IEEE 1904.2 UMT

Base Architecture and Operation

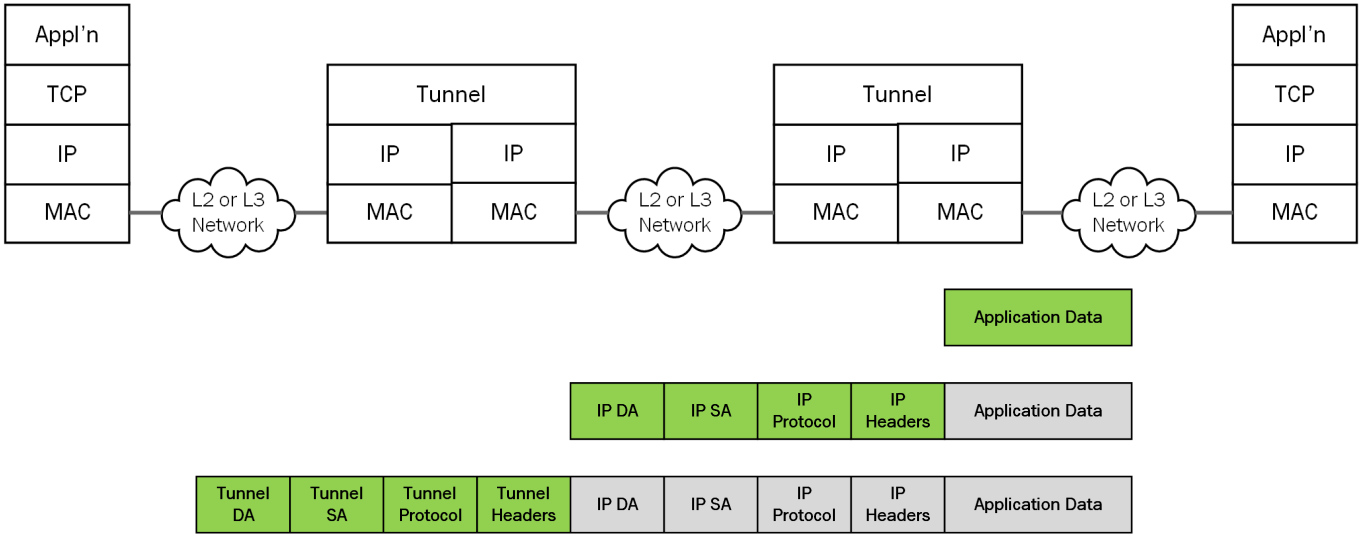
Kevin A. Noll, Tibit Communications

Key Operational Ideas

- Although called a “tunnel”, UMT is different than most other tunneling protocols

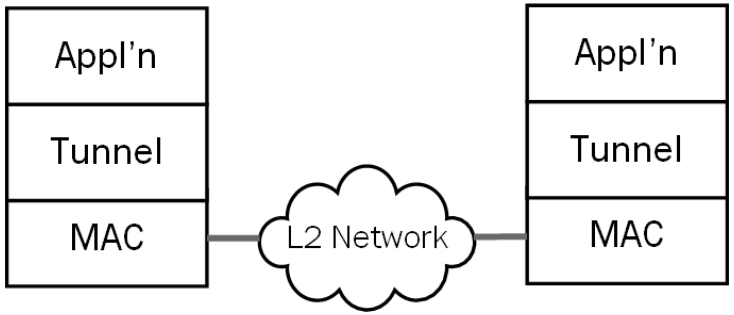
Typical Tunnel

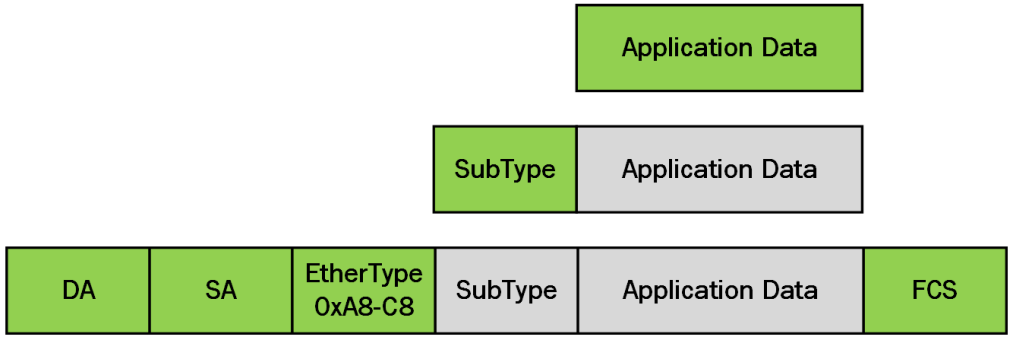
- Pre-defined and fixed endpoints (in a one-to-one relationship) for each “tunnel” or “session”
- Applications use their native stack and the tunnel endpoint ”captures” traffic that is to be tunneled



