ITSF 2013

Time & Sync in Telecoms

5-7 November, Lisbon, Portugal

In Partnership With:

Secure 1588 in HeNB / Femtocell Application

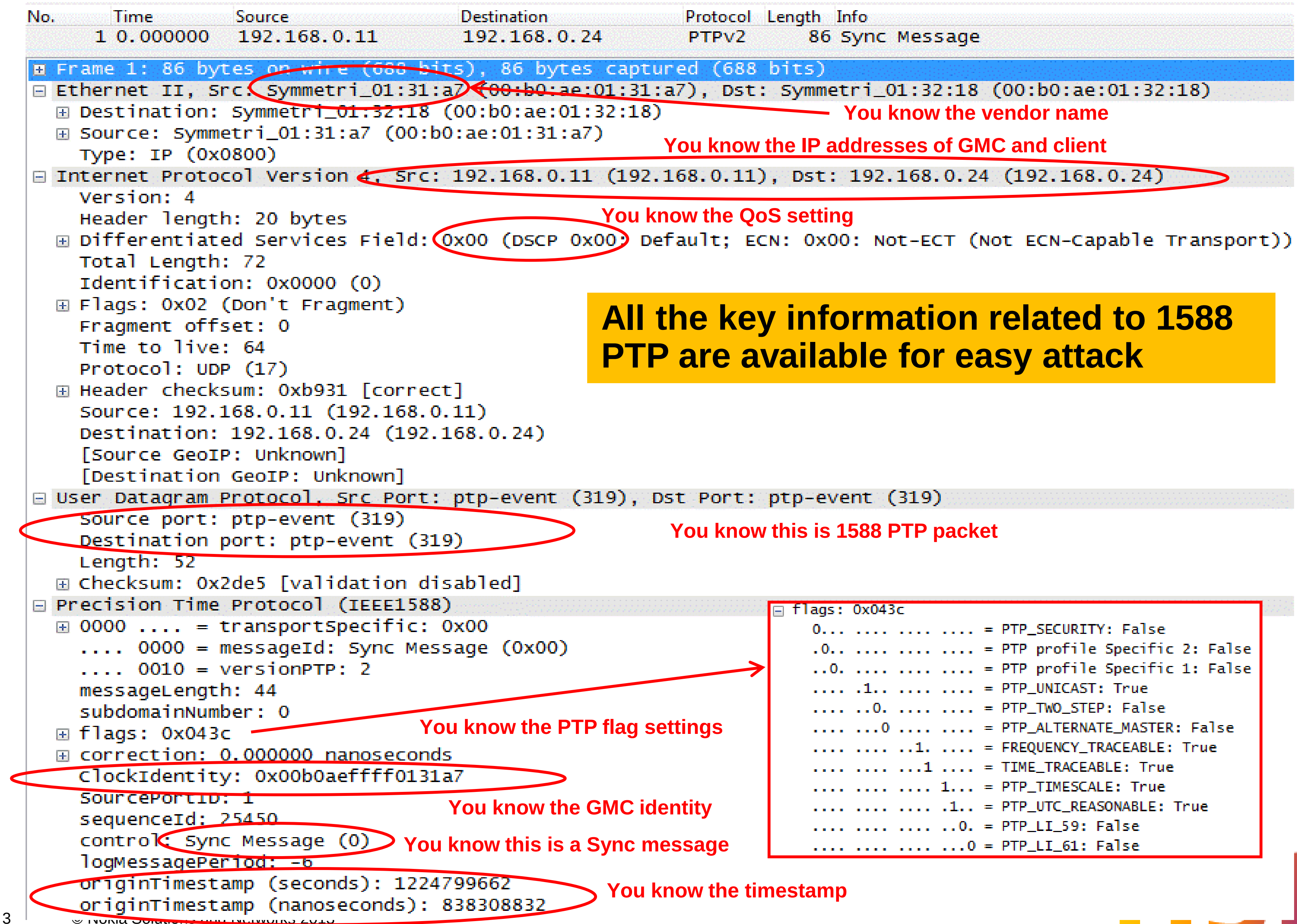
David T Chen, PhD
Senior Transport Architect
Nokia Solutions and Networks

5th November, 2013

BACKGROUND

- In 3GPP TS 33.320 (“Security of Home Node B (HNB) / Home evolved Node B (HeNB)”, Section 6.3.1 “Clock Synchronization Security Mechanisms for H(e)NB, it says: “The H(e)NB requires time synchronization with a time server. The H(e)NB shall support receiving time synchronization messages over the secure backhaul link between H(e)NB and the SeGW.”
- IEEE1588-2008 is the only timing over packet synchronization that can meet LTE/LTE-A frequency/phase/time-sync requirements, not NTP. However, IEEE 1588 is challenging to secure since it involves a large number of nodes that can spans across a large geographic area or multiple backhaul service provider domains.

1588 outside IPsec Tunnel Tells All (Sync)



```
No.      Time        Source           Destination       Protocol  Length  Info
      1 0.000000    192.168.0.11     192.168.0.24      PTPv2        86 Sync Message

⊞ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
⊟ Ethernet II, Src: Symmetri_01:31:a7 (00:b0:ae:01:31:a7), Dst: Symmetri_01:32:18 (00:b0:ae:01:32:18)
  ⊞ Destination: Symmetri_01:32:18 (00:b0:ae:01:32:18)
  ⊞ Source: Symmetri_01:31:a7 (00:b0:ae:01:31:a7)
    Type: IP (0x0800)
⊟ Internet Protocol Version 4, Src: 192.168.0.11 (192.168.0.11), Dst: 192.168.0.24 (192.168.0.24)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 72
    Identification: 0x0000 (0)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
  ⊞ Header checksum: 0xb931 [correct]
    Source: 192.168.0.11 (192.168.0.11)
    Destination: 192.168.0.24 (192.168.0.24)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ User Datagram Protocol, Src Port: ptp-event (319), Dst Port: ptp-event (319)
    Source port: ptp-event (319)
    Destination port: ptp-event (319)
    Length: 52
  ⊞ Checksum: 0x2de5 [validation disabled]
⊟ Precision Time Protocol (IEEE1588)
  ⊞ 0000 .... = transportSpecific: 0x00
    .... 0000 = messageId: Sync Message (0x00)
    .... 0010 = versionPTP: 2
    messageLength: 44
    subdomainNumber: 0
  ⊞ flags: 0x043c
  ⊞ correction: 0.000000 nanoseconds
    ClockIdentity: 0x00b0aeffff0131a7
    SourcePortID: 1
    sequenceId: 25450
    control: Sync Message (0)
    logMessagePeriod: -6
    originTimestamp (seconds): 1224799662
    originTimestamp (nanoseconds): 838308832
```

```
⊟ flags: 0x043c
    0... .... .... .... = PTP_SECURITY: False
    .0.. .... .... .... = PTP profile Specific 2: False
    ..0. .... .... .... = PTP profile Specific 1: False
    .... .1.. .... .... = PTP_UNICAST: True
    .... ..0. .... .... = PTP_TWO_STEP: False
    .... ...0 .... .... = PTP_ALTERNATE_MASTER: False
    .... .... ..1. .... = FREQUENCY_TRACEABLE: True
    .... .... ...1 .... = TIME_TRACEABLE: True
    .... .... .... 1... = PTP_TIMESCALE: True
    .... .... .... .1.. = PTP_UTC_REASONABLE: True
    .... .... .... ..0. = PTP_LI_59: False
    .... .... .... ...0 = PTP_LI_61: False
```

