TSN IN LINUX

Levi Pearson

10 Nov 2017

OUTLINE

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. Future features
6. Resources
7. Time Sync and Media Clocks
8. Standards Overview

PRESENTER

Contact info

Levi Pearson
Principal Engineer, Harman International
`levi.pearson@harman.com`
`levipearson@gmail.com`

Goals

- Give `tc` developers some context about TSN
- Get feedback on existing TSN-related `tc` features
- Get suggestions on best approaches for further features

Disclaimer

This represents my personal understanding, not an official interpretation of any of the relevant standards.

OUTLINE

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. Future features
6. Resources
7. Time Sync and Media Clocks
8. Standards Overview

WHAT IS TSN?

Time Sensitive Networking

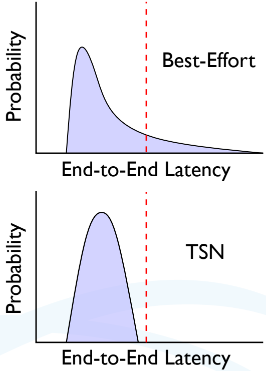
- An IEEE 802.1 (LAN/MAN standards) task group
- Set of standards/enhancements produced by the task group

Goals

- Provide bounded *worst-case* delays
- Precise synchronization, coordination, phase-alignment

Mechanism

- Global time synchronization
- Resource reservation and access control
- Traffic shaping and/or scheduling



THEORY OF OPERATION

HARMAN

Establish time-sync and stream reservation domains

- gPTP ensures link peers are gPTP-capable
- SRP uses **Domain** attribute to establish reservation capability

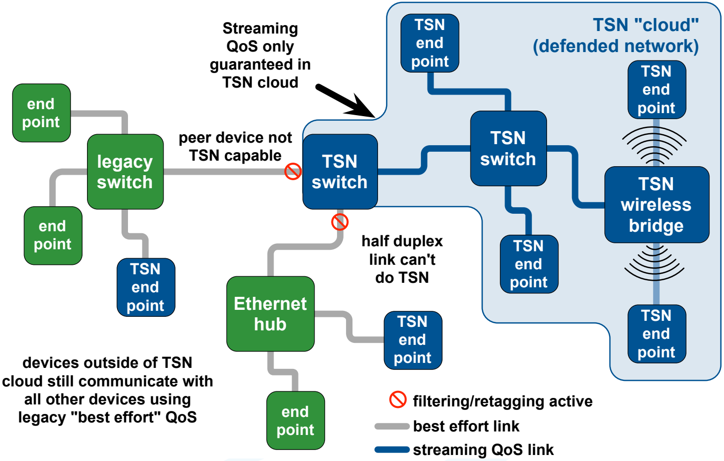
Reservations are established

- Talker uses SRP to advertise a stream (ID + priority/VLAN + bandwidth)
- Listener joins VLAN and uses SRP to join the stream
- Forwarding tables updated along talker/listener paths

Streaming traffic is forwarded

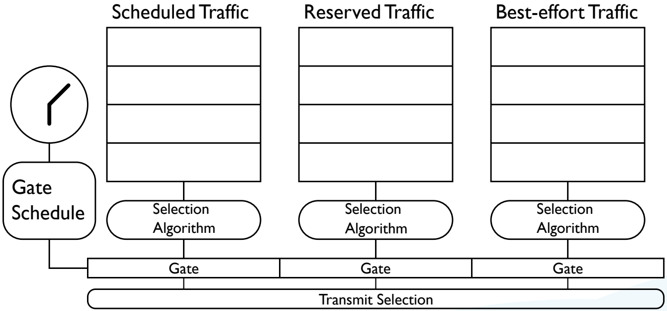
- Talker transmits at regular intervals w/timing information
- Bridges prioritize & shape stream traffic
- Listener receives traffic and recovers timing information

TSN DOMAIN


Streaming QoS only guaranteed in TSN cloud
TSN "cloud" (defended network)
TSN end point
TSN end point
TSN switch
TSN switch
TSN wireless bridge
TSN end point
TSN end point
end point
legacy switch
peer device not TSN capable
end point
TSN end point
half duplex link can't do TSN
Ethernet hub
TSN end point
end point
devices outside of TSN cloud still communicate with all other devices using legacy "best effort" QoS
🚫 filtering/retagging active
best effort link
streaming QoS link

TSN FORWARDING

Queues, shapers, and gates



OUTLINE

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. Future features
6. Resources
7. Time Sync and Media Clocks
8. Standards Overview

SENDING AND RECEIVING STREAMS

Packet sockets

- AVTP is a layer-2 protocol; data must be built and unpacked from raw frames
- Many small & frequent frames, `sendmsg`/`recvmsg` have significant overhead

Class measurement intervals

- "Class A" traffic = 125μs interval; "Class B" = 250μs
- Streams must not exceed frames/interval of their reservation

Class shaping

- Frames transmitted from a SR-class queue must obey CBS to reduce bursts
- Using HTB shaper w/DA classifier, a small number of streams can be shaped

