

Dimensionamento de Túneis LSPs Através da Ferramenta GLASS: Uma Proposta de Implementação de um Novo Módulo de Re-roteamento Dinâmico

Klenilmar L. Dias^{1,2,3}, Klissiomara L. Dias², Antônio Jorge. G. Abelém^{1,3}, Kelvin L. Dias^{1,3}

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

²Faculdade Seama – Macapá, AP – Brasil

³Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal do Pará
(UFPA) - Belém, Pa - Brasil

{kldias,kld}@seama.edu.br, {abelem,kld}@ufpa.br

Abstract. *The GLASS (GMPLS Litghwave Agile Switching Simulator) is a tool for simulation of next generation optical networks. It is based on free and open source code developed in JAVA. This paper aims at describing a new feature added to the GLASS, in order to facilitate the dynamic rerouting of LSPs (Label Switched Paths) in GMPLS networks scenarios based on the RSVP-TE signaling protocol.*

Resumo. *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.*

1. Introdução

As tecnologias de redes têm evoluído muito rápido, tanto nos aspectos de capacidade de transmissão quanto no que diz respeito à confiabilidade, distâncias alcançadas, métodos de compartilhamento do meio e tipos de aplicações suportadas. Todo este desenvolvimento tem tido como principais forças motoras a evolução dos meios físicos, em particular, das fibras ópticas, o aprimoramento de técnicas de transmissão, a integração crescente de recursos multimídia às aplicações e o desenvolvimento da microeletrônica.

Em função da crescente convergência de aplicações e tipos de tráfego observados nas redes de comunicação, que têm capacidade limitada, os conceitos de qualidade de serviço e priorização de tráfego passaram a ser objetos de freqüente estudo. Hoje em dia, ainda não se dispõe de ferramentas completas para realizar os testes e análises para a implementação de qualidade de serviço (QoS) em um sistema de comunicação.

Diante da carência de recursos para se criar laboratórios de sistema de comunicação bem estruturados, diversos segmentos acadêmicos têm optado por testar e

certificar propostas através de técnicas alternativas de avaliação de desempenho, em especial a simulação. O uso de simulação tem sido uma ferramenta poderosa no processo de projetar e avaliar arquitetura, tecnologias e protocolos de comunicação de dados.

Outros meios de auxílio no processo de projetar e avaliar arquitetura, tecnologias e protocolos de comunicação de dados é através da avaliação analítica e experimentos práticos. A avaliação analítica é útil para demonstrar o impacto geral de uma determinada arquitetura em um cenário de comunicação de dados, os argumentos de entrada de um protocolo ou de mudança em condições específicas da rede. Porém, requer modelos simplificados da realidade, limitando o número de variáveis consideradas e as fontes de não-determinismo, já que os resultados são obtidos alterando argumentos em fórmulas.

Já os experimentos práticos, são baseados em um conjunto de execuções de um protocolo sobre um sistema de comunicação de dados real, portanto, produzem os resultados mais precisos. Apesar disso, devido à grande influência da topologia e dos sistemas e configurações envolvidos, os resultados coletados são dependentes de um cenário específico, o que torna difícil sua reprodução em outro. Para protocolos, essa técnica requer que o mesmo já esteja implementado antes que seu funcionamento geral possa ser testado.

A simulação se encontra no nível intermediário, entre a avaliação analítica e os experimentos práticos. Ela permite que o projetista do cenário de comunicação de dados ajuste o nível de detalhe, atendo-se apenas aos recursos desejados [Jain, 1991].

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. Contudo, mesmo em cenários e topologias mais simples o uso de técnicas de simulação é atrativo seja pela facilidade da coleta de dados e plotagem destes em forma de gráfico, bem como pela análise passo a passo dos diferentes protocolos de comunicação existentes. Através da simulação, podem ser feitos o projeto, dimensionamento e avaliação de desempenho de sistemas de comunicação de dados.

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 e proprietário 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.

As Redes Ópticas de Nova Geração são promissoras como mecanismo de suporte ao crescimento acelerado de tráfego. Essas redes, apesar de oferecerem uma grande quantidade de largura de banda, podendo levar a pensar que não há necessidade de gerenciá-las, apresentam problemas semelhantes quando comparadas com redes baseadas em pacotes.

Devido a esse fato, o MPLS (*Multiprotocol Label Switching*) [Rosen et al. 2001] evoluiu para o GMPLS (*Generalized MPLS*) [Mannie 2004] para fornecer ao plano de controle (sinalização e roteamento) a possibilidade de encaminhar (comutar) em vários domínios diferentes, tais como domínios que utilizam pacotes (redes IP), tempo (TDM) e comprimento de onda (WDM). O GMPLS permite que a identificação de um determinado caminho (LSP – Label Switched Path) seja feita, por exemplo, através de comprimento de onda em uma rede óptica.

