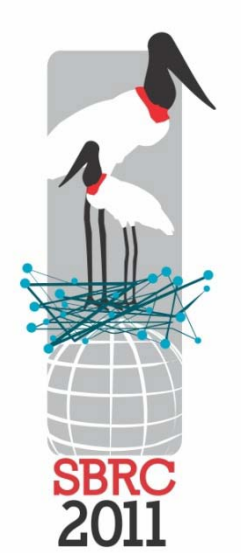
XXIX Simpósio Brasileiro
de Redes de Computadores
e Sistemas Distribuídos

30 de maio a 3 de junho de 2011
Campo Grande • Mato Grosso do Sul
http://sbrc2011.facom.ufms.br

SBRC
2011

XVI Workshop de Gerência e Operação de Redes e Serviços (WGRS 2011)

ANAIS



XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos
30 de maio a 3 de junho de 2011
Campo Grande, MS

XVI Workshop de Gerência e Operação de Redes e Serviços (WGRS)

Editora
Sociedade Brasileira de Computação (SBC)

Organizadores
Paulo André da Silva Gonçalves (UFPE)
Fábio Moreira Costa (UFG)
Ronaldo Alves Ferreira (UFMS)

Realização
Faculdade de Computação
Universidade Federal de Mato Grosso do Sul (UFMS)

Promoção
Sociedade Brasileira de Computação (SBC)
Laboratório Nacional de Redes de Computadores (LARC)

Capa: Venise Melo
Produção Editorial: Lucilene Vilela Gonçalves, Ronaldo Alves Ferreira

Cópias Adicionais:
Sociedade Brasileira de Computação (SBC)
Av. Bento Gonçalves, 9500 – Setor 4 – Prédio 43.412 – Sala 219
Bairro Agronomia – CEP 91.509-900 – Porto Alegre – RS
Fone: (51) 3308-6835
E-mail: sbc@sbc.org.br

Dados Internacionais de Catalogação na Publicação (CIP)

Workshop de Gerência e Operação de Redes e Serviços (16.: 2011 : Campo Grande, MS).

Anais / XVI Workshop de Gerência e Operação de Redes e Serviços; organizadores Paulo André da Silva Gonçalves... et al. – Porto Alegre : SBC, c2011.
189 p.

ISSN 2177-496X

1. Redes de computadores. 2. Sistemas distribuídos. I. Gonçalves, Paulo André da Silva. II. Título.

Promoção

Sociedade Brasileira de Computação (SBC)

Diretoria

Presidente
José Carlos Maldonado (USP)

Vice-Presidente
Marcelo Walter (UFRGS)

Diretor Administrativo
Luciano Paschoal Gaspary (UFRGS)

Diretor de Finanças
Paulo Cesar Masiero (USP)

Diretor de Eventos e Comissões Especiais
Lisandro Zambenedetti Granville (UFRGS)

Diretora de Educação
Mirella Moura Moro (UFMG)

Diretora de Publicações
Karin Breitman (PUC-Rio)

Diretora de Planejamento e Programas Especiais
Ana Carolina Salgado (UFPE)

Diretora de Secretarias Regionais
Thais Vasconcelos Batista (UFRN)

Diretor de Divulgação e Marketing
Altigran Soares da Silva (UFAM)

Diretor de Regulamentação da Profissão
Ricardo de Oliveira Anido (UNICAMP)

Diretor de Eventos Especiais
Carlos Eduardo Ferreira (USP)

Diretor de Cooperação com Sociedades Científicas
Marcelo Walter (UFRGS)

Promoção

Conselho

Mandato 2009-2013
Virgílio Almeida (UFMG)
Flávio Rech Wagner (UFRGS)
Silvio Romero de Lemos Meira (UFPE)
Itana Maria de Souza Gimenes (UEM)
Jacques Wainer (UNICAMP)

Mandato 2007-2011
Cláudia Maria Bauzer Medeiros (UNICAMP)
Roberto da Silva Bigonha (UFMG)
Cláudio Leonardo Lucchesi (UFMS)
Daltro José Nunes (UFRGS)
André Ponce de Leon F. de Carvalho (USP)

Suplentes – Mandato 2009-2011
Geraldo B. Xexeo (UFRJ)
Taisy Silva Weber (UFRGS)
Marta Lima de Queiroz Mattoso (UFRJ)
Raul Sidnei Wazlawick (PUCRS)
Renata Vieira (PUCRS)

Laboratório Nacional de Redes de Computadores (LARC)

Diretoria

Diretor do Conselho Técnico-Científico
Artur Ziviani (LNCC)

Diretor Executivo
Célio Vinicius Neves de Albuquerque (UFF)

Vice-Diretora do Conselho Técnico-Científico
Flávia Coimbra Delicato (UFRN)

Vice-Diretor Executivo
Luciano Paschoal Gaspary (UFRGS)

Membros Institucionais
CEFET-CE, CEFET-PR, IME, INPE/MCT, LNCC, PUCPR, PUC-RIO, SESU/MEC, UECEM UERJ, UFAM, UFBA, UFC, UFCG, UFES, UFF, UFMG, UFMS, UFPA, UFPB, UFPE, UFPR, UFRGS, UFRJ, UFRN, UFSC, UFSCAR, UNICAMP, UNIFACS, USP

Realização

Comitê de Organização

Coordenação Geral
Ronaldo Alves Ferreira (UFMS)

Coordenação do Comitê de Programa
Artur Ziviani (LNCC)
Bruno Schulze (LNCC)

Coordenação de Palestras e Tutoriais
Nelson Luis Saldanha da Fonseca (UNICAMP)

Coordenação de Painéis e Debates
José Augusto Suruagy Monteiro (UNIFACS)

Coordenação de Minicursos
Fabíola Gonçalves Pereira Greve (UFBA)

Coordenação de Workshops
Fábio Moreira Costa (UFG)

Coordenação do Salão de Ferramentas
Luis Carlos Erpen De Bona (UFPR)

Comitê Consultivo
Antônio Jorge Gomes Abelém (UFPA)
Carlos André Guimarães Ferraz (UFPE)
Francisco Vilar Brasileiro (UFCG)
Lisandro Zambenedetti Granville (UFRGS)
Luci Pirmez (UFRJ)
Luciano Paschoal Gaspary (UFRGS)
Marinho Pilla Barcellos (UFRGS)
Paulo André da Silva Gonçalves (UFPE)
Thais Vasconcelos Batista (UFRN)

Realização

Organização Local
Brivaldo Alves da Silva Jr. (UFMS)
Edson Norberto Cáceres (UFMS)
Eduardo Carlos Souza Martins (UFMS/POP-MS)
Hana Karina Sales Rubinstejn (UFMS)
Irineu Sotoma (UFMS)
Kátia Mara França (UFMS)
Luciano Gonda (UFMS)
Lucilene Vilela Gonçalves (POP-MS)
Márcio Aparecido Inácio da Silva (UFMS)
Marcos Paulo Moro (UFGD)
Massashi Emilson Oshiro (POP-MS)
Nalvo Franco de Almeida Jr. (UFMS)
Péricles Christian Moraes Lopes (UFMS)
Renato Porfírio Ishii (UFMS)

Mensagem do Coordenador Geral

Sejam bem-vindos ao XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2011) em Campo Grande, MS. É um prazer e uma distinção organizar um simpósio de tamanha relevância para a Computação no Brasil, mais ainda por ser a primeira vez que a Região Centro-Oeste tem o privilégio de sediá-lo. O SBRC é um evento anual promovido pela Sociedade Brasileira de Computação (SBC) e pelo Laboratório Nacional de Redes de Computadores (LARC). Ao longo dos seus quase trinta anos, o SBRC tornou-se o mais importante evento científico nacional em Redes de Computadores e Sistemas Distribuídos e um dos maiores da área de Informática no país.

O SBRC 2011 está com uma programação bastante rica, de qualidade diferenciada e que consiste em: 18 sessões técnicas de artigos completos que abordam o que há de mais novo nas áreas de redes de computadores e sistemas distribuídos; três sessões técnicas para apresentação de ferramentas selecionadas para o Salão de Ferramentas; cinco minicursos, com quatro horas de duração, sobre temas atuais; três palestras e três tutoriais com pesquisadores de alto prestígio internacional; e três painéis sobre assuntos de interesse da comunidade. Além dessas já tradicionais atividades do simpósio, ocorrerão em paralelo oito workshops: XVI Workshop de Gerência e Operação de Redes e Serviços (WGRS), XII Workshop da Rede Nacional de Ensino e Pesquisa (WRNP), XII Workshop de Testes e Tolerância a Falhas (WTF), IX Workshop em Clouds, Grids e Aplicações (WCGA), VII Workshop de Redes Dinâmicas e Sistemas P2P (WP2P), II Workshop de Pesquisa Experimental da Internet do Futuro (WPEIF), I Workshop on Autonomic Distributed Systems (WoSIDA) e I Workshop de Redes de Acesso em Banda Larga (WRA).

O desafio de organizar um evento como o SBRC só pode ser cumprido com a ajuda de um grupo especial. Eu tive a felicidade de contar com a colaboração de inúmeras pessoas ao longo desta jornada. Meus sinceros agradecimentos aos membros dos Comitês de Organização Geral e Local por realizarem um trabalho de excelente qualidade e com muita eficiência, a qualidade da programação deste simpósio é fruto do trabalho dedicado dessas pessoas. Sou grato a Faculdade de Computação da UFMS por ter sido uma facilitadora ao longo de todo o processo de organização, desde a nossa proposta inicial até o fechamento da programação. Gostaria de agradecer, também, ao Comitê Gestor da Internet no Brasil (CGI.br), às agências governamentais de fomento e aos patrocinadores por reconhecerem a importância do SBRC e investirem recursos financeiros fundamentais para a realização do evento. Com o apoio financeiro recebido, foi possível manter os custos de inscrição baixos e oferecer um programa social de alta qualidade.

Em nome do Comitê Organizador, agradeço a todos os participantes pela presença em mais esta edição do SBRC e desejo uma semana produtiva, agradável e com estabelecimento de novas parcerias e amizades.

Ronaldo Alves Ferreira
Coordenador Geral do SBRC 2011

Mensagem do Coordenador de Workshops do SBRC 2011

Os workshops são uma parte tradicional do que hoje faz do SBRC o principal evento da área no país, sendo responsáveis por atrair uma parcela cada vez mais expressiva de participantes para o Simpósio todos os anos. O SBRC 2011 procurou manter essa tradição, com a realização de workshops já considerados parte do circuito nacional de divulgação científica nas várias subáreas de Redes de Computadores e Sistemas Distribuídos, como o WTF (Workshop de Testes e Tolerância a Falhas), o WCGA (Workshop em *Clouds*, *Grids* e Aplicações), o WGRS (Workshop de Gerência e Operação de Redes e Serviços) e o WP2P (Workshop de Redes Dinâmicas e Sistemas P2P). Incluímos também nesta lista de iniciativas bem sucedidas o WRNP (Workshop da Rede Nacional de Ensino e Pesquisa), que cumpre o importantíssimo papel de fazer a ponte entre as comunidades técnica e científica da área.

Como novidade em 2011, e reconhecendo o surgimento e o fortalecimento de novas linhas de pesquisa de expressiva importância dentro da comunidade brasileira de Redes e Sistemas Distribuídos, procuramos incentivar a criação de novos workshops dentro do Simpósio. Foi com esse intuito que introduzimos pela primeira vez no SBRC a chamada aberta de workshops, por meio da qual membros da comunidade foram convidados a submeter propostas de workshops inéditos para realização em conjunto com o SBRC 2011. Em resposta à chamada, recebemos nove propostas de alta qualidade, das quais oito foram aceitas e seus respectivos proponentes convidados a organizarem os workshops no SBRC em Campo Grande. Das oito propostas aceitas, cinco tratavam dos workshops já tradicionais acima mencionados, e uma referia-se à segunda edição de um workshop mais recentemente criado, mas que teve sua primeira edição realizada de forma muito bem sucedida no SBRC 2010, o WPEIF (Workshop de Pesquisa Experimental da Internet do Futuro). As outras duas propostas foram resultado direto da chamada aberta de workshops e resultaram na adição de dois novos eventos ao leque do SBRC, o WRA (Workshop de Redes de Acesso em Banda Larga) e o WoSiDA (*Workshop on Autonomic Distributed Systems*), ambos com ótima aceitação pela comunidade, a julgar pelos números de submissões de trabalhos recebidos.

Esperamos que 2011 seja mais um ano de sucesso para os workshops do SBRC, em particular para aqueles criados nesta edição do Simpósio, e para que eles continuem contribuindo como importantes fatores de agregação para os avanços promovidos pela comunidade científica da área de Redes e Sistemas Distribuídos no Brasil.

Aproveitamos para agradecer o inestimável apoio recebido de diversos membros da comunidade e, em particular, da Organização Geral do SBRC 2011.

A todos, um excelente SBRC em Campo Grande!

