

UNIVERSIDADE FEDERAL DO AMAZONAS
PRO REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE APOIO À PESQUISA
PROGRAMA INSTITUCIONAL DE INICIAÇÃO CIENTÍFICA

AUTO-CONFIGURAÇÃO DE PARÂMETROS
DE ROTEAMENTO EM REDES COM AGENTNOX

Bolsista: Ricardo Bennesby da Silva, Fapeam

MANAUS
2011

UNIVERSIDADE FEDERAL DO AMAZONAS
PRO REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE APOIO À PESQUISA
PROGRAMA INSTITUCIONAL DE INICIAÇÃO CIENTÍFICA

RELATÓRIO PARCIAL
PIB - E - 0007/2010
AUTO-CONFIGURAÇÃO DE PARÂMETROS DE
ROTEAMENTO EM REDES COM AGENTNOX

Bolsista: Ricardo Bennesby da Silva, Fapeam
Orientador: Profº Drº Edjard Souza Mota

MANAUS
2011

Todos os direitos deste relatório são reservados à Universidade Federal do Amazonas, ao Núcleo de Estudo e Pesquisa em Ciência da Informação e aos seus autores. Parte deste relatório só poderá ser reproduzida para fins acadêmicos ou científicos.

Esta pesquisa é financiada pela Fundação de Amparo à Pesquisa -Fapeam, através do Programa Institucional de Bolsas de Iniciação Científica da Universidade Federal do Amazonas

RESUMO DO RELATÓRIO

Neste trabalho apresentamos uma proposta de solução para o problema de roteamento entre domínios ou Sistemas Autônomos formados por redes baseadas no protocolo Openflow e gerenciadas por um sistema operacional de redes.

O objetivo deste trabalho é implementar um componente que realize o roteamento entre diferentes domínios. Ao longo do relatório apresentamos ilustrações que esclarecem como o componente foi implementado.

Apresentamos uma introdução a respeito do tema para contextualização do leitor, uma revisão bibliográfica expondo o estado da arte da literatura a respeito do assunto abordado, as metodologias adotadas para o desenvolvimento das atividades, resultados obtidos, discussões e conclusões, onde mostramos nossa opinião acerca da contribuição das pesquisas e do que foi desenvolvido.

Realizamos testes com três sistemas autônomos e foi possível determinar a rota para pacotes que não eram direcionados à rede da máquina de origem. Apesar de limitações de tempo e recursos computacionais para realização de testes mais completos, os resultados foram satisfatórios.

LISTA DE FIGURAS E TABELA

Figura 1- Topologia de rede com switches e roteadores	12
Figura 2- Mecanismo de roteamento para fluxos pacotes que chegam ao controlador	12
Figura 3- Roteamento no NOX com BGP.....	14
Tabela 1- Estrutura da tabela de roteamento.....	13

SUMÁRIO

INTRODUÇÃO	7
REVISÃO BIBLIOGRÁFICA	8
MÉTODOS UTILIZADOS	10
RESULTADOS E DISCUSSÕES	11
CONCLUSÕES	16
FONTES E REFERÊNCIAS BIBLIOGRÁFICAS	17
CRONOGRAMA	18

INTRODUÇÃO

Para facilitar a gerência das complexas redes de computadores, oferecendo uma abstração para suas aplicações, existe o NOX [1], um sistema operacional de redes. Por meio dele as aplicações de gerenciamento observam as entidades da rede sem utilizar parâmetros de baixo nível, como endereços IP e portas dos switches aos quais os usuários estão conectados. Para controlar e observar a rede o NOX utiliza um protocolo chamado Openflow [2], que permite acesso e controle dos switches e roteadores da rede, bem como aos fluxos de pacotes enviados e recebidos.

A Internet utiliza protocolos de roteamento para que seja possível a troca de pacotes entre os inúmeros sistemas finais conectados à rede. O roteamento consiste em determinar rotas pelas quais os pacotes de dados devem seguir da fonte ao destino.