As redes ópticas estão em constante expansão para atender a demanda de informações requisitadas pelos novos aplicativos. Para evitar que essas expansões causem problemas ao seu funcionamento ou acabem sendo muito dispendiosas, é necessário estimar qual será o crescimento do tráfego na rede resultante. Obter uma estimativa confiável não é um procedimento trivial, mas é de fundamental importância para que o processo de expansão se torne eficiente.

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 [Rosen et al. 2001] e GMPLS [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) desenvolvida e implementada em JAVA, através de esforço conjunto dos grupos High Speed Network Technologies Group e Internetworking Technologies Group vinculados à Advance Network Technologies Division no NIST (National Institute of Standards and Technology) [NIST]. Esta ferramenta é adequada tanto para usuários que estão interessados em analisar o comportamento de redes com tecnologia MPLS/GMPLS quanto para aqueles cujo objeto de estudo é a camada óptica sem o uso de protocolos baseados na tecnologia MPLS/GMPLS.

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 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 escreve uma visão geral da ferramenta. A Seção 4 descreve um resumo das características implementadas nos novos métodos inseridos no *framework* do simulador. Na Seção 5 são apresentados o cenário e os experimentos de simulação para avaliar o novo módulo. Finalmente, a Seção 6 conclui o artigo e apresenta possíveis trabalhos futuros.

2. Trabalhos Relacionados

Existe uma quantidade considerável de trabalho em simulação de redes de comunicação de dados, gerando várias ferramentas. Tais ferramentas podem ser classificadas em *simuladores* e *ambientes de teste*. Como exemplos de ambientes temos o Delayline [Ingham et al. 1994], Dummynet [Rizzo, 1997], ComFIRM [Leite, 2000] e o x-sim [Brakmo et al. 1996]. Estas ferramentas não permitem que todas as condições sejam testadas, porque elas são limitadas por condições existentes.

Simuladores permitem que qualquer condição específica conhecida seja testada. Em geral, o projetista tem muito mais domínio sobre o experimento com simuladores do que com ambientes de teste. Porém, o modelo a ser simulado representa uma visão abstrata do protocolo real e da rede (com topologia, conectividade, perda de pacote, padrões de tráfego, etc.). Desenvolver um modelo de simulação envolve partir de premissas que simplifiquem o cenário de modo a focalizar-se nos aspectos relevantes do estudo. De qualquer forma, a precisão dos resultados da simulação depende de quão válidas são as premissas iniciais [Little, 2008].

Como exemplos de ferramentas temos o trabalho apresentado em [Fidalgo et al. 2004] que 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 de extensão denominado MPLS Network Simulation (MNS) [Ahn et al. 2002].

Em [Muhammad et al. 2001] é apresentado o Simmcast, um *framework* de simulação de eventos discretos baseado em processos que auxilia no projeto e avaliação de protocolos multicast. É baseado em Java e permite que os protocolos simulados sejam especificados utilizando um *framework multi-thread* orientado a objetos.

O trabalho apresentado em [Nascimento et al. 1999] descreve as arquiteturas física e funcional do protótipo RENATA (Redes Neurais Aplicadas ao Tráfego ATM). Esta ferramenta de simulação 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 [Cunha et al. 2003] é apresentada uma análise comparativa de algumas das principais ferramentas de simulação utilizadas para avaliação de desempenho de redes e projetos de protocolos. Neste trabalho foram selecionadas, para estudos mais aprofundados, três destas plataformas de simulação: NS-2, GloMoSim e NCTUns.

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. Visão Geral da Ferramenta

O GLASS, como descrito anteriormente, é uma ferramenta de simulação desenvolvida em Java através de esforço conjunto dos grupos de pesquisa do NIST [NIST].

É uma ferramenta de simulação de redes ópticas que permite modelar e avaliar o desempenho de protocolos de roteamento, restauração e sinalização em redes ópticas [GLASS Overview]. O GLASS é adequado tanto para usuários que estão interessados em analisar o comportamento de redes WDM (Wavelength Division Multiplexing) quanto para aqueles cujo objeto de estudo está relacionado às redes ópticas de nova geração com plano de controle baseado na tecnologia GMPLS.

O GLASS tem uma estrutura de simulação, que permite a avaliação e protocolos de roteamento, restauração e sinalização em um ambiente de rede óptica e permite que os pesquisadores e projetistas de redes ópticas estudem o comportamento dos algoritmos e protocolos sem a necessidade de se construir um cenário real. Este simulador é a evolução do MERLin (Modeling Evaluation and Research of Lightwave Network) [MERLin] uma ferramenta de simulação focada na avaliação e modelagem dos algoritmos de atribuição de rota e comprimento de onda em redes WDM.

