

BAMSDN: Uma Ferramenta para a Exploração Dinâmica e Flexível de Recursos Baseada em Modelo de Alocação de Banda e SDN/OpenFlow

Eliseu Torres¹, Rafael F. Reale², Leobino N. Sampaio¹, Joberto S. B. Martins³

¹Programa de Pós-Graduação em Ciência da Computação (PGCOMP) – UFBA

²Instituto Federal da Bahia – IFBA

³Programa de Pós-Graduação em Sistemas e Computação (PPGCOMP) – UNIFACS

eliseu.torres@ufba.br, reale@ifba.edu.br,
leobino@ufba.br, joberto.martins@unifacs.br

Abstract. *Bandwidth allocation models (BAM) provide an efficient and practical mechanism for dynamic and flexible resource allocation in a network for application classes. This article presents BAMSDN, a tool that dynamically allocates the “bandwidth” resource in a MPLS-based network through the SDN paradigm and the OpenFlow protocol. The tool presents a modular and low-cost solution that allows a SDN controller to dynamically manage the Label Switched Paths (LSPs) allocation and their respective bandwidths on the OpenFlow switch output ports, but without requiring the development of intelligence or traffic engineering for the LSPs allocation. The environment supports the inclusion of multiple BAMs and other traffic engineering modules. BAMSDN was developed in the Python language and can be reused in any Linux environment with Mininet, OpenVswitch, OpenFlow and POX controller for different network scenarios.*

Resumo. *Modelos de alocação de banda (BAM) oferecem um mecanismo eficiente e prático para a alocação dinâmica e flexível de recursos numa rede para classes de aplicações. Este artigo apresenta BAMSDN, uma ferramenta que, através de BAMs, aloca o recurso “banda” dinamicamente para uma rede MPLS através do paradigma SDN e protocolo OpenFlow. A ferramenta apresenta uma solução modular e de baixo custo que permite um controlador SDN gerenciar, de forma dinâmica, a alocação de LSPs (Label Switched Paths) e suas respectivas larguras de banda nas portas de saída de switches OpenFlow, sem a necessidade do desenvolvimento de uma inteligência ou engenharia de tráfego. O ambiente suporta a inclusão de múltiplos BAMs e outros módulos de engenharia de tráfego. A BAMSDN foi desenvolvida na linguagem Python e pode ser reutilizada em qualquer ambiente Linux com Mininet, OpenVswitch, OpenFlow e controlador POX para diferentes cenários de rede.*

1. Introdução

Em cenários de redes de comunicação atuais e emergentes, tais como 5G, nuvem, redes ópticas, redes MPLS e Internet das Coisas (IoT), a alocação dinâmica, flexível e

customizada de recursos é fundamental para lidar com a dinamicidade do tráfego de entrada das redes. Os modelos de alocação de banda (do inglês, *Bandwidth Allocation Models* – BAM) possuem os atributos para resolver tais requisitos, uma vez que permitem a definição de classes de aplicações e o controle da distribuição dos recursos entre as classes [Martins et al. 2015].

A implementação de BAMs ganha novas perspectivas com o paradigma das Redes Definidas por Software (*Software-Defined Networks* – SDN) [Kreutz et al. 2014], que prevê programabilidade e controle logicamente centralizado da rede. A partir de um controlador é possível utilizar um BAM na otimização da alocação de recursos de redes (banda) através do agrupamento de serviços para diferentes classes de aplicações, problema frequentemente encontrado nas redes corporativas e provedores de serviços. Assim, os altos custos de OPEX e CAPEX de redes puramente MPLS são reduzidos.

Em função dos benefícios do uso conjunto de modelos de alocação de banda e o paradigma SDN, o presente artigo apresenta a ferramenta BAMSDN (*Bandwidth Allocation Models through Software-Defined Networks*). A ferramenta aloca o recurso “banda” dinamicamente para uma rede MPLS, utilizando BAMs em redes definidas por software, implementadas através do protocolo OpenFlow. Ao implementar BAMs através de SDN é possível superar dificuldades inerentes ao problema de alocação de recursos por meio de uma solução de baixo custo. A programabilidade da rede e sua visão global a partir do plano de controle possibilita à BAMSDN adequar a disponibilidade dos recursos à demanda dos usuários de aplicações, através do gerenciamento dinâmico da alocação de LSPs (*Label Switched Paths*) e suas respectivas larguras de banda nas portas de saída de switches OpenFlow. Além disso, a ferramenta permite ao desenvolvedor ou projetista de rede fazer uma migração suave de redes baseadas no MPLS para uma arquitetura de controle baseada no SDN/OpenFlow, incluindo todas as suas vantagens, mas sem a necessidade do desenvolvimento de uma inteligência ou engenharia de tráfego para a alocação de LSPs.

