SBP-2 in a bridged environment

Peter Johansson
Congruent Software, Inc

NCITS T10 SBP-2 Study Group
Huntington Beach, CA
15 September, 2000
Bridge model

- Common cycle clock (synchronized to one portal)
- Distributed resources *per* portal
  - Configuration ROM (distinct EUI-64’s)
  - Buffers, CSRs and control logic
  - Asynchronous / isochronous routing information
- Internal fabric implementation-dependent
Bridges change IEEE 1394 assumptions

- Remote addressing errors
  - Bus ID invalid
  - Path temporarily unavailable
- Discard virtual IDs upon notification
  - Initiated by bus reset, message follows
- Remote transaction timeouts
  - Longer than local split timeout
- Cannot rely upon bus reset for synchronization
Virtual node IDs

- Each bus in the net has a unique 10-bit ID
- Each node on a bus has a unique 6-bit ID
  - Portals map between assigned virtual_ID and the node’s PHY ID generated by bus reset
  - Master mapping maintained by alpha portal
- Virtual node IDs do not have to be refreshed
- Invalid virtual node IDs return address errors
Remote split transaction timeout

- All remote transactions are split
- Each bridge along the path can delay the packet
  - Outbound for request, returning for response
- $2 \sum \text{MAX\_FORWARD\_TIME} + \text{SPLIT\_TIMEOUT}$
- Remote timeout determined by bridge “hops”

<table>
<thead>
<tr>
<th>Bridge hops</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote timeout (sec)</td>
<td>7</td>
<td>13</td>
<td>26</td>
<td>51</td>
<td>103</td>
<td>410</td>
</tr>
</tbody>
</table>

- Assume 0.2 seconds for each timeout
- Pathological topology exceeds 6 minutes!
Bridge-aware device timeouts

- **Separate time limits for each split transaction**
  - SPLIT_TIMEOUT for local requests
  - Remote timeout for requests sent to other buses

- **Transaction label (tlabel) cannot be reused until timer expires!**
  - Because remote timeout may be very long, this eliminates some simplistic approaches to transaction label reuse
SBP-2 synchronization and bridges

- SBP-2 relies upon bus reset to synchronize state between controller and target
  - Also applies to its derivative, IEEE P1394.3
- Virtual IDs (by themselves) are insufficient to extend the usefulness of legacy protocols
  - Bus reset changes the state of the target
  - Controller is unaware of these changes
Target actions after bus reset

- **Current SBP-2 behavior upon bus reset**
  - Clear all task sets
  - Commence reconnect timers for initiators
- **Desired behavior of bridge-aware SBP-2**
  - Clear task sets for local initiators, only
  - No affect on logins from remote initiators
Net topology examples

- Initiator uses virtual ID for target
- Target uses virtual IDs for initiator and buffer
- Bus reset does not necessarily alter virtual IDs
- Bus reset plus “revalidate” message indicates necessity to recorrelate EUI-64 with virtual ID
Net topology examples (cont.)

- Initiator uses virtual IDs for buffer and target
- Target uses virtual ID for initiator
- Target uses local ID for buffer
  - Target responsible to track physical ID of buffer
Initiator and target use virtual ID for buffer

Initiator and target use local ID for each other
- Target monitors physical ID of initiator
- No RECONNECT necessary after bus reset
Initiator uses virtual IDs for buffer and target
Target uses virtual IDs for initiator and buffer
Invalid initiator / buffer virtual ID

- Initiator originally on bus ID 42
  - ORBs constructed to reflect this
- Net topology change (not shown) “dirties” bus ID
- Initiator bus ID changes to 5
- Necessary to reestablish correlation between EUI-64 and virtual ID for initiator
Initiator revalidates virtual IDs?

- Target CHECK CONDITION fault
  - Read or write of initiator memory \texttt{resp_addr_error}
- Target task set aborted by the fault
  - Need to communicate status to initiator
- Unfortunately, initiator’s \texttt{status\_FIFO} address may no longer be valid
- This approach is unworkable!
Target revalidates virtual IDs

- Target differentiates local and virtual IDs
  - Local node ID is $\text{FFC0}_{16}$ OR’ed with any 6-bit ID
  - Virtual node ID is anything else
- No CHECK CONDITION on remote address error
  - Target’s task set is not aborted
- Target uses DEP methods to rediscover virtual ID for a particular EUI-64
Fly in the ointment ...

- **Legacy SBP-2 targets unaware of buffer nodes’ EUI-64s**
  - Targets monitor initiator EUI-64, only
  - Work-around possible in typical case where initiator and buffer are identical

- **Possible solution: identify remote buffer addresses by a 16-bit “handle”**
  - New management functions to create handles and associate them with EUI-64

- **Target determines corresponding virtual ID**
Work-arounds for legacy targets

- **Design support into the bridges themselves**
  - Increased design complexity for bridges
  - “Tax” on bridges to solve out-of-scope problems
  - Potentially unbounded problem space

- **Service proxy on same bus as legacy target**
  - Enabled by 1394 TA DEP specification
  - Service proxy fully bridge-aware
  - Remote bridge-aware controller interacts with service proxy exactly as with native target
Service proxy for a legacy target

- Service proxy implements revised, bridge-aware extension of protocol used by target
- Controller issues commands to proxy
- Proxy issues commands to target
- Direct data flow between controller / target best
  - Avoids secondary copy by proxy
Service proxy for an SBP-2 target

- Service proxy fetches O RBs from initiator and signals them to target
  - Target fetches ORBs from proxy but …
  - Buffer pointers still reference source, not proxy
- Efficient data flow between target and buffer
- Upon bus reset, proxy adjusts O RBs and signals them to target—no task set abort in proxy!
Information sources

- **Working group reflector**
  - STDS-1394-1@IEEE.org
  - Majordomo@Majordomo.IEEE.org (subscriptions)

- **Working group web site**
  - http://grouper.ieee.org/groups/1394/1/

- **Working group chair**
  - Peter Johansson
  - PJohansson@ACM.org
Contact information

Peter Johansson
Congruent Software, Inc.
98 Colorado Avenue
Berkeley, CA  94707

(510) 527-3926
(510) 527-3856 FAX

PJohansson@ACM.org