O GLASS foi implementado utilizando como base a plataforma SSF (*Scalable Simulation Framework*) [SSFNet] que oferece componentes de redes como *hosts*, roteadores, enlaces de comunicação e alguns protocolos de redes. A Figura 1 mostra a arquitetura do GLASS com o *framework* SSF.

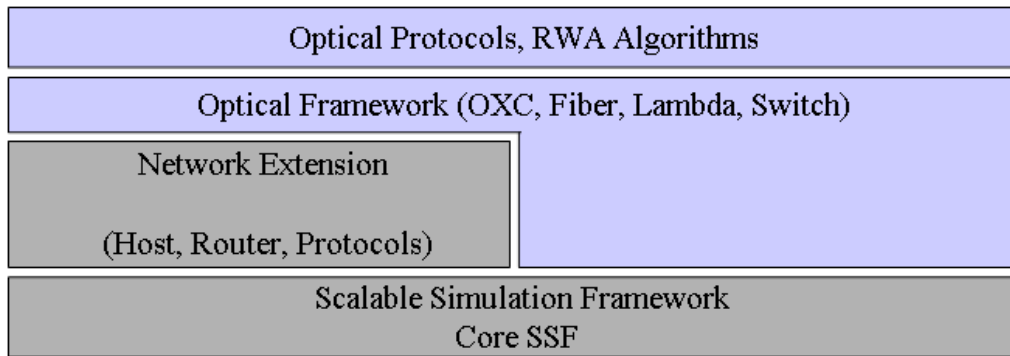


Figura 1. Arquitetura do Simulador com SSF

O simulador utiliza a DML (*Domain Modeling Language*) [DML] para projetar a topologia e gerar os *scripts* de simulação. A DML também permite um alto nível de descrição dos componentes e da configuração da topologia. Através desta linguagem é possível descrever os aspectos estáticos da rede, como a topologia da rede e suas propriedades, bem como os aspectos dinâmicos do cenário, que envolvem os protocolos que serão utilizados e os eventos que estão associados ao cenário.

A topologia de uma rede no GLASS é representada por dois planos. O plano físico que é formado por componentes tais como OXCs (Optical Crossconnect), enlaces, fibras ópticas, *switches*, LSR (*Label Switched Router*) entre outros, e um plano lógico que contém informações de rota óptica, caminhos ópticos, canal óptico, etc. A Figura 2 ilustra os dois planos descritos.

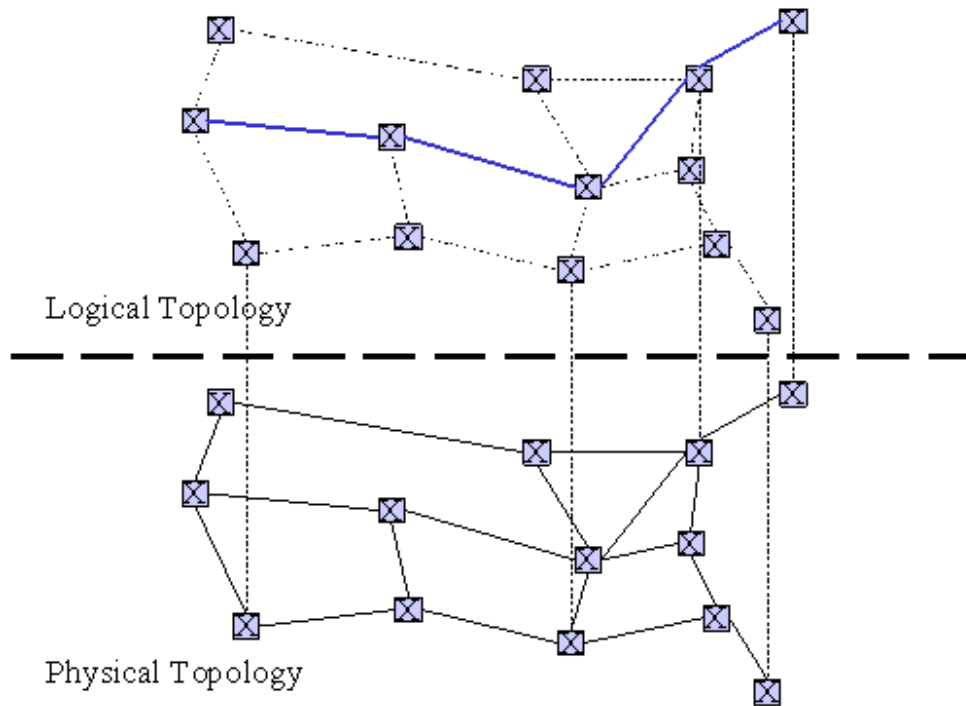


Figura 2. Topologias do Simulador GLASS

No plano físico cada componente pode ser configurado e possui vários parâmetros de configuração como é ilustrado na Figura 3.

