Clock Recovery and Channelized SDH/SONET

Tao Lang

Wintegra Inc.

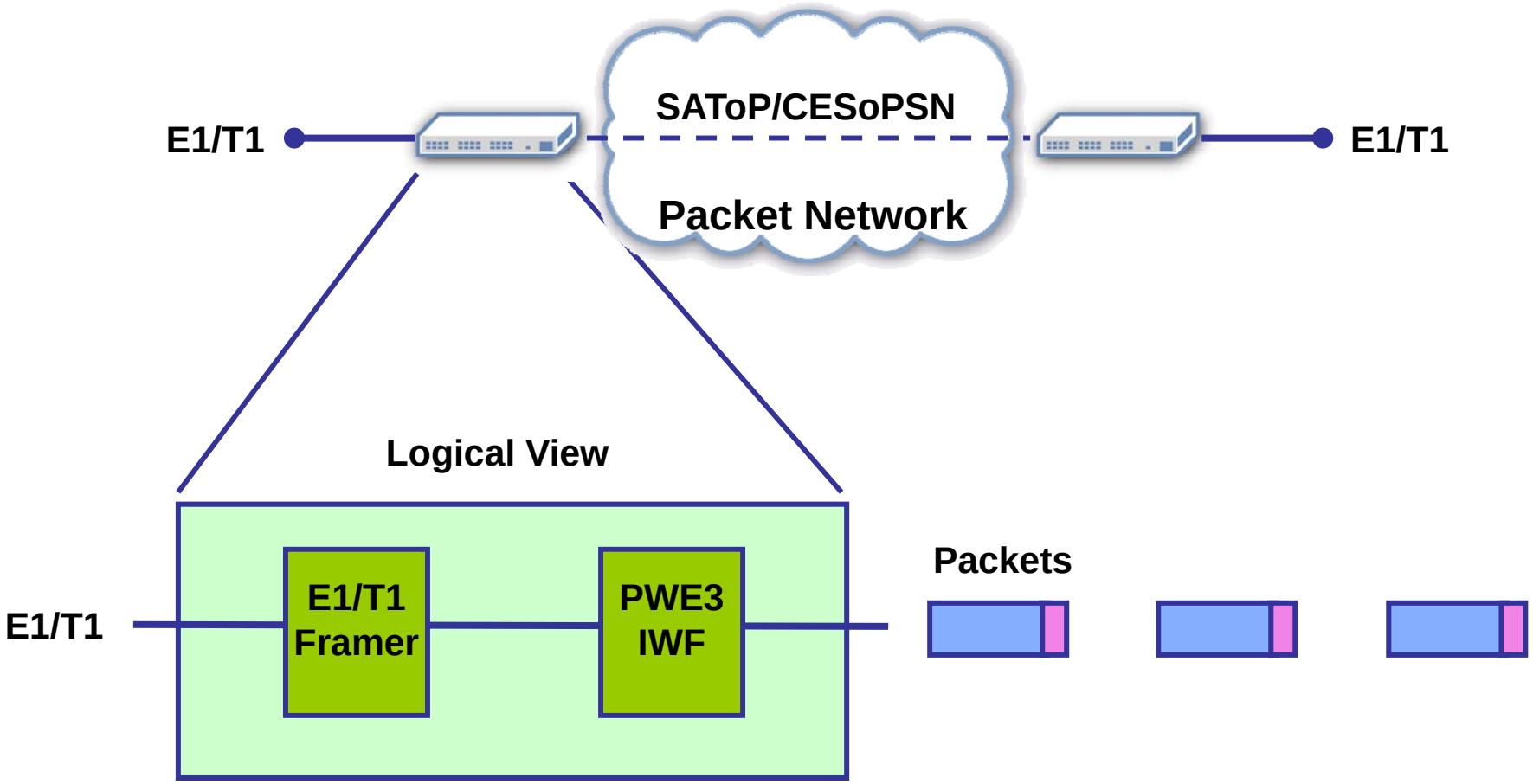
tao@wintegra.com

+1-512-345-3808

A Simple Agenda

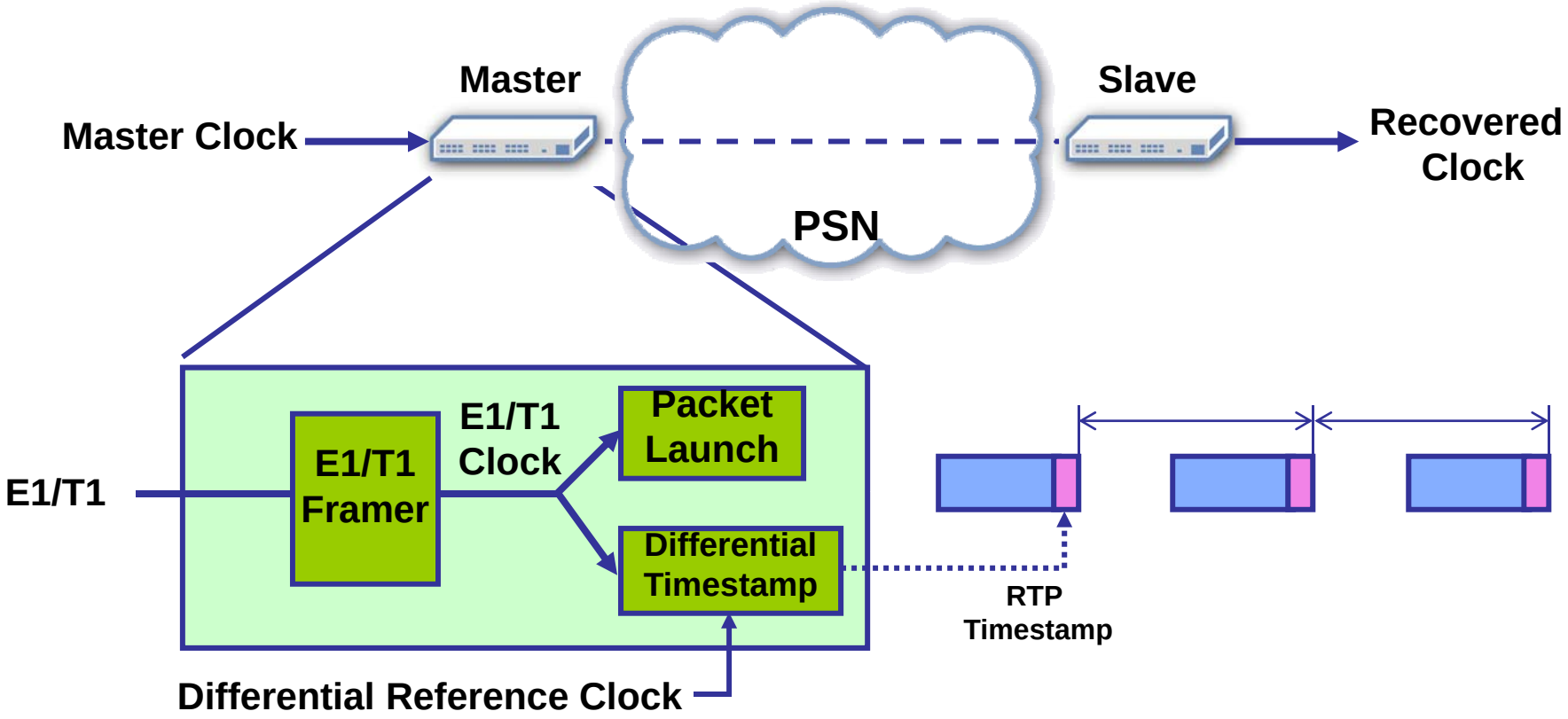
- Problem Statement
- Solution
- Summary