Fábio M. Costa
Coordenador de Workshops do SBRC 2011

Mensagem do Coordenador do WGRS

O Workshop de Gerência e Operação de Redes e Serviços (WGRS) é um evento promovido pela Sociedade Brasileira de Computação (SBC) e tem como objetivo prover um espaço para a apresentação de pesquisas e atividades relevantes na área de gerenciamento e operação de redes e serviços. Dessa forma, o evento contribui para a integração da comunidade brasileira de pesquisadores e profissionais atuantes nessa área. Além disso, o WGRS também atua como um fórum para a apresentação e discussão de soluções utilizadas por provedores e usuários de sistemas de gerenciamento de redes.

Nesta 16ª edição, a comunidade continuou a prestigiar o WGRS com um excelente número de submissões. Ao todo, 34 artigos foram submetidos para serem avaliados. O Comitê de Programa foi constituído por 30 pesquisadores. Esse comitê contou ainda com o apoio de avaliadores externos para a condução do processo de avaliação de artigos. Cada artigo recebeu 3 avaliações independentes e, ao final do processo de avaliação dos artigos submetidos, tivemos ao todo 102 revisões. Dentre os artigos submetidos, o Comitê de Programa optou por indicar os 13 melhores classificados para publicação e apresentação no evento, representando uma taxa de aceitação de 38%.

Nos Anais, encontram-se os textos completos dos artigos selecionados. Tanto a programação técnica do evento quanto os Anais estão organizados em quatro sessões: (i) Plataformas de Gerenciamento, Auto-gerenciamento e Auto-configuração, (ii) Desempenho e Qualidade de Serviço , Aprovisionamento de Redes e Planejamento de Capacidade , (iii) Gerenciamento de Serviços e Aplicações e (iv) Gerenciamento de Redes Móveis, Sem fio e de Sensores.

Gostaria de expressar o meu agradecimento aos membros do Comitê de Programa e aos revisores por terem aceitado participar voluntariamente dessa empreitada. Agradeço-os também pela competência e dedicação na realização do processo de avaliação e seleção dos artigos. Gostaria de expressar também os meus agradecimentos ao coordenador geral do SBRC 2011, Ronaldo Alves Ferreira (UFMS), e ao coordenador de Workshops do SBRC 2011, Fábio Moreira Costa (UFG), pela disponibilidade e orientações providas ao longo do processo. Agradeço também aos ex-coordenadores do WGRS, Aldri Luiz dos Santos (UFPR), Anelise Munaretto (UTFPR) e Mauro Fonseca (PUCPR) pela oportunidade e confiança ao me convidarem para essa empreitada. Finalmente, não poderia de deixar de expressar os meus agradecimentos aos autores que submeteram os seus trabalhos e que nos motivam a realizar anualmente este evento de interesse, visibilidade e sucesso crescentes.

Saúdo a todos os participantes do XVI Workshop de Gerência e Operação de Redes e Serviços com os votos de um excelente workshop e de uma excelente estadia em Campo Grande!

Paulo André da S. Gonçalves
Coordenador do WGRS 2011

Comitê de Programa do WGRS

Aldri Luiz dos Santos (UFPR)
Anelise Munaretto (UTFPR)
Antônio Tadeu Azevedo Gomes (LNCC)
Artur Ziviani (LNCC)
Bruno Schulze (LNCC)
Carlos Westphall (UFSC)
Célio Vinicius Neves de Albuquerque (UFF)
Edmundo Madeira (UNICAMP)
Fátima Duarte-Figueiredo (PUC Minas)
Horacio Oliveira (UFAM)
Joaquim Celestino Júnior (UECE)
José Augusto Suruagy Monteiro (UNIFACS)
José Marcos Nogueira (UFMG)
Jussara Almeida (UFMG)
Kelvin Dias (UFPE)
Linnyer Ruiz (UEM)
Lisandro Zambenedetti Granville (UFRGS)
Luciano Paschoal Gaspary (UFRGS)
Luis Henrique Costa (UFRJ)
Luiz Nacamura Júnior (UTFPR)
Luiz Henrique Andrade Correia (UFLA)
Manoel Camillo de Oliveira Penna Neto (PUCPR)
Marcelo Rubinstein (UERJ)
Marcial Fernandez (UECE)
Mauro Fonseca (PUCPR)
Michele Nogueira Lima (UFPR)
Nazareno Andrade (UFCG)
Paulo André da Silva Gonçalves (UFPE)
Raimir Holanda (UNIFOR)
Ronaldo Ferreira (UFMS)

Revisores do WGRS

Aldri Luiz dos Santos (UFPR)
Anelise Munaretto (UTFPR)
Antônio Tadeu Azevedo Gomes (LNCC)
Artur Ziviani (LNCC)
Bruno Almeida da Silva (UFPE)
Bruno Schulze (LNCC)
Carlos Westphall (UFSC)
Célio Vinicius Neves de Albuquerque (UFF)
Ewerton Salvador (UFMG)
Fátima Duarte-Figueiredo (PUC Minas)
Felipe Henriques (Centro Federal de Educação Tecnológica Celso Suckow da Fonseca)
Horacio Oliveira (UFAM)
Igor Monteiro Moraes (UFF)
Joaquim Celestino Júnior (UECE)
Jorge Lima de Oliveira Filho (UNICAMP)
José Augusto Suruagy Monteiro (UNIFACS)
José Geraldo Ribeiro Junior (UFRJ)
José Marcos Nogueira (UFMG)
Juliano Wickboldt (UFRGS)
Jussara Almeida (UFMG)
Kelvin Dias (UFPE)
Lisandro Zambenedetti Granville (UFRGS)
Luciano Chaves (UNICAMP)
Luciano Paschoal Gaspary (UFRGS)
Luis Henrique Costa (UFRJ)
Luiz Fernando Bittencourt (UNICAMP)
Luiz Henrique Andrade Correia (UFLA)
Luiz Nacamura Júnior (UTFPR)
Manoel Camillo de Oliveira Penna Neto (PUCPR)
Marcelo Rubinstein (UERJ)
Marcial Fernandez (UECE)
Mauro Fonseca (PUCPR)
Michele Nogueira Lima (UFPR)
Nazareno Andrade (UFCG)
Neumar Malheiros (UNICAMP)
Paulo André da Silva Gonçalves (UFPE)
Rafael Esteves (UFRGS)
Raimir Holanda (UNIFOR)
Ronaldo Ferreira (UFMS)
Revisores Anônimos

Sumário

XVI Workshop de Gerência e Operação de Redes e Serviços

♦

Sessão Técnica 1

Plataformas de Gerenciamento, Auto-gerenciamento e Auto-configuração

Um Servidor de Máquinas Virtuais Adaptado a Múltiplas Pilhas de Protocolos*

Rafael dos Santos Alves, Miguel Elias Mitre Campista e Luís Henrique Maciel Kosmalski Costa

[1]Grupo de Teleinformática e Automação – PEE/COPPE – DEL/POLI
Universidade Federal do Rio de Janeiro (UFRJ)

{santos,miguel,luish}@gta.ufrj.br

Resumo. *A Internet atual alcançou grande sucesso devido a características como o argumento fim-a-fim e a pilha TCP/IP. Entretanto, essa arquitetura dificulta a adição do suporte à mobilidade, à segurança e à qualidade de serviço. Nesse sentido, este trabalho apresenta um servidor de máquinas virtuais, uma aplicação capaz de criar e controlar roteadores virtuais em diferentes estações físicas. O servidor proposto facilita a implantação de soluções pluralistas baseadas em virtualização de computadores, permitindo a criação de redes virtuais sob demanda e a interação com os usuários dos recursos de rede. O servidor é implementado utilizando o conceito de* Web services *e um protótipo operacional com estações Xen existe atualmente.*

Abstract. *The Internet success is based on characteristics such as the end-to-end argument and the TCP/IP protocol stack. Nevertheless, this architecture hinders the addition of mobility, security and quality of service support. In this scenario, this work, proposes a virtual machine server, an application able to create and control virtual routers in different physical machines. The proposed server facilitates the deployment of pluralist solutions based on computer virtualization, allowing the creation of virtual network on demand and also the interaction between users and network resources. The server is implemented using Web services and a prototype with Xen stations is currently operational.*

1. Introdução

O enorme sucesso da Internet é consequência de decisões fundamentais tomadas em sua origem como a inteligência nas extremidades e a comutação de pacotes. Inicialmente, os requisitos fundamentais eram tornar a rede tolerante a falhas e prover suporte à heterogeneidade dos nós e dos serviços. O protocolo IP (*Internet Protocol*) surgiu com o papel de interconectar toda a rede e prover conectividade fim-a-fim, ao mesmo tempo em que mantinha o núcleo da rede simples, reservando aos nós das extremidades as tarefas mais complexas. Essas escolhas foram tomadas em momentos nos quais não se previa nós tão heterogêneos quanto alguns estáticos e outros móveis. Ainda, que a heterogeneidade fosse tamanha que alguns nós ou serviços não seriam confiáveis, ou que alguns serviços poderiam demandar tratamento diferenciado da rede como, por exemplo, requisitos de qualidade de serviço.

*Este trabalho foi realizado com recursos do FUNTTEL, FINEP, CNPq, CAPES e FAPERJ.

Muitos dos desafios da Internet atual estão relacionados com o atendimento de requisitos como mobilidade, segurança e qualidade de serviço em uma rede criada há décadas, sem prejudicar o seu funcionamento. Nesse sentido, muitos projetos vêm sendo desenvolvidos para propor uma nova Internet, também conhecida como Internet do Futuro. Um dos pioneiros foi o projeto *clean-slate* [Clark et al., 2004] que propôs recriar a Internet com base nas experiências adquiridas ao longo dos anos de operação. Há ainda propostas na direção oposta, que partem do pressuposto de que uma mudança drástica como essa é inviável economicamente e que a Internet deve continuar sendo adaptada à sua constante evolução [Rexford e Dovrolis, 2010]. Essa última abordagem, entretanto, tem que lidar com a multiplicidade de requisitos muitas vezes conflitantes em apenas uma arquitetura e correr o risco de não atender satisfatoriamente nenhum usuário. A solução única pode ser muito complexa e jamais alcançada. Assim, uma possibilidade que vem ganhando atenção é a solução pluralista na qual múltiplas redes virtuais com requisitos diferentes podem ser executadas em paralelo compartilhando o mesmo meio físico virtualizado. Para isso, o conceito de virtualização de máquinas é estendido para redes através de softwares hipervisores que gerenciam o acesso ao hardware de roteadores entre múltiplos roteadores virtuais[1]. Entende-se, portanto, que uma rede virtual é um conjunto de roteadores virtuais e os enlaces entre eles.

Este trabalho propõe a criação de um servidor de máquinas virtuais, uma aplicação capaz de criar, remover, migrar e gerenciar roteadores virtuais em diferentes máquinas físicas. Tal servidor facilita a implantação de soluções pluralistas baseadas em virtualização de computadores. Para isso, o servidor possui um repositório de imagens de máquinas que pode ser utilizado para transferir e instanciar remotamente roteadores virtuais personalizados. Tal característica permite a interação entre usuários confiáveis, administradores ou sistemas inteligentes, e a rede. Uma possibilidade tangível para oferecer interação aos usuários ainda mantendo certo controle é disponibilizar um conjunto de imagens pré-configuradas para que os usuários possam escolher entre elas conforme as suas necessidades. Uma vez que as imagens tenham sido transferidas e instanciadas, o servidor virtual ainda pode gerenciar a migração de roteadores ao vivo e destruir nós selecionados.

No sistema proposto, toda comunicação de usuários com o Servidor de Máquinas Virtuais ocorre através de mensagens do protocolo SOAP (*Simple Object Access Protocol*) [Box et al., 2000] sobre HTTP (*HyperText Transfer Protocol*). O Servidor de Máquinas Virtuais, por sua vez, se comunica com as máquinas físicas que hospedam as máquinas virtuais indicadas nas mensagens SOAP através de uma API de gerenciamento como, por exemplo, a `Libvirt` [Libvirt, 2011] ou a `XenAPI` [Citrix, 2011]. É importante observar que o serviço é totalmente transparente, o que significa que o servidor pode ser atualizado sem alterar a interface com seus clientes. Os resultados obtidos demonstram a operacionalidade do servidor através de provas de conceito realizadas em uma rede de testes no Laboratório do Grupo de Teleinformática e Automação (GTA) da UFRJ.

O restante deste trabalho está dividido da seguinte forma. A Seção 2 apresenta os trabalhos relacionados e descreve o modelo da arquitetura no qual está inserido o Servidor de Máquinas Virtuais proposto neste trabalho. A Seção 3 apresenta o Servidor de

[1]O termo máquinas virtuais e roteadores virtuais são usados neste trabalho de maneira intercalada.

Máquinas Virtuais, sua arquitetura e implementação. A Seção 4 apresenta o protótipo para testes do servidor implantado no laboratório do GTA. A Seção 5 apresenta os experimentos realizados e a Seção 6 conclui este trabalho.

2. Trabalhos Relacionados

Muitos trabalhos na literatura investigam soluções para a Internet do Futuro. Esses trabalhos podem ser divididos em duas categorias: puristas e pluralistas [Moreira et al., 2009]. A abordagem purista procura atender aos diversos requisitos da Internet através da utilização de uma única pilha de protocolos, que deve ser flexível o suficiente para atender as diferentes demandas. Por outro lado, a abordagem pluralista utiliza pilhas de protocolos em paralelo de forma que cada pilha atenda a um conjunto específico de requisitos.