A Internet é dividida em grupos de redes que seguem mesma política, chamados Sistemas Autônomos (SA). Milhares de sistemas autônomos concorrentes devem cooperar entre si para manter a conectividade global da Internet. Cada Sistema Autônomo (SA) codifica várias decisões de negócios, econômicas e de desempenho dentro da política de roteamento [4]. O sistema corrente de roteamento inter-domínio, o BGP (*Border Gateway Protocol*- Protocolo de Roteamento De Borda), é um protocolo que permite a transmissão de informações de alcançabilidade na rede e que cada AS expresse sua política usando um classificador ponderado que determina como um roteador no AS pode escolher uma dentre diferentes rotas possíveis para um destino e filtros que determinam rotas permitidas e rotas que ficam escondidas de cada AS vizinho. Como a Internet é composta de diferentes domínios, o sistema de roteamento deve prover autonomia, permitindo que os operadores de redes configurem esses rankings de maneira independente e sem nenhuma restrição nos filtros permitidos.

O NOX atualmente utiliza algoritmos com roteamento intra-domínio, por se tratar de uma solução inicialmente para redes locais. Mas para que o NOX possa ser utilizado na Internet atual é necessário que ele possua roteamento entre Sistemas Autônomos. O objetivo deste trabalho é implementar o algoritmo de roteamento inter-domínio (baseado no BGP) no NOX, e assim, configurar os parâmetros desse algoritmo de maneira autônoma posteriormente através da arquitetura AgentNOX [5].

REVISÃO BIBLIOGRÁFICA

Em [1] é apresentado o sistema operacional de redes NOX. É proposta uma solução para o problema de gerenciamento dos elementos físicos da rede, de modo análogo a um sistema operacional convencional, que gerencia o acesso aos recursos físicos de um computador. Outra semelhança está na importante função de prover uma abstração, com o intuito de o administrador não mais necessitar saber informações de baixo nível como endereços MAC ou IP para executar funções ou corrigir problemas na rede, mas apenas informações de alto-nível como o nome da máquina ou do usuário. Essa abstração é denominada interface programática. O controle e o gerenciamento da rede é realizado por aplicações que a gerenciam e que são executadas na camada provida por essa interface programática. O NOX é escrito em C++ e atualmente apresenta mais de 32.000 linhas de código. Em [8] há ferramentas que permitem fazer benchmarks em switches Openflow e as informações obtidas para a atual versão do NOX é que processa 60000 respostas por segundo.

Um trabalho intimamente relacionado com o do NOX é o do protocolo Openflow. Em [2] é proposta uma maneira de realizar experimentos de pequena ou larga escala nas redes de uso diário por meio do Openflow, que é baseado no Switch Ethernet. Este trabalho descreve esse switch, que conecta-se a um controlador (NOX) por um canal seguro (*Secure Channel*), apresenta uma tabela de fluxos (*Flow Table*) que associa uma ação para cada novo fluxo de entrada e um protocolo que permite que haja uma maneira padrão de comunicação entre o switch e o controlador. O Openflow permite que os pesquisadores possam executar seus experimentos em switches ou roteadores heterogêneos, propondo uma uniformização e eliminando a necessidade de os projetistas ou fabricantes terem de expor a arquitetura interna de seus produtos.

A respeito do protocolo de roteamento BGP há muitos livros, artigos e outros documentos relacionados, uma vez que este é o protocolo utilizado para roteamento inter-domínio na Internet. Uma obra muito conhecida é [4] que trata a respeito das arquiteturas, dos protocolos e da organização de redes TCP/IP (a Internet é uma rede TCP/IP). Utilizei os capítulos 8, 14 e 15 para entender os princípios do BGP.

