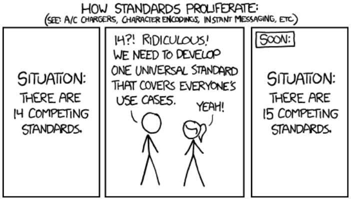
Using the Network as a Reliable Platform for Time-Sensitive Systems

Henrik Austad
haustad@cisco.com

Cisco Systems

Berlin, Oct 13, 2016


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

https://xkcd.com/927/

"This talk will dive into what TSN actually is, what it can provide and its benefits and finally what is currently missing in the kernel to support TSN."

about:henrik

- Software Engineer at Cisco's Telepresence group at Lysaker, Norway
- GNU/Linux-, OSS-enthusiast
- Realtime-/kernel-troubleshoo(t)ing
- Spend a lot of my time staring at traces thinking "huh?"
- Lives by "There's a script for that"
- Hardware-hoarder
- Currently working on the TSN driver for the kernel

TSN - Getting started

Source: `https://goo.gl/images/kZ987x`

Why we started with AVB

- AV setups can be a *lot* of cables
- Entropy always wins, especially so for cabling
- C90 and SX80 have fairly large backpanels, but not infinite (we always want more)
- A lot of potentionally unused AD/DA hw
- Analog cables require point-to-point, no trunking - inflexible!

Internal AVB demo

Each year R&D engineers shows off the most insane ideas to the other engineers.

- Take some networking gear
- a desktop computer with an i210 NIC
- an idle MX800D unit
- a somewhat enthusiastic engineer
- and blast AC/DC from Spotify via TSN/AVB on the MX800 speakers

Terminology

- AVB - Audio/Video Bridging
- TSN - Time Sensitive Networking
- Bridge, End Station, Talker & Listener
- (Time Sensitive) Stream
- Stream Reservation → guaranteed delivery
- Traffic prioritization → bounded latency
- SR Class A & B[1]
- gPTP-, SR-, AVB-*domains*

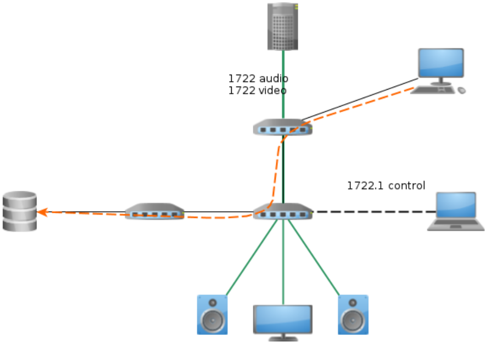
[1] and now also C & D

What is AVB good for?

- Sound and video is digital
- Allows for very flexible setups (analog is point-to-point)
- High audio-capacity
- Basic infrastructure
- Using open standards is the only sensible way
- Can use different network (802.3, 802.11v)
- Guaranteed delivery, not best effort

What is AVB good for?

- Sound and video is digital
- Allows for very flexible setups (analog is point-to-point)
- High audio-capacity
- Basic infrastructure
- Using open standards is the only sensible way
- Can use different network (802.3, 802.11v)
- Guaranteed delivery, not best effort



Evolved motivation - AVB → TSN

Soon became clear that AVB could do more than “just AV”

- Pro-AV
- Consumer AV
- Automotive (infotainment, system control, autonomous driving)
- Industrial applications (Control, Robotics, IIoT, “Industry 4.0”)
- Own protocol (see EF_STREAM[2] & EF_CONTROL)

... a bit like SCHED_DEADLINE for LANs...

[2] Experimental Format

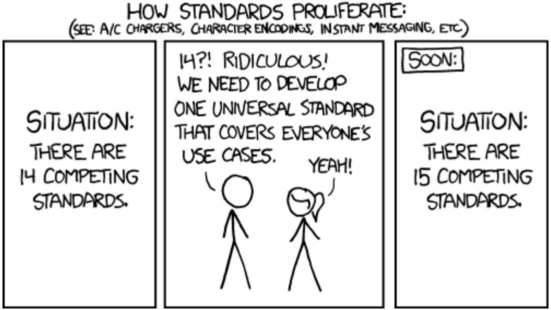
TSN - Details

u

Source: `https://goo.gl/images/13846A`

TSN

"A set of standards that govern the transmission of time-sensitive frames through a network".


