

Projeto de Redes Ópticas com Topologia em Anéis Hierárquicos Tolerantes a Falhas

Marcelo O. Lima, Elias Oliveira, Marcelo E. V. Segatto e Renato T. R. de Almeida

Resumo — Uma característica desejável em uma rede de telecomunicações é a capacidade de fornecer um serviço contínuo mesmo na presença de falhas. Neste trabalho nós apresentamos um algoritmo que utiliza a meta-heurística Busca Tabu para o projeto de topologias virtuais de redes ópticas baseadas em anéis hierárquicos tolerantes a falhas. Os resultados experimentais mostram que a nossa estratégia é capaz de obter uma boa ocupação da rede, baixo esforço computacional e possibilitar rotas alternativas no caso de falhas.

Palavras-Chave — Redes Ópticas WDM, Topologia Virtual, Heurísticas.

I. Introdução

Os avanços tecnológicos dos últimos anos, tornaram as redes ópticas de comunicação a melhor solução para a transmissão de dados em alta velocidade. O desenvolvimento da tecnologia de multiplexação por comprimento de onda (WDM) possibilitou o aumento da capacidade dos enlaces físicos, já que ela permite que diferentes comprimentos de onda ocupem o mesmo meio [1]. Devido a alta taxa de transmissão, é importante que o projeto da rede promova mecanismos de sobrevivência a eventuais falhas.

O número de nós que formam um anel tolerante a falhas (SHR - *Self-Healing Ring*) [1] é limitado pela capacidade requerida e pelo número de saltos entre dois nós quaisquer [2]. Isto torna-se importante na busca por métodos eficientes de inter-conexão entre tais anéis. Algumas características desejáveis em redes de anéis inter-conectados são: preservação da capacidade de sobrevivência dos anéis, roteamento eficiente e um mecanismo simplificado de controle da rede. Considerando estes fatores, Shi e Fonseka [2] propuseram uma arquitetura de rede em anéis hierárquicos tolerantes a falhas (HSHR - *Hierarchical Self-Healing Ring*). A arquitetura HSHR proposta consiste em um determinado número de níveis hierárquicos, cada um contendo um número de SHR's. Um nível superior é utilizado para conectar os anéis do nível imediatamente inferior, para isto é escolhido um nó, chamado *super nó*, de cada anel do nível inferior.

A topologia HSHR possui vantagens, tais como possibilitar um fácil re-roteamento de tráfego, gerenciamento relativamente simples e evolução da rede, pois um novo nó pode ser facilmente acrescentado em um anel do nível inferior.

O primeiro autor é aluno de IC, os demais são pesquisadores do Labtel (Laboratório de Telecomunicações), Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, Brasil, Campus de Goiabeiras, Av. Fernando Ferrari, s/n, Cx Postal 5011, 29060-970. <http://www.labtel.ele.ufes.br> e-mail: marcelo@labtel.ele.ufes.br, segatto@ele.ufes.br - Este trabalho é parcialmente financiado pelo CNPq.

Neste trabalho nós propomos o uso de heurísticas para o projeto de redes HSHR. Formulamos uma solução utilizando a meta-heurística *Busca Tabu* [3], baseando-se em nossos trabalhos anteriores com redes em anéis [4], [5].

II. Solução Heurística Proposta

O problema de projeto de topologia pode ser declarado como segue: tomando um conjunto de nós, e sua correspondente matriz de tráfego, é preciso conectar os nós de tal modo que resulte em um mínimo custo para a rede. Este problema pode ser formulado através de programação inteira-mista (MILP) [1], o qual pode ser resolvido por métodos exatos, ou combinando métodos de programação linear com alguma meta-heurística. Contudo não se tem a garantia de encontrar a solução ótima quando se utiliza uma meta-heurística, mas os resultados na literatura mostram o sucesso desta aproximação em muitos problemas de difícil resolução.

A rede HSHR é dimensionada definindo cada anel individualmente. O tamanho dos anéis e o número de níveis são definidos pela heurística utilizada. A demanda entre cada par de nós é suficiente para dimensionar um anel.

Nós utilizamos a conhecida meta-heurística *Busca Tabu* (BT) [3] neste trabalho. A função objetivo utilizada foi a ocupação da rede [1], porém nós também consideramos o processamento eletrônico em cada nó [6].

