

# Topologias virtuais confiáveis considerando múltiplos critérios para o contexto borda-como-serviço

Rafael L. Gomes<sup>1,2</sup>, Luiz F. Bittencourt<sup>2</sup>, Edmundo M. R. Madeira<sup>2</sup>

<sup>1</sup>Centro de Ciência e Tecnologia (CCT) - Universidade Estadual do Ceará (UECE)

<sup>2</sup>Instituto de Computação (IC) - Universidade Estadual de Campinas (UNICAMP)

{rafaellgom, bit, edmundo}@ic.unicamp.br

**Abstract.** *Nowadays, the Internet is part of our lives, being an essential way of communication. However, it suffers from limitations that prevent the Internet to guarantee Quality of Service (QoS) to the users. Recently, the scientific community presented the Edge as a Service (EaaS) paradigm as a possible approach to improve the Internet access service. EaaS deploys Virtual Networks (VNs) over Software Defined Networks (SDNs) to ease the management of Internet Service Providers (ISPs). The EaaS needs an algorithm to define virtual topologies for the VNs considering the resource utilization, the energy consumption and the service delivery reliability. Within this context, this paper presents the Fuzzy for Allocation (FUZA) algorithm to define reliable virtual topologies based on a fuzzy system considering energy consumption and bandwidth availability. The results suggest that the proposed algorithm can deploy reliable VNs, while improving bandwidth utilization and energy consumption.*

**Resumo.** *Atualmente a Internet faz parte do cotidiano das pessoas, sendo um meio de comunicação essencial. Contudo, esta sofre de limitações que impedem a Internet de garantir Qualidade de Serviço (QoS) aos usuários. Recentemente, a comunidade científica apresentou o paradigma Borda como Serviço (EaaS) como uma possível abordagem para melhorar o serviço de acesso a Internet. EaaS implanta redes virtuais (VNs) sobre Redes Definidas por Software (SDN) para facilitar o gerenciamento de Provedores de Internet (ISPs). O paradigma EaaS necessita de um algoritmo para gerar topologias virtuais para as VNs considerando a utilização de recursos, consumo de energia e a confiabilidade do serviço prestado. Dentro deste contexto, este artigo apresenta o algoritmo Fuzzy como base para Alocação (FUZA) para definir topologias virtuais confiáveis baseado em um sistema fuzzy que considera o consumo de energia e a disponibilidade de largura de banda. Os resultados sugerem que o algoritmo proposto pode implantar VNs confiáveis, enquanto melhora a utilização de largura de banda e o consumo de energia.*

## 1. Introdução

A sociedade vem adaptando a abordagem tradicional de comunicação baseada em chamadas de voz e/ou mensagens de texto para chamadas de vídeo em tempo real e/ou compartilhando informações em redes sociais. Porém, os Provedores de Internet (*Internet Service Providers* - ISPs) atuais não conseguem garantir Qualidade de Serviço (*Quality*

of Service - QoS) para este novo aspecto da Internet, o qual gera uma demanda elástica de recursos de rede no decorrer do dia.

Pesquisadores pelo mundo têm investigado formas de lidar com este novo cenário. A Borda como Serviço (*Edge as a Service* - EaaS) é uma abordagem proposta para melhorar a capacidade das redes de borda em prover acesso à Internet [Davy et al. 2014]. EaaS implanta Redes Virtuais (*Virtual Networks* - VNs) sobre Redes Definidas por Software (Software Defined Network - SDN) para proporcionar flexibilidade e gerenciabilidade à alocação de recursos e customização do comportamento da rede.

Tradicionalmente, os ISPs visam maximizar seus lucros, e duas métricas são diretamente relacionada a isso: (i) número de clientes e (ii) consumo de energia. O número de clientes é representado pela quantidade de Acordos de Nível de Serviço (*Service Level Agreements* - SLAs) ativos. Assim, quanto mais clientes, maior é o lucro dos ISPs. ISPs podem aumentar o número de clientes a partir de uma melhor utilização da Largura de Banda (*Bandwidth* - Bw) em suas infraestruturas de rede. Por outro lado, consumo de energia da infraestrutura de rede, bem como a Eficiência Energética (*Energy Efficiency* - EE), vêm se tornando pontos cada vez mais importantes a serem considerados pelos ISPs.

Aplicando a abordagem EaaS, os ISPs precisam atender os parâmetros do SLA, onde confiabilidade é um requisito chave para assegurar a QoS para os usuários. Confiabilidade de rede é vista como a probabilidade da rede manter a comunicação mesmo quando falhas ocorrem na infraestrutura [Lee et al. 2010]. Portanto, confiabilidade engloba não apenas ações reativas para gerenciamento pós-falha, mas também um planejamento estratégico pré-falhas, ou seja, fazer o gerenciamento e implantação dos serviços de forma que a ocorrência de falhas não afete a prestação destes serviços aos clientes.

A implantação de VNs é uma tarefa vital no planejamento estratégico necessário para prover um bom acesso a Internet. Uma das tarefas do processo de implantação de VNs é decidir quais componentes da infraestrutura de rede (enlaces e nós) farão parte de cada VN, ou seja, definir a topologia virtual da VN. Esta decisão deve englobar aspectos como planejamento, confiabilidade, eficiência energética, disponibilidade de Bw, dentre outros. Sendo assim, uma abordagem multicritério é capaz de aprimorar o processo de definição da topologia virtual. A ideia de sistemas Fuzzy é comumente usada como base para tomada de decisão com múltiplos critérios.

Dentro deste contexto, este artigo apresenta o algoritmo Fuzzy como base para Alocação (FUZA), o qual define topologias virtuais em EaaS considerando a EE e a Bw disponível no ISP, enquanto considera a confiabilidade necessária para a VN. O objetivo do algoritmo é definir uma topologia virtual para a VN, onde a topologia é planejada para ser confiável e com a Bw exigida (cumprindo os parâmetros do SLA), bem como, simultaneamente, reduzindo a Bw total alocada e o consumo de energia do ISP.

