Protocol Adaptation Layer (PAL) for IEEE 1394 over IEEE 802.15.3 ("wireless 1394")

Peter Johansson
Congruent Software, Inc.
IEEE 802 Tutorial
Portland, O R
July 13, 2004
What is a PAL?

- A “glue layer” on top of a lower level
- Hides low-level details of underlying layer
- Mimics high-level behavior of target protocol
- For example, IP1394 is a PAL that permits Internet protocol to be carried by IEEE 1394
What use is a PAL?

- Leverages applications already developed

- Applications developed for IEEE 1394 expect:
  - Read, write and lock transactions
  - Infrastructure CSRs and configuration ROM
  - Asynchronous and isochronous streams
Wireless products enabled

- Firmware developed for (wired) IEEE 1394 products can migrate to wireless domain
- Minimize reengineering between wired and wireless domains
1394 PAL ground rules

- Shall support IEEE 1394 TRAN layer functions
  - Read, write and lock
- Shall support isochrony and streaming data
- Shall coexist with other users of the underlying IEEE 802.15.3 transport
- Should behave “like” IEEE 1394
- Should conceal differences between IEEE 1394 and IEEE 802.15.3 physical and MAC layers
1394 PAL for IEEE 802.15.3 permits wireless devices to talk to each other

Not always interesting unless wireless devices can also talk to (wired) IEEE 1394 devices
Wired to wireless via bridges

- IEEE P1394.1 bridge isolates physical and link (MAC) layer differences in each domain from the other

- IEEE P1394.1 preserves TRAN layer similarities
  - Transaction routes configured autonomously
  - Explicit isochronous stream setup / tear down
1394 TA project scope

Develop a document that specifies methods to:

a) mimic IEEE 1394 infrastructure (transactions, isochrony, stream data, configuration ROM and CSR architecture) using the facilities of IEEE 802.15.3

b) implement IEEE P1394.1 bridge behaviors in the same domain. The methods are to be compatible with the simultaneous use of IEEE 802.15.3 by other protocols, e.g., Internet protocol.
Summary of PAL features

- Virtual bus management
- IEEE 1394 packet encapsulation
- Split transaction time-out
- Cycle time synchronization
- Isochronous streams
- Isochronous connection management
- Control and status registers
- Configuration ROM
Virtual 1394 bus within a piconet

- **Wireless 1394 coordinator co-located with PNC**
  - Assigns 6-bit virtual IDs to wireless 1394 DEVs
  - Distributes synchronized cycle time
Bringing a new device home

- New, out-of-box device doesn’t know it should associate with your piconet
- Possible metrics might choose incorrectly
  - Radio signal from your neighbor’s PNC might be stronger than your PNC’s signal
Introducing a device to the right piconet

- New device flashes amber LED (disconnected)
- User first presses HANDSHAKE button on the PNC, next the new device’s HANDSHAKE button
- Wireless 1394 coordinator admits only the one device whose HANDSHAKE button was pressed
Factory-configured components

- **DVD player is both PNC and wireless 1394 coordinator**
  - Access Control List (ACL) factory-initialized with MAC-64 IDs of wireless DTV and speakers
  - DTV and speakers automatically connect to virtual 1394 bus coordinated by the DVD player
Set DVD player’s COORDINATOR switch off and STB’s COORDINATOR switch on.

Introduce DVD player to STB (via HANDSHAKE).

DVD transfers its ACL to the STB (DTV and speakers automatically connect to virtual 1394 bus coordinated by the STB).
IEEE 1394 packet encapsulation

- Protocol ID identifies format that follows
  - Used only with 802.15.3 stream index zero
- 1394 SDU analogous to IEEE 1394 packet
  - Multiple 1394 SDUs permitted in a single 802.15.3 MSDU (isochronous data)
The 3-octet LLC header indicates that a SNAP header follows.