UMT – Key Operational Ideas

- A tunnel in UMT does not need to have fixed and pre-specified "tunnel endpoints"
- In UMT the application uses the tunnel directly
- Another way to look at it – UMT appears to the client as an alternate MAC layer



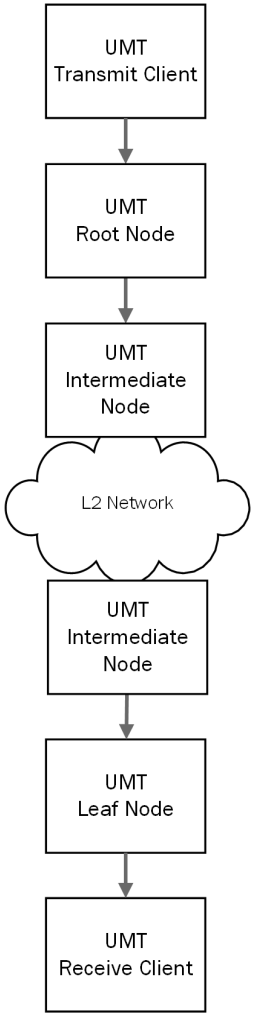


UMT – Key Operational Ideas

- When a frame enters the UMT “tunnel”, it could be delivered to {ONE|SEVERAL|ALL} UMT clients in a given Ethernet broadcast domain.
- The choice of {ONE|SEVERAL|ALL} depends on the DA requested by the UMT client.
- The UMT client will need to be updated with this in mind. For example, if IPv4 is expected to operate over UMT, ARP might need to be implemented over UMT so that automatic neighbor discovery works.

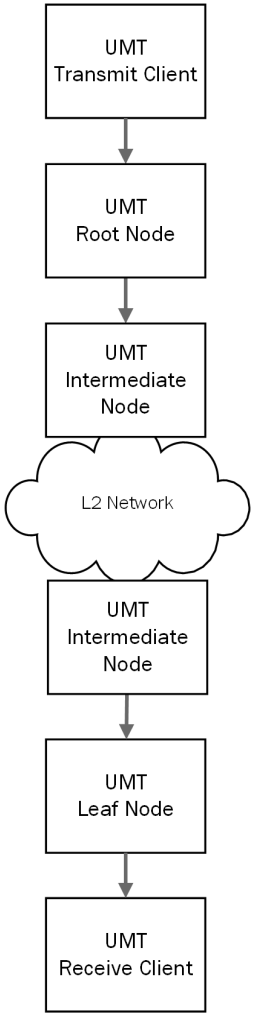
Architecture – Functional Units

- UMT Client – There are two client functions
  - UMT Transmitting Client (UMT-TxC) – Station sending management frames (e.g. OAM)
  - UMT Receiving Client (UMT-RxC) – Station receiving management frames (e.g. OAM)
  - There may be only one or both client functions in a station
  - There may be more than one instance of a client function in a station
  - UMT-TxC and UMT-RxC are only functional entities for the description of protocol operation.



