Deterministic Ethernet and the Intrinsic Value of Synchronization

Norman Finn
Cisco Fellow

WSTS June 2014

Success

Rah! Rah!

- **IEEE 1588 technology could conquer the world.** (Of time.)
- Growing acceptance, both in use and in experimentation, for:
  - Service provider radio mobile base station
  - Real-time process and machine control
  - Interference reduction in unlicensed radio applications
  - Wireless location services
  - Audio / video production, editing, and presentation
  - Cryptographic security applications
  - Delivering zero-congestion-loss and fixed-latency packet networking
  - Packetizing and generalizing the dedicated digital signal

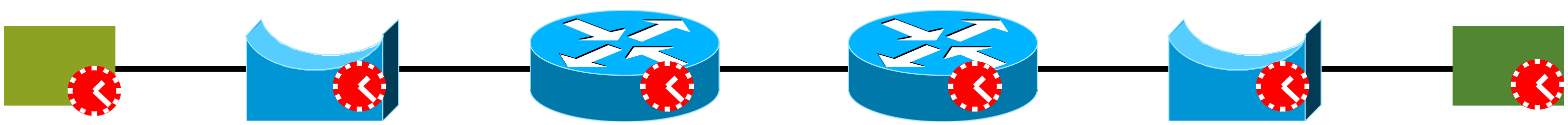
Brilliant!

- Let us take a moment to mention that the sync / followup technique of IEEE Std 1588 is **brilliant**.
- It gives the choice of two ways to **preserve proper layering** in the protocol stack:
  - Predict when the packet will emerge on the wire (insertion of transparent clock forwarding delay); or
  - Retrieve the exit time stamp and send a followup message.

Our thinking until recently has been …

- We have L2 network nodes (bridges = switches) and L3 network nodes (routers).
- We add **time-awareness** to these devices.
- Time-aware bridges and routers pass PTP PDUs through the network just like data, but with time information. (This is especially true in the various forms of the transparent clock.)
- Before:

- After:



This is one approach that can be taken

- The present approach does not violate the sacred principles of ISO Layering.
- But, neither does it take advantage of them.
- We are adding a feature to every bridge and router.
- But, when you look at some of the questions this raises, you realize that **there is another approach that will prove much less difficult to specify and to implement**.

That leads to this kind of diagram: (from 802.1AS-2011)

- A time-aware bridge from IEEE Std. 802.1AS-2005™.
- We have noticed, in P802.1ASbt, that the **MAC relay** in this picture **is not actually used** by 802.1AS.
- This makes one suspect that we using a suboptimal layering architecture.

