

Re-roteamento Dinâmico de Túneis LSPs na Ferramenta GLASS: Uma Proposta de Implementação

Klenilmar L. Dias^{1,2,3}, Antônio Jorge G. Abelém^{1,2}, Klissiomara L. Dias³, Kelvin L. Dias^{1,4}

¹Grupo de Estudos em Redes de Computadores e Comunicação Multimídia (GERCOM)
Universidade Federal do Pará (UFPA) – Belém, PA – Brasil

²Programa de Pós-Graduação em Engenharia Elétrica – Universidade Federal do Pará
(UFPA) – Belém, PA – Brasil

³Faculdade SEAMA – Macapá, AP – Brasil

⁴Departamento de Engenharia Elétrica e Computação – Universidade Federal do Pará
(UFPA) – Belém, PA – Brasil

{abelem,kld}@ufpa.br, klenilmar@gmail.com, kdias@seama.edu.br

Resumo. *Redes Ópticas de Nova Geração necessitam da utilização de características de QoS (Quality of Service) e Engenharia de Tráfego. O GLASS (GMPLS Lightwave Agile Switching Simulator) é uma ferramenta de simulação para Redes Ópticas de Nova Geração, livre e de código aberto, desenvolvida em Java. Este artigo visa apresentar uma nova funcionalidade no simulador para facilitar o re-roteamento dinâmico de LSPs (Label Switched Paths) em cenários de redes GMPLS, baseado no protocolo de sinalização RSVP-TE com o objetivo de auxiliar trabalhos de pesquisas, projeto e gerenciamento de Redes Ópticas com a tecnologia MPLS/GMPLS.*

1. Introdução

No estudo das redes, especificamente das Redes Ópticas de Nova Geração, a simulação aparece muitas vezes como a única ferramenta que se pode usar com relativa facilidade, dada a dificuldade em reunir equipamentos suficientes para experimentar laboratorialmente topologias complexas e determinados cenários hipotéticos de tráfego. Mas mesmo em cenários e topologias mais simples o uso de técnicas de simulação é atrativo seja pela facilidade da coleta de dados e plotagem desses dados em forma de gráfico, seja também pela análise passo a passo dos diferentes protocolos de comunicação existentes.

Existem diversos simuladores de tráfego de redes disponíveis, os que mais se destacam comercial e academicamente são o OPNET [OPNET] e o *Network Simulator (ns)* [VINT]. O OPNET é um simulador comercial largamente utilizado no âmbito corporativo, devido às suas funcionalidades e precisão nos resultados. Ele é mais utilizado em grandes empresas e operadoras de telecomunicações, mas restrito em outros ambientes, como o acadêmico, principalmente devido ao seu custo elevado.

O *ns-2* é utilizado principalmente por pesquisadores, por ter distribuição gratuita e código aberto. Tal fato o torna adequado a situações onde é necessário desenvolver

novas funcionalidades, como em teses e projetos de pesquisa aplicada. No entanto, a sua interface (textual) não é amigável ao usuário. A execução de um experimento de simulação no *ns-2* requer a elaboração de *scripts* em Tcl (*Tool Control Language*) e grande trabalho adicional para obter e visualizar os resultados. Também é comum que o usuário necessite programar em C++ para que as funcionalidades desejadas estejam disponíveis. Além disso, os protocolos e tecnologias no *ns-2*, em geral, são desenvolvidos para uso isolado, para resolução de problemas específicos. Integrá-los, normalmente, é uma tarefa árdua.

Nesse contexto, um dos grandes problemas encontrados pelos alunos de graduação e pós-graduação nos cursos de Computação e Engenharia Elétrica é o alto custo de ferramentas de simulação para análise de protocolos e arquitetura de alto nível tolerante a falhas, em cenários de Redes Ópticas utilizando as tecnologias MPLS (*Multiprotocol Label Switching*) [Rosen et al. 2001] e GMPLS (*Generalized Multiprotocol Label Switching*) [Mannie 2004].