O desempenho do algoritmo *FUZA* foi avaliado e comparado com alguns algoritmos existentes que podem ser aplicados para o contexto de implantação de VNs: RKSP [Eppstein 1994], MSPS [Parandehgheibi et al. 2014], BRAR [Gomes et al. 2016b] and RDP [Gomes et al. 2016a]. Os experimentos realizados avaliaram os benefícios do algoritmo *FUZA* e dos algoritmos existentes. Os resultados mostram a eficiência do algoritmo *FUZA* em definir VNs confiáveis, enquanto aprimora a EE e a utilização de Bw do ISP.

Este artigo está organizado da seguinte forma: a Seção 2 inclui os conceitos

básicos, ou seja, a apresentação do método de confiabilidade considerado, a descrição do contexto EaaS e os trabalhos existentes mais relacionados; a Seção 3 introduz o algoritmo proposto; a Seção 4 apresenta o resultado dos experimentos realizados; e a Seção 5 conclui o artigo e cita os trabalhos futuros.

## 2. Contexto

Nesta seção serão detalhados os conceitos chave para a compreensão deste artigo, englobando EaaS, o método de confiabilidade utilizado e alguns trabalhos relacionados ao processo de implantação de VNs em EaaS.

### 2.1. Edge-as-a-Service (EaaS)

EaaS habilita os provedores de serviço para usar virtualização e construir VNs elásticas [Davy et al. 2014]. EaaS desassocia a relação entre os operadores de rede e suas redes de acesso. Através do desenvolvimento de redes virtuais, pode-se dar suporte a funções adaptativas requisitadas sobre demanda a fim de flexibilizar a entrega de serviços. Em uma abordagem de integração entre SDN e VN, os componentes de hardware podem ser alocados em duas VNs e então acessadas por diferentes controladores de rede. O comportamento de uma VN é definido através da configuração do controlador responsável pela mesma, onde um controlador é responsável por cada VN. A Figura 1 ilustra o cenário EaaS tratado neste artigo, onde duas VNs são implantadas para diferentes clientes. Assim, cada VN é moldada de acordo com as características do cliente.

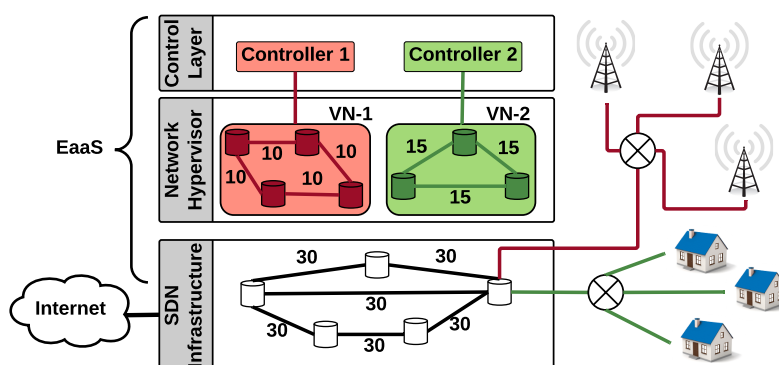


Figura 1. Cenário representando o contexto EaaS.

Os parâmetros da VN englobam algumas métricas, onde o foco deste artigo considera três aspectos: confiabilidade, eficiência energética e largura de banda. A confiabilidade é diretamente relacionada ao número de caminhos alternativos na rede, onde quanto maior é a redundância de caminhos, maior é a confiabilidade da rede. Quando um nó/enlace falha e o caminho primário é afetado, pode-se direcionar o tráfego para o caminho secundário [Lee et al. 2010]. Contudo, não é vantajoso para o ISP definir redundância completa (dois caminhos totalmente disjuntos) para um cliente o qual não deseja tal nível de confiabilidade, visto que isso representa desperdício de recursos.

Um algoritmo é aplicado pelo ISP para definir quais componentes da infraestrutura de rede serão associados a cada VN implantada. Por exemplo, um cliente pode negociar uma VN com um ISP, e este implantar a VN-1 vermelha (Figura 1), que possui Bw de 10 Mbps. O algoritmo proposto neste artigo executa no ISP para implantar as VNs após a especificação do SLA ser finalizada.

Quando parte da rede é comprometida, uma VN pode perder parte dos seus recursos. A definição de uma topologia virtual apropriada pode permitir a operação da VN com uma qualidade mínima definida no SLA através do aumento da Bw alocada no caminho ainda operacional. Dentro deste contexto, o algoritmo *FUZA* proposto neste artigo gera uma topologia virtual confiável que considera tanto consumo de energia quanto a disponibilidade de Bw. Aspectos sobre energia podem ser avaliados sobre diversas perspectivas, neste artigo considerada-se a Eficiência Energética (*Energy Efficiency* - EE) no contexto EaaS, que é definida como o montante de Bw alocada para os clientes em relação a quantidade de energia consumida pela infraestrutura de rede do ISP [Gomes et al. 2016b].

## 2.2. Confiabilidade de Rede

No contexto de implantação de VNs, inicialmente a topologia virtual é determinada, e posteriormente a confiabilidade da topologia gerada é calculada. Este artigo utiliza a confiabilidade da topologia virtual como restrição de adequabilidade da mesma. O método de medição de confiabilidade é aplicado para avaliar se a topologia virtual gerada atende a confiabilidade mínima requisitada. Desta forma, o método de confiabilidade usado é independente do algoritmo de geração da topologia virtual.

Neste artigo foi usado o método proposto por Li et al. [Li and Silvester 1984], o qual possui as seguintes vantagens: (i) foi proposto para redes genéricas (ou seja, este método não usa informações sobre os dispositivos físicos ou uma base de dados pré-definida); (ii) possui uma boa aproximação da confiabilidade com baixo custo computacional; (iii) é flexível para variar o número de falhas na rede; e (iv) é de fácil adaptação para quando se tem uma fonte e vários destinos.