A BAMSDN foi desenvolvida na linguagem Python e pode ser reutilizada em qualquer ambiente Linux com Mininet, OpenVswitch, OpenFlow e controlador POX para diferentes cenários de rede. A versão que será demonstrada suporta os modelos MAM (*Maximum Allocation Model*) [Faucher and Lai 2005] e RDM (*Russian Dolls Model*) [Pinto Da Costa Neto et al. 2008], mas pode, de forma modular, incluir outros BAM ou módulos de engenharia de tráfego. Através do MAM e RDM será demonstrada a capacidade do controlador em fazer preempções de forma dinâmica, assim como, oferecer garantias de banda mínima para as classes de tráfego.

Na parte seguinte deste artigo, a Seção 2 descreve a ferramenta BAMSDN com seus componentes e operação. A Seção 3 trata da demonstração do uso da ferramenta. Por fim, a Seção 4 apresenta a conclusão.

2. A ferramenta BAMSDN

A ferramenta BAMSDN¹ visa oferecer suporte à implantação prática de BAMs a partir dos benefícios e características do paradigma SDN. A ferramenta auxilia o plano de controle e de gerenciamento da rede através da criação de LSPs virtuais e da reconfiguração

¹disponível para download em <https://github.com/EliseuTorres/BAMSDN>

das alocações de banda por classe de serviço (*BCs - Bandwidth Constraints*) do modelo BAM adotado. As próximas subseções detalham os componentes da ferramenta proposta, assim como a interação entre os processos executados pelos mesmos.

2.1. Visão geral do funcionamento

A ferramenta BAMSDN reúne dois elementos que se integram de forma vantajosa no intuito de permitir a alocação de banda com dinamismo e flexibilidade em redes MPLS. São eles: i) Um modelo de alocação de banda (BAM) que define a distribuição e compartilhamento da banda disponível para classes de aplicações; e ii) Uma estratégia SDN, onde o protocolo OpenFlow atua no controle da operação da rede e efetivação das políticas determinadas pelo BAM.

Conforme discutido em [Martins et al. 2015], BAMs definem uma estratégia geral de alocação de recurso que permite três ações principais: i) um planejamento e agrupamento das aplicações em classes (CTs - Classes de Tráfego) com requisitos comuns (i.e.: QoS, SLA ou outro critério/ requisito da aplicação); ii) uma definição da quantidade de banda alocável por classe (BC - *Bandwidth Constraint*); e iii) uma estratégia ou comportamento (*behavior*) para a alocação e compartilhamento da banda disponível por classe. Na ferramenta BAMSDN tais ações são realizadas através do BAM que interage com os usuários da rede MPLS da seguinte forma: (i) atuando como um procurador (*broker*) para a alocação de banda; (ii) recebendo solicitações de LSP (*request*) com sua trajetória (*path*); e (iii) alocando uma LSP solicitada, quando a mesma é aceita (*setup grant*) ou rejeitada (*setup deny*) pelo BAM em função da disponibilidade de banda nos enlaces de sua trajetória.

O BAM utilizado, portanto, gerencia e aloca a banda disponível por enlace. Assim sendo, o estabelecimento de uma LSP numa trajetória envolvendo “n” enlaces implica numa análise por parte do BAM sobre a disponibilidade de banda e necessidade, ou não, de preempção para todos os enlaces na trajetória da LSP. É preciso destacar que a operação do BAM já faz a gerência dos recursos para as classes de aplicações e utiliza uma informação de rota provida para a BAMSDN.

