Design Considerations for Packet Networks supporting Synchronous Ethernet and IEEE 1588v2

Pierre Bichon
Consulting Engineer
pierre@juniper.net

November 2009

Some Quotes for Consideration

“A synchronous transmission layer will always be there, everywhere”

“Only the transmission layer can deal with synchronization, it cannot be done elsewhere”

- Transmission Engineer

“Why do you need sync ??? This is legacy ! Sync is going the way of the dinosaurs”

“When everything has moved to IP and Ethernet, you don’t need sync anymore”

- Packet Network Engineer

Food for Thought

Can we reconcile both circuit and packet worlds and create something "better" ?

Synchronization can bring a lot to the packet world

- Circuit emulation and radio hand-off support
- Very accurate ToD (network monitoring, transaction management, etc)

Networks are changing – some decisions to be made:

- Do we need sync everywhere ?
- Is sync part of the infrastructure?
- Is sync a service?
  - Or both ?

Agenda

1. **Packet-based Architectures**
2. **Deploying SyncE**
3. **Deploying IEEE1588v2**
4. **Deploying SyncE and IEEE1588v2 together**

Agenda

1 **Packet-based Architectures**

2 Deploying SyncE

3 Deploying IEEE1588v2

4 Deploying SyncE and IEEE1588v2 together

Packet-based Architectures
Change is the Only Constant

Packet networks have more diverse topologies than SDH

- Mesh, rings, dual homed rings, stars, dual homed stars, chains, …
- Number of nodes per island (ex.: ring) ?
- Access vs Aggregation vs Core

What packet technologies are used

- Ethernet, MPLS, IP, but also flavors (ETHoSDH, MPLS-TP)
- And associated protocols

Who owns what

  - Mobile carrier, wireline carrier, wholesale
  - Trusted and un-trusted zones
- Requires boundaries with adequate features

What is (to be) deployed

- Existing (recent) network
- Greenfield deployment
- Network extension

Every network is different

Packet-based Architectures
Sync Strategy

Synchronization strategy options

- No network assistance
  - End-to-end transparent 1588v2, spot insertion of frequency/phase at the edge
- Partial network assistance
  - Boundary and transparent clocking support on selected nodes
- Full network assistance
  - Layer-1 transport of Frequency with SyncE
  - IEEE1588v2 with TC (and BC)
  - Including hybrid mode

Definition of “network assistance”

- Clock transfer/maintenance capabilities
- QoS enforcement
- Traffic Engineering
- Monitoring

Agenda

1 Packet-based Architectures

2 Deploying SyncE

3 Deploying IEEE1588v2

4 Deploying SyncE and IEEE1588v2 together

Deploying SyncE
Challenge #1: Network Design

Challenge:

- ALL network components in the path need to support SyncE

Reality check: Network Boundaries

- SDH to SyncE boundaries still not well proven (clock quality, traceability, loops)
- What about other boundaries (IEEE1588v2)
- Boundaries need expensive oscillators
- Where to start SyncE (in the core?)
- Complexity of meshed networks (not ring-only any more, like SDH)

SyncE will be deployed in an ever-changing environment

- Different from SDH - can be deployed link by link (careful design & validation)

This is layer 1= hardware upgrades

Virtualization ("I want my own clock")

Deploying SyncE
Challenge #2: OAM

Existing SSUs to support ESMC (traceability, loops, …)

Boundaries

- SDH/SyncE
- OTN/SyncE

ESMC Performance

- When under 100% load at 100 GE
- Convergence in a distributed chassis

Adding TLVs will

- Be beneficial for monitoring
- Increase complexity (performance?)
- Introduce interop issues between SyncE vendors and with SSUs
- Phase support will increase this even further (see next slide)

Deploying SyncE
Challenge #3: Phase Support using ESMC

ESMC re-use is a good target, however

- Why build another protocol for phase support ?
- What does really matter: the protocol, or the end-result ?
- How different from IEEE1588 with BC/TC ?
  - Could be less flexible (more embedded into systems)
- Still expensive oscillator needed at every hop (or every line card)

Potential issues

- Phase accuracy on a distributed chassis (to be designed into the PFE)
- Inter-working with IEEE1588 (TC, or BC for non-Eth boundaries)
- Will we need a hardware upgrade again ?

