

# Agregação Dinâmica de Enlaces em Ambientes de Redes Definidas por Software

Ronaldo Resende Rocha Junior,  
Marcos A. M. Vieira, Antonio A. F. Loureiro

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais – Belo Horizonte, MG – Brasil

{ronaldorrjr,mmvieira,loureiro}@dcc.ufmg.br

**Abstract.** *The link aggregation technique is a solution to the link saturation problem. This technique combines several physical links to create a virtual link with the sum of the respective bandwidths. Since the use of Software Defined Networks (SDN) in corporate environments increases every day, the purpose of this work is to allow the SDN controller to configure the aggregation of links. Therefore, We have defined and implemented an architecture to dynamically aggregate links such that it is configurable, scalable, and autonomous. We evaluate three link aggregation algorithms: hash, traffic analysis, and round-robin. In our tests, the proposed system improves the distribution of the bandwidth, allows more aggregated interfaces, and increases the throughput between two endpoints.*

**Resumo.** *A técnica de agregação de enlaces é uma solução para o problema de saturação do enlace. Essa técnica combina várias interfaces físicas para criar um enlace virtual com a soma das respectivas bandas. Visto que a utilização das Redes Definidas por Software (SDN) em ambientes corporativos aumenta a cada dia, a proposta deste trabalho é permitir o controlador SDN configurar a agregação de enlaces. Para tanto, definimos e implementamos uma arquitetura que permite agregar enlaces de forma dinâmica, configurável, escalável e autônoma. Foram avaliados três algoritmos de agregação de enlaces: hash, análise de tráfego e round-robin. Em nossos testes, o sistema proposto melhora a distribuição da banda, permite maior quantidade de interfaces agregadas e aumenta a vazão dos tráfegos entre duas pontas.*

## 1. Introdução

As redes de computadores modernas são sistemas complexos e, por isso, a sua administração é um grande desafio tanto para profissionais quanto para pesquisadores [Ranum et al. 1998]. Os equipamentos de redes atuais, tais como roteadores, *firewalls* e *switches*, são considerados verdadeiras caixas pretas [Moore and Nettles 2001]. Cada empresa é responsável por criar seus sistemas operacionais e, mesmo que os algoritmos de roteamento sejam os mesmos, a forma como eles são implementados varia bastante, podendo, inclusive, levar a problemas de interoperabilidade. Esse foi o caso do padrão IEEE 802.11, que devido a problemas práticos de integração dos dispositivos de diferentes fabricantes, levou à criação da aliança Wi-Fi<sup>1</sup>.

---

<sup>1</sup><http://www.wi-fi.org/>

O gargalo de tráfego é um dos principais problemas existentes em ambientes de redes clássicos [Carter and Crovella 1996]. Isso acontece, por exemplo, quando vários usuários tentam utilizar um único servidor em uma rede, saturando o enlace que conecta o *switch* àquele servidor. A Figura 1 ilustra esse problema, onde quatro servidores (conectados em um único *switch* utilizando um enlace de 1 Gbps cada) tentam acessar recursos de um servidor remoto (que está conectado ao mesmo *switch*, utilizando apenas uma conexão de 1 Gbps) ao mesmo tempo.

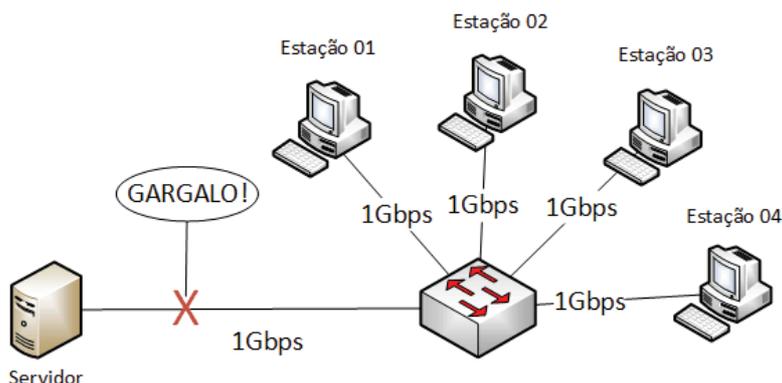


Figura 1. Exemplo de gargalo

Considerando que o fluxo de dados de cada usuário tenha o mesmo tamanho, podemos estimar que durante a concorrência do tráfego, caso todos os usuários tentem acessar o servidor em questão ao mesmo tempo, a banda média disponível para cada um será de apenas 250 Mbps. Vale salientar que essa é uma estimativa média realizada apenas para ilustrar o problema do gargalo, uma vez que em redes com fluxos reais TCP/UDP, essa taxa média dependerá do tamanho do fluxo sendo transmitido pelo usuário ao servidor [Judd 2015]. Isso demonstra que muitas vezes avaliações incorretas são feitas a respeito da lentidão em ambientes de rede, levando à troca ou *upgrade* desnecessários de equipamentos.

Uma rede com alta disponibilidade, tolerante a falhas e com alto potencial de transmissão dos seus enlaces é ideal em ambientes corporativos [Floyd and Jacobson 1995]. Através da implantação da agregação de enlaces, podemos conseguir essas características, já que essa técnica permite transformar diversos enlaces físicos em apenas um enlace virtual. Nos *switches* clássicos, é utilizado o protocolo aberto LACP (*Link Aggregation Control Protocol*) ou os diversos protocolos proprietários dos fabricantes (Cisco: *Port Aggregation Protocol*, Huawei: *Eth-Trunk*, Juniper: *Aggregated Ethernet*, entre outros).