The 5-octet SNAP header consists of an OUI (company_ID) and a Protocol ID:
- 00 A02D_{16} is 1394 Trade Association OUI
- 201_{16} specifies IEEE 1394 over IEEE 802.15.3
**PAL header for 1394 SDU**

- **Transaction code (tcode) determines format**

<table>
<thead>
<tr>
<th>tcode-dependent</th>
<th>expiration_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination_ID</td>
<td>tcode-dependent</td>
</tr>
<tr>
<td></td>
<td>tcode-dependen</td>
</tr>
<tr>
<td></td>
<td>tcode-dependen</td>
</tr>
</tbody>
</table>

- 0, 1, 4, 5, 9  Request
- 2, 6, 7, B_{16}  Response
- A_{16}  Stream (asynchronous or isochronous)
- E_{16}  PAL control
### PAL header for requests

Data payload never present for read requests

For write and lock requests, data payload present only if data_length is nonzero

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination_ID</td>
<td></td>
</tr>
<tr>
<td>source_ID</td>
<td></td>
</tr>
<tr>
<td>destination_offset</td>
<td></td>
</tr>
<tr>
<td>data_length</td>
<td></td>
</tr>
<tr>
<td>data (optional)</td>
<td></td>
</tr>
<tr>
<td>expiration_time</td>
<td></td>
</tr>
<tr>
<td>tl</td>
<td></td>
</tr>
<tr>
<td>tcode</td>
<td></td>
</tr>
<tr>
<td>pri</td>
<td></td>
</tr>
<tr>
<td>extended_tcode</td>
<td></td>
</tr>
</tbody>
</table>
PAL header for responses

- Data payload never present for write responses
- For read and lock responses, data payload present only if data_length is nonzero
**PAL header for stream data**

- **Nonzero isochronous bit specifies isochronous stream**
  - Data indicated to application at cycle_time
  - Permits multiple isochronous packets in a single 802.15.3 MSDU
- **Data payload present only if data_length is nonzero**
# PAL control header

<table>
<thead>
<tr>
<th>signature</th>
<th>expiration_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination_ID</td>
<td>result</td>
</tr>
<tr>
<td>data_length</td>
<td>timeout</td>
</tr>
<tr>
<td>data (optional)</td>
<td></td>
</tr>
</tbody>
</table>

- **Used for control messages exchanged by PALs**
  - Virtual bus management
  - Isochronous connection management
- **Data payload present only if data_length is nonzero**
Split transaction time-out

- **All wireless 1394 transactions are split**
  - Request subaction in one MSDU, response subaction in different MSDU

- **Difficult to control how much time the 802.15.3 MAC will use to transmit or retry an MSDU**

- **Split time-out controlled by expiration_time**
  - Transmitting PAL calculates expiration_time for requests by adding SPLIT_TIMEOUT to cycle time
  - Receiving PAL discards requests and responses if expiration_time is earlier than cycle time
Cycle time broadcast in beacon

<table>
<thead>
<tr>
<th>element_ID (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
</tr>
<tr>
<td>OUI (00 A02D_{16})</td>
</tr>
<tr>
<td>beacon_number</td>
</tr>
<tr>
<td>previous_bus_time</td>
</tr>
<tr>
<td>previous_cycle_time</td>
</tr>
</tbody>
</table>

- Each beacon contains an Application-Specific Information Element (ASIE) with master cycle time
- ASIE serves additional purposes:
  - Identifies presence of wireless 1394 coordinator
  - Supports HANDSHAKE mode for DEV introduction
Cycle time synchronization

- Upon MLME-BEACON-SYNC.indication, all wireless 1394 devices simultaneously sample CYCLE_TIME.
- Wireless 1394 coordinator transmits its sample in the next beacon.
Isochronous aggregation by transmitting PAL

- AV application informs PAL of optimal averaging window size for isochronous stream
- PAL accumulates at least this much data before encapsulating multiple 1394 SDUs in a single MSDU
Isochronous replay by receiving PAL

Aggregated isochronous payload (2444 octets) requires approximately 200 µs to transmit
- Assume 100 mbps UWB PHY