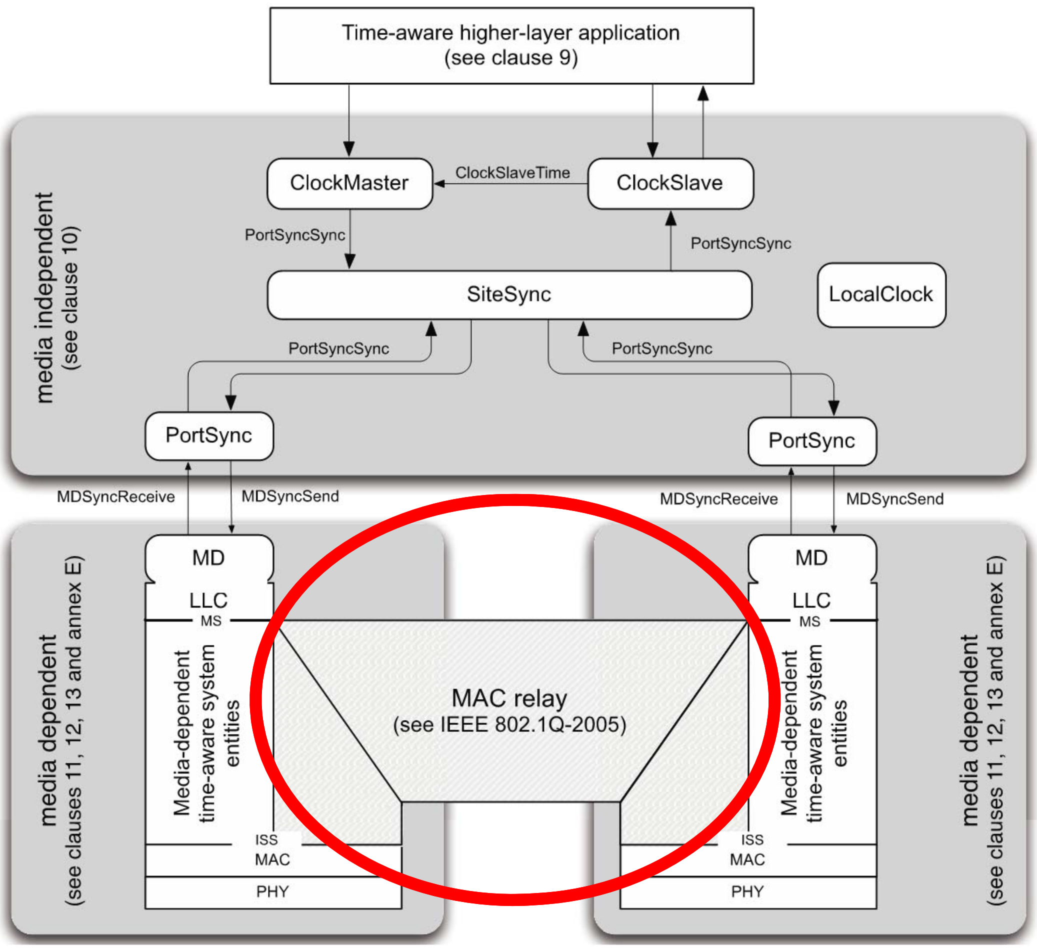


Figure 7-4—Time-aware system model

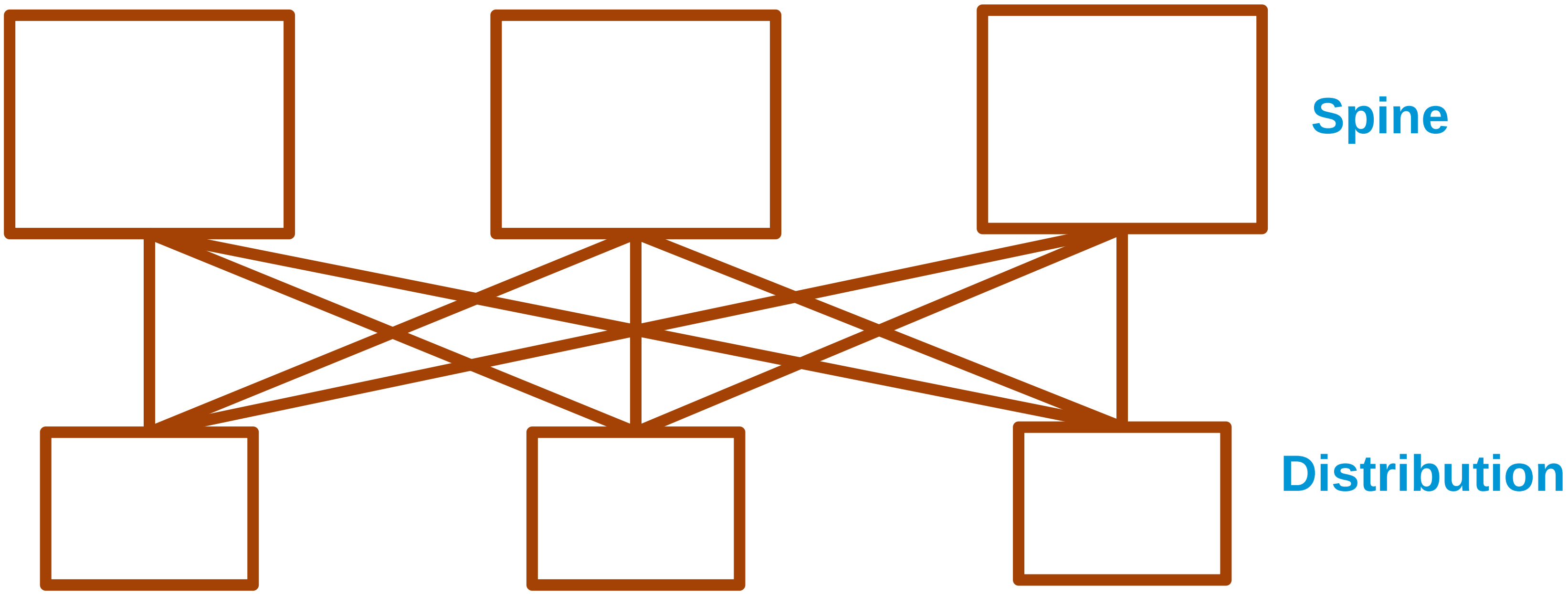
Some questions

Then and now

- In 1992, you could look at a box and say, “That box is a bridge,” or, “That box is a router.”
- In 1992 it made more sense to define a time-aware bridge and a time-aware router.
- But, my network has:
  - NAT engines, firewalls, load balancing appliances, multi-homed hosts, Link Aggregation, TRILL Rbridges, MAC-in-MAC bridges, ITU-T Ethernet protection switches, MPLS label switches, VPNs, bridge-route-bridge switches, etc.
  - VXLANs (virtual networks of virtual bridges and routers connecting virtual hosts)
  - Virtual switches (two boxes claiming to the world to be a single switch)
  - Software Defined Networking that (some use to) blur L2/L3 boundaries.
  - Equal-Cost Multi Pathing at both L2 and L3
  - Soft switches in server blades and/or virtual computers.
  - Proprietary extensions in Ethernet preambles and/or interframe gaps.

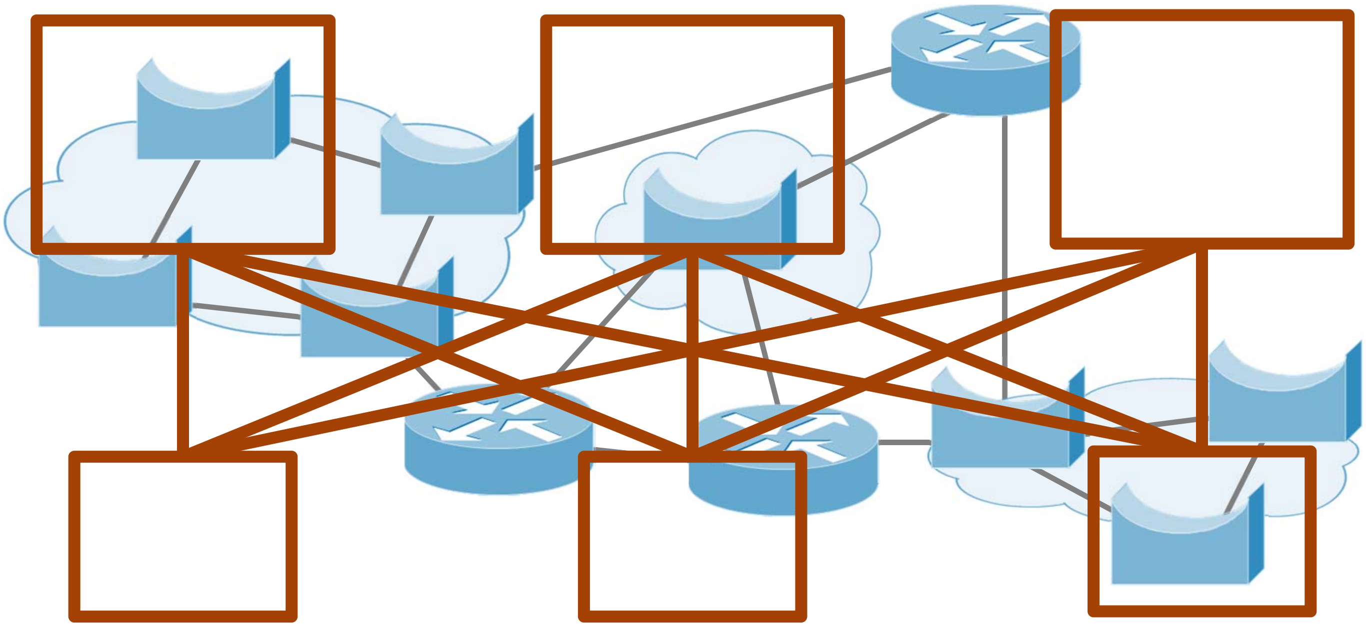
Why is this a problem? Example 1: Complex virtual data center networks