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

`https://xkcd.com/927/`

Standards

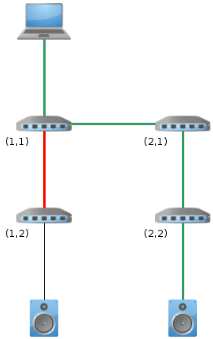
- IEEE standards
  - 802.1BA - AVB Systems
  - 802.1Q-2014 (Sec 34: FQTSS, Qav, 35: Stream Reservation, Qat)
  - 802.1AS-2011 (gPTP, also, 1588 v2)
  - 1722 / 1722a d16 AVTP (latest draft for new rev.)
  - 1733 (AVTP over RTP)
  - 1722.1 Discovery and enumeration (think plug'n'play-ish)
  - 802.1Qbu-2016 Frame preemption (bridges)
- IEC/ISO 61883 1-6 (Firewire)

Why L2 before L3?

- Simple End-Stations (why should a mic implement TCP/IP?)
- Pro-AV had little interest in very, *very* large networks
- multipath routing makes bounded latency difficult
- Must solve L2 before you can tackle L3
- Work in progress to get support for higher layer (IETF DetNet)

Stream Reservation

- MVRP: Declare membership to a VLAN, define SR class priority
- MSRP: Reserve network resources (up to 75%).
  - Talkers: available stream attributes
  - Listeners: acceptable attributes
- Success: all bridges OK
- Failure: *at least one* not OK
- Extraordinarily low packet-loss ratios
  $10^{-6} - 10^{-10}$, hard to guarantee *0*



MVRP: Multiple VLAN Registration Protocol
MSRP: Multiple Stream Reservation Protocol

MSRP Attributes

- Uses MRP Attribute Declaration
- Internal and external MSRPDUs.
- Talker advertise & Talker failed
- Listener Ready, Ready failed & Asking Failed
- Establish SRP domain boundary
- Connect a stream_id to a source and destination[3]

[3] dest can be *any* L2 address (unicast, multicast, broadcast)

TSN SR Classes

class	ID	Default pri	observation interval	Max Transit Time
A	6	3	125 μs	2 ms
B	5	2	250 μs	50 ms
“C”[4]			1333 μs	15 ms
“D”[4]			1451 μs	15 ms

Observation interval gives a minimum time between frames, but can be higher! (i.e. Class A *can* send frame every 250μs, but B *cannot* send every 125μs).

[4] Part of Automotive AVB profile

IEEE 802.1AS (gPTP) vs. 1588v2

A gPTP domain:

- MAC PDUs (L2) only.
- Only time-aware systems allowed.
- All talkers must be GM capable.
- No overlapping timing domains.
- Media independent sublayer.
- gPTP has Bridge (P2P transparent clock) and End Station. (ordinary clock)
- Clock accurate within 1μs over 7 hops
- Tracks time with ns granularity
- Simplifications to BMCA (→*faster* clock convergence).

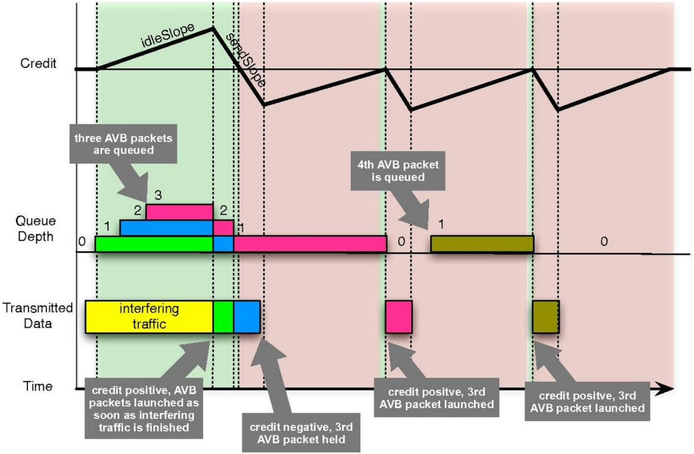
HW requirement: Credit Based Shaper

Required for Talkers
In *theory* only a single configure value for a NIC - *idleSlope*

- *idleSlope*
- *sendSlope*.
- maxFrameSize (For DMA engine)
- hiCredit (how much can you store)
- loCredit
- interference (MTU + bw for higher class)

