Prototipação de Solução de Roteamento de Fonte para Suporte à Engenharia de Tráfego em Ciência de Dados Intensiva

Domingos José Pereira Paraiso[1], Everson Borges[1,2], Edgard Pontes[2], Cristina Klippel Domincini[1], Magnos Martinello[2], Moisés Ribeiro[2]

[1]Instituto Federal do Espírito Santo (IFES)

[2]Universidade Federal do Espírito Santo (UFES)

domingos.paraiso@gmail.com, cristina.dominicini@ifes.edu.br

Abstract. *This article proposes a virtual environment for testing Traffic Engineering mechanisms for Data Intensive Science applications based on the RARE/freeRtr platform, allowing complex experiments to be tested before being transferred to a physical testbed with programmable P4 switches. The study defines network requirements for Data Intensive Science and demonstrates how these requirements can be met by the PolKA Source Routing protocol, enabling agile route migration, fault tolerance and policy-based load balancing.*

Resumo. *Este artigo propõe um ambiente virtual de experimentação de mecanismos de Engenharia de Tráfego para aplicações de Ciência de Dados Intensiva baseado na plataforma RARE/freeRtr, permitindo que experimentos complexos possam ser testados antes de serem transferidos para um testbed físico com switches programáveis P4. O estudo define os requisitos das redes para Ciência de Dados Intensiva e demonstra como esses requisitos podem ser atendidos pelo protocolo de Roteamento de Fonte PolKA, habilitando migração ágil de rotas, tolerância à falhas e balanceamento de carga baseado em políticas.*

1. Introdução

Pesquisas científicas atuais têm gerado conjuntos de dados digitais na escala de petabytes por meio de detectores, simulações e análises [Zurawski et al. 2021]. Muitas destas pesquisas exigem colaboração com outras instituições e pesquisadores geograficamente dispersos pelo mundo [Babik et al. 2020]. A troca de informações entre os envolvidos exige que grande parte desse enorme volume de dados seja compartilhado para consumo, análise e apresentação de resultados em tempo hábil para dar suporte a aplicações de DIS (Ciência de Dados Intensiva, do inglês *Data Intensive Science*) [Guiang et al. 2022].

Neste contexto, existem várias redes de alto desempenho que interconectam centros de pesquisa, como a ESnet nos Estados Unidos [Xiang et al. 2018] e a GÉANT [Valera-Muros et al. 2019] na Europa. Para atender os níveis de QoS (Qualidade de Serviço, do inglês *Quality of Service*) exigidos para essas redes, devem ser escolhidas ferramentas e métodos que permitam um uso mais eficiente dos recursos disponíveis. Em especial, como as aplicações de DIS exigem a troca de grandes volumes no menor tempo possível com a utilização de diversos fluxos agregados, a largura de banda disponível deve ser o mais próxima da sua capacidade total, utilizando todos os caminhos disponíveis. Essas aplicações requerem redes com tolerância a falhas, roteamento baseado em políticas de acordo com as classes de fluxos e administração centralizada, ágil e simplificada.

Uma solução para atendimento a esses requisitos rigorosos das aplicações de DIS é o uso da TE (Engenharia de Tráfego, do inglês *Traffic Engineering*) na rede [Babik et al. 2020]. Entretanto, os mecanismos dos protocolos tradicionais de roteamento não permitem atualizar dinamicamente os melhores caminhos para utilizar a capacidade total da rede de forma mais eficiente [Dominicini et al. 2020]. Para preencher essa lacuna, uma alternativa promissora é o uso de protocolos do tipo SR (Roteamento de Fonte, do inglês *Source Routing*) [Dominicini et al. 2021], que permitem a troca dinâmica de rotas de forma mais ágil, porque só necessitam atualizar as regras nos nós de entrada, que inserem um rótulo de rota no cabeçalho dos pacotes para selecionar um caminho, enquanto os nós internos só precisam fazer operações neste rótulo. Uma proposta inovadora existente na literatura é o protocolo PolKA [Dominicini et al. 2020], que explora o CRT (Teorema Chinês dos Restos, do inglês *Chinese Remainder Theorem*) para calcular o rótulo de rota. Cada nó do núcleo da rede calcula a porta de saída por meio de uma operação de resto da divisão entre o rótulo da rota e o identificador do nó.