O GLASS (*GMPLS Lightwave Agile Switching Simulator*) [GLASS Overview] é uma das poucas ferramentas voltada para simulação de Redes Ópticas de Nova Geração, livre e de código aberto e com uma interface amigável (gráfica). Entretanto, um dos requisitos para se obter Engenharia de Tráfego em Redes Ópticas de Nova Geração com as tecnologias MPLS/GMPLS, é a capacidade de re-rotear LSPs (Label Switched Path) já estabelecidos na rede, sob certas condições, ou políticas administrativas. Outra característica importante é quando um LSP é re-roteado devido a uma falha em um circuito de comunicação, e quando essa falha estiver resolvida, o LSP deverá ser restabelecido pelo caminho original. Atualmente, o GLASS na sua distribuição (versão 2.0.2) só tem incorporado o primeiro requisito descrito de Engenharia de Tráfego, isto é, re-rotear LSPs já estabelecidos na rede.

Este trabalho tem como objetivo apresentar a implementação do segundo requisito de engenharia de tráfego descrito, no simulador. O objetivo do novo módulo é fazer o restabelecimento dinâmico de LSPs primários, em caso da saída desses do estado de falha, visando o estudo da qualidade de serviço fim-a-fim para uma determinada aplicação em um cenário de redes ópticas com as tecnologias MPLS/GMPLS.

O restante deste trabalho está organizado da seguinte forma. Na próxima seção são apresentados alguns trabalhos relacionados ao oferecimento de ferramentas de simulação. A Seção 3 descreve as características implementadas nos novos métodos inseridos no *framework* do simulador. Na Seção 4 são apresentados o cenário e os experimentos de simulação para avaliar o novo módulo. Finalmente, a Seção 5 conclui o artigo e apresenta possíveis trabalhos futuros.

2. Trabalhos Relacionados

O trabalho apresentado em [Fidalgo et al. 2004] descreve o QNET, um simulador de redes IP desenvolvido com o objetivo de auxiliar trabalhos de pesquisa, projeto e gerenciamento de redes convergentes. O QNET fornece uma interface gráfica ao usuário, como o GLASS e o OPNET, para configurar os cenários e visualizar os resultados. Além disso, beneficia-se do potencial e flexibilidade proporcionados pelo *ns-2*. Entretanto, não possui implementação de módulo para redes ópticas integrado à ferramenta, apesar do *ns-2*, possuir o módulo MPLS Network Simulation (MNS).

O trabalho apresentado em [Nascimento et al. 1999] descreve as arquiteturas física e funcional do protótipo RENATA (Redes Neurais Aplicadas ao Tráfego ATM). Este ambiente permite o desenvolvimento de agentes inteligentes, baseado em redes neurais, destinados à gerência pró-ativa de redes ATM (*Asynchronous Transfer Mode*).

Em [Gonçalves et al. 2003], os autores descrevem uma nova versão para o trabalho apresentado em [Nascimento et al. 1999], chamado de RENATA 2, cuja evolução compreenderá o suporte à gerência pró-ativa de redes IP, MPLS e GMPLS. Dentre possíveis tarefas realizadas pelos agentes gerados pelo ambiente RENATA 2 estão o controle de tráfego e alocação de caminhos entre os *hosts* de um sistema de comunicação. Porém, não implementa o re-roteamento dinâmico de túneis MPLS.

Em [Rochol et al. 2007] são apresentadas as ferramentas mais usadas atualmente para que se possa fazer a avaliação de desempenho de protocolos de rede quando submetidos às mais diversas situações de tráfegos, tanto em ambientes cabeados e estruturados quanto em redes de difusão sem fio. Essas ferramentas são baseadas em plataformas de simulação de software livre.

Com o estudo dos trabalhos citados acima e de outros encontrados na literatura, observa-se pouca preocupação no desenvolvimento ou aprimoramento de ferramentas de simulação que permita modelar e avaliar o desempenho de protocolos de roteamento, restauração e sinalização tolerantes a falhas em redes ópticas com as tecnologias MPLS/GMPLS. Este artigo contribui com o acoplamento de um novo módulo com característica de engenharia de tráfego e baseado nas principais classes das mensagens *PATH* e *RESV* do protocolo de sinalização RSVP-TE (*Resource Reservation Protocol - Traffic Engineering*) para fazer o re-roteamento dinâmico de túneis LSPs.

3. Descrição da Implementação

As principais características implementadas nos métodos da classe *TEAgent*, *RSVP* e *MPLSForwardingTable* são descritas a seguir.