The credit based shaper

- why software won't do



three AVB packets are queued

4th AVB packet is queued

credit positive, AVB packets launched as soon as interfering traffic is finished

credit negative, 3rd AVB packet held

credit positve, 3rd AVB packet launched

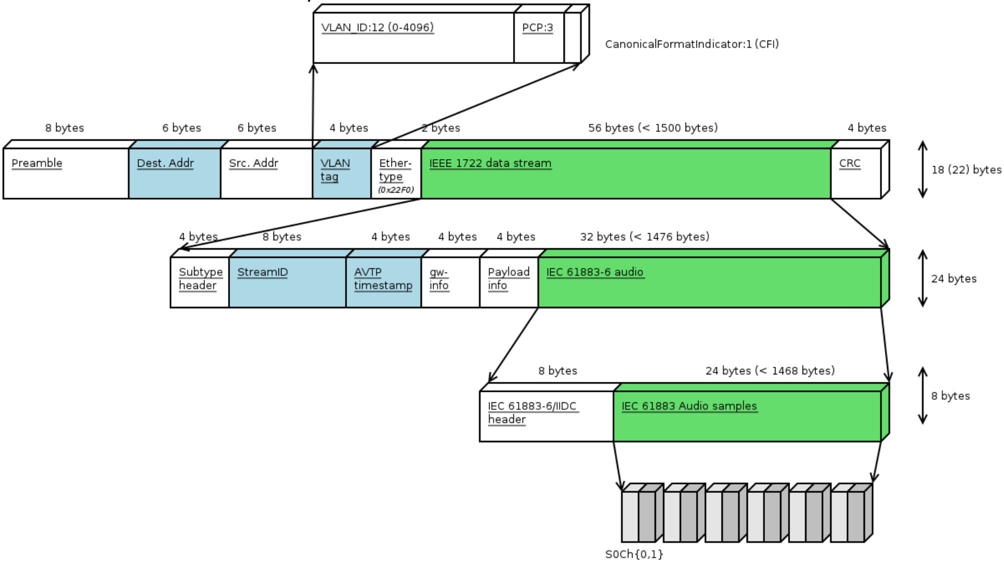
credit positve, 3rd AVB packet launched

https://en.wikipedia.org/wiki/File:Traffic-shaping.pdf

HW “almost-requirement”: PTP support

- Needed by both Listener and Talker
- Timestamp in PHY on ingress and egress
- Can do this in software, but gives large uncertainty
- Tight connection between NIC PTP circuit and audio-samplerate is nice (to avoid resampling)

AVTPDU Frame, IEC-61883-6 audio



How much could you add to it?

channels	streams	Mbps/stream	total	
1	118	6.34	118 ch	748 Mbps
2	**95**	**7.87**	**190**	**748**
4	68	10.94	272	744
8	**43**	**17.09**	**344**	**735**
16	25	29.38	400	734
24	18	41.66	432	749.95
32	**13**	**53.96**	**416**	**701.37**
40	11	66.24	440	728.64
48	9	78.53	432	706.75
56	8	90.82	448	726.53
61[5]	7	98.50	427	689.47

Class A, AM824, 48kHz, S16LE, 1Gbps link, 75% utilization

[5] 61 channels ought to be enough for anyone...

Picking at the TSN driver

Source: https://goo.gl/images/heF88x

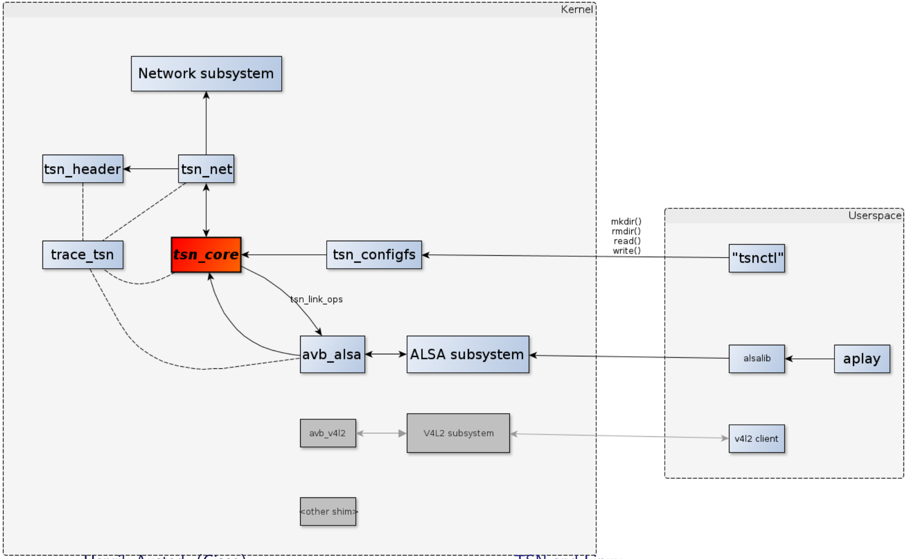
TSN Driver

- In the works since 2014[6]
- Renamed to TSN and sent for a wider review in June [7]
- Current status, rebased onto v4.8, reworked usage of i210 registers
- Very much in beta - but can be used to do fun things
- Introduces CONFIG_TSN, CONFIG_MEDIA_AVB_ALSA and CONFIG_IGB_TSN