Este trabalho tem como objetivo propor a implementação da técnica de agregação de enlaces em controladores SDN e, assim, não será utilizado o protocolo LACP, mas a estrutura de dados do protocolo OpenFlow. Através dessa premissa, os seguintes resultados foram obtidos:

- **Melhor distribuição da banda:** utilizando diferentes estratégias para a tomada de decisão ao criar os fluxos de transmissão de pacotes, as interfaces da agregação podem ter um balanceamento de tráfego real.

- **Agregação autônoma:** a agregação pode ser criada, modificada ou removida automaticamente pelo controlador sem a necessidade de intervenção do administrador de redes.
- **Protocolo aberto:** utilizando a estrutura de dados e tabelas presentes no Open-Flow, o fluxo de transmissão de pacotes entre os enlaces de uma agregação é gerenciado pelo controlador SDN. Isso faz com que o uso de protocolos de agregação proprietários não seja mais necessário.
- **Aumento da quantidade de interfaces agregadas:** uma das limitações do protocolo LACP é a utilização máxima de oito interfaces para a agregação. Ao utilizar o LACP na plataforma *Open vSwitch*, esse valor cai para quatro portas (uma limitação referenciada na documentação da plataforma). Ao migrar a gerência da agregação desses enlaces para o controlador SDN, essa limitação deixa de existir.
- **Velocidade de transmissão entre links:** outra limitação do protocolo LACP é a necessidade de utilizar interfaces com a mesma configuração de velocidade na agregação, ou seja, interfaces com velocidade 100 Mbps não podem ser agregadas com interfaces com velocidade de 1 Gbps. Essa limitação existe por motivos de implementações do protocolo, na forma como ele realiza e mantém a agregação configurada. Tal limitação não existiria em ambientes SDN.

O restante deste artigo está organizado como segue. Primeiramente, a Seção 2 discute os trabalhos relacionados. A Seção 3 revela a arquitetura proposta e os algoritmos de agregação de enlaces. A Seção 4 descreve os experimentos e os resultados obtidos. Por fim, a Seção 5 apresenta a conclusão e os trabalhos futuros.

## 2. Trabalhos Relacionados

### 2.1. LACP (*Link Aggregation Control Protocol*)

O protocolo aberto LACP (*Link Aggregation Control Protocol*) [Seaman 1999] permite a implantação da agregação de enlaces nos *switches* clássicos. Esse protocolo foi publicado em 1997 e sua última versão é de 2014, com o padrão 802.1AX-2014. O LACP é amplamente utilizado em ambientes clássicos de redes, mas a sua aplicação em redes SDN ainda não é muito pesquisada.

Pra criar o fluxo, o LACP analisa vários parâmetros de origem e de destino como MAC, IP, porta e etiqueta MPLS. A partir do hash desses parâmetros, o protocolo escolhe a porta para o fluxo, ou seja, todos os pacotes deste fluxo passam somente nessa interface. Quando surge um novo fluxo, um novo hash é calculado e, provavelmente, outra interface será utilizada para ele. Assim, o LACP usa todas as interfaces da agregação, espalhando os fluxos entre elas.

Para criar uma agregação de enlaces utilizando o protocolo LACP, o administrador de redes deve realizar a configuração de forma manual. Ao ser habilitado, o protocolo cria uma interface virtual e a quantidade de enlaces físicos não pode ser maior do que oito e todas as interfaces do *switch* devem estar configuradas com a mesma velocidade.

O protocolo LACP pode ser empregado em redes SDN através da plataforma *Open vSwitch* [Pfaff et al. 2009]. Essa plataforma representa a implementação de um *switch* virtual capaz de suportar vários protocolos de camada 2 e interfaces de gerenciamento como *sFlow*, espelhamento de porta e *NetFlow*. Por ser basicamente uma máquina virtual,

essa plataforma geralmente é instalada em estações ou servidores, impossibilitando sua utilização em equipamentos SDN físicos.

## 2.2. Outras abordagens

A plataforma *Open vSwitch* foi criada para ser um *switch* virtual capaz de executar em qualquer ambiente [Pfaff et al. 2009]. Pelo fato de ser bastante completa e robusta, logo foi incorporada nas redes SDN. Entretanto, ao considerarmos a técnica de agregação de enlaces, essa plataforma não é eficaz. A quantidade máxima de agregação de portas físicas é bastante limitada, não podendo ultrapassar quatro unidades (uma limitação da própria plataforma). Além disso, essa configuração precisa ser manual. Portanto, a solução não é escalável ou autônoma, sendo ineficiente em ambientes como data centers, que possuem centenas de equipamentos.

Em [Steinbacher and Bredel 2015], os autores propõem a implementação do protocolo LACP no Floodlight, um controlador aberto escrito em linguagem Java. Esse trabalho apenas discute a implementação teórica do protocolo, sendo que a configuração do código do protocolo em si não é elaborada.

Já os autores de [Bredel et al. 2014], [Ahn et al. 2009], [Al-Fares et al. 2010] e [Alizadeh et al. 2010] consideram o caminho de um tráfego TCP entre múltiplos enlaces, sem considerar uma possível agregação local deles. Porém, em todos eles são considerados caminhos através de múltiplos saltos. A diferença deste trabalho para os demais é a abordagem na camada dois, ou seja, a comunicação entre dois *switches* com enlaces agregados entre eles.

Por fim, em [Vencioneck et al. 2014], os autores apresentam uma nova plataforma para controlar o encaminhamento de pacotes substituindo o *Open vSwitch*. Apesar de ser inovador e possuir resultados promissores, esse estudo é complementar ao nosso, pois não considera a agregação de enlaces entre dois dispositivos.

