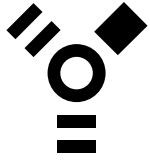
Playing with fire



Protocol analysis techniques for the Firewire® bus

Linux.conf.au 2011

Jonathan Woithe
(jwoithe@atrad.com.au)

1 Talk outline

- Introduction to Firewire
- The aim of the game
- The easy bit: capturing isochronous packets
- Capturing asynchronous packets
- Summary
- Acknowledgements

2 Introduction to Firewire

2.1 Properties

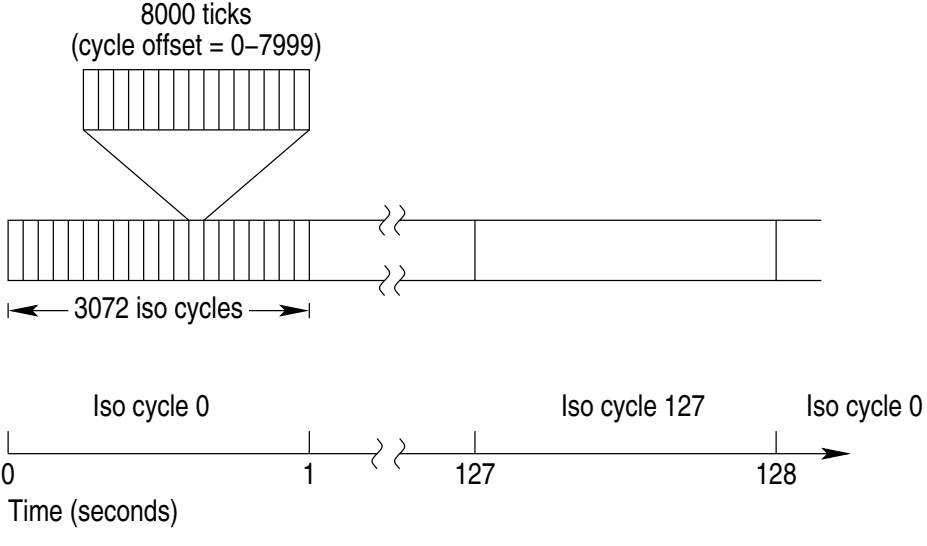
- High speed bidirectional serial bus.
- Speeds of 400 Mbps and 800 Mbps in use today.
- Up to 63 devices per bus.
- The bus implements a global address space.
- Each device is assigned a unique address range on the bus.
- Often abstracted as a register model – each device has a collection of individually addressable 32-bit (“quadlet”) registers.

2.2 Packet types

- Asynchronous (“async”):
  - Addressed to a specific bus address – that is, a particular address (register) on a given device.
  - No timing guarantees.
  - Used for device configuration and control.
- Isosynchronous (“iso”):
  - Broadcast packets, sent on one of 64 “channels”.
  - Guaranteed bandwidth and delivery timing.
  - Used for bulk data streaming (eg: audio, video).

2.3 Iso packet management

- A second is divided into 3072 “iso cycles”
- Each node can transmit one iso packet per iso cycle
- Timing within an iso cycle specified by a “cycle offset” (0-7999)
- An “iso clock” runs at 24.576 MHz (3072×8000) synchronously across the bus
- Iso clock drives an “iso counter” to count iso clock ticks for 128 seconds before wrapping around
- Device providing iso clock master (the “Isosynchronous Resource Manager”, or IRM) is negotiated by the bus
- A node must request its iso bandwidth requirements from the IRM. Can be denied if bus’s bandwidth already fully allocated.


8000 ticks
(cycle offset = 0–7999)
3072 iso cycles
Iso cycle 0
Iso cycle 127
Iso cycle 0
0
1
127
128
Time (seconds)

2.4 Firewire card types

- OHCI-compliant cards
  - Includes all commonly available firewire cards
  - Common interface for software drivers (quirks and firmware bugs notwithstanding)
- PCILynx cards
  - Utilise PCILynx chipset from Texas Instruments (originally designed for hardware bus analysers)
  - No drivers to permit 'normal' use
  - Very uncommon, but very useful for protocol analysis
  - IOI technology still sells a PCI version (IOI-1394TT) at "reasonable" cost (approx US$100 plus shipping).

3 The aim of the game

- Not all vendors of Firewire interfaces provide support for Linux driver writers.
- Sometimes it's useful to see the contents of packets when debugging vendor drivers.
- We want to:
  - Capture packets sent between a "supported" platform and a device
  - Observe data exchanges between a Linux driver and a device for debugging

4 The easy bit: capturing isochronous packets

- Iso packets are broadcast packets — can be seen by all nodes on the bus which care
- Can capture packets with 'normal' OHCI firewire cards
- No special hardware required.
- Example: using `dumpiso` from libraw1394:

  ```
  dumpiso firewire_dump.dat
  ```

  Use hex viewer and dumpiso(5) manpage to view.
- Example: using nosy-dump and a PCILynx card:

  ```
  nosy-dump --iso --cycle-start -v
  ```

  This dumps human-readable ascii/hex. Can also store to binary file ('-o' option) for replay later with '-i' option.
- Tools are opensource, so printout can be changed to suit protocol.

5 Capturing asynchronous packets

5.1 Why this isn't so easy

- Async packets are addressed only to receiver.
- OHCI specification mandates that packets to another address are dropped at hardware level.
- There are several approaches to work around this OHCI limitation.
- A useful fact: OHCI cards cannot capture traffic between two other bus nodes, but PCILynx cards *can*.

