

INSTITUTO FEDERAL DE SANTA CATARINA

VINÍCIUS DA LUZ SOUZA

**Ferramenta para Captura e Análise de
Mensagens OMCI em Redes GPON**

São José - SC

Março/2021

FERRAMENTA PARA CAPTURA E ANÁLISE DE MENSAGENS OMCI EM REDES GPON

Trabalho de conclusão de curso apresentado à Coordenadoria do Curso de Engenharia de Telecomunicações do campus São José do Instituto Federal de Santa Catarina para a obtenção do diploma de Engenheiro de Telecomunicações.

Orientador: Marcelo Maia Sobral

Coorientador: Roberto de Matos

São José - SC

Março/2021

Vinícius da Luz Souza

Ferramenta para Captura e Análise de Mensagens OMCI em Redes GPON/
Vinícius da Luz Souza. – São José - SC, Março/2021-

77 p. : il. (algumas color.) ; 30 cm.

Orientador: Marcelo Maia Sobral

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC

Campus São José

Engenharia de Telecomunicações, Março/2021.

1. Redes Ópticas. 2. GPON. 2. OMCI. I. Marcelo Maia Sobral. II. Instituto Federal de Santa Catarina. III. Campus São José. IV. Ferramenta para Captura e Análise de Mensagens OMCI em Redes GPON

VINÍCIUS DA LUZ SOUZA

FERRAMENTA PARA CAPTURA E ANÁLISE DE MENSAGENS OMCI EM REDES GPON

Este trabalho foi julgado adequado para obtenção do título de Engenheiro de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 11 de Março de 2021:

Marcelo Maia Sobral, Dr.
Orientador
Instituto Federal de Santa Catarina

Professor, Dr.
Instituto X

Professor
Instituto Y

Professor
Instituto Z

AGRADECIMENTOS

Às pessoas que estiveram comigo durante os anos de graduação. Meus familiares, que me deram forças para continuar quando precisei. Em especial, à minha namorada, que suportou comigo momentos de angústia e impaciência.

Aos meus amigos, que de uma cerveja em outra, ajudaram a reduzir a carga de conciliar faculdade e trabalho.

À empresa Intelbras, que disponibilizou os equipamentos para realização da monografia e desenvolvimento das atividades. Em especial, aos meus amigos de trabalho, pelo compartilhamento do conhecimento e pelas sugestões para executar da melhor forma possível a entrega deste projeto.

Aos professores do IFSC São José, pela excelência tanto no ensino técnico como no pessoal, que levarei para a vida. Em especial aos professores Marcelo Sobral e Roberto de Matos pelas orientações neste projeto final.

*“Todas as maiores invenções tecnológicas criadas pelo homem,
— o avião, o automóvel, o computador —
dizem pouco sobre sua inteligência,
mas falam bastante sobre sua preguiça.
(Mark Kennedy)*

RESUMO

As redes **GPON** (*Gigabit Passive Optical Network*) são redes de acesso baseadas na utilização de elementos passivos com o intuito de fornecer conexão através da fibra óptica. Graças ao aumento na demanda por altas taxas de transmissão, somado aos benefícios da utilização da fibra óptica e seu notável barateamento nos últimos anos, essa tecnologia vem ganhando mercado. No Brasil, a presença de **PPPs** (Provedores de Pequeno Porte) tem impulsionado ainda mais a utilização deste tipo de rede, aumentando a procura por equipamentos para **GPON**. Entretanto, apesar das normatizações da **ITU-T** (*ITU Telecommunication Standardization Sector*) G.984 e G.988, a incompatibilidade entre fabricantes é notória nessa tecnologia, fazendo com que alguns ativos da rede não se comuniquem adequadamente. Desta forma, este trabalho tem por objetivo desenvolver uma ferramenta que capture e analise mensagens de gerenciamento que trafegam na rede **GPON**, mais especificamente, as mensagens **OMCI** (*Optical network termination Management and Control Interface*). Esta ferramenta poderá auxiliar na análise da interoperabilidade de equipamentos, orientando engenheiros e analistas no desenvolvimento de produtos que utilizem esta tecnologia.

Palavras-chave: Redes Ópticas. GPON. OMCI.

ABSTRACT

GPON (*Gigabit Passive Optical Network*) are access networks based on the use of passive elements in order to provide connection through optical fiber. Thanks to the increase in demand for high transmission rates, added to the benefits of using optical fiber and its notable cheapness in recent years, this technology has been gaining market share. In Brazil, the presence of small internet providers has further boosted the use of this type of network, increasing the demand for equipment for **GPON**. However, despite the regulations of **ITU-T** G.984 and G.988, the incompatibility of the manufacturers is notorious in this technology, causing some network assets to not communicate properly. Thus, this work aims to develop a tool that captures and analyzes management messages that travel on the **GPON** network, more specifically, the **OMCI** (*Optical network termination Management and Control Interface*) messages. This tool can assist in the analysis of interoperability between equipment, guiding engineers and analysts in the development of products that use this technology.

Keywords: Optical Networks. GPON. OMCI.

LISTA DE ILUSTRAÇÕES

Figura 1 – Topologia de Rede de Telecomunicações Moderna	22
Figura 2 – Cenários FTTx	24
Figura 3 – Relação entre ONU, T-CONT e GEM	28
Figura 4 – Quadro <i>downstream</i> GTC	28
Figura 5 – Quadro <i>Upstream Bandwidth Map</i>	30
Figura 6 – Quadro GEM	30
Figura 7 – Quadro <i>Upstream</i>	31
Figura 8 – Quadro PLOAM	31
Figura 9 – Quadro OMCI	33
Figura 10 – Exemplo <i>MAC Bridged LAN</i> OMCI	37
Figura 11 – Cenário de captura da ferramenta	38
Figura 12 – ONU 110 - Utilizada como capturador	39
Figura 13 – Fluxo das mensagens OMCI no capturador	41
Figura 14 – Diagrama geral do <i>software</i> analisador	43
Figura 15 – Trecho de imagem gerada pelo <i>software</i> analisador	45
Figura 16 – Visão geral do <i>software</i> analisador	48
Figura 17 – <i>Software</i> analisador executado em modo contínuo	49
Figura 18 – Ajuda do analisador executado em modo <i>OneShot</i>	50
Figura 19 – Exemplo de execução do <i>software</i> em modo <i>OneShot</i>	51
Figura 20 – Visão geral da ferramenta	52
Figura 21 – Testes com OLTs diferentes - Configuração 1	56
Figura 22 – Testes com OLTs diferentes - Configuração 2	56
Figura 23 – Testes com OLTs diferentes - Configuração 3	57
Figura 24 – Testes com OLTs diferentes - Configuração 4	57
Figura 25 – Cenário testes com OLTs diferentes	57
Figura 26 – Teste OLT 8820i com configuração básica	58
Figura 27 – Relatório verificação teste OLT 8820i com configuração básica	60
Figura 28 – Cenário testes com mensagens OMCI manipuladas	62
Figura 29 – Referencia analisador cenário VoIP	62
Figura 30 – Comparativo relatório verificador antes e depois das alterações na OLT	64
Figura 31 – Resultado analisador após segunda e terceira modificações na OLT	65
Figura 32 – Cenário testes comparativos	66
Figura 33 – Teste OLT 8820i com configuração básica e opção <i>show-zero-pointers</i>	69
Figura 34 – Diagrama de classes do software analisador	77

LISTA DE TABELAS

Tabela 1 – Características gerais GPON	26
Tabela 2 – Exemplo - ME (<i>Managed Entity</i>) <i>IP host config data</i> (134)	35
Tabela 3 – Exemplo - ME <i>TCP/UDP config data</i> (136)	36

LISTA DE ABREVIATURAS E SIGLAS

Alloc-ID *Allocation Identifier*

APON *ATM Passive Optical Network*

ATM *Asynchronous Transfer Mode*

BER *Bit Error Ratio*

BIP *Bit Interleaved Parity*

BPON *Broadband Passive Optical Network*

BW *Bandwidth*

CPU *Central Process Unit*

CRC *Cyclic Redundancy Check*

DBA *Dynamic Bandwidth Assignment*

DHCP *Dynamic Host Configuration Protocol*

DNS *Domain Name System*

DSL *Digital Subscriber Line*

DS3 *Digital Signal 3*

E1 *E-Carrier 1*

E3 *E-Carrier 3*

EPON *Ethernet Passive Optical Network*

FEC *Forward Error Correction*

FPGA *Field Programmable Gate Array*

FSAN *Full Service Access Network*

FTTB *Fiber To The Building*

FTTC *Fiber To The Curb/Cabinet*

FTTH *Fiber To The Home*

FTTN *Fiber To The Neighborhood*

FTTP *Fiber To The Premises*

FTTx *Fiber To The X*

FXS *Foreign eXchange Subscriber*

GEM *GPON Encapsulation Method*

GPON *Gigabit Passive Optical Network*

GTC *GPON Transmission Convergence*

HEC *Header Error Control*

HFC *Hybrid Fiber Coax*

ID *Identifier*

IP *Internet Protocol*

IPv4 *Internet Protocol version 4*

IPv6 *Internet Protocol version 6*

ISP *Internet Service Provider*

ITU *International Telecommunication Union*

ITU-T *ITU Telecommunication Standardization Sector*

LAN *Local Area Network*

MAC *Media Access Control address*

ME *Managed Entity*

MIB *Management Information Base*

MPLS *Multi-Protocol Label Switching*

OAM *Operation, Administration and Maintenance*

ODN *Optical Distribution Network*

OLT *Optical Line Terminal*

OMCC *Optical network termination Management and Control Channel*

OMCI *Optical network termination Management and Control Interface*

ONT *Optical Network Termination*

ONU-ID *Optical Network Unit Identifier*

ONU *Optical Network Unit*

P2P *Peer-to-Peer*

PCBd *Physical Control Block downstream*

PCB *Physical Control Block*

PC *Personal Computer*

PLend *Payload Length Downstream*

PLI *Payload Length Indicator*

PLOAMd *Physical Layer Operations, Administrations and Maintenance downstream*

PLOAMu *Physical Layer Operations, Administrations and Maintenance upstream*

PLOAM *Physical Layer Operations, Administrations and Maintenance*

PMD *Physical Media Dependent*

PoE *Power over Ethernet*

PON *Passive Optical Network*

PPP *Provedores de Pequeno Porte*

PSync *Physical Synchronization*

PTI *Payload Type Indicator*

QoS *Quality of Service*

SDH *Synchronous Digital Hierarchy*

SDK *Software Development Kit*

SIP *Session Initiation Protocol*

SO *Sistema Operacional*

SoC *System on Chip*

T1 *T-Carrier 1*

T-CONT *Transmission Container*

TCP *Transmission Control Protocol*

TDM *Time Division Multiplexing*

TOS *Type Of Service*

UDP *User Datagram Protocol*

UNI *User-Network Interface*

UPON *Universal PON*

US *Upstream*

VEIP *Virtual Ethernet Interface Point*

VLAN *Virtual Local Area Network*

VoIP *Voice Over IP*

WAN *Wide Area Network*

xDSL *Digital Subscriber Line*

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Objetivos	20
1.2	Organização	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Redes PON	21
2.1.1	Elementos de redes PON	22
2.1.2	Arquiteturas FTTx	24
2.2	GPON	25
2.2.1	Enquadramento GPON	27
2.2.2	PLOAM	31
2.2.3	OMCI	32
3	ANALISADOR OMCI	38
3.1	Capturador OMCI	38
3.1.1	Identificação e encaminhamento das mensagens OMCI no capturador	40
3.1.2	Configuração da ONU a ser monitorada	41
3.2	Software Analisador	42
3.2.1	Módulo Receptor	43
3.2.2	Módulo Analisador	44
3.3	Módulo Verificador	46
3.3.1	Verificação de atributos obrigatórios	46
3.3.2	Verificação de tamanho das MEs	47
3.3.3	Verificação de configuração sem permissão	47
3.4	Visão geral do software Analisador	48
3.5	Modo de utilização do software desenvolvido	48
3.6	Visão geral da ferramenta	51
4	TESTES E RESULTADOS	53
4.1	Testes e objetivos	53
4.2	Testes com OLTs diferentes	55
4.2.1	Cenário	55
4.2.2	Resultados	58
4.3	Testes com mensagens OMCI manipuladas	61
4.3.1	Cenário	61
4.3.2	Resultados	63

4.4	Testes comparativos	65
4.4.1	Cenário	65
4.4.2	Resultados	67
5	CONCLUSÕES	70
5.1	Trabalhos Futuros	71
5.1.1	No capturador	71
5.1.2	No <i>software</i> analisador	72
	REFERÊNCIAS	74
	APÊNDICES	76
	APÊNDICE A – DIAGRAMA DE CLASSES	77

1 INTRODUÇÃO

O acesso a Internet banda larga vem crescendo ano após ano ao redor do mundo, impondo um crescimento médio anual de quase 13% (ANATEL, 2019). No Brasil o crescimento acontece na mesma proporção, colocando o país na sexta posição, entre os dez com maior número de assinantes de banda larga fixa no mundo (ANATEL, 2019). Os provedores ou prestadoras de pequeno porte (PPPs), juntos, somam 27% do *market share* do mercado de banda larga fixa no Brasil (ABRANET, 2019). Nesses provedores, de modo a oferecer um serviço diferenciado, é comum o fornecimento do serviço de banda larga com acesso via fibra óptica até a casa do assinante, o chamado FTTH. Em função do avanço no uso das redes via fibra óptica, os acessos à banda larga realizados através deste meio já ultrapassaram os realizados através de tecnologias xDSL (*Digital Subscriber Line*), que dominavam o mercado há pouco tempo atrás (ABRANET, 2020). Apesar do grande mercado já existente e do forte crescimento previsto, a maior parte das tecnologias de acesso utilizadas para fornecer conexão via fibra óptica no Brasil são importadas, de modo que poucos desses produtos são desenvolvidos em território brasileiro.

A tecnologia GPON (*Gigabit Passive Optical Network*) oferece uma rede de acesso através de componentes passivos capazes de prover velocidades na ordem de *Gigabits* por segundo. Nessa tecnologia, o equipamento mestre, denominado OLT (*Optical Line Terminal*), faz o gerenciamento dos equipamentos escravos, as ONUs (*Optical Network Unit*) ou ONTs (*Optical Network Termination*), por meio do protocolo OMCI (*Optical network termination Management and Control Interface*). O protocolo OMCI é definido pela ITU (*International Telecommunication Union*) e especifica recursos nas ONUs/ONTs que podem ser gerenciados pela OLT. Estes recursos são representados em forma de entidades, que se relacionam entre si para permitir o gerenciamento das ONUs/ONTs. Além das configurações da rede GPON, o OMCI permite a aplicação de configurações de rede *ethernet*, como configurações de VLAN (*Virtual Local Area Network*), IP (*Internet Protocol*) e até configurações de serviços, como o IEEE 802.1X (IEEE, 2020). É por meio deste protocolo que a OLT define o modo de funcionamento de cada ONU/ONT na rede.

Apesar da especificação definida pela ITU, são percebidas divergências na comunicação entre equipamentos de diferentes fabricantes, de modo que se torna comum equipamentos escravos (ONUs ou ONTs) de determinado fabricante não serem compatíveis com equipamentos mestre (OLTs) de outro fabricante. Essa incompatibilidade reside justamente nas trocas de mensagens OMCI, fazendo com que as configurações não sejam aplicadas corretamente nas ONUs/ONTs. Elas podem ser observadas no relacionamento entre as entidades OMCI ou até mesmo no conteúdo das mensagens, impossibilitando

a configuração da rede e impactando diretamente em seu funcionamento. Como alguns exemplos de tais divergências, é possível citar algumas configurações, como: configuração de um determinado fluxo de dados, rotas estáticas ou até mesmo configuração de uma conta **VoIP** (*Voice Over IP*) em uma **ONU/ONT**. Essas incompatibilidades limitam a utilização de equipamentos de diferentes fabricantes, fazendo com que o cliente seja, por vezes, forçado a utilizar os dispositivos de um mesmo fornecedor, principalmente em cenários onde já existe uma infraestrutura em execução.

Visando reduzir os impactos da incompatibilidade, parte do trabalho de desenvolvedores de equipamentos **GPON** é realizar a compatibilização dos dispositivos que compõem a rede. Um exemplo deste tipo de trabalho pode ser dado por um fabricante de **ONU/ONT** que deseja ter seu equipamento compatível com uma **OLT** de outro fabricante. Ele terá de verificar como são estruturadas as mensagens desta **OLT** para adaptar o *firmware* de seu equipamento, de modo que os torne compatíveis. Para realizar tal verificação, são necessários dispositivos com capacidade de análise em redes **GPON**. Entretanto, o alto custo e a baixa oferta destes equipamentos tornam o trabalho de compatibilização caro e, por vezes, moroso. Isso porque o desenvolvedor pode ter de buscar formas alternativas para fazê-lo, por não dispor de tais recursos. Sendo assim, tendo em vista o rápido crescimento desse mercado, tais empecilhos podem ser fortemente prejudiciais, implicando até numa possível inviabilização do projeto.

Considerando o cenário emergente do acesso à banda larga, o alto crescimento das redes de acesso **FTTH** e as dificuldades de compatibilidade entre os equipamentos de mercado, este trabalho teve por objetivo desenvolver uma ferramenta de captura e análise do fluxo de mensagens do protocolo **OMCI**, usado para fins de gerenciamento entre os equipamentos **OLT** (mestre) e **ONU/ONT** (escravo) de uma rede **GPON**. A ideia da ferramenta proposta foi capturar as mensagens **OMCI** que trafegam na rede e analisá-las. A partir da captura e da análise realizada, a ferramenta proporciona uma melhor visualização das mensagens trocadas entre os equipamentos, além de indicar possíveis falhas na comunicação entre eles, facilitando a depuração de problemas ou incompatibilidades entre os dispositivos que compõem a rede. Sendo assim, a ferramenta pode auxiliar em atividades de desenvolvimento de dispositivos **GPON**.

Neste trabalho foi desenvolvida uma ferramenta de análise de mensagens **OMCI** composta por dois módulos, denominados capturador e analisador. O módulo capturador foi desenvolvido modificando o *firmware* de uma **ONU** do fabricante Intelbras, cujo objetivo era capturar as mensagens **OMCI** que trafegam na rede **GPON** e entregá-las ao módulo analisador. O módulo analisador, por sua vez, foi composto por um *software* executado em um computador, cujo objetivo foi processar as mensagens **OMCI** e apresentá-las ao usuário da ferramenta. A apresentação destas mensagens foi realizada a partir de diagramas, que mostram a interação das entidades **OMCI** nas mensagens enviadas pela

OLT. Além da apresentação por meio de diagramas, o *software* de análise foi responsável por executar verificações nas mensagens capturadas. O resultado destas verificações foi disponibilizado por meio de relatórios, que apontam falhas que podem indicar a origem de incompatibilidades entre os dispositivos que compõem a rede **GPON**. Desta forma, a ferramenta pode auxiliar na depuração de problemas relacionados com a comunicação entre **OLT** e **ONU/ONT**.

1.1 Objetivos

Este trabalho teve como objetivo geral desenvolver uma ferramenta de captura e análise do fluxo de mensagens do protocolo **OMCI**. Ela tem a finalidade de auxiliar a análise da interoperabilidade entre **ONUs** e **ONTs** de diferentes fabricantes, estabelecendo também procedimentos de verificação de conformidade com o protocolo **OMCI**. Como objetivos específicos, é possível destacar:

- **Criar um dispositivo de captura **GPON** baseado em uma **ONU**:** para possibilitar a captura das mensagens **OMCI**;
- **Desenvolver um software para análise das interações entre as entidades **OMCI**:** para possibilitar uma visualização do relacionamento entre as entidades;
- **Estabelecer um procedimento de verificação de conformidade nas mensagens enviadas pela **OLT**, com base nas especificações **GPON**:** para indicar possíveis causas de incompatibilidade entre **OLT** e **ONU/ONT**.

1.2 Organização

O capítulo 2 descreve os principais conceitos de uma rede **PON** (*Passive Optical Network*), além de apresentar os elementos e as possíveis arquiteturas deste tipo de rede. Ainda neste capítulo, é descrito o funcionamento da tecnologia **GPON**, indicando detalhes sobre seu funcionamento, incluindo o modo de enquadramento dos dados e suas formas de gerenciamento, realizadas por meio do protocolo **OMCI**. O capítulo 3 descreve as etapas de desenvolvimento da ferramenta proposta, bem como as estratégias e ações adotadas em cada caso, até chegar na solução final. O capítulo 4 tem por objetivo descrever os testes utilizados para validar a ferramenta desenvolvida, bem como os resultados obtidos. Por fim, o capítulo 5 traz as conclusões obtidas a partir desta monografia e indicações de possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é realizada a fundamentação teórica sobre os conteúdos utilizados no desenvolvimento da monografia. Nele foram descritos conceitos de redes **PON**, bem como seus elementos e as arquiteturas utilizadas. Após tais definições, foram descritos os princípios e terminologias das redes **GPON**, indicando detalhes sobre o enquadramento **GPON** e as formas de gerenciamento neste tipo de rede.