## 2.3. Trabalhos Relacionados

Nesta seção serão descritos os trabalhos chave relacionados a definição de topologias virtuais e SDN, os quais consideram a confiabilidade da rede. A Tabela 1 resume os trabalhos descritos, enfatizando a diferença entre eles e a proposta deste artigo. Cada coluna da Tabela 1 especifica uma característica do trabalho: *Contexto* apresenta o ambiente de rede para qual o trabalho foi projetado, e *Critério* é o conjunto de informações nas quais o trabalho se baseia para implantar VNs.

Soualah et al. [Soualah et al. 2014] propõem uma estratégia para mapeamento de redes virtuais dentro de ambientes em nuvem para aprimorar a recuperação quando ocorrem falhas nos equipamentos (enlaces e roteadores). Adicionalmente, a proposta apresentada na referência [Soualah et al. 2014] visa contornar a complexidade exponencial do mapeamento. Contudo, esta abordagem foca em ambientes de nuvem, além de não considerar aspectos de energia.

Cheng et al. [Cheng et al. 2012] apresentam um método para gerar um *ranking* de nós mais adequados e um algoritmo guloso para combinar nós de uma VN para nós na infraestrutura de rede. Da mesma forma, Mano et al. [Mano et al. 2014] descrevem um método para otimizar a implantação de redes virtuais sob múltiplos domínios, sem revelar informações privadas de cada domínio. Apesar de ambas as propostas focarem em implantação de redes virtuais, elas são baseadas em contextos específicos (capacidade de gerar *ranking* e múltiplos domínios) e não consideram aspectos chave deste artigo como consumo de energia, confiabilidade e planejamento estratégico.

Parandehgheibi et al. [Parandehgheibi et al. 2014] propõem um conjunto de algoritmos para lidar com o problema de capacidade de sobrevivência em redes multi camada. Os autores descrevem o Minimum Survivable Path Set (MSPS), o qual é o número mínimo de caminhos necessários para que um par de nós consiga se comunicar caso uma única falha ocorra na infraestrutura de rede. Portanto, o algoritmo MSPS é uma abordagem para trazer resiliência para a rede, sendo uma solução existente a ser comparada com o algoritmo *FUZA* na seção de experimentos.

Uma abordagem popular para lidar com aspectos de confiabilidade em diversos ambientes de rede é a aplicação do algoritmo Reliable K-Shortest Path (KSP) [Eppstein 1994], o qual encontra o menor caminho e mais outros caminhos adicionais até atingir a confiabilidade desejada.

Em propostas anteriores, os autores desenvolveram dois algoritmos para geração de topologias virtuais de acordo com a confiabilidade desejada: o Bandwidth and Reliability According to Redundancy (BRAR) [Gomes et al. 2016b] e o Relative Disjoint Paths (RDP) [Gomes et al. 2016a]. O algoritmo *BRAR* define caminhos alternativos com a melhor relação entre a confiabilidade de rede e a Bw alocada para a VN. O algoritmo *RDP* gera VNs considerando a disponibilidade de Bw nos enlaces e o consumo de energia geral do ISP. Contudo, estes trabalhos não focam simultaneamente na disponibilidade de Bw, eficiência energética e confiabilidade.

**Tabela 1. Trabalhos Relacionados**

Referência	Contexto	Critério
[Soualah et al. 2014]	Cloud	Falhas
[Cheng et al. 2012]	VN	Rank dos nós
[Mano et al. 2014]	VN	Múltiplos domínios
[Eppstein 1994]	Genérico	Algoritmo KSP
[Parandehgheibi et al. 2014]	Genérico	Sobrevivência
[Gomes et al. 2016b]	VN	Confiabilidade e Bw
[Gomes et al. 2016a]	VN	Energia e Bw
Algoritmo <i>FUZA</i> (este trabalho)	EaaS	Confiabilidade, Energia e Bw

### 3. Algoritmo *FUZA*

Esta seção apresenta o algoritmo *FUZA* para a geração de topologias virtuais confiáveis, o qual é apresentado no Algoritmo 1. O algoritmo *FUZA* é um algoritmo guloso exato, que tem como base a ideia de caminhos relativamente disjuntos, ou seja, é gerado um caminho inicial e posteriormente são adicionados enlaces ao mesmo a fim de alcançar a confiabilidade desejada, enquanto considera-se simultaneamente a eficiência energética e largura de banda disponível para determinar qual redundância de caminhos é a mais adequada. A notação apresentada na Tabela 2 é usada para descrever o algoritmo.

A variável  $p$  é a responsável pelo controle da redundância aplicada na topologia virtual. Por exemplo,  $p = 0$  representa o caso sem redundância, ou seja, a topologia da rede virtual terá apenas um caminho para cada nó destino. Por outro lado,  $p = 1$  é o caso de redundância completa, ou seja, a topologia terá dois caminhos completamente disjuntos para cada nó destino. Da mesma forma,  $p = 0.5$  é o caso onde metade dos enlaces do caminho primário serão usados como base para a definição do caminho secundário.

**Tabela 2. Notação Utilizada**

<b>Símbolo</b>	<b>Descrição</b>
$G$	grafo representando a infraestrutura de rede
$s$	nó fonte/raiz que representa o cliente
$D$	conjunto de destinos o qual deseja-se conectar
$L$	conjunto de enlaces da infraestrutura de rede
$N$	conjunto de nós da infraestrutura de rede
$l$	enlace entre dois nós
$w_l$	custo/peso do enlace $l$
$\wp$	quantidade perto de infinito
$\varepsilon$	quantidade próxima a zero
$e$	número de enlaces a serem atualizados
$p$	percentual de redundância desejado ( $0 \leq p \leq 1$ )
$T_1$	árvore com os componentes de rede de $s$ para os nós em $D$
$G'$	grafo alternativo com $w_l$ atualizados
$T_2$	árvore com os caminhos alternativos gerados a partir de $G'$
$G_f$	topologia final para a rede virtual (junção de $T_1$ e $T_2$ )
$Rr$	confiabilidade requisitada pelo cliente
$R$	confiabilidade da rede virtual $G_f$
$best$	melhor topologia virtual encontrada até o momento
$Bw_l$	largura de banda disponível no enlace $l$
$BwO_l$	largura de banda original do enlace $l$
$BwR$	largura de banda requisitada pelo cliente
$E_S$	energia gasta por um nó
$E_L(x)$	energia consumida por um enlace quando $x$ Mbps são alocados
$En_{Max}$	máximo de energia consumida pela infraestrutura de rede
$Score$	nível de adequabilidade da topologia segundo o sistema fuzzy

