

Analysis for FlexE control model & A YANG Data Model for Flex Ethernet(FlexE)

[draft-xiaobn-ccamp-flex-e-yang-mod-00](#)

[draft-wang-ccamp-flex-e-control-analysis-00](#)

Xiaobing Niu (ZTE)

Qilei Wang (ZTE)

Introduction

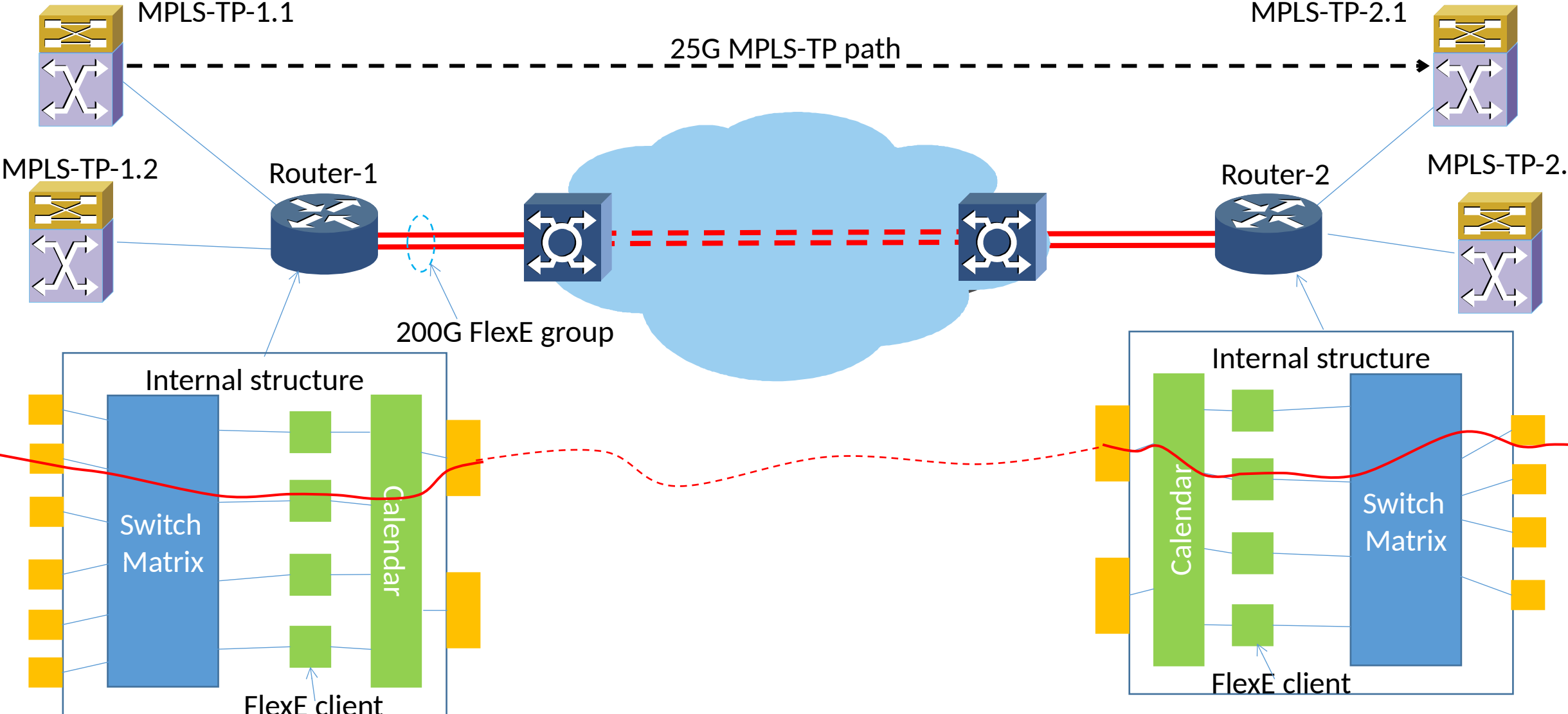
- This presentation try to derive what can we do with control method to FlexE according to the detailed encapsulation process described in [ITU-T G.8023].
- draft-xiaobn-flexe-yang-mod-00 was presented during last IETF meeting in Bangkok. Now replace it with draft-xiaobn-ccamp-flexe-yang-mod-00. Updates mainly include:
 - reference [ITU-T G.8023], [ITU-T G.709] and [ITU-T G.798].
 - restructure the model into two parts: “ietf-flexe-types” and “ietf-flexe-configuration”.
- A new draft-wang-ccamp-flexe-control-analysis-00, which include feature analysis of the FlexE client, FlexE group and adaptation between them, was submitted to help understand the model draft.