## 3. Arquitetura

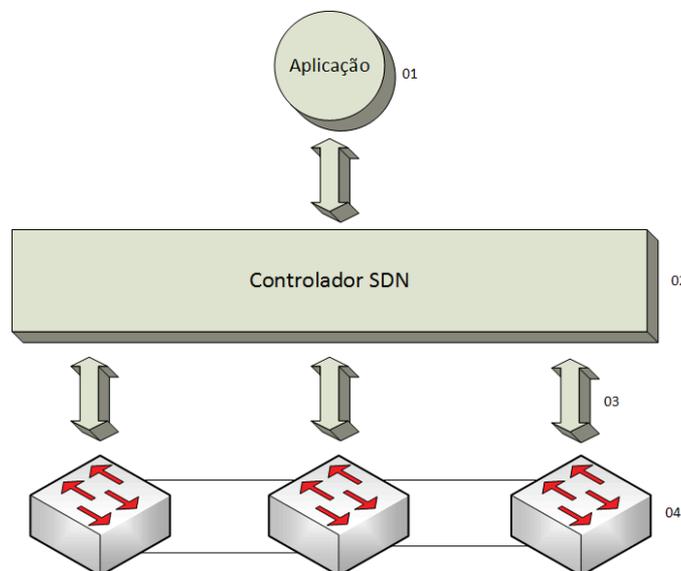
Esta seção apresenta a arquitetura proposta neste trabalho, implementação, identificação dos enlaces de agregação e os algoritmos de agregação de enlace.

### 3.1. Descrição

A Figura 2 retrata a arquitetura proposta:

- **Aplicação (01):** Nesta camada, foram criadas as regras de agregação de enlace. Exemplo: as políticas de encaminhamento de pacotes de acordo com a utilização de cada interface. No total foram criadas três aplicações, que são representadas por três arquivos na linguagem *Python* que são definidas nos parâmetros de entrada quando o controlador é iniciado.
- **Controlador (02):** responsável por gerenciar a rede local. Executa as decisões da aplicação sobre as políticas de roteamento, encaminhamento, redirecionamento e balanceamento de carga. O controlador envia regras para a tabela de fluxos dos *switches*. É utilizado o controlador Ryu.
- **Interface de ligação sul (Southbound) (03):** Protocolo de comunicação entre o controlador e os equipamentos de rede. Utiliza o padrão *OpenFlow*, já que existem funções pré-definidas no controlador Ryu que facilitaram o desenvolvimento das aplicações.

- **Switch (04)**: responsável por comutar os pacotes. Em cada equipamento de rede, o cliente *OpenFlow* é configurado utilizando a tabela de encaminhamento de pacotes. Em nosso ambiente, são criadas instâncias virtual do *Open vSwitch* pela plataforma Mininet.



**Figura 2. Arquitetura proposta neste trabalho**

### 3.2. Implementação

O sistema foi desenvolvido utilizando o controlador Ryu, que gerencia todos os detalhes entre as conexões entre *switches*, interpretando e convertendo os pacotes de rede em objetos fáceis de usar. O aplicativo do *switch* escuta as mensagens de pacotes de entrada, mantém uma tabela interna de endereços MAC e adiciona as regras de encaminhamento nos *switches* à medida em que as conexões são identificadas utilizando o protocolo *OpenFlow*.

Essencialmente, três passos devem ser seguidos para criar uma aplicação Ryu: o registro e inicialização da aplicação ocorre antes dos *switches* serem inseridos no domínio do controlador Ryu e permite que a instância da aplicação inicialize os dados que serão compartilhados em toda a rede. Por exemplo, a aplicação *switch* inicializa uma tabela para manter informações dos endereços MAC das estações ligadas nos *switches* e as respectivas portas às quais elas estão conectadas. Em seguida acontece a inicialização de um *switch* que se conecta à controladora Ryu, ou seja, um *switch* é inserido no domínio do controlador Ryu, e assim o aplicativo pode verificar todas as suas características através do evento *EventOFPSwitchFeatures*. Geralmente, nesse processo, todas as regras de encaminhamento estáticas são adicionadas ao *switch*. Na aplicação *switch*, uma regra é adicionada para que qualquer pacote sem regra de encaminhamento seja enviado para o controlador. Por fim realiza-se o envio de pacotes e neste passo o aplicativo pode escolher monitorar o evento *EventOFPPacketIn*, que é responsável pelo envio de pacotes. Isso ocorre sempre quando o *switch* não possui uma regra específica para enviar aquele pacote ou caso tenha uma regra geral de encaminhamento que o instrui a enviar o pacote para o controlador analisá-lo.

Cada uma das nossas aplicações é uma modificação do aplicativo *simple\_switch.py* presente na plataforma Ryu, pois nela implementamos toda a inteligência e tomada de decisão do controlador, deixando o *switch* responsável apenas por encaminhar os pacotes de acordo com as regras presentes em sua memória em tempo de execução. Essas aplicações possuem duas partes: a primeira é o processo automático de identificação dos enlaces de agregação, e a segunda o algoritmo de agregação de enlace que implementam as políticas de encaminhamento de pacotes.

### 3.3. Identificação dos enlaces de agregação

Quando o controlador SDN recebe a notificação da inclusão de um *switch* em seu domínio, ele verifica os enlaces desse *switch*. Caso sejam identificadas mais de uma ligação entre um mesmo par de *switches*, o controlador assume que existe uma agregação de enlaces entre esse par e cria uma tabela para ela. Com esse mecanismo, não há a necessidade da configuração manual por parte do administrador de redes. Esse processo torna a agregação de enlaces autônoma, escalável e dinâmica.