O protocolo PolKA foi testado em equipamentos comerciais programáveis distribuídos na Europa conectados por enlaces de 10Gbps [Dominicini et al. 2021], com o mesmo desempenho no plano de dados dos métodos de encaminhamentos tradicionais. Posteriormente, [Borges et al. 2022c] demonstrou em uma topologia emulada a integração do PolKA com a plataforma RARE/freeRtr como forma de automatizar o plano de controle e permitir a criação de túneis de forma ágil e simplificada. Por fim, [Borges et al. 2022b] demonstrou a criação de uma rede *overlay* com túneis PolKA para validar a transferência intensiva de dados em 10Gbps e 100Gbps em um *testbed* composto por *switches* programáveis na Europa, Estados Unidos e Brasil. Os experimentos preliminares mostraram que foi possível classificar e balancear os fluxos com altas taxas de tráfego usando mecanismos de PBR (Roteamento Baseado em Política, do inglês *Policy-Based Routing*) para explorar os múltiplos caminhos da rede. Esse último trabalho também mostrou que existe uma lacuna para conseguir implementar nos *testbeds* físicos as soluções projetadas no ambiente emulado, pois o ambiente emulado não consegue representar as particularidades do ambiente físico. Como existe acesso limitado para uso de um *testbed* global que envolve múltiplos parceiros, o resultado é um alto tempo de ocupação dos recursos compartilhados para executar experimentos complexos.

Com essa motivação, nossa primeira contribuição é definir requisitos das redes para DIS (Seção 2) e projetar experimentos que mostrem como estes requisitos podem ser atendidos por protocolos de SR. A segunda contribuição é um ambiente virtual de experimentação de protocolos de roteamento para aplicações de DIS baseado na plataforma RARE/freeRtr (Seção 4), que permita que experimentos complexos possam ser testados para posteriormente serem transferidos para um *testbed* físico composto por switches programáveis P4 com o mínimo tempo de implantação. Por fim, a terceira contribuição é explorar o ambiente virtual proposto para avaliar se o protocolo PolKA (Seção 3) consegue atender aos requisitos levantados, tais como agilidade na mudança de rotas, tolerância à falhas e balanceamento de carga baseado em políticas (Seção 5).

2. Requisitos de rede para aplicações de DIS

O artigo [Ioannou et al. 2020] mostra que projetos de pesquisas em física de alta energia, genoma e astronomia geram um grande volume de dados que precisam ser transferidos para outros locais. Ele cita o uso de *Data Transfer Nodes* (DTNs), que são conjuntos de

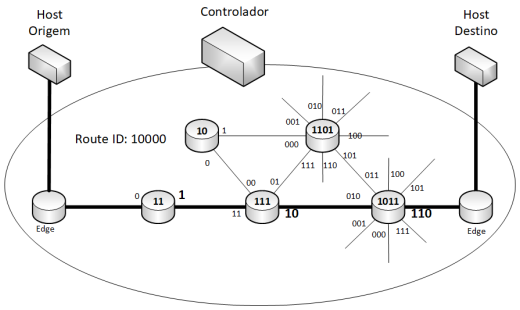

Figura 1. Protocolo PolKA.

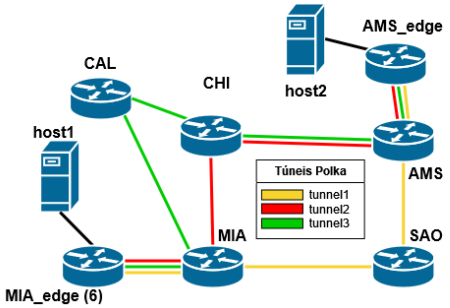

Figura 2. Topologia.

hardware e software com grande velocidade de leitura e gravação, além de ferramentas de transferência de dados, que precisam atingir taxas de 100Gbps. Os requisitos elencados são: uma rede de alta capacidade com políticas específicas para os diferentes fluxos das pesquisas de acordo com classes de serviço, além de DTNs de alto desempenho. O Departamento de Energia dos Estados Unidos gerencia uma rede que integra pesquisas de diversos locais do mundo (e.g., Chile, Nova Iorque, Los Angeles, Hong Kong e Singapura). Em [Zurawski et al. 2021], é apresentado um relatório com os requisitos das aplicações de DIS, que compartilham um volume de dados muito alto usando enlaces de alta capacidade (100Gbps) e compartilham grandes volumes de dados (entre 30 e 1536 TB/dia). Cada pesquisa usa diferentes protocolos de transferência de arquivos, que funcionam sobre o protocolo TCP. Dessa forma, é necessário gerenciar várias conexões simultâneas com diversos fluxos de dados e explorar de forma eficiente a capacidade da rede.