What can we do to FlexE?

- FlexE is just an **interface technology**. What we believe is: **only some initial configuration is needed** for this interface (i.e., FlexE group and FlexE client). The FlexE interface enabled can be used in the same way as traditional Ethernet with no dynamic protocol involved any more.
- Initial configuration below should be provided first:
 - FlexE group configuration
 - FlexE client configuration

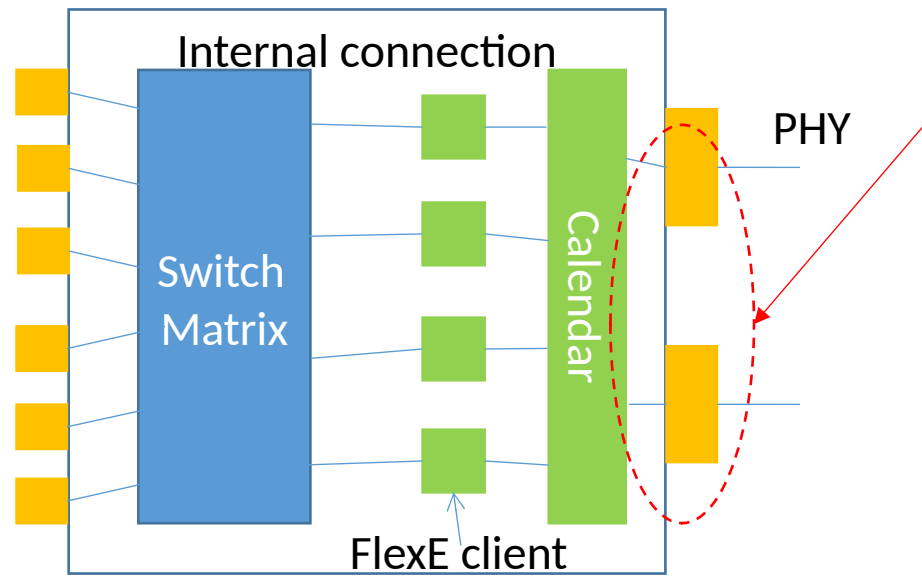
Emphasis

Network Scenario Example



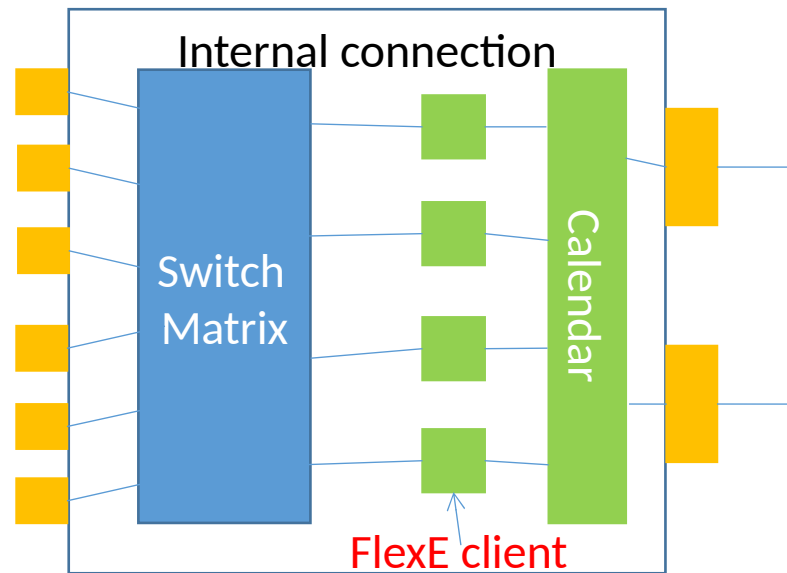
Interface configuration -- FlexE group

- Bonded several PHYs together, each PHY is consisted of 66B blocks.
- Insert section monitoring overhead at FlexE source to determine the status of the link between two FlexE shims. [ITU-T G.8023]



