



# UMT Layering

Glen Kramer

[glen.kramer@broadcom.com](mailto:glen.kramer@broadcom.com)

# 802.1 MAC Bridge Architecture

□ Per 802.1 architecture, MAC Relay entity is located above the Media Access Method Specific Functions (802.n)

□ We need to decide where UMT sublayer fits in this architecture:

- As part of Media Access Method (below MAC Relay) or
- As part of Higher Layer entities (above MAC Relay)

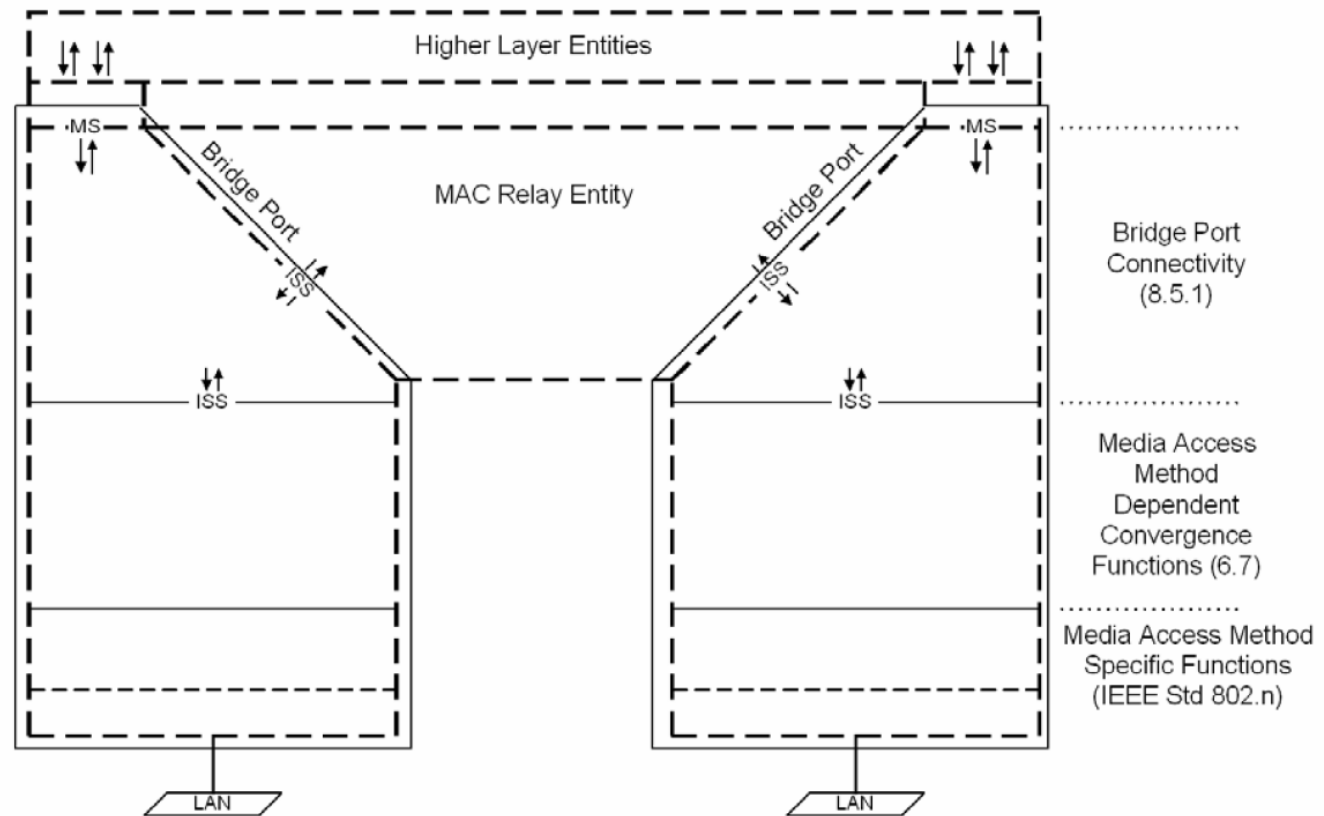
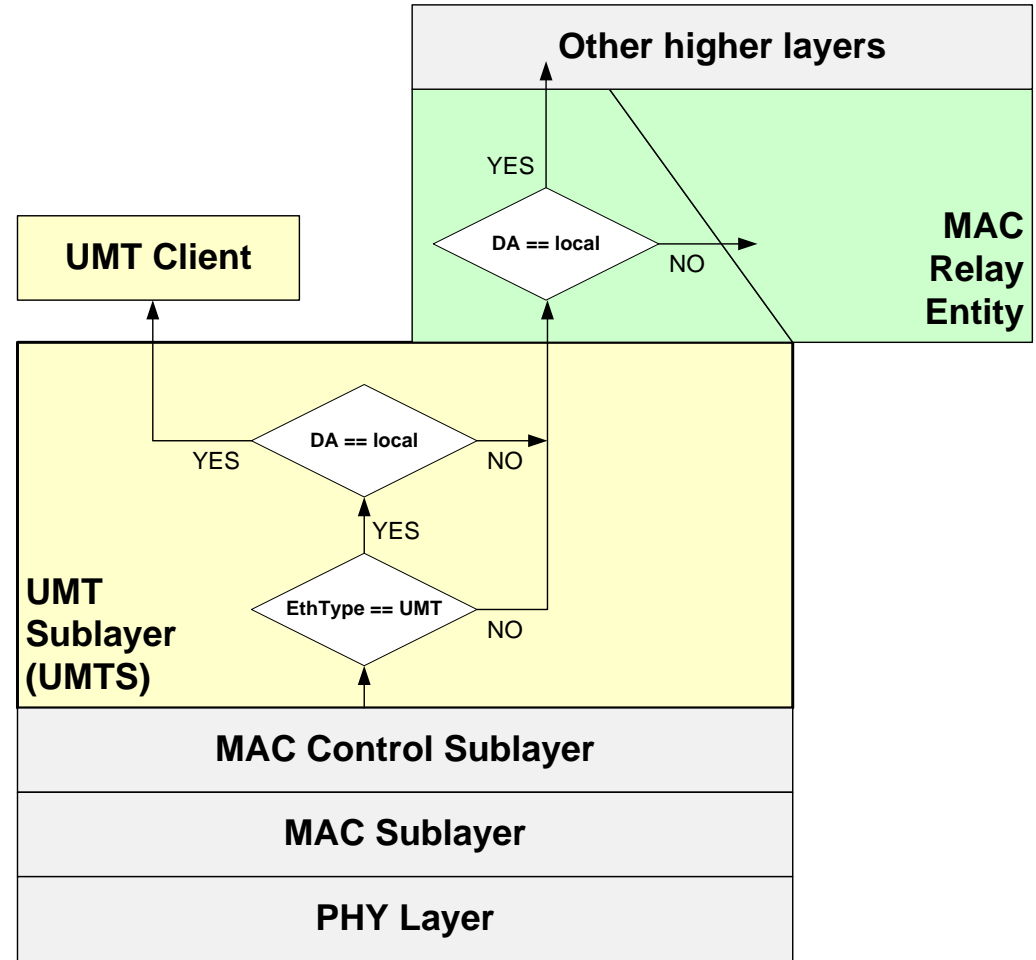


