

Utilização da Lógica Fuzzy em Redes PON Definidas por Software

Ricardo Girnis Tombi, Michael Prieto Hernández, Regina Melo Silveira

Resumo—As redes ópticas passivas (PON) são cada vez mais utilizadas pelas operadoras de telecomunicações como redes de acesso, e possuem mecanismos de controle de recursos bem definidos para o tráfego *downstream* e *upstream*. Entretanto, a integração de outras tecnologias como a lógica Fuzzy e as redes definidas por software (SDN – *Software Defined Network*), podem ser utilizadas como alternativa para otimizar ou aumentar o desempenho destas redes de acesso. Este trabalho demonstra a utilização da lógica Fuzzy para alocação de recursos em redes PON, e apresenta como a mesma pode ser flexível e de simples implementação em conjunto com uma arquitetura SDN considerando tipos distintos de tráfego gerados por aplicações de IoT (*Internet of Things*), voz e vídeo, que apresentam necessidades específicas. Os resultados obtidos comprovam a viabilidade da lógica Fuzzy para subsidiar o controle de alocação de recursos, garantindo a qualidade de serviços em redes PON.

Palavras-Chave—PON, Redes Definidas por Software, Lógica Fuzzy.

Abstract—Passive optical networks (PON) have been increasingly used by telecom operators as access networks and have well-defined resource control mechanisms for downstream and upstream traffic. However, the integration of other technologies such as Fuzzy logic and software defined networks (SDN) can be used as an alternative to optimize or increase the performance of these access networks. This work demonstrates the use of Fuzzy logic for resource allocation in PON networks, and presents how it can be flexible and simple implementation in conjunction with an SDN architecture considering different types of traffic generated by IoT (*Internet of Things*), voice and video applications, which have specific needs. The results obtained prove the feasibility of Fuzzy logic to subsidize the control of resource allocation, guaranteeing the quality of services in PON.

Keywords—PON, Software Defined Networks, Fuzzy Logic.

I. INTRODUÇÃO

As redes ópticas passivas (PON) são atualmente uma alternativa importante para as operadoras de telecomunicações como redes de acesso, pois podem oferecer grande capacidade de recursos aos usuários da Internet cujo tráfego cresce cada vez mais devido a aplicações como vídeos sob demanda e de alta definição, IoT, jogos interativos e VoIP (*Voice over Internet Protocol*). As arquiteturas de redes PON são definidas através de um conjunto de padronizações em órgãos como IEEE e ITU-T, e seus principais componentes são: i) *Optical Network Unit* (ONU), que agrega o tráfego dos usuários proporcionando o acesso dos mesmos à rede PON; ii) *Optical Line Terminal* (OLT), que aloca recursos para as ONUs em um

canal compartilhado e faz a interface entre a PON e o *backbone* da rede; iii) *Optical Distribution Network* (ODN) é a infraestrutura, composta pelas fibras ópticas e *splitters*, que conecta as ONUs às OLTs nos provedores. A figura 1 apresenta a topologia de uma PON típica.

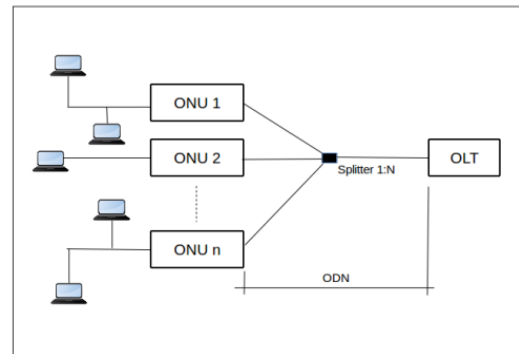


Fig. 1. Rede PON

Entretanto a infraestrutura apresentada não possui a flexibilidade necessária para promover inovações rápidas e para provisionar novos recursos no tempo esperado que as aplicações atuais demandam [1-3]. Apesar das evoluções tecnológicas das redes PON, o atendimento aos parâmetros de Qualidade de Serviço (QoS) ainda são um desafio [4-5]. A estrutura destas redes é complexa e rígida, baseada em um conjunto de protocolos e algoritmos distribuídos em equipamentos proprietários, que não oferecem o dinamismo que os usuários requerem, como já acontece em outros ambientes virtualizados como os *data centers*. O trabalho de Yang [6] exemplifica de que modo as redes PON atuais são dependentes de configurações manuais e periódicas, gerando custos de operação e manutenção, além de não garantir completamente o atendimento de QoS das aplicações, subutilizando recursos, pois os fluxos dos diferentes clientes não podem ser atendidos através de um controle de alocação inteligente, dinâmico e automático.