Um outro trabalho sobre o roteamento inter-domínio é o [3], que também possui um vasto conhecimento na área de rede de computadores e aborda a organização e a sua divisão em 5 camadas. Para este trabalho utilizei o capítulo 4 que trata das camadas de rede, que explica dentre outros, o protocolo BGP.

A referência [6] é um trabalho específico sobre o protocolo BGP. Mostra detalhes avançados do BGP, desempenho, controle de políticas, escalabilidade, uso em redes

empresariais, dentre outros.

Mas apesar do uso em larga escala do protocolo BGP, alguns problemas ainda precisam ser resolvidos. Os desafios envolvidos na melhoria desse protocolo são abordadas em [11]. Nele os autores mostram os avanços das pesquisas na área e quais aspectos do protocolo recebem mais atenção. Eles mostram suas perspectivas do por que alguns problemas não têm solução até hoje e algumas soluções propostas não foram adotadas.

Todos os protocolos da Internet são padronizados e mantidos por uma instituição conhecida como *Internet Engineering Task Force* (IETF). A cada novo protocolo ou nova versão de protocolo é emitido um documento na Internet chamado RFC (*Request For Comments*). O RFC do BGP é o [7]. Ele especifica detalhes do protocolo, como tipos de mensagem, formato das mensagens, tamanho, dentre outras informações importantes para quem quiser informações que para quem quiser implementar o protocolo é obrigado a seguir para que ele possa ser usado na Internet. Caso contrário ele simplesmente não funcionará.

MÉTODOS UTILIZADOS

Para realizar este trabalho foram utilizadas algumas ferramentas e métodos que temos à disposição. Inicialmente começamos pesquisando o estado da arte do problema, por meio de leitura de livros e de artigos na área. Dessa maneira, foi possível analisar as soluções já existentes, os pontos positivos e negativos da abordagem atual e saber como conduzir o trabalho de modo a cumprir os objetivos.

Primeiro estudamos os aspectos da linguagem de programação C++ por meio de livros e tutoriais. Concomitantemente estudamos o estado da arte do roteamento na Internet e do roteamento no NOX. Começamos então a estudar como implementar componentes em C++ para o NOX, construindo o arcabouço de ferramentas a serem utilizadas para o desenvolvimento da meta principal, por meio da implementação de um componente nessa linguagem de programação.

Para desenvolvimento, utilizamos uma máquina virtual Openflow e um ambiente que emula uma rede em um PC, o Mininet, ambos providos pela Universidade Stanford (Estados Unidos) e o software de emulação VmWare. A máquina utilizada possui Sistema Operacional Windows 7, emulando a máquina virtual executando no Ubuntu por meio do VmWare e o ambiente de desenvolvimento é o Eclipse, com um plugin para C++. A configuração básica é um processador AMD Turion II Dual-Core (2 núcleos), 3 GB de RAM e 300 GB de HD.

Existem soluções que implementam o BGP, como [9] e [10], por exemplo, mas percebemos que seria necessário adaptar o código para que pudesse ser executado como um componente do NOX, pois haveriam muitas inconsistências em tentar portá-lo diretamente e que seria mais vantajoso implementar o BGP do início desenvolvendo-o já como um componente, utilizando a API do NOX. Desse modo torna-se possível utilizar o NOX para gerenciar de modo mais fácil redes que utilizam esse protocolo, para determinar o melhor caminho para que uma rede seja alcançada por outra.

Foram estudados também as estruturas de dados essenciais aos componentes do NOX, seus eventos e como definir ações aos switches, por meio da análise de outros componentes implementados, e do uso da listas de discussão do NOX, do Openflow e do Mininet, onde é possível entrar em contato com os criadores dessas ferramentas ou com outras pessoas que saibam esclarecer as dúvidas que eventualmente surgem.