### 3.4. Algoritmos de agregação de enlace

Em redes SDN, temos a liberdade de implementar nosso próprio algoritmo de agregação de enlaces. Podemos classificar os algoritmos de agregação como do tipo inter-fluxo ou intra-fluxo. O tipo inter-fluxo ocorre quando cada fluxo é separado em caminhos distintos, mas não há divisão do fluxo. O tipo intra-fluxo ocorre quando existe a divisão de um fluxo em vários sub-fluxos com caminhos distintos. Diante disto, durante este trabalho, identificamos e realizamos a agregação utilizando três técnicas distintas que serão descritas a seguir.

Vale ressaltar que a plataforma de emulação Mininet não suporta a configuração do protocolo LACP entre dois *switches* virtuais. Por esse motivo, optamos por realizar a implementação de um algoritmo parecido com o LACP, batizado de Tabela *Hash*, para superar essa limitação.

#### 3.4.1. Tabela *Hash*

As estações ligadas nos *switches* começam a transmitir dados. Quando o *switch* recebe o primeiro pacote do fluxo, ele o retransmite para o controlador, que identifica os campos disponíveis (endereço MAC de origem e destino, endereço IP de origem e destino) e calcula um *hash*, que determina qual interface será utilizada para transmitir os dados daquele fluxo. O controlador insere a regra de encaminhamento no *switch*, que encaminha os pacotes daquele fluxo pela interface calculada pelo controlador. O Algoritmo 1 representa a implementação desse procedimento.

---

#### Algorithm 1 Algoritmo implementando a política **Hash**

---

```

1: function CALCULA HASH(pacote)
2:   valor hash(mac origem, mac destino, ip origem, ip destino)
3:   return valor
4: end function

```

---

### 3.4.2. Análise de Tráfego

O *switch*, ao receber o primeiro pacote do fluxo, o retransmite para o controlador, que identifica a utilização da banda de cada interface da agregação para tomar a decisão de qual interface será utilizada para transmitir os dados daquele fluxo em questão. A prioridade é escolher interfaces com baixa utilização de banda, permitindo assim uma melhor distribuição do tráfego entre todas as interfaces. Apesar, dessa escolha ser realizada apenas no começo da transmissão do tráfego, eventualmente ela será refeita já que as regras criadas possuem um tempo de expiração de 30 segundos após ocorrer inatividade. O controlador insere a regra de encaminhamento no *switch*. Em seguida, o *switch* encaminha os pacotes daquele fluxo pela interface escolhida pelo controlador. Tal técnica é classificada como do tipo inter-fluxo. O Algoritmo 2 representa a implementação desse procedimento.

---

#### Algorithm 2 Algoritmo implementando a política **Análise de Tráfego**

---

```

1: function INTERFACE COM MENOR TRAFEGO(switch)
2:   Recupera o tráfego de todas as interfaces
3:   Identifica a interface com menor utilização de tráfego
4:   return interface
5: end function

```

---

### 3.4.3. Round-Robin virtual

A diferença entre esta técnica e as duas anteriores é a quantidade de regras criadas. Ao invés de criar regras para apenas um fluxo para cada transmissão de uma origem para um destino, o controlador cria regras para todas as origens e todos os destinos utilizando todas as interfaces da agregação. Porém, cada regra é criada com prioridades diferentes, fazendo com que o *switch* escolha apenas uma interface para transmitir os pacotes.

Entretanto, de tempos em tempos, o controlador muda as prioridades de todas as regras, forçando o *switch* a mudar constantemente a interface que irá utilizar para transmitir os dados. Escolhemos que essa atualização seja feita a cada 0,2 segundos. Esse valor foi determinado após a realização de vários testes para identificar qual levaria ao melhor Índice Justiça[Jain et al. 1984], que é um indicador utilizado para determinar se os usuários ou aplicativos estão recebendo uma parcela justa de banda da rede. Essa técnica é do tipo inter-fluxo, mas um fluxo pode passar por duas interfaces distintas ao longo do tempo. O Algoritmo 3 representa a implementação desse procedimento.

---

#### Algorithm 3 Algoritmo implementando a política **Round-Robin virtual**

---

```

1: while True do
2:   próximo_índice  $\leftarrow$  (índice+1)%Número_de_Regras
3:   Regra[próximo_índice].prioridade  $\leftarrow$  50;  $\triangleright$  Aumenta prioridade da próxima regra
4:   Regra[índice].prioridade  $\leftarrow$  1;  $\triangleright$  Diminui prioridade da regra atual
5:   índice  $\leftarrow$  próximo_índice;  $\triangleright$  Atualiza índice da regra atual
6:   Sleep  $\tau$  ms
7: end while

```

---

## 4. Avaliação Experimental e Resultados

A proposta deste artigo inclui uma abordagem experimental. Para atingir os objetivos propostos, um ambiente virtual com três componentes foi testado: a plataforma Mininet, em sua versão mais recente, é capaz de criar redes virtuais complexas para rodar as simulações de transmissão de dados; o protocolo de comunicação Openflow, na versão 1.4, é utilizado para controlar os equipamentos SDN; a plataforma Ryu, na versão 4.9, é o controlador SDN que utiliza a linguagem Python.

Utilizamos a ferramenta *iperf* para gerar tráfego TCP e UDP entre todas as estações e servidores das topologias. Os testes variam entre apenas uma estação enviando dados até todas as estações enviando dados. O *iperf* não considera os custos adicionais dos protocolos como, por exemplo, os cabeçalhos *Ethernet*, IP, UDP ou TCP. Por isso, nos resultados apresentados, a banda máxima teórica disponível não foi atingida.