Caveats

- Less flexible than IEEE1588 (non Ethernet portions of the network ?)
- Virtualization ("I want my own clock")

SyncE is one part of the puzzle

Agenda

Deploying IEEE1588v2

Challenge: Network Design in an “un-deterministic” packet world

Ultimate target

- Phase accuracy (reach the microsecond level)
- Increase hop count between Master and Slaves
    - while maintaining frequency / phase accuracy

Jitter and wander

- PDV: QoS
- Network load: Traffic Engineering & CAC

Experience being built*

- Microwave
- xDSL – asymmetric bandwidth
- Network design for phase support

* not detailed here for the sake of time

Deploying IEEE1588v2

Challenge #1: Engineer for Jitter Part 1: Latency

PTP : 64-128 B
- reference: 128B

Similarity
- L2 vs L3
- IPv4 vs IPv6 vs MPLS LER/LSR
- IP vs L2/L3 VPNs

Centralized vs distributed

No QoS: Max goes to the roof

QoS or RFC2544: similar

What matters is the baseline (min/avg)
- The more information, the better (max to be limited)

Fully loaded system
- All ports
- Large switching / routing table
- Impact of other processing

Latency (microseconds)	Min/Avg	Max
1 GE - NPU	200	300
1 GE – L2 (centralized)	5	10
1 GE – L3 (centralized)	5	10
1 GE – L3 (distributed)	10-15	20
10 GE – L2 (centralized)	2	4
10 GE – L3 (centralized)	2	4
10 GE – L3 (distributed)	15-20	30

Vendor Dependent !

Deploying IEEE1588v2

Challenge #1: Engineer for Jitter   Part 2: QoS choices

Vendor Dependent !

Strict Priority Queue

- Always served (CAC needed) – no extra delay
- PTP + CES (+ VoIP) (+OAM) – packet sizes small and consistent
- Some max of SPQ – depends on the rest of queuing

High Priority Queue

- Credit could be negative: could be starved / buffered
- PTP: high drop probability, low buffer size
- VoIP, Business High, OAM

Network Control Queue

- Routing protocols have different requirements (could be buffered)
- Different profile from PTP traffic

N packets on the driver after the queue: MTU has an impact (less on 10GE) – check jumbo frames

Deploying IEEE1588v2

Challenge #1: Engineer for Jitter — Part 3: Implementations

Demystifying:

- "Switching matrix" - distributed systems are more sophisticated
- ETHoSDH may not be so perfect

To be checked

- Distributed systems (between line cards)

Vendor Dependent !

QoS with SPQ can compensate for bursts

- Avoid peaks of temporary buffering
- Avoid packet drops

What also matters is the long term PDV

- Engineer traffic along with QoS (see next topic)

Deploying IEEE1588v2

Challenge #1: Engineer for Jitter Part 4: QoS and TC

Good quality NEs down to the access are a good thing anyway

- Needed for other things (Business High, VoIP, LTE, etc)

Is TC a way to compensate

- For poor quality switches
- For the use of QoS
- Is that really simpler

Balance complexity / cost of

- Hardware-based QoS devices with SPQ
- TC capable devices

TC is the ultimate goal

- Will eventually be needed for high accuracy phase
- Not yet clear if needed everywhere
- Centralized vs distributed systems
  - requires intelligent calculation/distribution of residence time across Packet Forwarding Engines
- TC at 10GE / 100GE

JUNIPER NETWORKS

Deploying IEEE1588v2

Challenge #2: Engineer for Wander Part 1: Symmetry

Forward / reverse path

- Layer 3 routing protocols may be asymmetric, but MPLS solves the problem
- No more issues with asymmetric delays due to data rate steps (1GE, 10GE)
- Potential inter-work with Network Monitoring tools (TWAMP) to calibrate it

Check the NEs have the same behavior in upstream / downstream (QoS impact)

Physical networks:

- use fiber instead of copper
- xDSL: we have to live with it – slaves to cope with it