Figure 8-3—MAC Bridge architecture

## 1. UMT sublayer filters based on UMT type

- Non-UMT frames are passed to higher layers
- UMT frames are checked for local/non-local DA
  - Non-local frames are passed to MAC Relay
  - Local frames are passed to local UMT Client

## 2. MAC Relay entity checks frames for local/non-local DA (again)

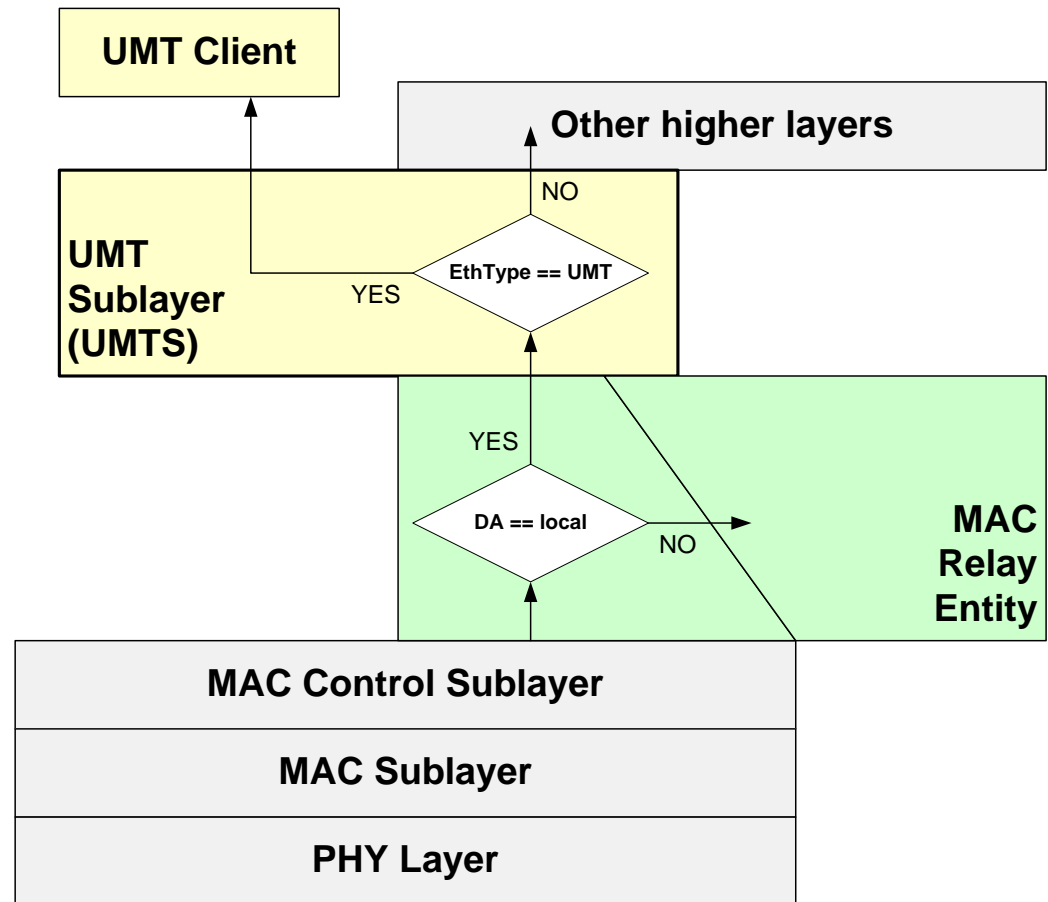


## 1. MAC Relay filters frames based on DA.

- Frames with non-local DA are relayed
- Frames with local DA are passed to UMT sublayer

## 2. UMT sublayer filters frames based on UMT type

- Non-UMT frames are passed to higher layers
- UMT frames are passed to local UMT Client



# UMT sublayer above MAC Relay

- ❑ When we need to do the same operation several times at different sublayers (like checking `DA == local/non-local`), this is a sign that we are trying to push some functions to a wrong sublayer.
  - I.e., pushing UMT sublayer to be below MAC Relay.
- ❑ It appears that placing UMT sublayer above MAC Relay is the proper architecture.
  - All frames with non-local DA are treated identically (i.e., relayed), regardless of their EtherTypes.