Entre as arquiteturas que utilizam a abordagem purista pode-se citar a arquitetura baseada em papéis [Clark et al., 2004]. Nesse modelo não existe a multiplicidade de camadas evitando, de forma inerente, o problema comum dos protocolos da Internet que é a violação de camadas. Para substituir as camadas, módulos que os autores chamam de papéis são utilizados para permitir a modularização do desenvolvimento dos protocolos. O diferencial importante quando comparado ao modelo em camadas é a inexistência de níveis hierárquicos entre os papéis. É importante notar que os papéis devem ser blocos bem conhecidos e padronizados, permitindo assim a criação de serviços bem definidos.

A arquitetura DONA (*Data-Oriented Network Architecture*) [Koponen et al., 2007] parte do princípio que a Internet passou de centrada em estação para centrada em dados. Portanto, os usuários não estão mais interessados em quem ou onde podem adquirir conteúdo, mas na obtenção desse conteúdo em tempo hábil. Um dos principais representantes do conceito centrado em estação é o sistema de resolução de nomes, o DNS (*Domain Name System*), que responde a solicitações de endereços IP recebendo como parâmetro o nome do servidor de destino. Ao invés do DNS, a arquitetura DONA propõe o uso de primitivas *anycast* baseadas em nomes inseridos acima da camada de rede.

Entre as arquiteturas pluralistas pode-se destacar a arquitetura CABO (*Concurrent Architectures are Better than One*) [Feamster et al., 2007] que propõe utilizar máquinas virtuais de modo que em cada rede virtual uma configuração ou até mesmo uma pilha de protocolos distinta seja utilizada. Dessa forma, múltiplas redes, com diferentes características estão disponíveis em um mesmo substrato físico. A principal motivação por trás da arquitetura CABO é permitir que os ISPs (*Internet Service Providers*) ofereçam serviços diferenciados aos seus clientes, o que atualmente não é possível já que nenhum provedor de serviço possui roteadores em todo o percurso fim-a-fim entre todos os usuários da Internet. Para isso, é proposta a separação entre provedores de serviço e provedores de infraestrutura. Neste caso, os provedores de infraestrutura forneceriam recursos computacionais e de rede para os provedores de serviço, através de acordos comerciais. Um provedor de serviço, por sua vez, pode utilizar recursos de diferentes provedores de infraestrutura e com isso, construir um caminho fim-a-fim de roteadores, possibilitando a oferta de serviços diferenciados aos usuários finais.

O projeto franco-brasileiro Horizon [Horizon Project, 2011] tem por objetivo desenvolver uma arquitetura para a Internet baseada nos conceitos do pluralismo e de inte-

ligência intra e inter-redes virtuais. Nesse projeto, para cada pilha de protocolos existe um conjunto de máquinas virtuais instanciadas ao longo da rede, executando essa pilha. Além disso, para garantir um desempenho mínimo para as redes propõe-se a criação de um plano de pilotagem. Esse plano tem por objetivo sensoriar e atuar na rede. Através de observações colhidas por um conjunto diverso de equipamentos de medidas e do conhecimento acumulado no plano de conhecimento, o plano de pilotagem deve decidir se alguma mudança de configuração deve ser realizada na rede. Em caso positivo, o plano de pilotagem deve acionar os procedimentos de software necessários para que a tarefa seja realizada. O plano de pilotagem é responsável por coordenar os planos de gerenciamento, de controle e de virtualização através, por exemplo, da alteração de parâmetros de configuração de um protocolo de roteamento. Os resultados das alterações implementadas são colhidas como conhecimento a ser disseminado pela rede.

Uma tecnologia importante em muitas das abordagens pluralistas é a virtualização de computadores [Popek e Goldberg, 1974, Egi et al., 2008]. Através da separação dos recursos computacionais, diferentes pilhas de protocolos podem conviver em um mesmo ambiente. Vale notar também que essa tecnologia pode ser utilizada para a construção de redes experimentais para realização de testes de ambas as abordagens, como já é feito em projetos como o PlanetLab [Chun et al., 2003] e GENI [GENI, 2011].

Este trabalho propõe um servidor de máquinas virtuais voltado para a abordagem pluralista que tem como principal objetivo a automatização do processo de gerenciamento de máquinas e, em última instância, de redes virtuais. Vale notar que este trabalho encontra-se no escopo do projeto Horizon, embora, sob o ponto de vista funcional, pudesse ser utilizado por outras propostas, como por exemplo, o CABO, que depende de uma infraestrutura muito semelhante a necessária pelo projeto Horizon, formada por máquinas com suporte à virtualização que oferecem suporte a diferentes redes virtuais.

3. O Servidor de Máquinas Virtuais

O Servidor de Máquinas Virtuais provê um conjunto de serviços Web. Podem figurar como clientes do Servidor de Máquinas Virtuais, por exemplo, um administrador da rede, ou um sistema autônomo de gerenciamento, em última instância, qualquer agente interessado em monitorar ou alterar recursos da rede. Além das tarefas de criação de máquinas virtuais e de redes virtuais, serviços básicos do servidor, um conjunto de serviços extras foi adicionado ao servidor tornando-o um controlador de redes virtuais. Dessa forma, esse servidor pode ser utilizado como ferramenta para a realização das tarefas definidas pelo administrador. Por exemplo, caso o administrador perceba um gargalo de desempenho de uma rede virtual em um determinado nó, uma requisição para o aumento de recursos (memória, CPU etc.) pode ser feita através de um serviço provido pelo servidor.

De acordo com a proposta deste trabalho, cada um dos roteadores físicos é equipado com alguma tecnologia, como o Xen [Barham et al., 2003] ou VMWare [VMware, 2011], que os torna capazes de hospedar sistemas virtualizados. A função básica do Servidor de Máquinas Virtuais é criar sob demanda máquinas virtuais nos nós físicos da rede e configurar seus enlaces de forma a garantir que a rede esteja ativa e conectada após o término de toda operação.

3.1. Arquitetura do Servidor

A Figura 1 apresenta o cenário no qual o Servidor de Máquinas Virtuais é utilizado no contexto do Projeto Horizon. Na mesma figura, também podem ser observados o plano de pilotagem proposto no projeto Horizon e uma máquina com suporte a virtualização. Sempre que o plano de pilotagem requisitar um dos serviços providos pelo Servidor de Máquinas Virtuais, uma mensagem deve ser enviada utilizando o protocolo SOAP sobre HTTP para o servidor. O Servidor de Máquinas Virtuais, por sua vez, comunica-se com as máquinas físicas que hospedam as máquinas virtuais indicadas na mensagem SOAP e que serão acessadas através de uma API de gerenciamento como, por exemplo, a `Libvirt` ou a `XenAPI`. É importante observar que, para o plano de pilotagem, a forma como o serviço vai ser realizado é totalmente transparente, permitindo que o Servidor de Máquinas Virtuais seja atualizado, mudando a implementação de seus serviços, sem alterar a interface com seus clientes.

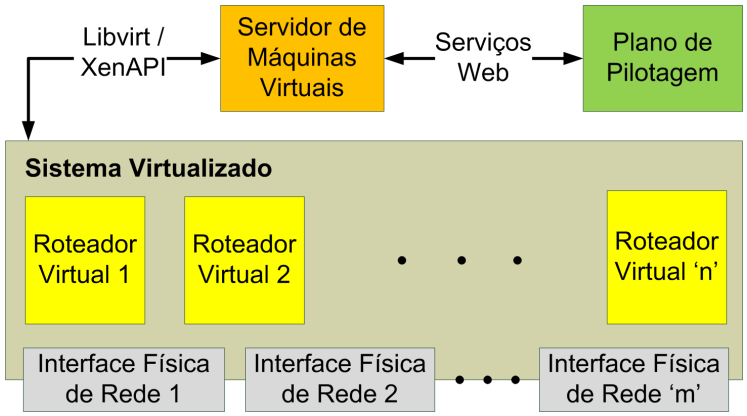


Figura 1. Arquitetura do Servidor de Máquinas Virtuais proposto.

Na maioria dos serviços oferecidos, o cliente deve esperar uma mensagem de retorno informando o resultado da operação. Em caso de falha, um relatório com os motivos da falha é enviado ainda por serviço Web ao cliente.

É importante notar que não existe qualquer restrição conceitual ao tipo de protocolo de comunicação instalado na máquina virtual. A limitação que pode existir é do ponto de vista da implementação. Um determinado protocolo pode não estar disponível para o sistema operacional desejado, por exemplo.

A Figura 2 mostra como o sistema evolui ao longo do tempo. Nesse exemplo, assume-se que o serviço requisitado é o de criação de uma máquina virtual (`createVirtualMachine`). Na primeira interação do sistema (Mensagem 1), o plano de pilotagem - o cliente - envia uma mensagem SOAP contendo uma requisição do serviço. Ao receber a mensagem, o Servidor de Máquinas Virtuais identifica o sistema Xen indicado na mensagem e envia (Mensagem 2), através de uma biblioteca de gerenciamento de sistema virtualizado, uma requisição para a criação de uma máquina virtual com as características definidas na mensagem que o plano de pilotagem enviou.

Em seguida, o sistema Xen tenta criar a máquina virtual solicitada e envia o resultado da operação ao Servidor de Máquinas Virtuais (Mensagem 3). O termo "criar máquina virtual" aqui utilizado refere-se ao processo de definição de um arquivo que será

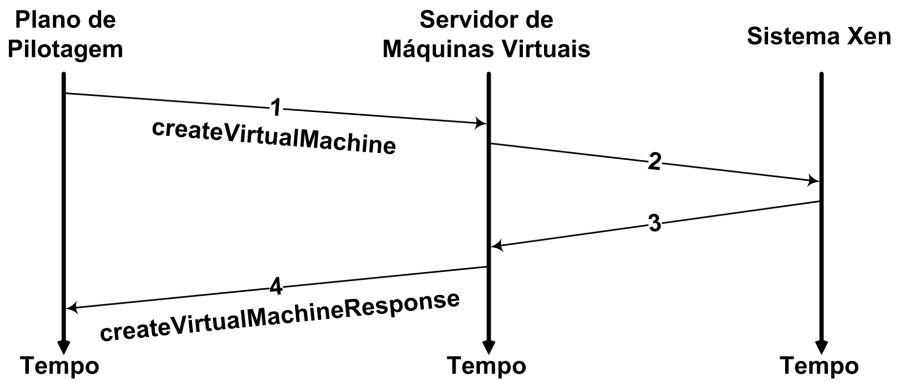


Figura 2. Evolução do sistema no tempo.

utilizado como disco virtual para a máquina hóspede e outros parâmetros, como tamanho da memória, número de interfaces de rede etc. Finalmente, o servidor cria uma mensagem XML com o resultado da operação e a envia ao plano de pilotagem (Mensagem 4).

Um ponto importante a ser ressaltado é a localização do arquivo que será utilizado como disco virtual. Esse arquivo deve conter o sistema operacional desejado, com as funções requeridas pelo serviço. Por exemplo, caso a máquina virtual requisitada seja um roteador IPv4, o arquivo deve conter um sistema operacional Linux, com suporte a IPV4 e o protocolo de roteamento RIP (*Routing Information Protocol*). Esse arquivo pode ser armazenado em locais diferentes. O Servidor de Máquinas Virtuais pode também armazenar diferentes imagens de sistemas operacionais e transferí-las para as estações físicas no momento em que máquinas virtuais forem criadas.

3.2. Implementação

Um protótipo do Servidor de Máquinas Virtuais foi implementado e implantado dentro do laboratório do GTA com o objetivo de realizar análises de prova de conceito. Os serviços atualmente disponíveis estão descritos na Tabela 1.

O serviço de criação de máquinas virtuais (`createVirtualMachine`) utiliza os parâmetros passados pela requisição do serviço para criar um nó virtual no nó físico da rede especificado. Vale destacar que esse serviço tem duas variações principais. Na primeira, o Servidor de Máquinas Virtuais envia o arquivo utilizado como disco virtual para o servidor físico que irá abrigar a nova máquina virtual. Na segunda, o disco deve existir em algum local acessível pelo servidor físico que abriga o roteador virtual. O serviço de criação de redes virtuais (`createVirtualNetwork`) deve criar um conjunto de nós virtuais em máquinas físicas da rede. Além disso, para que os nós criados formem uma rede, deve-se realizar o mapeamento entre a interface física indicada e a interface virtual criada, além da configuração dos endereços de rede.

Quando o serviço de criação de máquinas virtuais é utilizado transferindo o disco virtual, os detalhes referentes ao tipo de máquina virtual a ser utilizada devem ser definidos. Alguns serviços foram criados com o objetivo de mostrar aos clientes os tipos de sistema operacional disponíveis no Servidor de Máquinas Virtuais. O serviço `getAvailableOSes` apresenta os sistemas operacionais disponíveis (Linux, Windows etc.), já o serviço `getAvailableArch` apresenta as arquiteturas disponíveis e final-

Tabela 1. Serviços oferecidos pelo Servidor de Máquinas Virtuais.