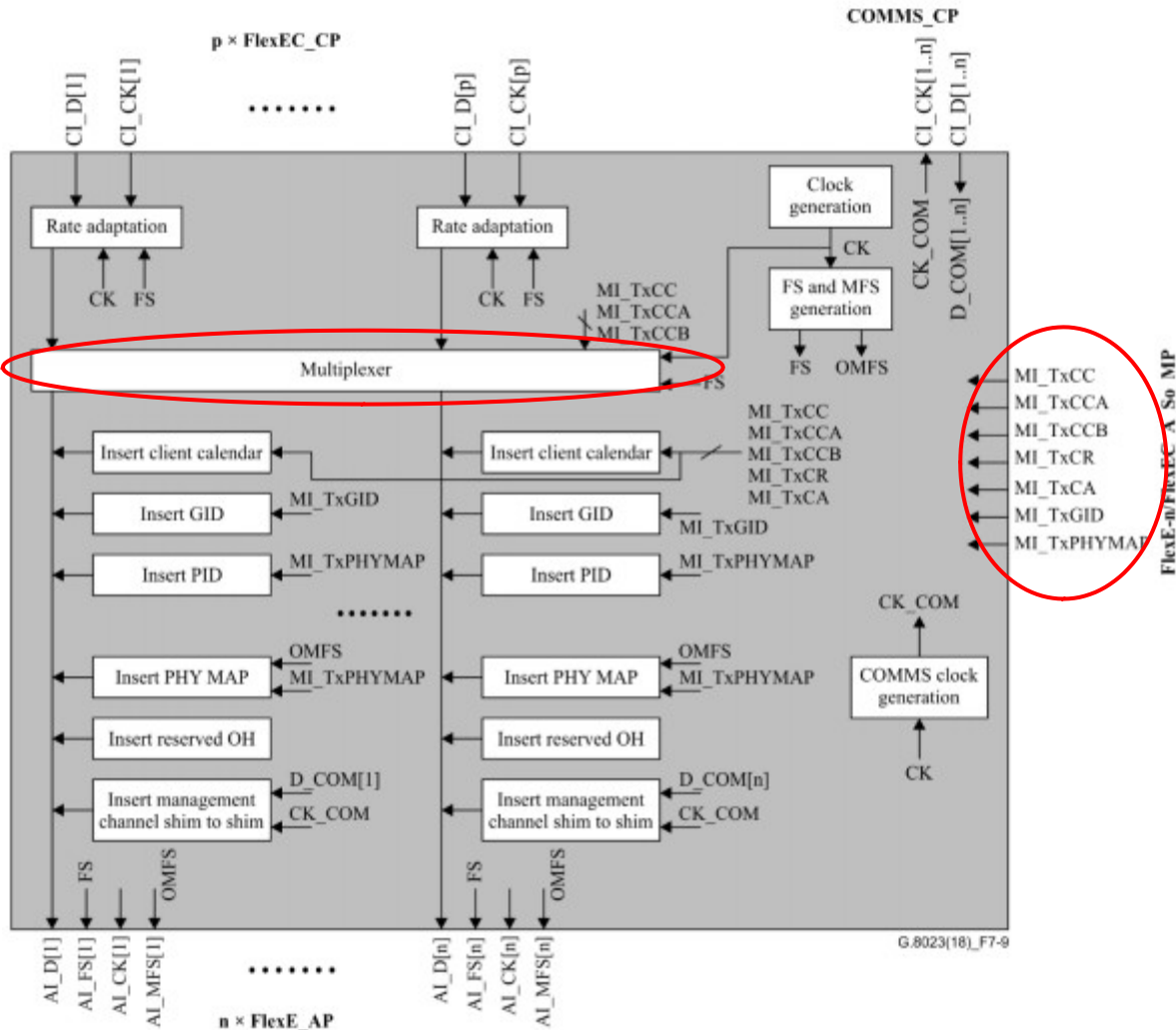
Interface configuration -- FlexE client

- A FlexE Client is an Ethernet flow based on a MAC data rate.
- According to the description in [ITU-T G.8023], there is no overhead defined for monitoring a FlexE client, so **the trail concept for FlexE client does not exist. FlexE client TTP is a null function.** Modelling FlexE client as a network layer is not correct.