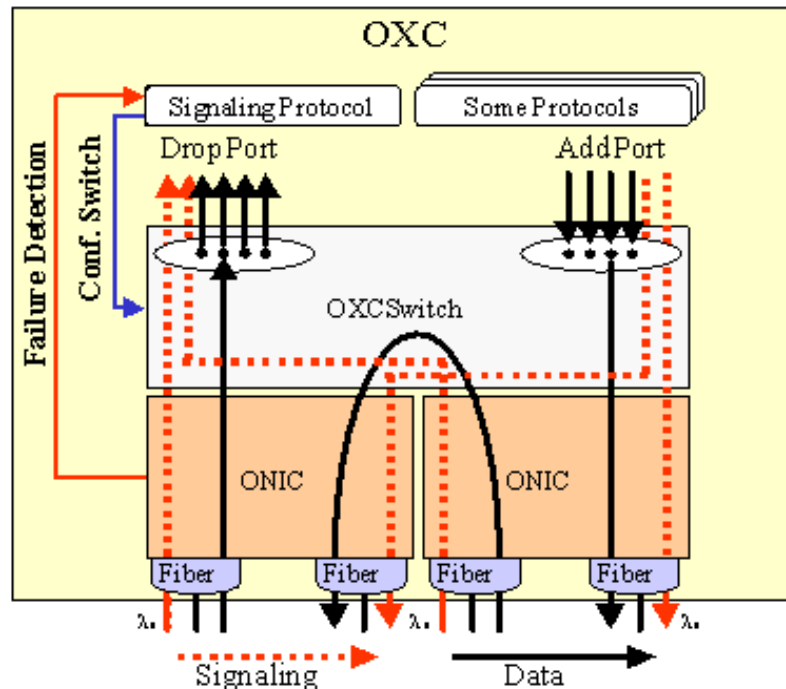


Figura 3. Componente do plano físico

O componente OXC fornece tanto a comutação O/O/O (*Optical-Optical-Optical*), quanto a O/E/O (*Optical-Eletronic-Optical*). A configuração do componente é feita por um protocolo de sinalização. O GLASS possui interfaces necessárias para configurá-la corretamente.

O plano lógico, ilustrado na Figura 2, reside em cima do plano físico da topologia do simulador. Estes componentes são utilizados como interface entre o simulador e os algoritmos implementados nele. Os componentes que constituem o plano lógico são apresentados na Figura 4. Cada componente tem uma variedade de atributos e métodos. Por exemplo, o componente rota óptica (*OpticalRoute*) do simulador contém o resultado do algoritmo de roteamento.

O caminho óptico (*Optical Path*) é resultado do algoritmo de atribuição de comprimento de onda. O caminho óptico contém vários canais *lambda*, representados pelo componente canal óptico (*OpticalChannel*). Cada canal contém uma lista de segmentos, que na figura corresponde ao componente *OpticalChannelSegment*, e cada segmento é atribuído ao *lambda* correspondente, representado na figura pelo componente de mesmo nome.

O plano lógico viabiliza aos implementadores dos algoritmos e protocolos um modo flexível de criar e estabelecer conexões.