[6] http://mailman.alsa-project.org/pipermail/alsa-devel/2014-May/077087.html

[7] https://lwn.net/Articles/690998/

Driver architecture



Network hooks

```
#if IS_ENABLED(CONFIG_TSN)
int (*ndo_tsn_capable)(struct net_device *dev);
int (*ndo_tsn_link_configure)(struct net_device *dev,
                              enum sr_class class,
                              u16 framesize,
                              u16 vid, u8 add_link);
#endif  /* CONFIG_TSN */
```

Currently added for Intel's igb-driver (I210 NIC)

About the “shims”

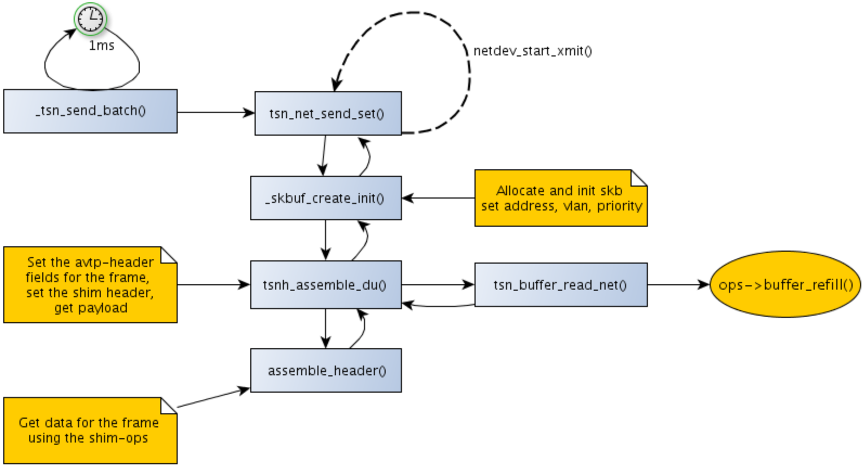
Basically a thin wrapper between systems

Defines a set of operations tsn_core will call into.

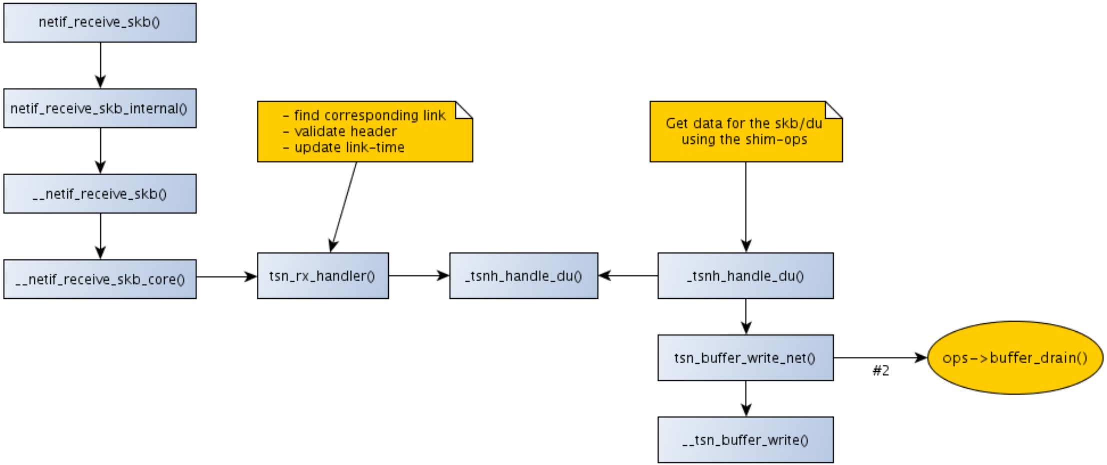
```
struct tsn_shim_ops {
  int (*probe)(struct tsn_link *link);
  [...]
  size_t (*buffer_refill)(struct tsn_link *link);
  size_t (*buffer_drain)(struct tsn_link *link);

  int (*media_close)(struct tsn_link *link);
  [...]
  void (*assemble_header)(struct tsn_link *link,
                          struct avtpdu_header *header,
                          size_t bytes);
}
```