Dado este cenário, a proposta deste trabalho é apresentar uma nova arquitetura para redes PON por meio de controle SDN com algoritmo para alocação de banda baseado na aplicação da teoria da lógica Fuzzy.

Este trabalho está estruturado da seguinte maneira: a seção II apresenta como as redes PON podem se beneficiar do paradigma das redes definidas por software, a seção III relaciona trabalhos correlatos, a seção IV detalha a proposta do

modelo do controlador baseado na lógica Fuzzy, a seção V apresenta os resultados experimentais obtidos com a utilização do controlador para alocação de recursos, e a seção VI apresenta a conclusão e as considerações finais.

II. REDES PON CONTROLADAS POR SOFTWARE

As redes PON podem se beneficiar do uso do paradigma de uma rede SDN para melhorar sua flexibilidade e proporcionar a dinâmica necessária exigida pelas mudanças de perfil de tráfego das aplicações atuais. Por meio desta nova estrutura, os equipamentos baseados em sistemas fechados e proprietários, gerenciados por um provedor de acesso por meio de configurações estáticas, darão lugar a um sistema com um controle mais aberto e com visão mais abrangente dos elementos da rede. Além da possibilidade de configurações automáticas, é esperada redução de custos de operação e manutenção de uma estrutura que atualmente opera de forma descentralizada, com grande quantidade de dispositivos.

Usuários podem ter mais poder sobre suas aplicações através da possibilidade de interagir com a rede por uma interface usuário-rede, como um aplicativo intuitivo, que fornece ao mesmo a escolha do serviço e respectiva qualidade de experiência (QoE – *Quality of Experience*). Os usuários também podem ser informados sobre os parâmetros de execução de sua aplicação para, em seguida, ajustar as suas escolhas mantendo um ciclo de realimentação com este objetivo. Segundo Kerpez [3], há muitas funções de controle e gerenciamento necessárias para o acesso em banda larga, que, como em uma SDN, podem migrar de *firmwares* embutidos em equipamentos de rede dedicados, para controladores de *software* executados em *hardware* padrão de mercado, em uma nuvem privada ou pública, economizando custos e aumentando as capacidades.

III. TRABALHOS RELACIONADOS

Há trabalhos desenvolvidos que apresentam algumas formas de como redes PON podem ser inseridas em uma arquitetura SDN. O trabalho realizado por Chitimalla [4] propõe uma solução para o problema de como tratar de forma diferenciada os fluxos de uma OLT para suas ONUs, de acordo com as demandas dos usuários e aplicações. No mecanismo proposto um controlador SDN é conectado entre as ONUs e a OLT. Uma aplicação lê os *buffers* dos usuários e envia os dados para o controlador SDN que executa o algoritmo *Application-Aware SDN-Enabled Resource Allocation* (AASRA) para alocar novos pesos para as filas da OLT. Desta forma a OLT recebe a informação do controlador e implementa os novos pesos para atendimento diferenciado das filas. O trabalho realizado por Yang [6] desenvolve uma interface para a OLT se comunicar com o controlador OpenFlow, o qual também está conectado a uma interface para monitoração da aplicação do usuário. No mecanismo proposto, após receber o *status* do fluxo, o controlador decide como distribuir e ajustar a largura de banda baseada em demandas de QoS de diferentes prioridades, entretanto não é especificado no artigo as regras utilizadas para esta decisão. Com relação à lógica Fuzzy os autores de [7] propõe um algoritmo utilizando Fuzzy para alocação de banda em redes PON de acordo com as necessidades de QoS das aplicações, entretanto fora do ambiente SDN. De acordo com a revisão da literatura não é do nosso conhecimento um trabalho que utilize o paradigma SDN com seu controlador baseado em lógica Fuzzy em redes PON, sendo assim este trabalho deve cobrir esta lacuna.