Inicialmente, o Algoritmo 1 define um laço para iterar entre os possíveis fatores de redundância ( $p$ ). Dentro do laço, é criada uma árvore inicial  $T_1$  com o nó  $s$  como raiz executando-se o algoritmo de definição de caminho (função  $PathDefinition(Grafo, No)$ ). Esta função encontra o melhor caminho para os nós determinados (em  $D$ ) de acordo com o peso dos enlaces.

Portanto,  $T_1$  contém os enlaces pertencentes aos caminhos de  $s$  para os nós contidos em  $D$ . A linha 4 atribui a  $e$  o número de enlaces a serem atualizados para gerar a redundância na topologia.  $e$  é calculado como um percentual do número de enlaces em  $T_1$  de acordo com  $p$  ( $0 \leq p \leq 1$ ).

Após executar a função  $PathDefinition$ , o Algoritmo 1 atualiza o custo de cada enlace na rede, criando um novo grafo  $G'$ . Para criar este novo grafo, o algoritmo substitui o peso  $w_l$  dos enlaces de acordo com a redundância, ou seja, ele substitui o peso  $w_l$  dos  $e$  primeiros enlaces de  $T_1$  de acordo com a função  $UpdateLink(Link, Link)$  apresentada no Algoritmo 2.

No próximo passo, o algoritmo encontra a árvore  $T_2$  no grafo  $G'$  (o qual possui os pesos dos enlaces atualizados) a partir da execução da função  $PathDefinition$ . Pos-

---

**Algoritmo 1** Fuzzy como base para Alocação (FUZA)

---

```
1:  $p = 0$ ; ▷ Caso sem redundância
2: enquanto  $p \leq 1$  faça ▷ Redundância  $\neq$  Completa
3:   Árvore  $T_1 = PathDefinition(G, s, D)$ ;
4:    $e = p * |T_1|$ ;
5:   para todo Enlace  $j \in T_1$  faça
6:     para todo Enlace  $i \in G$  faça
7:        $UpdateLink(j, i)$ ;
8:     fim para
9:   fim para
10:  Árvore  $T_2 = PathDefinition(G', s, D)$ ;
11:  Grafo  $G_f = MergePaths(T_1, T_2)$ ;
12:   $Score = Fuzzy(Bw_{Impact}(G_f), En_{Impact}(G_f))$ ;
13:  se ( $Score < best$ ) então
14:     $R = Reliability(G_f)$ ;
15:    se ( $R > Rr$ ) então
16:       $best = Score$ ;
17:    fim se
18:  fim se
19:   $Incrementar(p)$ ;
20: fim enquanto
```

---

teriormente, o algoritmo faz a junção das árvores  $T_1$  e  $T_2$  (linha 11) para criar um grafo com os componentes de rede (enlaces e nós) que existem nas duas árvores, resultando na topologia final  $G_f$ .

Após a definição de  $G_f$ , o algoritmo verifica se esta é a melhor solução encontrada até o momento, baseado no sistema fuzzy a ser descrito na Seção 3.1, sobre as perspectivas de largura de banda disponível (função  $Bw_{Impact}(Grafo)$ ) e consumo de energia (função  $En_{Impact}(Grafo)$ ), ambas descritas nos Algoritmos 3 e 4, respectivamente.  $best$  é uma variável para identificar a topologia virtual mais adequada. Se a “melhor” opção é encontrada, checa-se se  $G_f$  atende a confiabilidade requisitada pelo cliente no SLA ( $Rr$ ). A confiabilidade  $R$  de  $G_f$  é calculada de acordo com o método mostrado na Seção 2.2.

Finalizando o laço, a redundância desejada é incrementada para permitir que menos enlaces que foram utilizados no caminho primário sejam considerados na busca por um caminho alternativo. O valor do incremento é configurado pelo administrador da rede, por exemplo: 0.1, 0.25, ou 0.5 por iteração. Nos experimentos realizados, considerou-se um incremento de 0.25, visto que ele representa uma maior abrangência das possíveis variações de topologia. Portanto, este valor de 0.25 resulta em um número plausível de possibilidades, enquanto que evita uma busca excessiva por topologias com uma pequena variação (muito similares).

O Algoritmo 2 é utilizado para realizar a atualização do custo dos enlaces, o qual tem por objetivo evitar que os enlaces já alocados sejam evitados na busca por um caminho alternativo. Enquanto o número de enlaces ( $e$ ) não é alcançado, o custo do enlace é substituído por  $\varnothing$  para evitar o seu uso, e após  $e$  enlaces serem processados, o custo dos enlaces é substituído por 0, a fim de encorajar a alocação do mesmo.

---

**Algoritmo 2** *UpdateLink(Enlace  $j$ , Enlace  $i$ )*

---

```
1: se ( $i == j$ ) então
2:   se ( $e > 0$ ) então
3:      $w'_j = \wp$ ;
4:      $e = e - 1$ ;
5:   senão
6:      $w'_j = 0$ ;
7:   fim se
8: fim se
```

---

Com relação ao impacto da alocação de  $G_f$  na largura de banda, este cálculo é feito de acordo com o Algoritmo 3, onde  $Bw_l$  é a largura de banda disponível no enlace  $l$  e  $BwR$  é a largura de banda requisitada pelo cliente no SLA.