You know the vendor name

You know the IP addresses of GMC and client

You know the QoS setting

All the key information related to 1588 PTP are available for easy attack

You know this is 1588 PTP packet

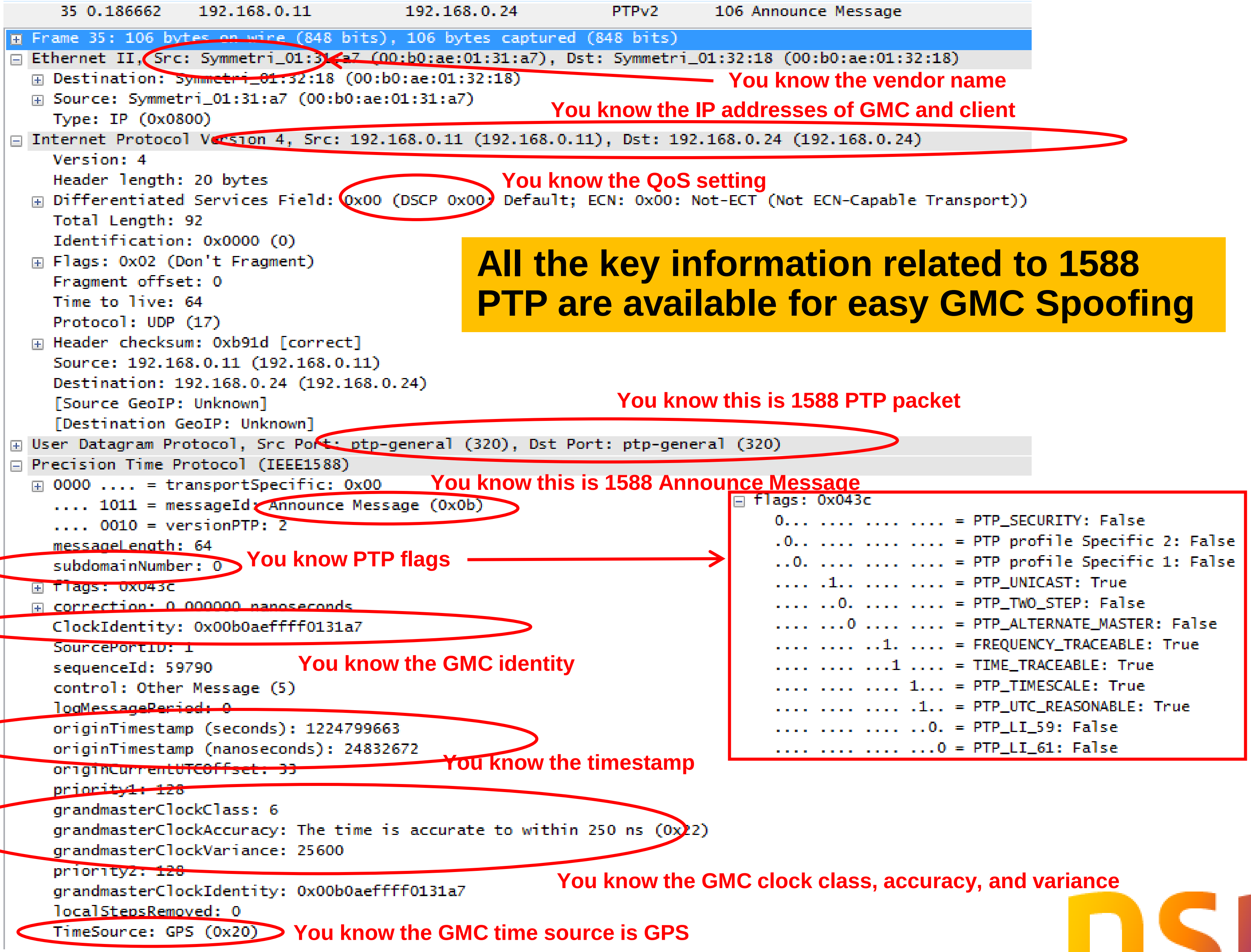
You know the PTP flag settings

You know the GMC identity

You know this is a Sync message

You know the timestamp

1588 outside IPsec Tunnel Tells All (Announce)



```
    35 0.186662     192.168.0.11          192.168.0.24          PTPv2      106 Announce Message
⊞ Frame 35: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
⊟ Ethernet II, Src: Symmetri_01:31:a7 (00:b0:ae:01:31:a7), Dst: Symmetri_01:32:18 (00:b0:ae:01:32:18)
  ⊞ Destination: Symmetri_01:32:18 (00:b0:ae:01:32:18)
  ⊞ Source: Symmetri_01:31:a7 (00:b0:ae:01:31:a7)
    Type: IP (0x0800)
⊟ Internet Protocol Version 4, Src: 192.168.0.11 (192.168.0.11), Dst: 192.168.0.24 (192.168.0.24)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 92
    Identification: 0x0000 (0)
  ⊞ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
  ⊞ Header checksum: 0xb91d [correct]
    Source: 192.168.0.11 (192.168.0.11)
    Destination: 192.168.0.24 (192.168.0.24)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊞ User Datagram Protocol, Src Port: ptp-general (320), Dst Port: ptp-general (320)
⊟ Precision Time Protocol (IEEE1588)
  ⊞ 0000 .... = transportSpecific: 0x00
    .... 1011 = messageId: Announce Message (0x0b)
    .... 0010 = versionPTP: 2
    messageLength: 64
    subdomainNumber: 0
  ⊞ flags: 0x043c
  ⊞ correction: 0.000000 nanoseconds
    ClockIdentity: 0x00b0aeffff0131a7
    SourcePortID: 1
    sequenceId: 59790
    control: Other Message (5)
    logMessagePeriod: 0
    originTimestamp (seconds): 1224799663
    originTimestamp (nanoseconds): 24832672
    originCurrentUTCOffset: 33
    priority1: 128
    grandmasterClockClass: 6
    grandmasterClockAccuracy: The time is accurate to within 250 ns (0x22)
    grandmasterClockVariance: 25600
    priority2: 128
    grandmasterClockIdentity: 0x00b0aeffff0131a7
    localStepsRemoved: 0
    TimeSource: GPS (0x20)
```

```
⊟ flags: 0x043c
    0... .... .... .... = PTP_SECURITY: False
    .0.. .... .... .... = PTP profile Specific 2: False
    ..0. .... .... .... = PTP profile Specific 1: False
    .... .1.. .... .... = PTP_UNICAST: True
    .... ..0. .... .... = PTP_TWO_STEP: False
    .... ...0 .... .... = PTP_ALTERNATE_MASTER: False
    .... .... ..1. .... = FREQUENCY_TRACEABLE: True
    .... .... ...1 .... = TIME_TRACEABLE: True
    .... .... .... 1... = PTP_TIMESCALE: True
    .... .... .... .1.. = PTP_UTC_REASONABLE: True
    .... .... .... ..0. = PTP_LI_59: False
    .... .... .... ...0 = PTP_LI_61: False
```

You know the vendor name

You know the IP addresses of GMC and client

You know the QoS setting

All the key information related to 1588 PTP are available for easy GMC Spoofing

You know this is 1588 PTP packet

You know this is 1588 Announce Message

You know PTP flags

You know the GMC identity

You know the timestamp

You know the GMC clock class, accuracy, and variance

You know the GMC time source is GPS

Security Concerns of IEEE1588-2008 PTP

- **Unlike NTP that was developed with security in mind from the beginning, IEEE1588 PTP's security provisioning is an after-thought**
- **Denial of Service Attack (GMC/Slave-aware Attack)**
  - An unsecure PTP message exchanges (ANNOUNCE, SYNC, DELAY_REQ, DELAY_RESP, etc.) will have the PTP packets easily captured (e.g., Wireshark) and show the IP addresses of the GMC and clients, UDP port numbers, 1588 frame headers in detail, etc.
