Proposal for a new OAM layering model
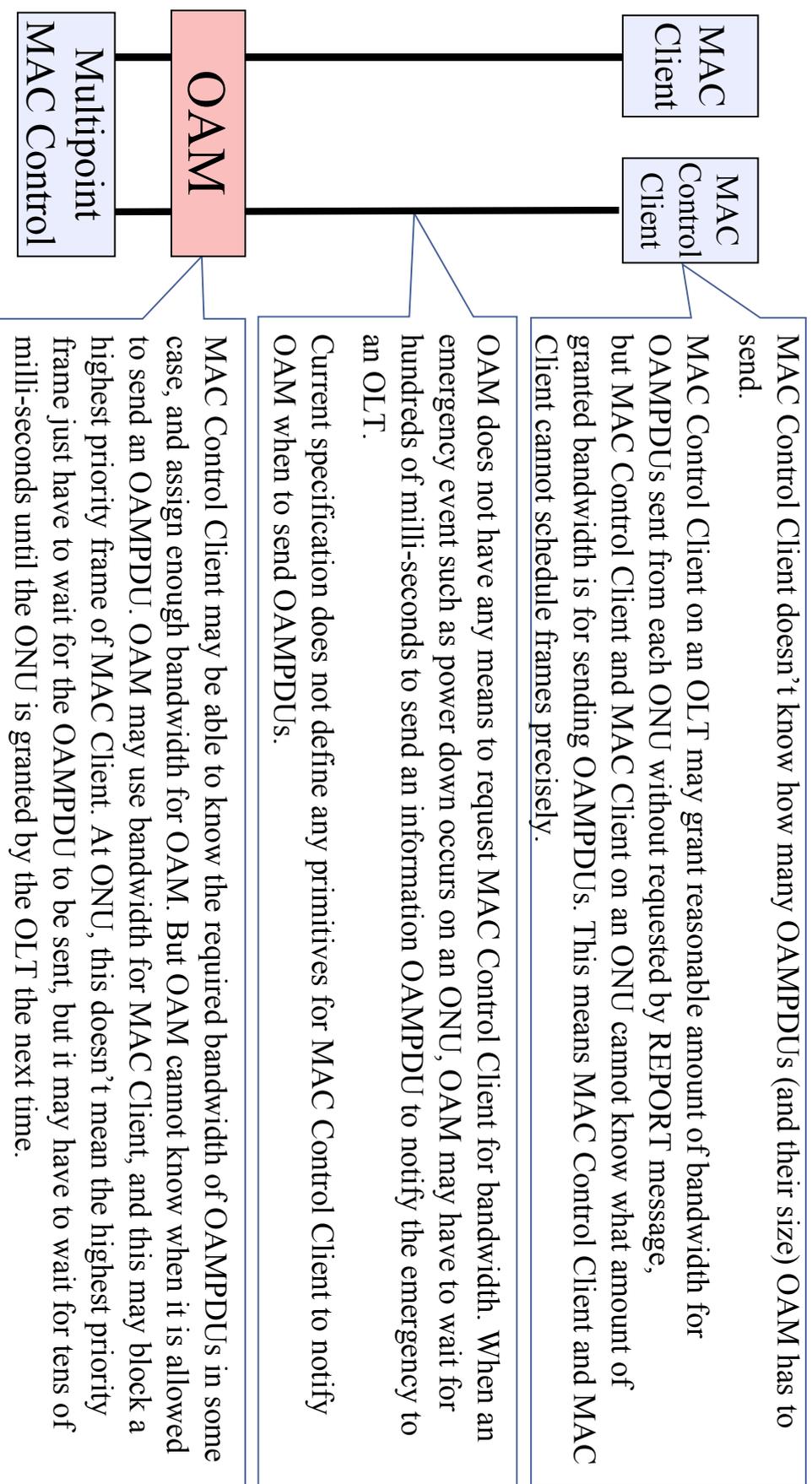