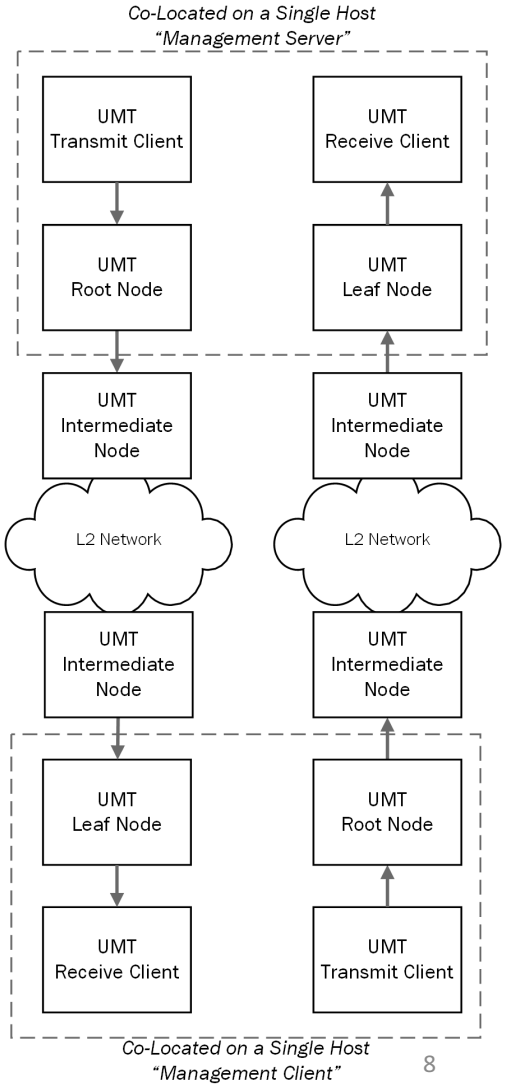
Architecture – Functional Units

- UMT Root Node (UMT-RN) – Encapsulates management frames transmitted by UMT-TxC
- UMT Leaf Node (UMT-LN) – Decapsulates management frames sent by UMT-RN and transmits the management frame to the intended UMT-RxC
  - UMT-RN and UMT-LN are separate because there could be a few to many relationship between RN and LN (e.g. one RN and many LN) and it is possible that only a receive function is needed in a given node.
  - Separation helps describe the needed functions in the standard. Implementation can be different.
- UMT Intermediate Node (UMT-IN) – MAC-layer bridge/switch forwards UMT frames toward UMT-LN based only on L2 forwarding rules found in 802.1D, 802.1Q, and related standards



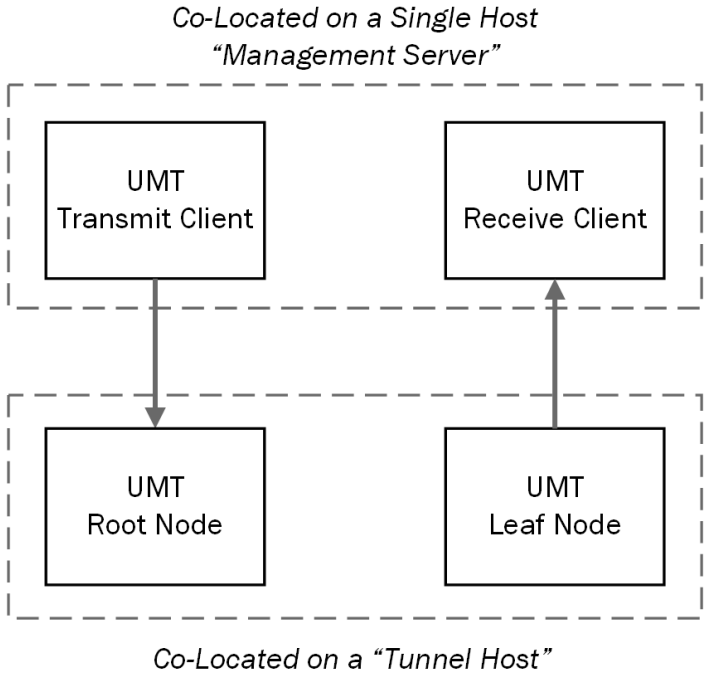
What About a Real System?

- UMT-TxC and UMT-RxC are the software that want to use UMT, e.g. OAM
- Highly probable that any UMT-TxC will also be an UMT-RxC and vice versa
- UMT-RN and UMT-LN are to be specified as a single layer



What about a real system?