2.2. Características gerais e benefícios

Redes MPLS acrescentam ao IP um mecanismo de estabelecimento LSPs com o controle ou não das rotas (“*path*”) e recursos utilizados pelas LSPs (MPLS-TE e MPLS). A migração de uma rede MPLS para um plano de controle puramente SDN/OpenFlow requer, em princípio, um mecanismo inteligente capaz de realizar duas tarefas: i) definir a rota a ser alocada pelo OpenFlow nas tabelas de fluxos para a trajetória da LSP; e ii) alocar a banda necessária para os serviços associados à LSP.

Por tais motivos, na ferramenta BAMSDN, o BAM aloca e compartilha a banda disponível entre as LSPs e usa um roteamento padrão ou um roteamento explícito para a trajetória da LSP estabelecida. Assim sendo, a ferramenta BAMSDN apresenta os ganhos de permitir uma migração suave de uma rede MPLS ou MPLS-TE para uma solução SDN/OpenFlow e uma experimentação modular com a utilização de novos métodos de roteamento, engenharia de tráfego e alocação de banda.

2.3. Arquitetura e seus componentes

A arquitetura e os principais componentes da BAMSDN estão ilustrados na Figura 1. Como é possível observar, a ferramenta é composta de 3 módulos, a saber: (i) Núcleo do controlador, responsável pelo controle das solicitações por recursos de rede (banda); (ii) BAM, que faz a verificação dos recursos e aplica o modelo de alocação de banda disponível; (iii) Gerência de filas, responsável pela criação e remoção de filas nas portas de saída dos *switches* e também configurar e reconfigurar a largura de banda de cada fila.

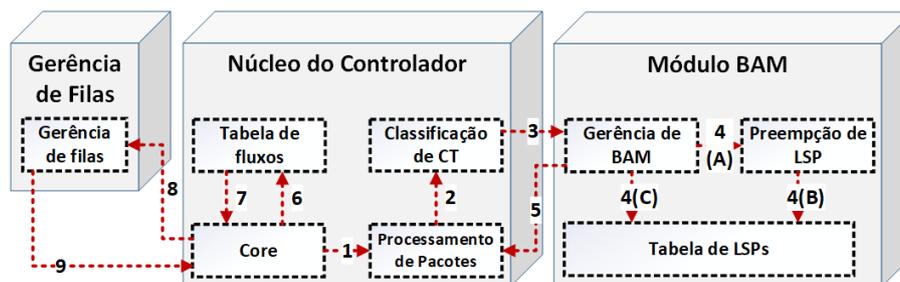


Figura 1. Visão geral dos módulos da BAMSDN.

2.4. Fluxo de Admissão

O fluxo de admissão implementado pela BAMSDN acontece a partir de uma mensagem de *PacketIn* enviada por um *Switch*. Neste fluxo são realizadas três principais ações, a saber:

- Controle de admissão: Nesta etapa, o pacote recém chegado ao core do controlador BAMSDN é reencaminhado para o componente de processamento de pacotes que extrai comparações (*matching*)² e encaminha para o componente de classificação de classes de tráfego que identifica a CT correspondente e gera uma instância de um LSP.
- Verificação de recursos: Esta etapa é iniciada pelo módulo BAM após a classificação de CT. A instância do LSP é encaminhada para o componente gerência de BAM responsável por verificar a disponibilidade de recursos em todos os enlaces do caminho em conformidade com o BAM definido para os mesmos. Após a verificação, caso existam recursos, o LSP é encaminhado para o componente Tabela de LSPs que adiciona o novo LSP na tabela de LSPs. Se existir necessidade de liberação de recursos, o componente Preempção de LSP é acionado para fazer a remoção dos recursos na tabela de LSPs.
- Execução: Nesta etapa, o componente core do módulo Núcleo do controlador recebe mensagens de controle do componente Gerência de BAMs que são processadas conforme segue: (i) Caso a mensagem de controle seja bloqueio (*None*), a tabela de fluxos é atualizada com uma mensagem de *FlowMod* para bloquear o pacote; (ii) Caso haja necessidade de liberação de recursos, uma mensagem *FlowMod* é enviada para atualizar a tabela de fluxos e remover as regras referente a liberação de recursos. O componente Gerência de filas também é acessado para remover ou reconfigurar as filas das portas de saída dos *switches*; (iii) caso o LSP seja aceito, uma mensagem de controle *FlowMod* é enviada para a tabela de

²IP de origem, IP de destino, porta de origem e porta de destino entre outros

fluxo com informações de configuração das rotas do novo pacote e o componente Gerência de filas é acessado para criar uma nova fila com a BC correspondente ou reconfigurar uma fila já existente.