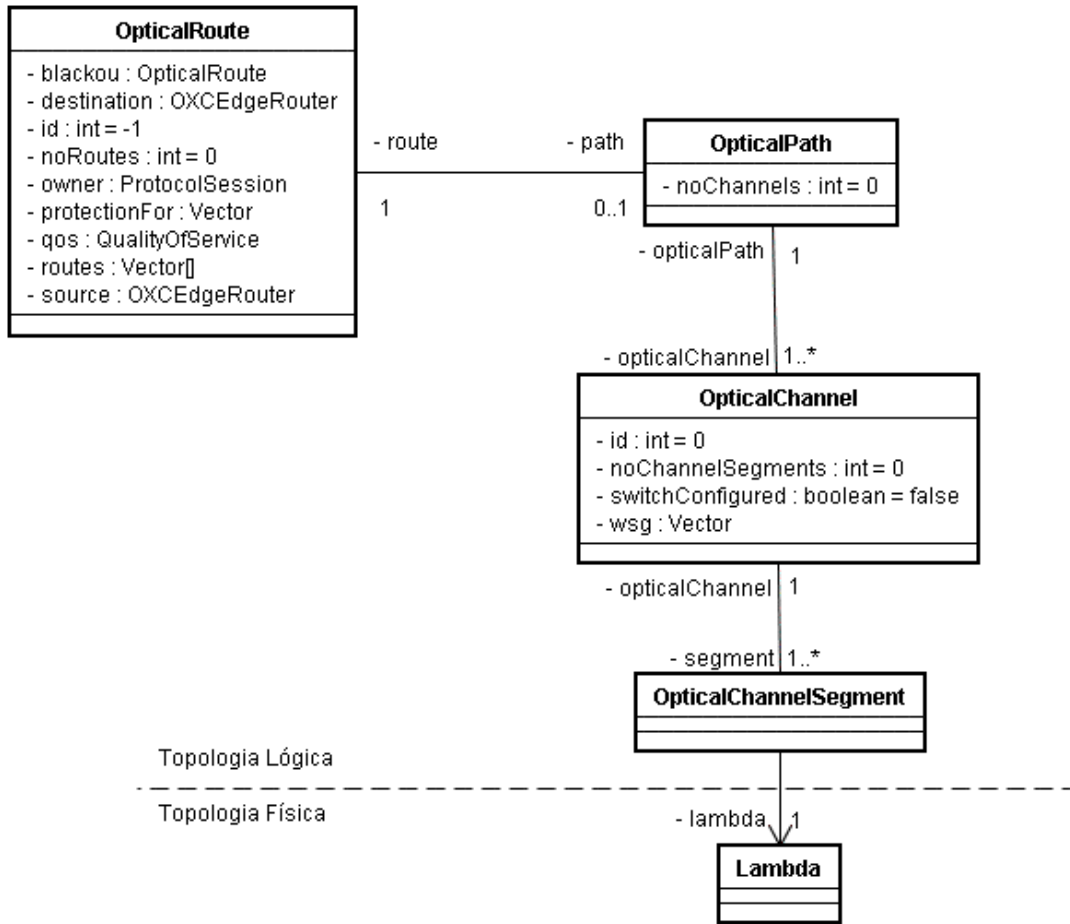


Figura 4. Componentes do Plano Lógico

A distribuição atual do simulador (versão 2.0.2) não possui no seu módulo de recuperação de falhas de túneis LSPs a característica de re-roteamento dinâmico de túneis LSPs primários quando da saída desses do estado de falha. Dificultando assim, a implementação de cenários tolerantes à falhas para o estudo das características de Engenharia de Tráfego.

4. Descrição da Implementação

Por questões de legibilidade, o diagrama de classes apresentado na Figura 5 ilustra apenas as principais classes e alguns métodos diretamente envolvidos na solução. Para ser possível a inserção do novo módulo de re-roteamento, inicialmente foi necessário realizar ajustes no código (na versão distribuída do simulador) para que este compilasse perfeitamente, por exemplo, chamadas de métodos que não existiam ou existiam com outro nome. A técnica de Engenharia Reversa se fez necessária para um melhor entendimento dos pacotes implementados, a partir do qual foi possível evoluir o simulador, inserindo a funcionalidade apresentada neste artigo.

As classes TEAgent, RSVP e MPLSForwardingTable, originárias do GLASS, precisaram ser estendidas para contemplar a solução apresentada, conforme explicado a seguir.