---

**Algoritmo 3** *BwImpact(Grafo  $G_f$ )*

---

```
1:  $Sum = 0$ ;
2: para todo Enlace  $l \in G_f$  faça
3:    $Sum + = \frac{BwR}{Bw_l}$ ;
4: fim para
5: retorne  $Sum$ ;
```

---

O impacto da alocação de  $G_f$  no consumo de energia é medido de acordo com o Algoritmo 4. Primeiramente, a função calcula o consumo de energia dos comutadores (do inglês *switch*) da rede, onde  $E_S(Node)$  é a energia gasta pelo nó de acordo com a Equação (1) [Mahadevan et al. 2009], e  $En_{Max}$  representa o máximo de energia consumida pela rede (ou seja, a energia consumida quando todos os componentes estão operacionais em máxima capacidade).

$$E_S(o) = P_{ch} + (N_o * Pl) + \sum_{c=0}^C (NumPorts_c * E_c) \quad (1)$$

$P_{ch}$  é a energia consumida pelo chassi do *switch*;  $Pl$  é a energia consumida pelas portas de transmissão não ativadas, e  $N_o$  é o número de placas de transmissão conectadas ao *switch*  $o$ . A variável  $C$  representa as possíveis configurações de velocidade das portas de transmissão, e  $E_c$  é a energia consumida por cada porta executando a uma velocidade  $c$ ;  $NumPorts_c$  é o número de portas configuradas para a velocidade  $c$ . De acordo com a referência [Mahadevan et al. 2009],  $P_{ch}$  é normalmente 50 Watts e  $Pl$  é 40 watts.

Adicionalmente, o Algoritmo 4 calcula o aumento na energia consumida nos enlaces devido ao aumento na alocação de largura de banda requisitada ( $BwR$ ) no mesmo.  $BwO_l$  é a largura de banda original do enlace  $l$ , enquanto que  $E_c(x)$  é a largura de banda consumida quando  $x$  Mbps são alocados no enlace (segundo a Equação 2, a qual é baseada na referência [Mahadevan et al. 2009]).

$$E_c(x) = \begin{cases} 0.4, & \text{se } 0 < x \leq 10Mbps; \\ 0.5, & \text{se } 10 < x \leq 100Mbps; \\ 1 & \text{se } 100 < x \leq 1Gbps; \\ 0, & \text{caso contrario;} \end{cases} \quad (2)$$



---

**Algoritmo 4**  $En_{Impact}(Graph G_f)$ 

---

```
1:  $Sum = 0$ ;  
2: para todo Nodes  $o \in G_f$  faça  
3:    $Sum + = \frac{E_S(o)}{En_{Max}}$  ;  
4: fim para  
5: para todo Link  $l \in G_f$  faça  
6:    $Sum + = \frac{E_C(BwO_l+BwR-Bw_l) - E_C(BwO_l-Bw_l)}{En_{Max}}$  ;  
7: fim para  
8: retorne  $Sum$ ;
```

---

Numa visão geral, o Algoritmo 1 constrói uma árvore inicial e, a partir de uma atualização dos pesos dos enlaces, busca-se por um caminho alternativo através da adição de novos enlaces a árvore inicial. Esta adição é limitada pelo fator de redundância definido. A atualização do custo dos enlaces é usada para evitar o uso dos enlaces que já foram alocados, mas sem descartá-los como uma opção, forçando o algoritmo a procurar caminhos alternativos para alcançar os nós em  $D$ . Executando o Algoritmo 1 gera-se uma topologia para VN com caminhos alternativos considerando aspectos de consumo de energia, largura de banda disponível e a confiabilidade requisitada pelo cliente.

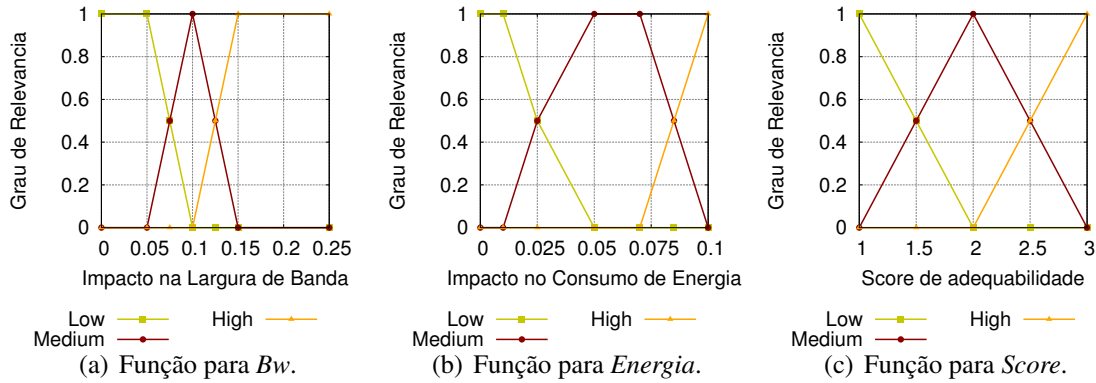
### 3.1. Sistema fuzzy para medição de adequabilidade

No conceito tradicional de conjuntos, um elemento pertence totalmente ou não a um certo conjunto. Por outro lado, os conjuntos *Fuzzy* definem funções de adesão limitadas ao intervalo  $[0, 1]$ , para expressar o grau de adesão de um elemento em relação ao conjunto em questão, representado por  $\mu(x)$  [Abedin et al. 2011]. Além disso, os conjuntos *Fuzzy* são uma ferramenta para lidar com problemas de tomada de decisão, devido a duas razões: (i) lidam com o conceito de “grau de satisfação”; e (ii) têm uma estrutura matemática para manipular informações vagas.

Inicialmente, os valores de consumo de energia ( $En_{Impact}$ ) e largura de banda ( $Bw_{Impact}$ ) são convertidos em variáveis linguísticas, a partir das funções de adesão respectivas, ilustradas nas Figuras 2(a) e 2(b). Posteriormente, as variáveis linguísticas geradas são aplicadas no conjunto de regras (apresentado na Tabela 3) e produzem um conjunto de variáveis linguísticas relacionadas a função de adesão *Score*, a qual é usada como saída e é mostrada na Figura 2(c). A configuração definida do conjunto de regras visa favorecer o uso de caminhos com pequeno impacto na Bw disponível (valores *Low* e *Medium*) e que consigam, ao menos, manter o consumo de energia atual do ISP (valor *Low*).