Assim, podemos elencar os seguintes mecanismos que a Engenharia de Tráfego deve fornecer para habilitar aplicações de DIS: (i) configuração simples de túneis que permitam selecionar qualquer caminho para uso otimizado dos recursos de rede, (ii) classificação de fluxos de tráfego de acordo com classes de serviço, e (iii) seleção e migração ágil de caminhos baseada em políticas que considerem as classes de serviço. Nas próximas seções, demonstramos no ambiente virtual proposto que o protocolo de SR PolKA integrado à plataforma RARE/freeRtr é capaz de fornecer esses mecanismos.

3. O protocolo PolKA

No protocolo de SR PolKA [Dominicini et al. 2021], o controlador usa o CRT para explicitar todos os nós por onde o pacote deve passar e calcular um rótulo de rota (*routeID*), que será adicionado ao pacote no nó de entrada. Em cada nó do núcleo da rede, a porta de saída (*portid*) é obtida por uma operação de resto da divisão entre o *routeID* e o identificador do nó (*nodeID*). A Figura 1 mostra um exemplo de uma rota com *routeID* 10000, representada pelo caminho em destaque[1]. Em cada nó, é calculado o resto da divisão de 10000 pelos identificadores dos nós, resultando na porta de saída. No nó 11, o resto da divisão resulta em 1, o pacote segue para o nó 111. O resto da divisão agora é 10 e o pacote segue para o nó 1011, onde o cálculo resulta em 110 e o pacote chega ao nó de destino. A capacidade do protocolo PolKA representar qualquer caminho de rede por meio de um rótulo de rota será explorada neste trabalho para criar túneis que podem ser configurados de forma ágil com uma simples modificação na origem do tráfego.

[1]Todos os números representam polinômios binários em GF(2) [Dominicini et al. 2020].

```
interface tunnel3                                    |  access-list fluxo3
 description POLKA tunnel MIA -> CAL -> CHI -> AMS   |   sequence 10 permit  6 40.40.1.0 255.255.255.0 \
 tunnel vrf v1                                       |          all 40.40.2.2 255.255.255.255 all tos 128
 tunnel source loopback0                             |  exit
 tunnel destination 20.20.0.7                        |
 tunnel domain-name 20.20.0.1 20.20.0.5 20.20.0.4 20.20.0.3  |
 tunnel mode polka                                   | ipv4 pbr v1 sequence 30 fluxo3 v1 nexthop 30.30.1.2
 vrf forwarding v1                                   |
 ipv4 address 30.30.3.1 255.255.255.252              |
 no shutdown                                         |
 no log-link-change                                  |
 exit                                                |
```

Figura 3. Configuração.

IP src	Link	IP dst
10.10.1.1	MIA SAO	10.10.1.2
10.10.2.1	MIA CHI	10.10.2.2
10.10.3.1	MIA CAL	10.10.3.2
10.10.4.1	MIA MIAedge	10.10.4.2
10.10.5.1	SAO MAS	10.10.5.2
10.10.6.1	AMS AMSedge	10.10.6.2
10.10.7.1	AMS CHI	10.10.7.2
10.10.8.1	CHI CAL	10.10.8.2
40.40.1.2	host1 MIAedge	40.40.1.1
40.40.2.2	host2 AMSedge	40.40.2.1

Loopback	
MIA	20.20.0.1
SAO	20.20.0.2
AMS	20.20.0.3
CHI	20.20.0.4
CAL	20.20.0.5
MIAedge	20.20.0.6
AMSedge	20.20.0.7

Túnel PolKA		
	MIA	AMS
tunnel1	30.30.1.1	30.30.1.2
tunnel2	30.30.2.1	30.30.2.2
tunnel3	30.30.3.1	30.30.3.2

Tabela 1. Tabela de Endereços.

4. Ambiente virtual de experimentação

O RARE/freeRtr é um software de controle de rede, usa *sockets* UDP e suporta protocolos tradicionais e novos, como o PolKA [Borges et al. 2022a]. Pode ser usado para emular redes ou como plano de controle para dispositivos de hardware, e suporta diferentes plano de dados, como DPDK e P4. É possível emular topologias e testar soluções de rede antes de implementar em um ambiente real.

O ambiente de testes utilizado foi um computador com processador Intel i7, 12Gb de RAM, Linux Debian 11 e VirtualBox 7. A topologia utilizada neste trabalho representa um subconjunto dos nós do *testbed* o Global P4 Lab, que possui *switches* programáveis P4 na Europa, Estados Unidos e Brasil, conforme Figura 2. Para emular essa topologia, criamos 9 VMs (Máquinas Virtuais, do inglês Virtual Machines) com 1Gb de RAM rodando Debian 11. Nas VMs que funcionaram como roteadores instalamos o RARE/freeRtr .