Internal port performing encapsulation function for upper layer packets (e.g., MPLS packets)

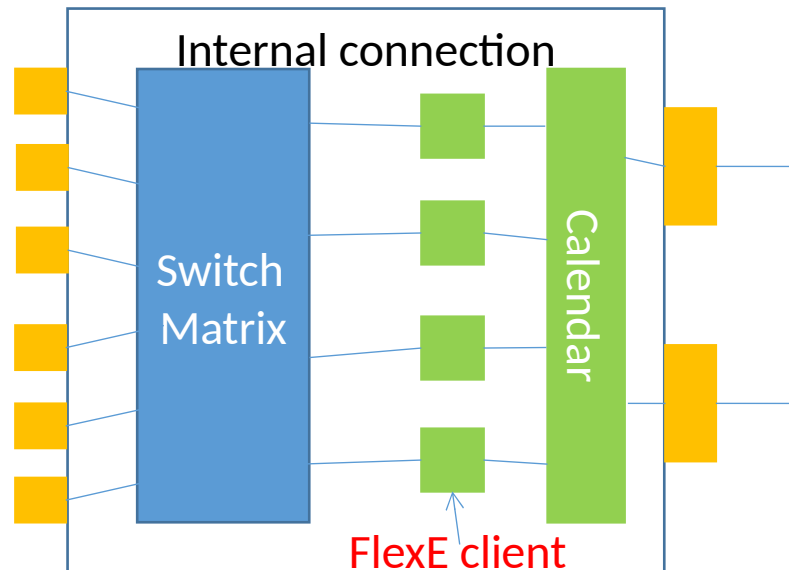
Adaptation between FlexE client and group



- Figure 7-9 from [ITU-T G.8023] about adaptation process at source node.
- External inputs include:
 - TxCC, TxCCA and TxCCB
Used to configure FlexE client number and type A or type B calendar configuration.
 - TxGID - group number
 - TxPHYMAP - PHY ID and MAP
- Multiplexer function performs the action of assigning individual FlexE Clients to specific calendar slots of the FlexE group.
 - No slot switching
 - Slot allocation information is insert into the overhead through the "insert client calendar" function.
 - No need to intervene the slot allocation process with external tools (e.g., NMS/Controller)

Transformation of MAC into FlexE client

- According to Figure 8-6 of [ITU-T G.8023] and [IEEE 802.3], It can be seen that the process mainly includes the 64B/66B encoding, as well as MAC frame check sequence generation and frame counting. The FlexE client stream is generated at the determined FlexE Client MAC rate and 64B/66B encoded.
- Behavior performed is finished in FlexE client internal port automatically. No need to intervene this process with external tools (e.g., NMS/controller)



FlexE configuration model

```
module: ietf-flexe-yang
+--rw flexe-configuration
+--rw flexe-groups
| +--rw flexe-group* [group-number]
|   +--rw group-number          uint32
|   +--rw group-attributes
|     +--rw flexe-group-available-bandwidth?  rt-types:bandwidth-ieee-float32
|     +--rw calendar-slot-granularity?        fe-types:calendar-slot-granularity
|     +--rw flexe-phy-type?                   fe-types:flexe-phy-type
|     +--rw bonded-phys
|       +--rw flexe-phys* [phy-number-in-group]
|         +--rw phy-number-in-group          uint8
|         +--rw local-phy-interface?         if:interface-ref
|         +--rw remote-phy-interface?       if:interface-ref
|     +--rw flexe-instances
|       +--rw flexe-instance* [flexe-instance-number]
|         +--rw flexe-instance-number        uint8
|         +--rw unavailable-sub-calendar-slot-list* [sub-calendar-slot-id]
|           +--rw sub-calendar-slot-id      uint8
|         +--rw unequipped-flexe-instance* [flexe-instance-number]
|           +--rw flexe-instance-number     uint8
|     +--rw tx-calendar?                   fe-types:calendar-AorB
|     +--rw rx-calendar?                   fe-types:calendar-AorB
|     +--rw tx-calendar-neg?               enumeration
|     +--rw reply-ca-mode?                 enumeration
+--rw flexe-clients
+--rw flexe-client* [client-number]
+--rw client-number          uint16
+--rw bandwidth
| +--rw signal-type?        flexe-client-signal-rate
| +--rw mac-rate?           rt-types:bandwidth-ieee-float32
+--rw client-interface?     if:interface-ref
+--rw group-number?         uint32
```