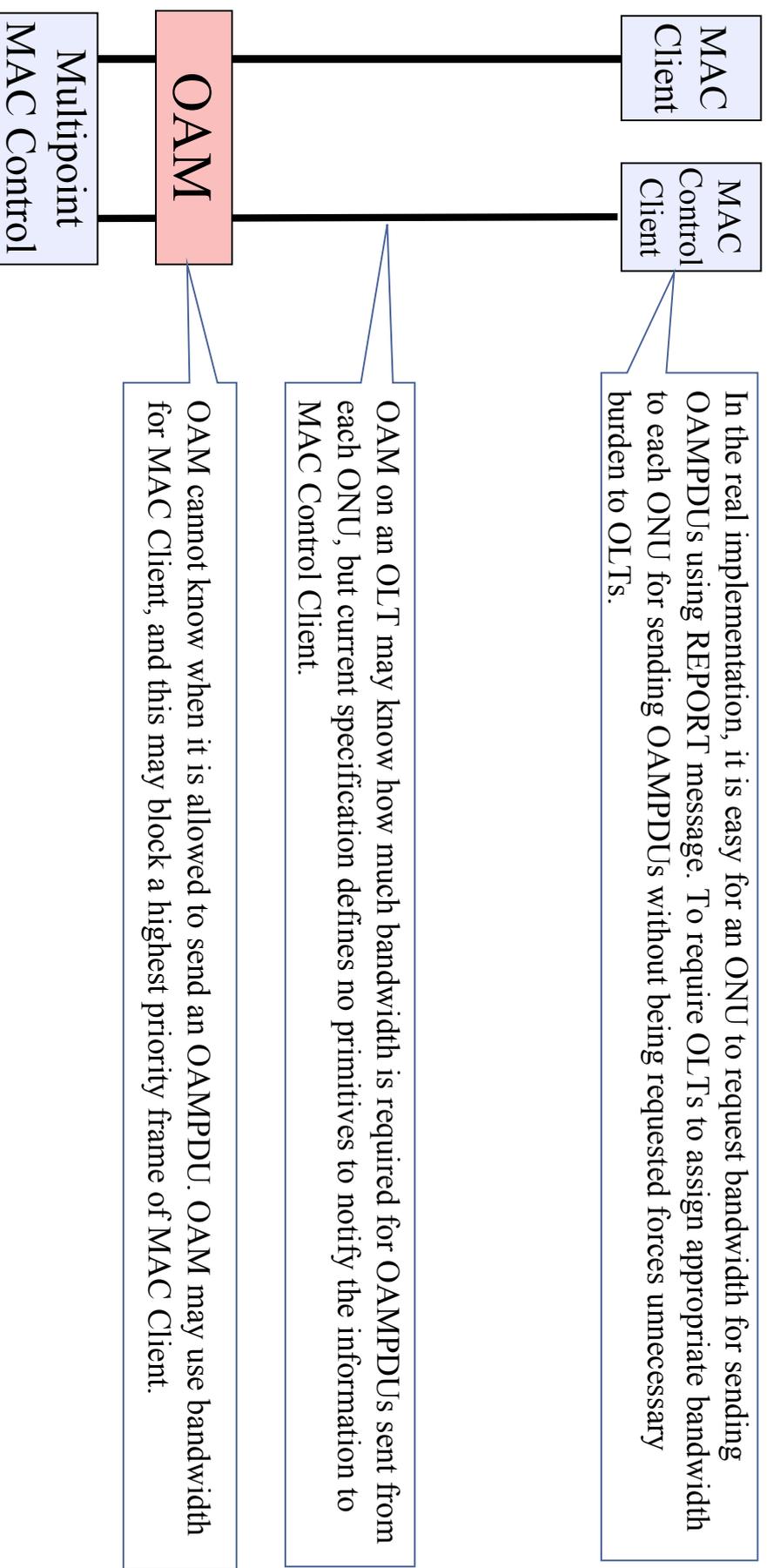
11/1/2002