IV. MODELO PROPOSTO PARA O CONTROLADOR SDN-FUZZY

Um controlador Fuzzy é baseado na lógica Fuzzy, uma extensão da lógica booleana tradicional [8]. Esta lógica de controle é adequada para sistemas não lineares e sistemas onde conjuntos nebulosos podem ser utilizados no lugar dos tradicionais [9], e onde os controles tradicionais utilizam equações complexas e inflexíveis, não permitindo ajustes de acordo com o dinamismo do fenômeno a ser controlado e a percepção dos desenvolvedores e operadores.

A lógica Fuzzy é representada por números e conjuntos Fuzzy, e operações matemáticas e lógicas com estes elementos para se obter o controle de um sistema. O controlador Fuzzy opera através de processos denominados fuzzificação, inferência e defuzzificação. Na fuzzificação o valor de uma variável de entrada do sistema é convertida em valores linguísticos como alto, baixo, muito, pouco, e são representados por conjuntos Fuzzy. Cada um destes conjuntos Fuzzy são delineados em torno de certos limites e são definidos através de funções de pertencimento que determinam quanto uma variável de entrada pertence àquele conjunto. Quanto mais se aproxima de 1 mais certo é que aquela variável pertença a um determinado conjunto, e na aproximação de 0 o oposto ocorre. Uma característica importante da lógica Fuzzy é sua flexibilidade relacionada com o grau de pertencimento de uma variável, que pode assumir valores de pertencimento em mais de um conjunto, fazendo com que decisões mais adequadas sejam tomadas pelo controlador. Para maiores detalhes sobre a lógica Fuzzy o leitor é recomendado a verificar [8, 10-13].

O controlador proposto neste trabalho segue a arquitetura da figura 2, e é composto por: uma interface de entrada que recebe as variáveis de entrada do sistema; um processo de fuzzificação, que transforma os termos linguísticos em variáveis Fuzzy, por meio de conjuntos Fuzzy e funções de pertinência; uma máquina de inferência que aplica as regras Fuzzy e determina as ações que o controlador vai executar; um processo de defuzzificação que vai gerar um valor de saída a ser aplicado ao sistema, através da interface de saída.

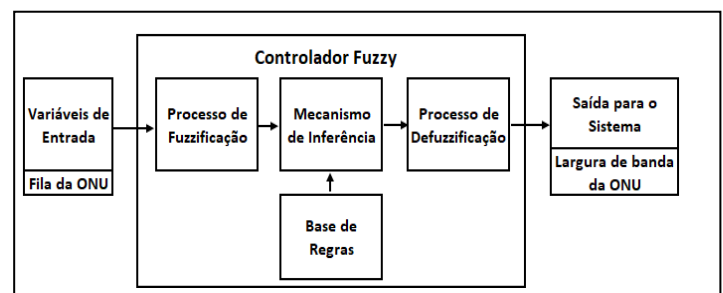


Fig. 2. Controlador Fuzzy

O grau de sensibilidade aos requisitos de QoS depende de cada aplicação, desta forma os parâmetros definidos para esta arquitetura são ajustados para garantir QoS de aplicações sensíveis ao atraso como voz e vídeo interativos, cujos valores de atraso não devem superar 150ms [14].

Os conjuntos Fuzzy são definidos por meio de funções de pertinência relevantes ao ambiente. Importante destacar que podem ser modificados de acordo com a situação e conhecimento do ambiente, proporcionando flexibilidade ao sistema. A tabela 1 apresenta as definições das variáveis de entrada e saída do controlador.

TABELA I. DEFINIÇÕES DAS VARIÁVEIS DE ENTRADA E SAÍDA

Variável	Entrada	Saída
Descrição	Ocupação do <i>buffer</i> da ONU	Alocação de banda da ONU
Unidade	%	%
Símbolo	F_{Tx} (T: termo linguístico; x: id da ONU)	B_{Tx} (T: termo linguístico; x: id da ONU)
Universo de Discurso	$U(F_{Tx}) = [0, 100]$	$U(B_{Tx}) = [0, 100]$
Termos Linguísticos	$T = \{\text{muito_baixa, baixa, media, alta, muito_alta}\}$	$T = \{\text{muito_baixa, baixa, media, alta, muito_alta}\}$

As variáveis de entrada e saída são definidas através dos seus conjuntos Fuzzy demonstrados pelo teorema 1, que tem como objetivo a manutenção do baixo nível de ocupação das filas e a otimização do slot de transmissão das ONUs.