2.5. Utilização da ferramenta

A ferramenta BAMSDN é utilizada a partir do ambiente de emulação Mininet que foi desenvolvido para ser executado apenas na plataforma Linux. Usuários de outras plataformas (e.g., Windows, MacOS X) poderão instalar o ambiente através de uma máquina virtual. Esta seção apresenta detalhes da utilização da ferramenta, tais como: (i) Aquisição, configuração e softwares adicionais necessários para rodar os experimentos; (ii) Execução dos Experimentos; (iii) Reconfiguração dos parâmetros de experimentação (largura de banda) dos enlaces e das classes de tráfego a partir do código fonte.

2.5.1. Aquisição da ferramenta e requisitos do ambiente

Para a execução da ferramenta no Mininet é necessário instalar o controlador POX e o gerador de tráfego iperf3. Também será necessário instalar o gerenciador de versão (git) para fazer o *download* da ferramenta. A instalação do (git) deverá ser feita através do gerenciador de pacotes da plataforma Linux em uso.

A ferramenta está disponível na URL <https://github.com/EliseuTorres/BAMSDN>. Para obtê-la, o usuário deve executar o comando “*git clone https://github.com/EliseuTorres/BAMSDN*” no terminal Linux. Após o *download* é necessário configurar o ambiente de experimentação. Para isso, o usuário deverá acessar a pasta do BAMSDN e executar o comando “*sudo ./init.sh*” que moverá os *scripts mam.sh* e *rdm.sh* para o diretório */pox* e as pastas *MAM* e *RDM* para o diretório *pox/ext*. Além disso, tais *scripts* criarão o diretório topologia na pasta *home* do usuário e moverão *lab.py* e demais *scripts* para a pasta “*/topologia*”.

2.5.2. Execução dos Experimentos

Para executar os experimentos será necessário que o usuário faça o login no ambiente utilizando o *shell* (SSH) a partir de dois terminais. Um dos terminais será utilizado para rodar a topologia e o outro para rodar o controlador. Para executar o controlador o usuário deverá acessar o diretório *pox/* e executar “*sudo ./mam.sh* ou *sudo ./rdm.sh*”. Com o outro terminal deverá acessar o diretório *topologia/* e executar “*topo.py*” para subir a topologia no Mininet.

Após a execução da topologia, o ambiente de experimentação será inicializado e disponibilizará uma CLI do Mininet. Para dar continuidade aos testes, na CLI do Mininet é necessário abrir o *xterm* para cada *host* em execução. Para isso, digite “*xterm h1 h2 h3 h6*”, em seguida com o *host* (h6) executar os *scripts* do servidor contidos na pasta *scripts/* e por fim execute o iperf3 para conectar os *hosts* (h1, h2 e h3) ao servidor utilizando o comando “*iperf3 -c 10.0.0.6 -p porta_de_destino*”. As portas de destino variam de 5001 a 5200 e são divididas em 3 subconjuntos de portas, cada subconjunto representando uma classe de tráfego (CT). Após a execução do iperf será possível visualizar tanto no cliente

quanto no servidor as informações de transferência de dados com a largura de banda para cada classe de tráfego.

2.5.3. Reconfiguração dos parâmetros de experimentação

A implementação de modelos de alocação de banda em um controlador SDN não é algo trivial. Por esta razão, esta ferramenta utiliza uma topologia fixa (Figura 2) e os parâmetros de reconfiguração da ferramenta estão limitados apenas à largura de banda para cada classe de tráfego e às portas TCP utilizadas para mapear as classes de tráfego. Para fazer a reconfiguração da largura de banda acesse o código fonte do arquivo `controller.py` e altere a variável `TOTAL_BANDWIDTH` para modificar a largura de banda máxima do enlace. A variável `queue_bw` altera os valores das BCs para a largura de banda de cada fila de uma classe de tráfego CT. Também será necessário acessar o código fonte `lsp.py` e fazer as mesmas alterações na variável `TOTAL_BANDWIDTH` e em `_BC0`, `_BC1` e `_BC2`.