- **Physical network**



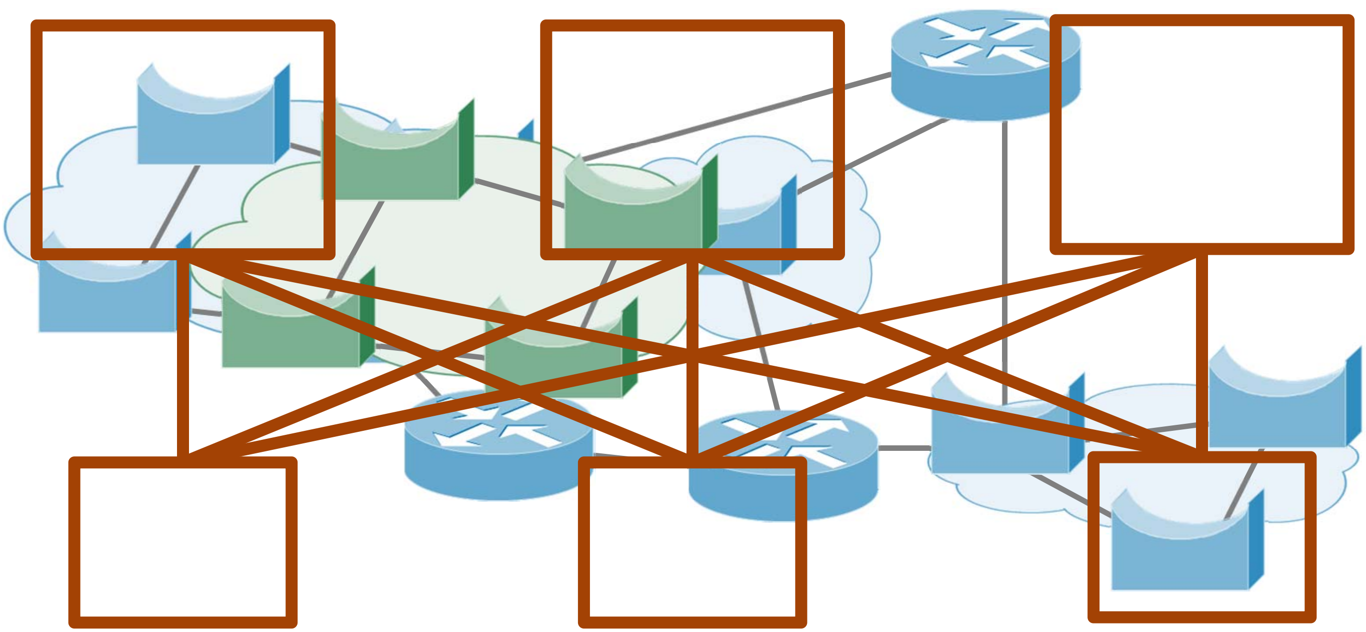
Why is this a problem? Example 1: Complex virtual data center networks

- **Physical network** creates multiple virtual network topologies.
- It simulates Virtual Network **A (blue)**



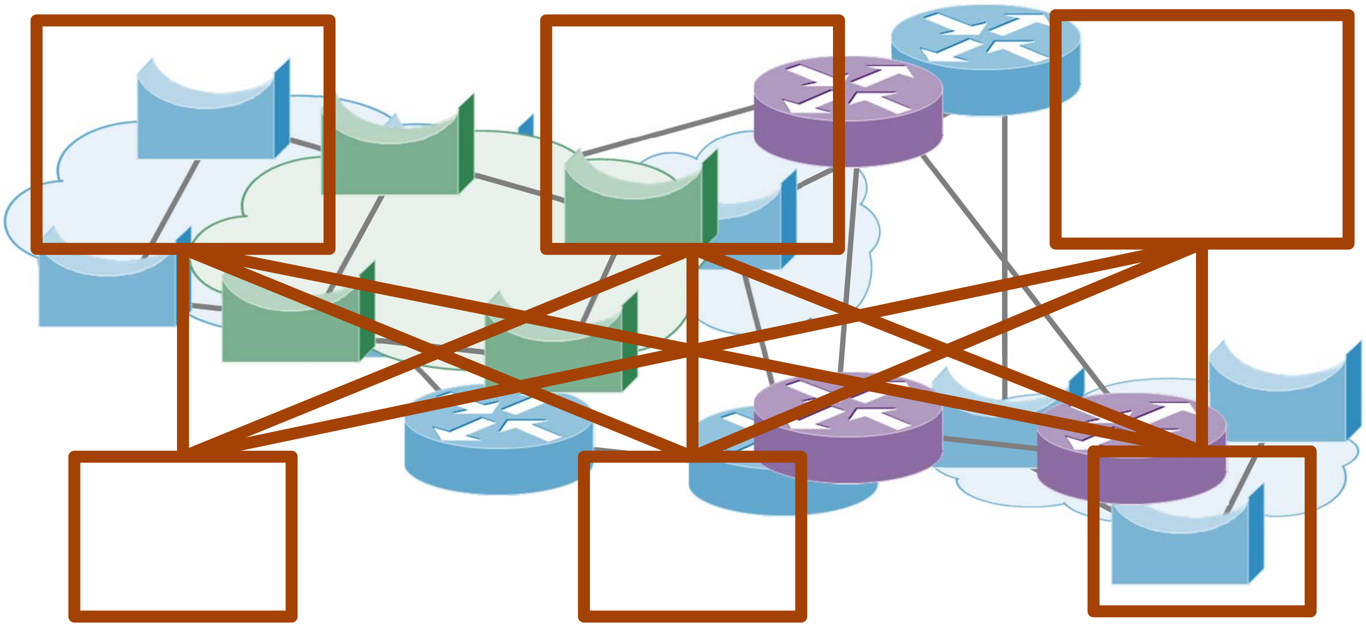
Why is this a problem? Example 1: Complex virtual data center networks

