OTN Support Update
P802.3bs 400 Gb/s Ethernet Task Force

Steve Trowbridge

Alcatel-Lucent

Key Elements of OTN Support

- See "OTN Support: What is it and why is it important?", July 2013
  - A new rate of Ethernet (e.g., 400 Gb/s) fits into the corresponding rate OTN transport signal
  - All Ethernet PHYs of a given rate are mapped the same way and can be interconnected over the OTN (e.g., same PCS for all 100 Gb/s PHYs gives a single canonical format ("characteristic information" in ITU-T terminology) that can be mapped
  - Optical modules for Ethernet can be reused for OTN IrDI/client interfaces at the corresponding rate

A new rate of Ethernet (e.g., 400 Gb/s) fits into the corresponding rate OTN transport signal

- Assumption – the OTN mapper/demapper will terminate and regenerate any Ethernet FEC code, correcting errors at the OTN ingress since the FEC is chosen to correct single-link errors but not double-link errors
- Assumption – the OTN mapper/demapper may trans-decode/trans-encode back to 64B/66B to avoid MTTFPA reduction for OTN transported signal
- Based on these assumptions, the encoded data rate of the OTN-mapped 400 Gb/s Ethernet would be no more than 400 Gb/s x 66 / 64 = 412.5 Gb/s ±100ppm. Since the 400 Gb/s OTN container would presumably be designed to also transport four "lower order" ODU4s, there should be no concern that it is large enough to carry 400 Gb/s Ethernet based on the assumption that the canonical form is near this rate.
- Any Ethernet bits in excess of this rate are likely to be part of a FEC that is not carried over OTN

Discussion in Architecture and Logic ad hoc groups of 400GbE transport over OTN

- ITU-T mapped 100GbE over ODU4 as a set of deskewed and serialized PCS lanes, which included the PCS lane BIP in the alignment markers – a format common to every 100GbE PMD
- Good consensus so far that there should be an “OTN reference point” that identifies the exact information expected to be transported over OTN
- Early architecture discussion is that we shouldn’t require a common logical lane architecture across all 400GbE PMDs. A consequence if we still put BIP in the lane alignment markers is that it would be segment by segment instead of end-to-end (see further slides). If we need BIP (or any other overhead, e.g., a management channel) to be end-to-end, we will have to find somewhere besides the lane alignment markers to put it