Na BT diferentes *movimentos* podem ser definidos a fim de determinar a vizinhança ao longo da busca. A todo tempo os movimentos que são feitos para gerar uma nova solução são incluídos em uma lista de movimentos proibidos, a *lista tabu*, onde permanecem por um determinado número de iterações. Esta lista tem a função de diversificar a busca, evitando temporariamente que a mesma retorne a soluções anteriormente visitadas. O algoritmo termina quando uma solução melhor que um definido *limitante inferior* é encontrada, ou um número máximo de iterações é alcançado, ou o limite de tempo é ultrapassado.

Para representar o problema nós definimos um vetor de valores inteiros onde cada um deles representa um nó da rede. Em nossos experimentos consideramos apenas hierarquias de anéis com dois níveis.

Consideremos uma rede com $N = \{v_1, v_2, \dots, v_n\}$, onde $v_i \in N$ é um nó desta rede. Os super nós estão entre estes nós. No entanto, nosso vetor solução é dividido em duas partes $\{S, V\}$: a parte S é um vetor de ponteiros para os nós em V , e V é uma permutação de N . Por exemplo, se $S = [0, p_1, p_2, \dots, p_k]$ dizemos que $S[0] = 0$ aponta para o super nó de um anel do nível inferior, localizado na posição 0 do vetor V , e $S[1] = p_1$ aponta para um outro super nó

em V . O tamanho de qualquer anel do nível inferior pode ser calculado pela quantidade de nós entre aqueles apontados por p_i e p_{i-1} , $p_i \in S$. A restrição de mínimo tamanho dos anéis é imposta por esta estrutura, ficando assegurado que todos os anéis do primeiro nível terão no mínimo dois nós.

Nesta estrutura o primeiro anel sempre se inicia na primeira posição do vetor V , v_0 , se estendendo até o nó imediatamente anterior àquele apontado por p_1 . O segundo anel consistirá desde o nó apontado por p_1 , até o nó imediatamente anterior àquele apontado por p_2 , e assim por diante. Desse modo, nós podemos mapear os dois níveis da topologia em anéis hierárquicos.

O movimento entre soluções, descrito anteriormente, é obtido simplesmente pela permutação dos nós no vetor solução V . Além disso, eventualmente a busca é diversificada, quando são mudados aleatoriamente os números da parte S do vetor solução. Com isso, nós alteramos a estrutura dos super nós e o tamanho dos anéis do nível inferior.

III. Resultados

Nesta seção serão apresentados os resultados numéricos obtidos para o problema de projeto de topologia virtual em anéis hierárquicos de dois níveis. O algoritmo descrito anteriormente foi aplicado as duas redes de tamanhos diferentes, a primeira com 24 nós e a segunda com 36. Ambas as redes possuem distribuição de tráfego uniforme entre todos os pares de nós.

Foi utilizada a linguagem de programação C para implementar o algoritmo, juntamente com o compilador gcc . Todos os experimentos foram realizados em uma máquina com sistema operacional Linux, processador Pentium IV de 1GHz e 1GBytes de memória.

A meta-heurística BT foi empregada no estudo da influência do tamanho dos anéis sobre a ocupação da rede e o processamento de tráfego nos nós. O processo de busca foi interrompido assim que foi encontrado um valor estável para a ocupação da rede. A figura 1 mostra a característica de convergência do algoritmo implementado. Como podemos notar, a convergência ocorre com menos de 20 minutos em ambas as redes.

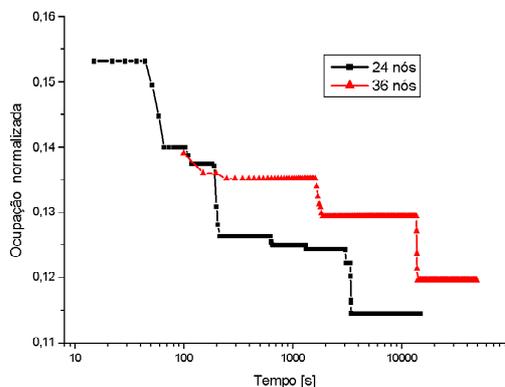


Fig. 1. Característica de convergência do algoritmo implementado para redes em anéis hierárquicos.

A influência do tamanho dos anéis foi verificada da seguinte forma: para um dado tamanho de anel no primeiro nível, o algoritmo busca o melhor valor para a ocupação da rede e, simultaneamente, calcula o processamento de tráfego nos nós, definido como a diferença entre o tráfego que chega e o que é destinado a cada nó, i.e., o tráfego que é processado pelo nó e que não é destinado a ele [6].