  - A Denial of Service Attack floods the queues/processors with arbitrary, modified, or replayed PTP messages to deny the synchronization between slave and master clocks
- **GMC/Slave-aware PTP Message Manipulation (Man-in-the-Middle Attack)**
  - Removal of PTP messages, delay of PTP messages to cause timeout, modification of PTP messages to cause discarding
- **GMC/Slave-unaware PTP Message Tempering (Man-in-the-Middle Attack)**
  - Selective PTP message delay, making the slave clock over- or under-compensate its offset to the GMC
  - Timestamp modification, causing confusion to the slave's servo algorithm
- **GMC Spoofing (DOS)**
  - Pose as GMC to win the master election in BMC, or make a node with a poor clock class winning the election
- To properly protect the PTP traffic from the above threats and to comply with 3GPP TS33.320 securing the synchronization messages requirement, putting PTP message inside the IPSEC tunnel using ESP mode is the desired solution
  - Note that AH mode (RFC 2402) does NOT prevent man-in-the-middle attack since NO protection for confidentiality

IEEE1588-2008 Annex K Security Provisioning (Experimental)

- IEEE1588-2008 Annex K defines an experimental security extension to PTP and the security protocol is composed of two basic mechanisms:
  - An integrity protection mechanism through HMAC (Hash-based Message Authentication Code), which uses the Message Authentication Code (MAC) to verify that a received message was transmitted by an authenticated source, was not modified in transit, and it is fresh (i.e., not a message replay)
  - A challenge-response mechanism, which is used to affirm the authenticity of new sources and to maintain the freshness of the trust relations.
- The participants in the PTP security protocol communicate through security associations (SAs).
  - The SA is unidirectional, and it protects traffic going from the source to the destination
  - Each node maintains a table of incoming SAs, which it uses for verification of incoming traffic, and a table of outgoing SAs, which it uses for protection of outgoing traffic
  - An SA can be shared by a single sender and multiple receivers
- Confidentiality is not required for the security of IEEE 1588 messages since time information is public in the network; therefore, encryption is avoided to simplify security

IEEE1588-2008 Annex-K Issues

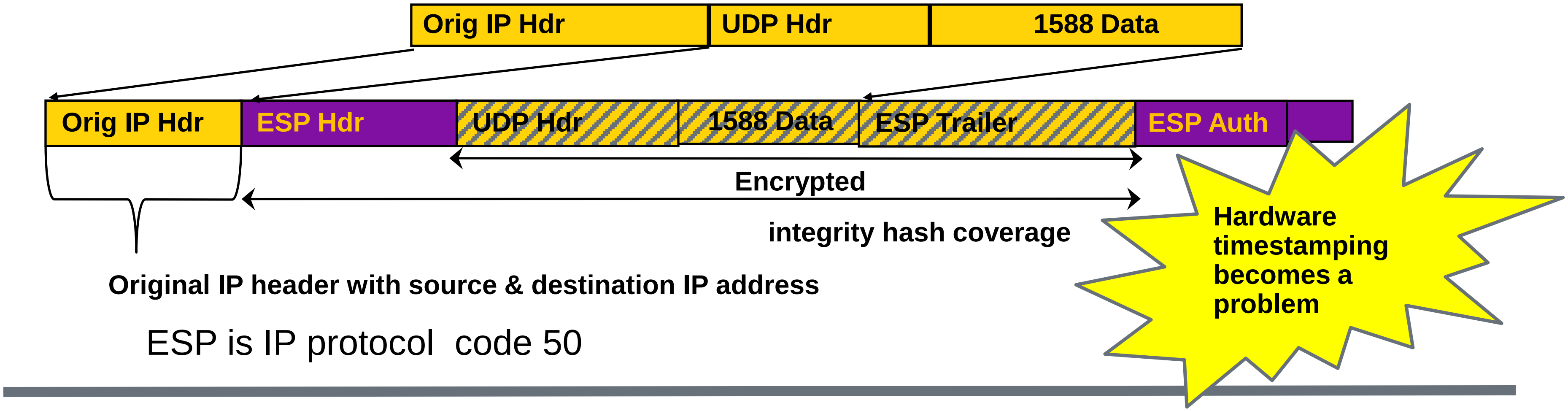
- Securing PTP messages in one-step clocks presents a challenge since MAC of Sync and Delay_Resp messages must be calculated in real-time while the frame is being sent over the GMC PTP egress port
  - Hardware implementation is almost a MUST
- The HMAC-SHA1-96 or HMAC-SHA256-128 MAC is suboptimal due to its long block size (512 bits) and the delay in MAC calculations
- The three-way handshake in the challenge-response mechanism is unnecessary and could be replaced by a one-way authentication
- There is no key distribution scheme specified (out-of-scope) that would allow nodes to join the network at any time
- HMAC is not as efficient as faster and cheaper alternatives such as GMAC (Galois Message Authentication Code in RFC4543), XCBCMAC (eXtension Cipher Block Chaining in RFC3566) etc.
- Checking with most 1588 vendors NSN has engaged, NO ONE has yet implemented the Annex-K in IEEE1588-2008
- **It is recommended to disregard the IEEE1588-2008 Annex-K security provisioning and resort to the end-to-end IPSEC tunnel in ESP mode to encrypt the 1588 packets**