Por fim, as variáveis linguísticas vindas do sistema de inferência são convertidas em um valor real de acordo com o método Weight Average Maximum (Equação 3), pois é um método eficaz e de baixo processamento, encaixando-se no escopo do trabalho que visa realizar as alocações de VN em tempo real. Este método produz um valor numérico considerando o peso médio dos maiores valores (pico das variáveis linguísticas mostradas na Figura 2(c)).

$$Score = \frac{(1 * \mu_H(x)) + (2 * \mu_M(x)) + (3 * \mu_L(x))}{(\mu_H(x) + \mu_M(x) + \mu_L(x))} \quad (3)$$



**Figura 2. Funções de Adesão**

**Tabela 3. Conjunto de Regras**

Energia	Operação	Largura de Banda	Score
High	E	High	Low
High	E	Medium	Low
High	E	Low	Low
Medium	E	High	Low
Medium	E	Medium	Medium
Medium	E	Low	Medium
Low	E	High	Medium
Low	E	Medium	High
Low	E	Low	High

Portanto, o resultado oriundo da Equação 3 representa o *Score* associado a topologia de rede gerada pelo Algoritmo 1. Na Equação 3,  $\mu_H(x)$  é o grau de adesão da variável *High*,  $\mu_M(x)$  é o grau de adesão da variável *Medium*, e  $\mu_L(x)$  é o grau de adesão da variável *Low*. Os valores 1, 2, e 3 são os máximos das variáveis *High*, *Medium*, e *Low*, respectivamente, como mostrado na Figura 2(c).

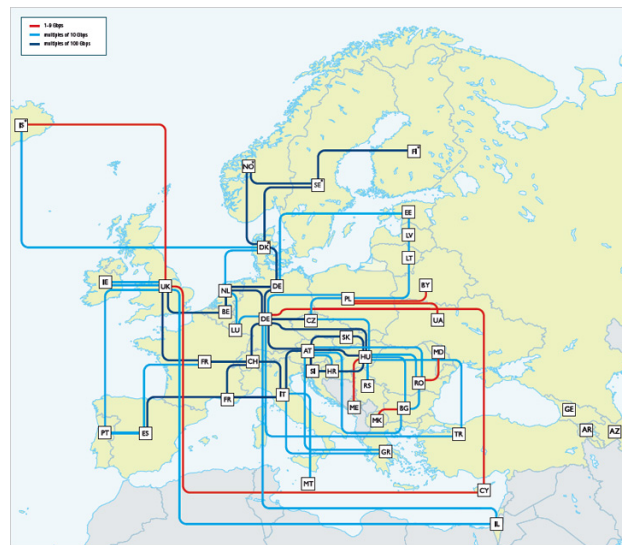
## 4. Resultados

Esta seção apresenta os experimentos realizados para avaliar o algoritmo *FUZA* com relação ao processo de definição de topologias virtuais. A fim de realizar os experimentos foi desenvolvido um simulador<sup>1</sup> de alocação de redes virtuais, o qual possibilita a análise dos principais aspectos relacionados a definição de topologias virtuais: alocações realizadas com sucesso, eficiência energética, disponibilidade de largura de banda e a conectividade das redes virtuais quando ocorrem falhas na infraestrutura de rede (onde a ordem das falhas foi gerada de forma aleatória). A Seção 4.1 descreve o cenário usado na avaliação, enquanto a Seção 4.2 discute os resultados e analisa o desempenho dos algoritmos avaliados.

### 4.1. Configuração do Cenário

Os experimentos utilizaram a topologia da rede GEANT, a qual possui quarenta nós e sessenta enlaces, como ilustrado na Figura 3. Cada enlace da rede GEANT foi configurado com 1Gpbs de largura de banda disponível.

<sup>1</sup><http://bitbucket.org/rafaellgom/vn-allocation/>



**Figura 3. Topologia da rede GEANT.**

Os experimentos visam avaliar a capacidade da proposta solucionar um conjunto de requisições de VN com os seguintes parâmetros: (i) conjunto de nós a serem conectados (escolhidos de forma uniforme); (ii) duração das requisições (valor médio de 50 unidades de tempo); e, (iii) Bw requisitada (valor médio de 100 Mbps). Foram gerados aleatoriamente quinhentos conjuntos, onde cada conjunto era composto de cem requisições de redes virtuais.

Normalmente, os modelos de tráfego de rede assumem que o intervalo de tempo de chegada e a duração dos fluxos, os quais resultam na demanda de tráfego, seguem uma distribuição exponencial [Chen 2007]. Portanto, neste trabalho a largura de banda requisitada e a duração das requisições de VN são geradas a partir de um processo Poisson, visto que ambos podem ser considerados relacionados a demanda de tráfego.