- **Physical network** creates multiple virtual network topologies.
- It simulates Virtual Network **A (blue)**, **B (green)**



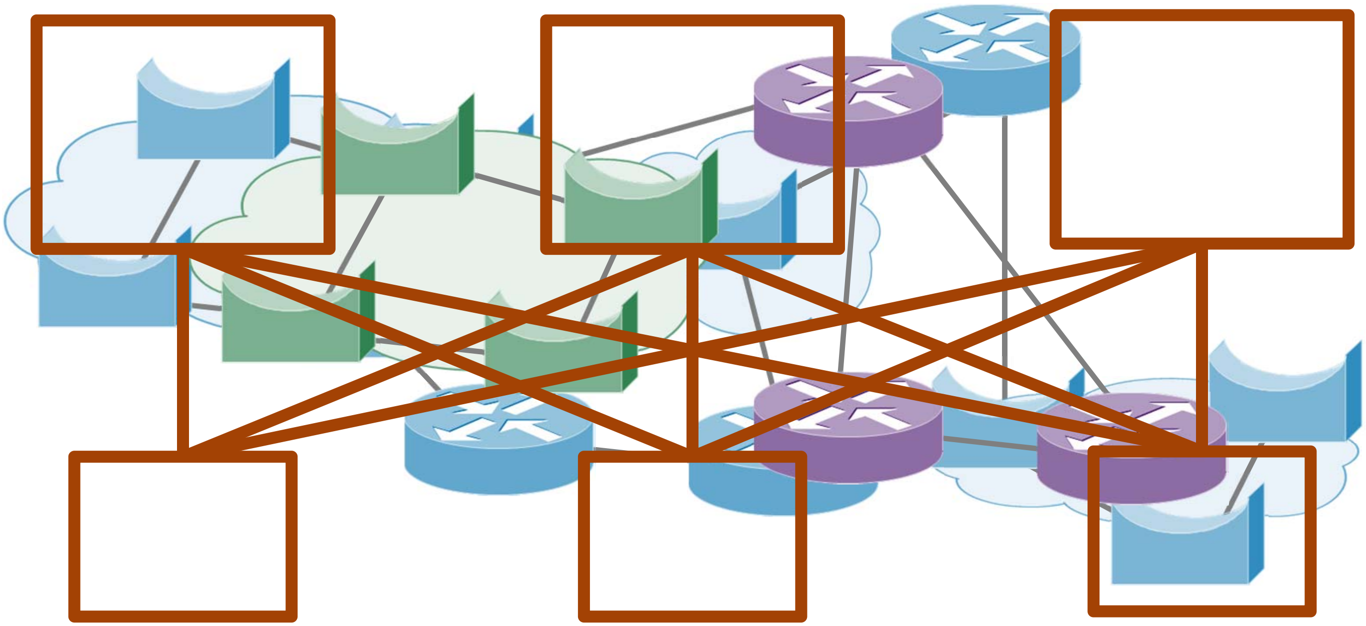
Why is this a problem? Example 1: Complex virtual data center networks

- **Physical network** creates multiple virtual network topologies.
- It simulates Virtual Network **A (blue)**, **B (green)**, and **C (purple)**.



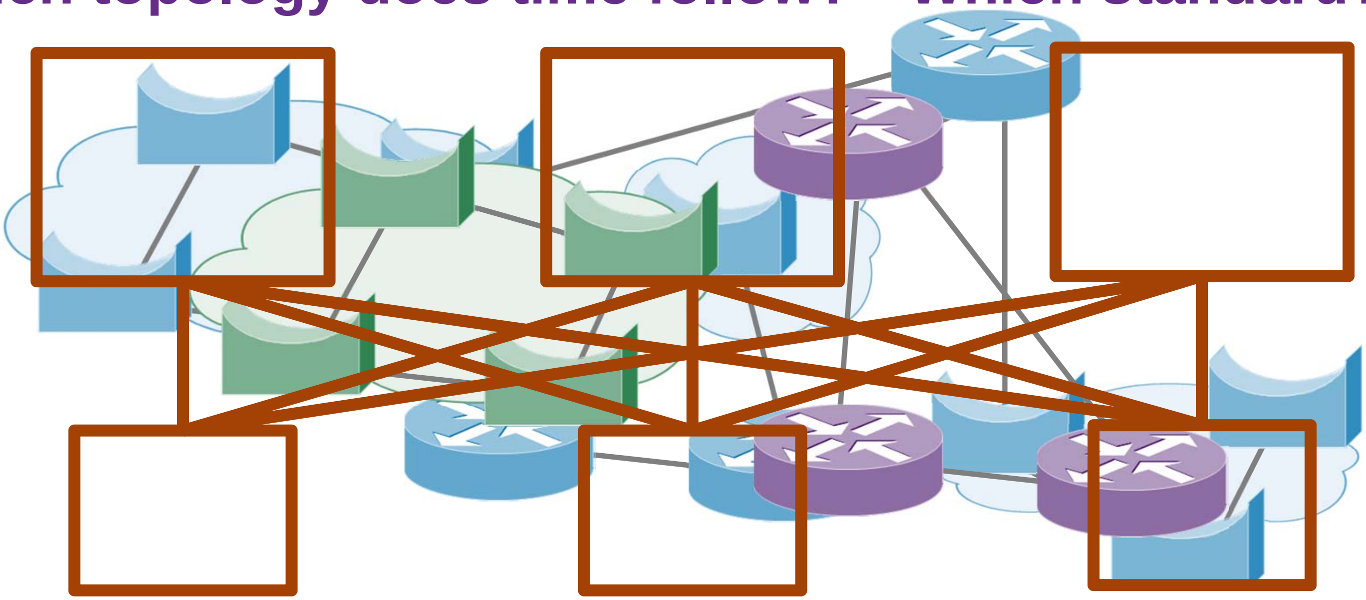
Why is this a problem? Example 1: Complex virtual data center networks