What matters is that the asymmetry keeps stable

Deploying IEEE1588v2

Challenge #2: Engineer for Wander Part 2: Network Load

New rules of NGMNs

- Network convergence
- Multiservice

Link load

- Core: 50%
- Edge/Access: US not at 100%
  - Fiber P2P star: DS 100%
  - Ring or MW chain: 100% on first DS link, then less and less

L2 or L3 protocols: little control (shortest path at best)

MPLS : brings control

- MPLS DiffServ Traffic Engineering to control congestion over paths
- MPLS CAC to control traffic classes
- Ramp up can better be managed

Wholesale islands

- Require strong SLAs

Deploying IEEE1588v2

Challenge #2: Engineer for Wander Part 3: Delay jumps

Network failures, re-routing, operational mistakes

- There are also some planned/known delay jumps
- Special case of AMP

What matters : a delay jump is stable and deterministic

MPLS DiffServ Traffic Engineering to manage delay jumps

- TE path computation
- Detection
  - MPLS OAM P2P and E2E
- Protection
  - Fast Reroute (FRR): node - link - path
  - Make before break, Non-revert mode

Deploying IEEE1588v2
The role of MPLS for Sync – Also Multicast

Usual assumptions:

- Too complex to deploy and manage
- Access networks are not multicast capable

Converged networks require multicast anyway (multiplay)

Multicast MPLS is

- Simpler to manage
- More deterministic than IP multicast

Traffic Engineering can be used for Multicast also

- Including protection

Benefits

- Scale the PTP Master (direct impact on cost and stability)
- Increase PTP performance

Deploying IEEE1588v2
The role of MPLS for Sync – Sync as a Service

What is Sync: in a packet network, is sync part of transmission, or a service, or part of network control ? Something else ?

Usual assumption: MPLS cannot be everywhere, too expensive

New rules apply

- MPLS moving from routing to transmission
  - “The Purple Line” - Kireeti Kompella
- MPLS is already in the core
- MPLS is already in the aggregation
- MPLS can be as far as in the access (MPLS-TP is an example)

MPLS to bring flexibility

- Decouple transport and service
  - « Seamless MPLS » (draft-leymann-mpls-seamless-mpls-00.txt; Oct. 20th, 2009)
  - Sync as a service
  - 1588 Masters are Service Nodes
- Direct and protected connections to the Masters
  - E-Line, L2 VPN, L3 VPN
- Network engineering without constraints: place / move your Service Nodes freely

Agenda

1 Packet-based Architectures

2 Deploying SyncE

3 Deploying IEEE1588v2

4 Deploying SyncE and IEEE1588v2 together

Comparing SyncE and IEEE1588v2

	Synchronous Ethernet	IEEE 1588v2
Ease of Deployment	Disruptive	Non-Disruptive (except if TC and BC are needed)
Cost of Deployment	Inexpensive for greenfield Expensive for upgrade (Capex +Opex)	Inexpensive (except if TC and BC are needed)
Ubiquity	No (Ethernet only)	Yes (Ethernet, xDSL, Microwave, Cable, FTTx, femtocell, etc)
Frequency/Phase Future-proof	No - Frequency only (Except if ESMC used for Phase But HW upgrades)	Yes (Frequency & Phase)
Deterministic behavior	Yes	Under certain conditions (better if using BC and TC)
Requires Careful Engineering	Yes	Yes
Virtualization	No	Yes (if overlay)

Combining SyncE and IEEE1588v2

Why combined SyncE & IEEE1588v2 ?

- Algorithms to enhance quality of both; clock failover; cross boundaries

It is not only a slave problem

- Also in any boundary node (islands)
- It makes sense to plan BC and SyncE in the same NE
- Will eventually make sense to combine TC and SyncE in the same NE

Vendor Dependent !

May well be required in every Node / POP

  - Where applications are to be supported

Open question: what is the best way to implement ?

- Combined algorithms ("hybrid mode")
- Combined protocols (ESMC & TC ?), including OAM
- Hardware upgrade ?

SyncE & IEEE1588v2 are complementary

Key Takeaways

Adding clocking to packet networks can be gradual

- Actual deployment will take time

MPLS and Traffic Engineering can play an important role

- e.g. “Sync as a Service”


JUNIPER
NETWORKS
TheNewNetworkIsHere.com
THE NEW
NETWORK
IS HERE.

everywhere