Teorema 1 (Conjuntos Fuzzy): Considere o sistema,

$$F_{\text{muito_baixa}}(x) = \begin{cases} 1 & , x \leq 0,1 \\ -10x+2 & , 0,1 \leq x \leq 0,2 \end{cases} \quad (1)$$

$$F_{\text{baixa}}(x) = \begin{cases} 10x-1 & , 0,1 \leq x \leq 0,2 \\ -5x+2 & , 0,2 \leq x \leq 0,4 \end{cases} \quad (2)$$

$$F_{\text{media}}(x) = \begin{cases} 5x-1 & , 0,2 \leq x \leq 0,4 \\ -1,67x+1,67 & , 0,4 \leq x \leq 1 \end{cases} \quad (3)$$

$$F_{\text{alta}}(x) = \begin{cases} 1,67x-0,67 & , 0,4 \leq x \leq 1 \\ -0,11x+1,11 & , 1 \leq x \leq 10 \end{cases} \quad (4)$$

$$F_{\text{muito_alta}}(x) = \begin{cases} 0,07x-0,67 & , 10 \leq x \leq 25 \\ 1 & , x \leq 25 \end{cases} \quad (5)$$

$$B_{\text{muito_baixa}}(x) = \begin{cases} 1 & , x \leq 0,1 \\ -0,5x+1,5 & , 1 \leq x \leq 3 \end{cases} \quad (6)$$

$$B_{\text{baixa}}(x) = \begin{cases} 0,5x-0,5 & , 1 \leq x \leq 3 \\ -0,05x+1,14 & , 3 \leq x \leq 25 \end{cases} \quad (7)$$

$$B_{\text{media}}(x) = \begin{cases} 0,05x-0,14 & , 3 \leq x \leq 25 \\ -0,04x+2 & , 25 \leq x \leq 50 \end{cases} \quad (8)$$

$$B_{\text{alta}}(x) = \begin{cases} 0,04x-1 & , 25 \leq x \leq 50 \\ -0,05x+3,5 & , 50 \leq x \leq 70 \end{cases} \quad (9)$$

$$B_{\text{muito_alta}}(x) = \begin{cases} 0,05x-2,5 & , 50 \leq x \leq 70 \\ 1 & , x \geq 70 \end{cases} \quad (10)$$

No processo de inferência sobre os valores fuzzificados das variáveis de entrada um conjunto de regras é aplicado seguindo o modelo *if-then*, sendo importante destacar que devido à característica dos distintos graus de pertencimento das variáveis há a possibilidade do acionamento de várias regras de forma simultânea pois uma variável pode pertencer a mais de uma regra com seus devidos graus de pertencimento. Após a aplicação das regras de inferências um conjunto de variáveis de saída é obtido.

A variável de saída final é obtida através da agregação (método min) do conjunto de saída, sendo que o gráfico resultante é defuzzificado através do método COG (*Center Of Gravity*). O método do centro de gravidade é o mais utilizado em sistemas Fuzzy pois fornece como resposta o valor mais adequado do gráfico de saída através da equação a seguir, onde u é o valor da variável de saída e $\mu(u)$ o seu grau de pertinência [10].

$$U = \frac{\int_{\min}^{\max} u \cdot \mu(u) \, du}{\int_{\min}^{\max} \mu(u) \, du} \quad (11)$$

V. RESULTADOS EXPERIMENTAIS

O modelo proposto na seção anterior foi implementado através de uma rede EPON simulada em código Java. O algoritmo do controlador SDN baseado na lógica Fuzzy foi desenvolvido com a utilização da biblioteca *jFuzzyLogic* [15-16], e inserido na rede simulada para gerenciar os recursos no canal compartilhado entre as ONUs e a OLT. A rede simulada está demonstrada na figura 3, e parte do código é apresentado no apêndice I.