E1/T1 Pseudo-wire



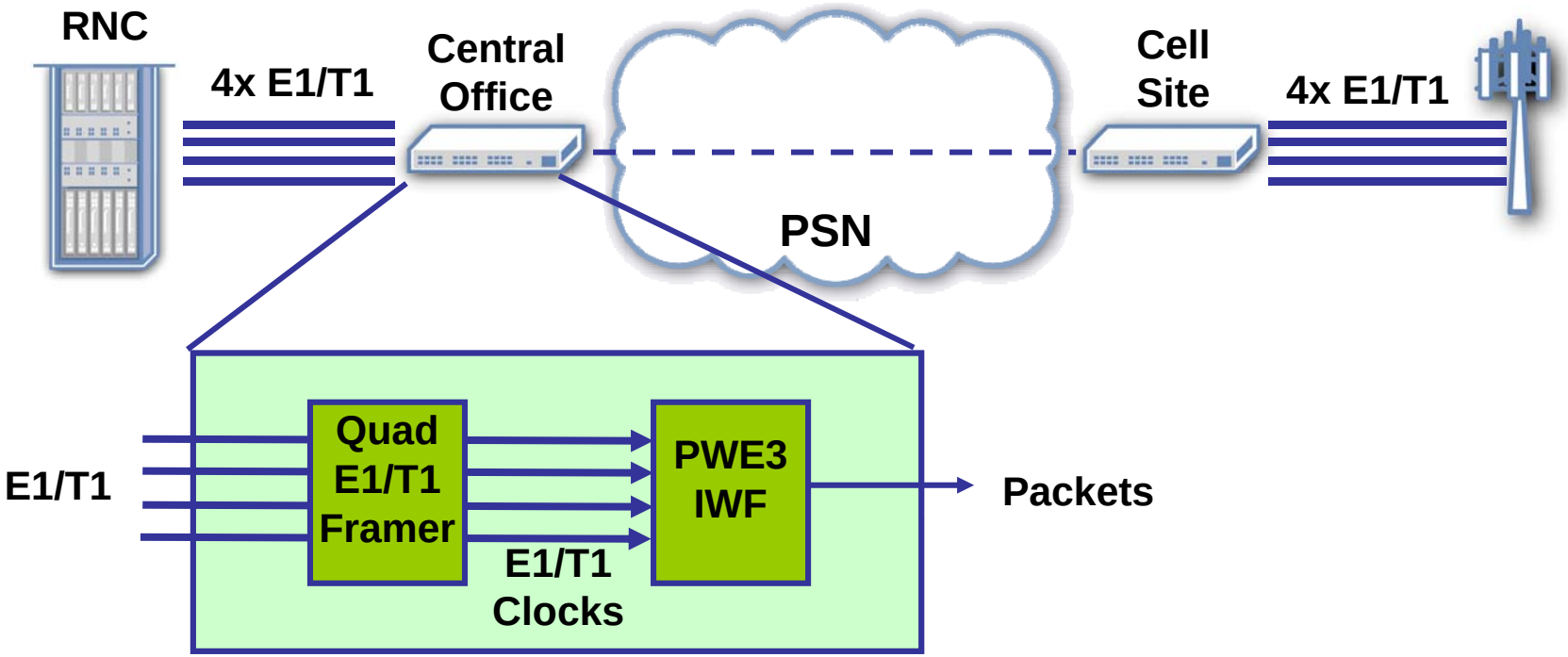
PWE3: Pseudo Wire Emulation Edge-to-Edge
IWF: Interworking Function

Clock Recovery



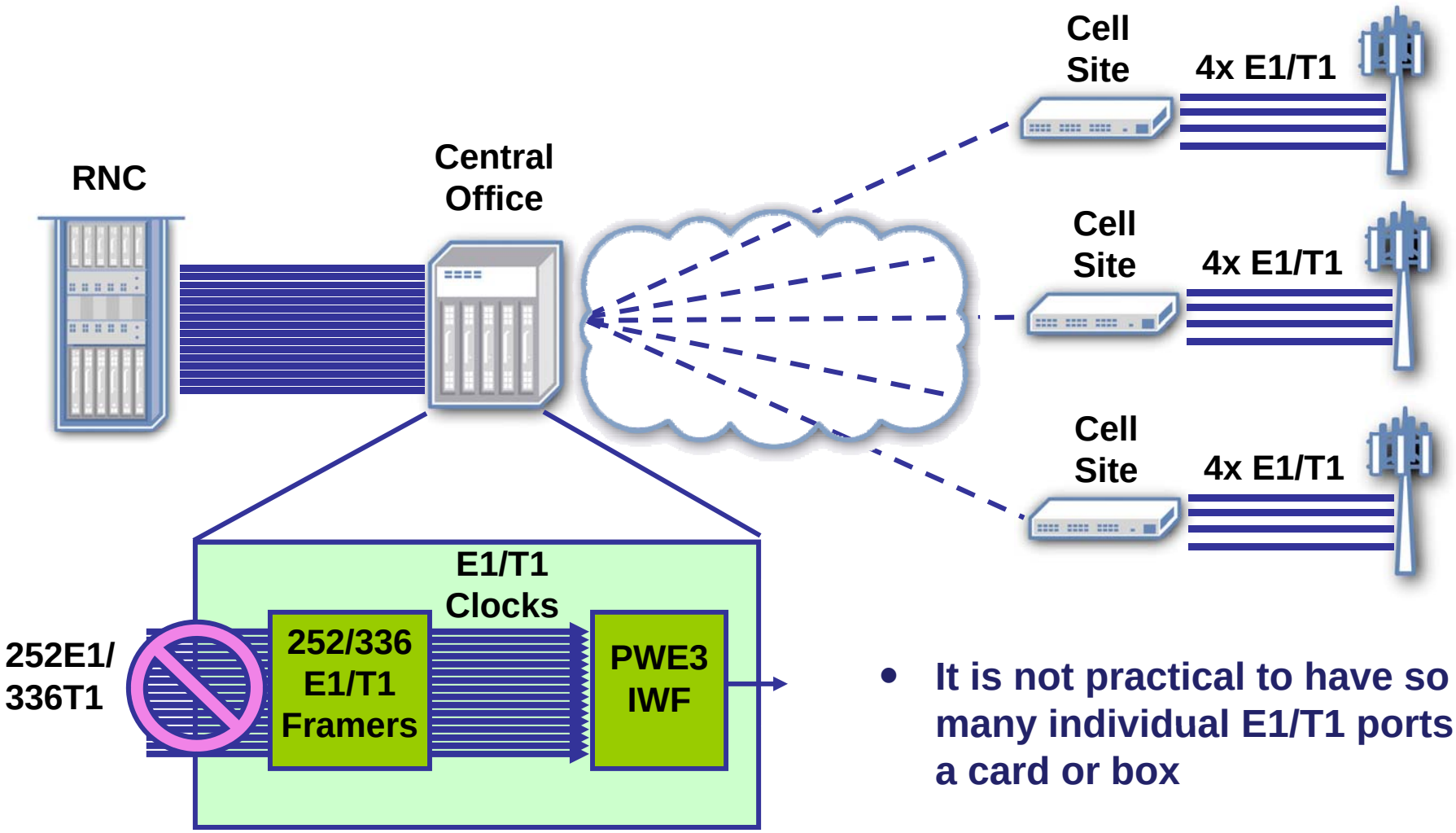
- For adaptive clock recovery, master clock is encoded in packet launch time
  - Slave will have to recovery the master clock from packet inter-arrival time
- For differential clock recovery, master clock is encoded in timestamp
  - Slave will have to recovery the master clock from differential timestamp