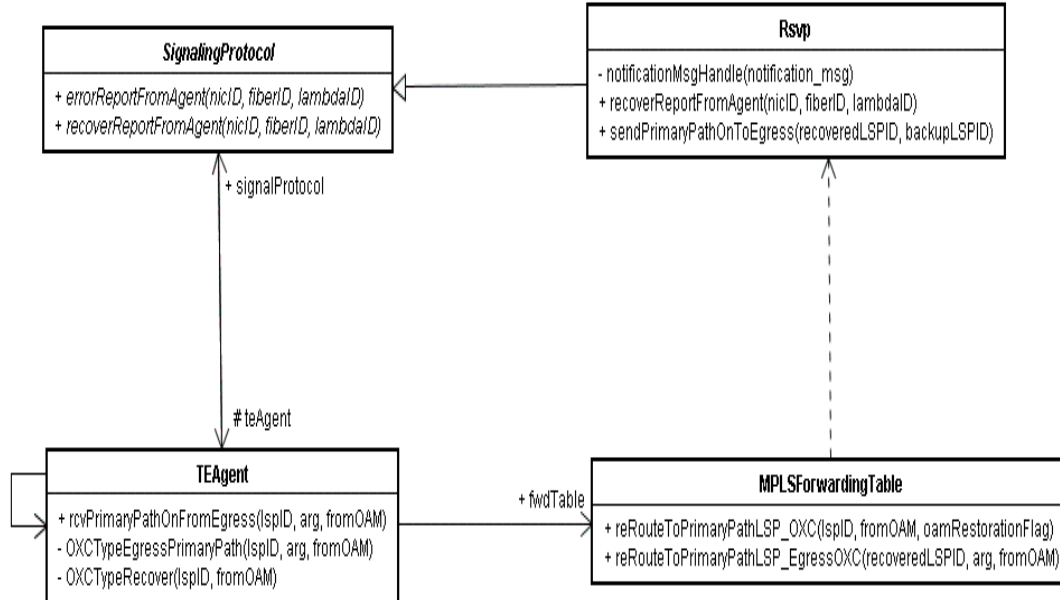


Figura 5. Principais classes e métodos participantes da solução

4.1. 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, é feito o reporte do evento ao protocolo de sinalização RSVP-TE, passando como parâmetro informações da interface do elemento que deve ser recuperado. Como parte deste processamento, é feita uma chamada ao método *recoverReportFromAgent* da classe *RSVP*, o qual é explicado a seguir.

4.2. 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*

4.2.1. Método recoverReportFromAgent

O método *recoverReportFromAgent* é um método abstrato¹ herdado da classe *SignalingProtocol*. Por este motivo, o mesmo teve de ser sobrescrito na subclasse

¹ Método declarado apenas com a assinatura (tipo de retorno, nome e lista de parâmetros), sem processamento algum, em uma classe abstrata. Subclasses que herdam da classe que declara um método abstrato devem sobrescrever (overriding) o método abstrato, caso desejem fornecer processamento específico.

RSVP, a fim de implementar o processamento desejado. Desta forma, o processamento implementado tem como função principal executar a recuperação do túnel LSP. A seguir são descritas as principais características do novo código implementado nesse método.

- a. Obtenção de um vetor contendo informações acerca dos LSPs que devem ser recuperados.
- b. Obtenção das informações necessárias à mensagem *RESV*. Enviado em resposta às mensagens *PATH*, mas usado também para remover reservas da rede.
- c. 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.
- d. 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 é realizada a verificação do tipo de componente, no caso *OXC*, e em seguida o método *OXCTypeRecover* da classe *TEAgent* é invocado.

O método *OXCTypeRecover* delega parte do seu processamento para um novo método criado na classe *MPLSForwardingTable* chamado *reRouteToPrimaryPathLSP_OXC*. Este método é explicado na Seção 4.3.

4.2.2. Método *sendPrimaryPathOnToEgress*

O método *sendPrimaryPathOnToEgress* foi criado com o objetivo de auxiliar o método *reRouteToPrimaryPathLSP_OXC*, localizado na classe *MPLSForwardingTable*, 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*. Em seguida, invoca o método *rcvPrimaryPathOnFromEgress* da classe *TEAgent*, responsável pela verificação do tipo de componente.

4.3. Classe *MPLSForwardingTable*

4.3.1. Método *reRouteToPrimaryPathLSP_OXC*

- a. 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.
- b. 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 túneis LSPs.

5. Análise da Implementação

5.1. 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, ilustrada pela Figura 6, foi elaborada para que se tenham caminhos redundantes em que se possa utilizar o módulo tolerante a falhas de re-roteamento dinâmico de LSPs. Neste trabalho, foi realizado apenas um experimento para avaliar a implementação.

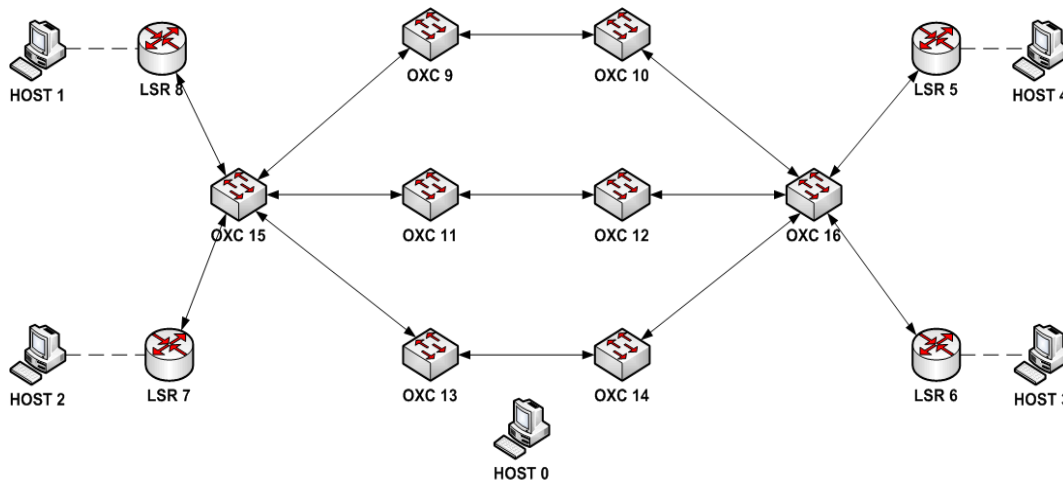


Figura 6. Topologia utilizada na simulação

Nessa topologia, 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, 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.