RESULTADOS E DISCUSSÕES

Uma abordagem para resolver problemas em gerenciamento de redes é o Sistema Operacional para Redes NOX, que analisa cada fluxo de pacotes da rede que ele controla por meio de aplicações executadas em sua interface programática e toma determinadas ações. Assim, o administrador pode criar e ter aplicações que auxiliem a controlar grandes redes. A Internet, maior rede de computadores do mundo, poderia ser gerenciada utilizando o Sistema Operacional de Redes.

Inicialmente este trabalho seria auto-configuração de parâmetros de roteamento em redes com AgentNOX [5]. O AgentNOX utiliza a arquitetura NOX executando como aplicações agentes, que realizam o monitoramento da rede de maneira autônoma, fazendo com que o trabalho do administrador seja somente de definir políticas.

Devido à complexidade do problema e de alguns passos que precisam ser tomados antes, decidiu-se reduzir o escopo do trabalho. Para esse trabalho de PIBIC implementamos um componente de roteamento baseado no BGP, que poderá ser a base para qualquer configuração autônoma de roteamento entre diferentes domínios.

A Internet é formada por grupos de redes. Cada grupo de redes está sob uma autoridade administrativa que obedece às mesmas políticas, independente dos algoritmos de roteamento que são executados em seus roteadores. Esses grupos são chamados de Sistemas Autônomos (SA). Cada SA possui protocolos de roteamento interno e externo [4]. Analisando o NOX, descobrimos que ele possui apenas um algoritmo de roteamento intra-domínio, o OSPF (Open Shortest Path First- Caminho Mais Curto Primeiro Aberto). Mas a Internet possui um protocolo de roteamento entre domínios largamente utilizado, chamado BGP (Border Gateway Protocol- Protocolo de Roteamento de Borda), atualmente em sua versão 4 [4]. A figura 1 mostra a estrutura de uma rede com máquinas ligadas a switches e roteadores:

Mas por que o NOX não possui um protocolo de roteamento inter-domínios? A resposta é que o NOX foi desenvolvido inicialmente para redes empresariais. Mas pelo que já foi discutido anteriormente, sabendo das vantagens que o Sistema Operacional de Redes pode trazer e a série de dificuldades e prejuízos que ainda há devido à complexidade para gerenciamento humano, propomos desenvolver um componente que implemente o roteamento inter-domínios, baseado no BGP.

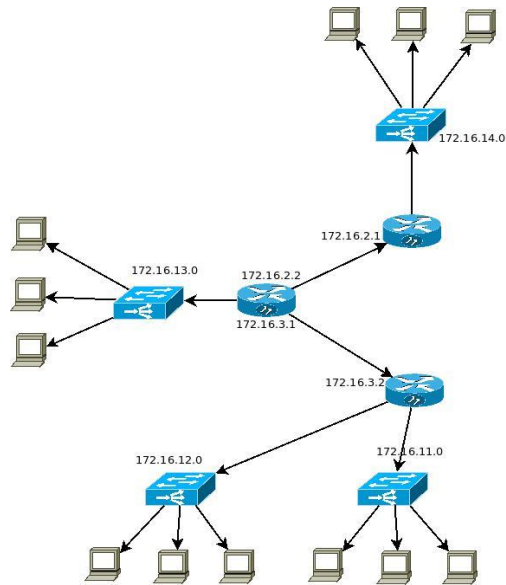


Figura 1- Topologia de Rede com Switches e Roteadores

O NOX permite que sejam criados componentes nas linguagens Python e C++. Para fins de obter um desempenho melhor o componente de Roteamento foi desenvolvido em C++.

A ideia foi criar um componente que deve ser uma dependência do componente switch, que realiza o roteamento dentro de uma rede, como mostra a Figura 2. O componente switch foi alterado para suportar a funcionalidade de roteamento inter-domínio.