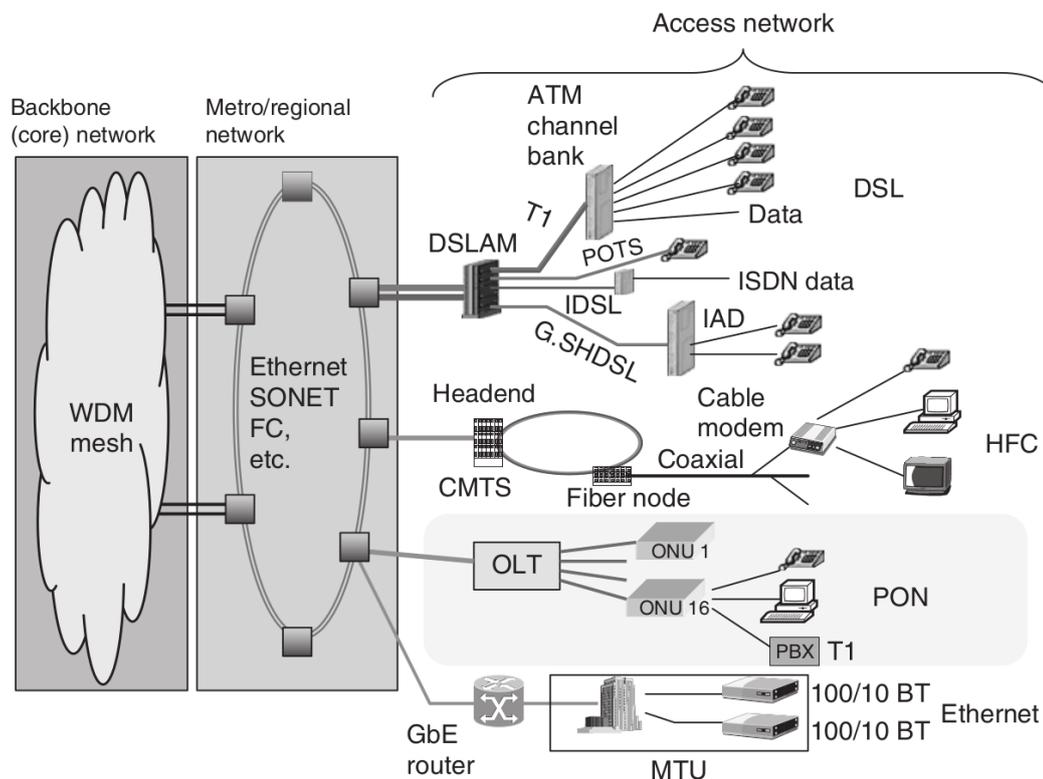
2.1 Redes PON

Arquiteturas modernas de redes de telecomunicações, como por exemplo, a Internet, se baseiam em três divisões principais em sua topologia: redes *backbone* (ou rede de núcleo, ou rede *core*), redes metro e redes de acesso. Cada um desses três tipos de rede possui sua atribuição e posicionamento dentro da topologia. No nível mais alto, estão as redes *backbone*, que são responsáveis pelo transporte em longas distâncias como, por exemplo, enlaces nacionais e internacionais. Num nível intermediário, encontram-se as redes metro, que são responsáveis pela organização e roteamento em nível local ou regional. Já na linha de frente, ou seja, na borda da topologia, estão as redes de acesso (CEDRIC, 2007). Elas são responsáveis por fornecer acesso ao usuário final, garantindo sua conexão à rede, podendo ser de origem corporativa ou, conforme utiliza-se nesta monografia, a Internet.

Conforme descrito na Figura 1, cada tipo de rede possui suas respectivas tecnologias para transporte de dados, as quais visam atender as demandas inerentes de cada camada. Seguindo essa lógica, existem algumas tecnologias pelas quais um **ISP** (*Internet Service Provider*) pode fornecer acesso a seus usuários, que podem variar de acordo com a necessidade de banda do assinante ou levando em consideração outros sistemas que já estão em uso. Segundo KUROSE e ROSS (2013), os dois tipos de tecnologias de acesso predominantes são o acesso à cabo e o **DSL** (*Digital Subscriber Line*). Provedores que fornecem o acesso via cabo, através do **HFC** (*Hybrid Fiber Coax*), normalmente utilizam essa tecnologia pois já possuem uma infraestrutura para fornecimento de TV a cabo, assim como provedores que fornecem acesso através do **DSL** utilizam a infraestrutura já existente de telefonia para fornecer o acesso à rede.

A tecnologia **PON** é uma rede de acesso que utiliza a fibra óptica como meio físico. Apesar de ser uma tecnologia já existente desde a década de 90, ela ganhou popularidade nos últimos anos devido ao aumento na demanda por bandas de dados cada vez mais largas, necessidade de serviços *triple play* (dados, voz e vídeo) e queda nos custos dos equipamentos e meios ópticos (CEDRIC, 2007). Além disso, por utilizar somente elementos

Figura 1 – Topologia de Rede de Telecomunicações Moderna



FONTE: CEDRIC (2007)

passivos na distribuição, seu custo de manutenção é baixo em relação a outras redes que necessitam de elementos ativos para extensão. Outra característica importante das redes PON diz respeito ao uso da fibra óptica, pois este meio não sofre com interferências eletromagnéticas, facilitando sua utilização e distribuição (KUROSE; ROSS, 2013).

2.1.1 Elementos de redes PON

Apesar da indicação de rede passiva no nome PON, a utilização de elementos ativos é necessária para a transmissão e recepção dos sinais ópticos. Os componentes ativos da tecnologia PON estão localizados nas bordas da rede e são os denominados OLT (*Optical Line Terminal*) e ONT (*Optical Network Termination*) ou ONU (*Optical Network Unit*).

A OLT é o equipamento que normalmente está localizado no ponto de distribuição do fornecedor da rede. Ela é responsável pelo controle do tráfego nas direções *downstream* (da OLT para a ONT) e *upstream* (da ONT para a OLT). Para possibilitar o tráfego bidirecional em uma única fibra, diferentes comprimentos de onda são utilizados para cada um dos sentidos. No sentido *downstream*, a OLT envia dados e voz no comprimento de onda de 1490 nanômetros e vídeo no comprimento de 1550 nanômetros. No sentido contrário, a ONT envia dados e voz através do comprimento de onda de 1310 nanômetros. Uma OLT é geralmente desenvolvida com mais de uma porta PON, sendo que cada porta

pode conter até 32 **ONTs**, de acordo com sua definição (veremos que houve expansões de acordo com a evolução dos protocolos **PON**). Cada porta **PON** deve ter capacidade para suportar clientes a uma distância de até 20 quilômetros e, dependendo do protocolo utilizado na rede, as taxas de dados podem variar de 155 Mbps até 2.5 Gbps. Essas taxas normalmente são assimétricas, ou seja, a banda designada para *downstream* não é a mesma designada para *upstream*, porém o modo simétrico também pode ser utilizado (KEISER, 2006).

Os equipamentos **ONU** e **ONT** são os dispositivos que estão localizados no fim da rede óptica. Possuem uma interface em seu lado *upstream* que faz a conexão com a rede **ODN** (*Optical Distribution Network*) e podem possuir diferentes interfaces de conexão no seu lado *downstream*. De acordo com Keiser (2006), em seu lado *downstream*, as **ONTs** possuem interfaces elétricas que fazem a conexão diretamente com os equipamentos do cliente, podendo conter interfaces *Ethernet* de diferentes velocidades, interfaces **T1** (*T-Carrier 1*) ou **E1** (*E-Carrier 1*), **DS3** (*Digital Signal 3*) ou **E3** (*E-Carrier 3*), conexões telefônicas, interfaces **ATM** (*Asynchronous Transfer Mode*) e saídas de vídeo digital e analógico. Ainda de acordo com sua definição, a **ONU** é um dispositivo que normalmente é colocado em ambientes externos, localizados, por exemplo, na calçada das ruas. Elas devem ser resistentes a variações de temperatura e utilizar banco de baterias para que não tenham seu funcionamento interrompido. Além disso, em seu lado *downstream*, possuem interfaces que fazem a conexão com o cliente através de cabos de par trançado, cabo coaxial, podem possuir um outro link óptico independente ou até mesmo uma conexão sem fio. Já de acordo com CEDRIC (2007), o uso dos termos **ONU** e **ONT** não segue uma regra clara nos padrões da série G.983. Nesta monografia, a menos que seja indicado, não há distinção entre esses dois equipamentos.

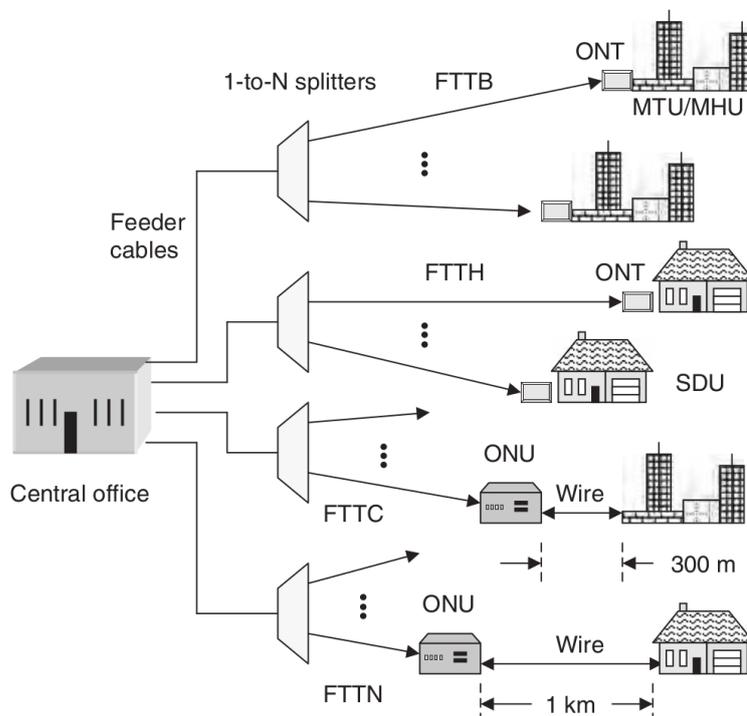
Os elementos passivos presentes em uma rede **PON** são os cabos ópticos, os *splitters* ópticos e os acopladores de comprimento de onda. Os cabos ópticos tem a função de guiar o sinal até o seu destino e são basicamente divididos em cabos multimodo e cabos monomodo. Tradicionalmente, os cabos monomodo são utilizados para longas distâncias, enquanto que cabos multimodo são mais utilizados para conexões locais. Os *splitters* ópticos são responsáveis pelas divisões do sinal óptico, podendo variar de 1 entrada para 2 saídas até 1 entrada e 128 saídas, sendo as os valores mais comuns de 1:8, 1:16 e 1:32. Para cada divisão, há um valor de perda característico. Por fim, os acopladores de comprimento de onda tem a função de refletir ou permitir a passagem de determinados comprimentos de onda. Eles funcionam como um filtro tanto para o lado da **OLT**, quanto para o lado da **ONU** (KEISER, 2006).

2.1.2 Arquiteturas FTTx

As arquiteturas de redes PON possibilitam a utilização em cenários ponto-a-ponto e cenários ponto-multiponto. Segundo CEDRIC (2007), os cenários ponto-a-ponto foram adotados apenas em alguns testes iniciais. Neste caso, uma quantidade maior de fibras e transceptores são necessários em ambos os lados da comunicação. Em contrapartida, as arquiteturas ponto-multiponto se baseiam no compartilhamento de uma mesma fibra, de modo a atender diversos assinantes. Apesar de essas arquiteturas disponibilizarem taxas menores para transmissão de dados, já que a banda é compartilhada, sua utilização é adequada para o cenários dos ISPs. No caso dos provedores, este cenário torna o investimento na compra de fibras ópticas bem mais baixo, já que não é necessária uma nova fibra para cada novo cliente que deseja ter acesso a rede.

O fornecimento de enlaces de rede utilizando tecnologias PON é normalmente chamado de FTTx. A sigla é proveniente do inglês *Fiber-To-The-x*, que podemos traduzir por fibra-até-o-x. Onde "x" indica o ponto terminal da fibra. Dessa notação surgem as nomenclaturas FTTH, FTTB, FTTC, FTTN, entre outros. Na figura a seguir, é considerada a presença de uma OLT no escritório central do provedor da rede.

Figura 2 – Cenários FTTx



FONTE: Keiser (2006)

Na Figura 2 são apresentados os principais cenários FTTx (*Fiber To The X*). A descrição de cada um segue a seguir:

- **FTTB** (*Fiber To The Building*): cenário onde a fibra sai do ponto de distribuição do

ISP e é levada até a construção, ou seja, um prédio, por exemplo. A partir daí, é utilizado por trançado ou cabo coaxial até o local do assinante;

- **FTTH** (*Fiber To The Home*): cenário em que a fibra sai do ponto de distribuição do ISP e é entregue diretamente na casa do assinante. Neste caso, há a possibilidade de utilizar uma **ONT** com conexão *Wireless* e portas de telefonia, por exemplo, para fornecimento ao usuário;
- **FTTC** (*Fiber To The Curb/Cabinet*): é possível traduzir "*curb*" como "borda de calçada". Sendo assim, neste cenário, a fibra que sai do ISP é levada até cerca de 300 metros de proximidade da casa do assinante, de onde a conexão é normalmente realizada através de cabo coaxial ou par trançado;
- **FTTN** (*Fiber To The Neighborhood*): semelhante ao **FTTC**, porém a fibra é entregue a nível de bairro e, desta forma, está mais distante do usuário final, a cerca de 1000 metros.

Além da definição de cada cenário, um termo mais genérico é utilizado para englobar todos os casos citados, cuja sigla é definida por **FTTP** (*Fiber To The Premises*) (KEISER, 2006).

2.2 GPON

Gigabit Passive Optical Network é um padrão definido pela ITU através da família G.984, que estabelece normativas para o funcionamento de redes *gigabit* através de uma infraestrutura de rede óptica passiva (**PON**). De acordo com Keiser (2006), o desenvolvimento deste padrão foi impulsionado em Abril de 2001 por meio do grupo **FSAN** (*Full Service Access Network*). Este grupo de grandes provedores, fundado nos anos 90, tinha como objetivo a padronização de uma rede de acesso óptica passiva, visando o barateamento dos custos de uma rede **PON** através da produção em escala, utilizando um padrão comum. As padronizações do **FSAN** posteriormente foram incorporadas pelo ITU na recomendação ITU-T G.983, denominado **APON** (*ATM Passive Optical Network*), em Outubro de 1998. O nome **APON** foi utilizado justamente pelo fato de a padronização ter sido baseada nas redes **ATM**. Por uma justificativa meramente comercial, essa padronização também pode ser chamada de **BPON** (*Broadband Passive Optical Network*) (CEDRIC, 2007).

Apesar da expectativa, as redes **ATM** não se tornaram o padrão universal de transporte para as diversas aplicações distribuídas. Por outro lado, a popularidade das redes Ethernet e a necessidade crescente da demanda por maiores bandas de transmissão em redes de acesso, impulsionaram o desenvolvimento do padrão **GPON** (CEDRIC, 2007). Com seu desenvolvimento baseado na recomendação G.983 (**BPON**), algumas de suas

características foram preservadas, além da principal novidade: a adição do suporte a redes Ethernet.

A família [ITU-T G.984](#) possui quatro especificações, partindo da [ITU-T G.984.1](#) até [ITU-T G.984.4](#). A primeira especificação, [ITU-T G.984.1](#), fornece uma visão geral do protocolo, indicando suas principais características e componentes. Algumas dessas características estão expostas na [Tabela 1](#).

Tabela 1 – Características gerais GPON

Parâmetro	Descrição	Valor
Bit Rate	Taxa de bits	1.2 Gbit/s UP, 2.4 Gbit/s DOWN 2.4 Gbit/s UP, 1.2 Gbit/s DOWN
Logical Reach	Distância entre ONU/ONT e OLT, desconsiderando as limitações da camada física	60 km
Physical Reach	Distância entre ONU/ONT e OLT, com base nas limitações da camada física	10 km e 20 km
Differential Fibre Distance	Distância entre ONU/ONTs na fibra	20 km
Split Ratio	Taxa de divisão em uma única porta PON (clientes por PON)	Até 1:128
Differential Fibre Distance	Distância entre ONU/ONTs na fibra	20 km
UNI Service Examples	Exemplos de serviços oferecidos na porta da ONU/ONT	Ethernet, POTS, ISDN (BRI, PRI), T1, E1, DS3, E3, Digital Vídeo

Fonte – [ITU-T \(2008a\)](#)

A especificação [ITU-T G.984.2](#), [PMD](#) (*Physical Media Dependent*), traz informações físicas a respeito do funcionamento do protocolo [GPON](#), tais como: conversões óptico-elétricas e elétrico-ópticas, sincronismo de clock e mecanismos de correção de erro utilizando código corretor - [FEC](#) (*Forward Error Correction*). Além disso, também são informados detalhes a respeito de tempo de guarda entre pacotes e controle de potência nos transmissores das [ONUs/ONTs](#), que são ajustados automaticamente conforme a posição da [ONUs/ONTs](#) (mais próxima ou mais distante da [OLT](#)) ([ITU-T, 2003](#)).

A recomendação [ITU-T G.984.3](#), [GTC](#) (*GPON Transmission Convergence*) tem como objetivo principal determinar o funcionamento do transporte de quadros entre [OLT](#) e [ONU](#) e vice-versa. Ela especifica o formato do quadro [GPON](#), o funcionamento do *ranging method*, as normas de segurança e os procedimentos [OAM](#) (*Operation, Administration and Maintenance*) ([ITU-T, 2014](#)). Devido à importância do entendimento dos quadros [GPON](#) para a leitura desta monografia, a [subseção 2.2.1](#) apresenta os detalhes desta recomendação.

Por fim, a especificação ITU-T G.984.4, OMCI (*Optical network termination Management and Control Interface*), descreve um protocolo para gerenciamento e controle das ONUs e ONTs presentes na rede, com o intuito de possibilitar a interoperabilidade entre OLT e ONU de diferentes fabricantes. O funcionamento dessa interoperabilidade é o objeto principal pelo qual esta monografia foi proposta e a compreensão dessa especificação possui grande importância para o decorrer da leitura, por este motivo a subseção 2.2.3 foi descrita com os detalhes dessa recomendação (ITU-T, 2008b).

2.2.1 Enquadramento GPON

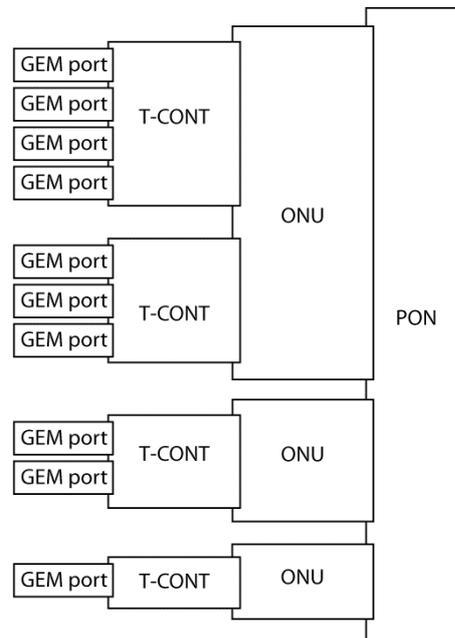
A especificação ITU-T G.984.3 (ITU-T, 2014) define como as funções da camada GTC são realizadas. Para realizá-las, é definido um meio de comunicação entre OLT e ONU. Neste contexto, são especificados os contêineres de transmissão, denominados T-CONTs (*Transmission Containers*). Os T-CONTs são definidos pela OLT por um ID (*Identifier*) de alocação, chamado de Alloc-ID (*Allocation Identifier*). Ou seja, um Alloc-ID se refere a alocação de um contêiner de transmissão (T-CONT). A OLT pode fornecer um ou mais T-CONTs para determinada ONU, sendo que cada T-CONT pode conter um ou mais GEMs (*GPON Encapsulation Method*). O GEM é quem irá encapsular os diferentes protocolos como o *Ethernet*, o SDH (*Synchronous Digital Hierarchy*), o IP (*Internet Protocol*), o TDM (*Time Division Multiplexing*), e o MPLS (*Multi-Protocol Label Switching*). A Figura 3 mostra como é realizada essa divisão. Na imagem, é exibida a visão da porta PON de uma OLT. Uma porta PON pode conter várias ONUs. Cada ONU pode conter um ou mais T-CONTs que, por sua vez, poderá ter um ou mais GEMs. Neste caso, conforme mencionado anteriormente, o GEM é quem fará o encapsulamento dos protocolos Ethernet, SDH, IP, etc.

