VOLTHA

OMCI-LIB-GO Update

2021/09/08

Code Changes

- Broke messagetypes.go (~4300 lines) and associated messagetypes_test.go into 15 ‘message type’ specific go files   (get.go, alarm.go, mibreset.go, …)
    - Allows for easier management of any future code changes / fixes
    - Unit test code coverage now easier manage understand
- Fully support the Extended Message Set
    - No API changes for existing code
- Uses latest (3/2020) version of G.988
    - New MEs support as well as existing MEs with new attributes
    - Improved decoding in the OMCI parser that generates the code
        - Of 288 ME’s in the G.988 document, all but 35 can be decoded
- Received baseline frame of 40 bytes (no length or CRC) is now valid
    - Line 289+ of ‘omci_cc.go’ has an vendor (adtran simulator) specific work-around. Should not be needed anymore.

Code Coverage

Entire Project: 97.4% of files and 70% of statements

Generated Subdirectory: 98.5% of files and 51.9% of statements

meframe Subdirectory: 80% of files and 55.4% of statements

Main Message Directory:

alarms.go	74.3%
avc.go	86%
create.go	82.5%
delete.go	85.5%
get.go	78.4%
getcurrent.go	69.4%
getnext.go	79.3%
layers.go	100%
messagetypes.go	100%
mibreset.go	76.6%
mibupload.go	77%
omci.go	90.6%
reboot.go	81.2%
set.go	77.3%
settable.go	81.5%
software.go	75.2%
synctime.go	79.3%
test.go	79.9%

37 MEs that are not yet code-generated

401: PPTP RS232/RS485 UNI

402: RS232/RS485 port operation configuration data

403: RS232/RS485 PM history data

404: L2 multicast GEM interworking TP (EPON)

405: ANI-E

23: CES physical interface PM history data

279: Protection data

409: PTM PM history data xDSL

411: Vectoring line configuration extensions

159: Equipment protection profile

417: Data gathering line test, diagnostic and status

166: xDSL line inventory and status data part 3

167: xDSL line inventory and status data part 4

297: Port-mapping package

428: FAST line configuration profile part 1

429: FAST line configuration profile part 2

430: FAST line configuration profile part 3

431: FAST line configuration profile part 4

303: Dot1ag MEP status

304: Dot1ag MEP CCM database

435: FAST line inventory and status data

439: OpenFlow config data

442: TWDM System Profile managed entity

317: RE config portal

318: File transfer controller

319: CES physical interface performance monitoring history data 2

320: CES physical interface performance monitoring history data 3

326: xDSL line inventory and status data part 6

455: Link aggregation service profile

327: xDSL line inventory and status data part 7

330: Generic status portal

331: ONU-E

347: IPv6 host config data

104: xDSL line configuration profile part 1

26 Managed Entities in use*

EthernetFrameExtendedPm
EthernetFrameExtendedPm64Bit
EthernetFramePerformanceMonitoringHistoryDataDownstream
EthernetFramePerformanceMonitoringHistoryDataUpstream
EthernetPerformanceMonitoringHistoryData
ExtendedVlanTaggingOperationConfigurationData
FecPerformanceMonitoringHistoryData
GalEthernetProfile
GemInterworkingTerminationPoint
GemPortNetworkCtp
GemPortNetworkCtpPerformanceMonitoringHistoryData
Ieee8021PMapperServiceProfile
MacBridgePortConfigurationData
MacBridgeServiceProfile
MulticastGemInterworkingTerminationPoint
MulticastOperationsProfile
MulticastSubscriberConfigInfo
Onu2G
OnuG
PhysicalPathTerminationPointEthernetUni
PriorityQueue
TCont
TrafficDescriptor
UniG
VirtualEthernetInterfacePoint
VlanTaggingFilterData

* Open ONU-Go Master branch 2021/08/27

OpenONU Adapter code impact

Extended VLAN Tagging Operation Configuration Data

- Two new attributes in latest standard (new VLAN table). No impact unless you want to support this new table

Multicast Operations Profile

- Previously was hand coded. Now fully code generated.
- Class Name and several attributes are named differently. Same functionality.

New JIRAs

- JIRA xxxx
    - All the updated code with new MEs mentioned before
    - Completed Extended Message Set support
    - Bump version to 2.0.0
    - Cleaned up README.md with much of this presentation’s information provided
- JIRA xxxx
    - Support for relaxed MIB Upload Next response message decode
    - Set ‘global’ value for relaxed handling (gopacket.NewPacket decode call does not have a way to pass down custom decode options such as this.
    - Message with ‘additional/new’ attributes will have an additional layer (derived from ErrorLayer) to communicate the additional attributes and binary data associated with the unknown attributes.

MIB UPLOAD Next example (error checks omitted)

```
RelaxedMIBUploadNextDecode = true
packet := gopacket.NewPacket(rxMsg, omci.LayerTypeOMCI, gopacket.NoCopy)
...
msgLayer := (*msg.OmciPacket).Layer(omci.LayerTypeMibUploadNextResponse)
msgObj, msgOk := msgLayer.(*omci.MibUploadNextResponse)
if msgObj.NextLayerType == omci.LayerTypeMibUploadNextResponseAttrError {
   …  do attr error decode steps if you need to  (TBD)
}
```