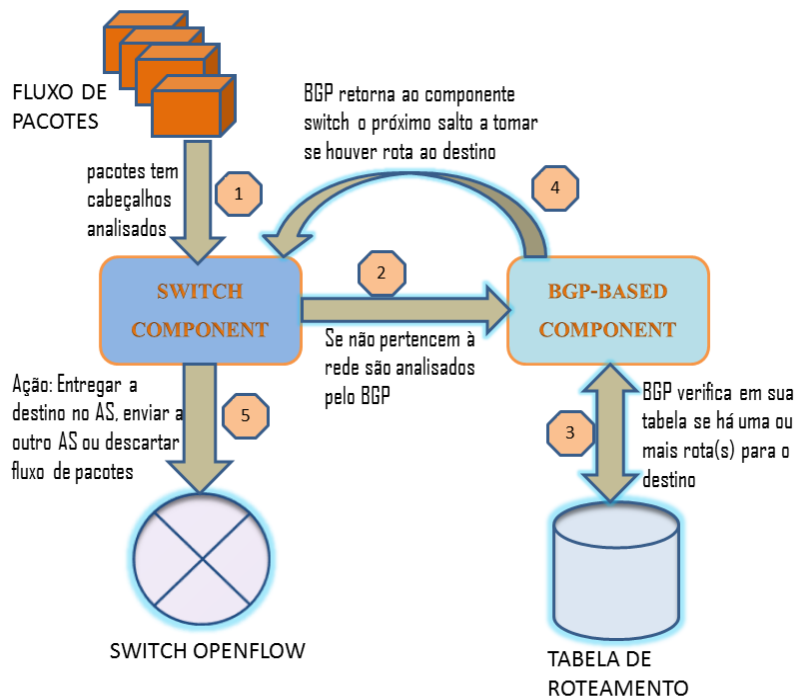


Figura 2- Mecanismo de roteamento para pacotes que chegam ao controlador

Desse modo, a cada evento de chegada de pacotes, o componente switch verifica pelo IP se os pacotes pertencem ou não à sua rede, para determinar o caminho para chegar ao destino. No componente tradicional, quando essa condição não é satisfeita, nenhuma ação é configurada, a entrada de fluxo não é adicionada na tabela do switch e os pacotes são descartados. Mas no componente switch modificado se os pacotes não pertencem à sua rede, o componente BGP é chamado, e sua tabela é consultada. Nessa tabela há informações de por quais sistemas autônomos passar para alcançar um determinado IP. Desse modo, havendo uma ou mais entradas na tabela, uma rota é escolhida e esta é repassada ao componente switch. Se houver uma rota possível, o controlador define a ação de repassar o fluxo de pacotes ao SA definido em sua tabela (próximo salto). Se não houver rota, os pacotes são descartados.

Começamos a definir as estruturas de dados da classe do componente BGP. Um dos atributos mais importantes é a tabela de roteamento, que é consultada quando o IP de destino do fluxo de pacotes que chega ao controlador não pertence a nenhuma máquina na rede. A tabela de roteamento possui o IP de destino e o caminho (identificação dos SAs) para se chegar a esse destino. A tabela 1 mostra duas colunas: A primeira é o IP da máquina destino do fluxo de pacotes. A segunda coluna (*AS_PATH*) refere-se a quais SAs os pacotes devem ser enviados para chegarem ao destino.

IP de destino	AS_PATH
192.168.1.0	1
192.200.2.0	1,3
200.163.1.0	1,5,6,7

Tabela 1- Estrutura da tabela de roteamento

A estrutura de dados utilizada para implementar a tabela foi um dicionário, chamado *multimap*. Ele permite inserir pares de valores de dois tipos definidos e fazer busca com um dos valores como chave. Em vez de utilizar o *map* resolvemos optar pelo *multimap* porque o *map* não permite inserir chaves repetidas, enquanto o *multimap* permite. Isso é importante para o nosso componente, pois pode haver chaves repetidas, ou seja, para um dado IP pode haver várias rotas diferentes, várias entradas na tabela com mesma chave. Essa é a importância do atributo *AS_PATH* referido anteriormente. Muitas vezes um destino pode ser alcançado passando por diferentes SAs. Quando isso ocorre o componente tem que decidir qual a rota

escolhida, ou seja, qual das entradas da tabela será utilizada para encaminhar o fluxo de pacotes ao seu destino. O BGP padrão utiliza vários critérios para decidir qual rota será adotada. Mas neste trabalho, definimos que a rota escolhida será sempre a menor, ou seja, a que passa pela menor quantidade de SAs para chegar ao destino. Se houver rotas de mesmo tamanho, a primeira de menor tamanho será escolhida.