A definição dos contêineres de transmissão (T-CONTs) possibilita a alocação dinâmica de banda na fibra por meio do DBA (*Dynamic Bandwidth Assignment*), realizado pela OLT. Existem diferentes tipos de T-CONTs, que possuem diferentes prioridades entre si, permitindo a reserva ou garantia de banda para os fluxos mais sensíveis, como voz e vídeo. Além disso, a OLT também pode fornecer funcionalidades de qualidade de serviço, denominado QoS (*Quality of Service*).

Inicialmente, além dos T-CONTs baseados em GEM, o padrão GPON previa também a utilização de T-CONTs baseados nos enlaces virtuais utilizados em redes ATM. Porém, devido a baixa ou nenhuma utilização desse recurso, a especificação desse método foi removida da versão de 01/2014 da norma ITU-T G.984.4, a qual indica que o método foi depreciado (ITU-T, 2008b). Por esse motivo, esta monografia está fundamentada somente nos T-CONTs baseados em GEM.

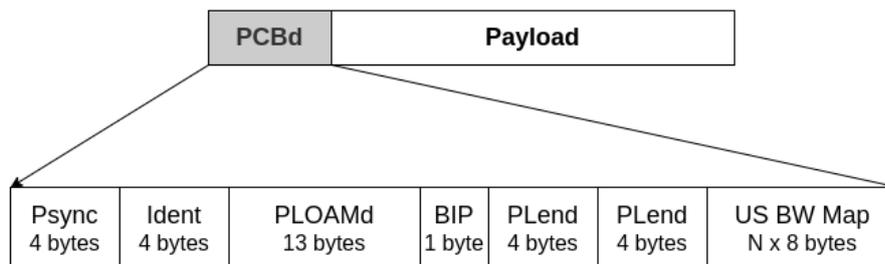
A transmissão no sentido *downstream*, ou seja, da OLT para a ONU é realizada por um quadro com tamanho fixo de $125\mu\text{s}$, dividido por um bloco denominado PCB

Figura 3 – Relação entre ONU, T-CONT e GEM



FONTE: Elaborada pelo autor.

(*Physical Control Block*) e um *GTC Payload*, conforme é mostrado na Figura 4. Considera-se inicialmente o bloco *PCBd* (*Physical Control Block downstream*), que possui as seguintes divisões:

Figura 4 – Quadro *downstream* GTC

FONTE: ITU-T (2014)

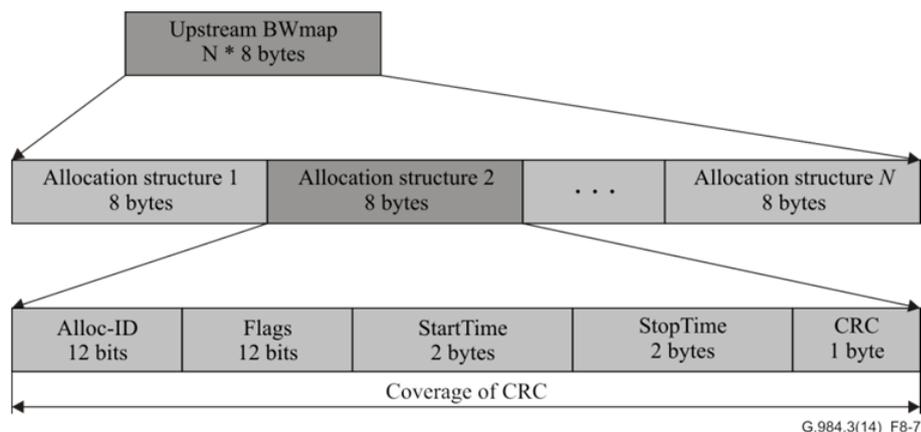
- **PSync** (*Physical Synchronization*) [4 bytes]: Possui um valor fixo para sincronização, ou seja, para a ONU encontrar o início do quadro. Este valor é: 0xB6AB31E0;
- **Ident** [4 bytes]: é subdividido em:
 - **FEC Ind** [1 bit]: indica se o **FEC** está habilitado no sentido *downstream*;
 - **Reserved** [1 bit]: reservado para uso futuro;
 - **Superframe Counter** [30 bits]: um contador do super quadro utilizado para sincronismo.

- **PLOAMd** (*Physical Layer Operations, Administrations and Maintenance downstream*) [13 bytes]: contém mensagens de **PLOAM** (*Physical Layer Operations, Administrations and Maintenance*), que possuem a funcionalidade de realizar procedimentos de operação, administração e manutenção em camada física;
- **BIP** (*Bit Interleaved Parity*) [1 byte]: Bits intercalados de paridade, utilizados para calcular a o número de erros no enlace;
- **PLend** (*Payload Length Downstream*) [4 bytes]: Especifica o tamanho do próximo campo (**BW Map**) e o tamanho da parte **ATM** do *payload* (mesmo com a parte **ATM** depreciada, esse trecho ainda é enviado). Esse campo é enviado duas vezes para proteção contra erros;
- **US** (*Upstream*) **BW** (*Bandwidth*) **Map** [$N \times 8$ bytes]: A **Figura 5** mostra as subdivisões deste quadro. Neste campo a **OLT** informa a **ONU** como devem ser enviados os quadros no sentido *upstream*, de acordo com os **T-CONTs** que a **ONU** possui. N indica o **T-CONT** e os 8 *bytes* estão divididos da seguinte forma:
 - **Alloc-ID** [12 bits]: indica o respectivo **Alloc-ID**, ou seja, o **ID** do **T-CONT** em questão;
 - **Flags** [12 bits]: algumas *flags* que indicam se determinada informação deve ser encaminhada no sentido *upstream* ou não;
 - **Start Time** [2 bytes]: momento (tempo) em que a **ONU** deve iniciar a transmissão deste **T-CONT**;
 - **Stop Time** [2 bytes]: momento (tempo) em que a **ONU** deve finalizar a transmissão deste **T-CONT** (*Transmission Container*);
 - **CRC** (*Cyclic Redundancy Check*) [1 byte]: cálculo CRC-8 conforme **ITU-T I.432.1** (**ITU-T, 1999**).

Voltando a **Figura 4**, a parte do quadro denominada **GTC Payload** é demonstrada a seguir. No *payload* estão as **GEMs**. Para cada **GEM** que a **ONU** possui, são definidos os campos de cabeçalho e de dados. A **Figura 6** mostra os detalhes, de modo que os campos são especificados da seguinte forma:

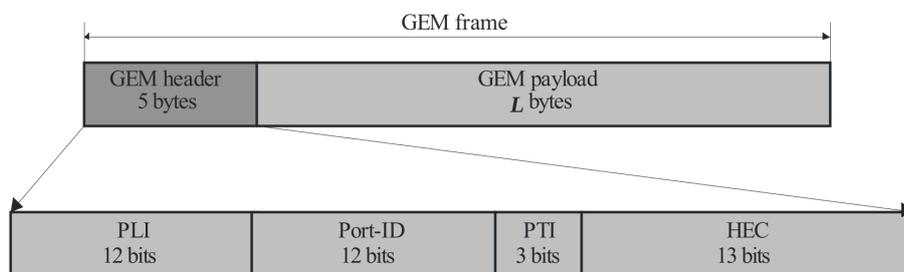
- **PLI** (*Payload Length Indicator*) [12 bits]: indica o tamanho do *payload*, que servirá de delimitação para encontrar o início do próximo cabeçalho **GEM Header**. Com 12 bits é possível que o tamanho do *payload* seja de, no máximo, 4095 *bytes*. Portanto, caso os dados a serem transmitidos ultrapassem os 4095 *bytes*, o quadro deverá ser dividido em mais de uma transmissão;
- **Port-ID** [12 bits]: é o identificador da **GEM**, podendo variar de 0 a 4095;

Figura 5 – Quadro *Upstream Bandwidth Map*



FONTE: ITU-T (2014)

Figura 6 – Quadro GEM



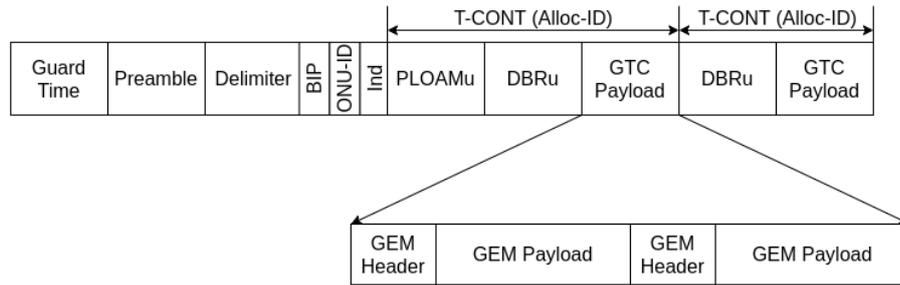
FONTE: ITU-T (2014)

- **PTI** (*Payload Type Indicator*) [3 bits]: é utilizado para indicar o tipo de conteúdo do *payload*, que pode ser um dado de rede do usuário, ou algum dado de procedimento **OAM**, por exemplo;
- **HEC** (*Header Error Control*) [13 bits]: utilizado para cálculo de códigos corretores de erro.

No sentido *upstream* cada **ONU** envia os quadros de acordo com o que é mostrado na **Figura 7**. Na imagem é possível verificar o exemplo da divisão de contêineres de transmissão, em que uma **ONU** pode possuir vários **T-CONTs**, que por sua vez podem encapsular várias **GEMs**. Os detalhes da transmissão em *upstream* não serão explanados neste trabalho, tendo em vista o objetivo de capturar somente pacotes em *downstream*.

O estabelecimento da comunicação nos dois sentidos permite a realização de procedimentos de operação, administração e manutenção (**OAM**) numa rede **GPON**. Estes procedimentos podem ser realizados utilizando três métodos. O primeiro deles, o método *Embedded OAM* é realizado utilizando os campos nos quadros *downstream* e *upstream* que levam informações de tempo real, como monitoramento de **BER** (*Bit Error Ratio*), por exemplo. Já os métodos **PLOAM** e **OMCI** utilizam um canal bidirecional (*upstream* e

Figura 7 – Quadro *Upstream*



FONTE: Elaborada pelo autor.

downstream) que permite a troca de mensagens entre OLT e ONU (ITU-T, 2014). Esses métodos OAM são descritos na subseção a seguir.

2.2.2 PLOAM

O método PLOAM é descrito ainda na ITU-T G.984.3 (ITU-T, 2014), pois as operações OAM realizadas fazendo o uso deste recurso possuem características de camada física. Utilizando esse método são realizadas operações de ativação de uma ONU, estabelecimento do canal OMCC (*Optical network termination Management and Control Channel*), configurações de criptografia entre OLT e ONU, chave de gerenciamento e sinalização de alarmes. Para este recurso são reservados 13 bytes nos cabeçalhos *downstream* (PLOAMd), conforme mostrado na Figura 4, e *upstream* (PLOAMu - *Physical Layer Operations, Administrations and Maintenance upstream*), conforme mostrado na Figura 7. Esses 13 bytes são divididos conforme apresenta a Figura 8. A especificação de cada campo do quadro é descrita a seguir.

Figura 8 – Quadro PLOAM

ONU-ID 1 byte	Message-ID 1 byte	Data 10 bytes	CRC 1 byte
------------------	----------------------	------------------	---------------

FONTE: Elaborada pelo autor.

- **ONU-ID (*Optical Network Unit Identifier*) [1 byte]:** indica o ID da ONU a qual a mensagem está endereçada. Este ID é atribuído durante o processo de *ranging*, realizado na ativação de uma ONU, e pode variar de 0 a 253. O valor 255 (0xFF) é utilizado para as mensagens em *broadcast*;
- **Message-ID [1 byte]:** indica o tipo da mensagem para que a interpretação do campo *Data* seja realizada de forma correta. A especificação 01/2014 da ITU-T G.984.3 descreve 18 tipos de mensagens no sentido *downstream* e 9 no sentido *upstream*;

- **Data [10 bytes]:** contém os dados da mensagem formatados de acordo com a especificação;
- **CRC (*Cyclic Redundancy Check*) [1 byte]:** contém o valor de CRC para detecção de erros.

2.2.3 OMCI

O padrão **OMCI** (*Optical network termination Management and Control Interface*), definido na ITU-T G.984.4 (ITU-T, 2008b), especifica um modo de gerenciamento de configurações, falhas e performance nas **ONTs** de uma rede **GPON**. Este protocolo é definido com o intuito de possibilitar a interoperabilidade entre diferentes fabricantes de dispositivos. Para realizar este gerenciamento, o protocolo **OMCI** utiliza o conceito de mestre e escravo, no qual a **OLT** assume o papel de mestre, aplicando configurações e efetuando o gerenciamento das **ONTs**, que assumem o papel de escravo. Dentre os recursos de gerenciamento do protocolo **OMCI** é possível citar:

- **Gerenciamento de configurações:** permite à **OLT** exercer funções de controle sobre as configurações da **ONT** (*GET e SET*), das quais é possível citar: configuração das interfaces **UNI** (*User-Network Interface*), configuração de **GEMs**, configuração de serviços, entre outros;
- **Gerenciamento de falhas:** possibilita o envio de notificações de falhas a partir da **ONT**, indicando à **OLT** sobre alguma anormalidade em seu funcionamento;
- **Gerenciamento de performance:** permite que a **OLT** realize o monitoramento da performance da **ONT** utilizando mensagens controle;
- **Gerenciamento de segurança:** permite que a **OLT** habilite ou desabilite a criptação dos dados transmitidos.

Para que seja possível realizar as operações de gerenciamento do protocolo **OMCI**, a **OLT** cria um canal denominado **OMCC** (*Optical network termination Management and Control Channel*) com a **ONT**. A definição deste canal é realizada utilizando mensagens **PLOAM**, conforme descrito na subseção 2.2.2, em que a **OLT** determina uma **GEM** que será utilizada como canal **OMCC**. Com o canal de gerenciamento definido, as mensagens **OMCI** seguem o formato exibido na Figura 9.

O campo **GEM Header** denominado no quadro **OMCI** segue a especificação já descrita na subseção 2.2.1 (Figura 6), carregando o valor da **GEM** do canal de gerenciamento **OMCC**. Os demais campos possuem as seguintes descrições:

Figura 9 – Quadro OMCI

GEM header (5 bytes)	Transaction correlation identifier (2 bytes)	Message type (1 byte)	Device identifier (1 byte)	Message identifier (4 bytes)	Message contents (32 bytes)	OMCI trailer (8 bytes)
-------------------------	---	--------------------------	-------------------------------	---------------------------------	--------------------------------	---------------------------

FONTE: ITU-T (2014)

- **Transaction correlation identifier [2 bytes]:** é o identificador da transação, utilizado para associar uma mensagem com sua resposta;
- **Message type [1 byte]:** este campo possui as seguintes subdivisões:
 - Bit 8 - *Destination Bit* (DB): de acordo com a ITU-T G.984.4, no OMCI este campo é sempre setado como 0;
 - Bit 7 - *Acknowledge Request* (AR): indica se a mensagem em questão deve ou não ter uma confirmação (ACK);
 - Bit 6 - *Acknowledgement* (AK): indica se a mensagem em questão é uma confirmação de outra mensagem;
 - Bits 5 a 1 - *Message Type* (MT): esses bits indicam o tipo da mensagem. Alguns exemplos são: *Create* (4), *Delete* (6), *Set* (8), *Get* (9), *Alarm* (16), entre outros. O tipo da mensagem indica como o campo *Message contents* deve ser interpretado. Cada um dos tipos e suas especificações estão definidos na recomendação ITU-T G.984.4.
- **Device identifier [1 byte]:** conforme especificação ITU-T G.984.4, este campo possui o valor fixo de *0x0A*;
- **Message identifier [4 bytes]:** os dois primeiros *bytes* deste campo são utilizados para indicar a entidade gerenciada (ME) e os dois últimos são utilizados para indicar a instância da entidade gerenciada;
- **Message contents [32 bytes]:** este campo possui o conteúdo em si da mensagem OMCI, ou seja, é nele que serão informados os parâmetros de gerenciamento da ONT;
- **OMCI trailer [8 bytes]:** no OMCI, os dois primeiros *bytes* desse campo é setado como *0x0000* e ignorado pelo receptor. Os próximos dois *bytes* são setados com o valor *0x0028*. Por fim, os últimos quatro *bytes* correspondem ao CRC definido na norma ITU-T I.363.5 (ITU-T, 1996).

Para realizar as operações de gerenciamento, o protocolo OMCI define as MEs (*Managed Entities*), ou entidades gerenciadas. Essas entidades funcionam como MIBs, que têm o objetivo de representar recursos e serviços presentes na ONT. É com elas que a OLT

faz o gerenciamento das funcionalidades disponíveis na **ONT**. Cada **ME** pode possuir até 36 *bytes*, onde estão inclusos os campos *message identifier* e *message contents*, descritos anteriormente (**Figura 9**). Para realizar a interpretação dos valores descritos em uma entidade gerenciada é preciso recorrer a norma **ITU-T G.988 (ITU-T, 2017)**, que define atualmente 369 **MEs** (versão 11/2017), com seus respectivos campos e valores esperados. O campo *message identifier*, descrito anteriormente, utiliza 2 *bytes* para identificação da **ME**. Isso significa que os valores podem variar de 1 a 65535, ou seja, o protocolo **OMCI** já prevê expansão para até 65535 **MEs**. Desta forma, da faixa que ainda não está em uso, os valores 453 a 65279 são reservados para padronização futura. A faixa de 172 a 239 é reservada para o **BPON** e as faixas de 240 a 255 e 65280 até 65535 são para uso livre, ou seja, abertas para utilização e definição de cada fabricante.

A título de exemplo, uma das **MEs** definidas na **ITU-T G.984.4** é a *IP host config data*. Esta **ME** possui a identificação 134 e é utilizada para configurar alguns parâmetros de **IP** na **ONT**. A **Tabela 2** mostra o que é definido na norma para esta **ME**.

O exemplo da **ME IP host config data** mostra alguns parâmetros de camada três que podem ser configurados via **OMCI**. Esta **ME** permite operações de *GET*, *SET* e *Test*, que deverá ser especificado no campo *Message type*, conforme mostrado na **Figura 9**. A partir da **Tabela 2** é possível observar que a somatória de *bytes* desta **ME** ultrapassa o estipulado no quadro **OMCI**. Isso acontece pois alguns tipos de mensagens **OMCI**, como *GET* e *SET*, possuem uma máscara de atributos (*attribute mask*) que é encaminhada nos primeiros 2 *bytes* do campo *message contents* do quadro **OMCI**. A partir dessa máscara, o receptor identifica quais atributos estão sendo passados na mensagem para interpretá-los de maneira correta. Com isso, caso a **OLT** queira configurar todos os parâmetros da entidade *IP host config data*, ela deverá encaminhar mais de uma mensagem com esta **ME**, variando a máscara de atributos e encaminhando apenas alguns campos por vez, de modo que não ultrapasse a quantidade de *bytes* especificada no quadro **OMCI**.

Nas *managed entities* definidas na padronização da **ITU-T** estão previstas funcionalidades de gerenciamento para recursos de rede de camada dois e três, além de outros serviços que a **ONT** pode ter. A norma define três níveis para utilização das **MEs**. Como primeiro nível, é especificado o *requirement*, que indica que a **ME** em questão é necessária para a compatibilização mínima da operação. Em segundo nível, classificadas como *conditional requirements*, estão as **MEs** que só são necessárias caso alguma função opcional seja utilizada. Em terceiro nível, as **MEs** com classificação *option* são aquelas que não são requeridas para a compatibilização mínima operacional (**ITU-T, 2008b**).