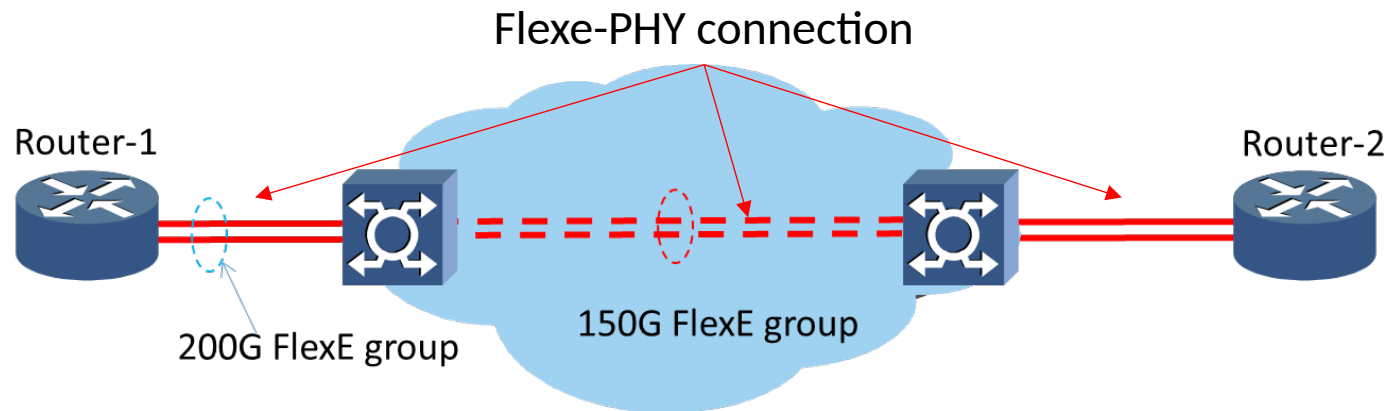
Information for configuring FlexE group include the group number, PHY information, mapping between logical PHY and physical interface, and flexe-instances;

One thing that we are not sure is whether flexe-instances should be included in the model, as for 200G or 400G PHY, the way of distributing instance number is fixed, even for unequipped case.

Initial configuration for FlexE client include: client number, mapping client number to a specific internal client interface, association of one FlexE client to a FlexE group.

Aware case configuration

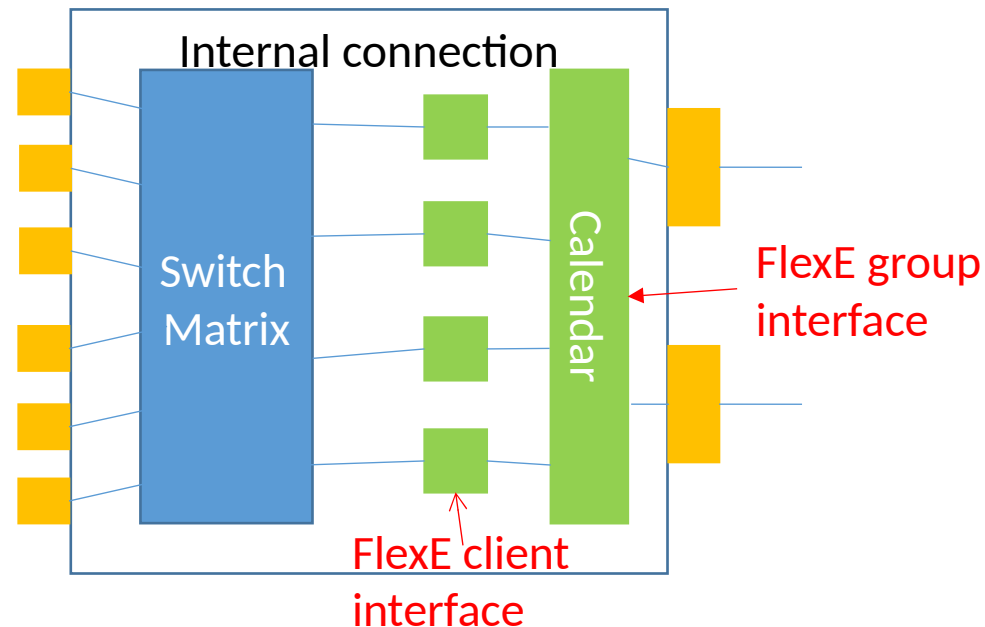
- One thing that we forget to put in the model.....
- Crunch of flexe group signal is actually done at each PHY level. The end-to-end PHY connection should be constructed first before configuring group.
- [RFC6002] defines switching type DCSC to describe the switching of whole digital channel on a single channel interface (ETY is in this scope). A new DCSC-flexe Switch type is suggested.
- As the unavailable slots are placed at the end of each relevant sub-calendar (the highest numbered slots), bandwidth information of each hop of PHYs can be used to infer the unavailable slot and maybe unequipped flexe instance information . Bandwidth of each hop should be included when constructing end-to-end flexe-phy.



```
augment "/te:te/te:globals/te:named-path-constraints/"
+ "te:named-path-constraint/te:explicit-route-objects/"
+ "te:route-object-include-exclude/te:type/te:label/"
+ "te:label-hop " {
description "bandwidth of each hop.";
leaf bandwidth {
type te-types:te-bandwidth;
config 'false';
description
"TBD";
}
}
```