Criamos uma VM de *template* no VirtualBox com a seguinte instalação: Debian 11, RARE/freeRtr, iperf3 e bwm-ng. Posteriormente, essa VM foi clonada e usamos scripts para automatizar a customização dos arquivos de configuração do RARE/freeRtr para cada nó. Para emular a topologia usamos o recurso de rede interna disponibilizado pelo VirtualBox. Os endereços IP das interfaces usados nos experimentos estão indicados na Tabela 1, os arquivos com as configurações do RARE/freeRtr, do VirtualBox, além dos *scripts* que automatizam os experimentos estão disponíveis no github[2].Não foi possível configurar o atraso de propagação das interfaces de rede exigido pelos experimentos, assim configuramos as interfaces entre os roteadores MIA e SAO para usar interfaces de rede ethernet da máquina física e inserimos um atraso de 20ms no sistema operacional da máquina hospedeira pelo comando tc. Para limitar a taxa de transmissão, usamos um recurso nativo do VirtualBox que permite criar diferentes limites nas interfaces de rede.

A criação de túneis PolKA, o controle de acesso e a política de roteamento PBR são configurados nos roteadores de borda, de acordo com os comandos freeRtr mostrados na Figura 3. No exemplo, a seção *access-list* indica que a rede 40.40.1.0/24 pode acessar a máquina 40.40.2.2 usando o protocolo 6 (TCP), o ToS (Tipo de Serviço, do inglês *Type of Service*) indicado no final do comando filtra apenas os pacotes com essa indicação. O túnel é criado na seção *interface* e a configuração *tunnel destination 20.20.0.7* indica que o túnel vai até o roteador de borda AMS, enquanto o *tunnel domain-name* informa a lista de roteadores que fazem parte do caminho explícito, que será internamente convertido pelo freeRtr em um *routeID* PolKA a ser encapsulado nos pacotes que passam pelo túnel. Na última linha, uma PBR é criada indicando que o controle de acesso *fluxo3* vai usar o túnel 3, pois foi informado o endereço 30.30.3.2 que é o IP do outro lado.

[2]https://github.com/domingosparaiso/wpeif-2023/

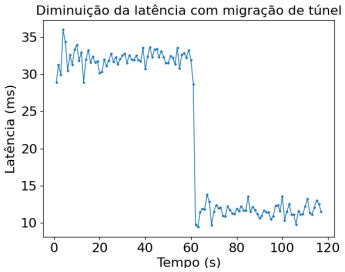

Figura 4. Exp. 1

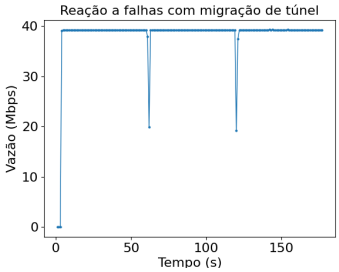

Figura 5. Exp. 2

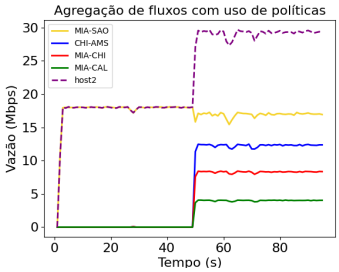

Figura 6. Exp. 3

5. Experimentos endereçando os requisitos de DIS

Foram realizados três experimentos usando a topologia da Figura 2. É importante ressaltar que o protocolo PolKA permite selecionar um túnel configurado na borda da rede por meio da seleção de uma PBR, sem nenhuma modificação no núcleo da rede. O primeiro experimento mostra a migração ágil para um caminho de menor latência para otimizar o desempenho de um determinado fluxo, requisito (i). Configuramos a PBR que redirecionou o fluxo para o túnel (MIA-SAO-AMS), por 1 minuto o comando *ping* enviou pacotes ICMP entre *host1* e *host2*, alteramos a PBR para encaminhar o fluxo pelo túnel (MIA-CHI-AMS) por mais 1 minuto. O gráfico da Figura 4 mostra a média de 10 rodadas de teste de latência em função do tempo.

O segundo experimento mostra a resposta ágil da rede na troca de caminhos para um fluxo de dados de 50Mbps. Configuramos uma PBR para usar o túnel 1 entre as máquinas *host1* e *host2*, transferimos dados UDP durante 2 minutos, trocamos a PBR para usar os túneis 2 e 3 por 2 minutos cada. Foram feitas 20 rodadas de teste e a vazão média dos resultados é mostrada na Figura 5. Podemos observar que o fluxo se manteve estável e a queda de vazão durante as trocas que foi recuperada rapidamente. Este mecanismo ágil pode ser explorado para otimizar o desempenho de um determinado fluxo quando algum caminho de melhor capacidade estiver disponível ou para reagir a uma falha no caminho original de um determinado fluxo atendendo ao requisito (ii).