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.

5.2. Resultados da Simulação

Partiu-se para a realização de simulações com o objetivo de avaliar o módulo de 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.

Simulou-se um evento de falha no LSP primário entre os nós 9 e 10 e a falha ocorre no instante 40s da simulação e o enlace retorna à condição inicial no instante 50s. Um evento de falha também é simulado no LSP de reserva entre os nós 13 e 14 e ocorre no instante 60s da simulação.

As Figuras 7 e 8 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 roteamento inserido no simulador.

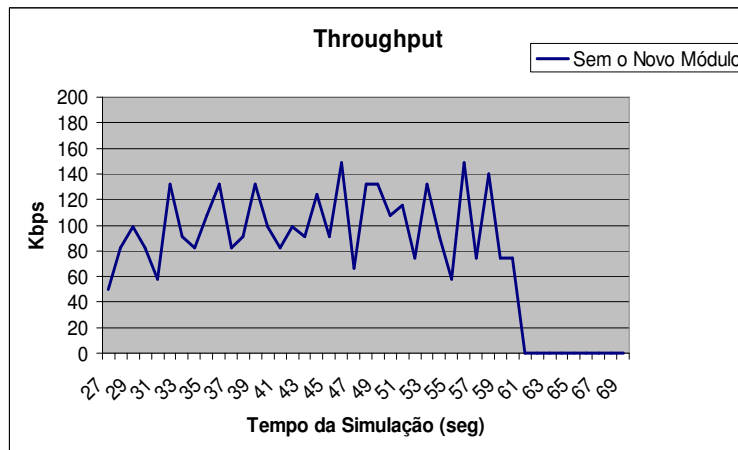


Figura 7. Throughput do Túnel LSP sem o Novo Módulo

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.

Decorrido 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 7 mostra que não houve tráfego, ocasionando uma diminuição total no parâmetro Throughput entre os host 6 e 8.

Na Figura 8 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 8 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.

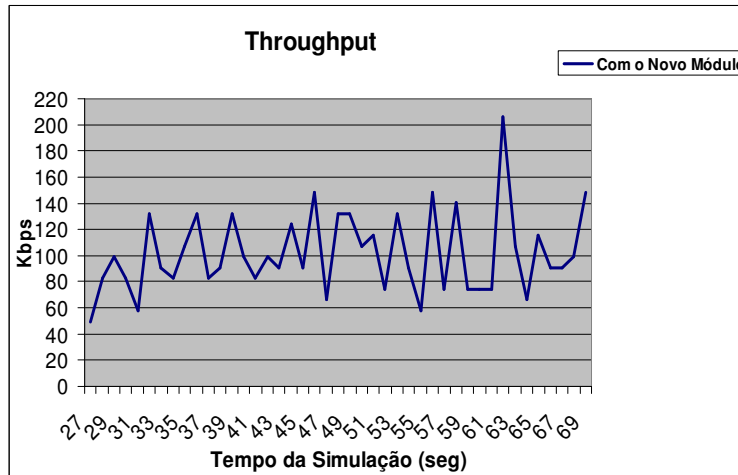


Figura 8. Throughput do Túnel LSP com o Novo Módulo

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 não teve uma diminuição total.

Nas Figuras 9 e 10 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.

Na Figura 9 podemos observar uma redução total na quantidade de pacotes após o primeiro evento de falha que ocorre no instante 40s.