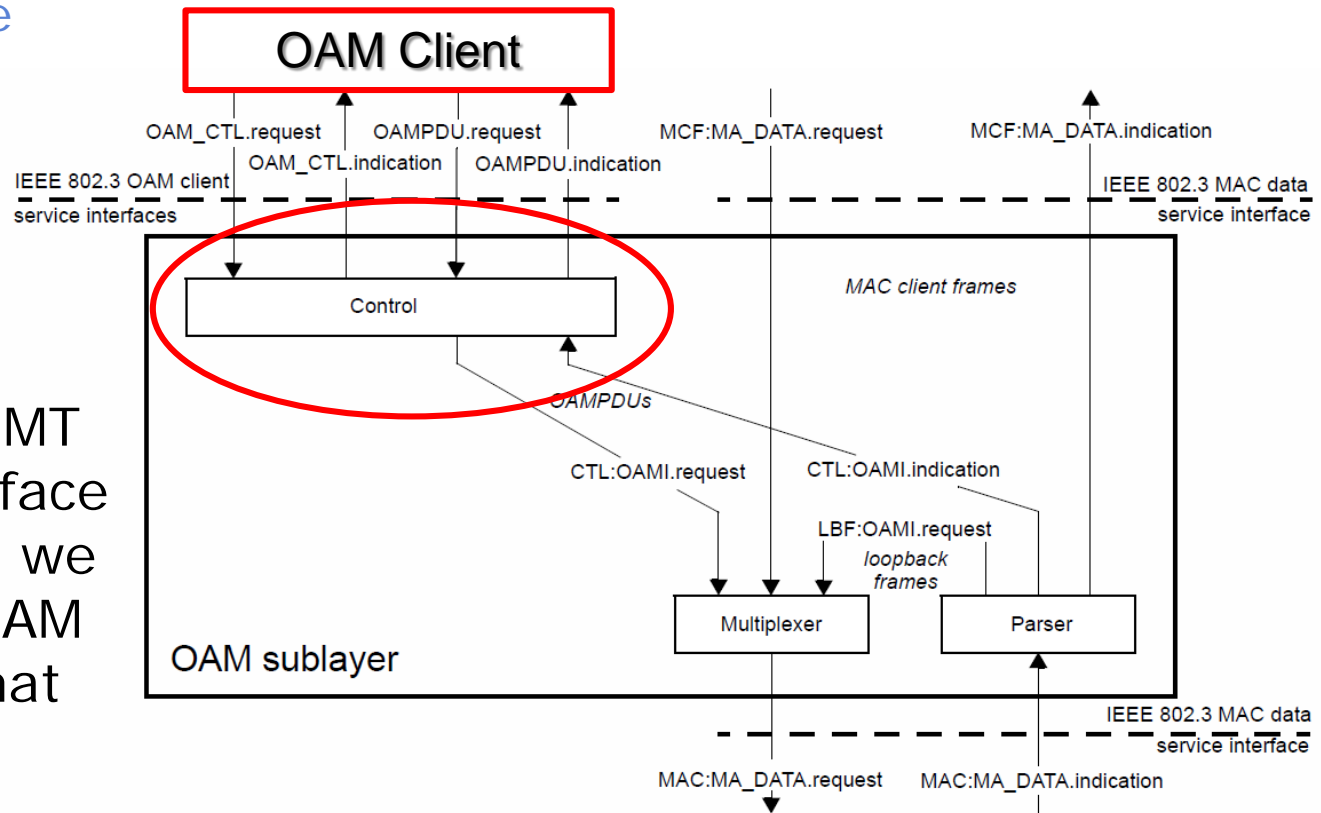
# Combining OAMPDUs and UMTTPDUs/Type 3

# Interfacing UMT with existing clients

- Besides the usual Parser and Multiplexor, the OAM sublayer implements a number of control functions, such as

- Discovery state diagram (See Fig 57-5)
- Transmit state diagram (see Fig 57-6)

- If we want the UMT sublayer to interface with OAM Client, we need to define OAM Client adaptor that implements the same Discovery and Transmit state diagrams



Instances of OAM internal service interfaces:

- CTL:OAMI.indication = Passes OAMPDUs to OAM Control
- CTL:OAMI.request = Passes OAMPDUs to Multiplexer
- LBF:OAMI.request = Passes loopback frames to Multiplexer

Instances of MAC data service interface:

- MAC=interface to subordinate sublayer
- MCF=interface to MAC client

# Discovery and Transmit SDs

- Both state diagrams control current state of the OAM sublayer.
  - Contain pdu\_timer, pdu\_cnt counter, etc.