PROBLEM for IEEE1588-2008 PTP inside IPSEC

- PTP inside IPSEC with AH (Authentication Header) mode does not fully prevent Man-in-the-Middle nor Denial of Service attack since the PTP payload is not encrypted and can be easily identified with a DPI engine
- PTP depends on PHY layer hardware timestamping on PTP messages to deliver microsecond (μs) phase accuracy and superb frequency synchronization performance; PTP packet within the encrypted ESP (Encapsulating Security Payload) cannot be identified by the hardware for PHY timestamping until it is decrypted at the application layer
  - One solution is to perform hardware PHY timestamping on every packets and only use the timestamp when the IPSEC packet is later identified as 1588 PTP packet (too expensive)
- Once 1588 PTP packets are inside the IPSEC tunnel, on-path support with intermediate nodes serving as BC (Boundary Clock) or TC (Transparent Clock) will no longer be possible
- IPSEC (ESP mode) encryption and decryption will incur extra latency / jitter and negatively impact the 1588 PTP packet filtering / servo algorithm to deliver superb performance

1588 PTP inside IPSec with Encapsulating Security Payload (ESP)

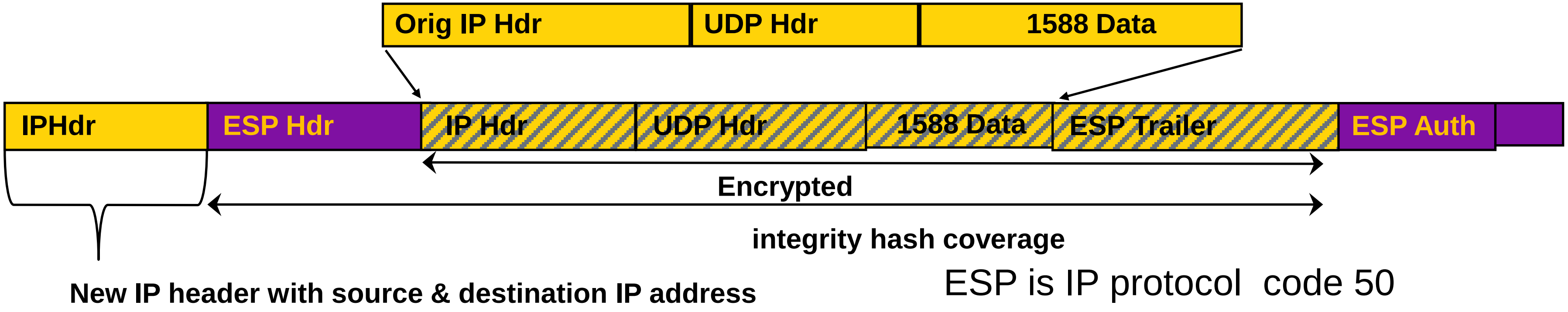
IPSEC Transport Mode



Original IP header with source & destination IP address

ESP is IP protocol code 50

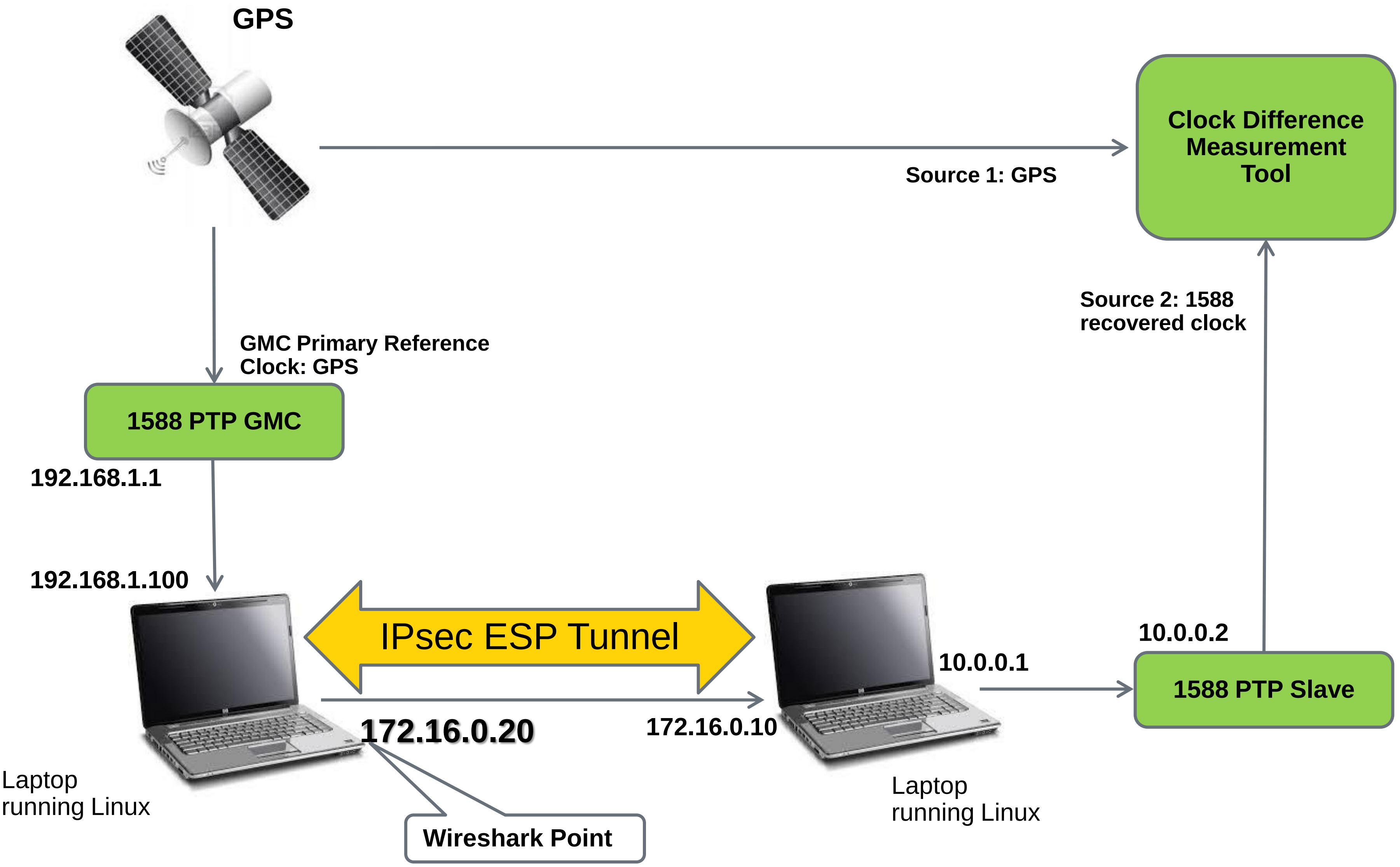
IPSEC Tunnel Mode



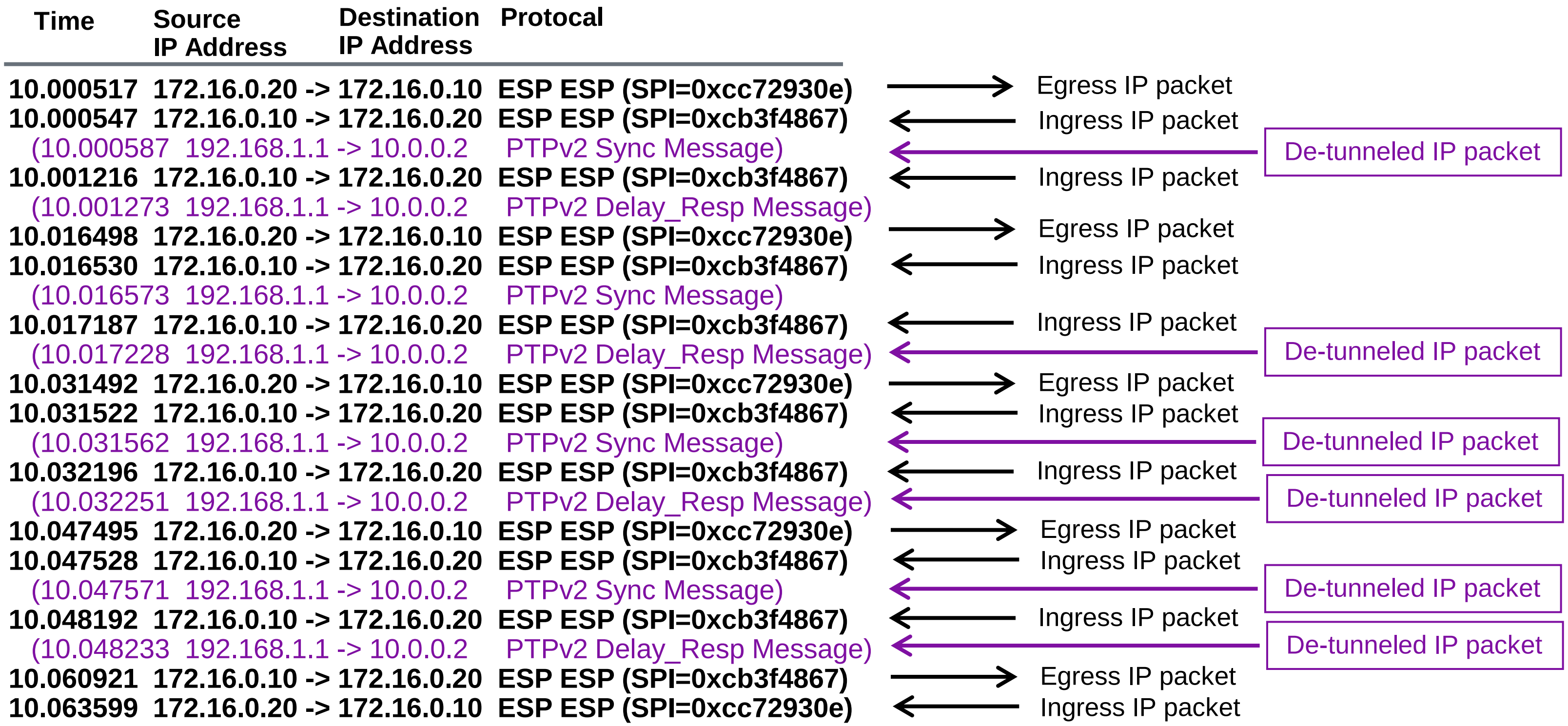
New IP header with source & destination IP address

