IEEE 1588 – aka PTP

Intro

This appears to be a complete tangent. However, stick with me. When I was an undergraduate, I was a bar steward at a private club in London. Amongst the clubs many treasures is a grandfather clock. The plaque on this read that Sir Walter Scott used that clock to set his watch by when in Edinburgh.

This may appear to be a bit "random!". Well, many years later I was writing up PTP for a training course, and started looking into "standard time". Until the 19th century, there was no need for time coordination. The sun came up, the sun went down. The days got longer, the days got shorter throughout the year. The "Town clock" was a thing, and in cities, this was often a group of clocks. In the Naval tradition, the ball dropping was a nautical phenomenon. For accurate navigation, ships needed to set their clocks to a known time, to track the progression of the sun through the sky, or stars crossing the night sky. The use of a sextant and accurate clock gave a good idea of one's location. We couldn't rely on sound (a gun firing) because even in the 17th century, the speed of sound was known to be relatively slow. In a large port like London, the delay from the observatory at the speed of sound would be problematic. So the visual indicator of a ball dropping – light obviously being significantly quicker than sound! So, the ball dropping in times square on New Years Eve owes its origin to this.

The "need" for standard time first came into play with the advent of the railway. Railways need timetables, without all the clocks telling the same time, chaos, and argument. Just a note here – A few months ago I went to get a train home. The train appeared to leave 20 seconds EARLY! Now, this sounds like a tall tale, but it did. There is, however, a potential excuse! We currently have two "standard times". Huh? Yes – there is UTC and TAI. UTC is often used – this is the reference that GPS uses. This incorporates "leap Seconds" at the time of writing, I think we are about 37 leap seconds "ahead". "ahead of what?" TAI is the International Atomic time. This is a network of labs with atomic clocks. These atomic clocks do not include leap seconds. All the time we have positive leap seconds, no problem. Unfortunately, there are actually negative leap seconds too.

Positive/negative leap seconds, so what? In our human linear modelled fallacy, we don't like having something placed at the same time. If we move time forward "a leap second" a whole number of timestamps never (by convention) existed, so no data needs to be worried about! If we now incorporate negative leap seconds, we "duplicate" the same timestamps, but these actually represent distinctly different "times!". There is obviously a lot of debate about this. We "could" have a flag that asserts "leap second correction time". This is extremely messy. We "could" ignore it, and accept that hopefully being broadly a wave function, we will likely get a fulcrum. The latter seems to be pragmatic for humanity's sanity! High speed trading would go horribly wrong potentially! Now THAT was a (fascinating) deviation!

Sir Walter Scott didn't need to worry about gravitational waves and LIGO (Laser Interferometry Gravitational-Wave Observatory). The need (as with the audio and video application of PTP) was to make sure the clock in Edinburgh was telling the same time as the one in London! I have the peculiar fortune to live in Brighton, Sussex. Brighton has some very odd relations with Time! This is strange coincidence, I'm sure. 1. The author Lewis Carrol lived in Lewes Crescent in East Brighton. His story Alice in wonderland included the "white rabbit" who "mustn't be late", if you are familiar with the story, the rabbit disappears down a hole near where Alice was lying on the grass, and Alice follows

him down the hole! They emerge on a lawn near the sea. This isn't a vivid imagination, this is a description of the private gardens in Lewes Crescent, and the private tunnel that emerges by the sea (at the bottom of low cliffs! 19th century children's fiction, are you feeling ok? Actually, yes – CERN – the huge physics experiment that crosses between France and Switzerland, and where Sir Tim Berners-Lee did his work on Hypertext Transfer Protocol, has a need for very precise timing. CERN have "white rabbit" PTP – a super high-resolution version of IEEE1588! The other Brighton"time" fame was the BT "speaking clock" this was located close to the end of my road at the end of Preston Park, next to the London to Brighton Railway line!

Swiss/French border Rabbit Hole!

So literature, the universal, the sub atomic, phones and trains, A nod to the Worldwide Web… anything else… oh ok. On the subject of CERN the closest I ever got was Montreux. This was where the Jazz festival is held every summer, and we were invited to see a product launch. This was where Shure introduced Axient (the analogue version). Axient was the reason for including the UK National Anthem performance by Jeff Beck in the video archive. This was my first live "tryout" of Axient, while we are at it then… At Montreux I had a lovely time watching some shows (by a few of our then clients) and saw Quincy Jones in one of the hospitality areas- a musician of the finest pedigree. Now THAT is a tangent!

A few years later we got invited to CERN, and sadly, never made it there. Something I regret!

PTP Mechanism

Like the train, we need to have a single reference for our time. We need at this stage to distinguish time, from "clock". In general conversation, clocks tell the time don't they? Not directly, with digital signals. A digital "clock" is concerned with the frequency. Very much the "ticking" of the clock, not the hands of the clock!