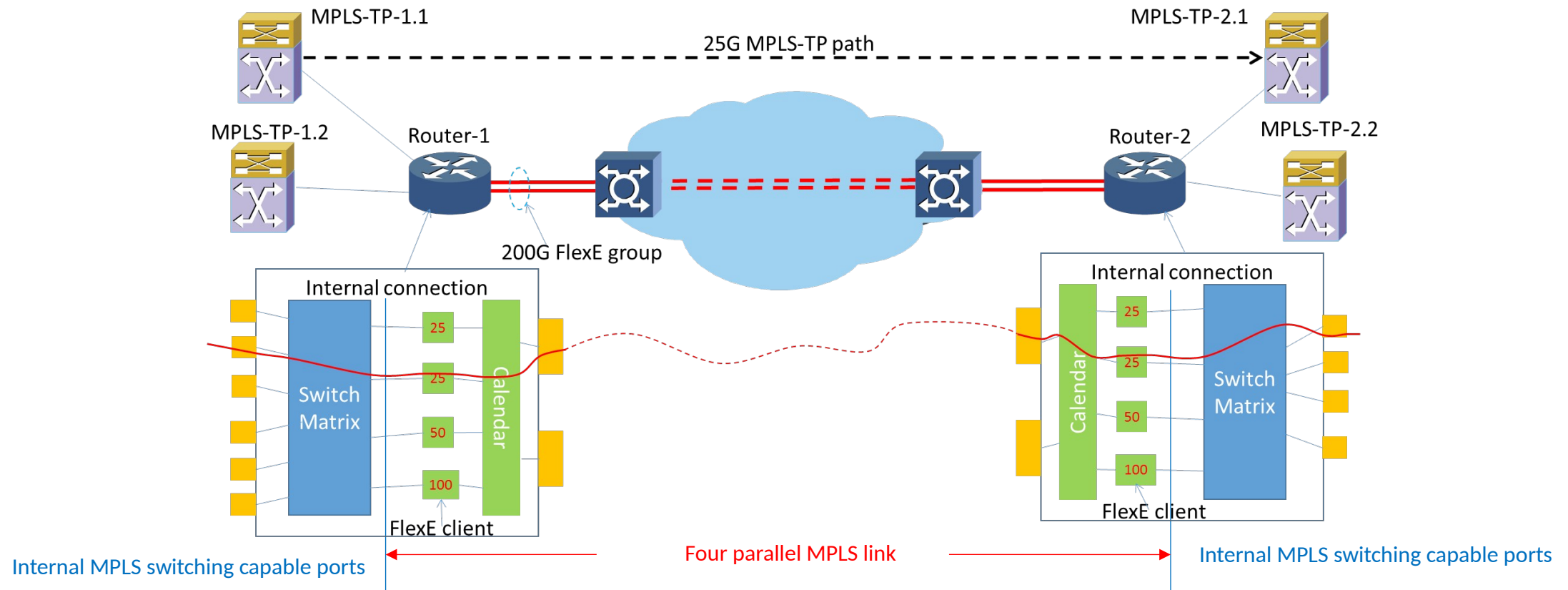
Other issues

- Whether interfaces for FlexE group and FlexE client need to be re-defined?
 - For the association interfaces between n *FlexE clients and a group, one equipment could be aware of this interface through group ID. This interface resides internal in the Calendar, maybe no need to expose it.
 - For FlexE client interface, it's just a MAC-based flow. Reuse of Ethernet interface module (e.g. , appendix A of RFC8343) should be feasible.



How can MPLS layer use flexe?

- Once the FlexE group and client is configured, for the client MPLS layer, the resource can be seen between Router-1 and Router-2 is four internal separate MPLS-TP switching capable ports with 25 G, 25G, 50G and 100G. . This is a multi-layer issue as defined in [RFC4206]. Connectivity of the lower layer is provided to MPLS layer. MPLS layer is not aware of the existence of FlexE.



Conclusion

- FlexE is just an interface technology. Only some initial configuration for FlexE group and client is needed for this interface. Then it can be used in the same way as traditional interface.

Next step

- Invite more people to join this work.
- Keep updating the analysis and model draft.

Thanks