A figura 3 mostra uma topologia com dois SAs (SA 1 e SA 2), com um controlador NOX gerenciando a rede, comunicando-se, trocando informações de alcançabilidade de rotas.

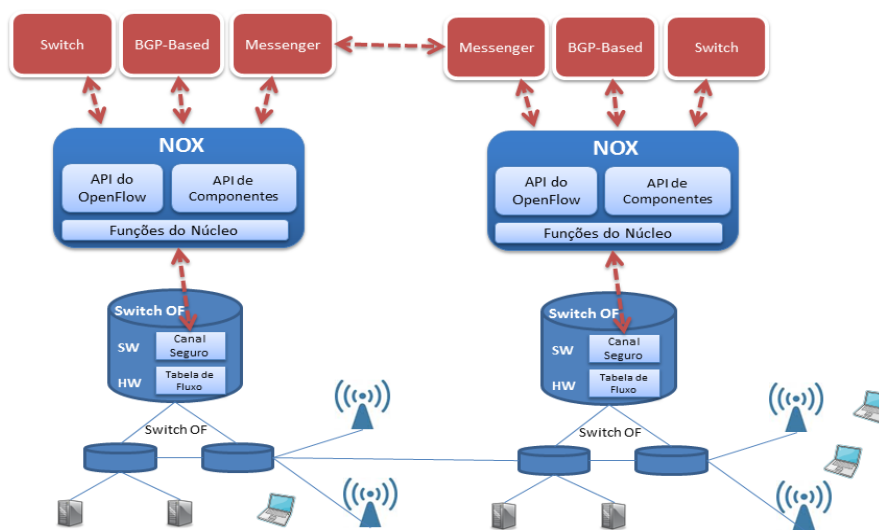


Figura 3- Roteamento no NOX com BGP

Para que a troca de informações entre dois NOX fosse possível fez-se necessário o uso de um componente que permite abrir uma conexão entre os dois controladores comunicantes. Esse componente é o *messenger*. Assim, quando o componente switch é executado, uma das primeiras ações tomadas pelo controlador é abrir uma conexão na porta 2603 com um controlador NOX vizinho e enviar uma mensagem solicitando a tabela de roteamento deste. Sem o *messenger* isso não seria possível, pois o NOX só abre uma conexão segura com os switches da rede que ele administra e uma conexão na porta 2603. Mas essa porta é exclusiva do *messenger*. Desse modo, a comunicação é segura.

Após essa fase começamos a trabalhar no repasse do fluxo de pacotes. Para que isso fosse possível foi necessário mudar o conjunto de ações associadas a cada novo fluxo de dados. Assim, se um determinado IP não pertence à rede, o componente BGP verifica seu destino, e o dado o número IP do próximo salto, ações são definidas

para mudar o destino dos pacotes. Assim, estes são entregues ao próximo salto, que também deve estar executando o BGP para que os pacotes recebam o mesmo tratamento.

Realizamos testes com três SAs e verificamos que foi possível rotear e repassar os pacotes ao destino como desejado. Devido às limitações da máquina não utilizamos um número maior de SAs. No entanto, em breve utilizaremos um computador dedicado para testes que envolvem muito processamento e espaço de armazenamento, para realizarmos testes com um número considerável de SAs.

CONCLUSÕES

Cumprindo com o que foi proposto como objetivo, implementamos um componente de roteamento inter-SA, capaz de fornecer ou receber rotas a outros SA ou determinar o caminho que um fluxo de pacotes deve seguir para chegar ao destino.