Apply PWE3 to Wireless Back-Haul Application



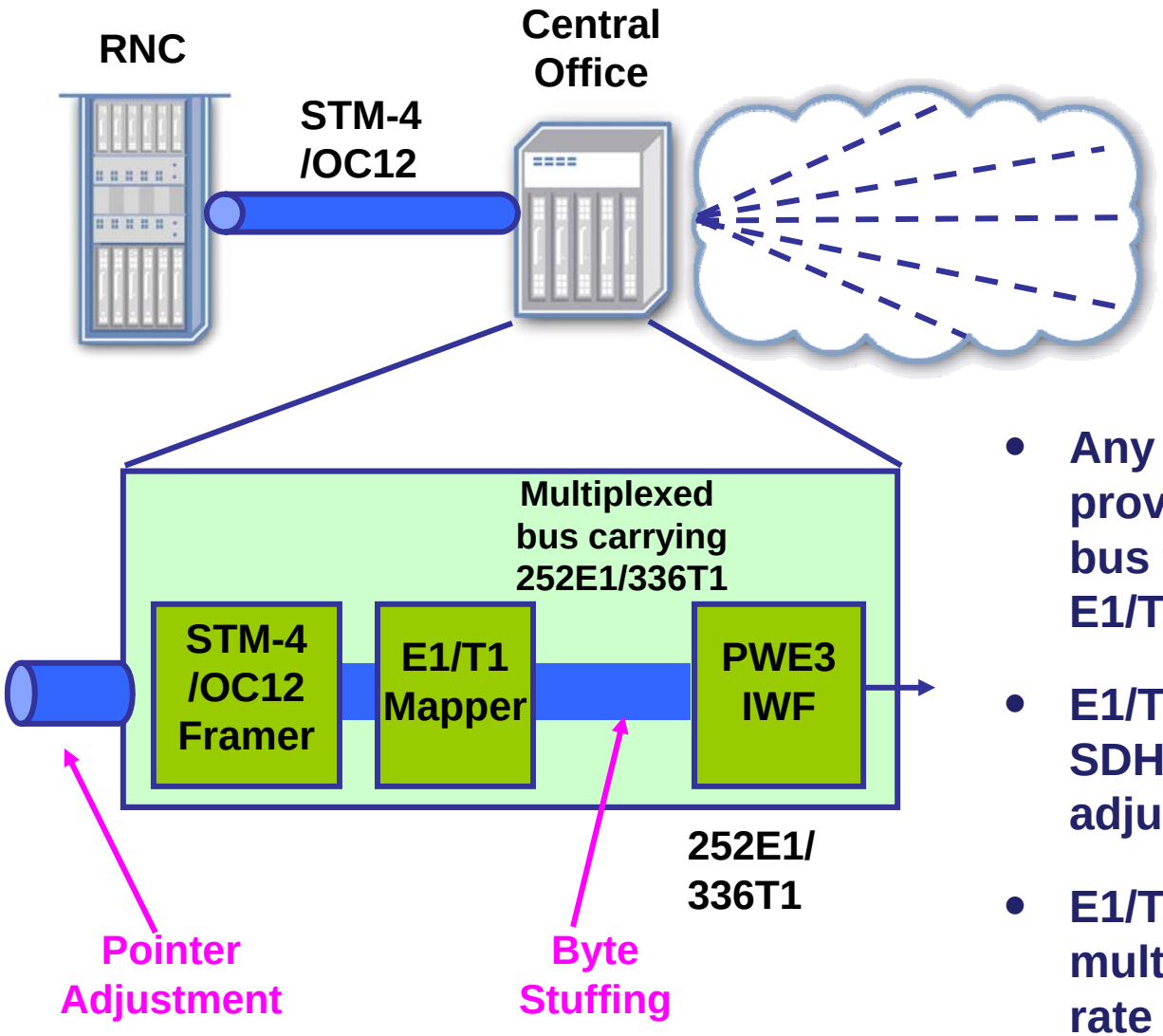
- Timing is distributed from RNC to base-stations
- Clock recovery wise
  - CO box is the master end → Encode the master clock in TDM-to-PSN direction
  - Cell-site box is the slave end → Recover the master clock in PSN-to-TDM direction

If We Try to Scale Up ……



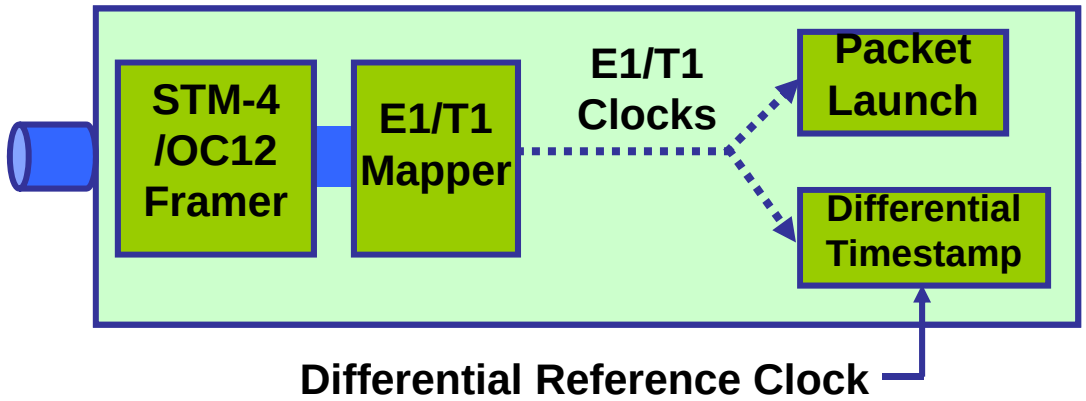
- It is not practical to have so many individual E1/T1 ports on a card or box
- Must use channelized SDH/SONET