BIP Considerations

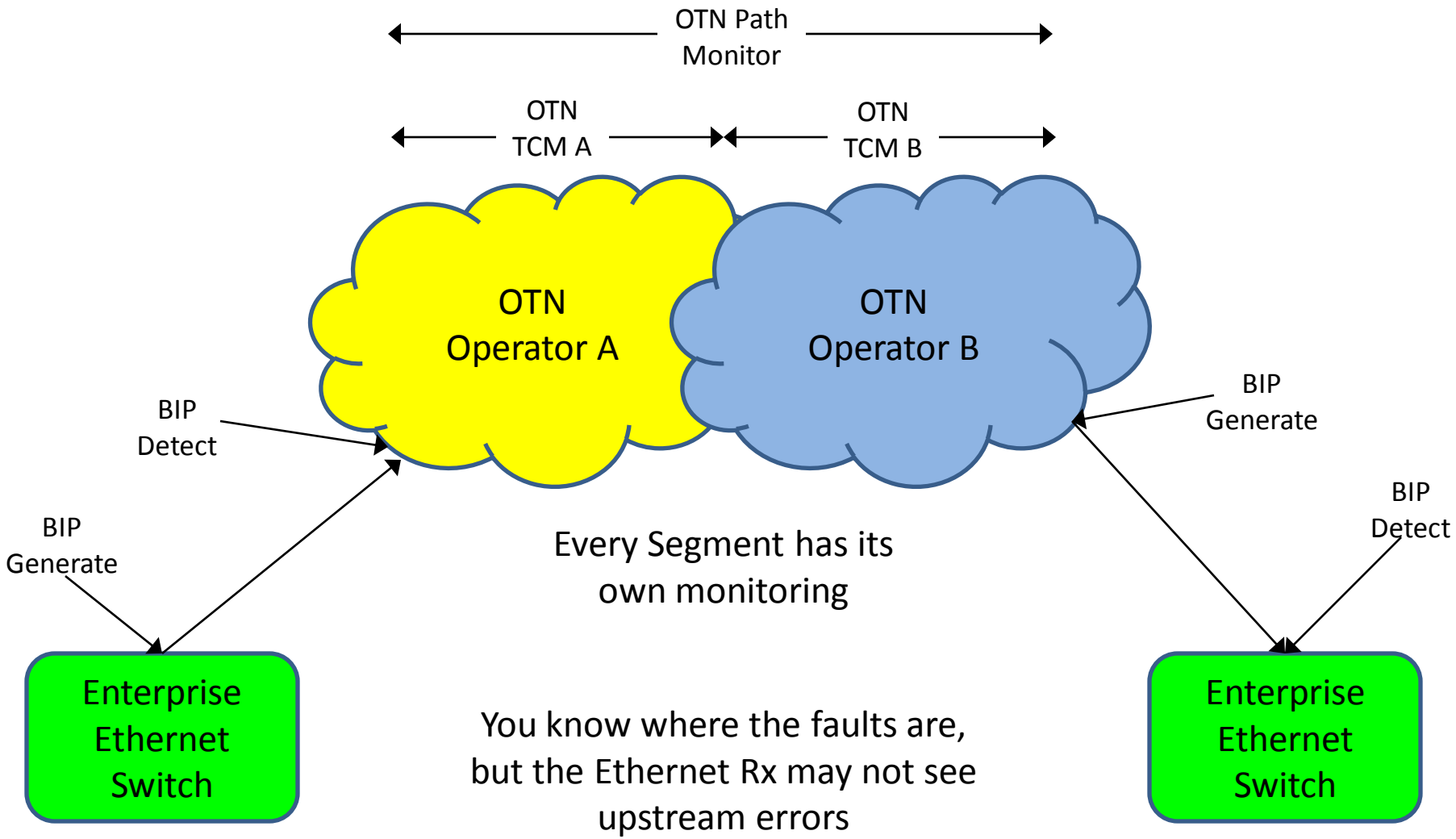
- P802.3ba introduced a BIP to detect link degradation below the level that would enter the high BER state
- Since all P802.3ba interfaces use the same logical lane striping, this allows the BIP to be carried in the lane alignment marker
- Two key use cases to be considered:
  - Service Assurance is best supported by an end-to-end BIP. You can see whether the connection is experiencing errors at the Rx, but in a multi-segment link, it is not simple to determine where the errors were introduced
  - Fault Isolation is best supported by a segment-by-segment BIP. You can detect whether errors are introduced in a particular segment just adjacent to that segment, but you don’t have a consolidated end-to-end view at the Ethernet Rx

BIP Considerations - continued

- Transport networks (e.g., OTN) tend to support multiple layers of error monitoring:
  - Path monitoring checks the connection end-to-end
  - Section monitoring checks an individual link segment
  - One or more layers of tandem connection monitoring can monitor between pre-determined points along the path, e.g., if the link traverses multiple operator networks, the section across each operator domain can be monitored independently; or monitoring the scope of a protection domain for working and protection paths
- Unlikely that Ethernet would decide to support multiple layers of monitoring as most Ethernet links are a single segment.