Shin Yoshida, Sumitomo
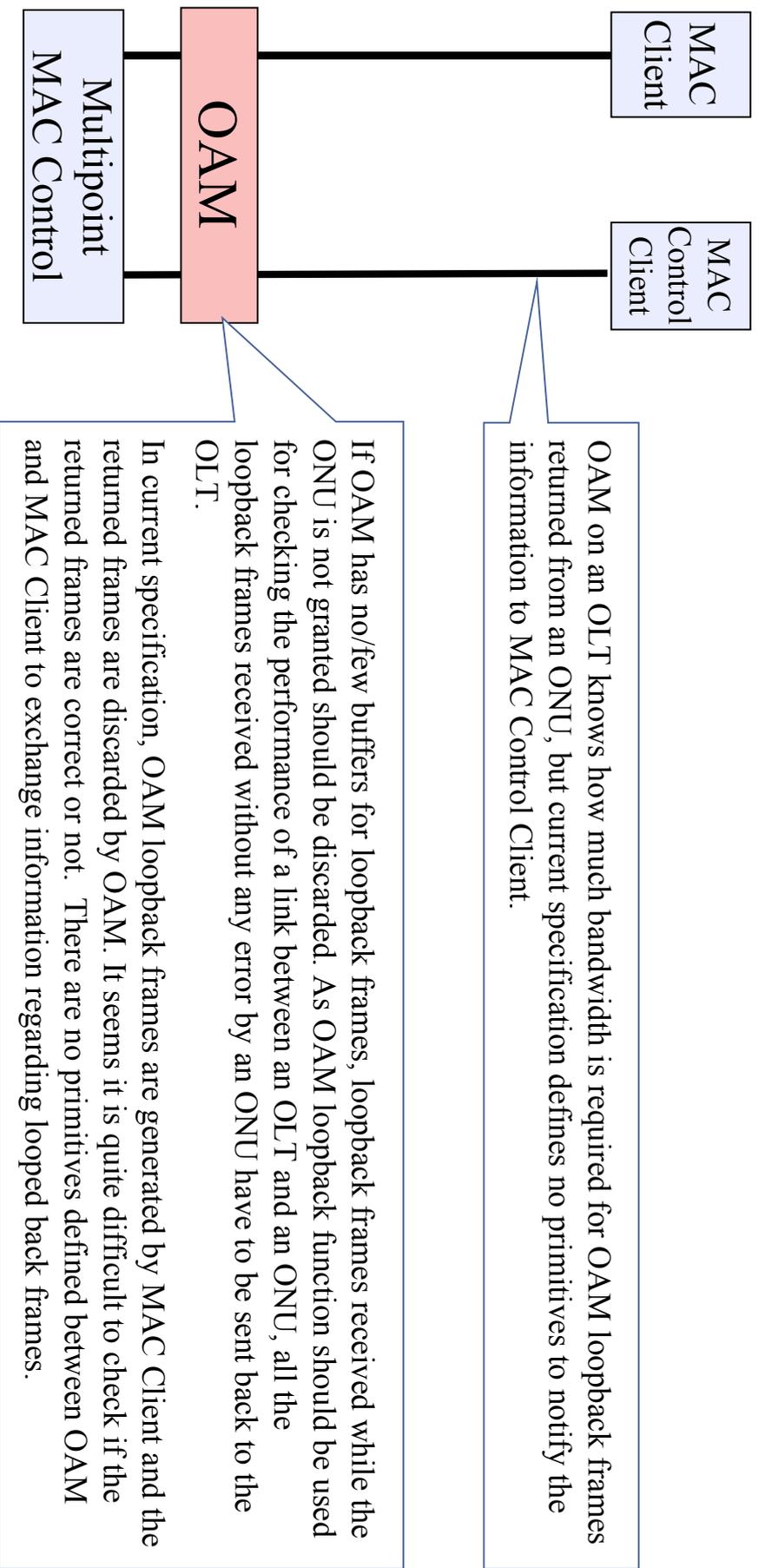
Problem on P2MP and OAM on an ONU