### 4.1. Topologia sem Agregação

Durante os testes iniciais, uma topologia sem agregação de enlaces foi criada para identificar o impacto da transmissão simultânea de pacotes entre várias origens e vários destinos. Tal topologia foi criada inspirada em um laboratório de nossa comunidade: oito estações distintas estão ligadas em um *switch* e quatro servidores estão ligados em outro *switch*. Todas as interfaces possuem velocidade máxima de transmissão de 100 Mbps e os dois *switches* estão ligados entre si. A Figura 3, substituindo a agregação por um único enlace, representa a topologia do laboratório.

A Tabela 1 representa as variações da banda (em Mbps) utilizada para as transmissões TCP de cada estação sem agregação de enlaces. A coluna banda utilizada representa o total das transmissões simultâneas. A coluna índice de Justiça (ou índice Jain) varia de a 0 (pior caso) a 1 (melhor caso) e é uma métrica para indicar o quão uniforme é a distribuição. Também fizemos experimento com UDP mas omitimos por espaço.

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça	
	1	2	3	4	5	6	7	8			
1	96,0	-	-	-	-	-	-	-	-	96,0	100,00%
2	53,80	27,8	-	-	-	-	-	-	-	96,20	99,82%
3	34,70	22,1	25,5	-	-	-	-	-	-	96,90	99,46%
4	20,90	14,8	17,1	14,5	-	-	-	-	-	96,90	99,12%
5	15,00	13,0	13,8	14,2	11,8	-	-	-	-	97,30	97,88%
6	7,49	15,2	12,1	12,4	12,1	11,7	-	-	-	97,19	98,69%
7	11,50	9,6	9,9	9,5	9,9	9,7	10,6	-	-	98,30	99,88%
8	11,00	8,6	9,5	9,7	11,5	9,8	8,4	7,7	-	98,18	98,70%

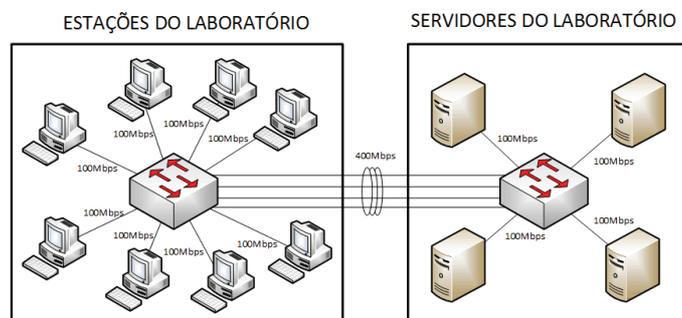
**Tabela 1. Transmissões TCP simultâneas sem agregação de enlaces (Valores em Mbps)**

Apesar de possuir um índice de justiça relativamente alto, quando todas as estações transmitem dados simultaneamente, elas utilizam em média uma banda de apenas 9,52 Mbps.

### 4.2. Tabela Hash

Em seguida, uma nova topologia foi criada no ambiente virtual utilizando a agregação de enlaces em quatro interfaces entre os *switches* interligados. Vale ressaltar que tal configuração não é possível de ser criada sem a utilização de um protocolo de agregação.

A Figura 3 representa essa nova topologia. Com isso, a banda disponível para transmissão de dados entre as estações e os servidores quadruplicou.



**Figura 3. Topologia do laboratório com agregação de enlaces**

Os resultados obtidos durante os testes dos algoritmo Tabela *Hash* são exibidos nas Tabelas 2 e 3. Nota-se que, em diversos casos, mesmo com o aumento de transmissões simultâneas, algumas estações foram beneficiadas com uma banda disponível maior do que as outras. Isso ocorre pelo fato dos fluxos serem distribuídos entre as interfaces da agregação fazendo com que a banda total consumida pela agregação não seja uniforme.

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça	
	1	2	3	4	5	6	7	8			
1	83,6	-	-	-	-	-	-	-	-	83,6	100,00%
2	83,6	83,9	-	-	-	-	-	-	-	167,5	100,00%
3	75,3	29,9	30,4	-	-	-	-	-	-	135,6	81,85%
4	76,5	86,3	32,9	32,6	-	-	-	-	-	228,3	84,36%
5	19,3	81,7	82,3	16,5	20,8	-	-	-	-	220,6	67,01%
6	31,9	29,6	31,7	31,4	33,4	32,4	-	-	-	190,4	99,87%
7	11,6	34,2	87,9	87,8	34,0	24,8	29,2	-	-	309,5	70,67%
8	28,0	84,2	18,2	84,4	18,8	9,6	17,8	40,0	-	301,0	64,00%

**Tabela 2. Transmissões TCP simultâneas com agregação de enlaces utilizando o algoritmo Tabela *Hash* (Valores em Mbps)**

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça	
	1	2	3	4	5	6	7	8			
1	88,5	-	-	-	-	-	-	-	-	88,5	100,00%
2	91,3	89,9	-	-	-	-	-	-	-	181,2	99,99%
3	53,5	95,5	42,6	-	-	-	-	-	-	191,6	88,69%
4	96,1	51,9	44,1	95,3	-	-	-	-	-	287,4	89,95%
5	43,7	45,6	95,9	52,1	52,2	-	-	-	-	289,5	90,00%
6	34,4	59,4	37,1	22,6	96,1	40,6	-	-	-	290,2	80,29%
7	46,0	95,3	44,5	53,8	48,0	51,4	43,3	-	-	382,3	91,20%
8	93,4	47,1	32,7	50,0	48,9	46,1	33,5	32,5	-	384,2	87,01%

**Tabela 3. Transmissões UDP simultâneas com agregação de enlaces utilizando o algoritmo Tabela *Hash* (Valores em Mbps)**

O índice de justiça é bem menor do que a topologia sem agregação, e a média da banda utilizada durante a transmissão de dados de todas as estações é de 37,62 Mbps para pacotes TCP e 48,02 Mbps para pacotes UDP.