Earliest possible MSDU receipt for cycle 11
- No allowance for retries

End-to-end latency of 1.25 ms for this example
Isochronous connection issues

- Talker requests channel time from PNC
- Talker transmits on assigned stream index
- Listener receives from same stream index
- How does controller manage this?
Wireless input and output plugs

- Patterned after IEC-61883 iPCR and oPCR— but different
- Internal PAL data structure (not exposed as a CSR)
  - 802.15.3 stream index
  - Connections
  - Maximum and average data payload
  - Latency
  - Averaging window
- Correlate stream index with device’s internal destination or source for isochronous data
Plug management SDU

- Supports two functions
  - PLUG and UNPLUG
- Recipient may reject function if latency too great
Controller transmits JOIN message to wireless 1394 coordinator at PNC
- Average payload, averaging window and latency

Wireless 1394 coordinator transmits PLUG request to talker
- Remembers JOIN information to monitor stream
Channel time allocation by talker

- Talker requests channel time from PNC
  - If channel time is available, PNC confirms request and assigns a stream index to identify the stream
- Talker confirms PLUG request and informs wireless 1394 coordinator which stream index will be used
- Coordinator sends PLUG request to listener
Connection established: STREAM STATUS message

After receiving PLUG confirmation from listener, wireless 1394 coordinator returns STREAM STATUS message to controller
- Channel time allocated by PNC
- Both talker’s and listener’s plugs programmed

No isochronous data flowing yet
Isochronous connection in use

- Controller issues “receive” command to listener
  - NOTE: AV/C monitors may not require a command
- Controller issues “transmit” command to talker (e.g., AV/C DIRECT SELECT command to a tuner)
- Isochronous data flow commences
Control and status registers

- Most CSRs are not applicable to wireless
- Wireless 1394 devices implement these CSRs
  - RESET
  - MESSAGE_REQUEST and MESSAGE_RESPONSE
  - CYCLE_TIME and BUS_TIME
- New CSRs specified by IEEE P1394.1 and wireless PAL
  - QUARANTINE and NET_UPDATE
- Bridge portals implement additional CSRs specified by IEEE P1394.1
Configuration ROM

- Supports all features specified by IEEE 1212 and IEEE 1394
- “Wireless 1394” differentiated by special entries in configuration ROM
  - ASCII bus identifier in bus information block
  - Bus-dependent information directory
**Bus information block**

- ASCII bus identifier is “0000”
- Consult bus-dependent information directory for more detail

<table>
<thead>
<tr>
<th>30₁₆ (“0”)</th>
<th>30₁₆ (“0”)</th>
<th>30₁₆ (“0”)</th>
<th>30₁₆ (“0”)</th>
</tr>
</thead>
<tbody>
<tr>
<td>capabilities</td>
<td>cycle_clk_acc</td>
<td>max_rec</td>
<td>mxr</td>
</tr>
<tr>
<td>node_vendor_ID</td>
<td></td>
<td>chip_ID_hi</td>
<td></td>
</tr>
<tr>
<td>chip_ID_lo</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bus information block capabilities

- Isochronous resource manager, bus manager and power manager have no analogs in “wireless 1394”
  - irmc, bmc and pmc bits are zero
- IEEE P1394.1 cycle master adjustment methods don’t work for wireless
  - adjustable bit is zero
- Cycle master capability is optional
  - cmc bit may be zero
Specifier_ID, Version and Revision entries identify the 1394 TA document that specifies the wireless PAL.

Wireless_Plugs describes the input and output wireless plugs implemented by the device.
1394 Trade Association Specification
- TS2003010 (May 7, 2004)
- Protocol Adaptation Layer (PAL) for IEEE 1394 over IEEE 802.15.3

Available for purchase
- http://www.1394ta.org/Technology/Specifications
Resources

1394 Trade Association
   http://www.1394ta.org

IEEE P1394.1

High Performance Serial Bus Bridges
   http://grouper.ieee.org/groups/1394/1
Contact information

Peter Johansson
Congruent Software, Inc.
98 Colorado Avenue
Berkeley, CA  94707
(510) 527-3926
(510) 527-3856 FAX
PJohansson@ACM.org