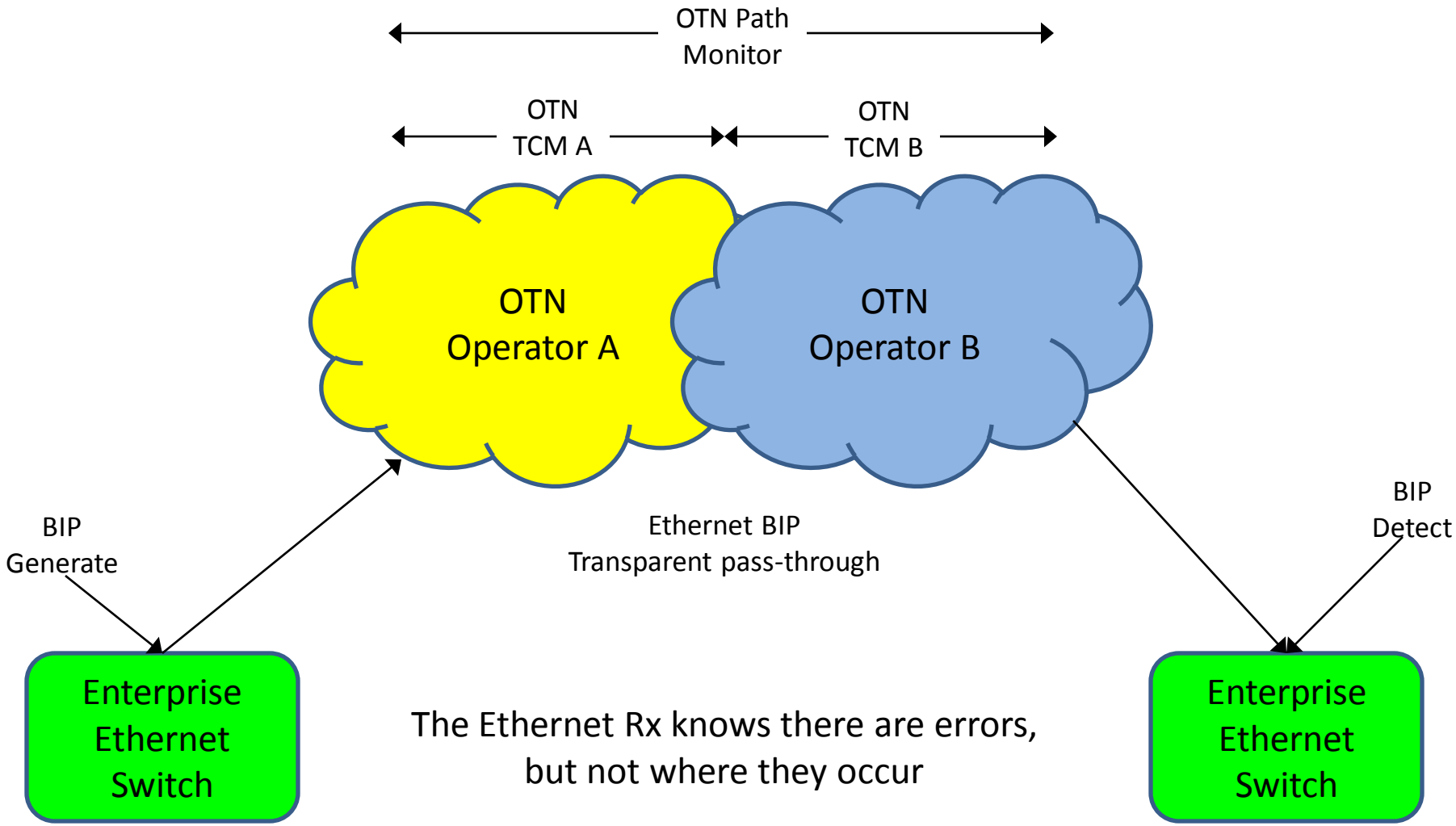
Fault Isolation Scenario

Per segment BIP (one direction of transmission)