A ocupação normalizada é obtida pela divisão da ocupação da rede pela soma de todas as demandas de tráfego, e o processamento normalizado é obtido de maneira semelhante, porém no numerador se utiliza o processamento de tráfego nos nós. A figura 2 mostra a ocupação e o processamento normalizados em função do tamanho dos anéis. Em ambas as redes a ocupação mínima é obtida quando o tamanho dos anéis é igual a um terço do tamanho da rede. Para anéis grandes a ocupação aumenta consideravelmente. Também é importante notar que o processamento nos nós aumenta significativamente com o tamanho dos anéis.

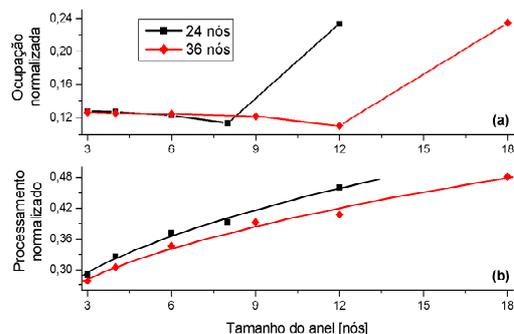


Fig. 2. (a) Ocupação e (b) Processamento normalizados, em função do tamanho do anel.

IV. Conclusões

Neste trabalho nós apresentamos uma estratégia de projeto de topologia e minimização da ocupação em redes ópticas, utilizando a meta-heurística Busca Tabu. Note que a solução ótima não era nosso principal objetivo, mas encontrar boas soluções com pouco tempo de processamento. Desenvolvemos uma codificação simples para o problema e conseguimos resolvê-lo para redes relativamente grandes, com menos de 20 minutos de processamento. Os resultados mostraram que para redes com distribuição uniforme de tráfego, existe um tamanho ótimo para os anéis. Verificamos também que o processamento de tráfego nos nós aumenta com o tamanho dos anéis.

REFERÊNCIAS

- [1] R. Ramaswami and K. N. Sivarajan, *Optical networks: A practical perspective*, Morgan Kaufmann Publishers, Inc, San Francisco, 2001.
- [2] J. Shi and J. P. Fonseka, *Hierarchical Self-Healing Rings*, IEEE/ACM Trans. Networking, 1995, Vol.3, pag.690-697.
- [3] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.
- [4] M. E. V. Segatto, Projeto de Topologias Virtuais de Redes Ópticas Tolerantes a Falhas, SBRT'04, pag.7-12, 2004.
- [5] M. O. Lima, E. Oliveira, *The Design of Hierarchical Self-Healing Rings Networks*, CONFETELE'2005, Portugal, April 2005.
- [6] R. T. R. Almeida, *Changing Paradigms on Virtual Topology and Traffic Routing Optimization*, CONFETELE'2005, Portugal, April 2005

Convergência de redes utilizando a plataforma Java – Exemplo de implementação

Gabriel Falcetti Esteca, Eduardo Martins Spina, Rodrigo C.L Rossetti, Rodrigo Filev Maia

Departamento de Engenharia da Computação e Sistemas Digitais – Escola Politécnica da USP
Av. Prof. Luciano Gualberto - Trav.3, Nº 380 – CEP – 05508-900 – São Paulo - SP

{gabriel.esteca,eduardo.spina,rodrigo.rossetti, rodrigo.maia}@poli.usp.br

Resumo - Este trabalho descreve a implementação de um sistema cuja finalidade é estabelecer a interconexão entre redes de tecnologias distintas. O principal desafio é tornar compatível a utilização de serviços entre redes síncronas e assíncronas. No trabalho é apresentado um protótipo que interconecta a Rede Pública de Telefonia Comutada, PSTN (*Public Switched Telephone Network*), e uma rede IP, através de uma plataforma implementada, em sua maior parte, sob a tecnologia JAVA. Este sistema funciona como uma caixa postal da rede telefônica, podendo-se armazenar e escutar mensagens tanto através de telefones fixos ou móveis quanto através de computadores pessoais. Neste trabalho destaca-se a implementação das interfaces de comunicação, notadamente a interface telefônica.

Palavras-chave – Convergência de redes, Java, Conversão de áudio, JNI, gateway convergente.

Abstract - This work describes an implementation of a computer system, which its purpose is to provide the convergence between existing network, considering the synchronism as the main aspects between them. The prototype proposed in this paper interconnects the Public Switched Telephone Network (PSTN), and an IP network, using the JAVA platform in order to implement the prototype. The service provided between networks is voice mail message exchanges. These messages may be stored and listened through the fixed telephone or mobile and personal computers. The main characteristic presented in this paper is about the communication interface, especially considering the PSTN interface.