Serviço	Descrição
createVirtualMachine	criação de máquinas virtuais
createVirtualNetwork	criação de redes virtuais
destroyVirtualMachine	exclusão de máquinas virtuais
getAvailableArch	arquiteturas disponíveis no servidor
getAvailableKernelVersions	versões de kernel disponíveis no servidor
getAvailableOSes	sistemas operacionais disponíveis no servidor
getPhysicalServerStatus	informações sobre um servidor físico
getRegisteredNodes	lista de nós registrados
getVirtualMachineSchedulerParameters	consulta parâmetros de escalonamento de máquina virtual
getVirtualMachineSchedulerType	consulta de tipo de escalonador de máquina virtual
getVirtualMachineStatus	informações sobre determinado domínio virtual
migrateVirtualMachine	migração de máquinas virtuais
registerNodes	registro de novo nó físico na rede
sanityTest	teste de sanidade sobre funcionamento do servidor
setVirtualMachineSchedulerParameters	ajuste de parâmetros de escalonamento de máquina virtual
shutdownVirtualMachine	desligamento de máquina virtual

mente o serviço `getAvailableKernelVersions` apresenta as versões de kernel disponíveis.

Além das tarefas básicas do Servidor de Máquinas Virtuais, tarefas adicionais foram implementadas. O serviço `destroyVirtualMachine` pode ser utilizado para destruir uma máquina virtual, por exemplo, quando a rede para a qual ela foi criada não é mais necessária. Em alguns casos a máquina virtual deverá ser desligada para ser religada depois de um tempo, por exemplo para economia de energia. O serviço `shutdownVirtualMachine` pode ser utilizado com esse intuito. Os serviços `getPhysicalServerStatus` e `getVirtualMachineStatus` têm por objetivo obter algumas informações gerais, como memória e número de processadores, acerca de uma máquina física e de uma virtual, respectivamente. Essas informações podem ser utilizadas pelo plano de pilotagem no processo de tomada de decisões. O serviço de migração (`migrateVirtualMachine`) pode ser utilizado para mover uma máquina virtual de um nó físico para outro, por exemplo, quando um determinado nó físico encontra-se sobrecarregado. Esse serviço utiliza o mecanismo de migração padrão do Xen [Clark et al., 2005], porém outros mecanismos podem ser utilizados, como por exemplo o mecanismo de migração com separação de planos em [Pisa et al., 2010].

Nos sistemas operacionais multitarefa, escalonadores de CPU são utilizados para dividir recursos de processamentos entre os processos. De forma semelhante, o Xen utiliza um escalonador de CPU para compartilhar recursos entre as máquinas virtuais. Entre os principais escalonadores utilizados no Xen pode-se citar o *Credit Scheduler* [Citrix, 2007], o SEDF (*Simple Earliest Deadline First*) [Leslie et al., 1996] e o BVT (*Borrowed Virtual Time*) [Duda e Cheriton, 1999]. Atualmente, o escalonador padrão do Xen é o *Credit Scheduler*. O tipo de escalonador utilizado pela máquina virtual pode ser consultado pelo serviço `getVirtualMachineSchedulerType`. Além disso, os parâmetros de escalonamento de uma máquina virtual podem ser consultados e alterados pelos serviços `getVirtualMachineSchedulerParameters` e `setVirtualMachineSchedulerParameters`, respectivamente.

Inicialmente, o Servidor de Máquinas Virtuais não tem conhecimento sobre as máquinas físicas às quais tem acesso. Para garantir que o Servidor de Máquinas Virtuais tenha conhecimento desses nós, o serviço `registerNodes` permite que um determinado nó seja registrado no servidor para futura administração, ou seja, o nome, a chave pública e os endereços IP são enviados ao servidor que armazena essas informações localmente. Os nós registrados podem ser consultados pelo serviço `getRegisteredNodes`.

O protótipo foi implementado utilizando a linguagem de programação Java. A biblioteca `Libvirt` [Libvirt, 2011], versão 0.7.5 foi utilizada para a realização de tarefas administrativas e a biblioteca `Axis2` [Perera et al., 2006] em sua versão 1.5.1 para a construção dos serviços Web. O servidor Web utilizado foi o `Tomcat` [Apache, 2011], versão 6. Completando o ambiente de desenvolvimento, a IDE (*Integrated Development Environment*) utilizada foi o `NetBeans` [Oracle, 2011a] versão 6.7.1. Adotou-se como plataforma de virtualização o Xen. Essa plataforma de virtualização possui grande apoio da comunidade acadêmica, o que a torna mais confiável, já que um grande número de pesquisadores e usuários comuns tem utilizado essa tecnologia ao redor do mundo [Egi et al., 2007, Clark et al., 2005, Cherkasova et al., 2007]. Além disso, o Xen é distribuído sob uma licença de código livre, permitindo que alterações em seu funcionamento sejam propostas.

A linguagem de programação Java [Oracle, 2011b] é uma linguagem orientada a objetos com suporte aos principais sistemas operacionais disponíveis. Além disso, conta com uma grande quantidade de bibliotecas publicamente disponíveis que foi um dos fatores determinantes para a escolha dessa linguagem para o protótipo. Além disso, o suporte a serviços Web em Java contaram pontos a favor nessa decisão. Finalmente, um programa Java é, em princípio, multiplataforma o que permite que ele seja executado em qualquer estação desde que esta possua uma máquina virtual Java.

A `Libvirt` é uma biblioteca para gerenciamento de sistemas virtualizados de código livre. A biblioteca foi originalmente desenvolvida em C e atualmente provê suporte para um grande conjunto de linguagens, a destacar Java e Python. Os recursos de gerenciamento da `Libvirt` são genéricos, ou seja, funções comuns à maioria dos sistemas de virtualização, por exemplo, Xen, VMware, OpenVZ, QEMU etc., são providas. Essa generalidade da biblioteca permite que mesmo no caso de alteração da plataforma de virtualização, a maior parte do código do Servidor de Máquinas Virtuais possa ser reaproveitada. Essa propriedade é altamente desejável para tornar o Servidor de Máquinas Virtuais útil em cenários mais amplos do que seria se comparado ao uso de uma biblioteca de administração específica para sistemas Xen.

A biblioteca `Axis2` desenvolvida pela *Apache Foundation* implementa o protocolo SOAP. É importante observar que essa biblioteca também é publicada sob uma licença de código livre. Atualmente, a Axis2 está implementada em C e em Java, sendo a última a implementação utilizada neste protótipo.

Caso alguma tarefa atualmente não disponível seja necessária, a adição de um novo serviço é simples e não afeta o funcionamento dos outros serviços previamente disponíveis. Cada serviço no Servidor de Máquinas Virtuais é implementado como um método em sua classe principal (`VirtualMachineServer`). Todo serviço deve ser

implementado como um método público que recebe um objeto da classe `OMElement` (*Object Model Element*) e retorna outro objeto da classe `OMElement`. Essa classe é oferecida pela biblioteca Axis2 e tem por objetivo armazenar um elemento XML, ou seja, depois de transformado em uma *string*, um objeto `OMElement` torna-se uma *tag* de uma mensagem XML. Neste caso, o elemento recebido como parâmetro é o conteúdo de uma mensagem SOAP, e o `OMElement` retornado será também o conteúdo da mensagem SOAP enviada como resposta pelo Servidor de Máquinas Virtuais.

3.3. Acesso ao Servidor de Máquinas Virtuais

Com o objetivo de facilitar a criação de sistemas de software que acessem o Servidor de Máquinas Virtuais, uma classe foi desenvolvida. Para cada serviço oferecido, a classe `HorizonXenClient` possui um método para a criação do conteúdo da mensagem. A Listagem 1 apresenta a API oferecida por essa classe.

Listagem 1. API oferecida pela classe `HorizonXenClient`.

```
public OMElement createVirtualMachinePayload(String phyServer, String
    vmName, String vmIP, String vmRAM);
public OMElement createVirtualNetworkPayload(Vector<String> phyServers,
     Vector<String> VMNames, Vector<String> IPs, Vector<String> RAMs,
    Vector<String> netInterface);
public OMElement destroyVirtualMachinePayload(String phyServer, String
    vmName);
public OMElement getAvailableArch();
public OMElement getAvailableKernelVersions();
public OMElement getAvailableOSes();
public OMElement getPhysicalServerStatusPayload(String phyServer);
public OMElement getRegisteredNodesPayload();
public OMElement getVirtualMachineSchedulerParametersPayload(String
    phyServer, String VMName);
public OMElement getVirtualMachineSchedTyplerePayload(String phyServer,
     String VMName);
public OMElement getVirtualMachineStatusPayload(String phyServer,
    String vmName);
public OMElement migrateVirtualMachinePayload(String sourcePhyServer,
    String destPhyServer, String vmName, String live);
public OMElement registerNodesPayload(Vector<PhysicalServer> phyServers
    );
public OMElement sanityTestPayload(String testString);
public OMElement setVirtualMachineSchedulerParametersPayload(String
    phyServer, String VMName, String Weight, String Cap);
public OMElement shutdownVirtualMachinePayload(String phyServer, String
     vmName);
```

Observa-se que todos os métodos retornam um objeto da classe `OMElement` que será utilizado como conteúdo de mensagens SOAP. As ocorrências `vmName` referem-se ao nome esperado para a máquina virtual. Esse nome será o nome acessível através dos recursos de administração de sistemas virtualizados e para interações futuras com o Servidor de Máquinas Virtuais. O parâmetro `phyServer` refere-se ao nome DNS externamente acessível do nó físico ou o endereço IP desse nó. Os parâmetros `vmIP` e `vmRAM` apontam, respectivamente, o endereço IP e o tamanho da memória RAM desejados para o novo domínio virtual.

Existem ainda parâmetros específicos para a função de migração de máquinas virtuais: `sourcePhyServer` e `destPhyServer` definem, respectivamente, os nós físicos de origem e de destino do domínio virtual a ser migrado; o parâmetro `live`, que pode receber os valores `true` ou `false`, define se a migração deve ser realizada ao vivo, ou seja, sem interrupção do funcionamento do domínio virtual. Para isso, as páginas de memória utilizadas pela execução da máquina virtual são armazenadas e transferidas para a máquina de destino para que a execução retorne no mesmo estado que estava antes da transferência.

Alguns serviços podem atuar sobre um conjunto de máquinas físicas e virtuais. Nesses casos os parâmetros esperados são vetores e a semântica é similar aos casos já apontados. Existe ainda o parâmetro `testString` do método de teste de sanidade. Esse parâmetro define a *string* que formará o corpo da mensagem de teste e que será retornada pelo servidor, caso o servidor esteja funcionando corretamente. Com relação ao serviço de alteração de parâmetros do escalonador, os parâmetros `Weight` e `Cap` são referentes aos parâmetros do *Credit Scheduler* [Citrix, 2007] utilizado pelo Xen.

É importante observar que a utilização da classe `HorizonXenClient` não é obrigatória. O cliente pode ser construído sem fazer uso dessa biblioteca. Não existe sequer limitação quanto à linguagem de programação, já que a utilização de um serviço Web permite essa flexibilidade. O único requisito é a necessidade de utilização do protocolo SOAP e de que o conteúdo da mensagem seja um XML válido com os campos esperados pelo servidor.

4. Protótipo

Para a realização de testes, um protótipo foi implantado no laboratório do Grupo de Teleinformática e Automação da UFRJ. Esse protótipo serve como prova de conceito para o Servidor de Máquinas Virtuais. A Tabela 2 apresenta os computadores utilizados e suas respectivas funções.

Tabela 2. Computadores no protótipo e suas configurações.

Computador	Arquitetura	Kernel	Processador	Memória
vms	i386	2.6.30-2	Core 2 2.13 GHz	2 GB
xen1	i386	2.6.32-5-xen	Celeron 2.8 GHz	3 GB
xen2	amd64	2.6.32-5-xen	Core 2 Duo 2.53 GHz	4 GB
xen3	i386	2.6.32-5-xen	Pentium 4 HT 3.4 GHz	2 GB

A Figura 3 representa a topologia da rede de testes. Todas as máquinas podem ser acessadas através do roteador que as conecta à Internet. A comunicação entre o Servidor de Máquinas Virtuais, hospedado na máquina `vms`, e as outras máquinas, ocorre através desse roteador. As estações `xen1`, `xen2` e `xen3` são máquinas que executam o Xen como software de suporte à virtualização e são operadas pelo Servidor de Máquinas Virtuais. No estado atual, o mecanismo de migração de máquinas virtuais do Xen é utilizado. Nesse mecanismo é necessário que o disco virtual esteja em algum local acessível para as duas máquinas físicas que participam do processo de migração. Essa limitação vem sendo estudada por alguns trabalhos na literatura. Por exemplo, o trabalho de [Mattos et al., 2011] apresenta uma solução para essa limitação

através de uma técnica de virtualização híbrida combinando as plataformas Xen e Openflow [McKeown et al., 2008].