5.2 The general idea

- Essentially use Tridge's "French Café" technique:
  - Capture async packets sent in response to a specific device action
  - Deduce protocol for controlling that action based on packet contents
  - Repeat for each device action
- Some protocols are easier than others. For example: some devices provide unrelated functionality from a given register depending on whether it was read or written.

- Example: setting headphone source on a MOTU 828mk3 audio interface:
  - Set source to ‘ADAT-B 7-8’:

```
54379 dest=0xffc0, tl=0x14, write_quadlet_request, src=0xffc2,
  offs=0xfffff0000c04, data=0x0000011e, ack_pending
```

  - Set source to ‘Main outs’:

```
write_quadlet_request, ...  offs=0xfffff0000c04, data=0x00000110
```

  - Observe data quadlet change only in bits 3–0.
  - Conclude bits 3–0 of device register 0xfffff0000c04 control headphone source, with 0x0 selecting ‘main out’.

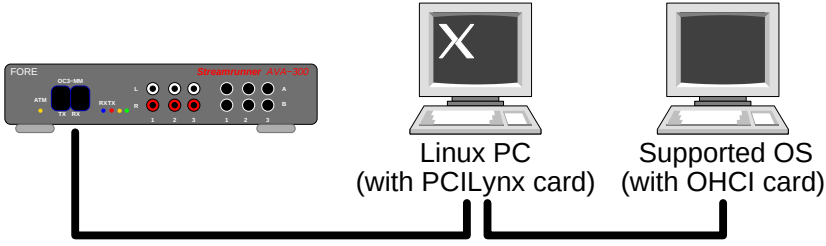
6 Options for asynchronouse packet capture

6.1 Hardware bus analysers

- Advantages
  - Independent of computers
  - Highly programmable and configurable
- Disadvantages
  - Very hard to come by
  - Extremely expensive
  - Often hard to drive
- Overall: not practically useful for most people

6.2 Using a PCILynx card

- The process
  - PCILynx PC placed between “supported OS” and firewire device



  - “Nosy” software used to capture packets (in mainline kernel as of 2.6.36).
  - Example: `nosy-dump -v`

- Advantages
  - Relatively cheap (US$100-400 depending on vendor).
  - Tools required are available under Linux.
  - Tools are opensource, so can be customised to suit specific analyses
- Disadvantages
  - Requires second PC.
  - “Special” PCILynx hardware is needed.

6.3 Using a suitable Power Mac

- 'Suitable' means a blue-and-white G3 or original "Yikes" G4 (with PCI graphics adapter). These used the PCILynx chipset.
- Use with Apple's Firebug software
- The process
  - Similar to PCILynx approach with slightly less flexibility
- Advantages
  - Some people have these lying around (!)
- Disadvantages
  - Somewhat rare and hard to find.
  - Hardware is dating and is hard to maintain.
  - Few people have done this so support is hard to come by.

6.4 Indirect analysis

- The process
  - OHCI-equipped Linux PC on firewire bus with “supported OS” and device.
  - Use Linux to query device registers.
  - Make a change on the supported OS.
  - Query device registers again and note differences.
- Example: setting headphone source to phones bus on MOTU Traveler
  - Use `scan-devreg` from ffado’s tests/ directory.
  - Start `scan-devreg`, then change phones source.
  - `scan-devreg` reports: `0x00000c04 changed from 00000504 to 00000501`

- Advantages
  - No special hardware or additional PCs required.
- Disadvantages
  - Some devices are more suitable than others (registers must implement read-back).
  - No information provided about ‘write enable’ bits.

6.5 Packet capture on “supported” platforms

- Example: bushound (`http://www.perisoft.net/bushound/`)
- The process
  - Install capture software on supported OS.
  - Control the device, capturing packets generated in response.
- Advantages
  - No additional hardware required beyond the “supported” platform.
  - Firewire bus topography is ‘normal’.
- Disadvantages
  - No FOSS capture software for common “supported” platforms.
  - Software required is often very expensive (US$700+).
  - Software which is free of charge is usually crippled in some way.

6.6 Using a virtual machine under Linux

- Advantages
  - No additional hardware required beyond a Linux PC.
  - In theory, potentially as useful as a PCILynx card.
- Disadvantages
  - No FOSS VM implements Firewire passthrough yet.

7 Summary

- Capturing packets on a firewire bus is not overly difficult.
- PCILynx cards are generally affordable and provide a flexible capture system.
- Other alternatives exist, albeit with less flexibility.
- It would be great if an open source virtual machine implemented firewire passthrough at some point.

8 Acknowledgements

- The trademarks of companies referred to thoughout this presentation are acknowledged
- FireWire and the FireWire symbol are trademarks of Apple Inc., registered in the U.S. and other countries. The FireWire logo is a trademark of Apple Inc.

9 Links

- FFADO project: `http://www.ffado.org`
- Nosy: part of mainline kernel as of 2.6.36
- libraw1394:
  `https://ieee1394.wiki.kernel.org/index.php/Libraries#libraw1394`
- IOI technology: `http://www.ioi.com.tw/`
- Bushound: `http://www.perisoft.net/bushound/`

Contacting me:

- jwoithe@physics.adelaide.edu.au