PTP by contrast is "precision TIME Protocol" it is concerned with the "hands of the clock"

This is where timing servos matter! PTP and other high precision timing references operate at a very low frequency. GPS – which is a phenomenal reference source signals civilian NMEA messages once per second. It is the exact spacing, driven by an atomic clock, that when decoded from multiple sources allows time recovery (in the case of my workplace) to be in the tens of nanoseconds range. PTP is typically for AES67, and ST2110 log -3 or 2^-3 0,125 seconds or 8 messages per second. Dante uses log -2 or 0.25 seconds – 4 messages per second.

So AES67 is doubly as good? It is not as simple as that I'm afraid. We want to generate a "clock" from timestamps. This is where we use servos.

The easiest way to understand servos, and how they relate to time, is via the railway clock. I get on a train with my best pocket watch, that I have meticulously set to Big Ben (or some other grand reference!) I need to ensure that I'm not taking a "distorted" snapshot. So I'll set my pocket watch, and in the traditional manner, leave it running next to my reference. I'll let both run for a while (I have visions of a suitable pause for some libation). Then I compare the two time pieces, and adjust my pocket watch to match the rate of the reference. I still remember analogue clocks, which had the little +/- adjustment in the middle of the dial. On the grandfather clock, this was achieved by moving the weight on the pendulum up and down. After another pause, check again. And another libation.

When satisfied that the two time pieces are running at the same rate – this is called syntonous, we begin to set the hands or "synchronise". Being assured we have a pocket watch that is both synchronous and syntonous with the reference we head to the station. At the station, we go to the station clock and set its syntonisation in a similar manner. Then, the synchronisation. We next get on the train, get off at the first station, and set up the station clock in the same manner. By the time we reach Edinburgh, we hope that all the station clocks we have set are keeping time.

This is the first part of the ptp process. We gently alluded to some key components. I am acting as a "local master". This is because I have set my pocket watch very carefully from the "grand master". PTP has a "master election process" or the BMCA (Best Master Clock Algorithm). BMCA has a priority criteria, in this example, I arrive at the station, and tell the station master "I'm the reference carrier from the grand master – this is the time at the grand master. I am effectively a "boundary clock" in PTP parlance.

PTP terms alluded to so far

Grand Master – this is at the Root of the PTP tree – the Reference that the "watch" is set to

Master – this is (in the case of the example) the synchronised pocket watch (initially)

Boundary clock – this is the clock at the individual station. A boundary clock can become a local master for downstream ordinary clocks (or even, but not perfect practice, downstream boundary clocks. The Boundary clock master declares its Grand Master's identity.

PTP Announce – this is when the grand master signals to its downstream followers (previously slave) clocks who it is and how "good" its clock is – there is a Clock quality calculation in PTP

Where the analogy begins to break down

Whilst we have dealt with the idea of hysteresis in a servo (gently introduced by the breaks for libation as the pocket watch is allowed to "relax" or settle, and is then checked, and adjusted). We could suspend the analogy, to deal with the differing realities inherent when trains and signalling are compared! In the UK trains do NOT travel at the speed of light, they don't even travel (by train standards that fast!)

Apologies for getting into "protocol weeds", the time has come I'm afraid!

We have the idea of a Grand Master, Boundary, and Ordinary clock. There is another kind, which is the most misunderstood one. The "transparent" clock. I will describe it here for completeness. It is 100% necessary to time experiments like the Large Hadron Collider at Cern, potentially LIGO etc. HOWEVER, given my existence in the real world of "Roadieing" as well as maybe having been around some big science, I'm still to be convinced that any normal practitioner measures time in a live sound system to the extent that a "Transparent" clock is significantly better than an ordinary clock, driven by a local boundary clock. A "transparent" clock is "transparent to time, it is definitely not "transparent to data and switching! The specific "benefit" is that a transparent clock has a "correction" field, which allows the switch that the ptp packet passes through to compare the ingress timestamp (in nanoseconds) with its egress timestamp, and then to "correct" the time by subtracting the "residence time" (the time it was in the switch) from its egress timestamp. Now, in an age of Broadcom Tomahawks, and Marvell Teralynx, we are in "cut through" switching mode – this means

that my 100 or even 800 Gbps interface upon receipt of the 48bit destination MAC address, immediately begins forwarding the packet, and it is already on its way out, before it has completely arrived! This absolutely minimises the delay caused by the switch, and therefore represents a tiny correction. In fact so insignificant, that the length of copper wire is of much greater significance!