Além das especificações de níveis entre as entidades gerenciadas, dentro de cada **ME**, um campo pode ser definido como obrigatório ou opcional, conforme visto no exemplo da **Tabela 2 (ITU-T, 2008b)**. A alternância destes campos varia a interpretação realizada pelo receptor, podendo fazer com que a **ONT** entre num estado inválido caso algum

Tabela 2 – Exemplo - ME (*Managed Entity*) *IP host config data* (134)

Atributo	Valor
Managed entity id	Identificador da instância da entidade gerenciada. (R) (obrigatório) (2 bytes)
IP options	Máscara de bits que habilita ou desabilita configurações IP como DHCP (<i>Dynamic Host Configuration Protocol</i>) e resposta a ping e <i>traceroute</i> . (R,W) (obrigatório) (1 byte)
MAC address	Indica o endereço MAC. (R) (obrigatório) (6 bytes)
ONT identifier	Uma <i>string</i> identificadora única para a ONT, cujo valor padrão é nulo. (R,W) (obrigatório) (25 bytes)
IP address	indica o endereço IP. Possui o valor 0 como padrão e, se informado, sobrescreve a configuração de DHCP. (R,W) (obrigatório) (4 bytes)
Mask	Máscara de subrede. Possui o valor 0 como padrão e, se informado, sobrescreve a configuração de DHCP. (R,W) (obrigatório) (4 bytes)
Gateway	<i>Gateway</i> padrão. Possui o valor 0 como padrão e, se informado, sobrescreve a configuração de DHCP. (R,W) (obrigatório) (4 bytes)
Primary DNS	DNS (<i>Domain Name System</i>) primário. Possui o valor 0 como padrão e, se informado, sobrescreve a configuração de DHCP. (R,W) (obrigatório) (4 bytes)
Secondary DNS	DNS primário. Possui o valor 0 como padrão e, se informado, sobrescreve a configuração de DHCP. (R,W) (obrigatório) (4 bytes)
Current address	Endereço IP atual. (R) (opcional) (4 bytes)
Current mask	Máscara de subrede atual. (R) (opcional) (4 bytes)
Current gateway	<i>Gateway</i> padrão atual. (R) (opcional) (4 bytes)
Current primary DNS	DNS primário atual. (R) (opcional) (4 bytes)
Current secondary DNS	DNS secundário atual. (R) (opcional) (4 bytes)
Domain name	Indica o nome de domínio fornecido pelo DHCP. Assume o valor nulo (25 bytes nulos) caso o servidor DHCP não tenha fornecido um nome de domínio. (R) (obrigatório) (25 bytes)
Host name	Indica o nome do <i>host</i> fornecido pelo DHCP. Assume o valor nulo (25 bytes nulos) caso o servidor DHCP não tenha fornecido um nome de domínio. (R) (obrigatório) (25 bytes)

Fonte – ITU-T (2008b)

Nota – As letras R e W mostradas na tabela indicam a permissão de leitura (R) e escrita (W).

atributo obrigatório não seja informado. Além disso, existem campos que possuem valores pré-definidos e que devem ser respeitados para o correto funcionamento do gerenciamento proporcionado pelo OMCI. A não conformidade de um fabricante com os atributos especificados pela norma pode ocasionar mal funcionamento ou até incompatibilidades entre diferentes marcas, gerando restrições de uso para o cliente GPON.

Conforme mostrado pelo exemplo da ME *IP host config data*, uma entidade OMCI

pode possuir diversos atributos, cada um deles contendo um tipo específico de dado, ou seja, um número (inteiro), uma sequência de caracteres (*string*), uma máscara de bits, etc. Um tipo de dado específico utilizado nos atributos das **MEs** é o ponteiro. Esse tipo de dado indica que aquele atributo faz referência a uma outra **ME**, ou seja, a uma outra entidade **OMCI**. Como forma de exemplo, a tabela a seguir descreve a **ME TCP/UDP (Transmission Control Protocol)/UDP (User Datagram Protocol) config data**.

Tabela 3 – Exemplo - **ME TCP/UDP config data** (136)

Atributo	Valor
Managed entity id	Identificador da instância da entidade gerenciada. (R) (obrigatório) (2 bytes)
Port ID	Especifica o número da porta que oferece o serviço TCP/UDP . (R,W) (obrigatório) (2 bytes)
Protocol	Indica o protocolo utilizado. (R,W) (obrigatório) (1 byte)
TOS/Diffserv Field	Especifica o valor de TOS (Type Of Service)/Diffserv do cabeçalho IPv4 (Internet Protocol version 4) . (R,W) (obrigatório) (1 byte)
IP host pointer	Aponta para as MEs IP host config data ou IPv6 (Internet Protocol version 6) host config data associada a esta entidade. (R,W) (obrigatório) (2 bytes)

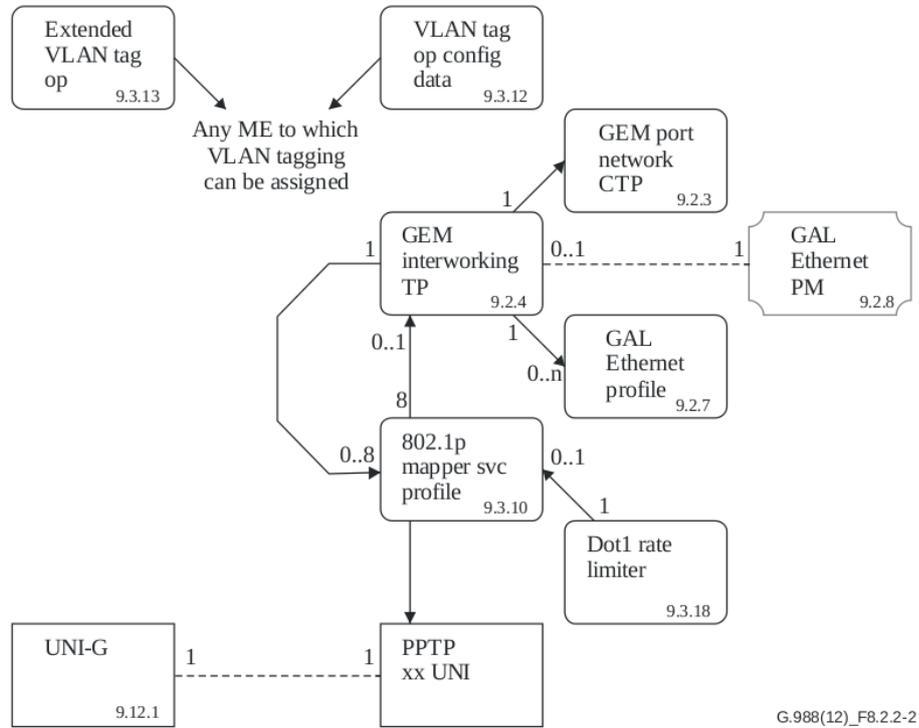
Fonte – ITU-T (2017)

De acordo com a **Tabela 3**, é possível verificar que existe a definição de um ponteiro na **ME TCP/UDP config data**. O atributo **IP host pointer** aponta para uma entidade do tipo **IP host config data** ou **IPv6 host config data**. Isso indica que o valor deste campo deve ser o número de uma instância de uma entidade **OMCI**, que neste caso devem ser as entidades **IP host config data** ou **IPv6 host config data**. O número da instância de uma entidade é definido pelo campo **Managed entity id**, presente em todas as **MEs**. Supondo que se tenha uma entidade do tipo **IP host config data** (conforme **Tabela 2**) cujo número de instância é 100 (definido pelo atributo **Managed entity id**) e uma entidade do tipo **TCP/UDP config data** queira fazer referência a ela, o atributo **IP host pointer** da **ME TCP/UDP config data** deverá conter o valor 100.

Com esses ponteiros entre as entidades, o protocolo descreve as interações (ou ligações) entre as **MEs**, o que torna possível a criação de serviços por meio da manipulação dos recursos disponíveis nos equipamentos que compõem a rede. Além deste tipo de ligação, as especificações da **ITU** também definem as ligações implícitas entre as **MEs**. Essas ligações são feitas com base apenas no número da instância da entidade, não tendo um atributo específico do tipo ponteiro, conforme exemplificado anteriormente. Neste caso, a ligação entre as duas entidades é dada apenas pelo número da instância. Considerando uma entidade A com número de instância 100, sua ligação é estabelecida com a entidade B, cujo número de instância é o mesmo, ou seja, 100. As normas **ITU-T (2008b)** e **ITU-T (2017)** definem quais **MEs** possuem esse tipo de ligação entre si.

A Figura 10 foi retirada da especificação G.988 (ITU-T, 2017) e serve de exemplo para mostrar como são os relacionamentos entre as entidades OMCI.

Figura 10 – Exemplo *MAC Bridged LAN* OMCI



G.988(12)_F8.2.2-2

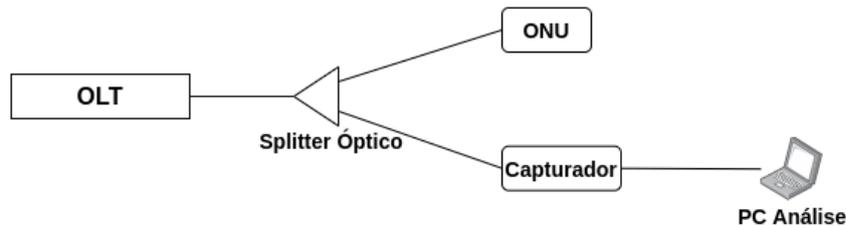
FONTE: ITU-T (2017)

Após a fundamentação realizada neste capítulo, o capítulo a seguir descreve as etapas de desenvolvimento da ferramenta proposta, chamada de analisador OMCI.

3 ANALISADOR OMCI

Este trabalho descreve o desenvolvimento de um analisador **OMCI**, que inclui um módulo capturador de mensagens **OMCI**, e outro que correlaciona e analisa as mensagens capturadas. A ferramenta consiste em dois terminais: um capturador e um *software* analisador, conforme mostrado na [Figura 11](#). O módulo capturador consiste de uma **ONU** com o *firmware* modificado, cujo objetivo é encaminhar as mensagens **OMCI** identificadas na porta **PON** para a porta *ethernet*. O segundo módulo é composto por um **PC** (*Personal Computer*), que executa o *software* analisador. Seu objetivo é receber as mensagens do capturador e analisá-las.

Figura 11 – Cenário de captura da ferramenta



FONTE: Elaborada pelo autor.

O desenvolvimento do projeto foi dividido em três etapas. A primeira, descrita na [seção 3.1](#), consiste no desenvolvimento do capturador, responsável por capturar as mensagens **OMCI** da rede **GPON**. A segunda e terceira etapas se baseiam no desenvolvimento do *software* que faz a análise das mensagens capturadas e estão descritas na [seção 3.2](#) e na [seção 3.3](#).

3.1 Capturador OMCI

Na ferramenta desenvolvida, a captura das mensagens **OMCI** que trafegam na rede **GPON** é realizada por uma **ONU** com o *firmware* modificado, cujo objetivo é identificar as mensagens **OMCI** recebidas e encaminhá-las no formato de um quadro *ethernet* para o módulo analisador. A **ONU** utilizada como capturador é a **ONU** modelo 110, do fabricante Intelbras, mostrada na [Figura 12](#). Conforme descrito no [Capítulo 2](#), esse equipamento funciona como escravo da **OLT** e está localizado na borda da rede óptica, com intuito de fornecer o acesso ao usuário final. O modelo 110 foi o escolhido por ser desenvolvido pela própria Intelbras, ou seja, a empresa detém o domínio sobre seu *firmware*, no qual foram realizadas as alterações para transformá-la num capturador de mensagens **OMCI**. O modelo escolhido possui uma interface com a rede **PON** (óptica) e uma interface do tipo *Ethernet* (elétrica), que faz a conexão com a rede do assinante. Entretanto, no cenário de

utilização como ferramenta de captura, a interface *ethernet* da ONU faz conexão com o PC no qual é executado o *software* analisador. Importante ressaltar que, em seu modo de funcionamento normal, a ONU não encaminharia as mensagens OMCI para a interface *ethernet*, uma vez que essas mensagens seriam utilizadas para configuração da própria ONU na rede PON.

Figura 12 – ONU 110 - Utilizada como capturador



FONTE: (INTELBRAS, 2020)

O *firmware* da ONU possuía, inicialmente, as características de funcionamento de uma ONU comum, ou seja, seu objetivo era funcionar como escravo da OLT, interpretando as mensagens OMCI e criando as configurações necessárias para o funcionamento da rede. Além disso, outros tipos de mensagens trafegam pela ONU, como mensagens PLOAM e pacotes de dados (*ethernet*, por exemplo). Todavia, a finalidade da ONU na ferramenta desenvolvida é capturar apenas as mensagens OMCI e encaminhá-las através da interface *ethernet* para o PC, sem que sejam aplicadas no equipamento as configurações contidas nestas mensagens. Sendo assim, nesta primeira etapa, foi desenvolvida a lógica para encaminhar as mensagens OMCI recebidas no *firmware* do capturador diretamente para a porta *ethernet*, sem que sejam chamadas as funções de configuração dentro do *software* do dispositivo. Este procedimento foi realizado para impedir que o equipamento crie configurações de rede que possam impactar no funcionamento do capturador, já que este seria o comportamento normal da ONU. O procedimento é necessário tendo em vista que o protocolo OMCI gerencia as configurações de rede na ONU e portanto, poderia atribuir configurações que bloqueassem o encaminhamento das mensagens pela interface *ethernet*. Além disso, nesta etapa, foram realizadas modificações visando o encaminhamento dessas mensagens pela interface de rede, as quais foram descritas na subseção 3.1.1.

O capturador monitora as mensagens recebidas por qualquer ONU que esteja conectada à rede óptica, porém somente uma de cada vez, com base no número de série e no *vendor ID* da ONU a ser monitorada. O *vendor ID* é uma identificação de quatro

letras que indica o fabricante do equipamento. Por exemplo, o *vendor ID* da Intelbras é ITBS. Sendo assim, essa fase do projeto também teve por objetivo o desenvolvimento de uma forma de configuração sobre qual dispositivo o capturador deve monitorar.

Após a execução de todas as alterações necessárias na ONU para que funcione de acordo com um capturador, um *firmware* do equipamento foi gerado e disponibilizado com as modificações realizadas.

3.1.1 Identificação e encaminhamento das mensagens OMCI no capturador

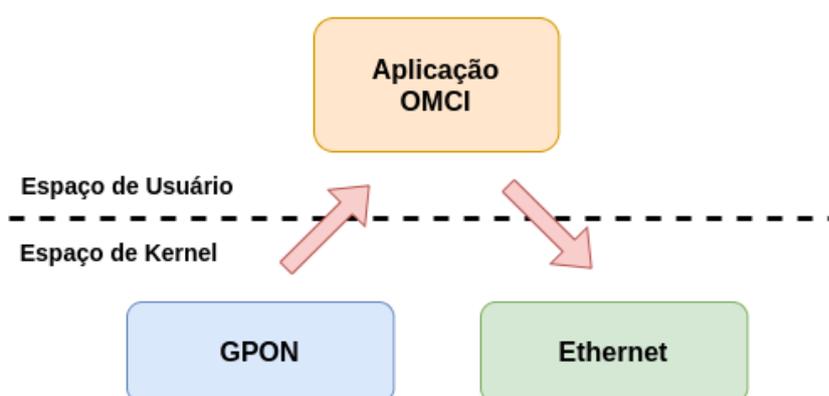
O *firmware* da ONU utilizada para desenvolvimento do capturador é baseado no sistema operacional Linux, com o SDK (*Software Development Kit*) fornecido pela fabricante do chipset PON. O modelo utilizado é o MC-88F6601, do fabricante Marvell. Este chipset faz parte da família AVANTA, que disponibiliza chipsets para o mercado de banda larga óptica. Ele é denominado como um chipset UPON (*Universal PON*), podendo ser utilizado em dispositivos EPON (*Ethernet Passive Optical Network*) e GPON. Seu objetivo é converter o sinal óptico em elétrico e vice-versa, fornecendo acesso ao usuário final da rede. O chipset utiliza o conceito de SoC (*System on Chip*), baseado em um processador ARM. Ele implementa as funções de camada MAC GPON e P2P (*Peer-to-Peer*) Ethernet, que são necessárias para a identificação e encaminhamento das mensagens OMCI capturadas. O SDK fornecido pelo fabricante implementa também o OMCI, interpretando as mensagens recebidas e aplicando as configurações solicitadas pela OLT. Dentre essas configurações é possível citar a habilitação de GEMs de dados e configurações de funcionamento da interface *ethernet*. O processamento de tais configurações é realizado em espaço de usuário, onde é possível identificar o recebimento das mensagens OMCI e providenciar o redirecionamento para a porta *ethernet*.

O capturador intercepta as mensagens OMCI com o objetivo de transmiti-las para o *software* analisador. Para efetuar este direcionamento, foi codificada a abertura de um *socket* no momento da criação do canal OMCC no *firmware* da ONU. Este canal é criado no processo de provisionamento da ONU. Conforme descrito no Capítulo 2, o canal OMCC é definido a partir de uma mensagem PLOAM enviada pela OLT, na qual, com base no número de série da ONU, é informada a GEM por onde serão encaminhados as mensagens OMCI. Sendo assim, é necessário que o capturador receba esta mensagem PLOAM para que seja possível capturar as mensagens OMCI vindas da OLT. Em outras palavras, caso o capturador seja conectado na rede óptica após a ONU a ser monitorada, é necessário que seja realizado um "reload" da criação do canal OMCC, que pode ser feito a partir de um comando na OLT, ou mesmo removendo e reinserindo a ONU a ser monitorada na rede óptica (importante ressaltar que o procedimento de "reload" deve ser feito na ONU que está sendo monitorada, e não no capturador). Após a criação do *socket* na ONU, para realizar o redirecionamento das mensagens através da porta *ethernet*, a mensagem

OMCI foi encapsulada num quadro *Ethernet*, utilizando os endereços MAC (*Media Access Control address*) de origem (definido como *bb:bb:bb:bb:bb:bb*), destino (definido como *aa:aa:aa:aa:aa:aa*) e protocolo (definido como *EtherType 0x88b5 - Local Experimental Ethertype*). Todos estes três campos foram definidos de modo arbitrário e foram descritos de forma fixa no *firmware* gerado (*hard coded*). Por conta desta implementação, o capturador deve ser sempre conectado diretamente no PC de análise para que sejam recebidas as mensagens na placa de rede do computador. Sendo assim, o fluxo das mensagens OMCI no capturador ocorre conforme o sentido descrito na Figura 13.

Conforme descrito no Capítulo 2, a comunicação no sentido *downstream*, ou seja da OLT para a ONU, utiliza o comprimento de onda de 1490 nm, enquanto que no sentido *upstream*, o comprimento de onda utilizado é de 1310 nm. Desta forma, o módulo óptico utilizado no capturador consegue capturar somente os quadros enviados em 1490 nm, o que impõe uma limitação para o sistema desenvolvido, pois só podem ser capturadas as mensagens no sentido *downstream*. Para que fosse possível capturar o que a ONU monitorada está enviando (sentido *upstream*), seria necessário uma alteração no *hardware* do capturador, substituindo o módulo óptico utilizado para um que recebesse quadros no comprimento de onda de 1310 nm. Ou seja, o sistema precisaria de dois módulos ópticos para que fosse possível capturar em ambos os sentidos. Além disso, seria necessário refletir o sinal enviado pela ONU monitorada, de modo que chegasse até o módulo óptico do capturador, o que poderia ser feito através de um arranjo de *splitters* ópticos.

Figura 13 – Fluxo das mensagens OMCI no capturador



FONTE: Elaborada pelo autor.

3.1.2 Configuração da ONU a ser monitorada

Conforme descrito anteriormente, o capturador pode monitorar qualquer ONU que esteja conectada a rede óptica, porém somente uma por vez. Portanto, é necessário que o usuário desta ferramenta informe qual será a ONU a ser monitorada. Para permitir esta configuração, foi alterado um dos *scripts* que são executadas na inicialização do equipamento. Este *script* foi modificada para que o número de série e o *vendor ID* da

ONU fossem configurados a partir de um novo arquivo que foi adicionado ao SDK do equipamento. Sendo assim, para configurar a ONU a ser monitorada, o usuário deve acessar o capturador via protocolo *telnet* e alterar o *vendor ID* e o número de série contidos no arquivo *vendor_sn_set*. Na prática, ao configurar o *vendor ID* e o número de série no capturador, o dispositivo irá se comportar como se ele mesmo fosse a ONU descrita pela tupla. Assim, ao receber uma mensagem PLOAM contendo o *vendor ID* e o número de série configurados, será executado o procedimento de criação do canal OMCC, seguindo o fluxo descrito na seção anterior.

Ainda nesse mesmo *script* executado na inicialização da ONU, foi inserido procedimento para que o transmissor (TX) do módulo óptico seja desligado. Esse procedimento foi realizado para garantir que o capturador somente escute o que está sendo enviado pela OLT na rede óptica, e não transmita nada. Ou seja, sua presença na rede não é identificada.

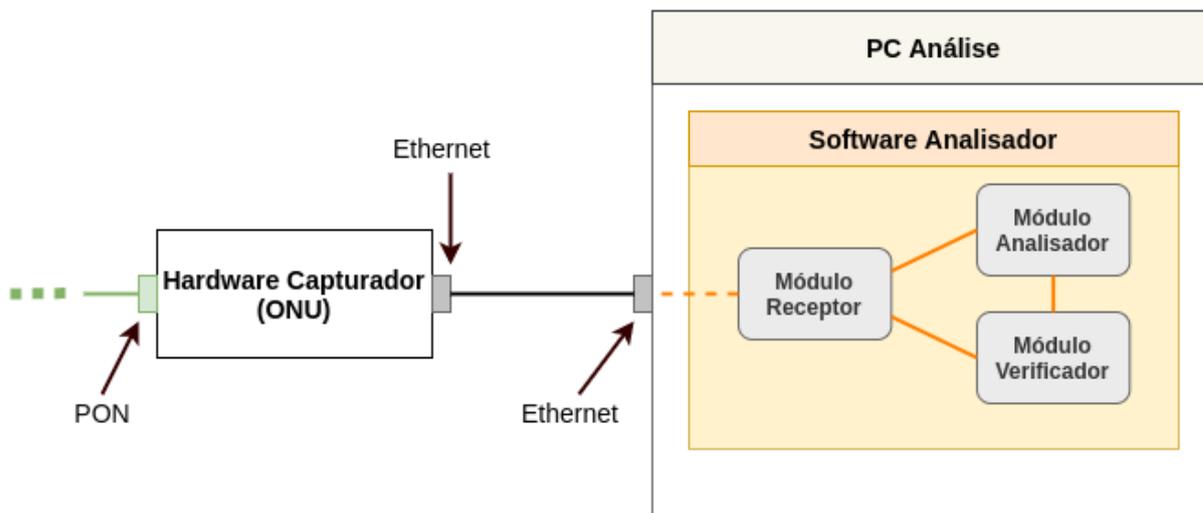
3.2 Software Analisador

O protocolo OMCI é responsável por estabelecer o relacionamento entre as MEs (*Managed Entities*), proporcionando o gerenciamento da ONU. Cada mensagem OMCI enviado pela OLT possui uma única ME, que pode referenciar ou ser referenciada por outra ME. Essa outra ME pode ter chegado, ou irá chegar, numa outra mensagem. Esta seção descreve o desenvolvimento do *software* responsável por essa análise. Seu objetivo é identificar e exibir as interações entre as MEs existentes na captura, indicando ao usuário como foram organizadas as mensagens enviadas pela OLT.