Service Assurance Scenario

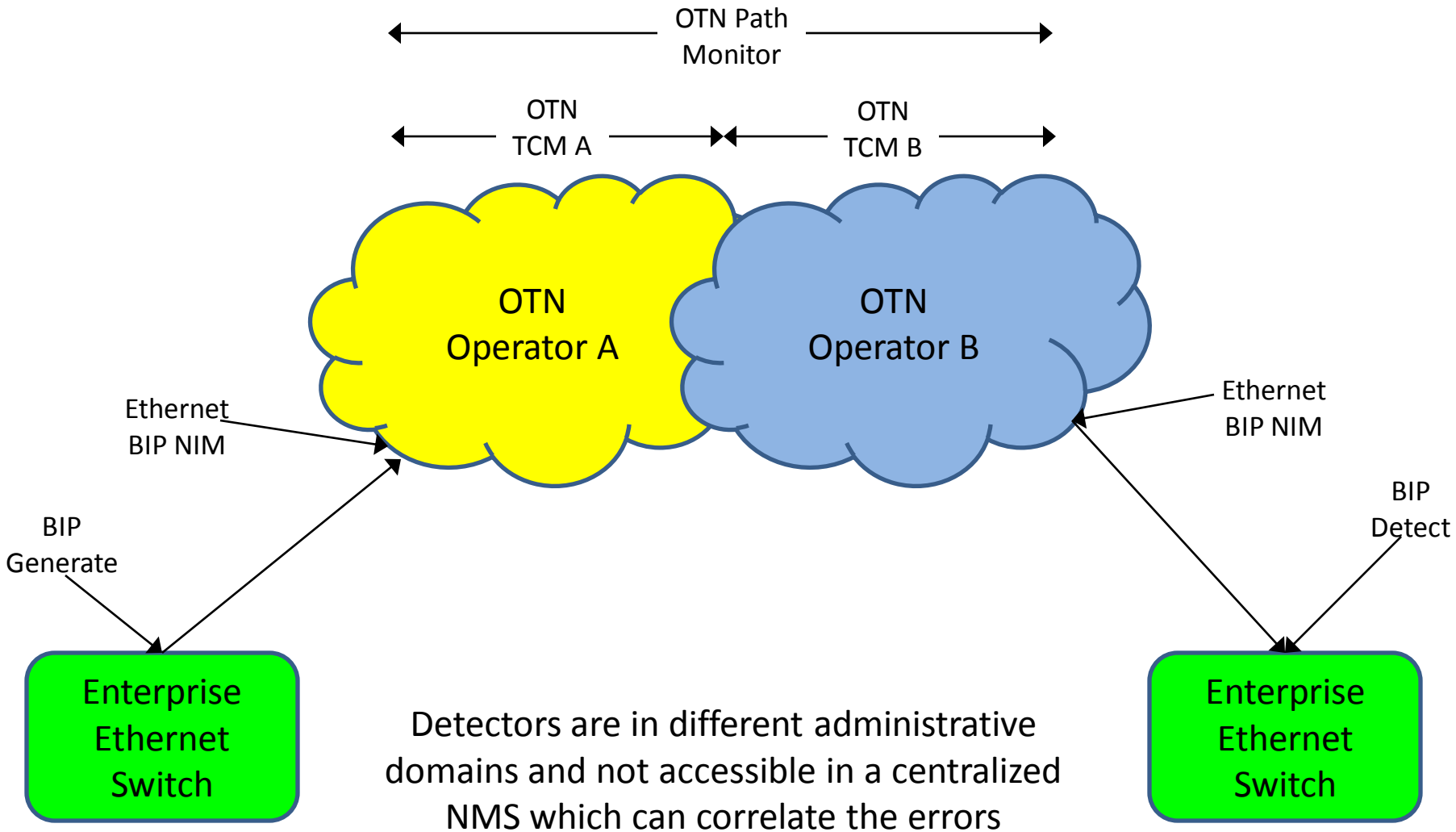
End-to-end BIP (one direction of transmission)