- **Physical network** creates multiple virtual network topologies.
- It simulates Virtual Network **A (blue)**, **B (green)**, and **C (purple)**.
- Complex encapsulations (e.g. VXLAN) divorce physical & logical.



Why is this a problem? Example 1: Complex virtual data center networks

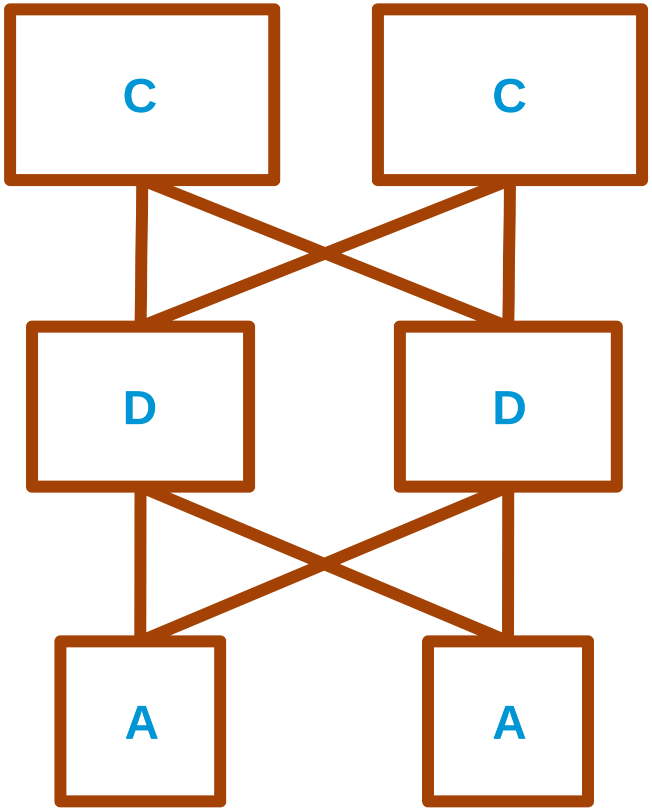
- **Physical network** creates multiple virtual network topologies.
- It simulates Virtual Network **A (blue)**, **B (green)**, and **C (purple)**.
- Complex encapsulations (e.g. VXLAN) divorce physical & logical.
- **Which topology does time follow?* Which standard?**



* "Add another topology" is not an answer – the hosts need time!

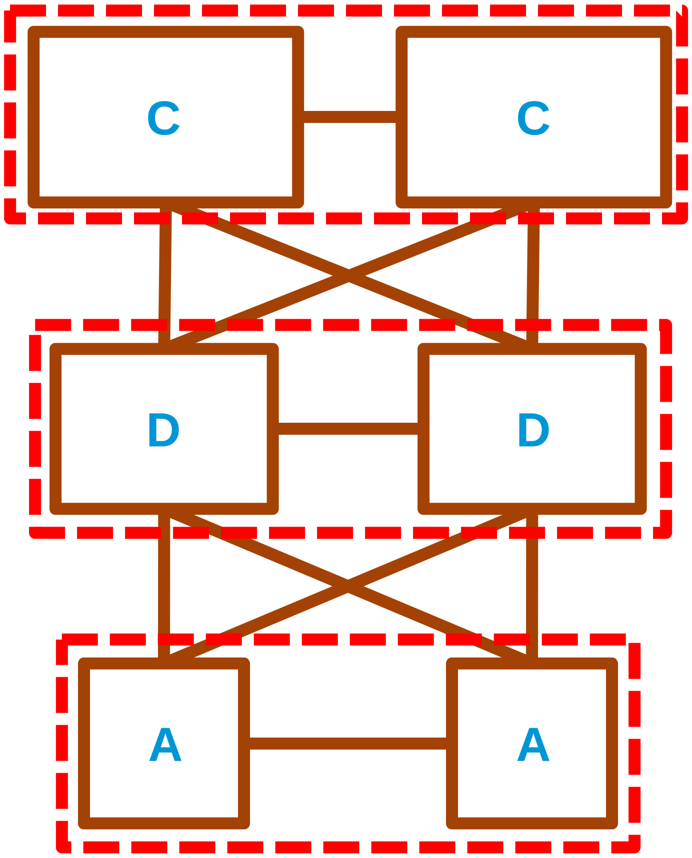
Why is this a problem? Example 2: Typical enterprise networks

- Three layers Core (routers), Distribution (routers), and Aggregation (switches = bridges or bridge + router).
- Thousands of Aggregation switches.
- Don’t imagine that there is only one layer of Aggregation switch / Wi-Fi Access Point.
- **Why would I want thousands of L2 1588 regions, one per Aggregation switch?**