Conforme exposto no início deste capítulo na Figura 11, o sistema conta com um PC, que é conectado ao *hardware* capturador (ONU). Neste PC é executado o *software* responsável pela análise e verificação das MEs enviadas pela OLT. A Figura 14 fornece uma visão geral da implementação realizada no desenvolvimento do *software* analisador.

O *software* desenvolvido neste projeto utiliza linguagem de programação Python e tem como plataforma nativa o SO (Sistema Operacional) Linux. Além disso, sua execução é realizada utilizando o terminal do SO, não tendo sido disponibilizada interface gráfica para operação. Para maiores detalhes da implementação foi disponibilizado o diagrama de classes do *software*, localizado no Apêndice A, Figura 34.

De acordo com a Figura 14, três módulos principais compõem o *software* analisador, são eles: receptor, analisador e verificador. Esta seção descreve a segunda das três etapas propostas nessa monografia, que contempla os módulos receptor e analisador, descritos nas subseções seguir. O módulo verificador é descrito na seção 3.3.

Figura 14 – Diagrama geral do *software* analisador

FONTE: Elaborada pelo autor.

3.2.1 Módulo Receptor

O desenvolvimento do *hardware* capturador descrito na [seção 3.1](#) possibilitou o encaminhamento das mensagens OMCI pela interface *ethernet* da ONU. Essas mensagens são entregues encapsulados em quadros *ethernet*, contendo portanto um MAC de origem, um MAC de destino e um protocolo (*EtherType*). Como é possível verificar, a comunicação entre o *hardware* capturador e o PC de análise é realizada através da interface *ethernet*. Com base nisso, o primeiro módulo desenvolvido no *software* analisador foi o receptor.

O módulo receptor tem a responsabilidade de monitorar e capturar as mensagens entregues pelo *hardware* capturador na interface *ethernet* do PC. Para executar essa operação, o *software* utiliza um *socket* do tipo *RAW*, o qual possibilita a captura de todos os tipos de pacotes a partir da camada dois do modelo TCP/IP. Além disso, só são capturados pacotes com o protocolo *0x88b5* (*Local Experimental*), que é o *EtherType* inserido pelo *hardware* capturador nas mensagens OMCI detectadas.

Ao receber um pacote da interface de rede do PC, o módulo receptor remove o cabeçalho *ethernet* e armazena-o num *buffer*, que será compartilhado com o módulo analisador, caso o usuário execute a operação de análise, ou verificador, caso o usuário execute operações de verificação. O receptor também desencapsula a mensagem OMCI e exibe algumas informações contidas nela com o intuito de mostrar que a captura está sendo feita e o que está sendo recebido em tempo real. As informações das mensagens exibidas são: *transaction correlation identifier*, *message type*, *acknowledge request*, *acknowledgement*, *managed entity (ME)* e *instance*.

Além de capturar os pacotes da interface de rede, o módulo receptor também disponibiliza ao usuário a opção de salvá-los em formato *.pcap*. Esta operação possibilita

que a captura seja restaurada futuramente no analisador, ou ainda, que ela seja visualizada no *software* de análise de rede Wireshark. Esta última funcionalidade tem por finalidade possibilitar ao usuário visualizar a captura numa ferramenta que é de uso comum para profissionais que trabalham com redes de computadores para análise de protocolos.

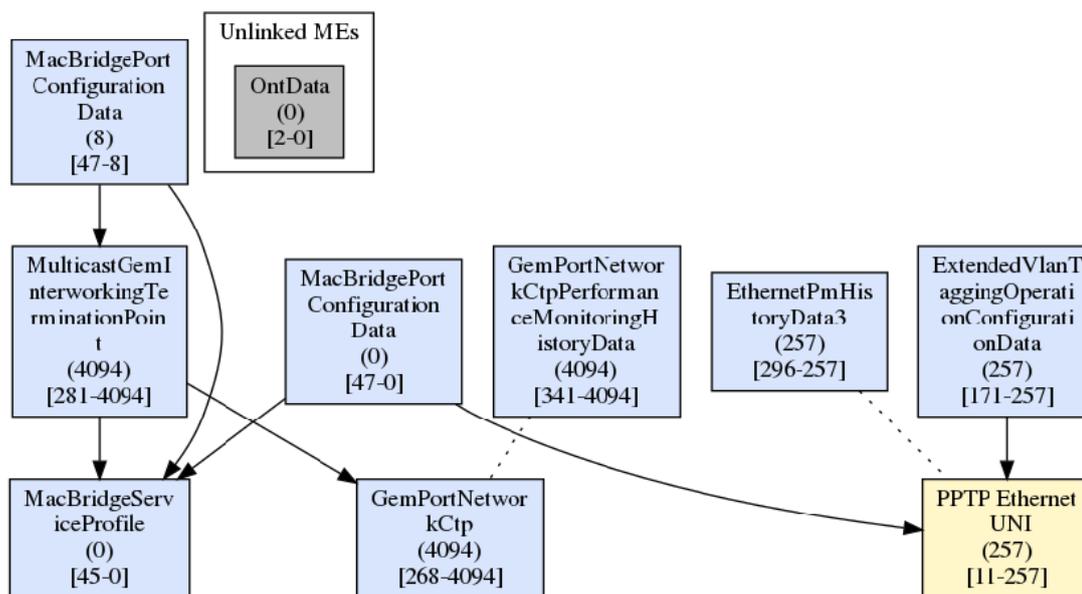
3.2.2 Módulo Analisador

O analisador é o responsável pela maior parte do processamento das mensagens OMCI capturadas. Ele é o responsável pela tradução, organização e análise das mensagens recebidas. Quando o usuário solicita ao *software* uma análise da captura, o módulo analisador extrai as informações de cada mensagem dos tipos *CREATE* e *SET*, e cria instâncias de classes para manusear todos os dados extraídos. Essas classes instanciadas nada mais são do que as próprias MEs, ou seja, as *Managed Entities* fornecidas pelas normas da ITU, que foram mapeadas em classes que possuem atributos conforme a especificação do protocolo OMCI. Dessa forma, é possível criar instâncias das entidades gerenciadas e manuseá-las como instâncias de objetos dentro do *software*. Com os objetos criados para cada classe de ME, o analisador faz uma varredura nos atributos de todos eles para identificar os apontamentos entre cada instância. Realizada esta análise, o usuário tem a opção de gerar uma imagem, que exhibe as ligações detectadas pelo analisador. A Figura 15 é um trecho de uma imagem de análise realizada com base numa captura de exemplo.

A base de MEs criada neste projeto teve como origem uma base já utilizada no setor de pesquisa e desenvolvimento da empresa Intelbras. Ela é uma base em formato texto, utilizada com frequência pelos desenvolvedores para visualização das mensagens OMCI no *software* Wireshark. Ela foi convertida para o formato de classes Python, conforme descrito no parágrafo anterior. A base possui cerca de 90 MEs, que englobam as principais entidades definidas na ITU-T G.988 (ITU-T, 2017).

Na Figura 15 são exibidas as entidades identificadas pelo analisador. Em azul estão as entidades que possuem ligação direta com alguma outra entidade, ou seja, algum de seus atributos é do tipo *pointer* (ponteiro) e aponta para a ME indicada através de uma flecha. A linha pontilhada entre algumas MEs indica uma ligação implícita, conforme definido na norma. Neste tipo de ligação, as entidades são conectadas pelo número de instância, e não por um atributo do tipo *pointer*. Na Figura 15, é o caso da ME *Ethernet Pm History Data 3*, implicitamente ligada com a ME *PPTP Ethernet UNI* pelo número da instância, neste caso, 257. Importante frisar que a norma define quais são as ligações implícitas possíveis, ou seja, no caso da ME *Ethernet Pm History Data 3*, a ITU-T G.988 define que ela só possui ligação implícita com a ME *PPTP Ethernet UNI*, por isso na imagem não são exibidas ligações entre essa entidade e outras entidades com o mesmo número instância (257).

O capturador desenvolvido neste trabalho, conforme seção 3.1, é capaz de capturar

Figura 15 – Trecho de imagem gerada pelo *software* analisador

FONTE: Elaborada pelo autor.

somente mensagens no sentido *downstream*. Com base nisso, o analisador desenvolvido faz sua análise somente nas mensagens deste sentido. Porém é no sentido *upstream* que a *OLT* identifica quais são as entidades presentes na *ONU* para poder criar e setar as configurações a partir das instâncias informadas por ela. Sendo assim, como o analisador não tem as mensagens no sentido *upstream*, ele exibe em amarelo as entidades que foram referenciadas pela *OLT*, mas que não tenham sido enviadas mensagens no sentido *downstream*, ou seja, as entidades apresentadas na cor amarela na saída do analisador são aquelas que foram referenciadas pela *OLT* no fluxo *downstream*, mas que não tiveram mensagens enviadas neste sentido. É o caso da *PPTP Ethernet UNI* no exemplo acima, ou seja, a *OLT* não enviou uma operação de *CREATE* ou *SET* em específico para esta *ME*, mas ela foi referenciada por outras entidades que tiveram operações de *CREATE* ou *SET* pela *OLT*, como as *MEs* *MAC Bridge Port Configuration Data*, *Ethernet Pm History Data3* e *Extended Vlan Tagging Operation Configuration Data*.

Além do que foi descrito nos parágrafos anteriores, o analisador também exibe um bloco denominado "*Unlinked MEs*". Este bloco exibe as entidades que tiveram operações de *CREATE* ou *SET* enviadas pela *OLT*, mas que não possuem ligações diretas ou indiretas com outras entidades.

Conforme mostrado na Figura 14, o módulo analisador foi colocado logo após o módulo receptor. Isso porque pressupõe-se uma sequência lógica para o usuário, na qual é realizada primeiramente a captura das mensagens, seguida da análise das instâncias das *MEs* *OMCI*. Entretanto, este módulo também permite ao usuário efetuar uma análise em uma captura já existente. Para isto, deve ser informado um arquivo no formato *.pcap* e, a partir dele, o analisador colocará em memória os dados das mensagens *OMCI*, permitindo

a realização de uma análise, conforme descrita anteriormente. Esta funcionalidade permite que o usuário utilize outras ferramentas para realizar a captura das mensagens OMCI da rede PON, e não somente a que foi desenvolvida neste projeto. Ou seja, desde que a captura esteja no formato adequado (*.pcap*), o analisador conseguirá analisá-la, independentemente de onde ela foi realizada. Isso dá flexibilidade ao usuário do *software* de análise. Todavia, é importante reafirmar que somente mensagens no sentido *downstream* serão considerados pelo analisador.

Como forma de visualização, o analisador permite também a exibição de cada ME. Esta exibição é feita com base no valor que aparece entre colchetes em cada entidade descrita na imagem gerada pelo *software*. Na Figura 15 é possível visualizar alguns exemplos. Sendo assim, ao indicar algum destes identificadores para o analisador, a instância de ME é exibida em formato texto para o usuário.

3.3 Módulo Verificador

De acordo com a ITU-T G.984.4, as MEs disponíveis no OMCI possuem atributos obrigatórios e não obrigatórios. Além disso, esses campos possuem tamanhos pré-determinados e permissões de leitura e escrita, que devem ser respeitadas. Desta forma, ao receber as mensagens com as MEs, a ONU conseguirá interpretar corretamente a informação que foi enviada pela OLT e aplicar as configurações ou operações por ela solicitadas. O módulo verificador foi desenvolvido com o objetivo de realizar verificações nas mensagens OMCI enviadas pela OLT, com o intuito de conferir se as regras impostas pelo protocolo foram respeitadas nas entidades configuradas. Foram desenvolvidas três verificações, descritas nas subseções a seguir.

3.3.1 Verificação de atributos obrigatórios

As definições das MEs (*Managed Entities*) contidas nas normas ITU-T G.984.4 ou G.988 descrevem entidades que possuem atributos a serem configurados pela OLT. Esses atributos podem ser obrigatórios ou não, conforme definido pela norma. A função de verificação de atributos obrigatórios visa conferir se, nas entidades configuradas pela OLT, todos os atributos obrigatórios foram informados.

Para realizar a verificação dos atributos obrigatórios, foi inserido na base de MEs utilizada pelo *software* a informação em cada atributo, indicando se ele é obrigatório ou não. Tendo essa informação, o procedimento de análise do módulo verificador consiste em primeiro converter as mensagens capturadas em estruturas conhecidas, ou seja, convertê-los em instâncias de classes de MEs, assim como acontece no módulo analisador. Justamente por esse processo ser realizado também no analisador, o módulo verificador utiliza o mesmo método para converter as mensagens OMCI em objetos de classes de MEs. Após

a conversão, o verificador percorre cada atributo de cada instância criada para verificar se ele é obrigatório ou não, e se está vazio. Caso esteja vazio (não tenha sido informado pela OLT) e seja um atributo obrigatório, é gerada uma saída para o usuário, indicando o nome da ME, o número da instância e o atributo que está em branco.

3.3.2 Verificação de tamanho das MEs

Além da definição de obrigatoriedade de cada atributo, a especificação da ITU determina tamanhos fixos para cada um deles. Essa definição é feita para que a ONU consiga interpretar a mensagem enviada pela OLT e vice-versa. O objetivo desta verificação é conferir se o tamanho total da ME confere com o que seria o esperado, de acordo com a definição da norma. Esta verificação leva em consideração todos os atributos configurados pela OLT e seus respectivos tamanhos somados, comparando o resultado com a somatória dos tamanhos desses atributos de acordo com o que está descrito na especificação.

Para realizar esta verificação, após converter todas as mensagens recebidas em objetos de MEs, o *software* percorre todos os atributos não nulos de todas as instâncias criadas e contabiliza o seu tamanho. O tamanho de cada atributo é definido como sendo a quantidade de *bytes* encontrados até o último *byte* nulo, definido como *0x00*, conforme especificado na norma da ITU. Cada tamanho de atributo é somado e, por fim, comparado com o tamanho esperado de acordo com a norma. Caso haja divergência, é gerada uma saída para o usuário, indicando o nome da ME, a instância e os valores esperado *versus* calculado.

3.3.3 Verificação de configuração sem permissão

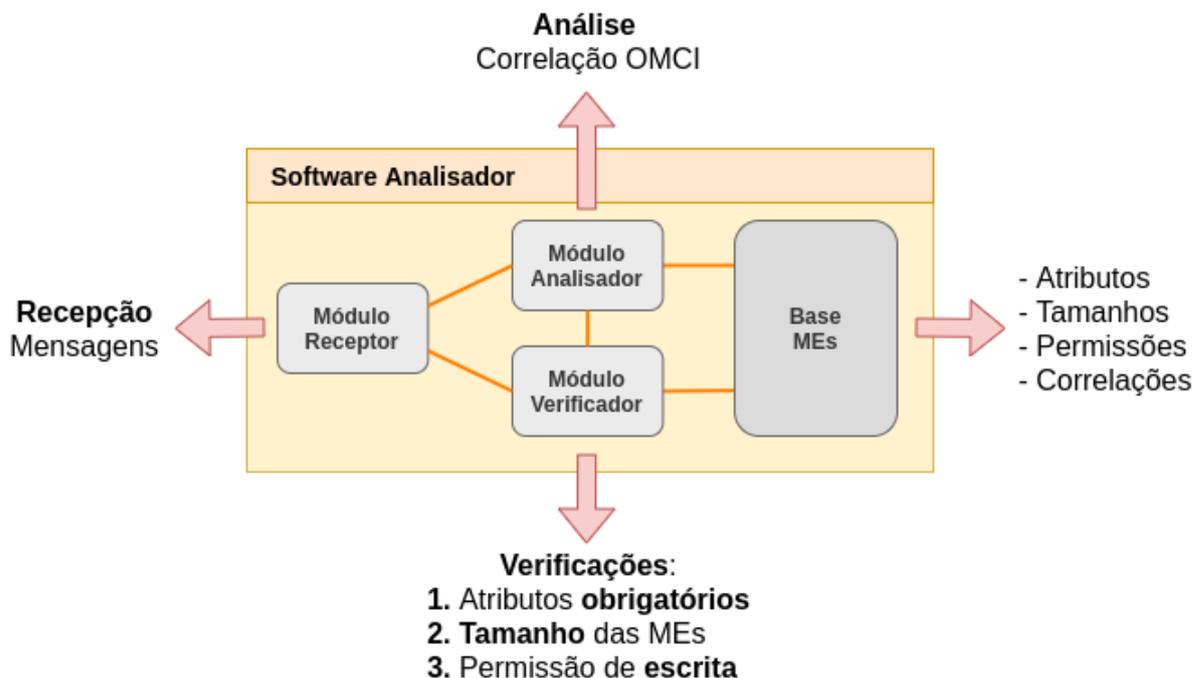
Ainda na especificação das MEs nas normas ITU-T G.984.4 e G.988, cada atributo de cada entidade definida possui uma permissão de leitura e escrita. Ou seja, do ponto de vista da OLT, o atributo pode ser somente leitura, somente escrita ou leitura e escrita. Esta verificação tem por objetivo checar se a OLT está tentando realizar alguma configuração (escrita) em um atributo que é somente leitura.

Para realizar a verificação de configuração sem permissão, foi inserido na base de MEs a informação em cada atributo, indicando se ele é somente leitura, somente escrita ou leitura e escrita. Tendo essa informação e todas as instâncias de entidades configuradas pela OLT, o *software* percorre cada atributo não nulo de cada entidade e verifica sua permissão, de acordo com a especificação. Caso o atributo tenha sido configurado e sua permissão seja somente leitura, uma saída é gerada para o usuário, indicando o nome da ME, a instância e o atributo que foi configurado de forma errônea.

3.4 Visão geral do software Analisador

Nas seções anteriores foram descritos os módulos do *software* analisador. Para melhor visualização, a Figura 16 apresenta uma visão geral do que foi desenvolvido. Ela descreve todos os seus componentes, bem como suas respectivas atribuições, de forma resumida.

Figura 16 – Visão geral do *software* analisador



FONTE: Elaborada pelo autor.

3.5 Modo de utilização do software desenvolvido

De acordo com o que foi proposto para este projeto, o *software* desenvolvido é executado em sistema operacional Linux. Sua execução é realizada utilizando linha de comando, apresentando dois modos de execução: contínuo ou *OneShot*. No modo contínuo, conforme Figura 17, o usuário pode navegar entre as funções disponibilizadas no menu, as quais são descritas a seguir:

- **Start Capture:** inicia uma captura. Após selecionar esta opção, o usuário deve informar a interface de rede do PC que deseja monitorar e a captura é iniciada. A origem da captura pode ou não ser o capturador desenvolvido nesta monografia. Caso não seja, basta que contenha um cabeçalho *Ethernet* com o campo *EtherType* igual a *0x88b5*, seguido da mensagem OMCI;

- **Load Capture:** carrega uma captura. Após selecionar esta opção, o usuário deve informar um arquivo no formato *.pcap* (*Wireshark*), do qual serão extraídas as mensagens e geradas as entidades **OMCI**;
- **Save Capture:** salva uma captura. Com a captura realizada, esta opção tem a função de salvá-la em formato *.pcap* (*Wireshark*);
- **Analyse:** analisa uma captura. Esta é a opção que irá gerar o arquivo de imagem (*.png*) mostrando o relacionamento das entidades **OMCI**. Após selecioná-la, o usuário deve informar o nome do arquivo de imagem de saída;
- **Show Entity:** mostra uma entidade. A imagem gerada na opção *Analyse* possui um índice para cada **ME**, descrito entre colchetes e definido como: $\langle ME_ID \rangle - \langle Entity_ID \rangle$. Ao selecionar esta opção, o usuário deve informar o índice da entidade para que então, a **ME** seja exibida;
- **ME Verifier:** verificador de **ME**. Esta opção chama o modo verificador do *software*. Após selecioná-la, um submenu é exibido para que o usuário escolha uma das três verificações que o programa realiza. O resultado destas verificações são exibidos na tela. Ainda no submenu, há uma quarta opção para que seja gerado um relatório com o resultado das três verificações. O relatório é então gerado de acordo com o nome de arquivo fornecido pelo usuário, no formato *.txt*.