The Ethernet Rx knows there are errors,
but not where they occur

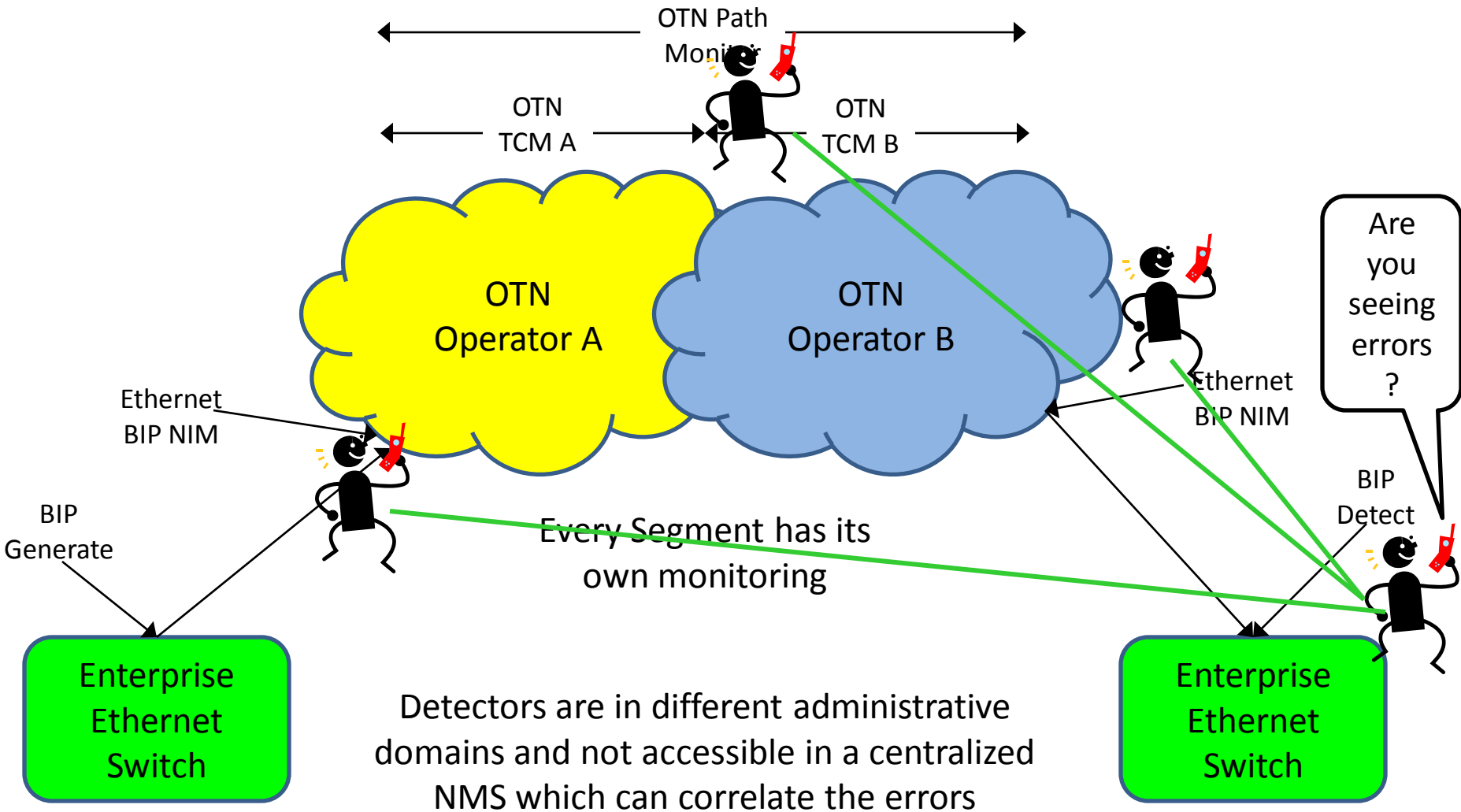
Fault Isolation Scenario

Using end-to-end BIP and non-intrusive monitoring (NIM) (one direction of transmission)



Fault Isolation Scenario

Using end-to-end BIP and non-intrusive monitoring (NIM) (one direction of transmission)



BIP Considerations - continued

- The scenario of the last slide is adequate for a persistent level of errors, but what about error bursts?
- Telco equipment historically collects error count information and stores in 15-minute and 24-hour bins. Will Ethernet equipment do such a thing?
- So you come in on Monday morning and see that you had an error burst over the weekend, ring up those with access to the other monitoring points and ask whether they observed errors between 3:15pm-3:30pm on Saturday afternoon

BIP Recommendation

- From discussion so far, the primary use case seems to be fault isolation rather than service assurance
- Segment by segment BIP provides the simplest fault isolation
- End-to-end BIP provides service assurance, and CAN do fault isolation at a coarse level by comparing error counts seen across what are likely different administrative domains
- While doing end-to-end BIP can meet both the needs of service assurance and fault isolation, supporting the secondary use case makes the primary use case enormously more cumbersome. Therefore, if we do BIP, it should be segment-by-segment. This also would allow BIP to remain in the alignment markers even if different PMDs are striped differently
- Another idea (credit to Dave Ofelt) is that if every PMD has FEC, do we even need BIP since you will get better information from the FEC corrected/uncorrectable error counts. This would also be segment-by-segment if not every PMD uses the same FEC.

Module Reuse

- Module reuse was facilitated by the fact that nothing below a CAUI chip-to-module interface cared about the or manipulated the bit values on the lanes – as long as OTN was striped into the same number of logical lanes as Ethernet, everything would work
- The following likely can be preserved: no idle insertion/deletion occurs below a CDAUI chip-to-module interface
- The following are possibly not be precluded by the 400GbE architecture:
  - Logical to physical lane multiplexing in a module may be on a block or FEC symbol basis rather than a bit basis
  - One (possibly Ethernet Frame Format dependent) FEC code may be replaced with another)