ESP is IP protocol code 50

1588 over IPsec ESP Tunnel with Linux Laptop



An Example Wireshark Output at 172.16.0.20 Port



```
Time       Source         Destination   Protocal
           IP Address     IP Address
10.000517  172.16.0.20 -> 172.16.0.10  ESP ESP (SPI=0xcc72930e)       →  Egress IP packet
10.000547  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.000587  192.168.1.1 -> 10.0.0.2    PTPv2 Sync Message)          ←  De-tunneled IP packet
10.001216  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.001273  192.168.1.1 -> 10.0.0.2    PTPv2 Delay_Resp Message)
10.016498  172.16.0.20 -> 172.16.0.10  ESP ESP (SPI=0xcc72930e)       →  Egress IP packet
10.016530  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.016573  192.168.1.1 -> 10.0.0.2    PTPv2 Sync Message)
10.017187  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.017228  192.168.1.1 -> 10.0.0.2    PTPv2 Delay_Resp Message)    ←  De-tunneled IP packet
10.031492  172.16.0.20 -> 172.16.0.10  ESP ESP (SPI=0xcc72930e)       →  Egress IP packet
10.031522  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.031562  192.168.1.1 -> 10.0.0.2    PTPv2 Sync Message)          ←  De-tunneled IP packet
10.032196  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.032251  192.168.1.1 -> 10.0.0.2    PTPv2 Delay_Resp Message)    ←  De-tunneled IP packet
10.047495  172.16.0.20 -> 172.16.0.10  ESP ESP (SPI=0xcc72930e)       →  Egress IP packet
10.047528  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.047571  192.168.1.1 -> 10.0.0.2    PTPv2 Sync Message)          ←  De-tunneled IP packet
10.048192  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       ←  Ingress IP packet
  (10.048233  192.168.1.1 -> 10.0.0.2    PTPv2 Delay_Resp Message)    ←  De-tunneled IP packet
10.060921  172.16.0.10 -> 172.16.0.20  ESP ESP (SPI=0xcb3f4867)       →  Egress IP packet
10.063599  172.16.0.20 -> 172.16.0.10  ESP ESP (SPI=0xcc72930e)       ←  Ingress IP packet
```

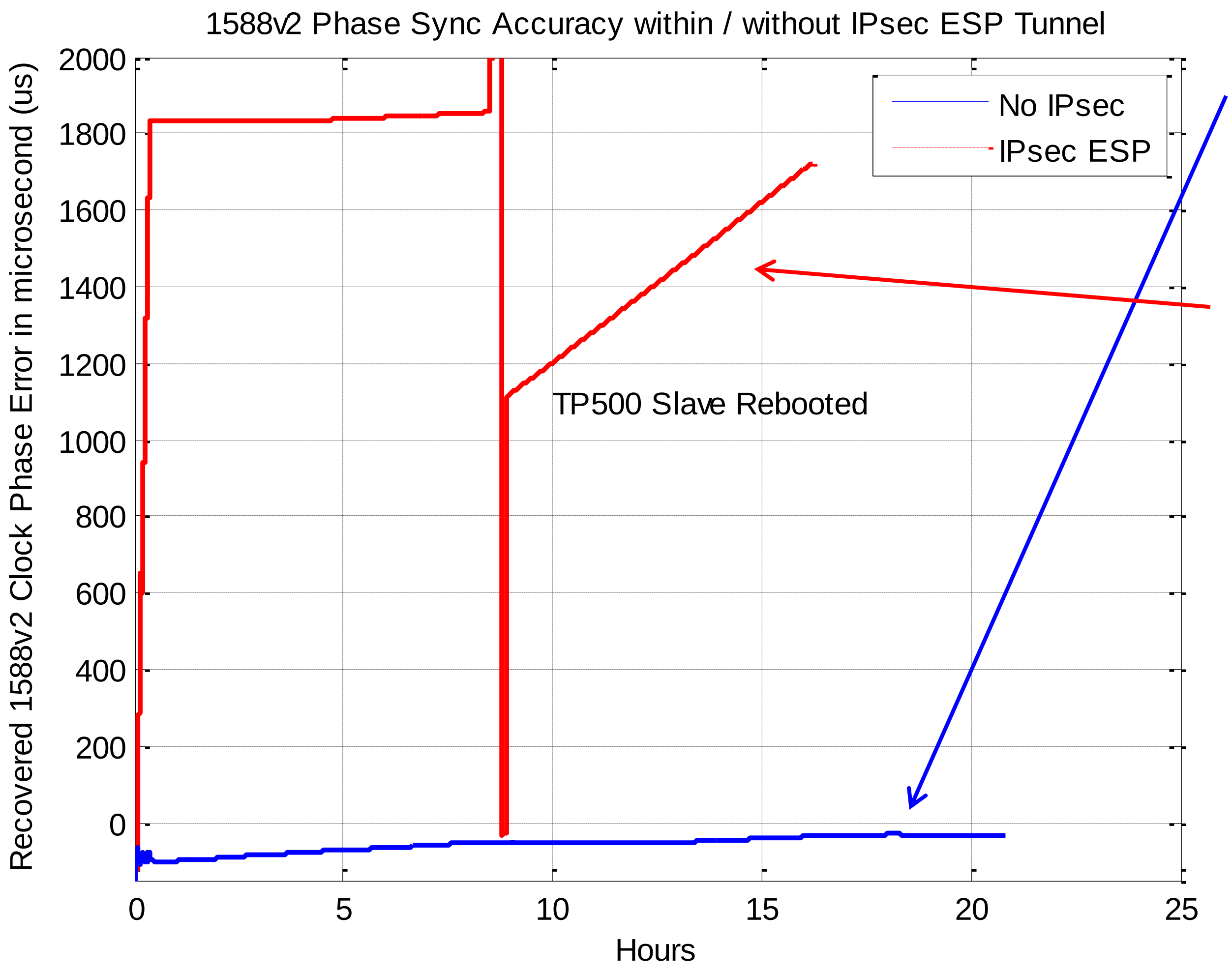
A 1588 Sync Packet Inside IPsec ESP Tunnel