MAC Client

MAC Control Client

OAM

Multipoint MAC Control

MAC Control Client doesn't know how many OAMPDUs (and their size) OAM has to send.

MAC Control Client on an OLT may grant reasonable amount of bandwidth for OAMPDUs sent from each ONU without requested by REPORT message, but MAC Control Client and MAC Client on an ONU cannot know what amount of granted bandwidth is for sending OAMPDUs. This means MAC Control Client and MAC Client cannot schedule frames precisely.

OAM does not have any means to request MAC Control Client for bandwidth. When an emergency event such as power down occurs on an ONU, OAM may have to wait for hundreds of milli-seconds to send an information OAMPDU to notify the emergency to an OLT.

Current specification does not define any primitives for MAC Control Client to notify OAM when to send OAMPDUs.

MAC Control Client may be able to know the required bandwidth of OAMPDUs in some case, and assign enough bandwidth for OAM. But OAM cannot know when it is allowed to send an OAMPDU. OAM may use bandwidth for MAC Client, and this may block a highest priority frame of MAC Client. At ONU, this doesn't mean the highest priority frame just have to wait for the OAMPDU to be sent, but it may have to wait for tens of milli-seconds until the ONU is granted by the OLT the next time.

Problem on P2MP and OAM on an OLT

MAC Client

MAC Control Client

OAM

Multipoint MAC Control

In the real implementation, it is easy for an ONU to request bandwidth for sending OAMPDUs using REPORT message. To require OLTs to assign appropriate bandwidth to each ONU for sending OAMPDUs without being requested forces unnecessary burden to OLTs.

OAM on an OLT may know how much bandwidth is required for OAMPDUs sent from each ONU, but current specification defines no primitives to notify the information to MAC Control Client.

OAM cannot know when it is allowed to send an OAMPDU. OAM may use bandwidth for MAC Client, and this may block a highest priority frame of MAC Client.

Problem on P2MP and OAM Loopback

MAC Client

MAC Control Client

OAM

Multipoint MAC Control

OAM on an OLT knows how much bandwidth is required for OAM loopback frames returned from an ONU, but current specification defines no primitives to notify the information to MAC Control Client.

If OAM has no/few buffers for loopback frames, loopback frames received while the ONU is not granted should be discarded. As OAM loopback function should be used for checking the performance of a link between an OLT and an ONU, all the loopback frames received without any error by an ONU have to be sent back to the OLT.

In current specification, OAM loopback frames are generated by MAC Client and the returned frames are discarded by OAM. It seems it is quite difficult to check if the returned frames are correct or not. There are no primitives defined between OAM and MAC Client to exchange information regarding looped back frames.

Remedy: Define OAM Client

◆ Define OAM Client in the same layer as MAC Client and MAC Control Client.