#### 4.3. Análise de Tráfego

Os resultados obtidos durante os testes do algoritmo Análise de Tráfego foram os piores em comparação aos outros e são exibidos nas Tabelas 4 e 5. Isso ocorre devido à criação

aleatória das regras instaladas nos *switches* a cada novo fluxo recebido. Além disso, o caminho de volta pode ser diferente do caminho de ida, o que impacta profundamente nas medições TCP.

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça	
	1	2	3	4	5	6	7	8			
1	81,7	-	-	-	-	-	-	-	-	81,7	100,00%
2	77,9	71,9	-	-	-	-	-	-	-	149,8	99,84%
3	31,3	77,1	26,7	-	-	-	-	-	-	135,1	79,66%
4	80,4	27,2	12,6	20,5	-	-	-	-	-	140,7	63,59%
5	39,1	86,1	86,8	13,5	17,0	-	-	-	-	242,5	69,40%
6	21,3	30,9	88,2	16,8	24,2	33,4	-	-	-	214,8	68,84%
7	33,1	13,3	28,5	37,0	24,4	29,2	48,9	-	-	214,4	90,04%
8	82,9	83,9	14,6	14,6	13,7	5,6	7,8	7,7	-	230,8	45,36%

**Tabela 4. Transmissões TCP simultâneas com agregação de enlaces utilizando o algoritmo Análise de Tráfego (Valores em Mbps)**

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça	
	1	2	3	4	5	6	7	8			
1	84,1	-	-	-	-	-	-	-	-	84,1	100,00%
2	93,4	93,5	-	-	-	-	-	-	-	186,9	100,00%
3	95,5	43,7	51,8	-	-	-	-	-	-	191,0	88,68%
4	96,8	46,2	50,4	96,3	-	-	-	-	-	289,7	89,98%
5	95,0	96,2	96,2	47,2	47,7	-	-	-	-	382,3	91,24%
6	96,2	49,5	38,6	96,2	57,2	46,3	-	-	-	384,0	88,20%
7	28,9	47,1	94,8	49,3	30,7	90,0	37,3	-	-	378,1	82,00%
8	24,3	50,7	22,8	28,9	91,2	45,5	20,2	92,6	-	376,2	74,06%

**Tabela 5. Transmissões UDP simultâneas com agregação de enlaces utilizando o algoritmo Análise de Tráfego (Valores em Mbps)**

A média da banda utilizada durante a transmissão de dados de todas as estações é de 28,85 Mbps para pacotes TCP e 47,02 Mbps para pacotes UDP. O índice de justiça é o menor de todos e no caso dos pacotes TCP corresponde à metade do índice da topologia sem agregação.

#### 4.4. Round-Robin Virtual

Os resultados obtidos utilizando o algoritmo *Round-Robin* virtual foram os mais promissores. Através dele, foi possível atingir um balanceamento de tráfego quase uniforme. As Tabelas 6 e 7 exibem os resultados desse algoritmo.

Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça	
	1	2	3	4	5	6	7	8			
1	83,1	-	-	-	-	-	-	-	-	83,1	100,00%
2	88,9	88,9	-	-	-	-	-	-	-	177,8	100,00%
3	94,5	93,9	86,0	-	-	-	-	-	-	274,4	99,82%
4	75,5	78,9	75,0	84,8	-	-	-	-	-	314,2	99,75%
5	49,0	72,7	67,0	59,4	54,9	-	-	-	-	303,0	98,10%
6	37,5	52,7	53,9	75,2	40,7	69,0	-	-	-	329,0	94,15%
7	61,3	43,1	39,0	52,2	34,7	51,3	49,6	-	-	331,2	96,99%
8	27,5	33,9	46,3	57,3	57,8	45,2	30,9	40,2	-	339,1	94,02%

**Tabela 6. Transmissões TCP simultâneas com agregação de enlaces utilizando o algoritmo Round-Robin Virtual (Valores em Mbps)**