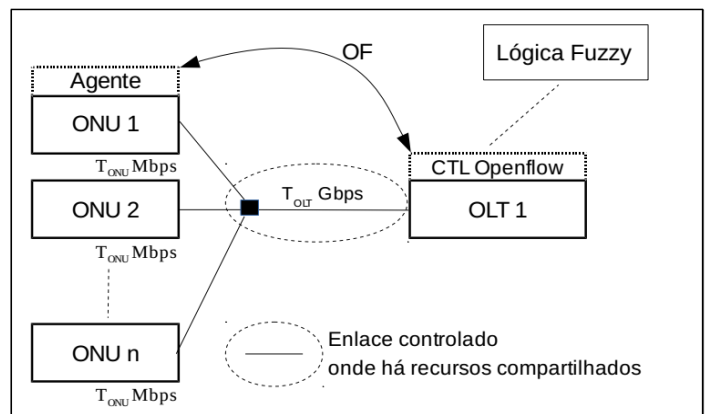


Fig. 3. Topologia da rede PON simulada

O canal compartilhado pelas ONUs (T_{OLT}) é de 1Gbps e o *buffer* de cada ONU é finito e possui 8MB de memória.

A figura 4 apresenta o atraso médio e máximo dos pacotes nas ONUs, de acordo com o aumento da carga nas suas

entradas. O tráfego gerado pelas ONUs (T_{ONU}) para ser enviado à OLT segue distribuição de Pareto, e enquanto $T_{OLT} \geq N \times T_{ONU}$ não haverá problema de sobrecarga do canal, pois o mesmo não terá seu limite excedido.

Conforme detalhado na descrição do modelo, o controlador é ajustado para atender QoS de tráfegos sensíveis ao atraso, e os resultados apresentados demonstram que os atrasos médios não superam 4ms e os máximos estão abaixo de 11ms (enquanto T_{OLT} não é excedido - % ocupação $T_{OLT} < 1$), satisfazendo às necessidades exigidas.

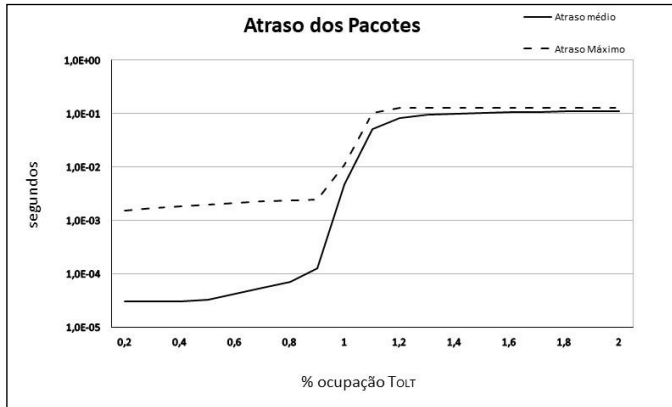


Fig. 4. Atrasos médio e máximo de entrega dos pacotes

Variações deste desempenho podem ser obtidas quando são modificadas as variáveis de entrada e os conjuntos Fuzzy, demonstrando a flexibilidade do controlador. Com esta característica, o controlador pode diferenciar as ONUs da rede com prioridades distintas, de acordo com seu SLA (*Service Level Agreement*). A prioridade pode ser realizada por meio da configuração das variáveis e conjuntos Fuzzy específicos para cada uma das ONUs adaptando-as às situações diversas.

Para demonstrar esta flexibilidade, outra simulação foi realizada onde o controlador Fuzzy se adapta ao novo cenário, e os resultados mais importantes são observados quando há sobrecarga do canal compartilhado ($T_{OLT} \leq N \times T_{ONU}$). Nesta situação seriam esperados valores semelhantes dos tempos médios e máximos de atraso, e das perdas de pacotes para as ONUs. Entretanto os resultados mostram valores menores para uma ONU com prioridade, onde um SLA é superior aos demais.

De acordo com a figura 5 pode-se notar que a ONU-A possui tempos nitidamente inferiores quando a carga oferecida à rede está entre 50% e 100%.