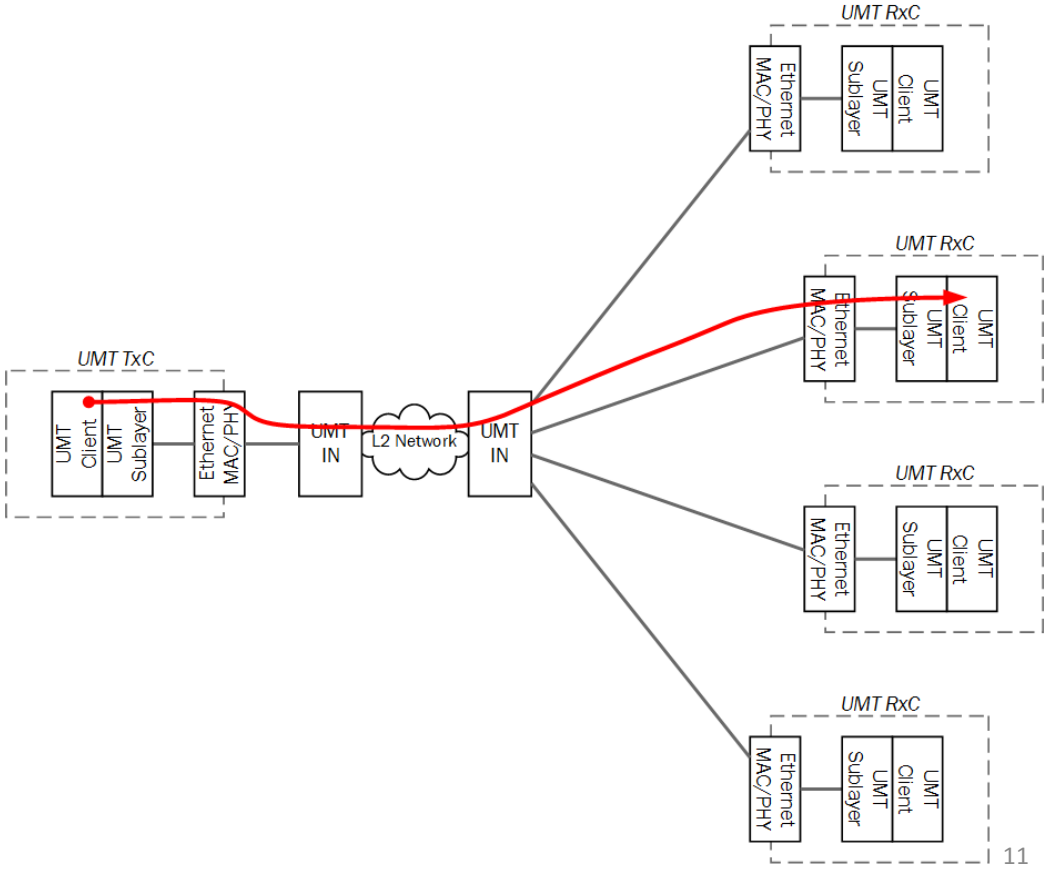
- UMT-TxC and UMT-RxC are most likely to be co-located on a host, but not necessarily.
- UMT-TxC/RxC and UMT-RN/LN are not required to be co-located.
- The implementation-specifics of the interface between the UMT Client functions and the Root/Leaf Node functions are outside the scope of 1904.2.