Why is this a problem? Example 3: Virtual switches

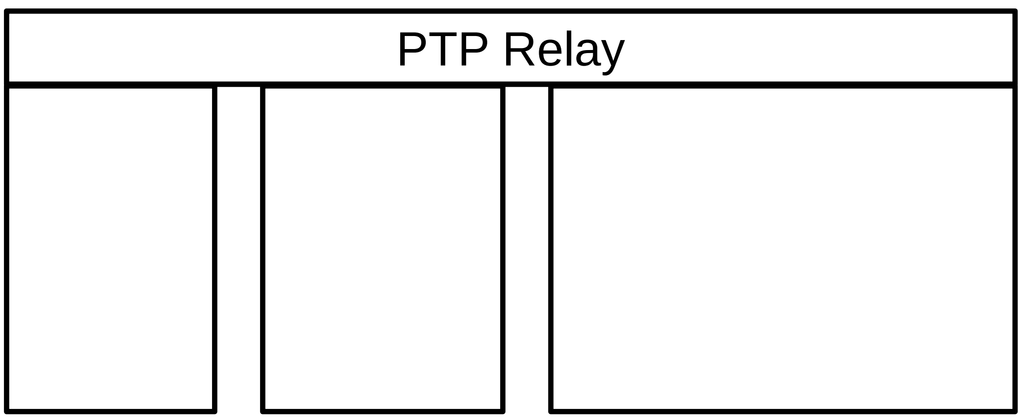
- Pairs of boxes each pretend (via proprietary protocols) to appear to the world to be a single box.
- Topology is then a tree – no bridging protocols needed!
- **But, how does this work for a "time-aware bridge?"** (Each box has to be visible as a separate entity for time purposes.)



Some answers

The Answer: **The PTP Relay (PTPR)**

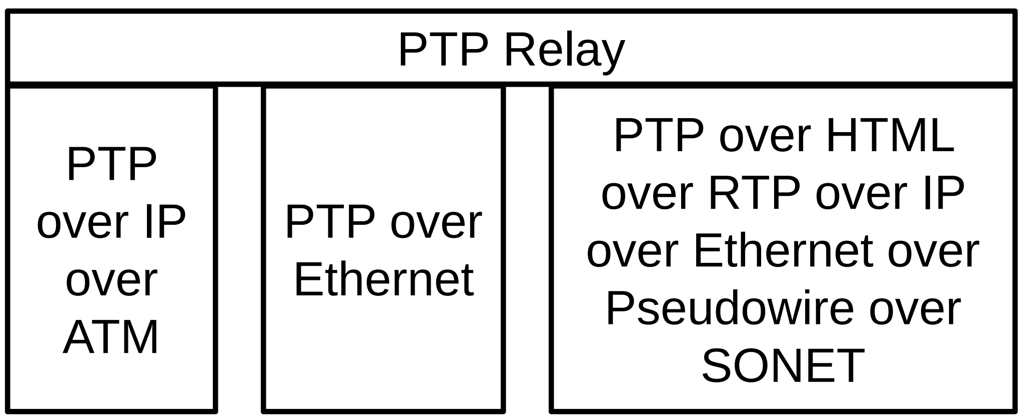
- **PTP is a user of the network(s), not integrated inside the networking functions.**
- Each PTPR has **multiple ports**.



The Answer: **The PTP Relay (PTPR)**

- **PTP is a user of the network(s), not integrated inside the networking functions.**

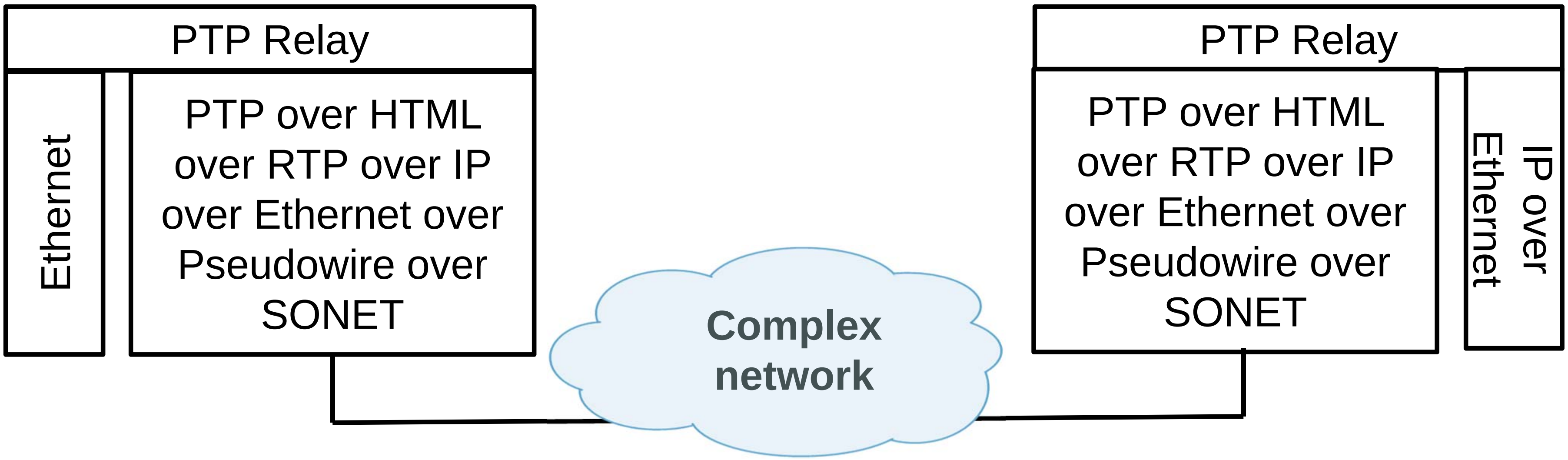
- Each PTPR port has an **independent protocol stack**.