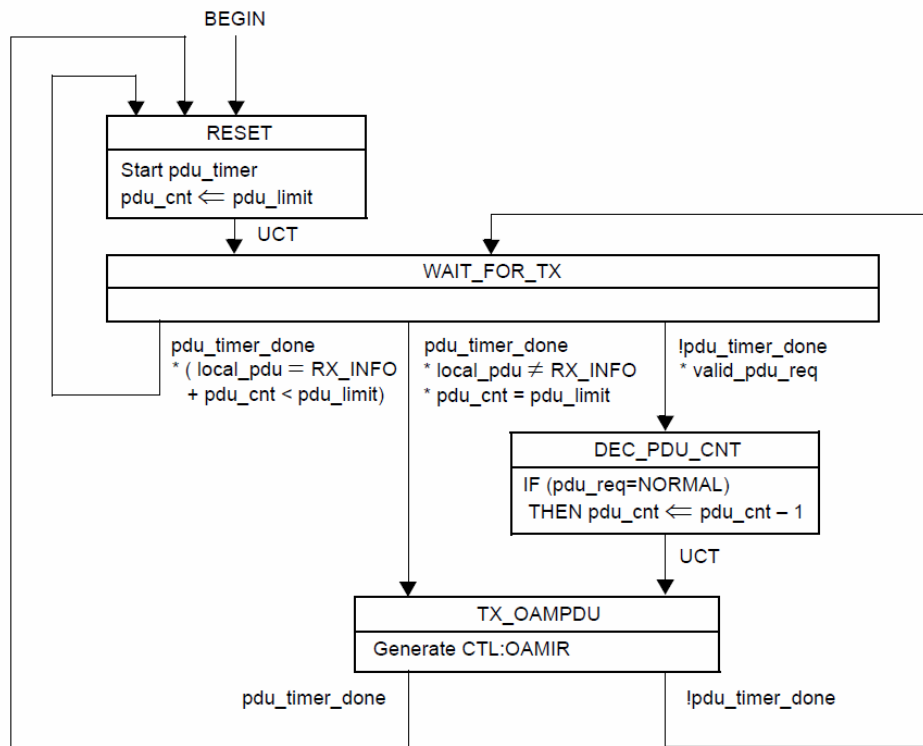


Figure 57-6—Transmit state diagram