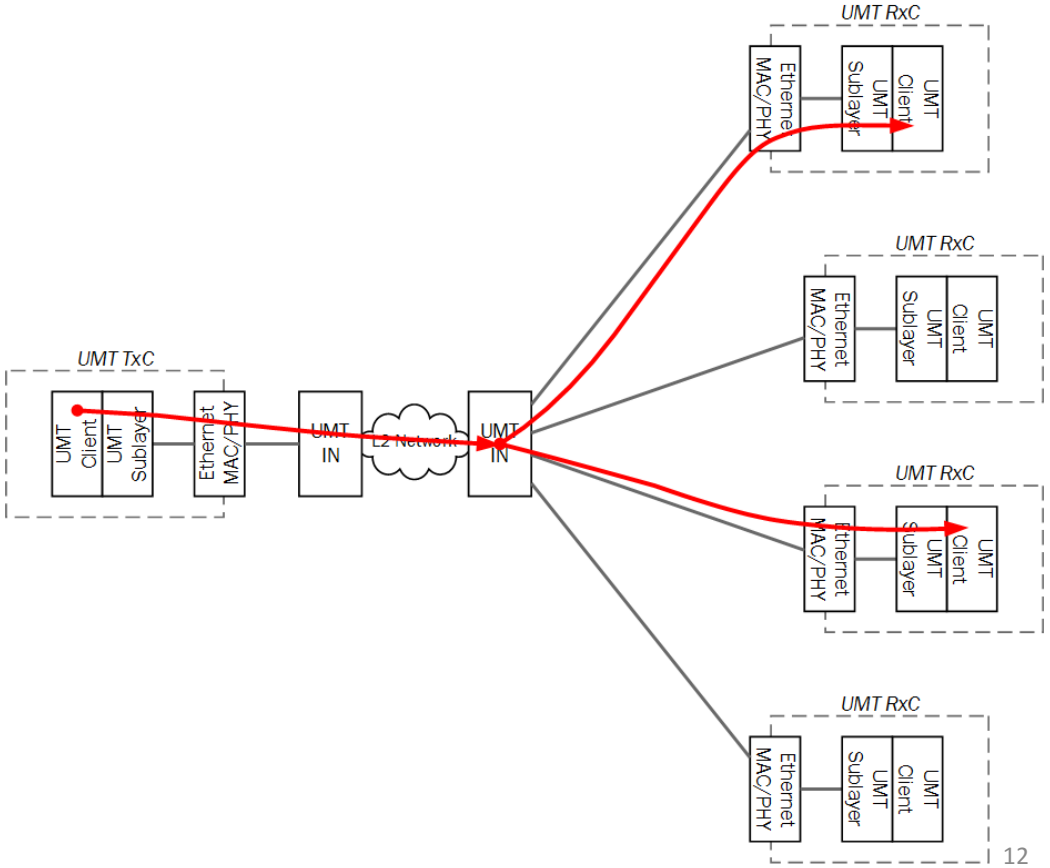
Addressing

- UMT frame DA MAY be a unicast address
- UMT frame DA MAY be broadcast address
  - For example – To enable node/client discovery
- UMT frame DA MAY be a multicast address
- UMT frame Ethertype must be set to the value assigned by IEEE-SA
- UMT frame may be carried within a VLAN (informative text may be added to describe this, but specifics are out of scope)
- UMT Payload addressing is out of scope