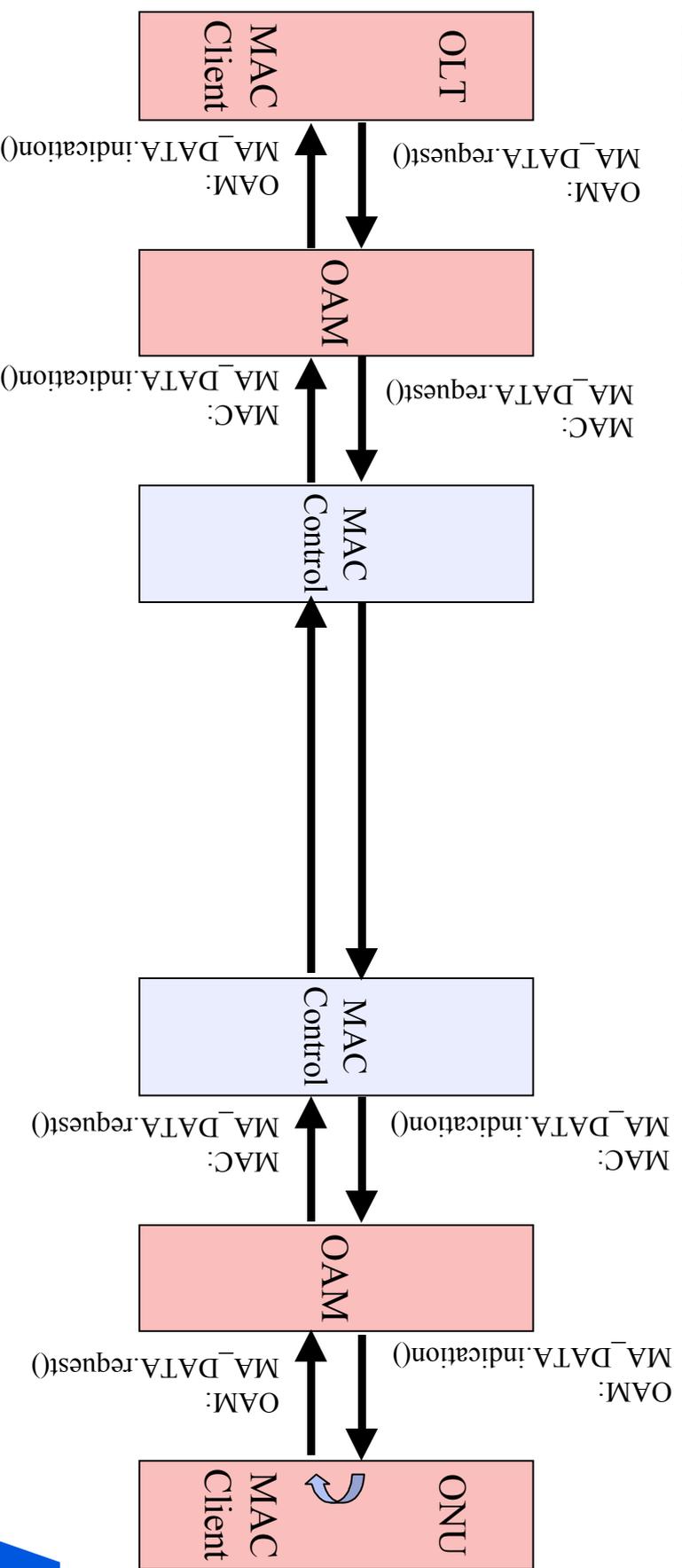◆ OAM Client should be able to exchange information closely with MAC Control Client (and MAC Client).
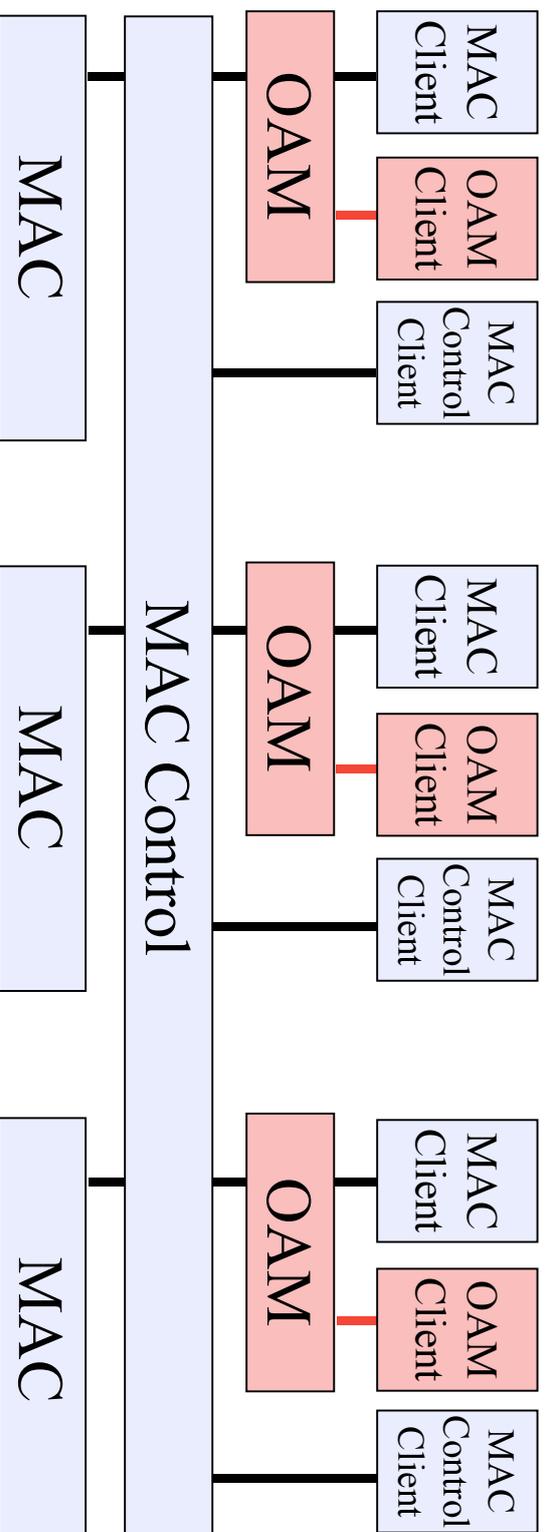
Remedy: define MA_OAM.req/ind primitives

◆ OAM Client issues MA_OAM.request() to ask OAM to send an OAMPDU.