Figura 17 – *Software* analisador executado em modo contínuo

```
vini@vini-laptop:~$ sudo omci_analyser -c
#####
##### Main Menu #####
#####

Choose an option and press ENTER:
  1) Start Capture
  2) Load Capture
  3) Save Capture
  4) Analyse
  5) Show Entity (ME)
  6) ME Verifier

  0) Exit

Option:
```

FONTE: Elaborada pelo autor.

No modo de utilização *OneShot*, o programa é executado como um comando Linux, executando o procedimento de acordo com o que foi informado na linha de comando do terminal. A [Figura 18](#) mostra a ajuda que é exibida ao usuário quando o comando é executado sem nenhum parâmetro ou quando é utilizado o parâmetro *-help*.

Figura 18 – Ajuda do analisador executado em modo *OneShot*

```
vini@vini-laptop:~$ omci_analyser --help
usage: omci_analyser [-h] [-c | -i INTERFACE | -a PCAP_FILE | -v PCAP_FILE]
                  [-o FILENAME] [--show-zero-pointers]

GPON OMCI Analyser

Mandatory Params:
  -h, --help          show this help message and exit
  -c                  Run in continuous mode
  -i INTERFACE        Interface to capture
  -a PCAP_FILE        Analyse and generate image from capture
  -v PCAP_FILE        Verify packets
  -o FILENAME         Output file
  --show-zero-pointers If -a option was selected, generate an image showing
                      pointers with zeroed instance

vini@vini-laptop:~$
```

FONTE: Elaborada pelo autor.

Conforme apresentado na [Figura 18](#), a sintaxe do comando a ser utilizada pelo usuário utiliza o seguinte formato:

```
omci_analyser [-h] [-c | -i INTERFACE | -a PCAP_FILE | -v PCAP_FILE] [-o FILENAME]
```

As opções disponibilizadas pelo do comando possuem as funções conforme descrito a seguir:

- **-h, -help:** chama a ajuda do comando;
- **-c:** executa o programa em modo contínuo, conforme explicado anteriormente;
- **-i INTERFACE:** executa o capturador na interface informada;
- **-a PCAP_FILE:** executa o analisador no arquivo informado;
- **-v PCAP_FILE:** executa o verificador no arquivo informado;
- **-o FILENAME:** arquivo de saída a ser gerado, podendo ser com as extensões *.pcap*, *.png* e *.txt*, a depender da operação selecionada;
- **-show-zero-pointers:** exibe os ponteiros para valores de intância zero nas entidades que não tiveram mensagens enviadas no *downstream*. Maiores detalhes sobre esse parâmetro foram descritos na [seção 4.4](#).

A [Figura 19](#) mostra um exemplo onde foram realizadas as três operações: captura, análise e verificação das mensagens OMCI.

Figura 19 – Exemplo de execução do *software* em modo *OneShot*

```

vini@vini-laptop:~$ sudo omci_analyser -i enp1s0 -o captura-00.pcap

Starting capture. Press CTRL + C to stop.

#####

33153 | Type: Get           | AR: 1 | AK: 0 | ME: 11 | Inst: 257
33154 | Type: Get           | AR: 1 | AK: 0 | ME: 11 | Inst: 258
33155 | Type: Get           | AR: 1 | AK: 0 | ME: 11 | Inst: 259
33156 | Type: Get           | AR: 1 | AK: 0 | ME: 11 | Inst: 260
33157 | Type: Set           | AR: 1 | AK: 0 | ME: 11 | Inst: 257
33158 | Type: Set           | AR: 1 | AK: 0 | ME: 11 | Inst: 258
33159 | Type: Set           | AR: 1 | AK: 0 | ME: 11 | Inst: 259
33160 | Type: Set           | AR: 1 | AK: 0 | ME: 11 | Inst: 260
^C
Saved sucessfully!

vini@vini-laptop:~$ omci_analyser -a captura-00.pcap -o captura-00.png
vini@vini-laptop:~$ omci_analyser -v captura-00.pcap -o captura-00.txt
vini@vini-laptop:~$ ls captura-00*
captura-00.pcap  captura-00.png  captura-00.txt
vini@vini-laptop:~$

```

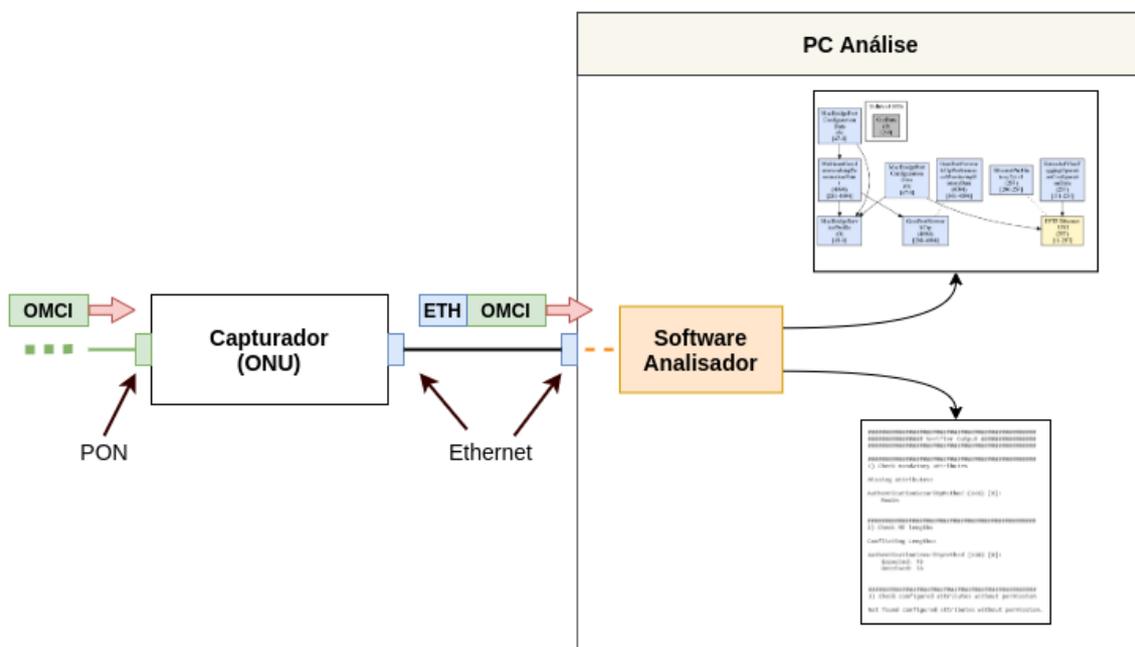
FONTE: Elaborada pelo autor.

3.6 Visão geral da ferramenta

Por fim, após a conclusão das etapas de desenvolvimento, foi obtida uma ferramenta que captura mensagens **OMCI** utilizando uma **ONU** que, por sua vez, as encapsula em quadros *ethernet* e encaminha para um computador. No computador é executado um *software* que faz a análise destas mensagens e gera duas saídas: um diagrama de correlação entre as entidades **OMCI** e um relatório de verificação de conformidade das **MEs**. A [Figura 20](#) representa uma visão geral da ferramenta desenvolvida.

O capítulo a seguir descreve os testes que foram realizados para avaliar o funcionamento da ferramenta.

Figura 20 – Visão geral da ferramenta



FONTE: Elaborada pelo autor.

4 TESTES E RESULTADOS

Neste capítulo são descritos os testes e resultados obtidos com a utilização da ferramenta desenvolvida. Na [seção 4.1](#) é descrita uma visão geral de cada teste e os objetivos de cada um deles. Nas seções em sequência são descritos os detalhes de execução dos testes realizados e os resultados obtidos.

4.1 Testes e objetivos

Com o intuito de verificar o funcionamento do capturador desenvolvido por meio de modificações no *firmware* da ONU Intelbras, bem como o funcionamento do *software* analisador executado em um PC, foram realizados três principais testes. Estes testes visam a validação da ferramenta como um todo, abrangendo tudo o que foi desenvolvido neste projeto. A seguir é apresentada uma breve descrição sobre cada um, indicando o objetivo e as métricas utilizadas.

Testes com OLTs diferentes: apesar das definições do protocolo OMCI, diferentes OLTs possuem diferentes formas de criação de fluxos e configurações nas ONUs/ONTs. Uma parte dessas diferenças se dá pela própria natureza do protocolo, pois suas definições utilizando entidades gerenciadas possibilitam uma certa flexibilidade na criação dos recursos. Além disso, diferentes OLTs podem enviar MEs diferentes, dependendo do que se deseja configurar, nas quais podem haver erros e inconsistências. Uma parte das incompatibilidades citadas na fundamentação teórica desta monografia pode ter origem em uma dessas causas. Sendo assim, o objetivo deste teste é verificar como a ferramenta se comporta analisando mensagens de diferentes OLTs. Ou seja, se as mensagens foram capturadas integralmente pelo capturador e se foram interpretadas de forma adequada tanto pelos módulos de análise quanto de verificação. As métricas para cada um dos módulos testados foram as seguintes:

- **Módulo capturador:** como métrica da integridade da captura foi realizada uma comparação, mensagem a mensagem, do que foi capturado no capturador, com o que foi gerado pela OLT, onde é utilizada uma função de espelhamento de mensagens OMCI. Através desta função de espelhamento a OLT duplica os pacotes OMCI enviados para a ONU em uma de suas portas *ethernet*, sendo possível a conexão de um PC para análise das mensagens que foram enviadas. Sendo assim, caso ambas as capturas (do capturador e da função de espelhamento) estivessem iguais em quantidade e tipos de mensagens, o resultado do teste seria dado como positivo;

- **Módulo analisador:** como métrica foi utilizada a captura "crua" das mensagens capturadas, onde são realizadas as verificações, diretamente nas mensagens das [MEs](#), das ligações exibidas na figura de saída. Também foram checadas se todas as entidades presentes na captura foram apresentadas na imagem gerada. Caso todas as entidades e ligações exibidas na imagem do analisador fossem identificadas nas mensagens e vice-versa, o resultado do teste seria considerado positivo;
- **Módulo verificador:** no caso do verificador, são identificados os erros indicados pelo relatório gerado e realizada uma checagem diretamente na mensagem para constatar se o erro realmente ocorreu. Se todos os erros indicados pelo verificador de fato existissem ao verificar as entidades diretamente nas mensagens capturadas, o resultado do teste seria considerado positivo.

Testes com mensagens manipuladas: no teste anterior, no módulo verificador, foi verificado se os erros descritos no relatório ocorreram de fato. Neste teste, o objetivo foi o contrário, ou seja, conferir se os erros não identificados pelo verificador de fato não ocorreram. Isso foi feito inserindo erros propositais em algumas [MEs](#), de forma que valide as três verificações realizadas pelo *software*. Além disso, foram omitidos os envios de algumas entidades para verificação também do módulo analisador. As métricas utilizadas foram as seguintes:

- **Módulo verificador:** foi gerado um relatório com a captura original e outro com a captura com os erros inseridos (propositais). Ambos foram comparados para identificar se os erros inseridos foram identificados pelo *software*. Se todos os erros propositais mais os erros anteriores da captura original estivessem presentes no relatório gerado pelo verificador, o resultado do teste seria dado como positivo;
- **Módulo analisador:** foram geradas duas imagens principais, uma a partir da captura original e outra com a captura omitindo o envio de duas [MEs](#): uma com apontamento (gerada originalmente com a cor azul) e outra sem apontamento (gerada originalmente com a cor cinza - *unlinked MEs*). As imagens foram comparadas para verificar se houve alteração em relação as [MEs](#) omitidas. Se a entidade com apontamento passasse a ser exibida na cor amarela e a entidade que antes era exibida na cor cinza, desaparecesse, o resultado seria considerado positivo.

Testes comparativos: o objetivo deste teste foi comparar a análise das ligações entre as [MEs](#) gerada pelo módulo analisador com a mesma análise gerada por uma ferramenta de mercado. Como métrica foi utilizada cada entidade mostrada na imagem gerada pela ferramenta comercial, comparando se ela estaria presente na análise do *software* desenvolvido, e vice-versa. Também foram verificadas as ligações entre as entidades,

comparando se estavam representadas de forma equivalente nas duas análises. O resultado deste teste seria dado como positivo se todas as entidades presentes na análise do *software* estivessem presentes na análise gerada pela ferramenta de mercado, e vice-versa. Com exceção das entidades em que fosse necessária análise do fluxo *upstream*, pois neste caso, a ferramenta desenvolvida neste projeto não dispõe da captura de dados neste sentido.

As seções a seguir descrevem os detalhes, indicando o cenário e o procedimento realizado em cada um dos testes de validação.

4.2 Testes com OLTs diferentes

O projeto desenvolvido nesta monografia baseou sua análise e verificação num protocolo cujo objetivo é a comunicação entre os dispositivos mestre (OLT) e escravo (ONU/ONT). As dificuldades de compatibilização mencionadas na [Capítulo 2](#) advém do níveis mais altos do protocolo GPON, ou seja, justamente da criação e organização das entidades gerenciadas (MEs) do OMCI. Sendo assim, toda a base do protocolo, por exemplo, o tamanho das mensagens, a forma de comunicação, as características físicas e até a forma de ingressar na rede (provisionamento) tendem a ser comum entre os fabricantes. Com base nisso, a ferramenta desenvolvida deve ser capaz de capturar e analisar mensagens OMCI de qualquer OLT, de modo que consiga exibir como estão relacionadas as entidades gerenciadas e verificar possíveis falhas presentes nas mensagens encaminhadas. O teste com diferentes OLTs visou validar esta característica, buscando garantir o funcionamento da ferramenta com OLTs de diferentes fabricantes, de forma que fosse possível exibir e analisar diferentes modos de organização das entidades gerenciadas do OMCI, deixando visível para o usuário o que está sendo encaminhado no sentido *downstream*, ou seja, da OLT para as ONUs/ONTs.

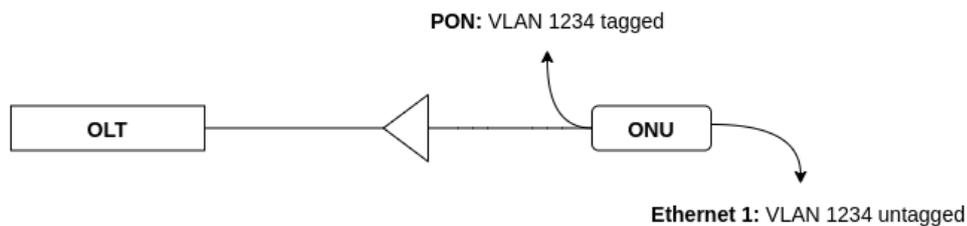
4.2.1 Cenário

Foram utilizadas duas OLTs para execução dos testes, ambas fornecidas pelo fabricante Intelbras. A primeira foi o modelo 8820i, cujo *firmware* foi desenvolvido pela própria empresa. A versão utilizada foi a 2.62. A segunda OLT utilizada nos testes ainda estava em desenvolvimento durante a realização dos testes. Ela também é fornecida pela Intelbras, porém a organização das mensagens OMCI enviadas por essa nova OLT é diferente do modelo 8820i. A utilização de uma OLT ainda em desenvolvimento foi relevante para a validação da ferramenta, pois esse é um dos cenários em que fora prevista a sua utilização. Para elaboração do cenário, foi utilizada uma ONU modelo 110B e uma ONT modelo 142NG, ambas do fabricante Intelbras. A ONU 110B possuía apenas uma porta *ethernet* e só funcionava em modo *bridge*, ou seja, não possuía capacidade de funcionamento em modo roteador. Já a ONT 142NG possuía 4 interfaces *ethernet* e

pôde funcionar em modo *bridge* ou *router* (roteador). As configurações utilizadas foram as seguintes:

- **Configuração 1 (ONU):** criado um único fluxo para a *ethernet 1* com a **VLAN** (*Virtual Local Area Network*) 1234 *untagged* na porta **LAN** (*Local Area Network*) da **ONU**. Este fluxo possuía uma **GEM** exclusiva e um **T-CONT** exclusivo. A **Figura 21** exibe o cenário gerado pela configuração;

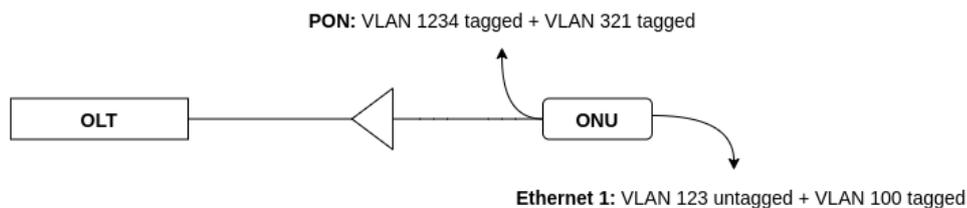
Figura 21 – Testes com OLTs diferentes - Configuração 1



FONTE: Elaborada pelo autor.

- **Configuração 2 (ONU):** criados dois fluxos para a *ethernet 1*, um com a **VLAN** 123 *untagged* na porta da **ONU** e outro com a **VLAN** 321 entre **OLT** e a **WAN** (*Wide Area Network*) da **ONU**, e entregando a **VLAN** 100 na porta **LAN** da **ONU**. Cada fluxo possuía uma **GEM** exclusiva e um **T-CONT** exclusivo. A **Figura 22** exibe o cenário gerado pela configuração;

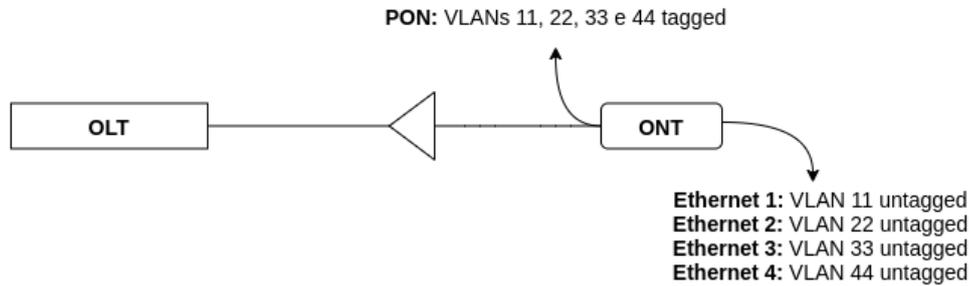
Figura 22 – Testes com OLTs diferentes - Configuração 2



FONTE: Elaborada pelo autor.

- **Configuração 3 (ONT):** criados quatro fluxos conforme listado a seguir. Cada fluxo possuía uma **GEM** exclusiva e um **T-CONT** exclusivo. A **Figura 23** exibe o cenário gerado pela configuração:
 - Fluxo para a *ethernet 1* com a **VLAN** 11 *untagged* na porta **LAN** da **ONT**;
 - Fluxo para a *ethernet 2* com a **VLAN** 22 *untagged* na porta **LAN** da **ONT**;
 - Fluxo para a *ethernet 3* com a **VLAN** 33 *untagged* na porta **LAN** da **ONT**;
 - Fluxo para a *ethernet 4* com a **VLAN** 44 *untagged* na porta **LAN** da **ONT**.

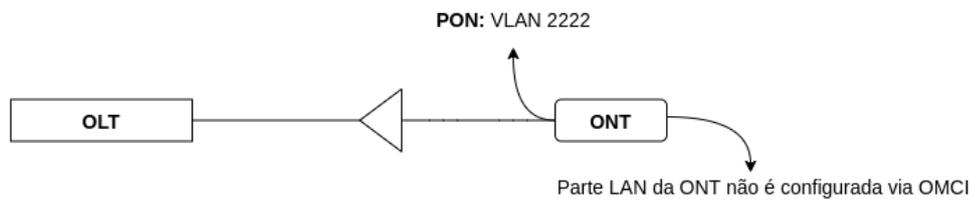
Figura 23 – Testes com OLTs diferentes - Configuração 3



FONTE: Elaborada pelo autor.

- **Configuração 4 (ONT):** configuração do tipo **VEIP** (*Virtual Ethernet Interface Point*) com a **VLAN 2222** (neste tipo de configuração, a **OLT** só configura até a interface **WAN** da **ONT**, deixando a configuração das portas **LAN** por conta da própria **ONT**). Este fluxo possuía uma **GEM** exclusiva e um **T-CONT** exclusivo. A **Figura 23** exibe o cenário gerado pela configuração.