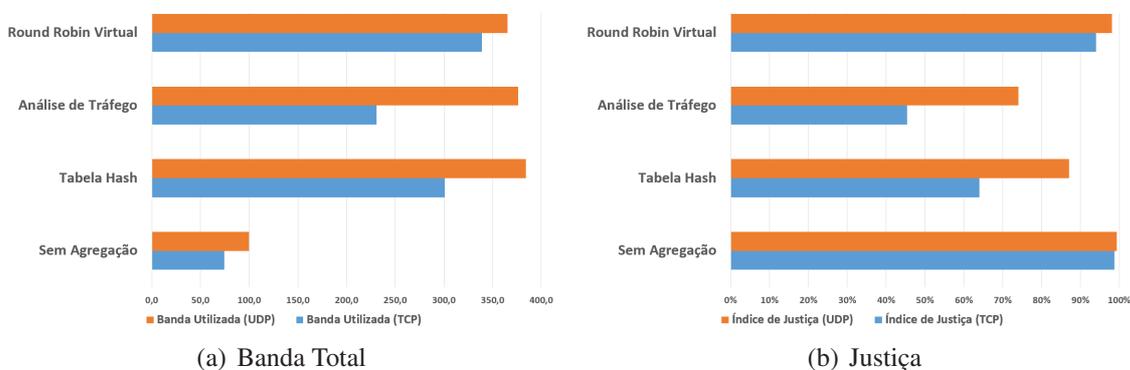
Transmissões Simultâneas	Banda utilizada por cada estação								Banda Utilizada	Índice de Justiça
	1	2	3	4	5	6	7	8		
1	99,9	-	-	-	-	-	-	-	99,9	100,00%
2	99,6	99,0	-	-	-	-	-	-	198,6	100,00%
3	94,2	90,7	90,3	-	-	-	-	-	275,2	99,96%
4	80,1	82,2	87,6	88,9	-	-	-	-	338,8	99,81%
5	70,6	70,8	61,7	71,3	65,6	-	-	-	340,0	99,69%
6	40,0	70,8	60,3	64,1	58,5	66,5	-	-	360,2	97,40%
7	51,8	38,0	58,7	55,4	44,5	53,6	62,8	-	364,8	97,80%
8	37,9	58,6	46,3	50,6	48,5	41,5	41,1	40,8	365,3	98,10%

**Tabela 7. Transmissões UDP simultâneas com agregação de enlaces utilizando o algoritmo *Round-Robin Virtual* (Valores em Mbps)**

O índice de justiça desse algoritmo ficou acima dos 94% nos testes TCP e acima dos 98% nos testes UDP. Além disso, as médias das bandas utilizadas durante as transmissões de dados de todas as estações foram de 42,38 Mbps e 45,66 Mbps, respectivamente. Esses valores correspondem a 84% e 91,4% da média de 50 Mbps, caso o balanceamento fosse uniforme para todas as transmissões.

#### 4.5. Discussão

A Figura 4(a) representa o comparativo das bandas totais utilizadas e a Figura 4(b) representa o comparativo dos índices de Justiça em cada teste. Por esses resultados podemos identificar que o algoritmo *Round Robin Virtual* teve desempenhos semelhantes entre o tráfego TCP e UDP tanto na distribuição de banda quanto no índice de justiça.

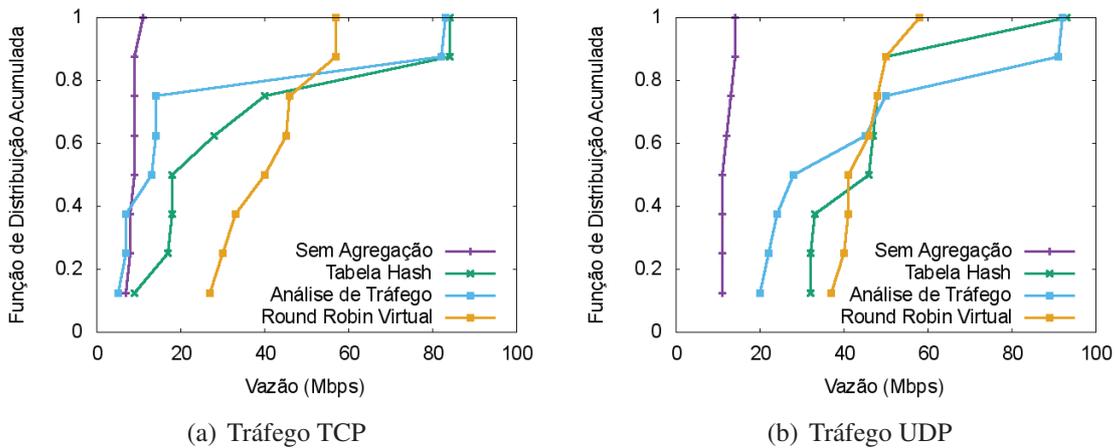


**Figura 4. Comparativo dos algoritmos de agregação**

As Figuras 5(a) e 5(b) representam a função de distribuição acumulada dos testes com oito fluxos simultâneos. Os gráficos demonstram que, tanto para o tráfego TCP quanto para o UDP, apenas o algoritmo *Round Robin Virtual* teve uma distribuição de tráfego tão uniforme como a topologia inicial com gargalo.

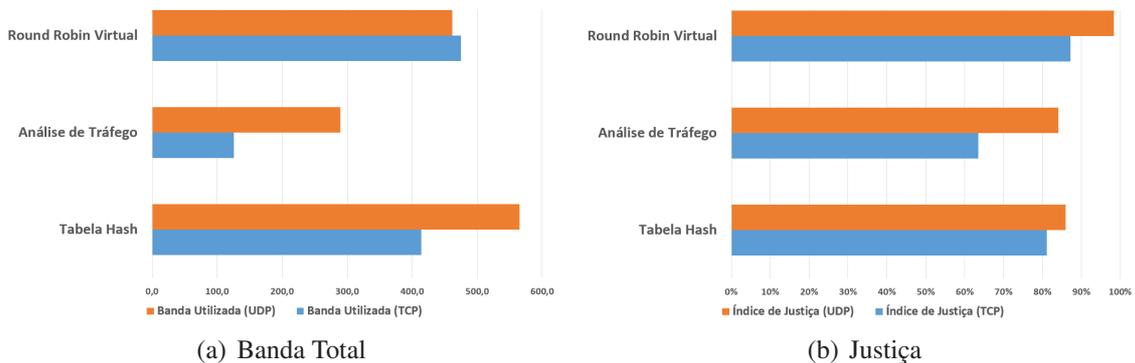
#### 4.6. Agregação com nove enlaces

Para fins de critérios comparativos entre nossa solução e o LACP, foi criada uma nova topologia semelhante à topologia da Figura 3 com algumas modificações: a adição de uma nona estação e a configuração de nove enlaces (oito com banda de 100 Mbps e um com banda de 1 Gbps). Essas modificações foram feitas para suprir a limitação do LACP de não ser capaz de agregar mais de oito enlaces e nem de utilizar enlaces com bandas diferentes. A Figura 6(a) representa o comparativo das bandas totais utilizadas e a