VLANS, PRIORITY, MULTI-QUEUE

HARMAN

Streaming traffic classification

- SR-class traffic is always VLAN-tagged (default VLAN 2)
- Default priority code point assignments for SR-classes (2 and 3)
- Streaming traffic is forwarded based on a unique multicast DA

Special configuration

- Full-duplex only, no pause frames, no jumbo frames
- New default mapping of priority code point to traffic class/queue
  - Old map: 1 0 2 3 4 5 6 7
  - New map: 1 0 4 5 6 7 2 3
- When hardware shapers are available, map SR-classes to shaped queues (`mqprio`)

OUTLINE

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. Future features
6. Resources
7. Time Sync and Media Clocks
8. Standards Overview

CREDIT-BASED SHAPER

- A qdisc-based software shaper + hardware offload driver interface was just merged
- Goal: Provide maximum spacing between SR-class frame transmits
  - Prevents bunching of frames throughout the domain
  - Protects SR-class as well as lower priorities
- Only Intel i210 driver support; other NICs are capable of offload as well
- Doesn't yet provide stream-based shaping; must be handled by a single application
- How do we provide the necessary per-stream shaping?

TIMED LAUNCH

- RFC was recently proposed for a general TXTIME cmsg for scheduling packet launch
- Sets a launch timestamp field associated with the `skbuf`
- Intel i210, newer NXP SoCs, possibly others have hardware offload capability
- Could form the basis of time-aware `qdiscs` as well as userspace scheduling
- Some details remain to be hashed out
  - What timescale should the timestamp be associated with?
  - Where to store the timestamp? New skb field or overload Rx timestamp?
  - How to deal with out-of-order and already-passed timestamps?
  - How to integrate with TPACKET rings?

OUTLINE

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. **Future features**
6. Resources
7. Time Sync and Media Clocks
8. Standards Overview

STREAM IDENTIFICATION

- New TSN standards have a more flexible notion of stream identification
- They examine frames and assign a `stream_handle` to them
- Doesn't necessarily correspond to a single PDU field
- Active stream identification can overwrite PDU fields, add VLAN tag, etc.
    - Translate a UDP Tx stream to the correct VLAN + multicast DA for TSN forwarding
    - Reverse translation for Rx stream
- The `stream_handle` is used for redundancy, shaping, etc.
- Could this be implemented with a `tc` classifier + actions, e.g. flower?

PER-STREAM SHAPING

Hierarchical CBS

- Built with the structure of the HTB shaper, but following CBS rules
- Provide an interface for SRP or other management to add/remove streams
- Hardware offload could manage hardware CBS bandwidth
- Could also take advantage of timed-launch hardware

Per-stream filtering and policing

- Provides Filters, Gates, and Flow Meters
- Filters match on `stream_handle` and priority and have a Gate
- Each Filter has 0+ filter specs that can drop frames; frame size or Meter-based
- Gate has a programmable control list, can be time-scheduled
- Meters follow MEF 10.3 (same as existing 802.1Q flow metering)

SCHEDULED TRAFFIC

Per-queue timed gates

- Eliminates interfering trafic by opening/closing queue gates on a cyclic schedule
- Requires precise PTP clock sync access from the shaper
- Needs interface for setting gate open/close schedule
- Mechanism used for different shapers; offload available on some hardware

Cyclic queuing and forwarding shaper

- Replaces CBS; provides more determinism
- Each SR-class gets two internal priority codes to be used in alternation
- Per-stream Filter Gate rules provide internal priority alternation
- Transmits proceed from alternating queues on each cycle via time-gated queues
- Frames are resident for one cycle time at each bridge hop

REPLICATION AND ELIMINATION

Concept

Duplicate a stream at a configured point in the network onto multiple paths, then recombine at another configured point, discarding duplicate frames. Can eliminate loss due to link failure.

Method

- Recognizes stream frames by `stream_handle` at split point
- Copies to configured egress ports, assigns new `stream_handle` for duplicates
- Edits frames to contain a sequence number
- Recognizes frames from the same stream and drops repeats at join point
- Edits frames to remove sequence number
- Retain original `stream_handle`

OUTLINE

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. Future features
6. Resources
7. Time Sync and Media Clocks
8. Standards Overview

TSN-RELATED GROUPS

- TSN Task Group (official): http://www.ieee802.org/1/pages/tsn.html
- TSN Task Group (unofficial): http://www.802tsn.org/
- IETF DetNet Working Group: https://datatracker.ietf.org/wg/detnet/about/
- Avnu Alliance: http://avnu.org/

DOCUMENTS

- Free access to 802.1 standards: http://ieeexplore.ieee.org/browse/standards/get-program/page/
- IEEE TSN intro: http://www.ieee802.org/1/files/public/docs2017/tsn-farkas-intro-0517-v01.pdf
- Avnu Whitepapers: http://avnu.org/whitepapers/

CODE AND DATA MODELS

- OpenAvnu protocol implementations: https://github.com/AVnu/OpenAvnu/
- Yang Models for 802.1: https://github.com/YangModels/yang/