Keywords – Convergence of networks, Java, Audio conversion, JNI, convergent gateway.

I. INTRODUÇÃO

Esse trabalho tem como finalidade a implementação de um sistema convergente que em princípio trabalha com dois tipos de rede: a rede pública de telefonia comutada PSTN e a rede de computadores baseadas no protocolo IP. Para atingir a meta proposta foi desenvolvido um “gateway convergente” (GC) entre as duas redes, capaz de permitir a interoperabilidade de diversos tipos de serviços, como o serviço de mensagens de áudio, conhecido também como *voice-mail*.

Usuários do GC podem enviar e receber mensagens de voz independentemente da rede em que estão, seja na rede PSTN

através de aparelhos telefônicos em geral ou na rede IP através de computadores comuns. O GC é capaz de tratar as diferenças de sincronicidade entre as redes citadas, através da implementação de regras de negócios (no próprio GC) e de repositórios de dados controlados pelo GC. [1],[2].

Este artigo está organizado nos seguintes itens para ilustrar a implementação deste protótipo de GC. O item 2 descreve sucintamente a arquitetura do sistema, o item 3 as tecnologias utilizadas. E por fim, os itens 4 e 5 descrevem respectivamente a implementação e resultados obtidos.

II. ARQUITETURA DO GATEWAY CONVERGENTE

O sistema é composto por um computador pessoal (PC) que interconecta os dois tipos de rede, como ilustrado na figura 1. Neste PC foi implementado o GC, composto de uma placa CTI (*Computer Telephone Integration*), responsável por receber e enviar dados pelo sistema telefônico. Através da placa CTI, o GC oferece um menu interativo, desenvolvido em JAVA, no qual os usuários podem optar por serviços e manipulá-los. Os dados obtidos são utilizados pelo GC para manipular o serviço utilizado pelo usuário em um servidor de mensagens (denominado *Multimedia Content Provider* - MCP) que está localizado dentro da rede IP. As mensagens geradas pelo serviço são gravadas e armazenadas até que o usuário destinatário da mensagem, independentemente da rede em que está, solicite-as. [1], [3].

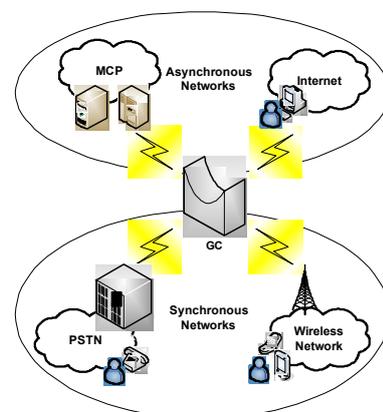


Figura 1: Arquitetura do sistema

A arquitetura proposta para a implementação do GC possui característica modular de tal forma a ser flexível para a

incorporação de outras funcionalidades para atender a mais serviços do que são oferecidos pelo protótipo atual. A figura abaixo ilustra a arquitetura implementada no GC:

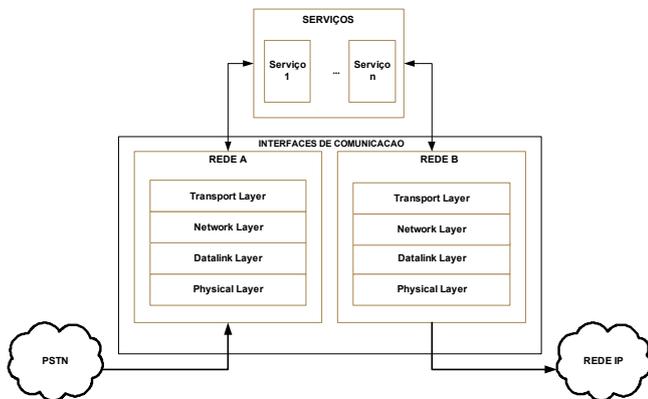


Figura 2: Arquitetura do GC

III. TECNOLOGIAS UTILIZADAS

As tecnologias utilizadas para implementar as interfaces de comunicação para o GC foram: placas CTI como interfaces para a rede PSTN e implementação de interfaces JNI para esta interface. Para a rede IP, a interface de comunicação foi Web-Services. Este padrão foi o escolhido por prover uma interface que, através de protocolos Internet, como o HTTP, permite o encapsulamento de tráfego multimídia de forma transparente ao servidor de aplicação.