Options for Module Reuse

- Option 1: Preserve the 802.3ba rule that no sublayers below the MAC care about bit values or manipulate the bit values on logical lanes (bit multiplexing only). Any FEC is done on the host board above a CDAUI. OTN may use a different FEC than Ethernet if it needs a stronger FEC to compensate for the higher bit-rate
- Option 2: Every FEC is client independent and not locked to the Ethernet (e.g., 66B block or alignment marker position) or OTN FAS position. Use the same FEC for Ethernet and OTN. A module can remove one FEC and apply another, and can combine logical into physical lanes, but cannot restripe the signal in a different way since OTN will stripe its signal into the same number of logical lanes
- Option 3 (most general, described in Norfolk) encode the OTN frame as 66B blocks (all data) and use whatever striping and FEC encoding mechanisms are used for Ethernet. OTN and Ethernet use the same FEC

Option #2: OTN Bit-rates using this scheme

	Working Assumption Bit-Rate
OTUC4 bit-rate without FEC	422.904 Gb/s
Add FEC, e.g. RS(528,514)	434.423 Gb/s
Logical Lane Rate (well within CEI-28G)	27.151 Gb/s
Ethernet Nominal Bit-rate	412.5 Gb/s
400G OTN Increase in bit-rate	5.31 %

100G OTN Increase in bit-rate	8.42 %

Smaller increase for 400G than for 100G, mainly due to RS(528,514) FEC rather than RS(255,239) FEC

Option#3 Amplification

- PCS is a logical serial stream of 66B blocks. Only physical instantiations are striped over physical or logical lanes
- Maintain the principle, as in 802.3ba, that idle insertion/deletion is not done below the PCS.
- Since any physical instantiation will need to be striped with lane markers, do idle insert/delete at the PCS only so the logical stream will be at the
  nominal MAC rate x 66/64 x (1-1/16384)
  so that any physical instantiation has room to insert lane markers as needed without idle insert/delete

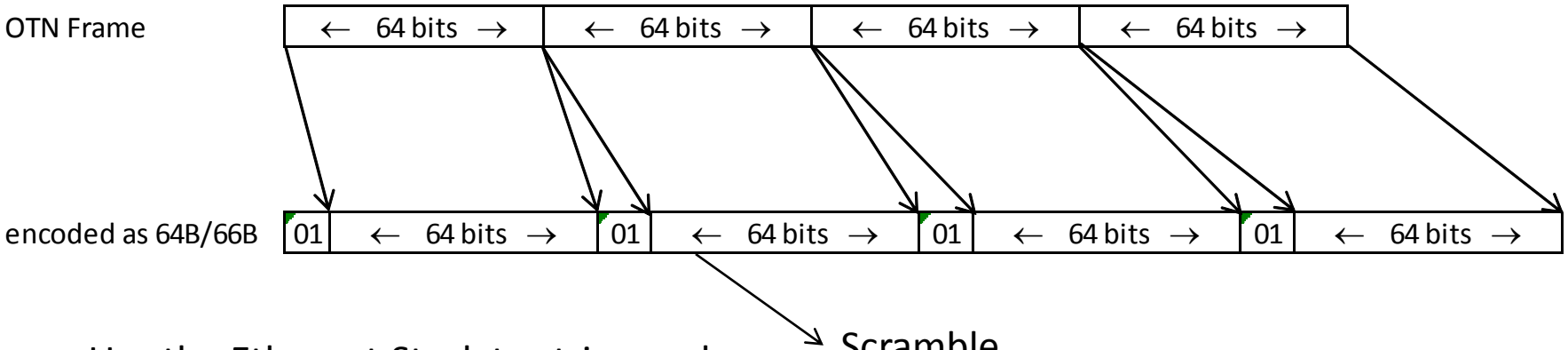
Option #3 Amplification continued

- Example physical instantiation could be exactly the format of Idea #1, produced by transcoding 64B/66B to 256B/257B, striping first into 100G groups, striping within each 100G group into 4 logical lanes on 10-bit symbol boundaries, inserting alignment markers on each lane, and applying an RS(528,514) code based on 10-bit symbols with alignment markers appearing in the first of each of 4096 Reed Solomon code blocks