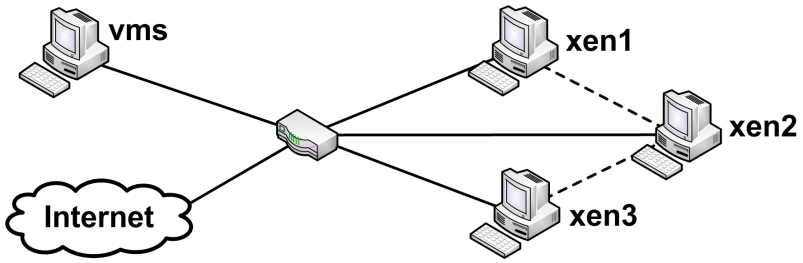


Figura 3. Topologia do *testbed*.

Todas as estações executam o sistema operacional Linux Debian. As versões de kernel e de arquiteturas utilizadas são descritas na Tabela 2. Com o objetivo de facilitar a autenticação do Servidor de Máquinas Virtuais, um mecanismo de SSH (*Secure Shell*) sem senha foi configurado entre o Servidor de Máquinas Virtuais e as máquinas. Mais especificamente, a autenticação é realizada através da utilização de chaves públicas configuradas previamente [Corp., 2003].

5. Experimentos e Observações

Experimentos foram realizados para avaliar algumas métricas de desempenho do Servidor de Máquinas Virtuais proposto. A partir da classe `HorizonXenClient`, dois clientes foram desenvolvidos para o Servidor de Máquinas Virtuais. O primeiro, um cliente JAR (*Java ARchive*) permite a requisição de serviços a partir da linha de comandos. O segundo um cliente JSP (*JavaServer Pages*) permite a requisição de serviços a partir de uma página Web.

O protótipo apresentado na Seção 4 foi utilizado para os experimentos, que têm por objetivo avaliar o consumo de recursos por alguns serviços do Servidor de Máquinas Virtuais. O cliente JAR foi utilizado nos experimentos sempre a partir de uma máquina externa ao protótipo, ou seja, além das quatro máquinas apresentadas no protótipo, uma quinta máquina assumiu o papel de cliente nos experimentos. Foram realizadas trinta rodadas experimentais para cada serviço e valores médios e de desvio padrão associados são apresentados.

Na Tabela 3 encontram-se os dados referentes à criação de máquinas virtuais. Para isso, duas possibilidades existem: transferência da imagem do servidor para as máquinas Xen e posterior inicialização da máquina virtual ou inicialização de máquina virtual já na máquina Xen. Nos experimentos, as duas variações do serviço (com ou sem transferência de disco virtual) foram avaliadas. Para os experimentos de criação com transferência de disco, duas métricas foram avaliadas: o tempo de execução do serviço para o cliente, que inclui o tempo de comunicação entre o cliente e o Servidor de Máquinas Virtuais, a transferência da imagem entre o Servidor de Máquinas Virtuais e a máquina Xen e o tempo de inicialização da máquina virtual; e o tempo de transferência de disco isoladamente.

Como visto, a transferência de disco tem impacto relevante no tempo total para a criação da máquina virtual. Entretanto, com ela não existe como pré-requisito a presença da imagem nos elementos de rede. Essa possibilidade oferece maior liberdade para que o

cliente possa criar sob demanda a sua máquina virtual personalizada. Além disso, ao comparar os tempos entre as máquinas `xen1` e `xen3` observa-se que a melhor configuração (memória e processamento maiores) da máquina `xen1` leva a um melhor desempenho desta máquina. Esse comportamento repete-se na comparação entre as máquinas `xen1` e `xen2`, onde a última possui menores tempos para o cliente, nas duas variações do serviço de criação de máquinas virtuais, e para a transferência de discos.

Tabela 3. Tempos na criação de máquinas virtuais (em segundos).

	Com transferência de disco		Sem transferência de disco
Computador	**Cliente**	**Transferência de disco**	**Cliente**
xen1	43,77 $\pm$ 1,35	35,47 $\pm$ 0,41	6,86 $\pm$ 0,25
xen2	42,20 $\pm$ 1,62	35,21 $\pm$ 0,13	5.72 $\pm$ 0,37
xen3	46,62 $\pm$ 5,80	36,52 $\pm$ 4,99	7,05 $\pm$ 1,74

A Tabela 4 apresenta o consumo de recursos de processamento no cliente e no Servidor de Máquinas Virtuais para as duas variações do serviço de criação de máquinas virtuais. Pode-se observar que os recursos de CPU exigidos no cliente são pequenos. A semelhança do processamento utilizado nos diferentes cenários corrobora o baixo consumo de recursos. Isso possibilita a utilização de clientes em dispositivos com baixa capacidade de processamento, como celulares, ou ainda a utilização de um sistema autônomo, no qual os clientes podem ser agentes móveis, por exemplo.

Tabela 4. Processamento utilizado (em segundos).

Computador	Com transferência de disco	Sem transferência de disco
xen1	0,82 $\pm$ 0,03	0,79 $\pm$ 0,01
xen2	0,79 $\pm$ 0,01	0,79 $\pm$ 0,01
xen3	0,81 $\pm$ 0,02	0,79 $\pm$ 0,01

Tanto o servidor como os clientes implementados, além da documentação de uso e instalação do Servidor de Máquinas Virtuais podem ser encontrados no seguinte sítio Web `http://www.gta.ufrj.br/~santos/vms`.

6. Conclusão e Trabalhos Futuros

Neste trabalho foi desenvolvido um servidor de máquinas virtuais adaptado a diferentes pilhas de protocolos. O servidor aqui apresentado é uma importante ferramenta no desenvolvimento de novas propostas de protocolos de comunicação para lidar com a complexidade e a multiplicidade de requisitos que ora se apresentam para a Internet.

O servidor foi implementado utilizando a linguagem de programação Java. Um conjunto de serviços está atualmente disponível e pode ser acessado com ajuda da classe de apoio desenvolvida. Essa classe, de uso opcional, foi desenvolvida também em Java e pode ser utilizada como forma de redução do tempo de desenvolvimento de clientes para o Servidor de Máquinas Virtuais.

Como trabalhos futuros, espera-se ampliar o número de serviços oferecidos pelo servidor, tais como a oferta de estatísticas mais detalhadas sobre os estados das máquinas físicas e virtuais da rede, tornando-o, na prática, um controlador de máquinas virtuais.

Além disso, o protótipo deve ser ampliado permitindo a exploração de um cenário maior do que o experimentado até o momento.

Referências

Apache (2011). Apache tomcat. http://tomcat.apache.org/. Acessado em março de 2011.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. e Warfield, A. (2003). Xen and the art of virtualization. Em *ACM Symposium on Operating Systems Principles (SOSP)*, pp. 164–177.

Box, D., Ehnebuske, D., Kakivaya, G., Mendelsohn, A. L. N., Nielsen, H. F., Thatte, S. e Winer, D. (2000). Simple object access protocol (SOAP) 1.1. http://www.w3.org/TR/2000/NOTE-SOAP-20000508/. Acessado em março de 2011.

Cherkasova, L., Gupta, D. e Vahdat, A. (2007). Comparison of the three CPU schedulers in Xen. *SIGMETRICS Performance Evaluation Review*, 35(2):42–51.

Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M. e Bowman, M. (2003). Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12.

Citrix (2007). Credit-based CPU scheduler. http://wiki.xensource.com/xenwiki/Credit-Scheduler. Acessado em abril de 2011.

Citrix (2011). Xen management API project. http://wiki.xensource.com/xenwiki/XenApi. Acessado em abril de 2011.

Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I. e Warfield, A. (2005). Live migration of virtual machines. Em *Symposium on Networked Systems Design & Implementation (NSDI)*, pp. 273–286.

Clark, D., Braden, R., Sollins, K., Wroclawski, J., Katabi, D., Kulik, J., Yang, X., Faber, T., Falk, A., Pingali, V., Handley, M. e Chiappa, N. (2004). New Arch: Future generation Internet architecture. Relatório técnico, MIT Laboratory for Computer Science and International Computer Science Institute (ICSI).

Corp., S. C. S. (2003). *SSH Secure Shell for Servers Version 3.2.9 - Administrator's Guide*. SSH Communications Security.

Duda, K. J. e Cheriton, D. R. (1999). Borrowed-virtual-time (BVT) scheduling: supporting latency-sensitive threads in a general-purpose scheduler. Em *Proceedings of the ACM symposium on Operating systems principles (SOSP)*, pp. 261–276.

Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Huici, F. e Mathy, L. (2008). Towards high performance virtual routers on commodity hardware. Em *ACM CoNEXT*.

Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Mathy, L. e Schooley, T. (2007). Evaluating xen for router virtualization. Em *International Conference on Computer Communications and Networks (ICCCN)*, pp. 1256–1261.

Feamster, N., Gao, L. e Rexford, J. (2007). How to lease the Internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64.

GENI (2011). Exploring networks of the future. http://www.geni.net. Acessado em março de 2011.

Horizon Project (2011). A new Horizon to the Internet. http://www.gta.ufrj.br/horizon. Acessado em abril de 2011.

Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S. e Stoica, I. (2007). A data-oriented (and beyond) network architecture. Em *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pp. 181–192.

Leslie, I., McAuley, D., Black, R., Roscoe, T., Barham, P., Evers, D., Fairbairns, R. e Hyden, E. (1996). The design and implementation of an operating system to support distributed multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1280–1297.

Libvirt (2011). The virtualization API. http://libvirt.org/. Acessado em março de 2011.

Mattos, D., Fernandes, N. C. e Duarte, O. C. M. B. (2011). Xenflow: Um sistema de processamento de fluxos robusto e eficiente para migração em redes virtuais. Em *Simpósio Brasileiro de Redes de Computadores (SBRC)*. Aceito para publicação.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. e Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *SIGCOMM Computer Communications Review*, 38(2):69–74.

Moreira, M. D. D., Fernandes, N. C., Costa, L. H. M. K. e Duarte, O. C. M. B. (2009). *Minicursos do Simpósio Brasileiro de Redes de Computadores (SBRC)*, capítulo Internet do Futuro: Um Novo Horizonte, pp. 1–59. SBC, Recife, PE.

Oracle (2011a). NetBeans. http://netbeans.org/. Acessado em março de 2011.

Oracle (2011b). Oracle technology network for java developers. http://www.oracle.com/technetwork/java/index.html. Acessado em abril de 2011.

Perera, S., Herath, C., Ekanayake, J., Chinthaka, E., Ranabahu, A., Jayasinghe, D., Weerawarana, S. e Daniels, G. (2006). Axis2, middleware for next generation web services. Em *International Conference on Web Services (ICWS)*, pp. 833–840.

Pisa, P., Fernandes, N., Carvalho, H., Moreira, M., Campista, M., Costa, L. e Duarte, O. (2010). OpenFlow and Xen-based virtual network migration. Em Pont, A., Pujolle, G. e Raghavan, S., editors, *Communications: Wireless in Developing Countries and Networks of the Future*, volume 327 of *IFIP Advances in Information and Communication Technology*, pp. 170–181. Springer Boston.

Popek, G. J. e Goldberg, R. P. (1974). Formal requirements for virtualizable third generation architectures. *Communications of the ACM*, 17(7):412–421.

Rexford, J. e Dovrolis, C. (2010). Future Internet architecture: clean-slate versus evolutionary research. *Communications of the ACM*, 53(9):36–40.

VMware (2011). VMware virtualization software for desktops, servers and virtual machines for public and private cloud solutions. http://www.vmware.com/. Acessado em março de 2011.

Sistema Multiagentes para Autogerenciamento Distribuído de Falhas em Redes Virtuais

Milton A. Soares Jr., Edmundo R. M. Madeira

[1]Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
Campinas – SP – Brasil

milton@lrc.ic.unicamp.br, edmundo@ic.unicamp.br

Abstract. *New proposals for the evolution of the Internet, as diversification of architectures and providers, may increase the complexity of a network whose proper operation is essential, given its importance to society and to global economy. A new theoretical framework has emerged to deal with the complexity of the networks through self-management. In this paper, the concepts of autonomic networks were applied in a virtualized environment through a multi-agent system, and experiments were performed with a focus on self-management failure, or self-healing, of virtual networks.*

Resumo. *Novas propostas para a evolução da Internet, como a diversificação de arquiteturas e provedores, podem aumentar a complexidade de uma rede cujo bom funcionamento é fundamental, dada sua importância para a sociedade e para a economia global. Um novo arcabouço teórico tem surgido para lidar com a complexidade das redes através do autogerenciamento. Neste trabalho, os conceitos das redes autonômicas foram aplicados em um ambiente virtualizado através de um sistema multiagentes e foram realizados experimentos com foco no autogerenciamento de falhas, ou autocura, de redes virtuais.*

1. Introdução

Com o avanço das plataformas de virtualização e o aumento do poder computacional do *hardware* foi possível imaginar o uso de virtualização na Internet para habilitar o pluralismo de arquiteturas e a separação de provedores de serviços e de infraestrutura [Turner and Taylor 2005]. Essa é uma estratégia *clean slate* para a Internet do futuro com o intuito de superar problemas na arquitetura atual [Anderson et al. 2005]. Embora a abordagem *one-size-fits-all* do TCP/IP, que mantem a complexidade nas extremidades da rede, tenha sido responsável pelo sucesso da Internet nos últimos 30 anos, muitos serviços atuais demandam garantias de banda, atraso, *jitter*, segurança, e até mesmo de funcionamento, que a arquitetura não é capaz de oferecer. Com as redes virtuais é possível implementar arquiteturas de rede especializadas para atender a um determinado tipo de serviço sobre uma mesma infraestrutura física.