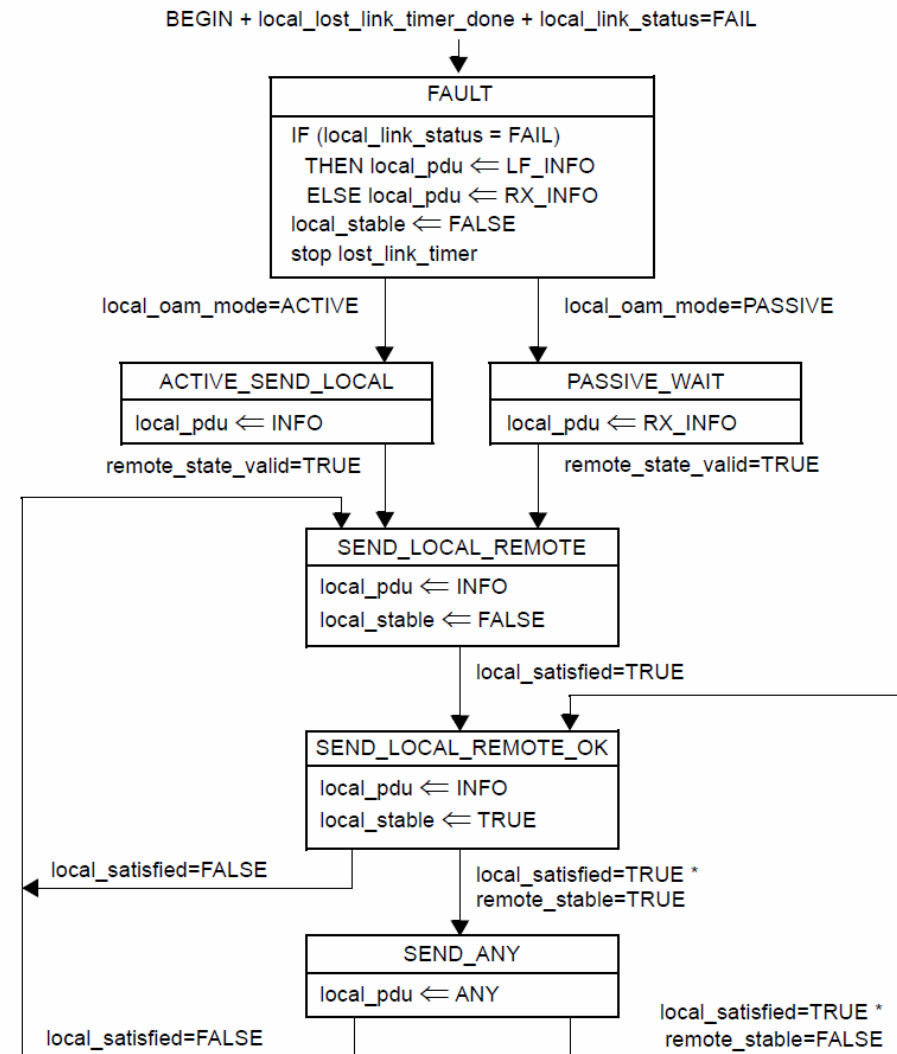
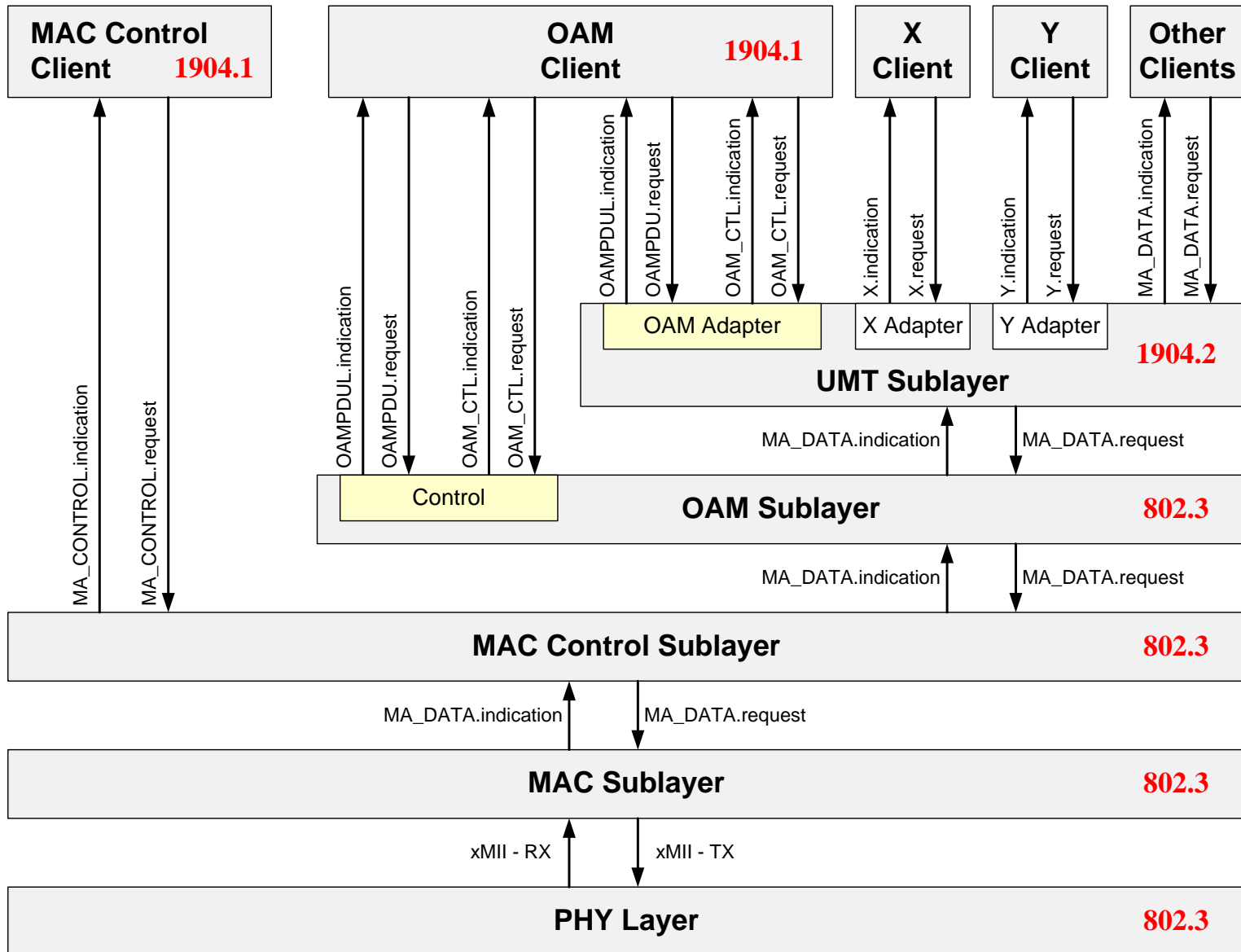