A sigla CTI refere-se a sistemas que possibilitam a interação do usuário através da rede telefônica por diversas linhas, com um computador, tornando-o capaz de servir como uma central de atendimento, recebendo dados, atendendo ligações e também as redirecionando. A atuação como interface entre os canais de comunicação telefônicos é a primeira fase necessária para a convergência das redes em questão. As funções de controle de uma placa CTI são implementadas em C, através do suporte fornecido por uma API (*Application Program Interface*), de modo a permitir ao desenvolvedor o uso de todas as características da placa. Através da API, o desenvolvedor deve implementar uma biblioteca no padrão DLL (*Dynamic Link Library*) para utilização da placa CTI. Como o sistema tem sua outra parte desenvolvida em JAVA, precisamos de uma interface que permita a utilização desta DLL, e para tal a tecnologia JNI (*Java Native Interfaces*) foi utilizada. Esta é a interface para código nativo elaborada pela Sun como parte do Java Development Kit (JDK). A JNI permite aplicações Java acessarem e executarem aplicações e bibliotecas escritas em outras linguagens como C, C++ e Assembly. A JNI também permite que códigos de diversas linguagens se utilizem de aplicações JAVA. A programação da placa CTI teve que ser necessariamente desenvolvida na linguagem C, pois era somente esta linguagem que as bibliotecas da placa suportavam. [3],[4],[5].

VI. IMPLEMENTAÇÃO

Através das funcionalidades implementadas em bibliotecas fornecidas pelo fabricante da placa CTI, elaborou-se uma DLL onde todas as funções que acessavam as bibliotecas da placa foram colocadas, de forma que atendessem às especificações da interface do GC. A utilização desta DLL em um programa

JAVA foi obtida através da implementação de código nativo JNI. Estas funções que constavam na DLL foram feitas utilizando o Visual C++ da Microsoft®, já que este era o único ambiente de desenvolvimento suportado pela placa CTI. Com isso era possível a partir da aplicação desenvolvida em JAVA acessar todas as funcionalidades da placa CTI como, gravar mensagens, tocar mensagens, capturar dígitos entre outras. [3],[4],[5].

As interfaces Web-Services, utilizadas para o envio e recebimento de dados entre o servidor de mensagens e o GC foram implementadas através de componentes para web-services disponíveis para o ambiente de desenvolvimento Eclipse¹ e com a utilização do servidor de aplicação Tomcat (da *Apache Software Foundation*).

V. RESULTADOS OBTIDOS & CONSIDERAÇÕES FINAIS

As tecnologias utilizadas mostraram-se adequadas aos objetivos propostos no estudo, destacando-se principalmente a implementação das interfaces JNI para o sistema de placas CTI, pois através das pesquisas feitas para o trabalho, não foram encontradas implementações deste tipo de sistema de comunicação através das tecnologias propostas.

O GC implementado permitiu a interoperabilidade de serviços entre diferentes redes, obtendo sucesso no tratamento da questão da sincronia entre infra-estruturas de comunicação de tecnologias distintas, principalmente em se tratando de interoperabilidade de serviços.

Uma grande parte do GC foi desenvolvida em Java, onde é processada toda a interação com o usuário a partir do menu interativo citado anteriormente. A arquitetura modular torna possível a criação de módulos independentes, ou seja, a manutenção ou ampliação do sistema se torna mais simples e direta, pois não há necessidade de reprogramação dos módulos existentes, e nem de procedimentos de integração complexos, acarretando em custos menores para futuras implementações.

VI. REFERÊNCIAS

- [1] Stallings, W. **High Speed Networks and Internets: performance and Quality of Service**. New Jersey, 2 ed. Prentice Hall, 2002.
- [2] Kurose, J.F.; Ross, K.W. **Redes de Computadores e a Internet**. 1 ed. São Paulo, Addison Wesley, 2003.
- [3] Computer Telephony Integration (CTI) in a SALT Environment. Acesso em Setembro de 2004, disponível em: <http://www.intel.com/network/csp/applnots/8986an.htm>
- [4] Intel Corporation. Acesso em Setembro de 2004, disponível em: <ftp://download.intel.com/network/csp/pdf/6925ds.pdf>
- [5] Sun Technology - Chapter 5: JNI Technology. Acesso em Outubro de 2004, disponível em: <http://java.sun.com/developer/onlineTraining/Programming/JDCBook/jni.html>

¹ Software inicialmente desenvolvido pela IBM. Atualmente é um software livre.