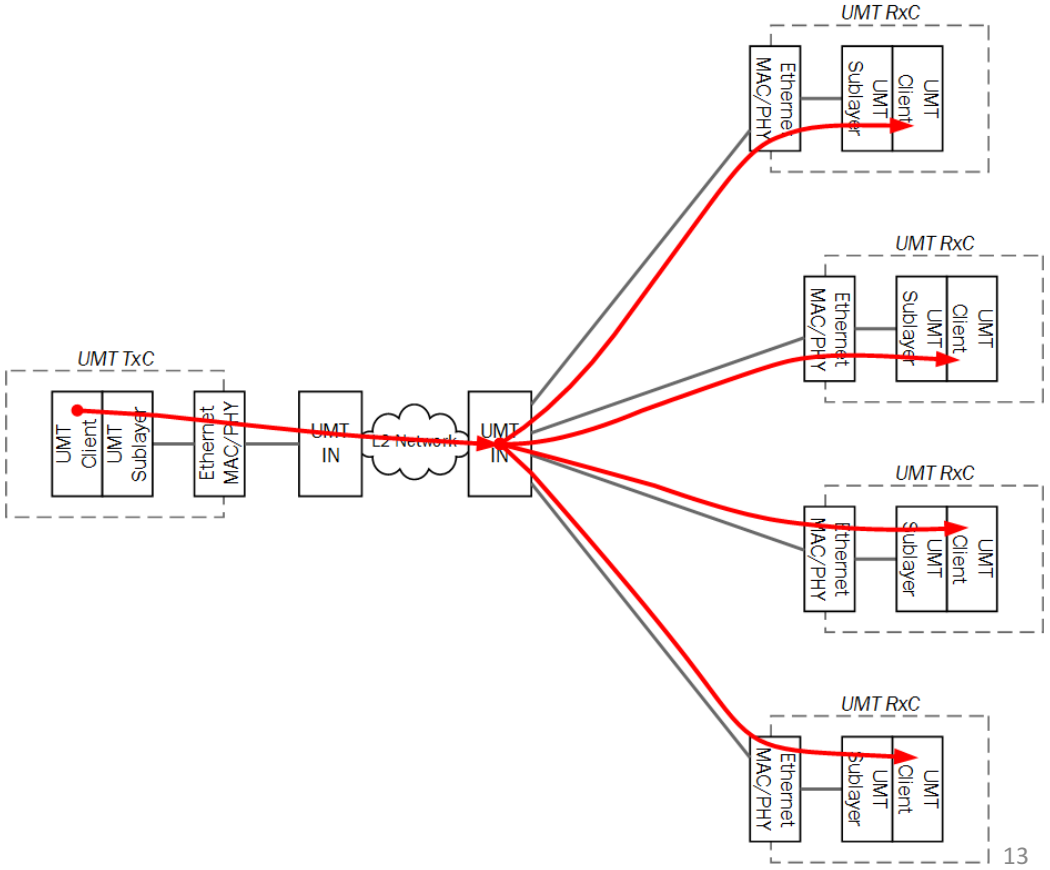
UMT Delivery with Unicast DA



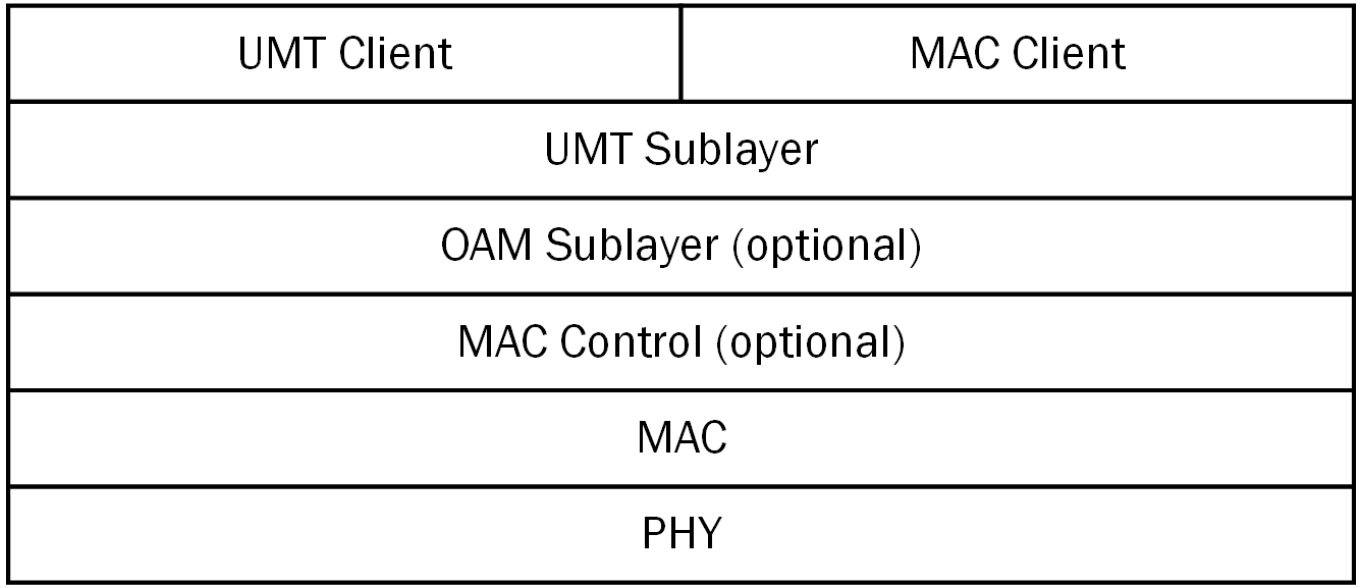
UMT Delivery with Multicast DA



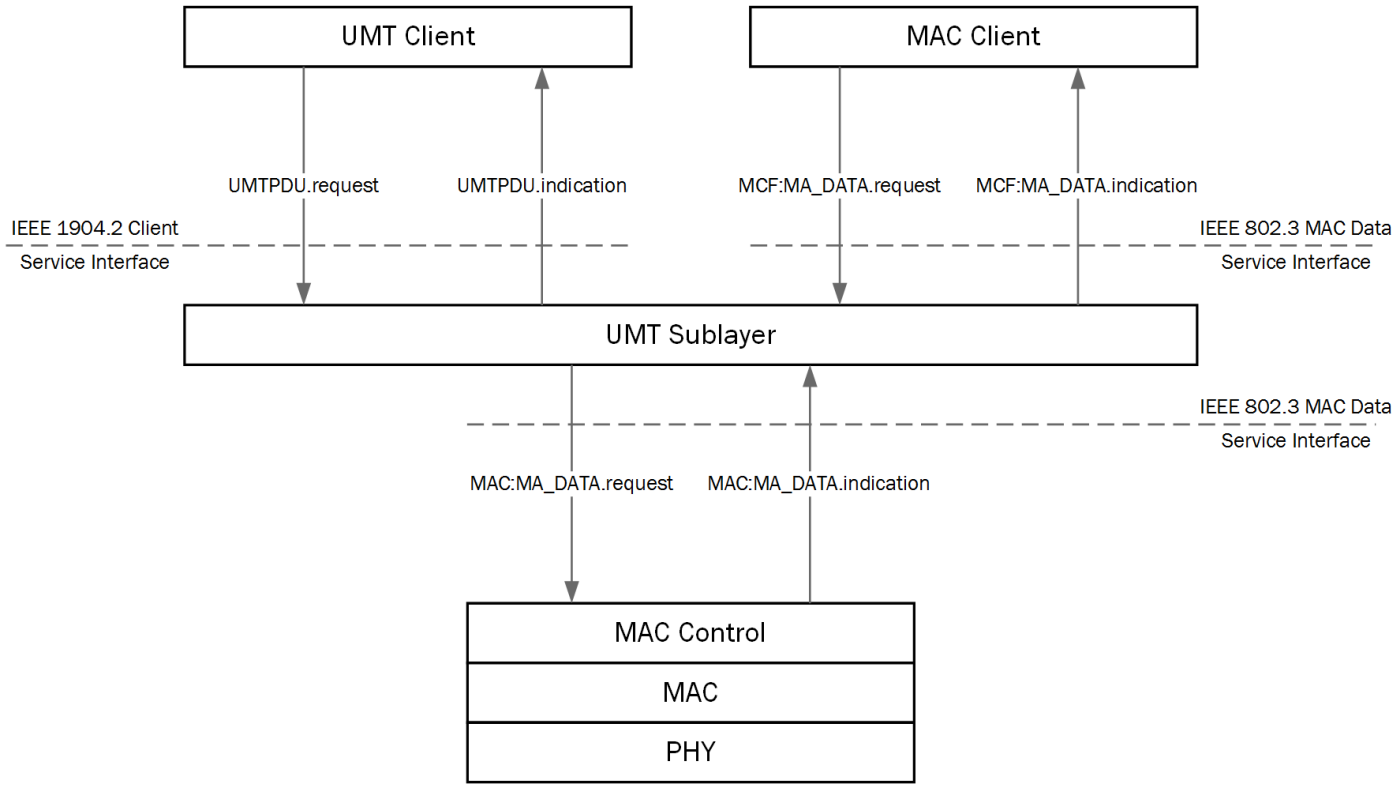
UMT Delivery with Broadcast DA



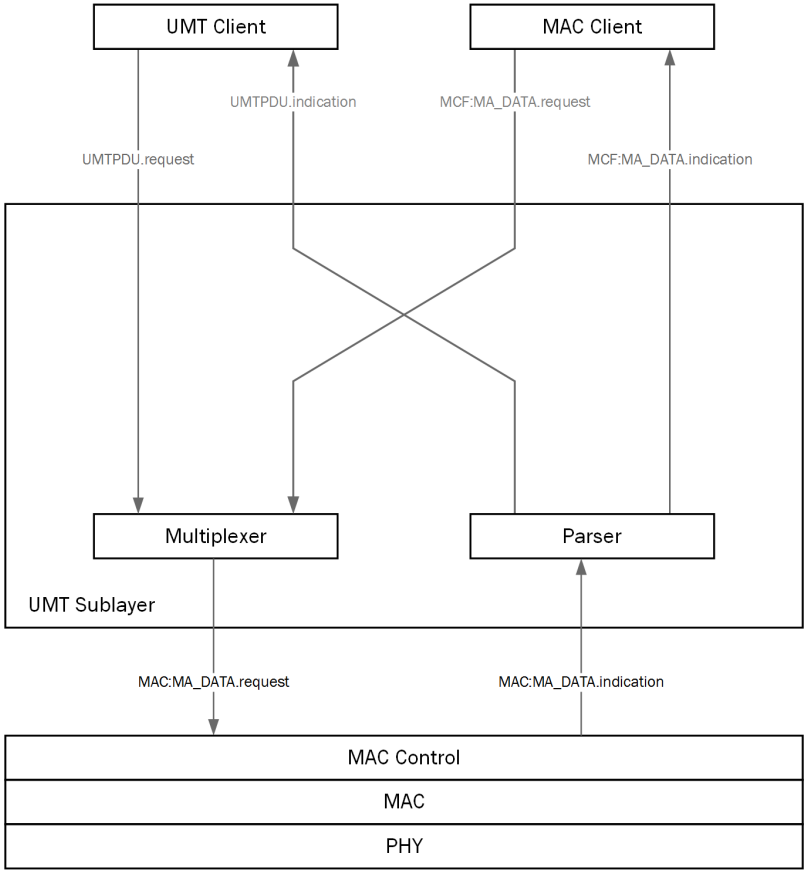
Proposed Protocol Stack for 1904.2



1904.2 Sublayer with Signals



1904.2 Sublayer Internals



1904.2 UMT Sublayer Operational Principles

- Receipt of MCF:MA_DATA.request
  - results in a call to MAC:MA_DATA.request with parameters identical to MCF:MA_DATA.request

- Receipt of MAC:MA_DATA.indication
  - UMT Parser looks at EtherType
  - If EtherType = UMT and SubType matches a UMT Client, results in a call to UMTPDU.indication to the UMT Client identified by the SubType
  - If EtherType != UMT, results in a call to MCF:DATA.indication with parameters identical to MAC:MA_DATA.indication.

1904.2 UMT Sublayer Operational Principles

- Receipt of UMTPDU.request
  - results in a call to MAC:MA_DATA.request with parameters to transmit a UMTPDU containing the UMT Client's data.