O plano de gerenciamento tem um papel fundamental nesse cenário. O provedor de infraestrutura precisa gerenciar as redes virtuais tanto para garantir os serviços contratados pelo provedor de serviços, quanto para obter uma boa utilização dos recursos físicos. A segmentação da rede de maneira ótima, robusta e segura é um desafio devido à complexidade do problema [Zhu and Ammar 2006, Yu et al. 2008, Houidi et al. 2008]. As ferramentas de gerenciamento atuais são centralizadas e dependentes de intervenção

humana, o que são fontes de defeitos, e não são capazes de lidar com a heterogeneidade de tecnologias.

Novas abordagens de gerenciamento baseadas em ferramentas da inteligência artificial e dos sistemas cognitivos tem surgido em resposta à crescente complexidade das telecomunicações. Na Internet, é proposta a inclusão de novos planos no modelo de referência para acomodá-las [Clark et al. 2003, Gaïti et al. 2006]. Entre as novas abordagens, a das redes autonômicas [Dobson et al. 2006] propõe lidar com a complexidade habilitando as redes a se autogerenciarem. Tarefas mais simples de configuração, otimização, recuperação de falhas e segurança podem ser realizadas pela própria rede, sem intervenção humana, liberando os administradores para tarefas mais complexas como a definição das políticas e objetivos da rede.

A utilização de máquinas virtuais em *data-centers* para consolidação de servidores, que utilizam técnicas de migração e *backup*, é uma prática comum nos dias atuais. Entretanto, roteadores possuem necessidades específicas que devem ser levadas em consideração. No plano de dados, os roteadores são responsáveis pelo transporte de dados, que é uma aplicação sensível a falhas. Problemas dessa natureza geram perdas que podem afetar os serviços que utilizam esse transporte. No plano de controle, os roteadores são responsáveis pelos algoritmos de roteamento que autoconfiguram as rotas ao custo de alguma latência e sobrecarga.

A proposta deste trabalho é o desenvolvimento de um sistema multiagentes para o autogerenciamento distribuído de redes virtuais. Sua arquitetura é baseada no ciclo autonômico e na base de conhecimento. Nós aplicamos o sistema em um cenário de falhas em que ele deve realizar o diagnóstico e o reparo de maneira autonômica. Também estudamos diferentes mecanismos de recuperação e seu impacto no tempo de reparo e consequentemente nas perdas de pacotes. Este trabalho está inserido no contexto do projeto Horizon [Horizon 2010]. O objetivo do projeto é a definição de uma nova arquitetura para a Internet do futuro baseada no pluralismo de arquiteturas, através de redes virtuais, e no plano de pilotagem, que inclui mecanismos inteligentes para adaptar protocolos a mudanças no ambiente.

Sistemas multiagentes também são utilizados em [Houidi et al. 2010] para manter os contratos e SLAs (*service level agreements*) em eventos de falhas de recursos e degradação severa de desempenho. Os agentes formam grupos baseados na similaridade dos nós físicos que são gerenciados por eles. A função de dissimilaridade também é utilizada para escolher onde serão instanciados os nós virtuais que tiveram problemas. Todas as ações dos agentes são executadas após o diagnóstico da falha. Nós implementamos e testamos situações em que o planejamento é antecipado, através de cálculos e difusões de informações realizados periodicamente. Nós também avaliamos a recuperação de roteadores virtuais através de arquivos que armazenam um estado anterior da memória do roteador para reduzir a latência da inicialização do sistema e da configuração das rotas.

Em [Marquezan et al. 2010] uma arquitetura de gerenciamento distribuída é apresentada com o objetivo de auto-organizar as redes virtuais para manter uma boa utilização dos recursos físicos. O algoritmo para auto-organização utilizado é baseado no ciclo de controle autonômico. Ele monitora os enlaces e busca minimizar a carga de tráfego na rede através da migração de nós virtuais. Nós implementamos o ciclo de controle au-

tonômico nos agentes do sistema de autogerenciamento distribuído para a autocura das redes virtuais.

O restante deste texto foi organizado da seguinte forma: a Seção 2 contém os conceitos básicos de redes virtuais e redes autonômicas que são utilizados neste trabalho. Na Seção 3, a arquitetura do sistema multiagentes desenvolvida neste trabalho é apresentada. Os detalhes de implementação do sistema no *testbed* para a execução dos experimentos são descritos na Seção 4. A Seção 5 apresenta resultados dos experimentos e os discute. A conclusão e os trabalhos futuros são mostrados na Seção 6.

2. Conceitos básicos

Esta seção aborda os conceitos básicos utilizados neste trabalho. O primeiro é o conceito de redes virtuais, as quais o sistema de autogerenciamento distribuído se destina. Outro conceito importante é o de redes autonômicas que possuem os fundamentos para o desenvolvimento do sistema multiagentes.

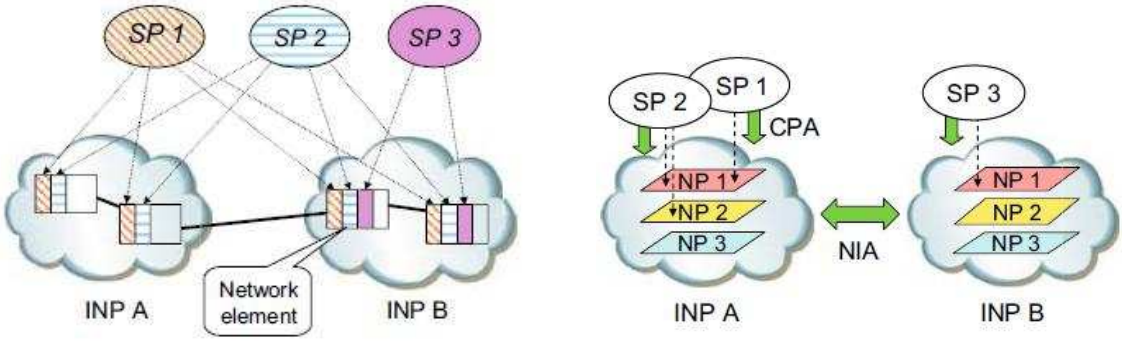
2.1. Redes virtuais

Virtualização é uma técnica empregada em vários cenários de redes de computadores. Nas VLANs (*virtual local area network*) é possível a criação de enlaces virtuais em uma rede local. VPNs (*virtual private network*) são utilizadas para inserir máquinas remotas dentro de uma rede através de um circuito virtual seguro. Com a técnica de tunelamento é possível encaminhar pacotes por redes de diferentes tecnologias. Em todos esses cenários, apenas o enlace está sendo virtualizado. Existem também as redes *overlay* que interligam aplicações e usuários, como as redes P2P (*peer-to-peer*).

As redes virtuais possuem uma abordagem diferente. A ideia por trás delas é a separação lógica dos recursos físicos de uma rede, incluindo dispositivos de comutação. Cada rede virtual pode ter seus próprios protocolos, algoritmos e configurações, de acordo com os objetivos dos serviços que estão sendo executados sobre ela, e deve ter isolamento, isto é, a operação das redes virtuais não deve causar interferência entre si, embora elas estejam sobre a mesma infraestrutura. As redes virtuais podem ser implementadas através de roteadores virtuais ou de separação dos planos de dados e de controle.

No primeiro caso, o roteador físico tem que ser capaz de fornecer e isolar *fatias* de seus recursos aos roteadores virtuais. Ele também tem que multiplexar e demultiplexar pacotes entre o mundo físico e o virtual. Roteadores virtuais podem ser criados via software através de plataformas conhecidas para virtualização de máquinas como o KVM [Habib 2008] e o Xen [Barham et al. 2003]. O KVM transforma o Linux em um *hypervisor* de virtualização total através de um módulo do kernel. Ele é baseado no emulador de máquinas Qemu e roda em arquiteturas x86 com a tecnologia Intel VT ou AMD-V. O Xen é um *hypervisor* de paravirtualização, o que, em geral, oferece um desempenho melhor, mas requer sistemas operacionais modificados ou específicos para ele nas máquinas virtuais.

Na segunda abordagem, há a separação do plano de dados e do plano de controle. Um controlador centralizado é responsável por funções de roteamento, controle de acesso, gerenciamento de tráfego etc. O controlador edita as tabelas de encaminhamento dos comutadores da rede e faz a inspeção dos fluxos que chegam. O OpenFlow [McKeown et al. 2008] é o principal produto dessa abordagem. Ele fornece o *software* do



(a) CABO (b) AGAVE

Figura 1. Propostas para o pluralismo de arquiteturas na Internet do futuro com base na virtualização [Boucadair et al. 2009].

controlador e um protocolo para comunicação segura entre o controlador e os comutadores.

As redes virtuais podem habilitar o pluralismo de arquiteturas na Internet do futuro. Uma das propostas é a do projeto CABO (*concurrent architectures are better then one*) [Feamster et al. 2007]. Ela visa a reestruturação da Internet através da separação de provedores de infraestrutura e de serviços. Nesse contexto, um provedor de serviços contrataria provedores de infraestrutura ao longo da comunicação fim-a-fim para o fornecimento de redes virtuais, que ficariam sob seu controle. Essa proposta está representada na Figura 1(a).

Outra proposta é a do projeto AGAVE (*a lightweight approach for viable end-to-end IP-based QoS services*) [Boucadair et al. 2009] que introduz os conceitos de NPs (*network planes*) e PIs (*parallel internets*). Os provedores de Internet criariam NPs internamente para atender determinada classe de tráfego, tendo o controle sobre eles. Os serviços com requisitos de QoS semelhantes seriam atendidos por este NP de uma maneira agregada. Através de interconexões horizontais com outros provedores que possuem NPs semelhantes formariam as PIs e, assim, o serviço poderia ser diferenciado de fim-a-fim. A Figura 1(b) apresenta a proposta do projeto AGAVE.

2.2. Redes Autonômicas

O manifesto da computação autonômica [IBM 2001] enfatiza que a complexidade tem sido um grande obstáculo para o desenvolvimento da área de TI, pois ela está crescendo além da habilidade humana de gerenciá-la. A computação autonômica é bio-inspirada no sistema nervoso autônomo que é responsável pela regulação do corpo de acordo com mudanças ambientais sem a necessidade do controle consciente. O objetivo dela é reduzir ou eliminar a intervenção humana no gerenciamento através das propriedades de autoconfiguração, auto-otimização, autocura e autoproteção.

A arquitetura de [Kephart and Chess 2003] é baseada em gerentes autonômicos distribuídos. O gerente autonômico realiza um ciclo de controle sobre o elemento gerenciado e utiliza uma base de conhecimento para armazenar as informações coletadas. A cada laço ele realiza as atividades de monitoramento, análise, planejamento e execução, o que alimenta o próximo ciclo através da base de conhecimento.

O cenário do aumento da complexidade, heterogeneidade, ubiquidade, conectividade e integração é o motivo que leva à necessidade de desenvolvimento de redes autonômicas [Braga et al. 2006]. As redes autonômicas são baseadas nos princípios da computação autonômica. Elas devem ser capazes de realizar o autogerenciamento a partir de políticas de alto-nível definidas pelos administradores ou inferidas através do conhecimento da aplicação. Essa segunda abordagem merece destaque pela propriedade de autoconhecimento baseado em técnicas de aprendizagem de máquina e ciência do contexto.

Para o autogerenciamento é preciso que a rede autonômica seja capaz de se autoconfigurar, auto-otimizar, autocurar e autodefender. Essas propriedades podem ser obtidas com a inclusão de gerentes autonômicos nos elementos de rede para executarem essas tarefas. O sistema de autogerenciamento deve ser distribuído, com cada gerente autonômico responsável por uma parte dos recursos da rede que é o seu elemento gerenciado. Isso evita a criação de um ponto único de falhas e permite maior escalabilidade.

Os gerentes devem atuar de maneira autonômica, mas compartilhar objetivos comuns. A atuação individual dos gerentes autonômicos sobre seus elementos gerenciados deve propiciar um ciclo autonômico maior para o autogerenciamento da rede de acordo com suas políticas e seus objetivos.

O FOCALE [Strassner 2007] é uma arquitetura autonômica para gerenciamento de redes. Ela também baseia-se em gerentes autonômicos sobre os recursos gerenciados, porém, eles não possuem um ciclo de controle estático. O ciclo pode ser definido pelo agente em tempo de execução, baseado no contexto e nas políticas de alto-nível. A arquitetura também define o uso de uma camada MTBL (*model-based translation layer*) para possibilitar a utilização de equipamentos legados no sistema de gerenciamento. Para a adaptação, a arquitetura utiliza técnicas de aprendizado e cognição para comparar, baseado em modelos e ontologias, se o comportamento atual é o mais adequado ou se ele deveria ser substituído. A arquitetura FOCALE foi aplicada como estudo de caso nos projetos *Beyond 3G Networks* e *Motorola's Seamless Mobility*.