Only known as an IPsec ESP packet with 154 bytes in length

Src/Dest IP addresses are for IPsec ESP end points

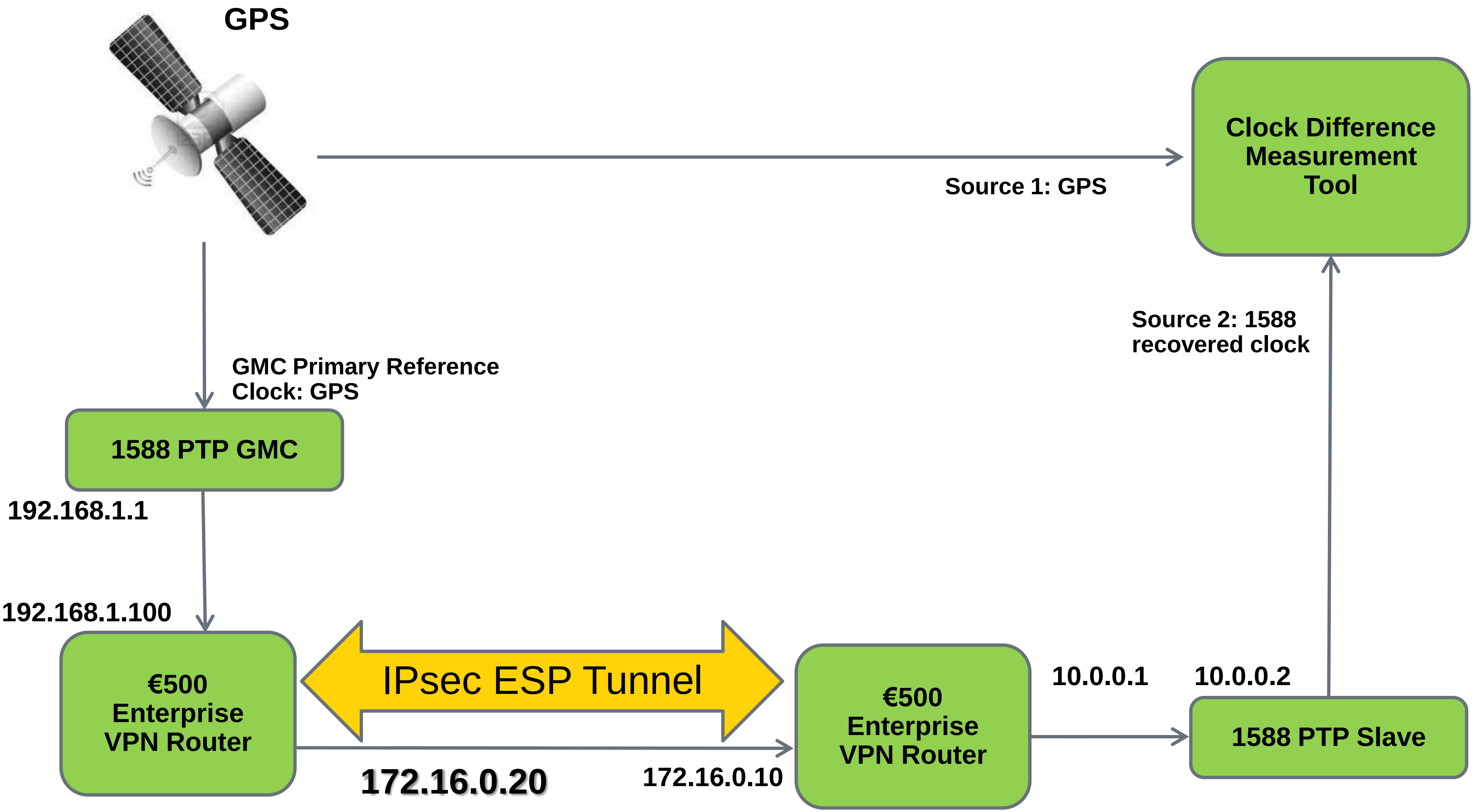
```
Frame 663 (154 bytes on wire, 154 bytes captured)
    Arrival Time: Jan  4, 2013 16:09:42.048244000
    [Time delta from previous captured frame: 0.000015000 seconds]
    [Time delta from previous displayed frame: 0.000015000 seconds]
    [Time since reference or first frame: 2.039529000 seconds]
    Frame Number: 663
    Frame Length: 154 bytes
    Capture Length: 154 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:esp]
Ethernet II, Src: Trendnet_10:58:0c (00:14:d1:10:58:0c), Dst: HewlettP_f4:22:cb (78:e7:d1:f4:22:cb)
    Destination: HewlettP_f4:22:cb (78:e7:d1:f4:22:cb)
        Address: HewlettP_f4:22:cb (78:e7:d1:f4:22:cb)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: Trendnet_10:58:0c (00:14:d1:10:58:0c)
        Address: Trendnet_10:58:0c (00:14:d1:10:58:0c)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: IP (0x0800)
Internet Protocol, Src: 172.16.0.10 (172.16.0.10), Dst: 172.16.0.20 (172.16.0.20)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 140
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
        0.. = Reserved bit: Not Set
        .1. = Don't fragment: Set
        ..0 = More fragments: Not Set
    Fragment offset: 0
    Time to live: 64
    Protocol: ESP (0x32)
    Header checksum: 0xe201 [correct]
        [Good: True]
        [Bad : False]
    Source: 172.16.0.10 (172.16.0.10)
    Destination: 172.16.0.20 (172.16.0.20)
Encapsulating Security Payload
    ESP SPI: 0xc4e99c13
    ESP Sequence: 270933
```