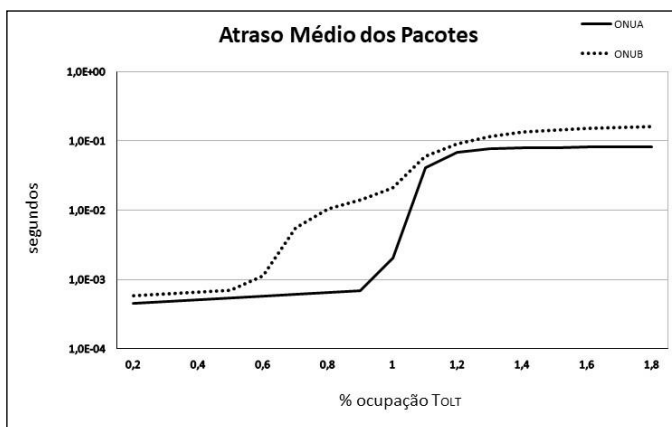


Fig. 5. Atrasos médios de entrega dos pacotes com $T_{OLT} \leq N \times T_{ONU}$

Pode-se notar também que acima da capacidade de T_{OLT} o atraso máximo dos pacotes da ONU-A continua menor apesar da diferença não ser tão acentuada. Desta forma fica demonstrada que a prioridade desejada foi obtida.

A figura 6 apresenta as perdas de pacotes quando há sobrecarga do tráfego no canal compartilhado excedendo a capacidade de T_{OLT} . A perda de pacotes na ONU-A é menor ao se comparar com a ONU-B nesta situação crítica, atendendo às expectativas de acordo com a prioridade de SLA dada para a ONU-A neste cenário. A diferença vai de 7198 pacotes perdidos a mais pela ONU-B quando a carga supera 20% de T_{OLT} e alcança 28696 pacotes perdidos a mais quando a carga supera 80% do total de T_{OLT} .

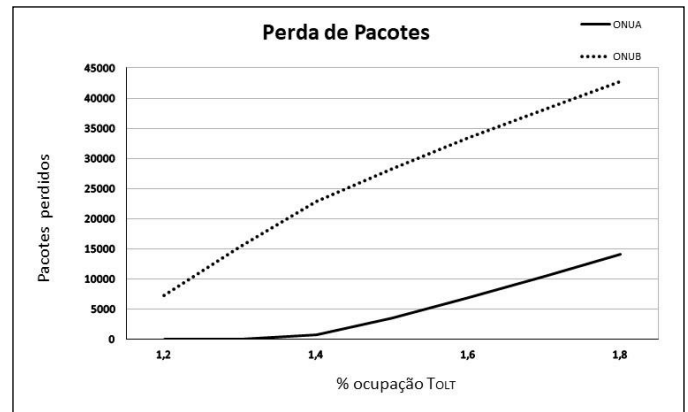


Fig. 6. Perda de pacotes com $T_{OLT} \leq N \times T_{ONU}$

VI. CONCLUSÕES

Neste trabalho foi desenvolvido um algoritmo para alocação de banda em redes PON baseado na lógica Fuzzy e apresentada a viabilidade de sua utilização através dos resultados obtidos em termos de atrasos médio, máximo e perda de pacotes, principalmente quando há aumento da carga na rede. Fator importante é sua flexibilidade para a manipulação de suas variáveis e simplicidade na construção, características que facilitam sua adoção em redes PON definidas por software, disponibilizando ao controlador da rede a possibilidade de alteração dos parâmetros e resultados de forma simples, através de sua característica linguística de acordo com a dinâmica da rede, para atender demandas das aplicações atuais. Esta característica foi demonstrada através da simulação de prioridades para o atendimento das ONUs por meio de SLAs distintos, resultando em desempenho superior em situação de sobrecarga da rede.

REFERÊNCIAS

- [1] N. Cvijetic, *SDN for Optical Access*, In: Advanced Photonics for Communications, OSA Technical Digest (online) (Optical Society of America, 2014), paper PM3C.4.
- [2] N. Bitar, *Software Defined Networking and Applicability to Access Networks*, In: Optical Fiber Communications Conference and Exhibition (OFC). IEEE, 2014.
- [3] K. Kerpez et al, *Software-Defined Access Network (SDAN)*, In: IEEE Communications Magazine, v. 52, n. 3, p. 152-159. IEEE, 2014
- [4] D. Chitimalla et. Al, *Application-Aware Software-Defined EPON Access Network*, In: IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, 2014
- [5] K. Li et. al, *QoE-based Bandwidth Allocation with SDN in FTTH Networks*, In: Network Operations and Management Symposium (NOMS). IEEE, 2014
- [6] H. Yang, *Experimental Demonstration of Remote Unified Control for OpenFlow-based Software Defined Access Optical Networks*, In: 39th