Figure 57-5—OAM Discovery state diagram

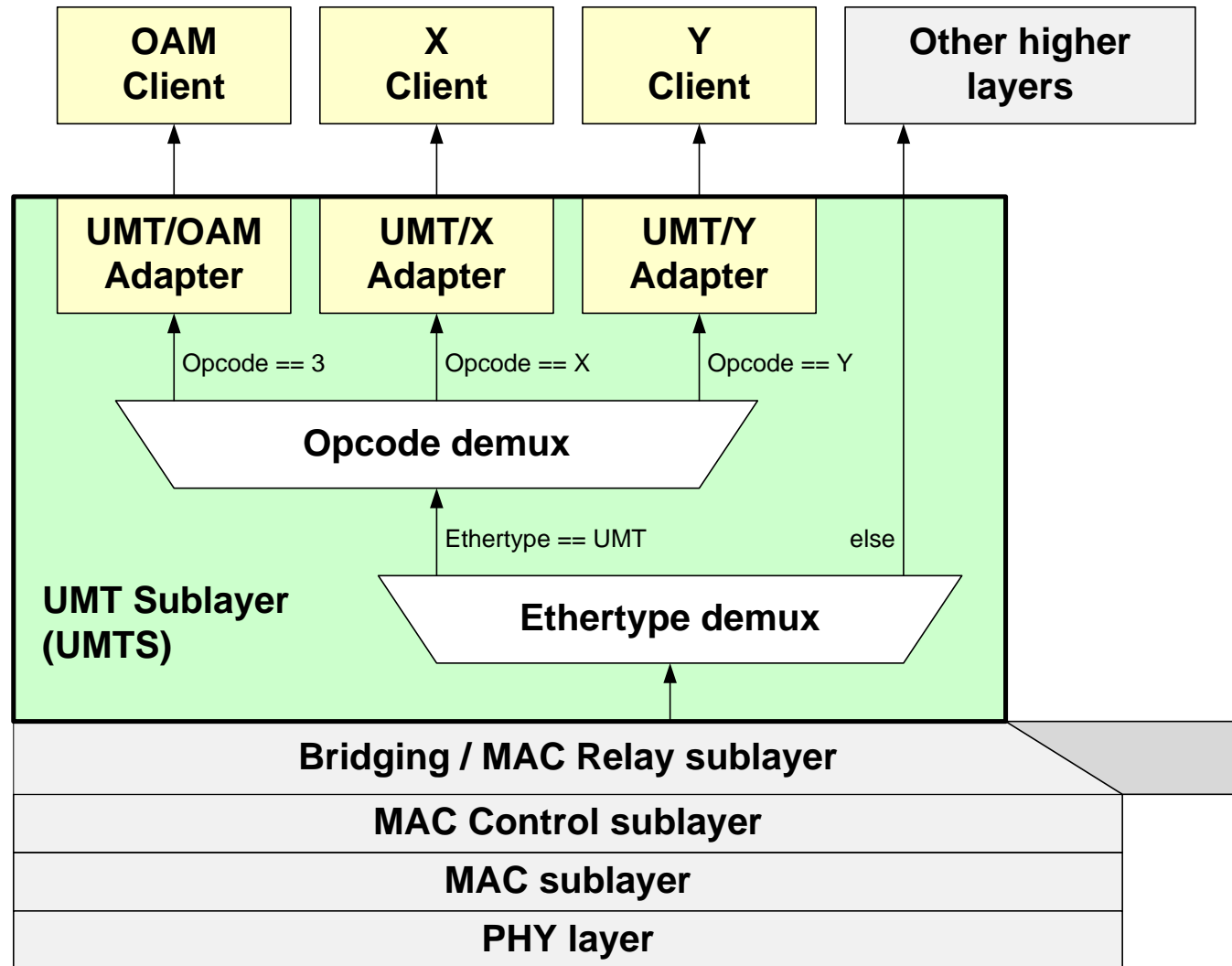


# Layering Diagram



# A look inside UMT sublayer

- ❑ In the receive direction, the UMT sublayer demultiplexes frames based on Ethertype first.
- ❑ Non-UMT frames are passed to higher layers
- ❑ UMT frames demultiplexed based on opcodes and passed to appropriate protocol adapter
- ❑ **The UMT sublayer never sees any frames with non-local DA**



- ❑ Sharing the same client between native transport channel and UMT transport channel appears problematic.
  - OAM is an example. Duplicating Discovery SD or Transmit SD in the UMT/OAM adapter will require the same OAM client to be discovered twice. Also requires independent OAM keep-alive messages on each transport channel.
  - If one channel goes down, should the OAM client deregister, or remain operational on the remaining channel? This is a new behavior for the OAM client.
  - Similar ambiguities exist for most other (stateful) protocols.
- ❑ We cannot allow multiple instances of the same protocol client.
  - For example, OAM client reflects the actual hardware state. Two clients for the same hardware will result in numerous conflicts and failures
    - On which channel the alarms are sent?
    - Both clients may attempt conflicting actions. Race conditions.
- ❑ Should we disallow sharing a client between the native transport channel and the UMT transport channel?
  - For example, for OAM this means use either OAMPDUs or UMTPDUs/Type3, but not both.



**Thank You**