It was important to deal with servo hysteresis in a human way. Set the clock to follow the reference, some "libation" confirm, adjust if necessary. This hysteresis is the magic bullet that deals with how we scale up the speed of our discipline 4 or 8 messages per second, to 48kHz or 10MHz. we aren't generating the clock. We are merely converting the observed passage of time, as represented by the nanosecond timestamps, and using these to align the rising edge of the internal oscillator generated clock. This has been done for decades. The "external word clock" or Black and burst in video is compared with the incoming discipline signal, and then, the internal rising edge is corrected by getting the internal oscillator to speed up and slow down until the clock edges are aligned – our 8hzsignal occurs 48000/8 or 6000 internal words per ptp sync packet. The servo works to ensure that these pulses are evenly spaced, before they are compared with the 48kHz word clock, which in turn generates the bit clock and mclock accordingly.

Clock election

This is normally the first thing to be talked about. This is because, in Audio Visual networks, we don't do quite the scale, so full understanding is seldom needed. Yes, setting the "hierarchy" is important. PTP v1 or IEEE1588:2002 was very simple. The device was either master, or not. If 2 devices said they were master, the "best" one won – this would be based on its stratum. Borrowing heavily from NTP the top tier is an atomic clock, and a GPS counts as an atomic clock! Then a "posh" oven-controlled crystal oscillator, then a voltage-controlled Crystal oscillator, then a normal Crystal oscillator, then software.. If the two contending devices were of equal quality, then the "winner" was the lowest value MAC Address! In IEEE1588:2010 or ptp v2. There is a Priority 1 and Priority 2 field. The lower priority 1 wins, if there are identical low priority 1 then priority 2 is checked the lower one wins. However, priority 2 isn't checked until the clock quality is compared. This is sensible, and prevents typos!

Deploying with datacentre switches

PTP is available on the latest generation of datacentre switches. Some implementations take their implementation of boundary clocks sensibly seriously. The Grand Master will transact with the boundary clocks in each switch. These boundary clocks are set to higher priority 1 than the master clock, so the master clock is the master clock, and the switch boundary clocks follow. One "trick" is that many enterprise switches can filter ptp messages from inter-switch links. This means that the downstream devices see the closest physical switch to them as their local master – they know who the Grand Master is (the authority for the local master). The local master can discipline the directly connected followers. The filter in the inter switch link means that another "local master" boundary clock, can be in the same VLAN, but its downstream ordinary clock devices are unable/unaware of the other switch master clock.

Why is this important?

PTP is bidirectional. It also assumes symmetric connections. This is where the upstream is the same data rate as the downstream. All following ordinary clocks have to communicate with their respective leader/master clock. This is to allow the delay over the wire to the master to be calculated. The ordinary clock device sends a message to the master labelled "delay request" it is

timestamped with the nanosecond time it left. It hits the master, who sends a delay request reply message with Received at, and replied to at time stamps. This allows the ordinary clock to calculate its delay in real time vs the upstream superior clock.

Important things that are often glossed over

1. The use of priority 2 in spine/leaf networks is important. It is often the case that we use MLAG (Multi-device Link Aggregation Groups) groups. This is where we have 2 or more switches arranged to form one logical entity. We often “dual home” switches or devices in this scenario. This protects against failure. It would be troublesome if all the members had identical priority 1 and priority 2 (we would have to pray that the MAC address tiebreaker worked predictably – chances are, it is annoyingly complex in mlag to guarantee this but hysteresis libation is preferable! This is why we use different priority 2 values for the members of an mlag! Downstream devices will see only 1 master clock, citing its authority from the Grand Master.
2. Boundary clocks can generate different sync intervals. If my GM is running 8 sync messages per second, and I need 4. It is entirely possible to have a “new” port on a boundary clock at the lower/higher or different sync interval. This caused much confusion early on “Dante uses ptp1 it wont work with ptp2 – yes and no. the original Dante ptp1 devices were boundary clocks. Primary and secondary represented two independent ptp “ports”. It is true that there wasn’t a codified boundary in ptpv1, however, this is the beauty of standards. The most long-lived standards are open to improvement. The need for multiple profiles of ptp in a network, and the tedium in documenting “non-standard” working implementations repeatedly is the sure indicator to the stakeholders, that its time for a revision. This is what happened to create VLANS in IEEE802.1q. there were myriad ways of achieving MAC ADDRESS Table segmentation, so the IEEE come along and codified it, problem solved interoperability achieved. One issue between ptpv1 and ptpv2 is that they are described as being “incompatible”. Strictly this is true. PTPv1 doesn’t have the required datatypes for v1 and V2 has more datatypes than v1 expects. The pragmatic solution is to place a v1 “port” on an interface, alongside a v2 “port” – these “ports” are in the application layer – so they aren’t on “wire” per se, they can talk v1 on the v1 port, the application can resolve the (identical) sync timestamps in fact IEEE1588:2010 actually contains translation tables precisely to achieve this!
3. There is no way to tell the difference between ptpv1 and ptpv2 below the application layer. The only difference when doing deep packet inspection is an indicator v2 on a v2 packet, and obviously v1 didn’t know it was v1, it just is…