- European Conference and Exhibition on Optical Communication (ECOC 2013)
- [7] S. Hwang et al, *Fuzzy Logic Embedded in Prediction-Based DBA for Differentiated Services on EPONs*, Tamkang Journal of Science and Engineering, Vol. 12, No. 1, pp. 73-84 (2009)
- [8] A.L. Zadeh, *Fuzzy logic = computing with words*, In: IEEE transactions on fuzzy systems, v. 4, n. 2, p. 103-111. IEEE, 1996
- [9] M.G. Simões e I.S. Shaw, "Controle e modelagem Fuzzy". Ed. Blucher. 2a Ed. (2007)
- [10] IEC, *IEC 61131-7 Programmable controllers - Part 7: Fuzzy control programming*, International Standard. IEC - International Electrotechnical Commission (2000)
- [11] Matlab, *Fuzzy Logic Toolbox*, The Math Works Inc. Natick. MA, 2017
- [12] N.K. Kasabov, *Foundations of neural networks, fuzzy systems, and knowledge engineering*. MIT. 2a Ed, 1998
- [13] L.B. Neto et al, "Minicurso de sistema especialista nebuloso". In: XXXVIII Simpósio brasileiro de pesquisa operacional, 2006
- [14] J.F. Kurose e K.W. Ross, "Redes de Computadores e a Internet: uma abordagem top/down". Pearson Education do Brasil, 6ª ed. São Paulo, 2013
- [15] P. Cingolani e J. Alcalá-Fdez, *jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming*, In: International Journal of Computation Intelligence Systems, v. 7, s. 1, p. 61-75, 2013
- [16] P. Cingolani e J. Alcalá-Fdez, *jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation*, Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference.

```
// Aloca valores de banda de acordo com as
// variáveis obtidas da lógica Fuzzy

fb.evaluate();

transm_onu[0] = (int)
((fb.getVariable("onua").getValue() * 0.01) *
remaining_dba_size);
transm_onu[1] = (int)
((fb.getVariable("onub").getValue() * 0.01) *
remaining_dba_size);

for (int id = 0; id < EponSimulator.num_onus;
id++) {
    report = report_array.getReport(id);
    if (report != null) {
        report.onu_tsize = transm_onu[id];
        report_array.overall_tsize +=
report.onu_tsize;
    } else {
        report = new Report(id, 0);
        report.onu_tsize = EponSimulator.REPORT_SIZE;
        report_array.addReport(report);
    }
}
}
```

APÊNDICE I

Parte do código que implementa a lógica Fuzzy através da biblioteca *jFuzzyLogic*, utilizada pelo simulador em Java.

```
import java.util.ArrayList;
import antlr.collections.List;
import net.sourceforge.jFuzzyLogic.FIS;
import net.sourceforge.jFuzzyLogic.FunctionBlock;
import net.sourceforge.jFuzzyLogic.plot.JFuzzyChart;
import net.sourceforge.jFuzzyLogic.rule.Variable;

public class FuzzyLogic {

    public static void fuzzyAllocation(long dba_size,
    ReportArray report_array) throws Exception {

        // carrega o código que executa a lógica Fuzzy
        int fuzzy = 0;
        String filename = "olt_fuzzy.fcl";
        FIS fis = FIS.load(filename, true);

        // carrega bloco de funções padrão Fuzzy
        FunctionBlock fb = fis.getFunctionBlock(null);

        // Configura as entradas do algoritmo
        double fila_onu[] = new double
[EponSimulator.num_onus];
        double utilizacao_fila_onu[] = new double
[EponSimulator.num_onus];
        int transm_onu[] = new int
[EponSimulator.num_onus];
        long remaining_dba_size = dba_size -
(EponSimulator.num_onus -
report_array.num_active_onus) *
EponSimulator.REPORT_SIZE;

        Report report;
        for (int id = 0; id < EponSimulator.num_onus; id++) {
            report = report_array.getReport(id);
            if (report != null) {
                fila_onu[id] = report.onu_qsize;
            }
        }
    }
}
```