3.1. Classe *TEAgent*

O *TEAgent* é a classe de Engenharia de Tráfego do simulador que faz a verificação dos eventos das mensagens *PATH Message* e *RESV Message* do protocolo RSVP-TE. Esta classe possui um método chamado *callback* que fica de tempos em tempos verificando eventos de falhas ou recuperação no circuito de comunicação MPLS/GMPLS. O método *callback* recebe como parâmetros as mensagens oriundas dos eventos que podem ocorrer. Na ocorrência de qualquer um dos casos, este método direciona a chamada para outro método chamado *interfaceRecoverReport* o qual reporta este evento ao protocolo de sinalização RSVP-TE, passando como parâmetro informações da interface do elemento que deve ser recuperado.

Foi observado que no momento da chamada, o método *interfaceRecoverReport* faz chamada ao método *recoverReportFromAgent* da classe *RSVP*. De acordo com os resultados da engenharia reversa, foi detectado que o método *recoverReportFromAgent* da classe *RSVP*, na implementação distribuída no GLASS, está declarado apenas com a assinatura, não tendo código algum implementado. Os procedimentos implementados neste método da classe *RSVP* são descritos a seguir.

3.2. Classe RSVP

As mensagens de *PATH* e *RESV* do Protocolo RSVP-TE são trocadas entre os elementos de rede para estabelecerem um caminho, ou LSP. Os Objetos utilizados pelas mensagens de *RESV* e *PATH* estão inseridos na classe *RSVP*.

3.2.1. Método *recoverReportFromAgent*

O método *recoverReportFromAgent* como descrito, na implementação atual do simulador, está declarado apenas com a assinatura, não tendo código algum implementado. Sendo assim a nova implementação tem como função principal de executar a recuperação do túnel LSP. A seguir são descritas as principais características do novo código implementado nesse método.

- Obtenção de um vetor contendo informações acerca dos LSPs que devem ser recuperados.
- Obtenção das informações necessárias à mensagem *RESV*. Enviado em resposta às mensagens *PATH*, mas usado também para remover reservas da rede.
- Construção de objeto contendo o endereço IP do elemento (ex: OXC) que vai ser recuperado, o código indicando que se trata de recuperação, o tempo de início da recuperação, o ID do nó a ser recuperado, entre outras informações necessárias.
- Verificação se o LSP, no momento, é de ingresso ou egresso. Se for ingresso, é dado início ao processo de re-roteamento. Se egresso, gera uma notificação MPLS informando como parâmetros: objetos da classe *SessionObject*, o objeto montado no item **c**, um valor booleano (true se *upstream* e false se *downstream*) e um objeto do tipo *ClassObect*. Para tanto, invoca o método *genNotificationMsg* da classe *Message*.

Se o LSP for de ingresso invoca o método *rcvRecoveryReportFromIngress* da classe *TEAgent*, ou seja, volta à chamada para um método dessa classe. O método *rcvRecoveryReportFromIngress* já vem declarado no simulador e tem como objetivo, receber a notificação de recuperação do elemento de ingresso passado para o de egresso. Este método verifica o tipo de componente, no caso OXC, e chama o método *OXCTypeRecover*. O método *OXCTypeRecover* teve sua implementação alterada, pois invocava um método da classe *MPLSForwardingTable* que estava com o corpo originalmente comentado e não compilável. Por este motivo, e por não ser útil para a funcionalidade da forma como estava, o método *OXCTypeRecover* passou a chamar um novo método criado na classe *MPLSForwardingTable* chamado *reRouteToPrimaryPathLSP_OXC*. Este método é explicado na seção 3.3.

3.2.2. Método *sendPrimaryPathOnToEgress*

O método *sendPrimaryPathOnToEgress* foi criado e implementado com o objetivo de auxiliar o método *reRouteToPrimaryPathLSP_OXC* no re-roteamento de túneis LSPs primário. O método *sendPrimaryPathOnToEgress* recebe como parâmetro o LSP primário e seu respectivo backup. Dessa forma, ao encontrar o LSR de egresso envia uma mensagem de notificação *PATH Message* e não mais *RESV Message*. Depois disso, o método *sendPrimaryPathOnToEgress* da classe *RSVP* invoca o método *rcvPrimaryPathOnFromEgress* da classe *TEAgent*. O método

rcvPrimaryPathOnFromEgress também é novo, tendo a mesma semântica do *rcvRecoveryReportFromIngress* para manter o padrão utilizado.

3.3. Classe MPLSForwardingTable

O novo método *reRouteToPrimaryPathLSP_OXC* tem suas características descritas a seguir.