◆ OAM issues MA_OAM.indication() to notify OAM Client that it received an OAMPDU.

MAC Client

OAM Client

OAM

MAC Control

MA_OAM.request()

MA_OAM.indication()
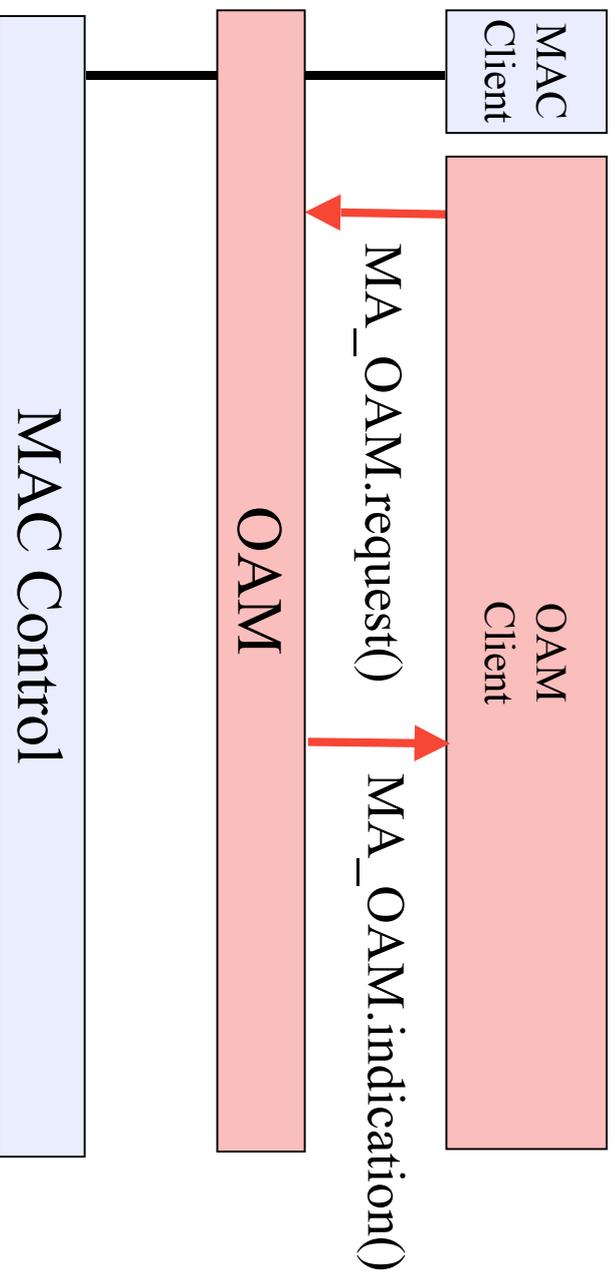
Remedy: OAM loopback at MAC Client

- OAM loopback frames are generated by MAC Client on an OLT,
- OAM loopback frames received by an ONU are looped back by MAC Client,
- OAM loopback frames received by an OLT are checked and discarded by MAC Client.

OLT

| MAC Client |

- OAM: MA_DATA.indication()
- OAM: MA_DATA.request()

| OAM |

- MAC: MA_DATA.indication()
- MAC: MA_DATA.request()

| MAC Control |

| MAC Control |

- MAC: MA_DATA.request()
- MAC: MA_DATA.indication()

| OAM |

- OAM: MA_DATA.request()
- OAM: MA_DATA.indication()

ONU

| MAC Client |

Conclusion

- Current OAM and P2MP specification cannot solve timing issues for sending OAMPDUs.

- Current OAM architecture may be able to solve most of the problem. But it requires a lot of hidden information path among OAM, MAC Control Client, MAC Client, Layer Management and so on.

- The proposed remedy solve the problem in a very simple way.

- This proposal does not cause any problem on P2P links.