The Answer: **The PTP Relay (PTPR)**

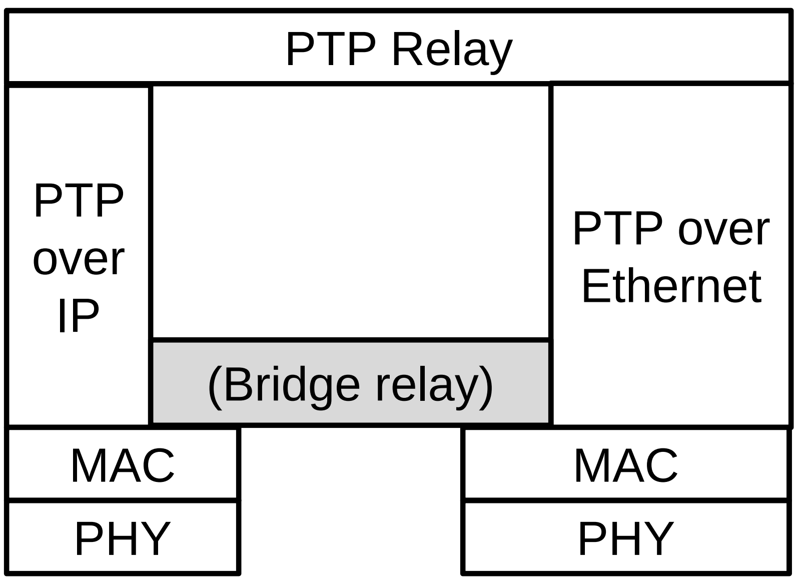
- **PTP is a user of the network(s), not integrated inside the networking functions.**

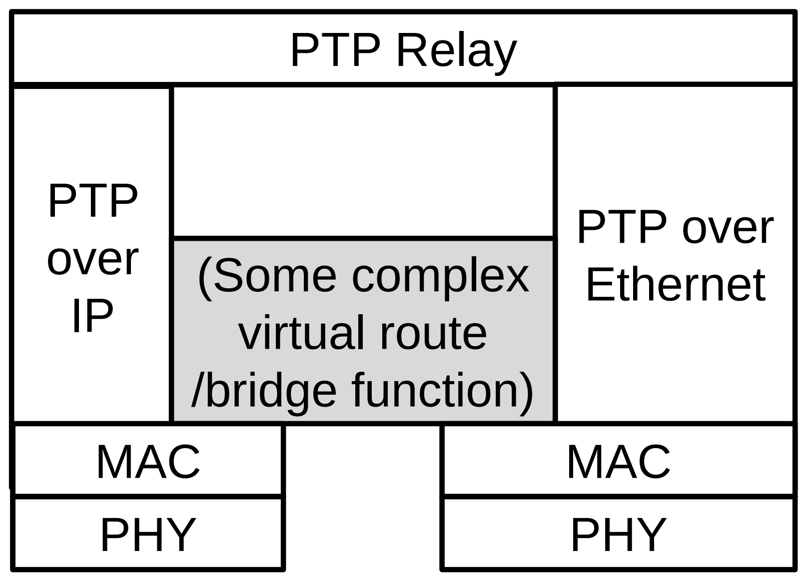
- The PTPRs are peers over their links, but each link is totally independent, just as bridges and routers connect different media.



The Answer: **The PTP Relay (PTPR)**

- It doesn't matter what else is in the box.
- A **router** + PTP Relay can use PTP over **Ethernet**!
- A **bridge** + PTP Relay can use PTP over **IP**!
- **Not understanding this is why the transparent clock causes so much confusion and controversy.**

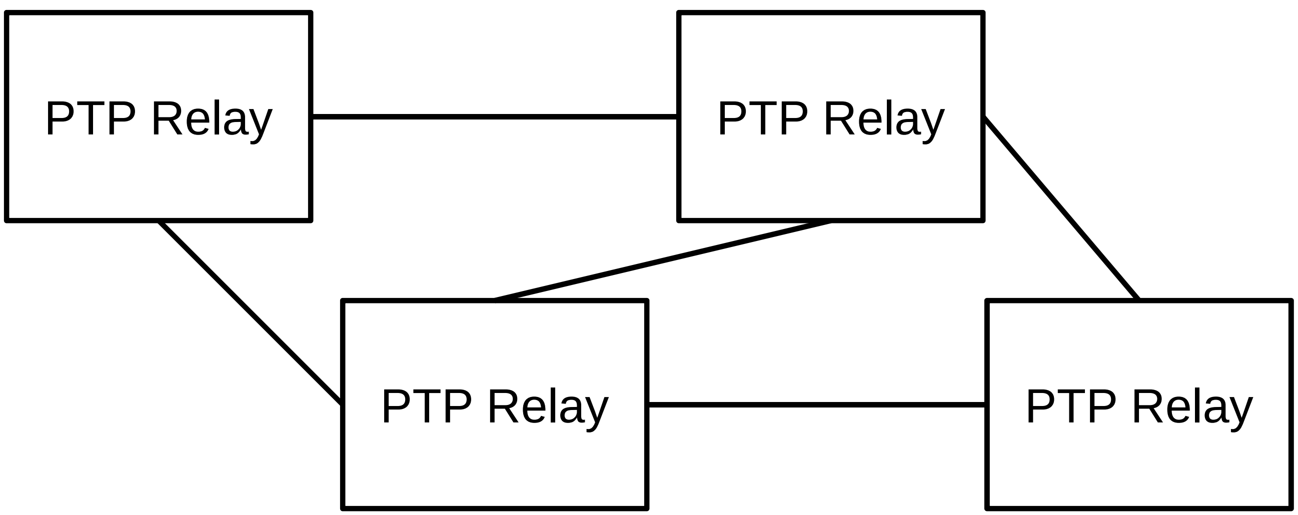




The Answer: **The PTP Relay (PTPR)**

- **PTP is a user of the network(s), not integrated inside the networking functions.**

- IEEE 1588, or **even a 1588 profile**, is concerned **only** with a network of PTPRs.