3.3.1. Classe MPLSForwardingTable

- Responsável por identificar os lambdas, que correspondem a distância entre valores repetidos num padrão de onda, do LSP de backup (tanto de entrada quanto de saída), além de também identificar os lambdas dos LSPs primários que devem ser ativados, tanto de entrada e saída. Uma vez feito isto, desconecta o par de lambdas (entrada e saída) do LSP de backup e conecta o par de lambdas (entrada e saída) correspondente ao LSP primário. Este procedimento é realizado para LSPs ingressos.
- No caso de LSPs egressos, certifica se o protocolo de sinalização sendo utilizado é o RSVP-TE. Se for invoca o método *sendPrimaryPathOnToEgress* (novo método criado) da classe *RSVP*.

Essas novas características implementadas nas classes acima citadas visam uma melhoria no simulador no que diz respeito às características de Engenharia de Tráfego já disponibilizadas com o acoplamento de um novo módulo de re-roteamento dinâmico de LSPs.

4. Análise da Implementação

4.1. Cenário

Nesta seção será apresentado um cenário de simulação para avaliar o módulo de re-roteamento dinâmico de LSPs no simulador. A topologia do cenário foi elaborada para que se tenham caminhos redundantes em que se possa utilizar o módulo tolerante a falhas de re-roteamento dinâmico de LSPs. Nessa topologia, ilustrada pela Figura 1, tem-se uma rede com 17 nós e cada enlace óptico têm capacidade de 10GB. Os nós 5, 6, 7 e 8 atuam como LERs (*Label Edge Routers*-Roteadores de Rótulos de Borda) e os nós 9, 10, 11, 12, 13, 14, 15 e 16 são OXCs (*Optical Crossconnect*), responsáveis por realizar a comutação óptica e comutação por rótulos. Os nós 1, 2, 3 e 4 são hosts conectados à rede e executam aplicações. No simulador, quem cria os caminhos comutados por rótulos é um no *host* que pode ou não estar conectado à topologia da rede simulada. Nesta simulação o nó que cria os túneis LSPs é o host 0.

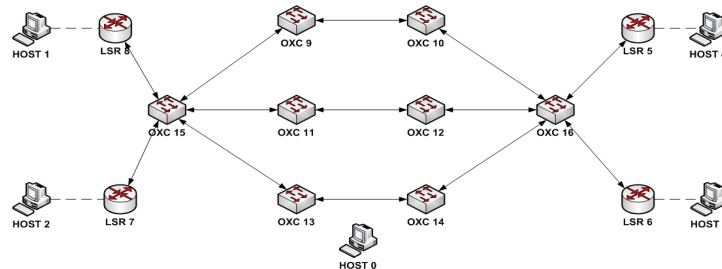


Figura 1. Topologia utilizada na simulação

O host 1 solicita tráfego do tipo UDP (User Datagram Protocol) para o host 3. Sendo assim, nesta simulação definiu-se que o tráfego teria como origem o nó *host 3* e o destino será o nó *host 1*. O túnel LSP Primário é criado entre os LERs 6 e 8, porém para criar esse túnel deve-se primeiramente criar um LSP entre os nós OXCs 15, 9, 10 e 16. O túnel LSP Reserva para o túnel LSP Primário é criado entre os LERs 6 e 8, porém para criar esse túnel de reserva deve-se primeiramente criar um LSP entre os nós OXCs 15, 13, 14 e 16.

4.2. Resultados da Simulação

Utilizando o cenário descrito, partiu-se para a realização de simulações com o objetivo de avaliar o módulo de re-roteamento dinâmico, comparando resultados de eventos de falhas e recuperação com o novo módulo inserido no simulador e com a distribuição padrão GLASS, sem o novo módulo. Os eventos são descritos a seguir:

- Evento de falha no enlace óptico entre os nós OXCs 9 e 10 dos LSP Primário, onde o evento ocorre no instante 40s da simulação.
- Evento de recuperação no enlace óptico entre os nós OXCs 9 e 10 do LSP Primário, onde o evento ocorre no instante 50s da simulação.
- Evento de falha no enlace óptico entre os nós OXCs 13 e 14 do LSP de Reserva, onde o evento ocorre no instante 60s da simulação.

As Figuras 2 e 3 mostram o comportamento do parâmetro Throughput (vazão), que é a métrica que corresponde a quantidade de bytes recebidos pelo destino do túnel LSP entre os hosts 6 e 8 para os eventos descritos. Na primeira é apresentado o comportamento do Throughput do Túnel LSP entre os host 6 e 8 sem o novo módulo de re-roteamento inserido no simulador.