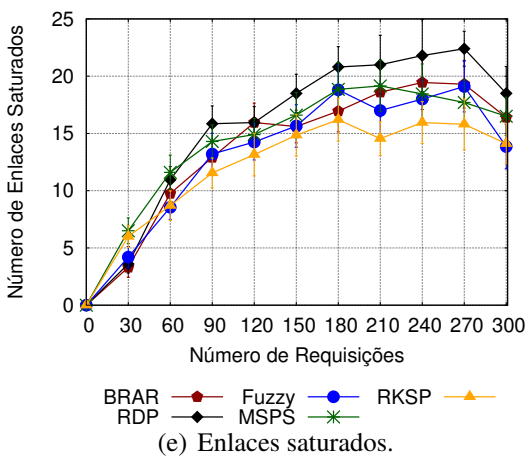
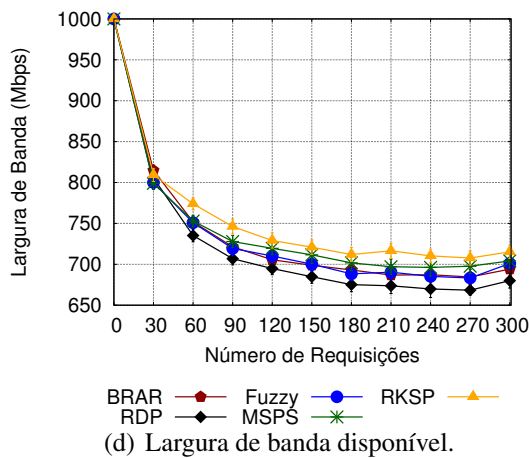
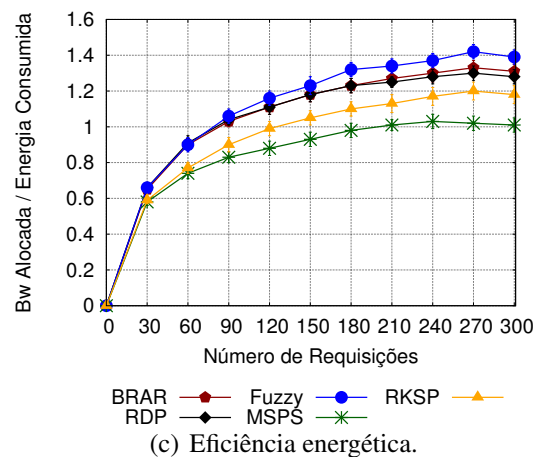
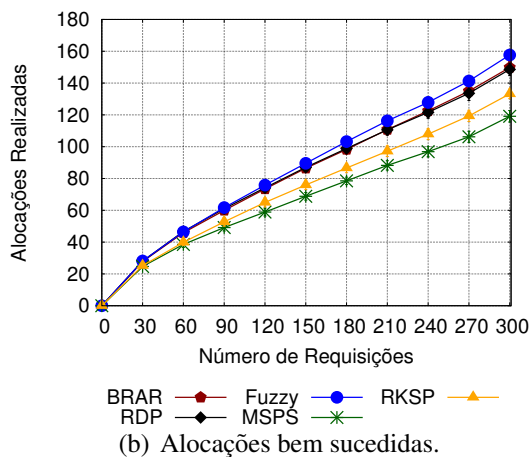
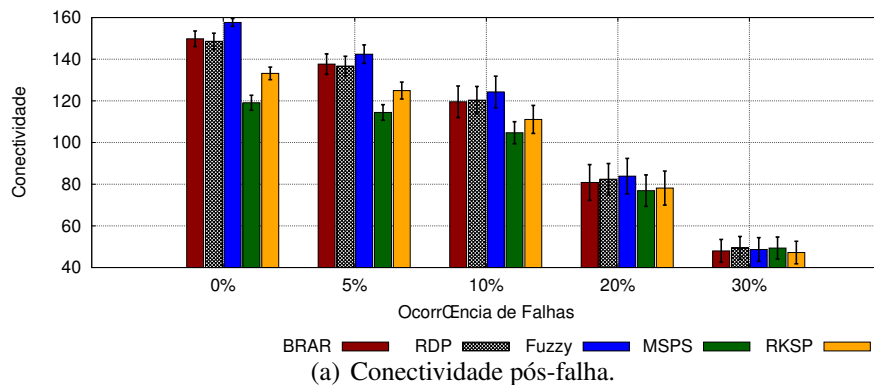
## 4.2. Resultados

Nesta seção apresenta-se o resultado dos experimentos realizados, comparando o desempenho do algoritmo *FUZA* com os algoritmos existentes descritos na Seção 2.3: RKSP [Eppstein 1994], o qual encontra os  $K$  caminhos mais curtos até atingir a confiabilidade desejada; MSPS [Parandehgheibi et al. 2014], um algoritmo que encontra o menor número de caminhos entre dois pares de nós que sobrevivem a qualquer falha de enlace na infraestrutura de rede; BRAR [Gomes et al. 2016b], que busca pela melhor relação confiabilidade e largura de banda; e RDP [Gomes et al. 2016a], um algoritmo de alocação para gerar VN considerando a confiabilidade mínima necessária, assim como a disponibilidade de largura de banda e consumo de energia.

Durante os experimentos foram avaliadas quatro métricas:

- Conectividade pós-falha (Figura 4(a)): o número de requisições bem sucedidas que continuam totalmente conectadas (alcançam todos os nós) após a falha dos componentes de rede;
- Alocações bem sucedidas (Figura 4(b)): o número de requisições solucionadas, ou seja, quantas requisições de VN o algoritmo foi capaz de alocar com a confiabilidade e a largura de banda desejadas;

- Eficiência energética (Figura 4(c)): montante de largura de banda alocada em relação ao consumo de energia consumida;
- Largura de banda disponível (Figura 4(d)): a largura de banda média disponível na infraestrutura de rede após as alocações serem realizadas;
- Enlaces saturados (Figura 4(e)): o número de enlaces que ficaram com menos de 10% de largura de banda disponível.



**Figura 4. Desempenho dos algoritmos avaliados.**

A Figura 4(a) mostra o número de alocações de VN que continuam operacionais (mantêm conectividade entre todos os nós envolvidos) após ocorrerem falhas. O eixo  $x$  da

Figura 4(a) representa o percentual de componentes de rede que aleatoriamente falharam, onde a falha de um enlace representa a remoção do mesmo da rede, enquanto a falha de um nó significa a remoção do nó e seus enlaces da rede. O nível de conectividade é diretamente relacionado ao número de componentes alocados pelo algoritmo para a VN, visto que quanto maior o número de componentes na VN, maior é a probabilidade de ocorrer uma quebra na conectividade quando falhas ocorrem.