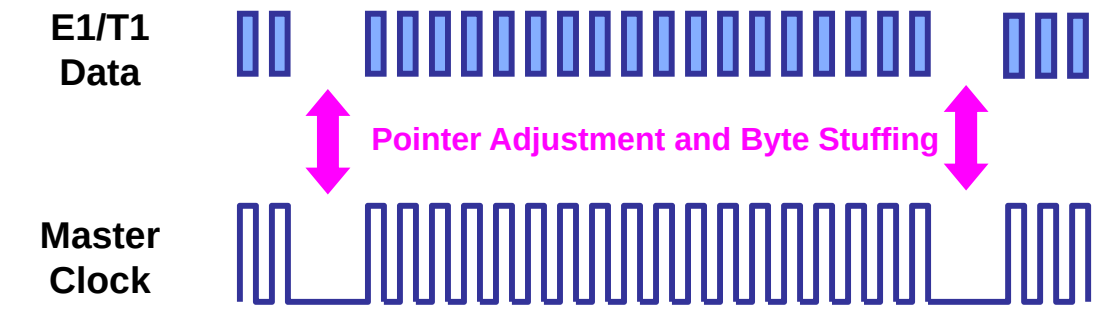
Adding STM-4/OC12 Interface ……

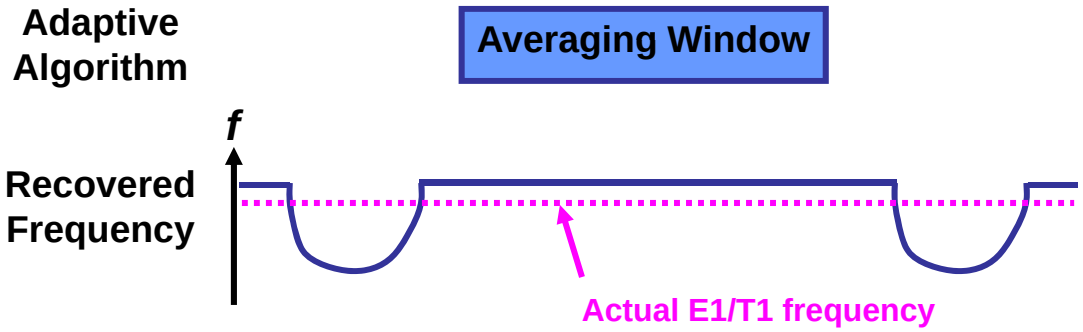


- Any off-the-shelf mapper only provides multiplexed parallel bus to carry large number of E1/T1 tributaries
- E1/T1 rate is adapted to SDH/SONET rate with pointer adjustment
- E1/T1 rate is adapted to multiplexed bus (e.g. SBI) clock rate with byte stuffing

A Degraded Master Clock





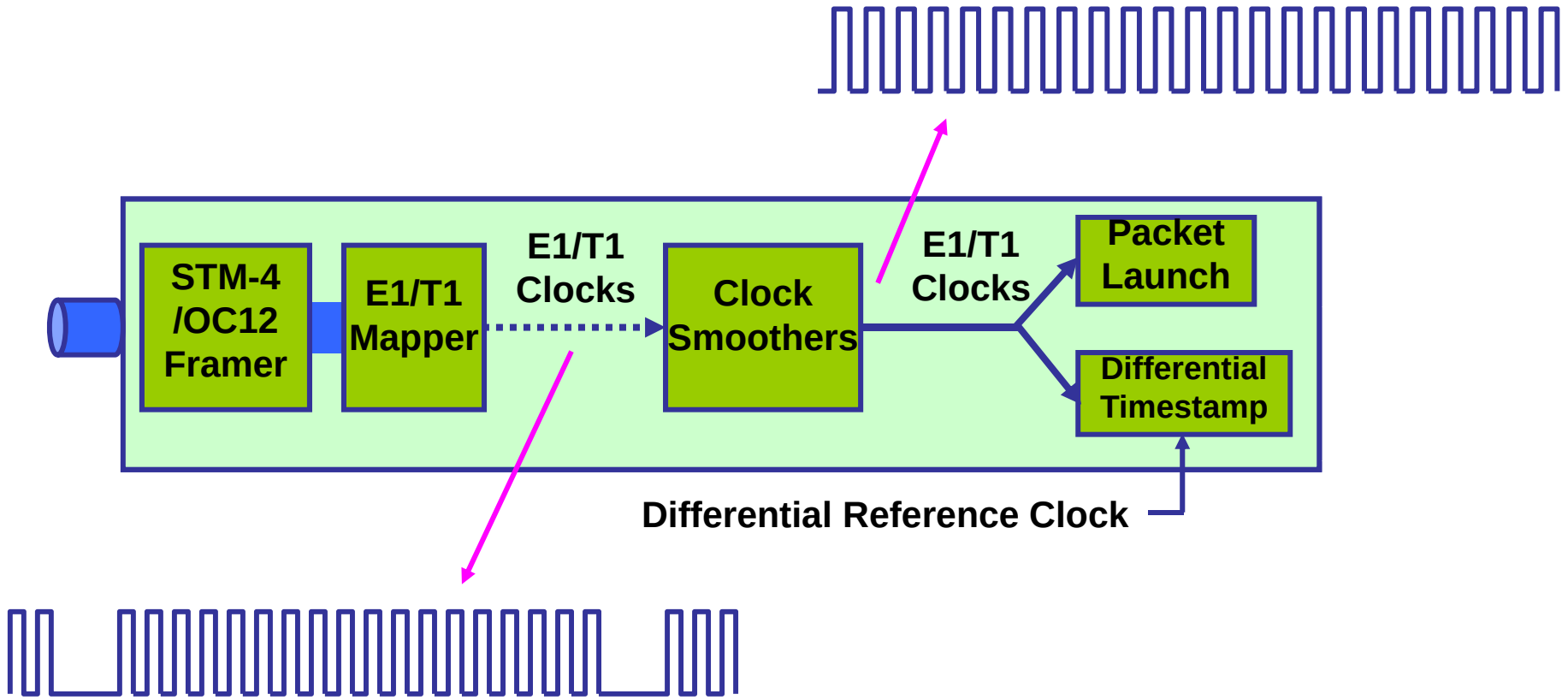


- The master clock seen by the PWE3 IWF is degraded
  - SDH/SONET pointer adjustment
  - Byte stuffing on multiplexed bus
- This will introduce wander to the master clock
- The wander frequency can be too low for the clock recovery algorithm to filter out

Problem Statement

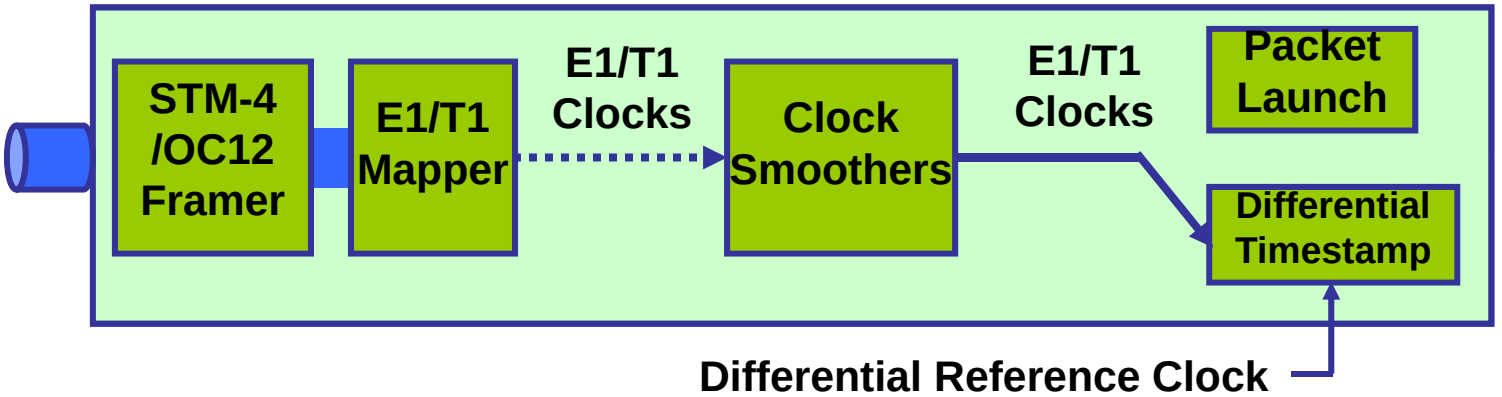
- E1/T1 clock recovery in channelized SDH/SONET has unique challenge
- Master clock will have wander due to pointer adjustment and byte stuffing
- Wander frequency can be very low when E1/T1, SDH/SONET and multiplexed bus frequencies are all close to each other
- Low frequency wander can not be filtered out by clock recovery algorithm running at the slave end