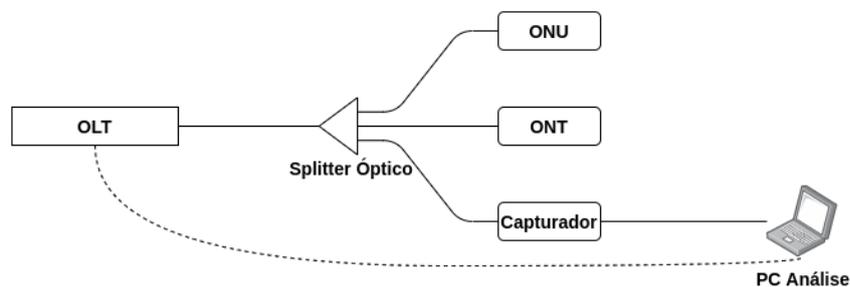
Figura 24 – Testes com OLTs diferentes - Configuração 4



FONTE: Elaborada pelo autor.

A **Figura 25** mostra o cenário preparado para a execução dos testes.

Figura 25 – Cenário testes com OLTs diferentes



FONTE: Elaborada pelo autor.

A conexão exibida entre **OLT** e o **PC** na **Figura 25** foi feita para realização da checagem das mensagens capturadas pelo capturador. Ambas as **OLTs** utilizadas no teste possuíam, a nível de desenvolvimento, uma funcionalidade de espelhamento do **OMCI**, na qual as mensagens eram direcionadas para uma das portas *ethernet* da **OLT**. Sendo assim, como o **PC** utilizado só possuía uma interface de rede, os testes foram realizados em

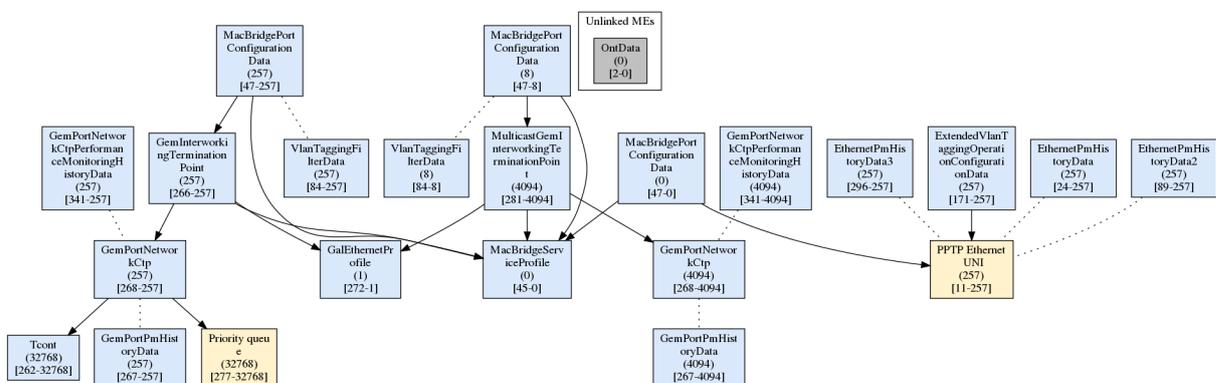
duas etapas. Na primeira, foram realizadas as quatro configurações capturando os dados com a interface de rede do PC conectada ao capturador desenvolvido neste projeto. Na segunda etapa, foram realizadas as mesmas configurações, porém com a interface de rede do PC conectada à porta espelho da OLT. Desta forma, foi possível comparar a captura do capturador desenvolvido diretamente com o que a OLT estava enviando.

4.2.2 Resultados

O resultado colocado como primeiro a ser verificado foi a integridade da captura. Ou seja, antes de qualquer análise, a ferramenta deveria ser capaz de capturar todas as mensagens enviados pela OLT sem corrompimentos ou perdas. Para isto, foram comparadas todas as mensagens provenientes da captura feita pelo capturador com as mensagens provenientes da captura direta da OLT. Como ambas as origens entregavam as mensagens no formato do Wireshark (.pcap), esta foi a ferramenta utilizada para realizar o comparativo. O resultado foi positivo para todas as quatro configurações realizadas em cada uma das OLTs. Todas as mensagens presentes nas capturas fornecidas pelas OLTs estavam presentes nas capturas realizadas pelo capturador, mantendo a integridade e a ordem do que foi enviado.

Em seguida foi executado o analisador para todas as capturas realizadas pelo capturador. O objetivo aqui foi verificar se as entidades, bem como suas ligações, mostradas na imagem gerada pelo módulo analisador estavam de acordo com o que era previsto, com base nas configurações realizadas. O intuito foi verificar a análise gerada de modo mais global, identificando se ocorreria algum erro de geração, ou se seria exibida alguma entidade que não estivesse dentro do contexto da configuração realizada, ou ainda, alguma ligação entre as MEs feita de modo incorreto. A título de exemplo, a Figura 26 mostra a imagem gerada pelo analisador para a configuração 1 com a OLT modelo 8820i.

Figura 26 – Teste OLT 8820i com configuração básica



FONTE: Elaborada pelo autor.

Todas as imagens geradas foram analisadas nos termos citados no parágrafo anterior

e o resultado foi dado como positivo. Não foram identificadas **MEs** sendo exibidas fora de contexto, nem mesmo ligações, implícitas ou não, errôneas. Todas as entidades exibidas em azul foram encontradas na captura e todas as que foram mostradas na cor amarela, de fato, não continham mensagens enviadas pela **OLT** no arquivo de captura. Além disso, a imagem gerada se mostrou legível e de fácil compreensão, proporcionando uma boa visão global das configurações realizadas pela **OLT**.

Em algumas análises foi identificada a presença da **ME PoE** (*Power over Ethernet Control* (349), que não havia sido descrita na base de **MEs** inicial e, portanto, foi exibida como uma **ME** padrão (*default*), a qual não possui a divisão dos atributos. Foi então inserida essa **ME** na base e executada novamente a análise, que passou a exibí-la como uma **ME** conhecida.

Após a verificação do módulo analisador, foi realizada conferência do módulo verificador. Neste módulo, conforme já descrito anteriormente, são realizadas três verificações nas entidades identificadas nas mensagens capturadas. Todas as três verificações foram executadas para todas as capturas realizadas pelo capturador, ou seja, nas quatro configurações em cada **OLT**. A **Figura 27** exibe o relatório gerado para a configuração 1 na **OLT** 8820i.

A primeira das verificações checa se todos os atributos obrigatórios com permissão de escrita foram setados pela **OLT**. A verificação é feita somente nas entidades configuradas por ela. Caso o verificador encontre algum atributo obrigatório com permissão de escrita que não tenha sido informado pela **OLT**, a **ME** e o atributo são informados no relatório de saída do verificador. Para validar o funcionamento desta verificação, os atributos informados no relatório foram comparados com o que está descrito na norma, consultado se eles são mesmo obrigatórios ou não. Com base nisso, foi concluído como positivo o resultado desta análise. Um caso interessante encontrado no relatório foi o da **ME T-CONT** (262), na qual o atributo *Policy* foi relatado. Esse atributo era somente leitura na definição da norma G.984.4, porém passou a ter permissão de leitura e escrita na norma G.988. Como a base de **MEs** foi descrita com base na G.988, o *software* identificou esse atributo como leitura e escrita, indicando o erro, já que a **OLT** não o configurou.

A segunda verificação realizada pelo *software* contabiliza o tamanho esperado da entidade com base nos atributos informados e confere com o que foi enviado pela **OLT**. Caso haja divergência, a **ME** é informada no relatório gerado, indicando o valor de tamanho esperado e o que foi recebido. O objetivo desta verificação seria validar se o tamanho dos atributos estavam sendo respeitados pela **OLT**, porém ela não se mostrou muito útil na prática. Isso porque a lógica para contabilizar o tamanho dos atributos foi verificar a posição do primeiro *byte* nulo da direita para a esquerda. Esta lógica foi adotada, pois as mensagens **OMCI** encaminhados pela **OLT** possuem tamanho fixo, de modo que eles sejam sempre preenchidos com *bytes 0x00* para fechar o tamanho pré-determinado de 53

Figura 27 – Relatório verificação teste OLT 8820i com configuração básica

```
#####  
##### Verifier Output #####  
#####  
  
#####  
1) Check mandatory attributes  
  
Missing Attributes:  
  
Tcont (262) [32768]:  
    Policy  
  
GemInterworkingTerminationPoint (266) [257]:  
    GAL loopback configuration  
  
MulticastGemInterworkingTerminationPoint (281) [4094]:  
    Multicast address table  
  
#####  
2) Check ME lengths  
  
Conflicting Lengths:  
  
Tcont (262) [32768]:  
    Expected: 2  
    Received: 1  
  
GemPortNetworkCtp (268) [257]:  
    Expected: 11  
    Received: 9  
  
ExtendedVlanTaggingOperationConfigurationData (171) [257]:  
    Expected: 24  
    Received: 23  
  
VlanTaggingFilterData (84) [257]:  
    Expected: 26  
    Received: 4  
  
VlanTaggingFilterData (84) [8]:  
    Expected: 26  
    Received: 4  
  
#####  
3) Check configured attributes without permission  
  
Not found configured attributes without permission.
```

FONTE: Elaborada pelo autor.

bytes. Sendo assim, a lógica utilizada não poderia ser aplicada a atributos do tipo inteiro, por exemplo. Além disso, os atributos mostrados na verificação como divergentes eram, na verdade, números ou máscaras de bit, nos quais o valor de *byte* (*0x00*) é aceitável, e não um problema em si. Desta forma, a conclusão final foi de que esta verificação não caracteriza um problema, de fato, gerado pela *OLT*.

A terceira verificação realizada pelo *software* tem por objetivo conferir se a *OLT* está tentando configurar algum atributo que não possua esta permissão, ou seja, caso a *OLT* tente configurar um atributo do tipo somente leitura, o *software* deve exibir esta tentativa, indicando um erro por parte da *OLT*. Em todas as configurações realizadas, não foi detectado nenhum erro nas *OLTs* utilizadas nos testes. Para validar o funcionamento desta verificação, foram revisados as mensagens das capturas e, de fato, não houve nenhuma configuração deste tipo. Apesar de não ter sido encontrado nenhum caso nos testes realizados, esta verificação foi tida como um possível indicador de problemas na configuração, já que pode ocorrer mau funcionamento da *ONU/ONT*, caso a *OLT* execute uma configuração num atributo sem permissão.

4.3 Testes com mensagens OMCI manipuladas

Conforme já descrito, o *software* desenvolvido possui os módulos analisador e capturador. A função central do analisador é interpretar as mensagens *OMCI* e exibir as entidades gerenciadas que foram identificadas nas mensagens enviadas pela *OLT*. O módulo verificador, por sua vez, tem como principal objetivo realizar verificações nestas entidades, de modo a exibir possíveis inconsistências ou fragilidades na configuração de tais *MEs*. Sendo assim, como uma forma de validação do funcionamento destes dois módulos, os testes com mensagens manipuladas foi planejado. Seu objetivo foi inserir possíveis erros nas configurações das entidades *OMCI* de forma proposital, com o intuito de verificar se essas alterações seriam exibidas pelo módulo analisador e se as falhas seriam identificadas pelo verificador.

4.3.1 Cenário

Visando uma maior abrangência dos testes realizados, o cenário para os testes com mensagens manipuladas utilizou uma configuração um pouco mais complexa do que somente os fluxos de rede criados nos testes com diferentes *OLTs* (seção 4.2). A utilização de um cenário *VoIP* necessita não só da criação de um fluxo de rede como nos cenários anteriores, mas além disso, necessita ainda de toda a configuração necessária para utilização dos serviços de voz sobre *IP*, como usuário, senha, servidor *SIP* (*Session Initiation Protocol*), porta, entre outros. Sendo assim, tendo em vista que este já era um caso previsto pelas normas da ITU, a configuração deste cenário foi a escolhida para

três **MEs** foram alteradas, cada uma delas com o objetivo de validar uma verificação do *software*. As modificações realizadas nesta etapa foram as seguintes:

- **ME Authentication Security Method (148)**: não enviado o atributo *username 1*. Este atributo era obrigatório com permissão de escrita, portanto, caso não fosse enviado, o módulo verificador deveria reportar na verificação 1;
- **ME SIP Agent Config Data (150)**: enviado o atributo *softswitch* com apenas 2 *bytes*. Este atributo possuía 4 *bytes*, portanto, caso não fosse enviado por completo, o módulo verificador deveria reportar na verificação 2;
- **ME VoIP Config Data (138)**: enviada operação de *SET* (configuração) para o atributo *Available signalling protocols*. Este atributo possuía somente permissão de leitura, ou seja a **OLT** não poderia configurá-lo. Portanto, caso ocorresse uma tentativa de configuração, o módulo verificador deveria reportar na verificação 3.

A segunda e terceira modificações realizadas no *firmware* da **OLT** tiveram como objetivo validar o funcionamento do módulo analisador. Na segunda modificação, foi removido o envio da primeira **ME** *Physical path termination point POTS UNI* (53), cuja instância era a de número 769, conforme [Figura 29](#). Ao omitir o envio desta **ME**, o resultado esperado era de que ela fosse exibida com a cor amarela na imagem gerada pelo analisador, isso porque ela era referenciada por outras **MEs**. Na terceira e última modificação realizada foi removido o envio da mesma **ME** (53), porém agora na instância 770. Como esta instância não estava sendo referenciada por nenhuma outra **ME** configurada pela **OLT**, o resultado esperado era que a entidade não seria mais exibida no bloco "*Unlinked MEs*", e fosse removida da imagem gerada pelo analisador.

Todas as três etapas de modificações realizadas na **OLT** foram acumulativas, ou seja, a terceira modificação continha também a segunda e a primeira.

4.3.2 Resultados

A primeira modificação realizada no *firmware* da **OLT** teve como objetivo validar as três verificações realizadas pelo módulo verificador. As figuras a seguir foram cortadas para exibirem somente as **MEs** e os atributos alterados nesta etapa. Elas mostram como era e como ficou o relatório gerado pelo verificador.

As alterações realizadas na **OLT** geraram os seguintes resultados nos relatórios gerados pelo verificador:

- **ME Authentication Security Method (148)**: a remoção do envio do atributo *Username 1* resultou na sua apresentação na verificação 1 realizada pelo *software*,

Figura 30 – Comparativo relatório verificador antes e depois das alterações na OLT

Antes	Depois
##### ##### Verifier Output ##### #####	##### ##### Verifier Output ##### #####
##### 1) Check mandatory attributes	##### 1) Check mandatory attributes
Missing Attributes: AuthenticationSecurityMethod (148) [0]: Realm	Missing Attributes: AuthenticationSecurityMethod (148) [0]: Username 1; Realm
##### 2) Check ME lengths	##### 2) Check ME lengths
Conflicting Lengths: AuthenticationSecurityMethod (148) [0]: Expected: 76 Received: 38	Conflicting Lengths: AuthenticationSecurityMethod (148) [0]: Expected: 51 Received: 30
##### 3) Check configured attributes without permission	##### 3) Check configured attributes without permission
Not found configured attributes without permission.	Attributes configured without permission: VoipConfigData (138) [1]: Available signalling protocols

FONTE: Elaborada pelo autor.

indicando que o atributo em questão não foi encontrado na respectiva **ME**. Além disso, essa alteração também impactou a verificação 2 do *software*, já que por consequência, o tamanho esperado da **ME** ficou menor (um atributo a menos);

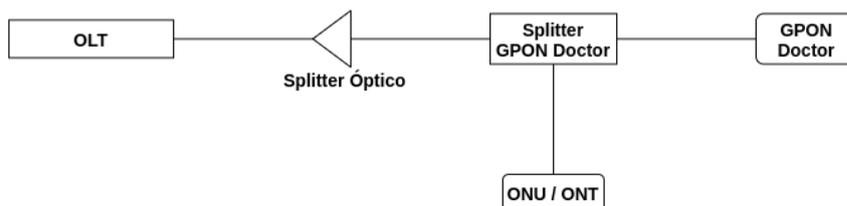
- **ME SIP Agent Config Data (150)**: a redução do tamanho do parâmetro *softswitch* em 2 bytes fez com que o *software* apresentasse essa diferença na verificação 2;
- **ME VoIP Config Data (138)**: o envio da operação de *SET* para o atributo *Available signalling protocols* resultou na apresentação deste atributo na verificação 3 do módulo verificador.

Diante do relatório gerado pela ferramenta, o resultado do teste foi dado como positivo.

A segunda e terceira modificações no *software* da **OLT** também tiveram resultados positivos do ponto de vista das alterações exibidas na imagem gerada pelo analisador. Como elas eram acumulativas, apenas o resultado final foi exibido na imagem a seguir. A **Figura 31** mostra que após a remoção das **MEs** *Physical path termination point POTS*

Diferentemente da ferramenta desenvolvida nesta monografia, o GPON Doctor detecta mensagens tanto no sentido *downstream* como *upstream*, além de capturar também outros tipos de mensagens como PLOAM e quadros *ethernet* encapsulados nas GEMs. A Figura 32 mostra o cenário utilizado nesta etapa de testes.

Figura 32 – Cenário testes comparativos



FONTE: Elaborada pelo autor.

A OLT utilizada nos testes foi a OLT 8820i do fabricante Intelbras. A ONU utilizada foi a 110B e a ONT foi a 142NG, ambas também produzidas pela Intelbras. O Splitter GPON Doctor mostrado na Figura 32 é um *splitter* óptico que acompanha o GPON Doctor. Seu objetivo é espelhar o tráfego para o equipamento de captura para que seja possível detectar mensagens nos dois sentidos (*UP* e *DOWN*). Os dispositivos ONU/ONT foram conectados diretamente neste *splitter*.

Tendo em vista o fim comparativo dos testes aqui descritos, nesta etapa foram utilizadas as mesmas quatro configurações dos testes com OLTs diferentes, descritos na seção 4.2. Para facilitar a leitura, as configurações foram reescritas também nessa seção:

- **Configuração 1 (ONU):** criado um único fluxo para a *ethernet* 1 com a VLAN 1234 *untagged* na porta LAN da ONU. Este fluxo possuía uma GEM exclusiva e um T-CONT exclusivo;
- **Configuração 2 (ONU):** criado dois fluxos para a *ethernet* 1, um com a VLAN 123 *untagged* na porta da ONU e outro com a VLAN 321 entre OLT e a WAN da ONU, e entregando a VLAN 100 na porta LAN da ONU. Cada fluxo possuía uma GEM exclusiva e um T-CONT exclusivo;
- **Configuração 3 (ONT):** criados quatro fluxos conforme listado a seguir. Cada fluxo possuía uma GEM exclusiva e um T-CONT exclusivo:
 - Fluxo para a *ethernet* 1 com a VLAN 11 *untagged* na porta LAN da ONT;
 - Fluxo para a *ethernet* 2 com a VLAN 22 *untagged* na porta LAN da ONT;
 - Fluxo para a *ethernet* 3 com a VLAN 33 *untagged* na porta LAN da ONT;
 - Fluxo para a *ethernet* 4 com a VLAN 44 *untagged* na porta LAN da ONT.
- **Configuração 4 (ONT):** configuração do tipo VEIP com a VLAN 2222 (neste tipo de configuração, a OLT só configura até a interface WAN da ONT, deixando a

configuração das portas LAN por conta da própria ONT). Este fluxo possuía uma GEM exclusiva e um T-CONT exclusivo.

4.4.2 Resultados

O resultado esperado do teste comparativo entre as duas ferramentas estava no módulo analisador do *software* desenvolvido. Esperava-se que a imagem das entidades OMCI gerada por este módulo fosse equivalente à imagem de análise gerada pelo GPON Doctor. Deveria ser levada em consideração também a limitação física da ferramenta desenvolvida, que só era capaz de capturar mensagens no sentido *downstream*.

A partir do cenário e das configurações descritas para este teste foram geradas capturas utilizando o GPON Doctor. As capturas foram limitadas a mensagens OMCI, com o intuito de deixar o campo de análise mais próximo da ferramenta desenvolvida. A partir das capturas, foram geradas as imagens de análise, que apresentavam os relacionamentos entre as entidades, para as quatro configurações utilizadas.

Ao comparar as imagens geradas pelo GPON Doctor com as imagens geradas pela ferramenta desenvolvida, foi possível perceber que estavam bem parecidas, de modo geral. Com exceção das entidades do bloco *Unlinked MEs*, todas as MEs presentes na análise gerada pelo *software* desenvolvido estavam presentes na análise do GPON Doctor. Além disso, as ligações entre as entidades, sejam elas ponteiros diretos ou ligações implícitas, também estavam presentes.

Analisando de forma mais minuciosa as imagens geradas pelos dois *softwares*, foi possível constatar duas MEs presentes na análise do GPON Doctor que não eram mostradas na análise da ferramenta desenvolvida. São elas: ME *Traffic Scheduler G* (instância 32768) e ME *Priority Queue Downstream* (instância 0). Os casos dessas duas MEs são descritos a seguir.