**Figura 5. Função de Distribuição Acumulada**

Figura 6(b) representa o comparativo dos índices de justiça nos testes nessa topologia. Nota-se que, em alguns resultados, a banda é superior a 400 Mbps. Isso ocorre devido ao tempo de execução dos testes, fazendo com que algum fluxo possa terminar antes de outros aumentando a vazão final.



**Figura 6. Comparativo dos algoritmos de agregação**

## 5. Conclusão e Trabalhos Futuros

Os resultados obtidos neste trabalho mostram que a nova abordagem proposta para agregação de enlaces em ambientes de redes SDN apresenta um melhor desempenho. O dinamismo, a complexidade e a escalabilidade dos ambientes de redes atuais justificam a necessidade de encontrar uma nova solução para o antigo problema de gargalo, que é a principal contribuição aqui apresentada.

A decisão de modificar o caminho dos pacotes de tempos em tempos é a melhor forma de balancear o tráfego entre todas as interfaces de uma agregação, dentre as propostas aqui discutidas. Isso fez com que a distribuição se tornasse mais justa e que não houvesse uma sobrecarga em algumas interfaces ou subutilização de outras, como pode ocorrer no protocolo LACP. Além disso, a nossa solução foi capaz de utilizar 84% da banda disponível transmitindo tráfego TCP e 91,4% transmitindo tráfego UDP. O índice

de justiça também foi bastante satisfatório, ultrapassando 94% nos tráfegos TCP e 98% nos tráfegos UDP.

Também foi implementada uma aplicação capaz de criar e remover enlaces da agregação de maneira dinâmica. Através de um simples comando, o controlador SDN consegue reconfigurar a agregação entre um par de *switches* sem a necessidade de configurações adicionais por parte do administrador de redes. Vale lembrar que, na solução clássica, as novas interfaces devem ser inseridas manualmente na interface virtual criada pelo protocolo.

Vale salientar que existem limitações para configurar o protocolo LACP no ambiente virtual Mininet, por isso foi realizada a implementação de um algoritmo similar que chamados de Tabela Hash. Pretendemos configurar um ambiente real para simulações, e com isso realizar os devidos testes com o estado da arte, a fim de validar nossa abordagem.

Como trabalho futuro, pretendemos integrar ao sistema uma ferramenta de predição de tráfego para permitir engenharia de tráfego em tempo real. Além disso, a reconfiguração da agregação quando uma interface se torna indisponível precisa ser implementada. Por fim, o impacto da reescrita dos fluxos de tempos em tempos precisa ser avaliada, já que isso pode ser o causador de eventuais atrasos na transmissão de pacotes.

## Referências

- Ahn, J. H., Binkert, N., Davis, A., McLaren, M., and Schreiber, R. S. (2009). Hyperx: topology, routing, and packaging of efficient large-scale networks. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–11.
- Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., and Vahdat, A. (2010). Hedera: Dynamic flow scheduling for data center networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 19–19, Berkeley, CA, USA. USENIX Association.
- Alizadeh, M., Greenberg, A., Maltz, D., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and Sridharan, M. (2010). Dctcp: Efficient packet transport for the commoditized data center. Technical report.
- Bredel, M., Bozakov, Z., Barczyk, A., and Newman, H. (2014). Flow-based load balancing in multipathed layer-2 networks using openflow and multipath-tcp. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, pages 213–214, New York, NY, USA. ACM.
- Carter, R. L. and Crovella, M. E. (1996). Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation*, 27–28:297—318.
- Floyd, S. and Jacobson, V. (1995). Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3:365–386.
- Jain, R., Chiu, D., and Hawe, W. (1984). A quantitative measure of fairness and discrimination for resource allocation in shared systems. *Digital Equipment Corporation. Tech. rep., Technical Report DEC-TR-301*.

- Judd, G. (2015). Attaining the promise and avoiding the pitfalls of tcp in the datacenter. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 145–157, Oakland, CA. USENIX Association.
- Moore, J. T. and Nettles, S. M. (2001). Towards practical programmable packets. In *in Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, pages 41–50.
- Pfaff, B., Pettit, J., Amidon, K., Casado, M., Koponen, T., and Shenker, S. (2009). Extending networking into the virtualization layer. In *Hotnets*.
- Ranum, M. J., Landfield, K., Stolarchuk, M., Sienkiewicz, M., Lambeth, A., and Wall, E. (1998). Implementing a generalized tool for network monitoring. *Information Security Technical Report*, 3:53—64.
- Seaman, M. (1999). Link aggregation control protocol. *IEEE* <http://grouper.ieee.org/groups/802/3/ad/public/mar99/seaman>, 1:0399.
- Steinbacher, M. and Bredel, M. (2015). Lacp meets openflow – seamless link aggregation to openflow networks. *TNC15 Networking Conference*.
- Vencioneck, R. D., Vassoler, G., Martinello, M., Ribeiro, M. R. N., and Marcondes, C. (2014). Flexforward: Enabling an sdn manageable forwarding engine in open vswitch. *10th International Conference on Network and Service Management (CNSM) and Workshop*, pages 296–299.