AVB L3 Transport SG

Aidan Williams

<aidan.williams@audinate.com>

Agenda

- Admin
- Problem statement
- Requirements
- Existing protocols
- Discussion
- Next steps

Admin

- AVB Transport Protocol Study Group
  - Is now a “Working Group”
  - Robert has asked for a new SG:
    - “AVB L3 Transport Study Group”
    - Work on PAR => WG
- Minutes
  - Volunteer for today needed…
- Editor
  - TBD, contact me if you’re interested

Goals for today

- Start discussion on:
  - Clearly defining the problem we're trying to solve
  - Identifying the requirements for AVB L3 transport
  - Evaluating various IP protocols that could be used in this space
  - Identifying the work that needs to be done

Problem Statement (1)

Next Generation AV Connector

Clean and Simple

Problem Statement (2)

- All those sockets are used to make connections between boxes
  - Replace them with "*virtual sockets*" and "*virtual wires*" running over a network
- Various media sources
  - DVD players, STBs, Computers, Hi Fi gear
- Various sinks / playout devices
  - Loudspeakers, displays, processors (sink+src)

- Our task: design an IP network system to get digital signals from the sources to the sinks
  - Time aligned, rate controlled, high quality transport
  - At least as easy to use as plugging in a cable!

Problem Statement (3)

- Some things that AVB L3 TP is *not*:
  - Transparent bridging of 1394 over IP
    - Exploit existing IETF protocols (e.g. RTP) as much as possible, analogous to L2-TP use of 61883
  - A file transfer protocol
    - Support analog inputs, “live” signals (i.e. not pre-recorded)
  - Finding media on boxes in a home network
    - E.g. recorded TV, movie files, MP3 files, etc
  - Remote control for devices
    - In a Hi-Fi, the copper connections between the devices exist before you hit the play button on your remote control
    - A remote control *might* cause network connections to be set up, but that shouldn’t be assumed
  - Need to avoid going too far up the stack

Problem Statement (4)

- Necessary network pieces:
  - High quality clock sync
  - QoS: admission control, traffic shaping
  - Automatic management of IP network details
    - Zeroconf: Addressing, mDNS, etc
  - Discovery of "*virtual sockets*"
  - Connection management for "*virtual wires*"
  - Media transport mappings

Requirements (1)

- “L3 Transport” implies the Internet Protocol suite
- Support simple, hardware-oriented implementations
  - E.g. chips
  - UDP/IP is easier to handle in hardware than TCP
  - Discovery and connection management should be simple to implement, not requiring TCP
- Multicast transport
  - UDP, by definition
- Unicast transport
  - Routable, many networks don’t support multicast routing
  - Lots of signals are point to point
  - Unicast works a *lot* better on some link layers than multicast (e.g. wireless)

Requirements (2)

- Re-use as much as possible
  - Lots of media transport mappings exist in the RTP world
  - The wheel does not need to be re-invented..
- Provide a transition story
  - Today's networks are largely IP and don't support the various AVB services
  - How do they migrate over time to support AVB services?
  - Can we usefully operate in a degraded mode on a non-AVB network?
  - What about networks with some AVB features, but not all?
- Support un-managed networks
  - Plug and play operation, like L2 bridges

Requirements (3)

- Scale to larger, managed networks
  - Avoid designing-in arbitrary limitations
    - E.g. a maximum supported end-to-end latency
  - Support DHCP address management, DNS
    - This will probably used by home networks too
  - Support centralised management of discovery and connection setup
  - Co-exist with things like firewalls

Existing Protocols (RTP 1)

- RTP consists of several pieces
  - RTP Data Transfer Protocol
    - Basic transport protocol header
    - Source time-stamping rather than presentation time
  - RTP Control Protocol (RTCP)
    - Sender relates media clock to reference clock
      - A shared reference clock is already available in AVB
      - Still need to send RTCP for non-AVB receivers
    - Receivers report quality
  - Session Description Protocol (SDP)
    - Essential information for the RTP session
    - Mime type => RTP packet payload type

Existing Protocols (RTP 2)

- Control protocols
  - RTSP – stop/start/pause, usually pre-recorded media
  - Interactive connection setup
    - SIP, H.323, …
- “RTP Profiles” map various media formats to RTP packets

Existing Protocols (RTP 3)

Some RTP Profiles:

Media Type	RTP Profile
16/24 bit LPCM Audio	RFC3190, RFC3351
RTP Payload Format for DV (IEC 61834) Video	RFC3189
RTP Payload Format for MPEG1/MPEG2 Video	RFC2250
RTP Payload Format for MPEG-4 Audio/Visual Streams	RFC3016
RTP Payload Format for BT.656 Video Encoding	RFC2431
RTP Payload Format for Uncompressed Video (BT.601, SMPTE 274M, SMPTE 296M)	RFC4175
RTP Payload Format for AC-3 Audio	RFC4184
… RTP Payload Format for MP3 Audio	RFC3119, RFC2250

Session Description Protocol

- Discovering the "*virtual sockets*"
- SDP, RFC2327
  - Describes RTP session characteristics
  - Gets embedded into other protocols
  - Straight-forward to map into service discovery
    - Advertise "*virtual sockets*"?
  - Easy to use for multicast that is always transmitting
    - Receiver just joins the correct addresses
  - Unicast needs signaling to get a flow started

SDP Example

- Multicast audio+video session
  - 2 separate RTP/UDP streams
  - Dynamic RTP protocol type (98) for **video/H263-1998**

```
v=0
o=bloggs 2890844526 2890842807 IN IP4 10.45.1.82
s=-
e=j.bloggs@example.com(Joe Bloggs)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 98
a=rtpmap:98 H263-1998/90000
```

Connection setup

- Connecting the "*virtual wires*"
  - Session Initiation Protocol (SIP), RFC3261
    - Commonly used for VoIP signalling
    - TCP and UDP variants
    - Text protocol, not friendly for hardware
    - Multi-step session setup
  - Real Time Streaming Protocol (RTSP), RFC2326
    - A control protocol (not a streaming transport protocol!)
    - TCP protocol only
  - Scope to develop a hardware-friendly, simple connection initiation protocol

Discussion

Next steps

- Using AVB sync/timestamps with RTP
  - Volunteer to do a presentation on this for the next call?
- Discuss problem statement and requirements on the reflector
  - I'll summarise the discussion at the next call