Solution



- Clock smoothing is necessary to remove wander at the master end

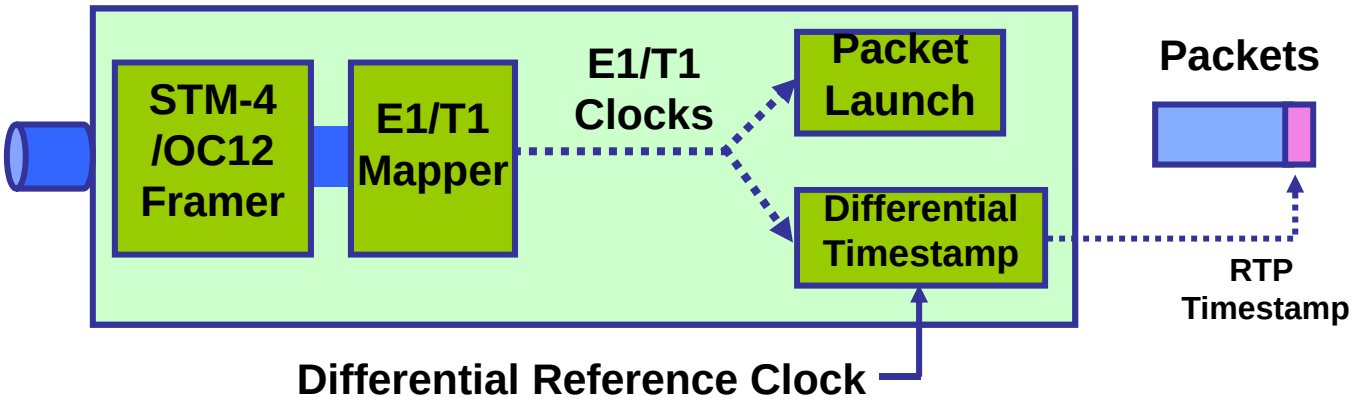
Differential Clock Recovery (DCR)



- Smoothed clock is used to generate differential timestamp
- Packet launch time is not critical
- Need one smoother per E1/T1
  - Assuming all E1/T1 are independent
  - Up to 252/336 smoothers for channelized STM-4/OC12

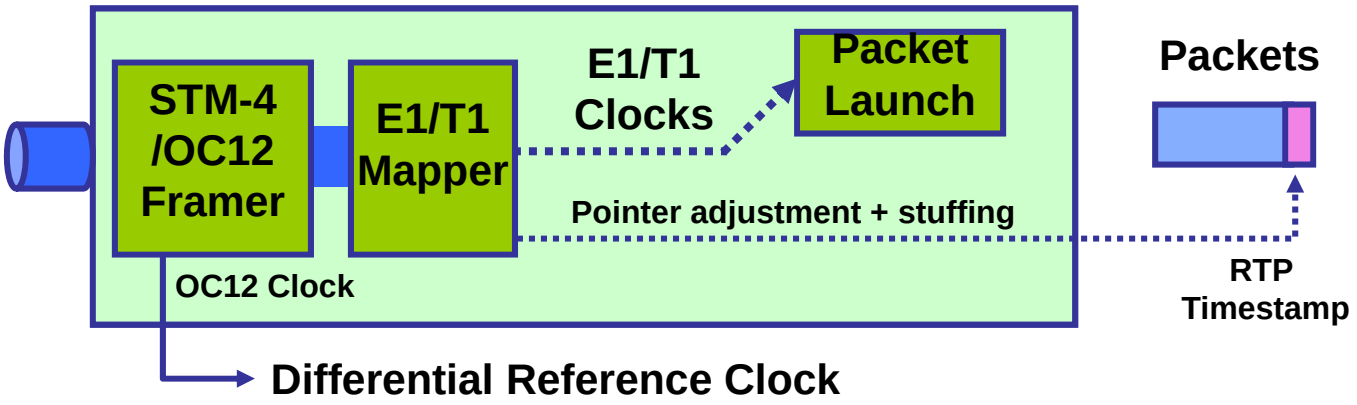
- A cost effective implementation is the key

Can We Get Rid of the Smoother?



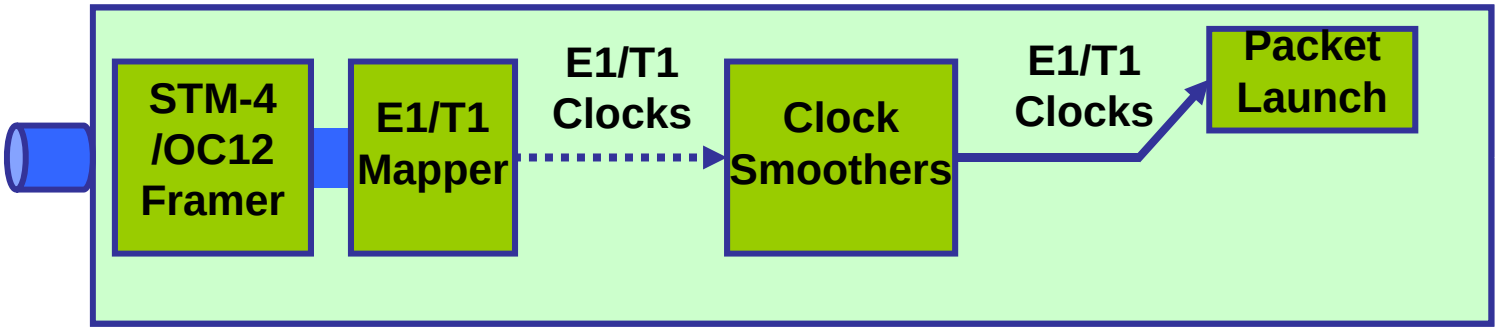
- The current way of generating differential timestamp (based on RTP) requires the E1/T1 clock
- To get rid of the smoother, differential timestamp must be re-defined

Can We Get Rid of the Smoother?



- The current way of generating differential timestamp (based on RTP) requires the E1/T1 clock
- To get rid of the smoother, differential timestamp must be re-defined
- One example is to pass the pointer adjustment and stuffing indication in the packets
- Requires standardization work to push forward

Adaptive Clock Recovery (ACR)



- Smoothed clock is used to launch packets
- Some implementation details
  - Clock distribution
    - One clock may drive multiple CESoPSN pseudo-wires if DS0s are from the same E1/T1
    - Load balancing is preferred to avoid transmitting all packets in a burst
- But do we need one smoother per E1/T1
  - How many independent clock sources or clock domains can ACR support?