O algoritmo do componente desenvolvido é básico. Apesar de realizar sua função principal ainda é muito simples em relação ao protocolo de roteamento BGP, utilizado atualmente na Internet.

O BGP tradicional é composto de inúmeros atributos que são utilizados na decidibilidade da rota a ser tomada pelos pacotes quando há mais de um caminho possível. No componente implementado o único critério de escolha é o tamanho da entrada na tabela de roteamento. Sempre o menor ou um dos menores (caso estes tenham o mesmo tamanho) é (são) escolhido(s).

Contudo, acreditamos que a contribuição tem sido grande, pois o NOX é uma inovação, uma maneira de resolver ou minimizar problemas de gerenciamento de redes. E como o NOX ainda não apresenta nenhum componente que realiza roteamento entre SAs, este trabalho tem grande valor. A seguir atualizaremos e melhoraremos alguns aspectos do componente, realizaremos testes com um número maior de SAs para avaliarmos questões como desempenho, comportamento do algoritmo e para corrigir erros.

FONTES E REFERÊNCIAS BIBLIOGRÁFICAS

[1] N.Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, "NOX: Toward an operating system for networks". ACM SIGCOMM Computer Communications Review: 2008.

[2] N. Mckeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "Openflow: Enabling Innovation in Campus Network". ACM SIGCOM Computer Communications: 2008.

[3] J. Kurose, K. Ross. "Redes de Computadores e a Internet- Uma abordagem Top-Down". 5. Ed.São Paulo: Pearson, 2010. 615 p.

[4] D. Comer. Internetworking with TCP/IP- Principles, Protocols and Architectures. 4. Ed. New Jersey: Prentice Hall, 2000. 750 p.

[5] A. Passito, E. Mota, R. Braga. "Towards an Agent-based NOX/Openflow Platform for the Internet". Workshop de Pesquisa Experimental na Internet do Futuro, 2010, Gramado, Brasil.

[6] R. Zhang, M. Bartell . "BGP Design and Implementation". 1. Ed. Indianapolis: Cisco Press, 2004, 672 p.

[7] T.J.Watson Reasearch Center. Request For Comments- RFC 1771- A Border Gateway Protocol 4 (BGP-4), 1995.

[8] Switch Openflow Consortium. Disponível em: www.Openflowswitch.org/wk/index.php/Oflops

[9] E. Kohler, R. Morris, B. Chen, J. Jannotti, M. Kaashoek. "The Click modular router". ACM Transactions on Computer Systems (TOCS), 18 v.:2000.

[10] Zebra- Implementação de Protocolos como BGP em: <http://www.zebra.org/>

[11] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure. "Open Issues in Interdomain Routing: A Survey". IEEE Network, November-December 2005.

CRONOGRAMA EXECUTADO

Nº	Descrição	Ago 2010	Set	Out	No v	Dez	Jan 2011	Fev	Mar	Abr	Mai	Jun	Jul
1	Realizar um estudo sobre os parâmetros que influenciam as políticas de roteamento nos domínios da Internet.	X	X	X	X	X	X						
2	Realizar um estudo sobre como o NOX pode lidar com a configuração de rotas, obedecendo as políticas de roteamento.				X	X	X	X	X				
3	Realizar um estudo sobre a linguagem C++ e componentes do NOX implementados em C++					X	X						
4	Preparação de arcabouço com ferramentas necessárias para implementação do componente BGP					X	X	X	X				
5	Implementação do Componente BGP							X	X	X	X	X	
6	Testar o funcionamento do componente e fazer medições de desempenho com métricas sugeridas na literatura pesquisada.									X	X	X	X
7	- Elaboração do Resumo e Relatório Final (atividade obrigatória) - Preparação da Apresentação Final para o Congresso (atividade obrigatória)												X