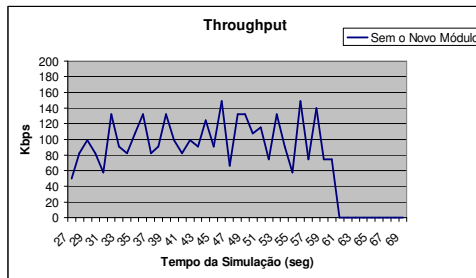


Figura 2. Throughput do Túnel LSP sem o Novo Módulo

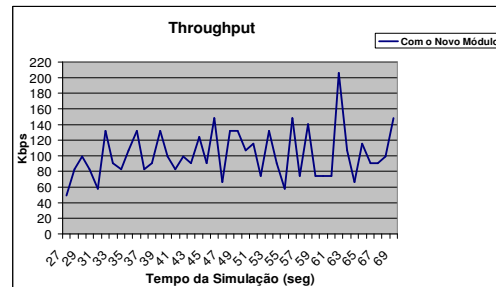


Figura 3. Throughput do Túnel LSP com o Novo Módulo

No instante 40s decorrido da simulação ocorre um evento de falha no segmento entre os OXCs 9 e 10 do Túnel LSP Primário. Nesse momento o tráfego do túnel LSP Primário é roteado para seu túnel LSP de Reserva fazendo com que um dos requisitos de engenharia de tráfego seja estabelecido. Decorrido 10s da existência do evento de falha no segmento do túnel LSP Primário, esse é restabelecido no instante 50s. Como o cenário é simulado sem o novo módulo após o restabelecimento do túnel LSP Primário o tráfego não é re-roteado para o Túnel LSP Primário. No instante 60s foi simulado um evento de falha no segmento entre os nós OXCs 13 e 14 do túnel LSP de Reserva. Como o tráfego não foi re-roteado para o túnel LSP Primário quando do seu restabelecimento,

a Figura 2 mostra que existe uma diminuição acentuada no parâmetro Throughput do tráfego entre os host 6 e 8.

Na Figura 3 é apresentado o comportamento do Throughput do Túnel LSP entre os host 6 e 8 com o novo módulo de re-roteamento inserido no simulador. A Figura 3 mostra que após o restabelecimento do segmento entre os OXCs 9 e 10 no instante 50s decorridos da simulação, o tráfego é re-roteado para o LSP Primário fazendo com que o segundo requisito de engenharia de tráfego seja estabelecido. Como no instante 60s é estabelecido um evento de falha no segmento entre os OXCs 13 e 14 do túnel LSP de Reserva, pode ser observado que o comportamento do valor de Throughput entre os host 6 e 8 ficou dentro dos parâmetros aceitáveis. Na Figura 4 e 5 é apresentada a quantidade de pacotes ópticos no Túnel LSP entre os host 6 e 8 após as ocorrências dos eventos sem e com o novo módulo de re-roteamento inserido no simulador.

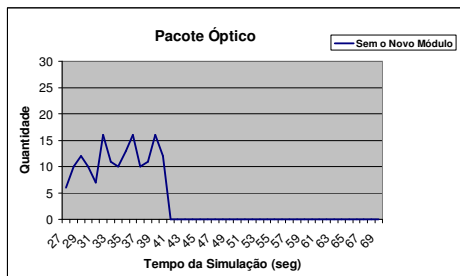


Figura 4: Sem o Módulo

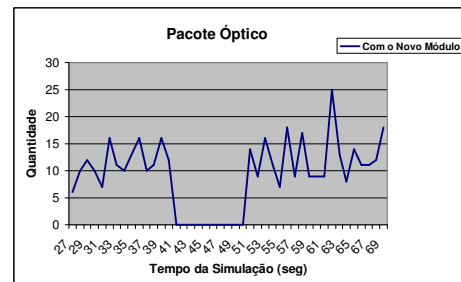


Figura 5: Com o Módulo

Na Figura 4 podemos observar uma redução total na quantidade de pacotes após o primeiro evento de falha que ocorre no instante 40s. Na Figura 5 também é apresentada a quantidade de pacotes ópticos no Túnel LSP entre os host 6 e 8 após as ocorrências dos eventos porém com o novo módulo de re-roteamento inserido no simulador. Com o novo módulo de re-roteamento dinâmico de LSPs acoplado no simulador, a quantidade de pacotes perdidos após as ocorrências diminuir sensivelmente ficando dentro dos parâmetros aceitáveis.