OUTLINE

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. Future features
6. Resources
7. **Time Sync and Media Clocks**
8. Standards Overview

PTP

Kernel services

- Hardware timestamping via MAC or PHY
  - Required for gPTP
- PTP Hardware Clocks
  - Adjustable from userspace via `adjtimex()`
  - Can get synchronized time via `clock_gettime()`
  - Event timestamp capture/compare interface
  - System/PHC cross-timestamp capability

Userspace services

- Multiple endpoint time-sync daemon implementations

MEDIA CLOCK MANAGEMENT

ALSA timestamp interface

- Need time, referenced to gPTP, when first sample was clocked
- ALSA has API for media-to-system timestamps; PTP provides system-to-PHC
- Not currently implemented in a precise way for many media devices

Measuring via PHC event capture

- 44.1KHz / 48KHz is too fast to service; need a hardware clock divider
- With an external divider, PHC event interface works for userspace control

Clock control

- External VCXO can be adjusted via kernel-exposed PWM interface
- Built-in SoC PLLs may be on-the-fly adjustable via clock API

OUTLINE

HARMAN

1. Introduction
2. TSN overview
3. TSN in Linux today
4. In-progress features
5. Future features
6. Resources
7. Time Sync and Media Clocks
8. **Standards Overview**

ORIGINAL TSN/AVB PROTOCOLS

- Generalized Precision Time synchronization Protocol (802.1AS)
  - gPTP, a profile of IEEE 1588, provides $<1\mu$s time sync
- Stream Reservation Protocol (802.1Qat)
  - SRP provides decentralized reservation/provisioning of resources
- Forwarding & Queueing for Time-Sensitive Streams (802.1Qav)
  - FQTSS defines a Credit-Based Shaper (CBS)
  - Has useful mathematical properties for calculating max burst
  - Under 250μs delay per bridge
- Audio/Video Transport Protocol (IEEE 1722)
  - AVTP is a Layer-2 transport based on IEC-61883 (Firewire A/V)
- A/V Device Enumeration, Connection management & Control (IEEE 1722.1)
  - AVDECC is a Layer-2 management protocol for AVB

PTP ENHANCEMENTS

802.1AS-REV - Timing and Synchronization for Time Sensitive Applications

- Both IEEE 1588 and 802.1AS are getting revisions and will be harmonized further
- Support for 802.11 Fine Timing Measurement
- Support for Link Aggregation
- Configurable for redundant paths and redundant GMs

SRP ENHANCEMENTS

802.1Qcc - SRP and Time-sensitive Networking Configuration

- Backward-compatble enhancements to MSRP, including TLV-style Attributes
- Scalability improvements to underlying protocol
- Includes configuration of new TSN features
- New configuration models: Fully distributed, fully centralized, centralized network/distributed user

802.1Qcp - YANG Data Model

- For use with NETCONF/RESTCONF configuration
- Alternate to both SNMP and SRP TLVs for Centralized models

802.1CS - Link-local Registration Protocol

- Replacement for MRP to implement SRP registration database

REDUNDANCY & PATH CONTROL

802.1CB - Frame Replication & Elimination

- Fault tolerance through redundancy
- Split and re-join streams at redundant path junctions

802.1Qca - IS-IS Path Control & Reservation

- Alternate to spanning trees for defining network topology
- If MSRP is not used, it can also specify bandwidth per traffic class

SCHEDULED TRAFFIC

802.1Qbv - Enhancements for Scheduled Traffic

- Synchronized, time-gated queues to precisely schedule traffic egress from queues
- Uses time slices and guard bands to do TDM over Ethernet
- Can eliminate interfering traffic, producing lowest-possible timing jitter

FRAME PREEMPTION

Preemption summary

Scheduled traffic can cause severe impact on lower-priority queues; preemption allows best-effort traffic to “fill in gaps” between precisely scheduled traffic even when it is larger than gaps.

802.1Qbu - Frame Preemption

Assigns traffic class queues as “express” or “preemptable”

802.3br - Interspersing Express Traffic

Performs MAC-level frame splitting and merging of preemptable frames

PER-STREAM FILTERING

802.1Qci - Per-Stream Filtering and Policing

- Identification of individual streams and assignment of unique ids
- Filtering based on unique id and priority, with wildcards for either
- Gates with an inner priority to change traffic class assignment
- Flow meters for per-stream flow-based filtering, MEF 10.3 algorithms

NEW SHAPERS

HARMAN

802.1Qch - Cyclic Queueing and Forwarding

- Uses per-stream filtering to send SR-class traffic to alternating queues
- Uses enhancements for scheduled traffic to alternatively open queue gates
- Achieves zero congestion loss and deterministic latency

802.1Qcr - Asynchronous Traffic Shaping

- Doesn't require time synchronization as cyclic queueing and forwarding does
- Prioritize urgent over relaxed traffic and re-shape per hop

MOBILE FRONTHAUL PROFILE

HARMAN

802.1QCM - Time-Sensitive Networking for Fronthaul

- A profile, like 802.1BA, describing how to use TSN protocols for fronthaul networks
- Developed jointly with CPRI Cooperation