3. Proposta de Arquitetura do Sistema Multiagentes

A modelagem orientada a agentes é um paradigma interessante para o desenvolvimento de sistemas distribuídos de autogerenciamento. Agentes são entidades autônomas que observam o ambiente e atuam sobre ele, podendo ter algum nível de cognição e se comunicar com outros agentes. Eles podem fazer o papel do gerente autonômico de um recurso da rede, seu elemento gerenciado. Portanto, o sistema para autogerenciamento de redes virtuais proposto neste trabalho será baseado em uma arquitetura multiagentes.

Esse sistema deve realizar a recuperação de falhas das redes virtuais. As falhas nos roteadores virtuais podem ocorrer por problemas neles próprios, nos nós físicos ou até mesmo nos enlaces físicos. No primeiro caso, o agente responsável pelo roteador virtual deve diagnosticar e avisar os demais sobre a falha. Nos outros casos, o agente também irá falhar e, portanto, os agentes vizinhos devem diagnosticar a falha.

No sistema multiagentes para autogerenciamento de redes viruais há apenas um tipo de agente que atua nos nós da rede física. Os agentes são responsáveis pelo monitoramento dos recursos dos nós físicos e virtuais: cpu, memória, armazenamento, interfaces

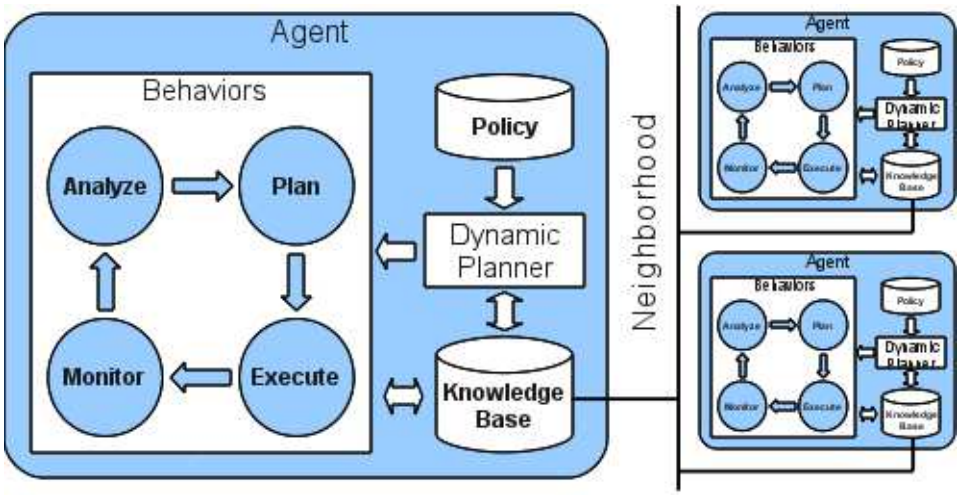


Figura 2. Arquitetura do gerente autonômico.

de rede etc. Eles também devem controlar os roteadores virtuais e instanciá-los em caso de falhas.

A arquitetura dos agentes é baseada nos comportamentos, na base de conhecimento, nas políticas e no DP (*dynamic planner*), como mostra a Figura 2. O sensoriamento, a cognição e a atuação dos agentes são realizados pelos comportamentos. Foram definidos quatro comportamentos: Monitoramento, Análise, Planejamento e Execução, para realizar atividades distintas da tarefa de autogerenciamento de falhas. Nossos agentes executam esses comportamentos periodicamente em sequência formando um ciclo de controle autonômico. Eles estão descritos a seguir:

Monitoramento: Coleta dados dos roteadores virtuais e alimenta a base de conhecimento.

Análise: Realiza o diagnóstico de falhas nos roteadores virtuais ou em nós físicos da vizinhança.

Planejamento: Calcula o custo do nó físico a partir da utilização de seus recursos. Quanto mais ocupado estiver, maior será o custo. E difunde essa informação para os outros agentes.

Execução: Verifica se todos os agentes da vizinhança já enviaram suas informações. Caso sim, o agente do nó físico de menor custo instancia os roteadores virtuais neste.

Nós definimos no arquivo de políticas a ordem de execução e os parâmetros dos comportamentos, como a taxa de execução do laço e o limite máximo de tempo que um agente pode ficar sem difundir informações para que seja considerada uma falha na rede física. O DP é responsável por interpretar o arquivo de políticas, alterar parâmetros dos comportamentos, e controlar o ciclo de vida do agente. Ele também tem acesso à base de conhecimento e pode atuar de acordo com informações contidas nela.

Para que o sistema multiagentes funcione de maneira correta é preciso que os agentes sejam sincronizados para executar algumas de suas ações, por exemplo: a execução do reparo da rede virtual só deve ser feita depois que todos os custos das

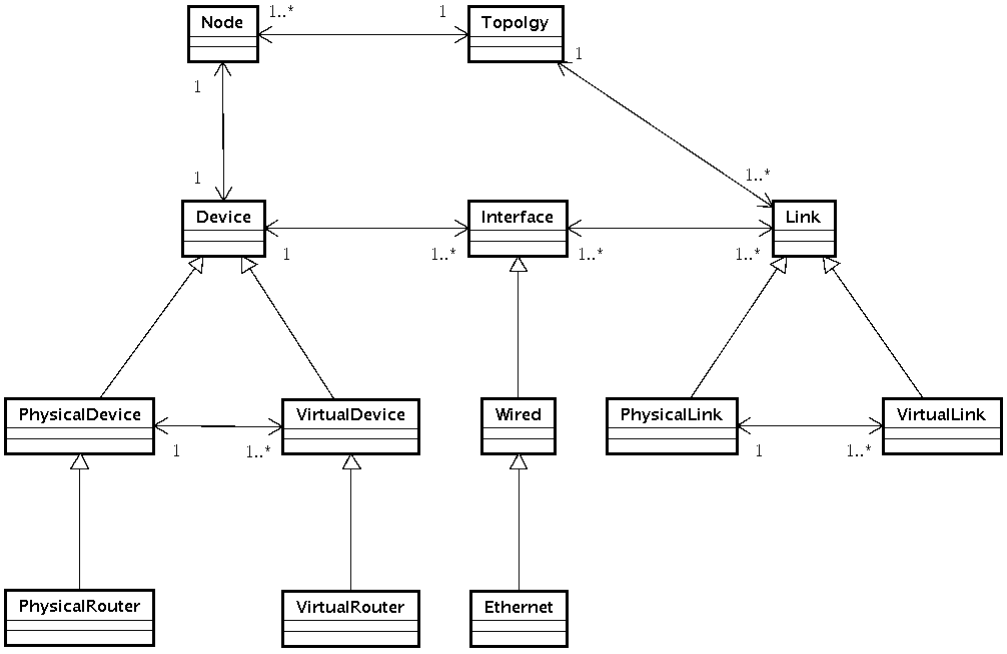


Figura 3. Modelo de informação que descreve a infraestrutura e as redes virtuais.

máquinas foram calculados, para garantir que apenas um agente irá executar a ação de instanciar o roteador virtual. Nós criamos essa sincronização através de informações na base de conhecimento que controlam os comportamentos.

A base de conhecimento serve de repositório para todas as informações do agente, que podem ser coletadas localmente, pelo monitoramento, ou remotamente, através da comunicação com outros agentes. A base de conhecimento possui um modelo de informação comum entre os agentes vizinhos para permitir a comunicação e a interpretação dos dados. O modelo de informação apresentado na Figura 3 representa as topologias das redes virtuais e seus mapeamentos na rede física. Esse modelo foi desenvolvido em trabalhos do projeto Horizon e é baseado em [Fajjari et al. 2010].

Os agentes podem se organizar em vizinhanças, o que limita o escopo de difusão da base de conhecimento. Através dessa comunicação seletiva os agentes formam visões situadas da rede. Um agente pode fazer parte de diversas vizinhanças e difundir diferentes informações para cada uma delas. Na arquitetura do sistema de autogerenciamento de redes virtuais existe apenas uma vizinhança com todos os agentes da rede, e todas a informações geradas pelo agente são difundidas para os vizinhos.

4. Implementação

Antes de aplicar o sistema em um ambiente real é importante testá-lo em um cenário controlado. A Figura 4 ilustra o *testbed* construído com essa finalidade.

A infraestrutura da rede é composta por três máquinas: *zeus*, *cronos* e *dionisio*, que são interligadas por dois comutadores, um de 100Mbps e um de 1Gbps. Sobre ela foi criada uma rede virtual com dois roteadores: *vrouter1A* e *vrouter2A*. As máquinas *artemis* e *apolo* são os *hosts* que farão uso da rede virtual para se comunicarem. Os caminhos foram definidos por rotas estáticas nos *hosts* e interfaces virtuais, que utilizam

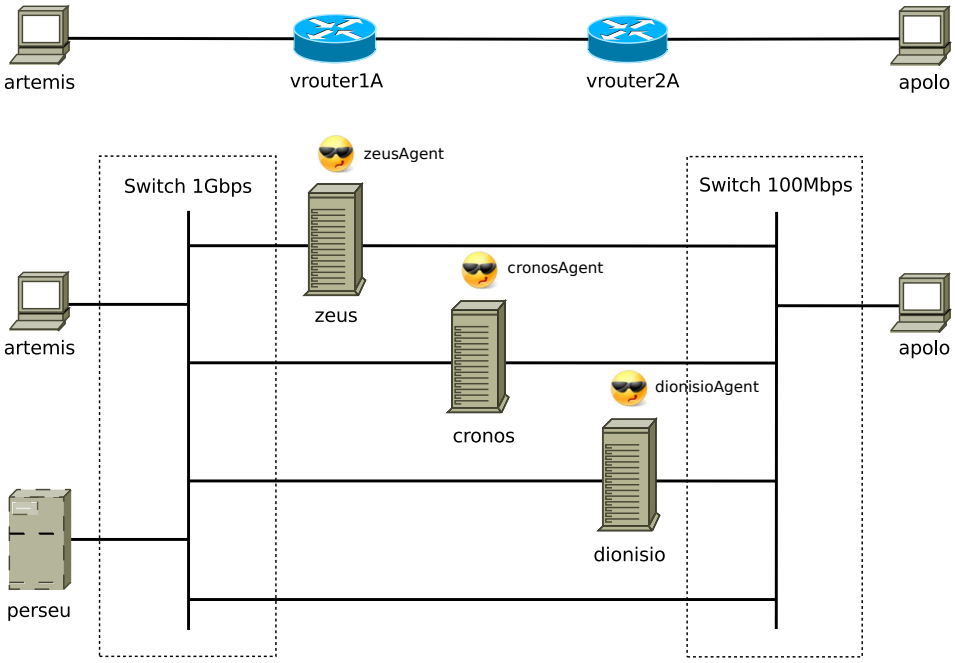


Figura 4. *Testbed* criado para validação do sistema de gerenciamento distribuído.

um endereçamento diferente da rede de controle. Na máquina *perseu* fica o repositório de arquivos, onde estão armazenados as imagens das máquinas virtuais. Tanto as máquinas físicas quanto as virtuais possuem o sistema operacional Debian GNU/Linux com a versão 2.6.32 do kernel.

Os roteadores virtuais foram criados com o KVM. Ele possui um bom desempenho e oferece virtualização total. O KVM possui suporte a SMP, *ballooning* de memória, interligação de interfaces físicas e virtuais através de comutação ou roteamento, e migração *live*, em que toda a memória da máquina virtual é copiada em tempo de execução antes do controle ser transmitido da origem para o destino.

O gerenciamento das máquinas virtuais tanto para a montagem do *testbed* quanto pelos agentes utiliza a biblioteca Libvirt [Coulson et al. 2010]. Ela fornece uma API para monitoramento e controle de diversas plataformas de virtualização, entre elas KVM, Xen e VMware. O uso da Libvirt no sistema de autogerenciamento distribuído de redes virtuais é importante por torná-lo independente de tecnologia.

Os agentes rodam nas máquinas físicas do núcleo da rede: *zeus*, *cronos* e *dionisio*, como mostra a Figura 4. Nós utilizamos a plataforma Ginkgo [Ginkgo Networks 2008] para a implementação dos agentes. Ela permite a criação de agentes leves e portáveis, o que facilita sua implementação em ambientes heterogêneos: roteadores, comutadores, de redes guiadas ou sem-fio. O Ginkgo é um *framework* que possui os blocos de construção básicos dos agentes e os mecanismos para comunicação intra-agentes e para o controle remoto dos agentes através de serviços web.

O experimento tem como objetivo validar o sistema de autogerenciamento distribuído e testar diferentes abordagens para o reparo das redes virtuais. O tempo de recuperação de um roteador virtual pode ser definido segundo a fórmula:

$$T_r = T_d + T_p + T_i + T_c$$

onde T_d é o tempo de diagnóstico da falha; T_p é o tempo gasto no planejamento da ação, que envolve trocas de informações entre os agentes; T_i é o tempo de instanciação da máquina virtual e T_c é o tempo de configuração do roteador através do algoritmo de roteamento.