Option#3 Implications for OTN

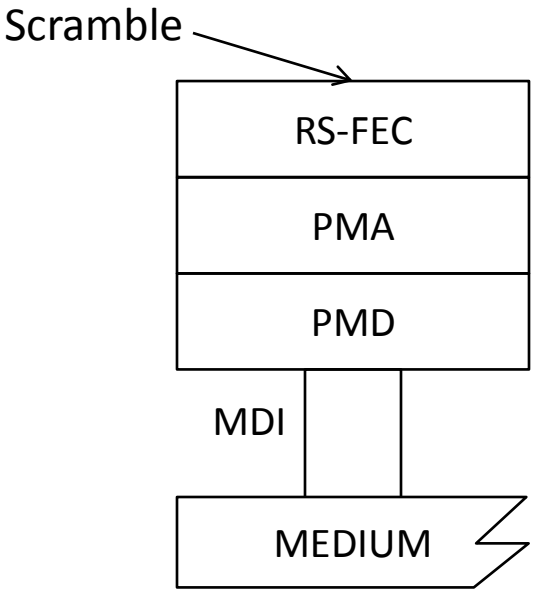
- Likely only possible if the same FEC code can be used for OTN applications as for Ethernet applications at about 6% higher bit-rate
- Would need to make OTN look like 66B blocks. Easiest way to do this and not lose any information in transcoding is to insert a “01” sync header after every 64 bits (all data)
- Since this is just part of the logical frame format, this doesn’t waste as many bits as it appears. 8 sync header bits are added to every 256 data bits in the “logical” frame format, but 7 of those bits are immediately recovered in 256B/257B transcoding and reused for the FEC code. So 0.39% net is added to the OTN frame to make it look like 66B blocks, then 2.724% overhead RS FEC added

Option #3 - Illustration of turning OTN frame into 64B/66B blocks



Use the Ethernet Stack to stripe and FEC encode the OTN frame when carrying over an Ethernet Module for an OTN IrDI or client interface

Could be OTN frame aligned as an OTUC4 frame without FEC is exactly 7648×64 bits, but not essential with scrambling



Option #3: OTN Bit-rates using this scheme

	Working Assumption Bit-Rate
OTUC4 bit-rate without FEC	422.904 Gb/s
64B/66B encoded	436.120 Gb/s
256B/257B transcoded	424.556 Gb/s
Insert Lane Markers	424.582 Gb/s
Add RS(528,514) FEC	436.146 Gb/s
Logical Lane Rate (well within CEI-28G)	27.259 Gb/s
Ethernet Nominal Bit-rate	412.5 Gb/s
400G OTN Increase in bit-rate	5.73 %

100G OTN Increase in bit-rate	8.42 %

Smaller increase for 400G than for 100G, mainly due to RS(528,514) FEC rather than RS(255,239) FEC

Option #3 - The module reuse aspect of OTN Support is satisfied if the following are true:

- There is an Ethernet sublayer reference point such as the PCS that is logically a serial stream of 64B/66B blocks
- No idle insertion/deletion occurs below the PCS (the serial stream of 64B/66B blocks), and hence the rest of the stack can deal with a constant-bit-rate (CBR) bitstream that is effectively an infinite-length packet.
- Note that any logical to physical lane interleaving that works for Ethernet also works for OTN since they are encoded the same way
- The link parameters and FEC coding gain have sufficient margin to meet the error performance target when running at approximately 5.73% higher bit-rate than necessary for 400G Ethernet. More likely to be true if all P802.3bs interfaces have FEC.

Option #3 - Details

Enabling 400 Gb/s Ethernet Module reuse for 400 Gb/s OTN IrDI/client interfaces

- No Idle Insertion/Deletion below the PCS
- The OTN frame is adapted to a CDAUI-like format by encoding as 64B/66B data blocks and using the Ethernet LLS, RS-FEC, and FLS sub-layers at 5.73% higher bit-rate
- All 400 Gb/s Ethernet PHYs are assumed to use FEC
- The FEC code is selected to have sufficient margin to meet the error performance target at 5.73% above the Ethernet bit-rate

Module Reuse Recommendations

- It seems unlikely that the architecture will preclude a module that may apply a different FEC or restripe the signal for an initial or subsequent generation PMD, so Option 1 likely does not cover all cases
- Option 2 brings the OTN and Ethernet bit rates slightly closer together than Option 3, but at the expense of precluding any restriping in the module and additional constraints on the choice of FEC
- Option 3 is the most general, and the OTN bitrate difference over Ethernet of 5.73% doesn't seem to be enough greater than the Option 2 difference of 5.31% to justify the additional restrictions
- Recommend Option 3 as a module reuse architecture

THANKS!