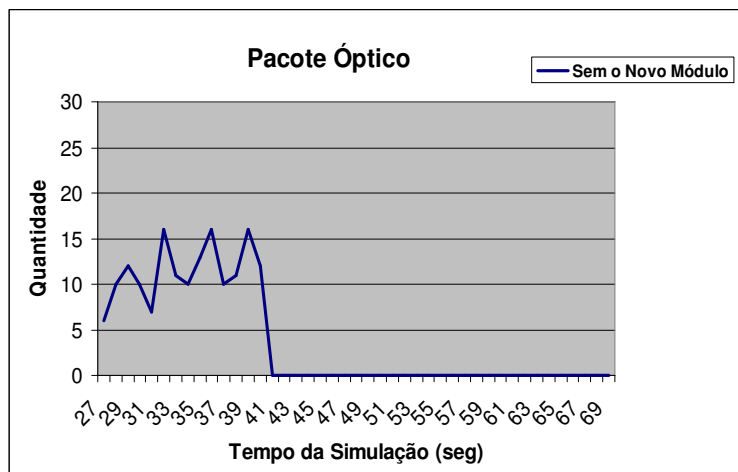


Figura 9. Pacote Óptico Sem o Novo Módulo

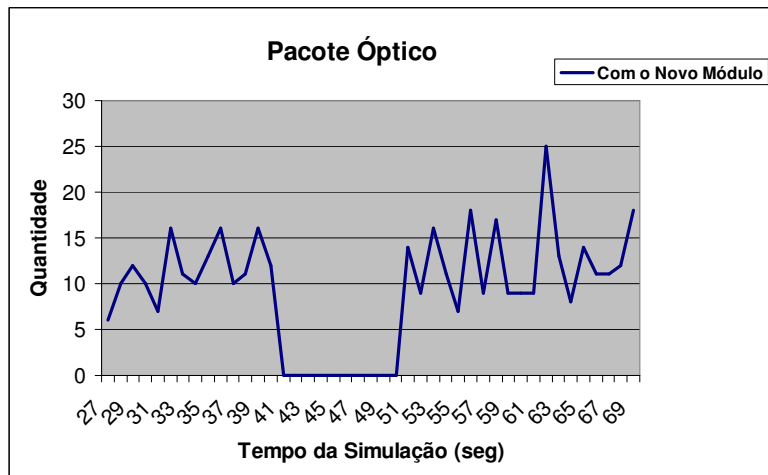


Figura 10. Pacote Óptico Com o Novo Módulo

Na Figura 10 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.

6. 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 existe 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 agregar ao 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

- Ahn, G.; Chun, W. Architecture of MPLS Network Simulator (MNS). Technical Report, Departamento de Computer Engineering, Chungman University of Korea. 2002.
- Andersson, L.; Swallow, G. The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols, IETF RFC 3468, February 2003.
- Brakmo, S. L.; Peterson, L. L. “Experiences with Network Simulation”, In ACM SIGMETRICS, Philadelphia, PP. 80-90, Maio, 1996.
- Cunha, L. S.; Rochol, J.; Dalto, L.; Fernandes, R. H.; Mori, O.. Plataformas de Simulação para Redes Fixas e Móveis: Características, Suporte, Instalação e Validação. In: 2nd Internacional Information and Telecommunication Technologies Symposium, 2003, Florianópolis. Proceedings of the 2nd Internacional Information and Telecommunication Technologies Symposium, 2003.
- DML, “Domain Modeling Language – Reference Manual”. Disponível em <http://www.ssfnet.org/SSFdocs/dmlReference.html>. Acessado em Março de 2008.
- 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.antd.nist.gov/glass/doc/html/overview/GLASS_Overview.htm; Acessado em: Março de 2008.
- 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.
- Ingham, D. B.; Parrington, G. D. “A Wide-Area Network Emulation Tool” Computing System, v.7, n.3, pp. 313-332, 1994.
- Jain, R. “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling”, John Wiley & Sons, New York, 1991, 685p.
- Leite, O. F. “ComFIRM – Injeção de Falhas de Comunicação Através da Alteração de Recursos do Sistema Operacional”, Dissertação de Mestrado, CPGCC – UFRGS, Porto Alegre, Dezembro, 2000.
- Little, C. M. “JavaSim Users Guide”, Public Release, Version 0.3. Disponível em <http://javasim.ncl.ac.uk/>. Acessado em Março de 2008.

- Mannie, E. (Editor). Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945. Outubro, 2004.
- MERLin, “Modeling Evaluation and Research of Lightwave Network Web Site”. Disponível em: http://w3.antd.nist.gov/Hsntg/prd_merlin.html. Acessado em Março de 2008.
- Muhammad, H. H.; Barcellos, M. P. Simmcast: Uma Ferramenta de Simulação para Avaliação de Protocolos Multicast. REIC. Revista Eletrônica de Iniciação Científica, v. 2, n.2, p. 1-8, 2001.
- 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.
- NIST, “National Institute of Standards and Technology Web Site”. Disponível em: <http://www.nist.gov/>. Acessado em Março de 2008.
- OPNET, “OPNET Web Site”. Disponível em: <http://www.opnet.com>. Acessado em Março de 2008.
- Rizzo, L. “Dummynet: A Simple Approach To The Evaluation of Network Protocols”, ACM Computer Communication Review, v. 27, n.1, Jan. 1997.
- Rosen, E.; Viswanathan, A.; Callon, R. Multiprotocol Label Switching Architecture. RFC 3031. Janeiro, 2001.
- SSFNet, “Scalable Simulation Framework Network Web Site”. Disponível em: <http://www.ssfnet.org/homePage.html>. Acessado em Março de 2008.
- 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 Março de 2008.