The PTP Relay

- The PTPR operates on Protocol Data Units (PDUs), not packets or frames! It **eats** PDUs on one port and **spews** PDUs on another port (or ports).
- The PTPR does **not** depend on a co-resident router or bridge to forward its transparent clock PDUs. It has its own rules.
- The box holding the PTP Relay can be a multi-port PC host without any bridging or routing, it can be a bridge, a router, an FCOE switch, a complex virtual combination, or anything else.
- **No standard cares!** A “time aware TRILL brouter” is **not undefined**. It’s **not punted** by 1588 to some other standard. The PTP Relay is **simply independent** of whatever else is in the box. (It’s called, “Layering”.)
- **PTP is its own network!** (I’d call it “over the top,” except that the PTP network is closer to the hardware than the underlying net!)

Four jobs for 1588 or for profiles

- **Link discovery.** The PTP Relays must find out, through configuration, management, and/or discovery protocols, about their neighboring PTP Relays, and for each one, select which addresses and protocols to use to reach that neighbor.
- **Boundary management.** Almost all the computers in the world are connected somehow. The PTP Relays must limit the link discovery process to appropriate bounds. Are the boundaries permeable or not?
- **Time routing.** The PTP Relays must elect various grades of master and slave clocks, provide or not provide for losses or errors, and compute one or more distribution paths for the time sync messages, through some combination of configuration, management, and/or routing protocols independently of any L2/L3 network routing protocols.
- **Time.** Almost forgot. All that cool syntonization and milliHertz bandwidth stuff. :)

I'm not upsetting 1588 implementations!

- I'm sure that I'm the first to think of this.
- 1588 has already started down this path, by specifying that the source MAC address changes at each hop.
- The existing PTP "translation device" is perhaps another way of looking at what I'm talking about.
- Look carefully; I hope that this layering model requires no changes to the the packet formats. You can still integrate the transparent clock into a router in your implementation.
- There is little or no change to the existing protocols. (Although more are needed!)
- But, unless we adopt proper layering, we will continue to argue over the wrong problems, and leave the right problems unsolved.
- **Worst case, a better-layered solution will supersede 1588.**

The conclusion

- “Virtual X” is a very bad thing, when it comes to PTP, for all values of X.
- Trying to piggyback on the L2 and L3 networks produces far more questions than answers, and generates far more work than it saves.
- Instead of having 1588 and/or profiles define a “time-aware X” for every X under the sun:
  - We let PTP Relays do their job on time.
  - We let networks do their job – connecting real and virtual devices, among which are PTP Relays. They do that really, really well, already.
- We get on with defining the protocols we need to do the time synchronization job.
  - Yes, some of them look similar to routing protocols. That gives us a boost.

Making the pig fly

- "That standard will be adopted when pigs can fly."
- Of course, anyone who works in the standards world knows that, given enough thrust, pigs certainly **can** fly.

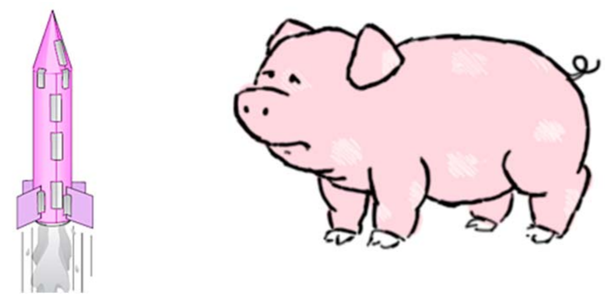
Making the pig fly

- **We have been trying to make the pig fly by genetically engineering a pig-bird** (by defining time-aware bridges, routers, Rbridges, etc.)**.**

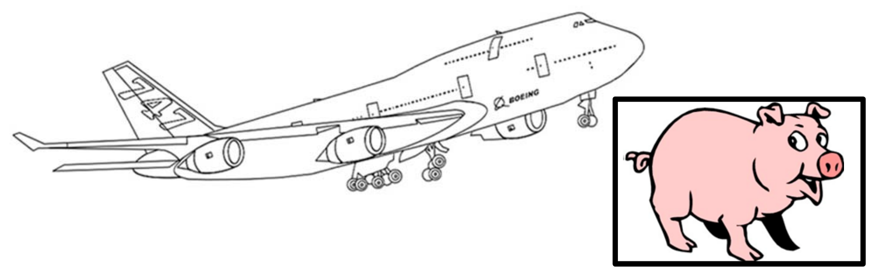
- At best, the result will be non-kosher chicken soup.
- At worst, we'll have feathers in our bacon.

Making the pig fly

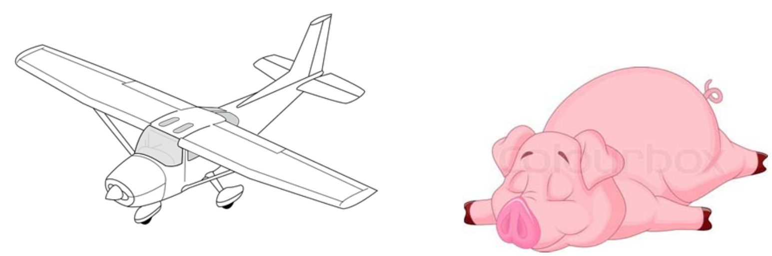
Behold! The ISO-layered flying pig.



- You can strap a rocket to his back and light it.



- You can crate the pig in a box and ship him in a 747.



- You can tranquilize him, put him in a Cessna, and hope he stays asleep until the pilot can land the plane.

Thank you.