Pushing frames



Receiving frames



Instantiating from userspace - ConfigFS

```
root@cerberus:~# modprobe tsn in_debug=1
root@cerberus:~# modprobe avb_alsa
root@cerberus:~# mkdir /config/tsn/eth1/link
root@cerberus:~# cd /config/tsn/eth1/link
```

```
root@cerberus:/config/tsn/eth1/link# for i in $(ls); do printf "%18s : %s\n" $i $(cat $i); done
       buffer_size : 16536
             class : B
           enabled : off
       end_station : Talker
         local_mac : 90:e2:ba:30:86:d3
  max_payload_size : 48
             pcp_a : 0x3
             pcp_b : 0x2
        remote_mac : 00:00:00:00:00:00
              shim : None
  shim_header_size : 8
         stream_id : 4074395330
           vlan_id : 2
```

Configuring a link

```
root@cerberus:/config/tsn/eth1/link# echo 65535 > buffer_size
root@cerberus:/config/tsn/eth1/link# echo 14:da:e9:2b:0a:c1 > remote_mac
root@cerberus:/config/tsn/eth1/link# echo 1337 > stream_id
root@cerberus:/config/tsn/eth1/link# echo 1 > vlan_id
root@cerberus:/config/tsn/eth1/link# echo alsa > shim
root@cerberus:/config/tsn/eth1/link# echo on > enabled
```

```
root@cerberus:/config/tsn/eth1/link# for i in $(ls); do printf "%18s : %s\n" $i $(cat $i); done
       buffer_size : 65535
             class : B
           enabled : off
       end_station : Talker
         local_mac : 90:e2:ba:30:86:d3
  max_payload_size : 48
             pcp_a : 0x3
             pcp_b : 0x2
        remote_mac : 14:da:e9:2b:0a:c1
              shim : none
  shim_header_size : 8
         stream_id : 1337
           vlan_id : 1
```

Creating a new ALSA device

```
root@cerberus:/config/tsn/eth1/link# aplay -L
null
    Discard all samples (playback) or generate zero samples (capture)

root@cerberus:/config/tsn/eth1/link# echo alsa > shim
root@cerberus:/config/tsn/eth1/link# echo on > enabled

root@cerberus:/config/tsn/eth1/link# aplay -L
null
    Discard all samples (playback) or generate zero samples (capture)
hw:CARD=avb,DEV=0
    Avb, AVB PCM
    Direct hardware device without any conversions

root@cerberus:/config/tsn/eth1/link# aplay -Dhw:CARD=avb /root/la_grange.wav
Playing WAVE '/root/la_grange.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Stereo
root@cerberus:/config/tsn/eth1/link#
```

Driver status

- Rebased onto 4.8
- net_device_ops hooks for configure and capable-test
- Normal testing is done on x86/amd64 and i210 NIC
- has an 'in_debug' mode
- avb_alsa shim for testing
- ndo-hooks in place
- register-config of i210 (idleSlope, Qav-mode)

New revision destined for LKML soon

Driver status

- Rebased onto 4.8
- net_device_ops hooks for configure and capable-test
- Normal testing is done on x86/amd64 and i210 NIC
- has an 'in_debug' mode
- avb_alsa shim for testing
- ndo-hooks in place
- register-config of i210 (idleSlope, Qav-mode)

Backlog

- v4l2-shim & raw_socket shim
- userspace client, "tsnctl"
- buffer management
- Proper integration with timing subsystem
- Syncing multiple streams
- Network-interface needs rework

New revision destined for LKML soon

References

- Very unofficial tarball dumping-ground: `https://lethe.austad.us/tsn/`
- TSN `https://en.wikipedia.org/wiki/Time-Sensitive_Networking`
- AVB `https://en.wikipedia.org/wiki/Audio_Video_Bridging`
- `https://en.wikipedia.org/wiki/IEEE_802.1Q`
- TSN Task Group: `http://www.ieee802.org/1/pages/tsn.html`
- AVnu Alliance: `http://avnu.org/`
- AVB bandwith calculator `https://abc.statusbar.com/`