Timerless missing ACK

Michael Johas Teener

ZAYANTE

Summary

- IEEE 1394-1995 and “a” have “managed timers”
  - Requires considerable knowledge of delays to optimize
  - Does not scale to new models
- P1394b removes most managed timers:
  - Wait for subaction gap to start arbitration (pipelined in B)
  - Wait for arb reset to guarantee fairness (pipelined current/nextA/nextB)
- Remaining timers not often used, *can* used fixed value:
  - Missing ACK (directed packet, missing destination or bad header CRC)
  - Lost grant

ZAYANTE

Personal philosophy

- Make all timeouts very, very robust
  - Not closely aligned with any physical requirements
  - Don't like to crowd the margins
- Implication is that all timeout limits should be very large
  - Any timeout will result in long idle bus
- Restrict use to error conditions that will never appear in normal operation

ZAYANTE

Missing ACK will happen in normal operation

- All types of probing operations can result in missing ACKs
  - Also software errors, which are much more likely than hardware failures
- Bad header CRC is an error condition, not concerned with this case
- Lost grant is an even more extreme error condition, not concerned with this case

ZAYANTE

Goals

- Positive, timerless detection of packet addressed to non-existent node.
- Non-goal: detection of corrupted packet header.
- Timer still needed for corrupted packet header, lost ACK, arbitration timeouts for other lost tokens.

ZAYANTE

Timerless detection of missing ACK (1)

All Links that do not respond with an ack to an explicitly addressed packet instead respond with a “NACK”

- The attached PHY shall send a NACK token back to the packet sender

All PHYs that do not respond to a PHY request packet shall send a NACK token back to the requester.

Timerless detection of missing ACK (2)

- When a PHY receives a data packet, it will record the direction that the packet came from, repeat the packet as appropriate and clear all NACK-received flags on each port (including the link port).
- If a PHY receives a NACK on all of its ports on which it transmitted the previous packet (including an active link), it will forward the NACK to the port or link that originated the packet.
- When a PHY or link that originated a packet receives a NAK on all ports, it knows it has a missing acknowledge and can do the appropriate processing.

ZAYANTE

Misc.

Border issues

- Border PHYs synthesize a NAK on subaction detection.

What is a NAK?

- A “NAK” is a control token of modest robustness. No big problem if lost (just falls back to worst case timeout).
- Two unassigned tokens now, plus various overloading possibilities (reverse direction GRANT, isolated DATA_END, etc.)

ZAYANTE