= root
5	Type 1	Type 2	don't care
6	Type 2	Type 1	don't care
7	Type 2	Type 2	don't care
8	Type 1	Type 2	don't care

When root contention occurs during tree ID: assuming outcome of negotiation during start up is as previous example, when node A detects contention it maintains PARENT_NOTIFY in order to become child of root. Node B detects contention and immediately changes signal to CHILD_NOTIFY. When node A receives CHILD_NOTIFY it changes PARENT_NOTIFY to IDLE and Tree ID process ends as normal.