3. Demonstração da Ferramenta

A ferramenta BAMSDN será demonstrada através de emulação no ambiente Mininet. O processo consistirá em demonstrar a capacidade do BAMSDN de fazer preempções, bloqueios e alocação de recursos através do mapeamento dos fluxos para LSPs. As próximas subseções descrevem a metodologia que será adotada e o ambiente de demonstração.

3.1. Metodologia

Dois experimentos foram projetados para a demonstração da BAMSDN. No primeiro experimento, será testado o comportamento do modelo MAM. Para isso, serão utilizadas três classes de tráfego distintas (i.e., CT0, CT1 e CT2) que representam aplicações de baixa, média e alta prioridade. Cada classe de tráfego terá sua restrição de banda BC (i.e., BC0, BC1 e BC2) que determinará o limite de largura de banda máximo que uma determinada classe deve utilizar em um enlace. As aplicações de média e alta prioridade não utilizarão todo o recurso de sua restrição de banda e as aplicações de baixa prioridade extrapolarão o seu limite de restrição de banda, fazendo com o que as novas solicitações de recursos sejam bloqueadas. No modelo MAM, os recursos de cada classe de tráfego são privados, não permitindo que sejam compartilhados com outras classes em um período de ociosidade.

Para o segundo experimento, adotamos a metodologia aplicada na primeira experimentação, porém, com será utilizado o modelo de alocação de banda RDM. Na fase inicial do experimento, será mantido o mesmo comportamento das aplicações de alta, média e baixa prioridade. Como o modelo RDM permite que os recursos das classes mais prioritárias sejam emprestados para as classes menos prioritárias, após exceder o limite de restrição de banda, as classes menos prioritárias requisitarão os recursos das classes mais prioritárias. No decorrer do experimento, ao executar as aplicações das classes de prioridade mais altas, estas requisitarão que as aplicações classes de prioridade mais baixas façam a devolução dos recursos emprestados por meio de preempções.

3.2. Ambiente de Demonstração

Na demonstração, os experimentos seguirão uma abordagem de emulação que reproduzirá uma rede OpenFlow através do Mininet. Sendo assim, serão utilizados: (i) Sistema Operacional *Ubuntu Server 15.04, x64*, versão do *kernel 3.19.0-30*; (ii) Máquina virtual *Virtual Box* versão 5.1.34; (iii) *Mininet*³, versão 1.8r11; (iv) Protocolo *OpenFlow*⁴, versão 1.0; (v) Controlador *OpenFlow POX*⁵, versão 0.2.0; e (vi) Gerador de tráfego *iPerf3*⁶, versão 3.0.7. Além de atender os requisitos da experimentação, a escolha de tais ferramentas se deu pelas *features* disponíveis, por ser software livre e existir uma ampla documentação para download.

O cenário da rede emulada na demonstração está ilustrado na Figura 2. O mesmo é constituído de três *hosts* que atuam como origem de tráfego (H_{S1} , H_{S2} e H_{S3}). Além disso, o ambiente possui um *host* que atua como destino (H_{D1}), um controlador OpenFlow que configura as regras de fluxos de acordo com as instruções do BAMSDN e, por fim, três *switches* que fazem a conexão entre os hospedeiros de origem e destino.

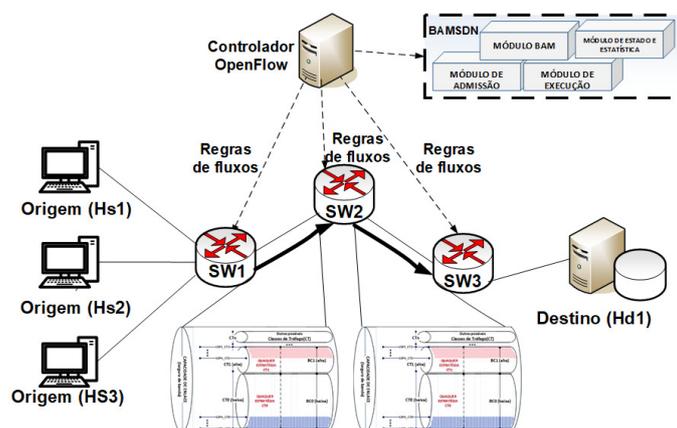


Figura 2. Cenário de rede OpenFlow utilizada nos experimentos de demonstração.