O terceiro experimento mostra como a distribuição de fluxos por caminhos distintos pode evitar limitações da rede. Para simular diferentes capacidades, limitamos as vazões dos enlaces: MIA-SAO, SAO-AMS e CHI-AMS em 20Mbps, MIA-CHI em 10Mbps, MIA-CAL e CAL-CHI em 5Mbps. Geramos três fluxos TCP com diferentes ToS, entre *host1* e *host2*. Cada ToS foi associada a uma PBR, inicialmente todas usaram o túnel 1, fazendo com que a vazão máxima ficasse menor que 20Mbps. Após 4 minutos, trocamos uma PBR para o túnel 2, e a outra para o túnel 3. A Figura 6 apresenta a média da vazão de 20 rodadas e mostra um aumento na vazão total (30Mbps), pois os fluxos trafegam por caminhos distintos até chegar ao *host* final. Esse mecanismo pode ser usado tanto para agregação de fluxos, quanto para balanceamento de carga atendendo ao requisito (iii).

6. Conclusões e Trabalhos Futuros

Este artigo propôs e implementou um ambiente virtual de experimentação para protocolos de roteamento baseado no software RARE/freeRtr, que permite experimentos complexos com rapidez de implantação em *testbeds* reais. Os testes realizados mostraram que o

PolKA é capaz de atender aos requisitos de aplicações de DIS: (i) agilidade na mudança de rotas, (ii) tolerância a falhas e (iii) balanceamento de carga baseado em políticas. Como trabalhos futuros, planejamos replicar o ambiente na plataforma *OpenStack* para maior escalabilidade e lidar com maior volume de dados, transferir os experimentos do ambiente virtual para um *testbed* físico e comparar o PolKA com outros protocolos de SR.

Agradecimentos

Agradecemos a FAPES e a CAPES (processo 2021-2S6CD, FAPES 132/2021) por meio do PDPG (Programa de Desenvolvimento da Pós-Graduação, Parcerias Estratégicas nos Estados) e as agências: FAPESP/MCTI/CGI.br (PORVIR-5G 20/05182-3, SAWI 20/05174-0) e FAPES (94/2017, 281/2019, 515/2021, 284/2021, 06/2022, 1026/2022, 941/2022). Este trabalho recebeu apoio do 2021 Google Research Scholar Award.

Referências

Babik, M. et al. (2020). Network capabilities for the hl-lhc era. In *EPJ Web of Conferences*, volume 245, page 07051. EDP Sciences.

Borges, E. et al. (2022a). Freerouter in a nutshell: A protocoland routing platform for open and portable carrier-class testbeds. In *Anais do I Workshop de Testbeds*, pages 36–46, Porto Alegre, RS, Brasil. SBC.

Borges, E. S. et al. (2022b). Demonstrating polka routing approach to support traffic engineering for data-intensive science. In *The International Conference for High Performance Computing, Networking, Storage, and Analysis*.

Borges, E. S. et al. (2022c). A lifecycle experience of polka: From prototyping to deployment at géant lab with rare/freertr. In *Anais do XIII WPEIF*, pages 35–40. SBC.

Dominicini, C. et al. (2020). Polka: Polynomial key-based architecture for source routing in network fabrics. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pages 326–334. IEEE.

Dominicini, C. et al. (2021). Deploying polka source routing in p4 switches. In *2021 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–3. IEEE.

Guiang, J. et al. (2022). Integrating end-to-end exascale sdn into the lhc data distribution cyberinfrastructure. In *Practice and Experience in Advanced Research Computing*, pages 1–4. PEARC.

Ioannou, I. et al. (2020). Data transfer node (dtn) tests on the géant testbeds service (gts). *GN4-3 project*.

Valera-Muros, B. et al. (2019). Is géant testbeds service compliant with etsi mano? In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 502–507. IEEE.

Xiang, Q. et al. (2018). Fine-grained, multi-domain network resource abstraction as a fundamental primitive to enable high-performance, collaborative data sciences. In *SC18: International Conference for HPC, Networking, Storage and Analysis*, pages 58–70.

Zurawski, J. et al. (2021). 2020 high energy physics network requirements review final report. `https://escholarship.org/uc/item/78j3c9v4`.