1588 Phase Accuracy over IPsec ESP Tunnel (Laptop Linux SW Routing & IPsec)

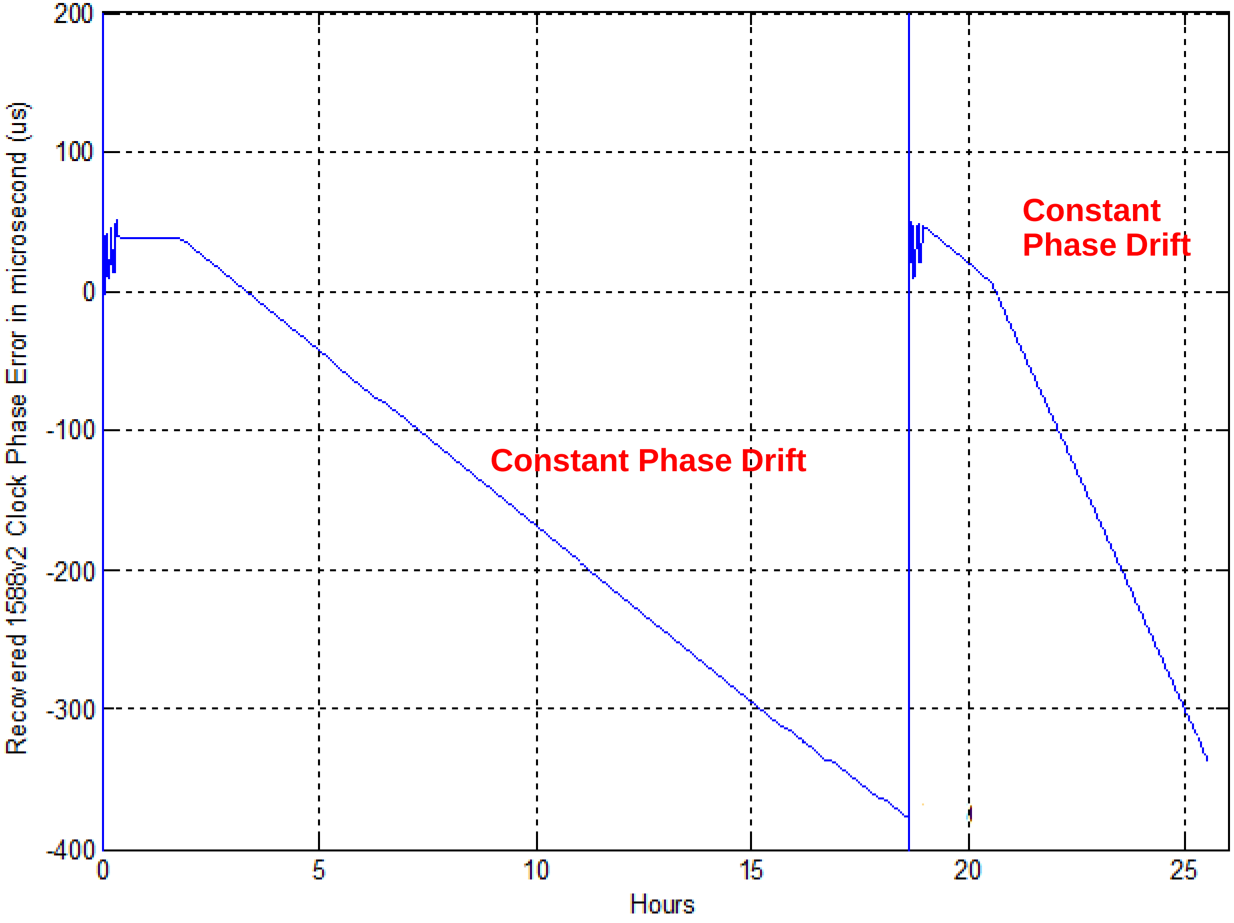


- 1588 outside IPsec ESP tunnel at least is converging to a stable phase sync accuracy, even with SW routing
- Putting 1588 inside IPsec ESP tunnel causes huge phase error (from µs to ms range) and PTP slave not able to converge

1588 over IPsec ESP Tunnel with an Enterprise VPN Router


GPS
Source 1: GPS
Clock Difference Measurement Tool
Source 2: 1588 recovered clock
GMC Primary Reference Clock: GPS
1588 PTP GMC
192.168.1.1
192.168.1.100
€500 Enterprise VPN Router
IPsec ESP Tunnel
€500 Enterprise VPN Router
10.0.0.1
10.0.0.2
1588 PTP Slave
172.16.0.20
172.16.0.10

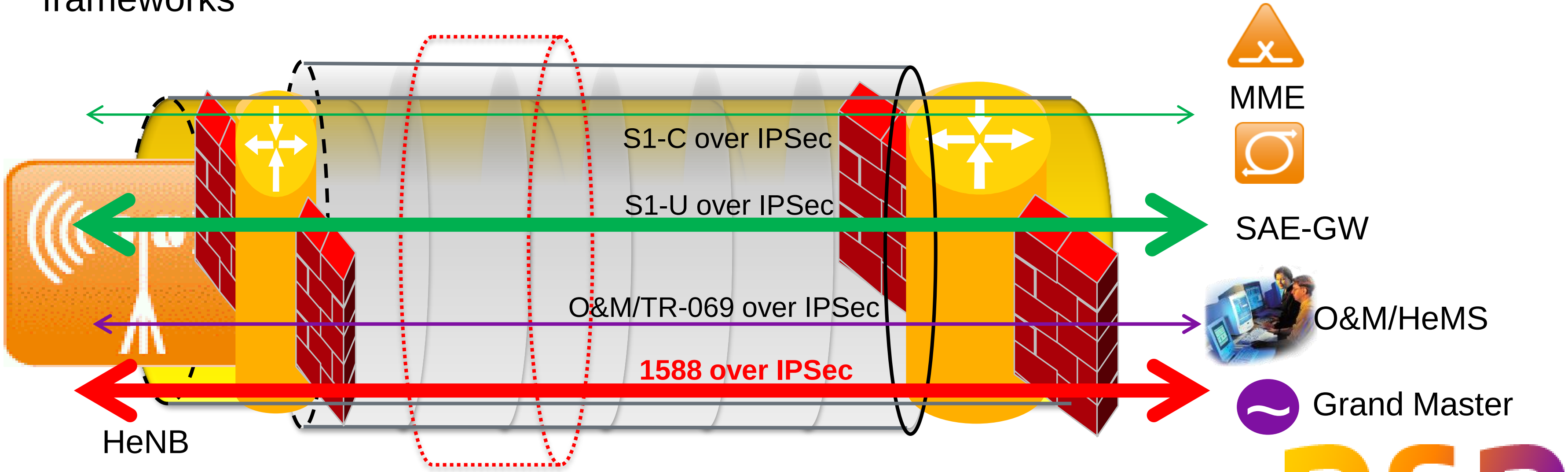
1588v2 Phase Accuracy over IPsec ESP Tunnel over a €500 Enterprise VPN Router



- ❑ The enterprise VPN router comes with a 400 MHz Cavium CPU and a L2-switch handling all traffic
- ❑ All forwarding plane functionalities are handled in the CPU on a separate process
- ❑ As the test result shows, getting reliable clocking for 1588v2 is very problematic, even before IPsec ESP tunnel
- ❑ No need to run the 1588v2 phase accuracy test over IPsec ESP

Summary

- We have analyzed 1588 PTP packets inside & outside the IPsec ESP tunnel and shown how the ESP can properly protect PTP traffic from known security threats
- We have performed lab validation on 1588 inside IPsec ESP tunnel to understand its performance impact to recovered clock phase accuracy:
  - Key finding ➔ Simply putting 1588 packets inside IPsec ESP tunnel will be problematic without dedicated packet forwarding engine and security accelerator engine on the VPN router endpoints
- Innovative Solutions are needed to secure 1588 inside IPsec ESP tunnel:
  - Solution needs to identify 1588 inside IPsec ESP Tunnel for HW timestamping instead of all packets at the ingress point
  - Solution needs to be completely compatible with existing IEEE 1588-2008 and IPSec frameworks


MME
S1-C over IPSec
S1-U over IPSec
SAE-GW
O&M/TR-069 over IPSec
O&M/HeMS
1588 over IPSec
Grand Master
HeNB