Dado que os experimentos foram planejados para avaliar a capacidade do BAMSDN em implantar dinamicamente a alocação de banda utilizando BAMs numa rede OpenFlow, assim como, verificar a capacidade da ferramenta de promover a reconfiguração das restrições de banda, foram definidos como fatores de análise a banda utilizada pelas LSPs, o número de preempções e bloqueios realizados pelo arcabouço. Assim, os parâmetros globais das emulações no cenário utilizado na demonstração serão configurados conforme a seguir: (i) enlaces com capacidades de 500 Mbps; três classes de tráfego: CT0, CT1 e CT2, com larguras de banda máxima de 5 Mbps, 10 Mbps e 20 Mbps, respectivamente; e, por fim, restrições de largura de banda (*Bandwidth Constraint – BC*), de acordo com a Tabela 1.

A partir dos fatores utilizados na tabela serão demonstradas as seguintes métricas de desempenho: (i) Taxa de utilização que consiste na largura de banda utilizada em

³<http://mininet.org/>

⁴<http://archive.openflow.org/>

⁵<https://github.com/noxrepo/pox>

⁶<https://iperf.fr/>

Tabela 1. Restrições de Banda (BC) por Classes de Tráfego (CT) que serão adotadas na demonstração.

	MAM			RDM		
	Max (%)	Max (Mbps)	CTs	Max (%)	Max (Mbps)	CTs
BC0	50	250	CT0	100	500	CT0 + CT1 + CT2
BC1	30	150	CT1	50	250	CT1 + CT2
BC2	20	100	CT2	20	100	CT2

relação à capacidade do enlace; (ii) Taxa de Preempções que fornece a percentual de preempções que ocorreram durante a emulação; (iii) Taxa de Bloqueios que indica o percentual de bloqueios ocorridos; e (iv) Requisições de LSPs que indica o número de solicitações de criação de novas LSPs.

4. Conclusão

Tendo em vista a redução dos custos de CAPEX e OPEX de redes MPLS, provedores de serviços têm vislumbrado o paradigma SDN como uma alternativa para implementação de soluções de baixo custo de mecanismos de estabelecimento de circuitos (LSPs) com garantias mínimas de QoS para diferentes classes de tráfego. Contudo, a migração de uma rede MPLS para um plano de controle puramente SDN/Openflow requer, em princípio, o desenvolvimento de uma certa “inteligência” que seja capaz de definir a rota a ser alocada pelo controlador nas tabelas de fluxos para a trajetória da LSP e alocar a banda necessária para os serviços associados à LSP. Por tais motivos, este trabalho apresenta a ferramenta BAMSDN. Uma solução de baixo custo que integra BAMs ao paradigma SDN implementados através do OpenFlow. A partir da visão global do controlador a BAMSDN é capaz de alocar de forma dinâmica e flexível os recursos limitados de banda disponíveis para as classes de tráfego de acordo com o BAM utilizado. A modularidade da ferramenta também permite que novos BAMs e componentes de engenharia de tráfego possam ser utilizados em diferentes cenários de redes.

Referências

- Faucher, F. L. and Lai, W. (2005). Maximum Allocation Bandwidth Constraints Model for DiffServ-aware MPLS Traffic Engineering. Request for Comments RFC 4125, Internet Engineering Task Force - IETF.
- Kreutz, D., Ramos, F. M. V., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76.
- Martins, J., Martins da Silva Bezerra, R., Durães, G., and Reale, R. (2015). Uma Visão Tutorial dos Modelos de Alocação de Banda (BAM – Bandwidth Allocation Models) como Mecanismo de Provisionamento de Recursos em Redes IP/MPLS/DS-TE. *RSC*, 5(2):144–155.
- Pinto Da Costa Neto, W., De Figueiredo Brito, S., and Martins, J. (2008). Algoritmos de Seleção de Caminho e Gerenciamento de Banda Compartilhada conforme ao Modelo RDM para Classes de Tráfego em Rede DS-TE. In *Anais do 26 Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2008*, pages 537–552, Rio de Janeiro, Brazil. SBC - Brazilian Computer Society.