Beating Effect

- First, there is this beating effect between asynchronous streams
  - Outlined in ITU G.8261 10.1.2.3
- Beating effect could introduce low frequency wander that can not be filtered out by ACR algorithm
- Clock domains must be limited for ACR applications

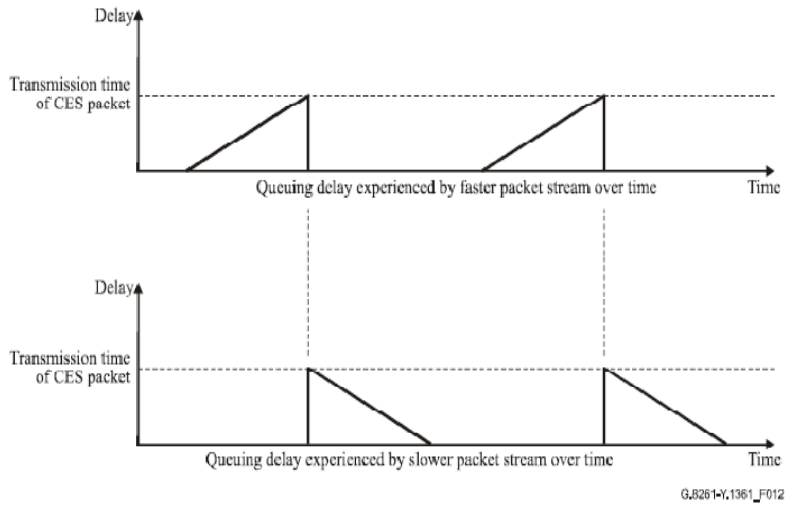


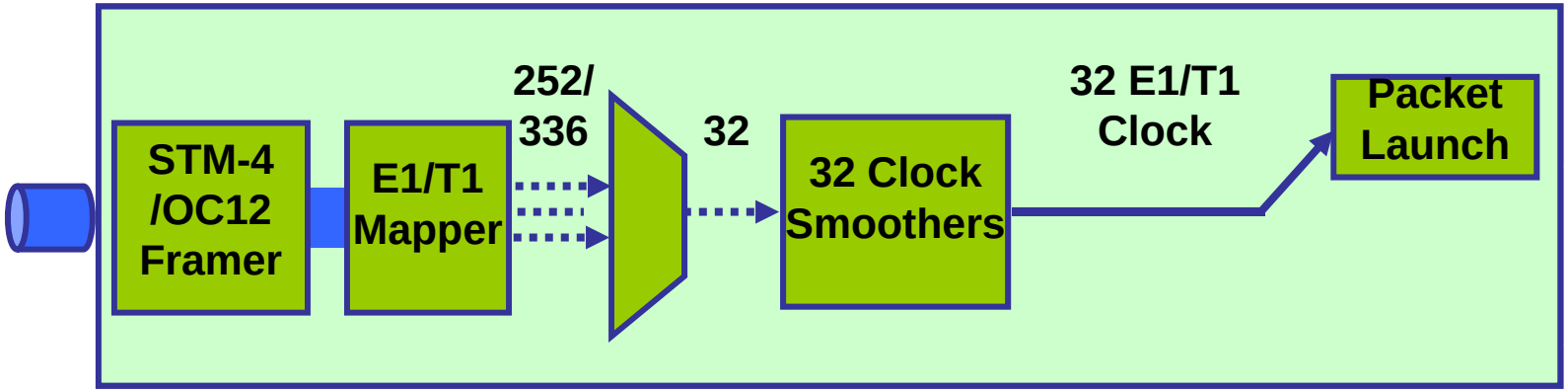
Figure 12/G.8261/Y.1361 – Delay profile experienced by beating packet streams

Eventually, the packets in stream 2 start to arrive at the switch or router ahead of those in stream 1, and the queuing delay is removed. At this point, it is stream 1 which now sees a queuing delay. This steadily declines until the packets in stream 1 arrive at the switch after the packets in stream 2 have completed transmission.

The duration of time that the packet streams experience queuing delay (i.e., the width of the triangles in Figure 12) is inversely proportional to the difference in rate between the two packet streams. Where the packet rates are very close, the duration may be extremely long. This long-term variation in the delay may cause a slow phase wander in any clock recovered from one of the packet streams.

Where multiple asynchronous constant bit rate streams share the same packet link, the effect may be additive. In the worst case, packets from all streams may line up exactly yielding the maximum queuing delay, although the frequency of this combined beat will reduce with the number of streams.