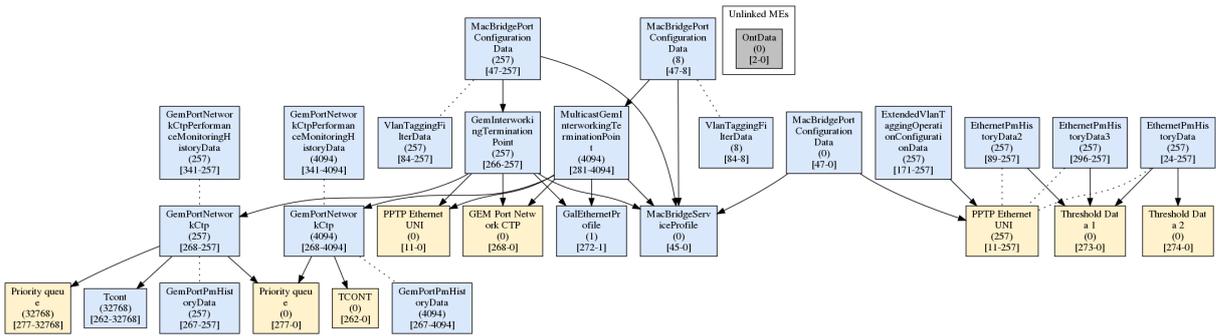
Conforme mencionado anteriormente, diferentemente na ferramenta desenvolvida neste projeto, o GPON Doctor possui capacidade de captura de mensagens nos dois sentidos (*UP* e *DOWN*). Ambas as MEs *Traffic Scheduler* (instância 32768) e *Priority Queue Downstream* (instância 0) foram detectadas pelo GPON Doctor no processo de *MIB Upload* executado entre OLT e ONU/ONT, no qual a ONU/ONT informa à OLT todos os recursos (ou entidades) que possui. É este processo que possibilita à OLT criar e setar configurações nas entidades presentes ONU. Porém, como é a ONU que informa os seus recursos à OLT, a informação relevante deste processo é gerada no sentido *upstream*, o qual não é capturado pela ferramenta desenvolvida. Com o intuito de minimizar o impacto deste caso no *software* de análise, foi implementada a exibição das entidades na cor amarela. As entidades exibidas nesta cor são aquelas que não tiveram nenhuma mensagem enviada em sentido *downstream*, mas que foram referenciadas por alguma outra entidade. Entretanto,

no caso da *ME Traffic Scheduler*, além de não ter sido enviada nenhuma mensagem no sentido *downstream* para esta entidade, ela não foi referenciada por nenhuma outra, o que impossibilitou a sua detecção sem o fluxo *upstream*.

No caso da *ME Priority Queue Downstream*, apesar de não ter sido enviada nenhuma mensagem para esta entidade no sentido *downstream*, ela era referenciada por outras *MEs*, o que deveria fazer com que ela fosse exibida em amarelo. Porém seu valor de instância é 0. De acordo com a ITU-T G.988, o valor de instância 0 deve ser utilizado com cuidado, pois a instância 0, de modo geral, significa um ponteiro nulo. Buscando apresentar os dados de uma forma mais clara, o *software* desenvolvido se baseou nesta informação. Ou seja, para entidades que não tiveram mensagens geradas no sentido *downstream*, ele considera ponteiros com instância zero como inválidos. Entretanto, ainda de acordo com a G.988, em alguns casos, um ponteiro com instância 0 pode apontar para uma entidade válida. E este é o caso da *ME Priority Queue Downstream*. Ela aponta para uma instância válida. Todavia, a verificação de validade ou não do número de instância também depende das mensagens no sentido *upstream*. Na mesma captura realizada, outras *MEs* como *Ethernet PM History data 2*, por exemplo, também apontariam para uma entidade com o valor de instância 0, porém essa ligação não é mostrada pelo GPON Doctor. Isso porque, por possuir acesso as mensagens *upstream*, ele conseguiu verificar que, neste caso, o valor de instância 0 não é válido, ou seja, o valor de instância 0 neste caso indica um ponteiro nulo. Desta forma, por não possuir acesso a captura no sentido *upstream* e com o objetivo de proporcionar uma visualização mais limpa (até mesmo mais parecida com a do GPON Doctor), a estratégia adotada no desenvolvimento do *software* de análise foi considerar estes casos como ponteiros nulos. Esse comportamento foi definido como o padrão, porém foi adicionado o parâmetro `-show-zero-pointers` ao programa. Ao informar este parâmetro, o analisador gera a imagem de análise indicando também os ponteiros para instâncias de valor zero, fazendo com que seja mostrada a *ME Priority Queue Downstream*, como é mostrado no GPON Doctor. No entanto, como era esperado, ao utilizar esta opção, outras entidades com valores de entidades zerados também foram exibidos na saída. A Figura 33 mostra o resultado gerado ao utilizar esta opção.

Em todas as capturas realizadas, as únicas diferenças encontradas no comparativo do GPON Doctor com a ferramenta desenvolvida foram as duas *MEs* citadas acima. A configuração 3 gerou mais de 80 entidades e as únicas diferenças encontradas foram com relação as duas *MEs* citadas nos parágrafos anteriores. Todas as demais entidades e ligações diretas e implícitas foram apresentadas de forma equivalente na comparação entre as duas análises. Sendo assim, o resultado dado para os testes comparativos foi positivo.

Figura 33 – Teste OLT 8820i com configuração básica e opção *show-zero-pointers*



FONTE: Elaborada pelo autor.

5 CONCLUSÕES

O projeto apresentado nesta monografia teve como objetivo o desenvolvimento de uma ferramenta de captura e análise de mensagens OMCI, que trafegam em redes GPON. A captura foi implementada com uma ONU, que teve seu *firmware* modificado para capturar esse tipo de mensagem. Além disso, foi proposto o desenvolvimento de um *software*, executado em um computador, cujo objetivo seria analisar as mensagens capturadas pela ONU.

A partir das modificações realizadas no *firmware* da ONU, foi possível direcionar as mensagens OMCI recebidas em sua interface óptica para a interface *ethernet*, possibilitando a entrega dessas mensagens ao *software* de análise. Para conferir se todas as mensagens enviadas foram capturadas pelo capturador, foram realizados testes com diferentes OLTs, comparando as mensagens enviadas por elas (com uma função de espelhamento de mensagens OMCI) com as mensagens capturadas na ONU. Foi observado que ambas as capturas, da OLT e do capturador, continham as mesmas mensagens, evidenciando o funcionamento do capturador.

No *software* de análise foi desenvolvida a lógica de interpretação das mensagens, que possibilitou a exibição em modo gráfico das interações entre as entidades OMCI, assim como a visualização das MEs recebidas pela ONU. As imagens de análise geradas pelo *software* foram comparadas diretamente com as mensagens capturadas, mostrando que todas as entidades OMCI foram identificadas, bem como as suas interações apresentadas de forma correta pela análise realizada. A análise gerada pelo *software* também foi comparada com a análise realizada por uma ferramenta já comercializada, apresentando forte semelhança entre ambas. A partir dos resultados observados nos testes, foi considerado como positivo o funcionamento da análise fornecida pelo *software* desenvolvido.

Ainda no *software* de análise, foram desenvolvidas três verificações para as entidades OMCI que foram identificadas na captura. Tais verificações tinham como objetivo sinalizar sobre possíveis problemas encontrados nas mensagens enviadas pela OLT. Para validar o funcionamento de tais verificações, foi utilizada uma OLT com alterações em seu *firmware*, cujo objetivo era gerar falhas propositalmente controladas e compará-las com o resultado da verificação fornecida pelo *software*. Como resultado foi observado que todos os erros inseridos na OLT foram identificados pelo *software*, validando o funcionamento das verificações realizadas por ele.

Sendo assim, é possível concluir que a ferramenta desenvolvida atendeu plenamente aos requisitos que foram propostos. As saídas geradas tanto pela ONU capturador quanto pelos módulos do *software* analisador se mostraram consistentes, podendo ser utilizadas

para depuração e análise de problemas ou incompatibilidades de uma rede GPON. E, apesar de ser possível realizar capturas somente no sentido *downstream*, conforme previsto, a análise deste fluxo permitiu demonstrar as interações geradas entre as entidades OMCI, bem como a realização de verificações que podem auxiliar na investigação e detecção de problemas relacionados a compatibilização entre os ativos da rede. Desta forma, a ferramenta se mostrou útil para atividades de desenvolvimento, podendo auxiliar analistas e engenheiros na elaboração e manutenção de produtos com tecnologia GPON.

5.1 Trabalhos Futuros

Alguns trabalhos futuros podem ser indicados visando a evolução da ferramenta desenvolvida. Como a ferramenta possui dois componentes, os trabalhos foram divididos entre propostas de trabalhos futuros no capturador e no *software* de análise.

5.1.1 No capturador

No capturador é possível destacar as seguintes propostas de trabalhos futuros:

Capturar fluxo *upstream*: uma análise superficial no *hardware* do capturador mostrou que, em princípio, seria possível capturar mensagens no sentido *upstream*, desde que o módulo óptico fosse alterado para um outro cujo comprimento de onda fosse 1310 nm. Além disso, seria necessário gerar uma reflexão com *splitters* ópticos. Este trabalho consistiria em analisar esta possibilidade e desenvolvê-la, caso realmente se mostrasse viável.

Otimização no fluxo das mensagens OMCI: conforme descrito na seção 3.1, as mensagens OMCI detectadas são tratados em espaço de usuário e encaminhadas via *socket* para a porta *ethernet*. Uma sugestão de trabalho futuro é realizar uma otimização neste fluxo, fazendo com que as mensagens não sejam encaminhadas à CPU do dispositivo, sendo redirecionados já em espaço de *kernel* para a interface *ethernet*. Pelo que foi analisado neste trabalho, tal otimização poderia ser feita diretamente nos *drivers* do equipamento, porém seriam necessárias alterações em nível de *kernel*, o que exigiria um cuidado com relação a tamanho dos *buffers*, enfileiramento, concorrência, entre outros pontos críticos a serem observados.

Capturar outros tipos de mensagens: as alterações no *firmware* da ONU realizadas nesta monografia tiveram como objetivo somente a captura das mensagens OMCI. Porém há outros tipos de mensagens que são recebidas por uma ONU de uma rede GPON, como mensagens PLOAM e os próprios pacotes de dados. Sendo assim, esta proposta sugere a realização de novas modificações no *firmware* da ONU, com o objetivo de capturar também esses outros tipos de mensagens, além do OMCI. A captura de tais

mensagens poderia auxiliar nas análises executadas pela ferramenta, fornecendo mais dados para a sua realização.

Desenvolver o *hardware* do capturador usando **FPGA (*Field Programmable Gate Array*):** esta proposta consiste em desenvolver um *hardware*, em princípio utilizando **FPGA**, que faça o trabalho do capturador. Este *hardware* deve se basear nas recomendações G.984.2 (ITU-T, 2003) e G.984.3 (ITU-T, 2014) para implementar as camadas **PMD** e **GTC** da tecnologia **GPON**. Desta forma, não seria mais necessária a utilização da **ONU** como capturador, proporcionando uma maior flexibilidade de *hardware* e possibilitando evoluções para as tecnologias subsequentes ao **GPON**. Além disso, o desenvolvimento de um *hardware* específico para a aplicação do capturador, possibilitaria aumentar o número de **ONUs** monitoradas, podendo identificar mensagens recebidas por qualquer **ONU** do barramento, ampliando as possibilidades de análise e verificação da ferramenta.

5.1.2 No *software* analisador

No *software* analisador é possível destacar as seguintes propostas de trabalhos futuros:

Melhorias gerais no *software*: algumas melhorias ou novas funcionalidades podem ser inseridas no *software* desenvolvido, visando entregar mais recursos aos usuários. A lista a seguir demonstra alguns exemplos:

- **Analisar mensagens no sentido *upstream*:** o *software* de análise verifica somente mensagens no sentido *downstream* devido a limitação do *hardware* capturador. Entretanto, por serem componentes separados, o *software* de análise poderia analisar também as mensagens *upstream*. Isso porque ele é capaz de analisar qualquer captura **OMCI** que esteja no formato *.pcap*, ou seja, caso essa captura tenha sido realizada por outro sistema que capture mensagens *upstream*, o *software* de análise poderia analisá-las também;
- **Compatibilizar com outros **SOs** (Windows e MAC):** atualmente o *software* é compatível somente com sistemas Linux. Esta sugestão propõe compatibilizá-lo com outros **SOs**;
- **Criar uma interface gráfica:** permitir a execução do *software* em modo gráfico, pois atualmente ele só é executado via terminal;
- **Aumentar a base de **MEs**:** a base utilizada nesta monografia possui cerca de 90 **MEs**, porém a última versão da ITU-T G.988 (ITU-T, 2017) possui mais de 300. Esta sugestão propõe o incremento dessa base.

Desenvolver mais verificações: atualmente o *software* realiza três verificações nas *MEs*. Uma proposta seria ampliar o número de verificações. Para fazer isto seriam necessárias algumas modificações para que o *software* não realizasse somente verificações na própria *ME*, mas que fosse possível realizar verificações entre *MEs* diferentes. Com isso seria possível realizar também verificações de dependências entre entidades, ampliando a abrangência de análise do *software*.

REFERÊNCIAS

- ABRANET. *Empresas de Internet somam 2,44 milhões de assinantes e chegam a 30% de market share*. 2019. Disponível em: <<http://www.abranet.org.br/Noticias/Empresas-de-Internet-somam-2%2C44-milhoes-de-assinantes-e-chegam-a-30%25-de-market-share-2620.html>>. Acesso em: 21 abr 2020. Citado na página 18.
- ABRANET. *Fibra ótica supera acessos x.DSL e assume ponta na banda larga fixa*. 2020. Disponível em: <<http://www.abranet.org.br/Noticias/Fibra-otica-supera-acessos-x.DSL-e-assume-ponta-na-banda-larga-fixa-2781.html?UserActiveTemplate=site#.XqIwtcZKhNw>>. Acesso em: 21 abr 2020. Citado na página 18.
- ANATEL. *Anatel divulga relatório sobre o mercado de banda larga brasileiro*. 2019. Disponível em: <<https://www.anatel.gov.br/institucional/noticias-destaque/2230-anatel-divulga-relatorio-sobre-o-mercado-de-banda-larga-brasileiro>>. Acesso em: 21 abr 2020. Citado na página 18.
- CEDRIC, L. *Passive optical networks: principles and practice*. [S.l.: s.n.], 2007. Citado 5 vezes nas páginas 21, 22, 23, 24 e 25.
- IEEE. *Ieee standard for local and metropolitan area networks—port-based network access control. IEEE Std 802.1X-2020 (Revision of IEEE Std 802.1X-2010 Incorporating IEEE Std 802.1Xbx-2014 and IEEE Std 802.1Xck-2018)*, p. 1–289, 2020. Citado na página 18.
- INTELBRAS. *ONU 110*. 2020. Disponível em: <<https://www.intelbras.com/pt-br/1-porta-gigabit-ethernet-onu-110>>. Acesso em: 24 jun 2020. Citado na página 39.
- ITU-T, I. recommendation i. 363.5. *B-ISDN ATM adaptation layer (AAL) specification type*, v. 5, 1996. Citado na página 33.
- ITU-T, R. I.432.1. *B-ISDN user-network interface – Physical layer specification: General characteristics*, Feb, 1999. Citado na página 29.
- ITU-T, R. G.984.2. *Gigabit-capable passive optical networks (GPON): physical media dependent (PMD) layer specification*, Mar, 2003. Citado 2 vezes nas páginas 26 e 72.
- ITU-T, R. G.984.1. *Gigabit-capable passive optical networks (GPON): General characteristics*, Mar, 2008. Citado na página 26.
- ITU-T, R. G.984.4. *Gigabit-Capable Passive Optical Networks (G-PON): ONT Management and Control Interface Specification*, Feb, 2008. Citado 5 vezes nas páginas 27, 32, 34, 35 e 36.
- ITU-T, R. G.984.3. *G. 984.3: Gigabit-capable Passive Optical Networks (G-PON): Transmission Convergence Layer Specification*, Jan, 2014. Citado 7 vezes nas páginas 26, 27, 28, 30, 31, 33 e 72.
- ITU-T, R. G.988. *ONU management and control interface (OMCI) specification*, Nov, 2017. Citado 5 vezes nas páginas 34, 36, 37, 44 e 72.

KEISER, G. *FTTX concepts and applications*. [S.l.]: John Wiley & Sons, 2006. v. 91. Citado 3 vezes nas páginas 23, 24 e 25.

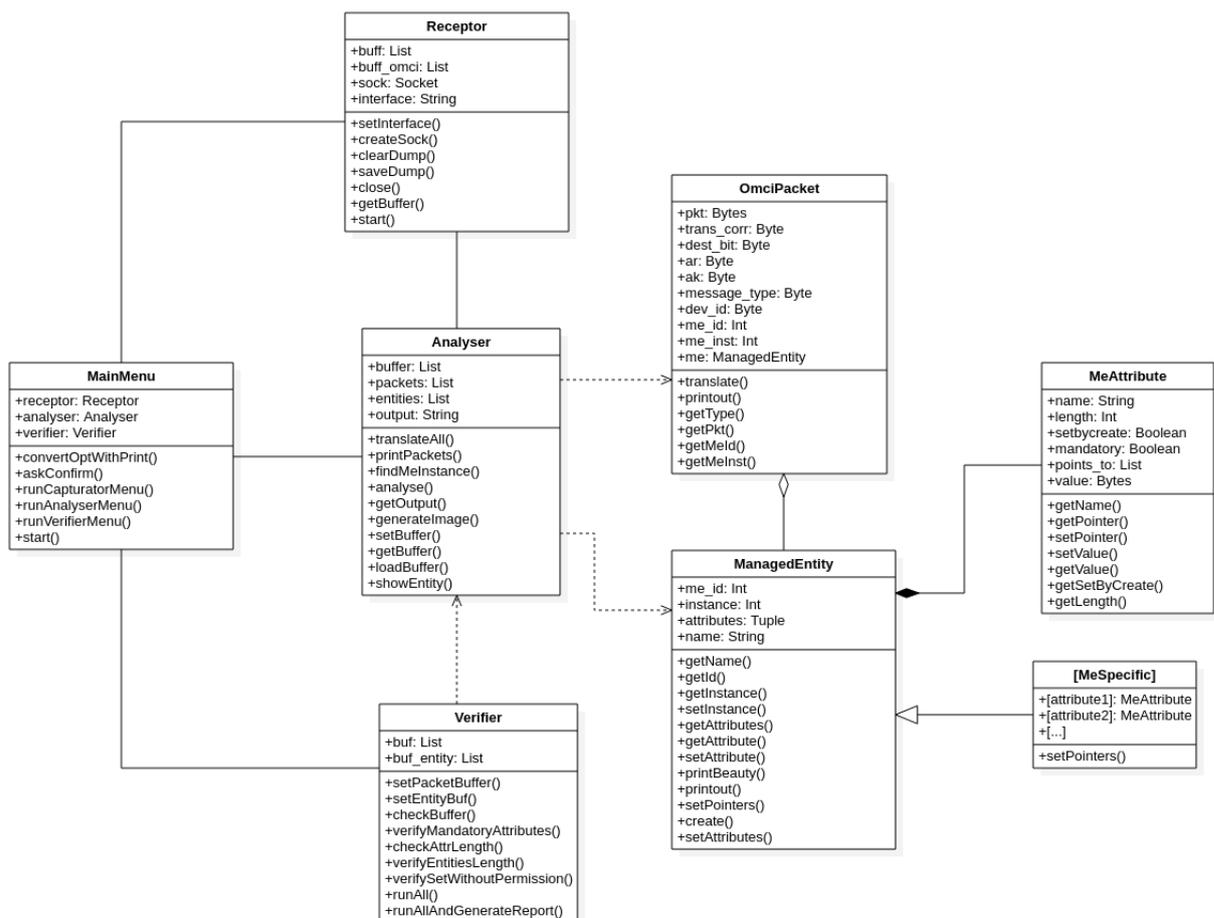
KUROSE, J. F.; ROSS, K. W. *Redes de computadores e internet: uma abordagem top-down. tradução Daniel Vieira; revisão técnica Wagner Luiz Zucchi*. [S.l.]: São Paulo. Pearson Addison Wesley, 2013. Citado 2 vezes nas páginas 21 e 22.

Apêndices

APÊNDICE A – DIAGRAMA DE CLASSES

A Figura 34 exibe o diagrama de classes do *software* analisador, que é executado no PC de análise. A classe *MainMenu* é responsável por chamar as classes *Receptor*, *Analysier* ou *Verifier*, de acordo com a opção selecionada pelo usuário, sendo essas opções: capturar (receber os pacotes do capturarador em tempo real), analisar (gerar diagrama de correlação de entidades) ou verificar (gerar relatório de conformidade), respectivamente. As mensagens capturadas são mapeadas na classe *OmciPacket*, das quais são geradas as MEs, de acordo com a classe *ManagedEntity*. A classe *ManagedEntity* é abstrata e é estendida por cada classe de ME proveniente da base de MEs, representadas no diagrama como *MeSpecific*. As MEs possuem atributos representados pela classe *MeAttribute*.

Figura 34 – Diagrama de classes do software analisador



FONTE: Elaborada pelo autor.