A análise básica dos resultados obtidos nas duas simulações procura evidenciar o fato de que o re-roteamento dinâmico de LSPs após um evento de recuperação, obtida através da introdução do novo módulo baseado nos objetos do protocolo RSVP-TE, permite um melhor balanceamento do uso dos recursos da rede, no caso caracterizado pela uso racional dos túneis LSPs.

5. Considerações e Trabalhos Futuros

O objetivo deste artigo foi descrever um novo módulo implementado no simulador GLASS para redes Ópticas MPLS/GMPLS. Este novo módulo tem como característica principal, executar o re-roteamento dinâmico de LSPs primários em caso do seu restabelecimento. Na última distribuição do GLASS, Versão 2.0.2, que é a utilizada neste trabalho, não existia a função de re-roteamento de túneis LSPs em cenários de redes MPLS/GMPLS. Dificultando assim, a implementação de requisitos de Engenharia de Tráfego como parâmetro de QoS (Quality of Service), tais como: *throughput*, *delay*, *jitter*, *taxa de perda de pacotes* e *taxa de erro*.

Assim o novo módulo visa implementar no simulador a função de re-roteamento dinâmico de LSPs primários utilizando características das mensagens de sinalização do protocolo RSVP-TE. Portanto, este artigo, através do acoplamento da nova funcionalidade, fornece uma outra alternativa de ferramenta de simulação para o estudo fim-a-fim do nível de qualidade de serviço em redes ópticas com as tecnologias MPLS/GMPLS.

No decorrer deste trabalho, verificou-se a necessidade de avaliar o novo módulo com cenários mais complexos a fim de provar o correto funcionamento da implementação realizada, bem como aprimorar algumas outras funcionalidades no simulador como a utilização de gerenciamento baseado em políticas para provisionamento dinâmico de *DiffServ* e *GMPLS*. O aprimoramento do módulo de plotagem dos gráficos em cima dos resultados gerados pela simulação e a adição de técnicas de simulação estocástica também se fazem necessários.

Referências

- Andersson, L.; Swallow, G. The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols, IETF RFC 3468, February 2003.
- Fidalgo, J.; Kamienski, C.; Farias, T.; Dias, K.; Almeida, F.; Sadok, D.; Sadok, J. QNet – Um Simulador Gráfico de Tráfego IP para Redes Convergentes. 22º Simpósio Brasileiro de Redes de Computadores – SBRC 2004, maio 2004, Rio Grande do Sul.
- GLASS Overview – Version: Draft 1.0. Disponível em: http://www-x-antnd.nist.gov/glass/doc/html/overview/GLASS_Overview.htm; Acessado em: Julho de 2007.
- Gonçalves, C.; Oliveira, A.; Andrade, R.; Castro, M.. Utilizando Redes Neurais para Predição de Falhas em Links de Redes Ópticas. Boletim Bimestral sobre Tecnologia de Redes, Rio de Janeiro, v. 7, 2003.
- Mannie E. (Editor). Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945. Outubro, 2004.
- Nascimento, A.; Franklin, M.; Oliveira, M. Desenvolvendo Agentes Inteligentes para a Gerência Pró-Ativa de Redes ATM. 17º Simpósio Brasileiro de Redes de Computadores – SBRC 1999, maio 1999, Bahia.
- OPNET, “OPNET Web Site”. Disponível em: <http://www.opnet.com>. Acessado em Julho de 2007.
- Rochol, J.; Souza, L.; Sewald, L.; Fernandes, R.; Mori, O. Plataformas de Simulação de Software Livre para Redes Fixas e Móveis: Características, Suporte, Instalação e Validação. <http://nsl.csie.nctu.edu.tw/>, Acessado em Julho de 2007.
- Rosen, E.; Viswanathan, A.; Callon, R. Multiprotocol Label Switching Architecture. RFC 3031. Janeiro, 2001.
- VINT Project, The Virtual InterNetwork TestBed. A Collaboration among USC/ISI, Xerox PARC, LBNL, and UCB. Url: <http://www.isi.edu/nsnam/vint/index.html>. Acessado em Julho de 2007.