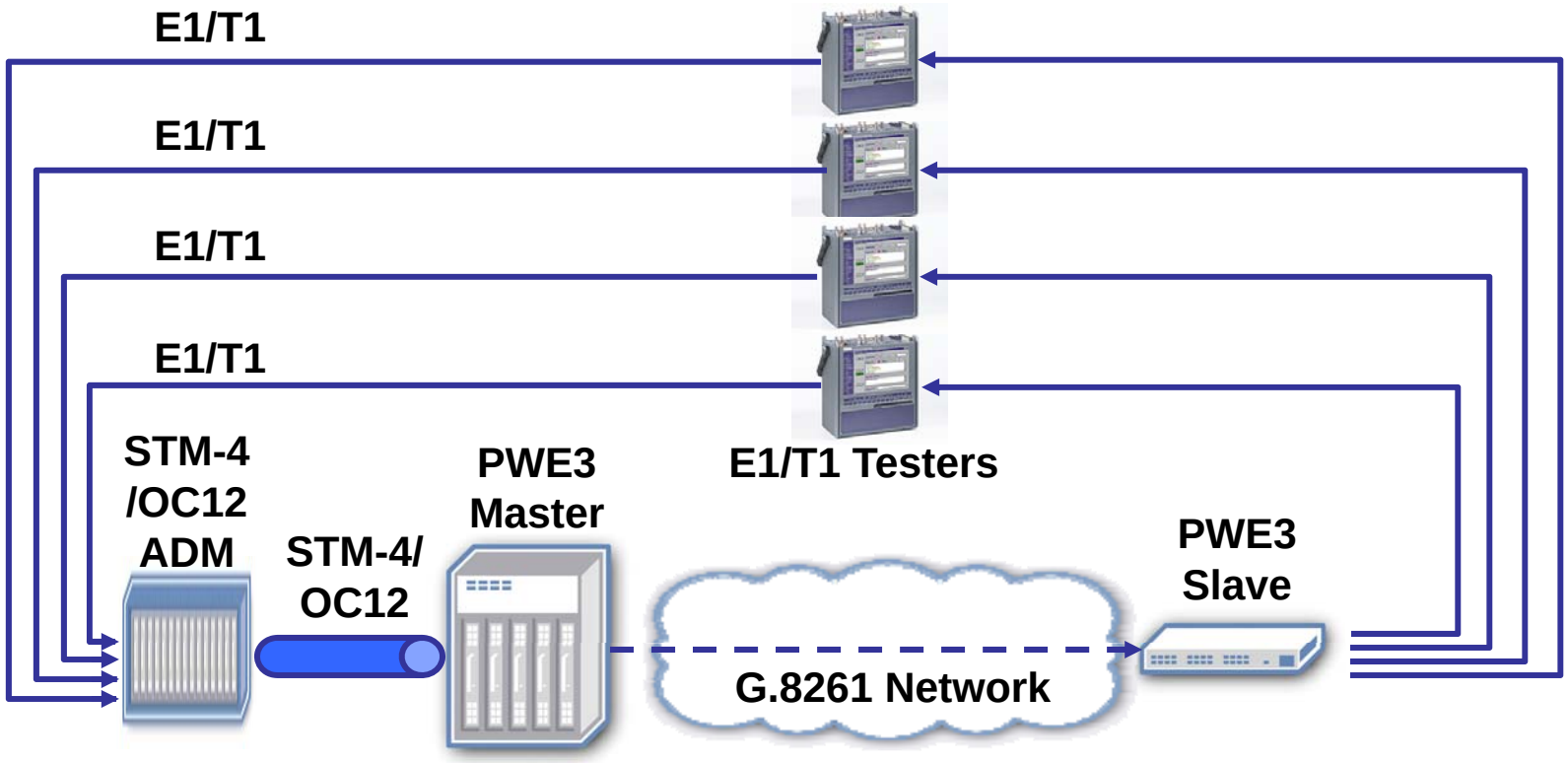
ACR with Limited Clock Domains



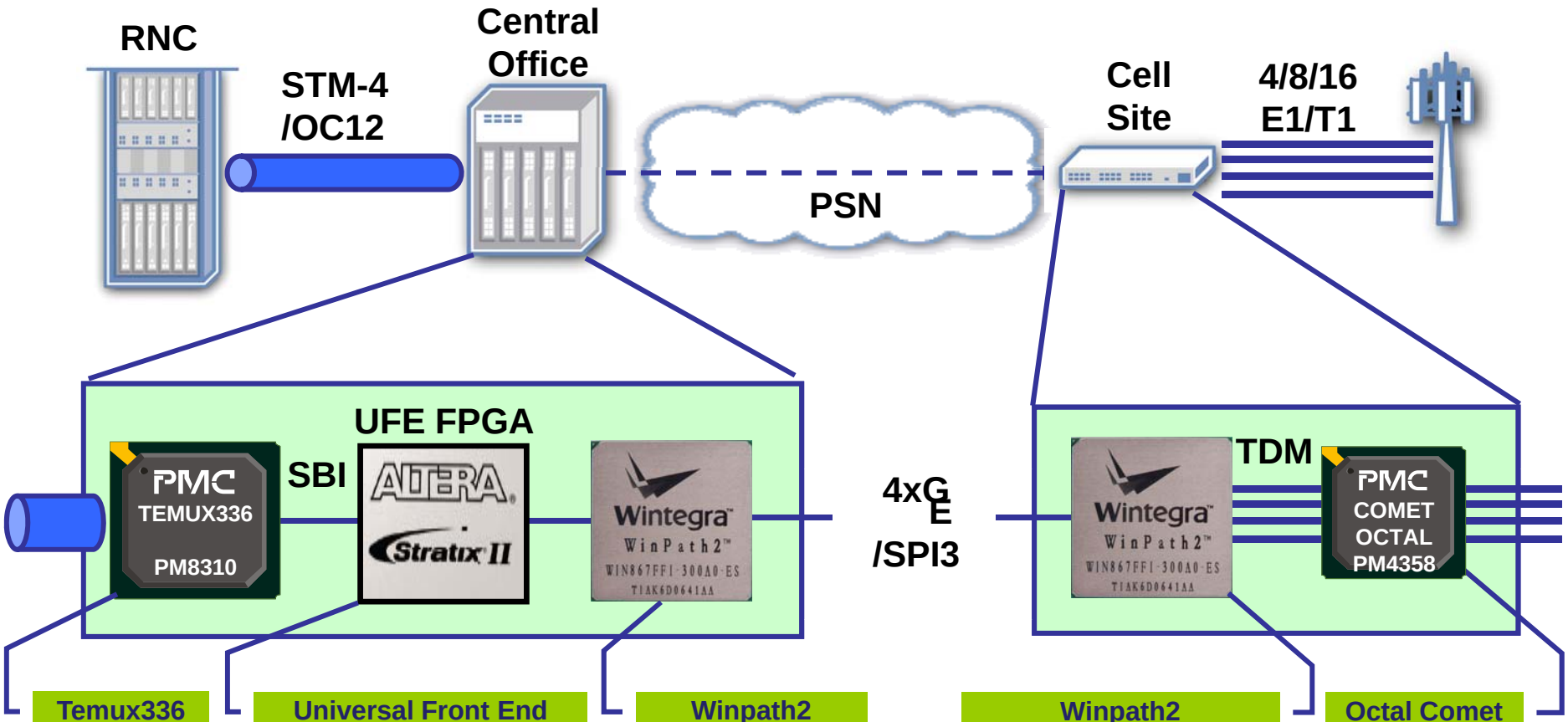
- One smoother per clock domain
- To save resource and cost, limited number of smoothers (e.g. 32 or less) may be implemented for ACR
- Some implementation details
  - Which E1/T1 is selected to drive the smoother? Failure protection is a must
  - Clock distribution
    - One clock may drive multiple SAToP/CESoPSN pseudo-wires
    - Load balancing is preferred to avoid transmitting all packets in a burst
- A robust implementation is the key

Clock Recovery Measurement

- Channelized OC12 (master) to E1/T1 (slave)
- Multiple E1/T1 testers all in internal timing mode to emulate multiple clock domains
- E1/T1 clocks independent of STM-4/OC12 clock
- G.8261 modeled packet network



A Real Implementation



Temux336
- OC12 framer
- Mapper
- T1/E1 framers

Universal Front End
(IP from Wintegra)
- Per DS0 service
  - Transparent
  - ATM TC
  - HDLC
- 336 clock recoveries
- 336 clock smoothers

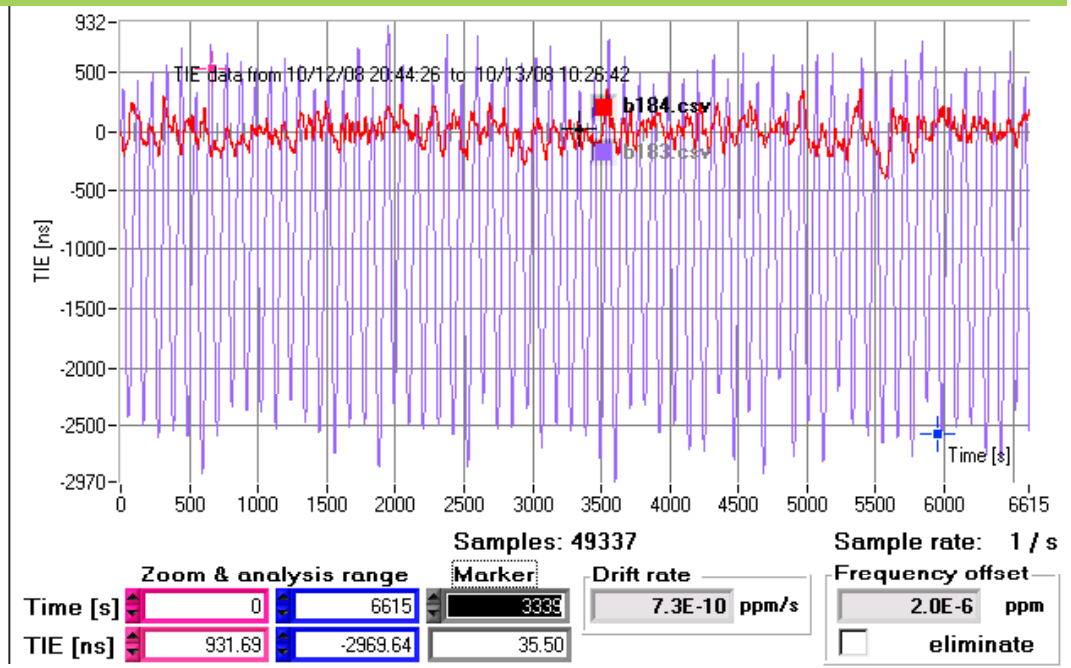
Winpath2
- SAToP
- CESoPSN
- ATM/IMA
- PPP/MLPPP
- FR/MFR
- ATM PWE3
- TM
- L2/L3 switch