O ponto “0%” da Figura 4(a), assim como na Figura 4(b), ilustra o número de requisições alocadas com sucesso (isto é, cumprem o SLA) quando a rede está totalmente operacional. Em geral, o algoritmo *FUZA* solucionou em média, pelo menos, 5% mais requisições que os demais algoritmos. Este melhor desempenho ocorre devido a característica do algoritmo *FUZA* em considerar tanto os aspectos de energia quanto a largura de banda, resultando em um melhor balanceamento de alocações por toda a infraestrutura de rede.

A Figura 4(c) mostra a eficiência energética, a qual avalia se o consumo de energia do ISP está sendo eficaz (quanto maior melhor). Assim como apresentado na Seção 3, a eficiência energética é a quantidade de Bw alocada para o conjunto de VNs pelo montante de energia gasto pela infraestrutura de rede. De acordo com a Figura 4(c), o algoritmo *FUZA* melhora a eficiência energética da rede em torno de 7%. Este melhor desempenho do algoritmo *FUZA*, em relação aos algoritmos existentes, deve-se a abordagem de avaliação considerando ambos os aspectos, de energia e Bw.

De acordo com a Figura 4(d), os algoritmos MSPS, RKSP, e BRAR obtêm um comportamento similar, onde o comportamento dos algoritmos MSPS e RKSP justifica-se pelo baixo número de alocações. Por outro lado, o algoritmo *FUZA* apresenta uma disponibilidade similar aos demais, mesmo realizando mais alocações de VNs. Adicionalmente, quando analisa-se os dados ilustrados na Figura 4(e), percebe-se que os algoritmos possuem um número próximo de enlaces saturados, apesar do algoritmo *FUZA* resultar em um maior número de alocações.

Baseado nos experimentos realizados, os resultados indicam que o algoritmo *FUZA* pode evoluir o processo de alocação de VNs para os ISPs. O algoritmo *FUZA*, apesar de ter um consumo de disponibilidade de Bw similar aos demais algoritmos, consegue alocar mais requisições (em torno de 5% maior) e eleva a eficiência energética do ISP (em torno de 7%). Da mesma forma, o algoritmo *FUZA* aloca VNs que mantêm a conectividade quando falhas na infraestrutura de rede ocorrem.

## 5. Conclusão

A aplicação da abordagem EaaS traz flexibilidade e gerenciabilidade para os ISPs, permitindo melhorar o uso da infraestrutura de rede e a prestação de serviço aos seus clientes. Contudo, a alocação de VNs ainda é um desafio em aberto. Portanto, este artigo propõe um algoritmo de definição de topologias virtuais multicritério, chamado de *FUZA*.

O foco do algoritmo *FUZA* é definir topologias virtuais para VNs, as quais são planejadas para serem confiáveis em frente a falhas (cumprindo os parâmetros do SLA), enquanto reduz a utilização de Bw e consumo de energia quando a rede está totalmente operacional. O algoritmo *FUZA* supera os algoritmos existentes analisados, atendendo um maior número de clientes (em torno de 5%) e obtendo uma maior eficiência energética (cerca de 7%). Estes fatos habilitam a maximização dos lucros do ISP.

Como trabalhos futuros, pretende-se estender o algoritmo proposto para aplicar uma abordagem multicritério ótima, como por exemplo fronteira de Pareto, bem como adicionar critérios adicionais ao sistema fuzzy.

## Agradecimentos

Os autores gostariam de agradecer a FAPESP, CAPES e CNPq pelo apoio financeiro.

## References

- Abedin, F., Chao, K.-M., and Godwin, N. (2011). A fuzzy group decision making process in a multi-agent negotiation environment. In *15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 311–318.
- Chen, T. M. (2007). Network Traffic Modeling. pages 326–339.
- Cheng, X., Su, S., Zhang, Z., Shuang, K., Yang, F., Luo, Y., and Wang, J. (2012). Virtual network embedding through topology awareness and optimization. *Computer Networks*, 56(6):1797 – 1813.
- Davy, S., Famaey, J., Serrat-Fernandez, J., Gorricho, J., Miron, A., Dramitinos, M., Neves, P., Latre, S., and Goshen, E. (2014). Challenges to support edge-as-a-service. *Communications Magazine, IEEE*, 52(1):132–139.
- Eppstein, D. (1994). Finding the k shortest paths. In *35th Annual Symposium on Foundations of Computer Science*, pages 154–165. IEEE.
- Gomes, R. L., Bittencourt, L. F., Madeira, E., Cerqueira, E., and Gerla, M. (2016a). State-Aware allocation of reliable virtual software defined networks based on bandwidth and energy. In *13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 418–423, Las Vegas, USA.
- Gomes, R. L., Bittencourt, L. F., Madeira, E. R., Cerqueira, E., and Gerla, M. (2016b). A combined energy-bandwidth approach to allocate resilient virtual software defined networks. *Journal of Network and Computer Applications*, 69:98 – 106.
- Lee, H.-W., Modiano, E., and Lee, K. (2010). Diverse Routing in Networks With Probabilistic Failures. *IEEE/ACM Transactions on Networking*, 18(6).
- Li, V. and Silvester, J. (1984). Performance Analysis of Networks with Unreliable Components. *IEEE Transactions on Communications*, 32(10):1105–1110.
- Mahadevan, P., Sharma, P., Banerjee, S., and Ranganathan, P. (2009). Energy Aware Network Operations. In *IEEE INFOCOM Workshops 2009*, pages 1–6.
- Mano, T., Inoue, T., Ikarashi, D., Hamada, K., Mizutani, K., and Akashi, O. (2014). Efficient virtual network optimization across multiple domains without revealing private information. In *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8.
- Parandehgheibi, M., Lee, H.-W., and Modiano, E. (2014). Survivable path sets: A new approach to survivability in multilayer networks. *Journal of Lightwave Technology*, 32(24):4741–4752.
- Soualah, O., Fajjari, I., Aitsaadi, N., and Mellouk, A. (2014). A reliable virtual network embedding algorithm based on game theory within cloud’s backbone. In *2014 IEEE International Conference on Communications (ICC)*, pages 2975–2981.