- Call to UMTPDU.indication
  - Occurs upon receipt of MAC:MA_DATA.indication and EtherType = UMT

1904.2 UMT Sublayer Function Definitions

- MCF:MA_DATA.request – Received from the superior MAC Client
  - (destination_address, source_address, mac_service_data_unit, frame_check_sequence)

- MCF:MA_DATA.indication – Sent to the superior MAC Client
  - (destination_address, source_address, mac_service_data_unit, frame_check_sequence, reception_status)

1904.2 UMT Sublayer Function Definitions

- MAC:MA_DATA.request – Sent to MAC Sublayer
  - (destination_address, source_address, mac_service_data_unit, frame_check_sequence)

- MAC:MA_DATA.indication – Received from MAC Sublayer
  - (destination_address, source_address, mac_service_data_unit, frame_check_sequence, reception_status)

1904.2 UMT Sublayer Function Definitions

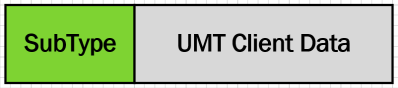
- UMTPDU.request – Received from UMT Client acting as a UMT-TxC
  - (destination_address, source_address, umt_data)

- UMTPDU.indication – Sent to UMT Client acting as a UMT-RxC
  - (destination_address, source_address, umt_data)

Architecture – Basic Flow 1/2

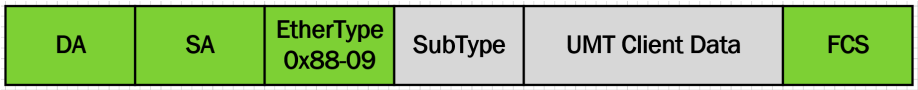
1. UMT-TxC constructs a PDU per its own standards/specifications. This becomes the UMT Payload

2. UMT-TxC sends UMT Payload to UMT Sublayer using UMTPDU.request



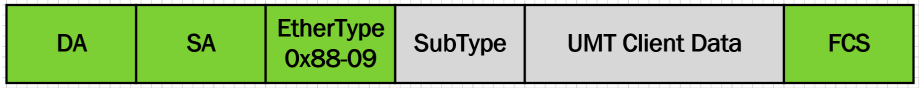
3. UMT Sublayer encapsulates the UMT Payload – creating the UMTPDU

4. UMT Sublayer delivers UMTPDU to the MAC Layer for delivery using MAC:MA_DATA.request
    1. DA is set to the intended recipient (UMT-RxC), SA is set to UMT-RN

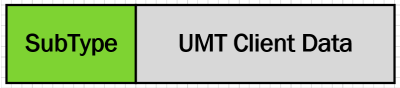


Architecture – Basic Flow 2/2

6. UMT-IN (if present) forwards UMTPDU toward UMT-LN
7. UMT-LN MAC Layer receives frame



8. MAC Layer delivers frame to UMT Sublayer using MAC:MA_DATA.indication



9. UMT Sublayer decapsulates UMT payload
10. UMT Sublayer forwards UMT payload to UMT-RxC using UMTPDU.indication

11. UMT-RxC receives UMT payload

Thanks!

Q&A