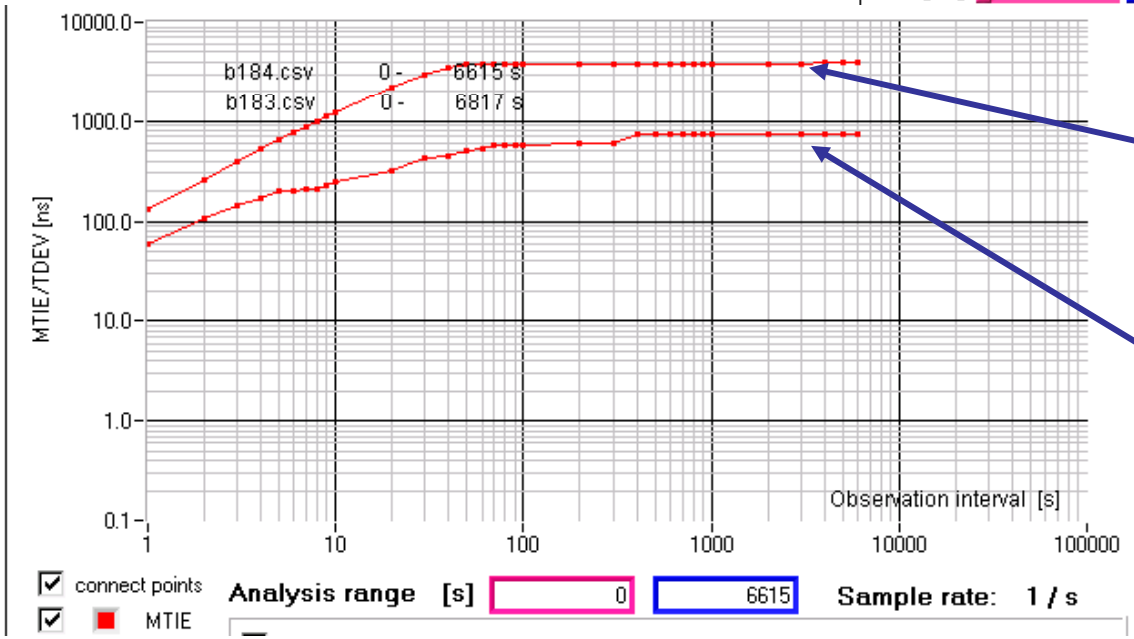
Winpath2
- SAToP
- CESoPSN
- ATM/IMA
- PPP/MLPPP
- FR/MFR
- ATM PWE3
- TM
- L2/L3 switch
- 16 clock recoveries

Octal Comet
- T1/E1 Framers

Test Result #1

- Channelized STM-4 to E1
- 1 clock domain
- No network
- STM-4 rate asynchronous (~0.001ppm offset) to the E1 under test
- TCXO at slave end
- Adaptive mode





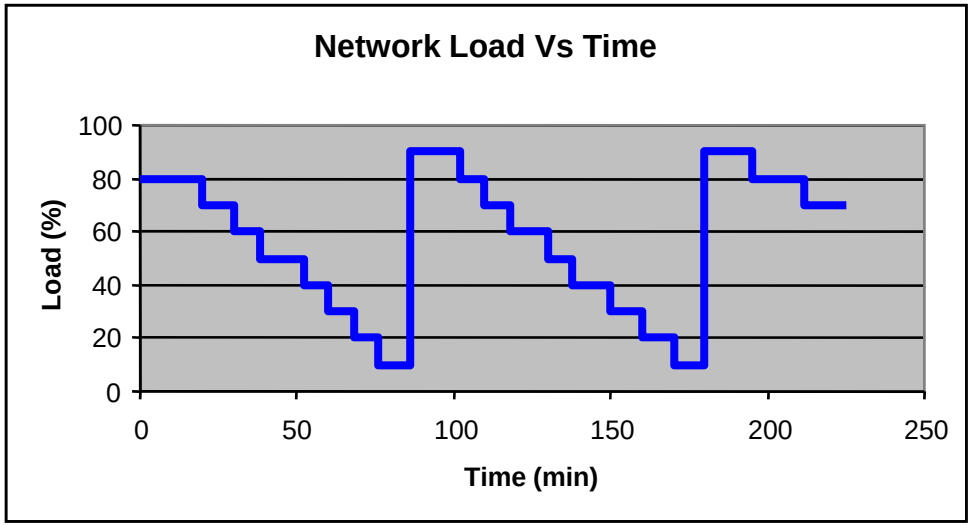
Without Clock Smoother
MTIE = 4us

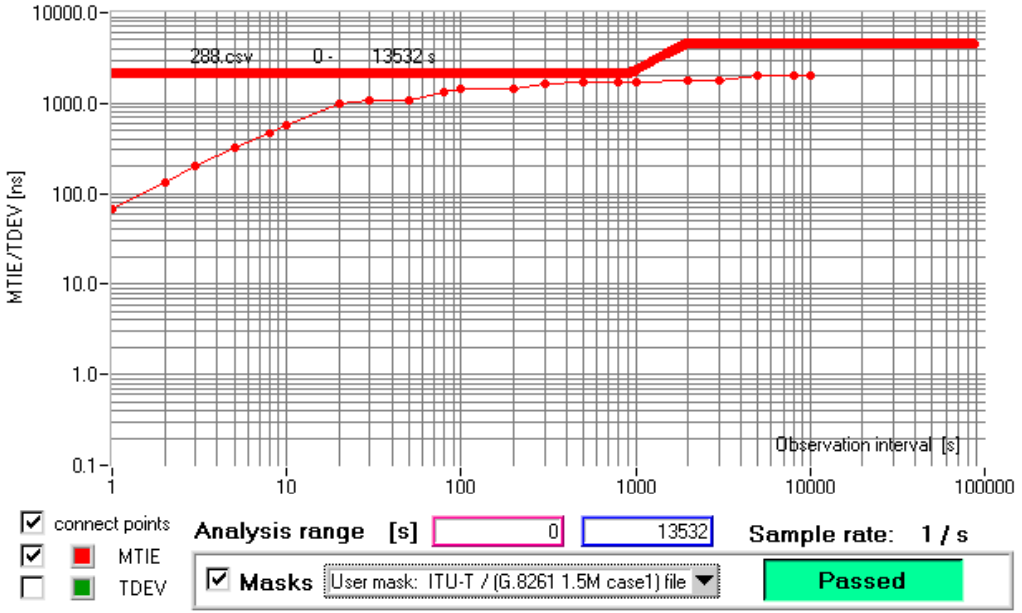
With Clock Smoother
MTIE = 800ns

Test Result #2

- 8 T1 pseudo-wires
- 4 clock domains
- OC-12 rate asynchronous (~0.001ppm offset) to the T1 under test
- 5 x GE switches
- Traffic loading per G.8261 VI.2.2.4
- 10% to 90% ramping
- TCXO at slave end
- With smoothers
- Adaptive mode

- MTIE = 2us





Summary

- Special care has to be taken when running E1/T1 clock recovery from channelized SDH/SONET
- The problem is identified as extra wander due to pointer adjustment and/or byte stuffing
- The problem can not be resolved by the clock recovery algorithm running at the slave end. It can only be resolved at the master end
- The implementation for ACR and DCR has different focus and considerations
- Solution is available in the market

Thank You

Tao Lang

Wintegra Inc.

tao@wintegra.com

+1-512-345-3808