Nos experimentos nós utilizamos duas abordagens para o planejamento da execução, uma pré e uma pós-diagnóstico. Na primeira, o comportamento Planejamento é executado a cada ciclo, calculando o custo da máquina e difundindo essa informação através da vizinhança. Na segunda, ele é executado apenas quando é diagnosticada uma falha pelo comportamento Análise do agente.

Também foram estudadas duas formas para a instanciação do roteador virtual. Uma delas inicializa o sistema da máquina virtual utilizando a imagem que está no repositório de arquivos na máquina *perseu*. A outra utiliza um arquivo com um estado anterior da memória, que também está no repositório de arquivos, gerado em um momento em que o roteador virtual já estava em funcionamento. Nesse caso, não é preciso inicializar ou reconfigurar as rotas.

As execuções foram realizadas com roteamento estático e dinâmico na rede virtual. O roteamento estático foi configurado manualmente nos roteadores virtuais. Para o dinâmico foi utilizada a suíte de roteamento Quagga [Ishiguro 2006], rodando o algoritmo de roteamento OSPFv2.

5. Resultados e discussão

Um experimento inicial foi realizado no *testbed* antes da implementação do sistema multiagentes. Nesse experimento foram realizadas migrações de um roteador virtual com roteamento estático enquanto um fluxo UDP de taxa constante de 500Kbps de *artemis* a *apolo* passava pela rede virtual. O tráfego foi gerado com a ferramenta Iperf. As curvas do gráfico da Figura 5 mostram a taxa de chegada em *apolo* e quando ela está em zero indica perdas na rede. Todas as migrações iniciaram próximas ao instante 8s.

Na primeira execução, a máquina virtual foi destruída na origem e inicializada no destino. Nela, a recuperação demorou 27s devido principalmente ao tempo de inicialização do sistema. Na segunda execução o estado do roteador virtual foi salvo no repositório de arquivos, ele foi interrompido e restaurado no destino. Mesmo gastando mais tempo com o salvamento, essa execução levou apenas 45% do tempo da anterior. A terceira e a quarta utilizam serviços de migração oferecidos pelo KVM. Na migração normal a máquina virtual é interrompida, a memória é copiada pela rede da origem ao destino onde, por fim, a máquina é reativada. Essa situação é parecida com a anterior, exceto pelo fato que a cópia é feita diretamente e não através de uma máquina intermediária e portanto ela é um pouco mais rápida, 37% do tempo da primeira execução. Nessas duas situações, no momento que as máquinas foram restauradas ocorreu um pico na rede. Isso ocorreu provavelmente porque no instante em que a memória da máquina virtual foi salva, havia pacotes no *buffer* que se somaram aos que estavam chegando no instante que ela foi reativada. A última execução foi feita com migração live, em que a memória é copiada da origem ao destino enquanto a máquina virtual está em execução e somente quando não houver mais nenhuma página modificada, o controle é transferido. Essa é

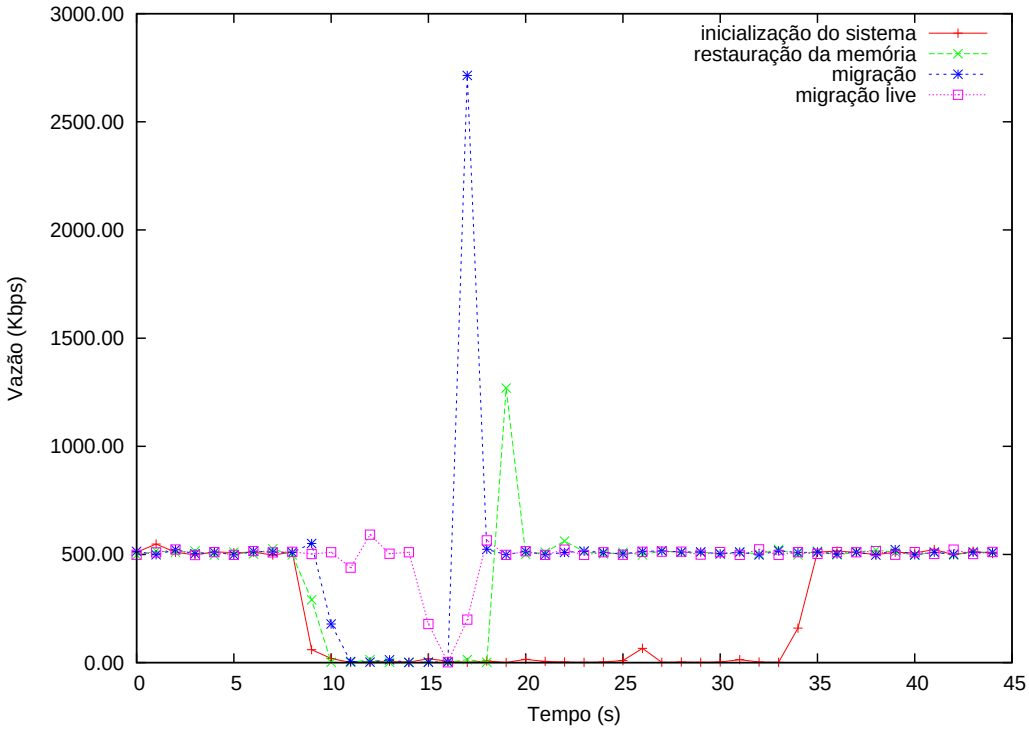
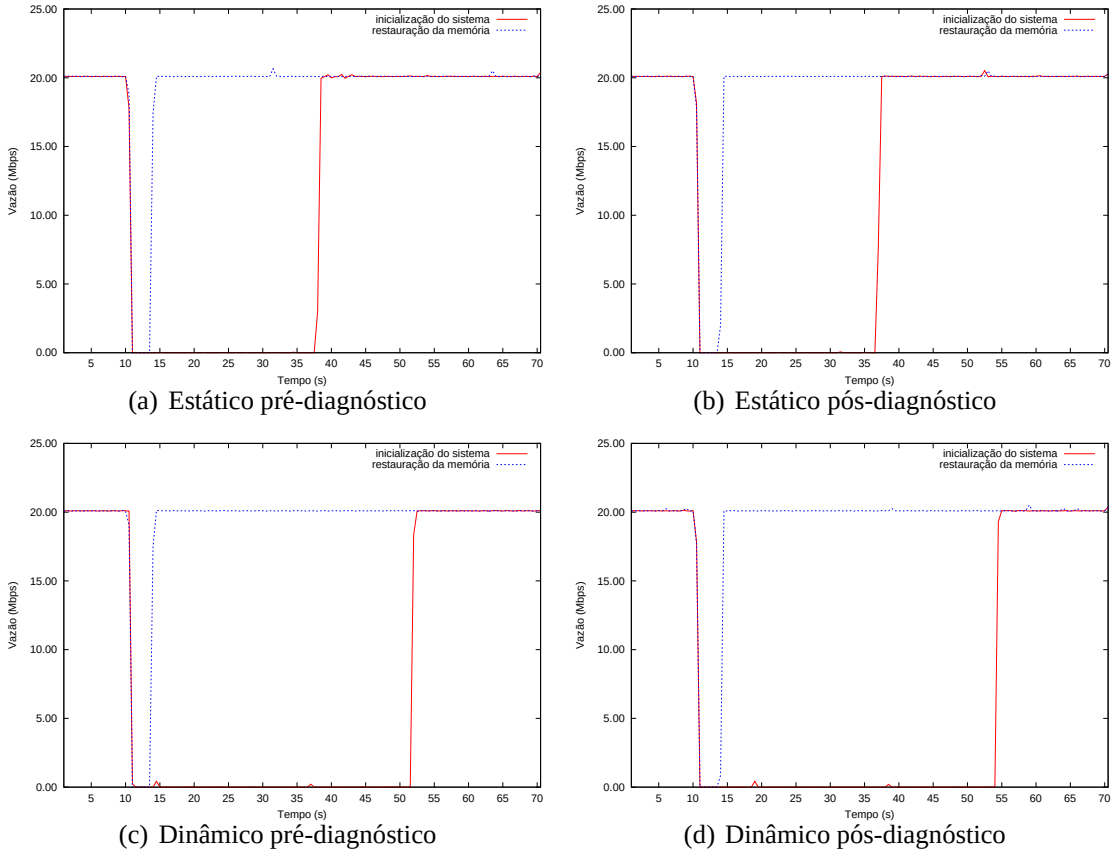


Figura 5. Gráfico do experimento sem o sistema multiagentes.

a melhor forma de migrar um roteador virtual, pois a transmissão foi interrompida por menos tempo, menos de 15% em relação à primeira execução.

Os gráficos da Figura 6 mostram os resultados do experimento com o sistema de gerenciamento distribuído. Os gráficos (a) e (b) apresentam as execuções com roteamento estático, e (c) e (d), com roteamento dinâmico. Em (a) e (c) estão os gráficos das execuções que utilizaram o planejamento pré-diagnóstico e em (b) e (d), os de planejamento pós-diagnóstico. Em cada gráfico estão sendo comparadas as abordagens de inicialização do sistema da máquina virtual e de restauração a partir de um arquivo com um estado anterior da memória. Em todas as execuções um fluxo UDP de taxa constante de 20Mbps de *artemis* a *apolo*, gerado pela ferramenta Iperf, está passando pela rede virtual. As curvas representam as taxas de chegada de pacotes ao destino ao longo do tempo.

Após 10s a máquina *zeus* é desconectada da rede. Os agentes em *cronos* e *dionisio* percebem a falha de *zeus* porque param de receber a difusão da sua base de conhecimento. No experimento os agentes realizam a difusão a cada 0,5s e quando eles percebem que a informação de algum nó físico está desatualizada a mais de 1s o comportamento Análise gera o diagnóstico de falha. Portanto, em todos os experimentos, o tempo de diagnóstico da falha varia entre 1 e 1,5s. No caso do planejamento pré-diagnóstico, os agentes iniciam imediatamente a execução, e no pós-diagnóstico, o comportamento Planejamento realiza o cálculo do custo que será enviado aos demais agentes na próxima difusão. O comportamento Execução verifica se a informação do custo de todos os nós físicos, com exceção do que falhou, está atualizada. O custo de *dionísio* é o maior porque ele já possui um roteador virtual em execução. Portanto, quem irá recuperar *vrouter1A* é o agente em *cronos*, que inicializa o sistema do roteador virtual ou recupera o estado da memória se



(a) Estático pré-diagnóstico (b) Estático pós-diagnóstico

(c) Dinâmico pré-diagnóstico (d) Dinâmico pós-diagnóstico

Figura 6. Gráfico do experimento com os agentes

encontrar o arquivo de *backup*. Ele difunde a informação de término da execução através da base de conhecimento e todos os agentes limpam as informações de planejamento.

As maiores variações estão na forma que a máquina virtual é recuperada, como pode ser visto na diferença do tempo de restabelecimento da conexão, que chega a ser quase 9 vezes maior quando o sistema tem que ser carregado. A diferença é maior nos casos em que um algoritmo de roteamento está sendo executado, pois o roteador virtual precisa reconfigurar suas rotas para restabelecer a conexão da rede. A recuperação através da inicialização do sistema demorou 41,1s com o planejamento pré-diagnóstico e 43,6s com o pós-diagnóstico. Com o roteamento estático a diferença é menor. Os tempos de recuperação foram de 27,5s com pré-diagnóstico e 26,4s com pós-diagnóstico.

Na abordagem em que o estado da memória é restaurado, o tipo de roteamento praticamente não interferiu, pois a máquina virtual2A não chegou a fazer alterações na tabela de roteamento. Tanto no cenário estático quanto no dinâmico, o tempo de recuperação foi de 3,1s com o planejamento pré-diagnóstico e de 3,5s com o pós-diagnóstico. Embora essa abordagem seja mais rápida, o salvamento, a transmissão e o armazenamento do estado da memória causam impactos significativos. É interessante a utilização de *ballooning* para reduzir a memória da máquina virtual.

As estratégias de planejamento pré e pós-diagnóstico possuem pouca diferença, e é mais perceptível na abordagem de restauração da memória, pois na outra abordagem o tempo é mais variável devido à inicialização do sistema. Porém, devemos levar em

consideração que o nosso *testbed* é pequeno e que em cenários maiores a sobrecarga na rede para o planejamento da execução pode aumentar bastante, causando uma latência maior. Nesse sentido é interessante a separação da rede em vizinhanças menores e mais locais para reduzir os impactos da difusão. A agregação de dados como é feita no cálculo local do custo também é uma estratégia para reduzir a sobrecarga.

Também seria possível combinar as estratégias anteriores e salvar o estado da memória do roteador virtual na máquina de menor custo calculado no planejamento prévio. Isso seria bom para evitar o armazenamento centralizado, o que gera um ponto único de falhas e aumenta a sobrecarga da rede.

6. Conclusão

Sistemas baseados em computação bio-inspirada geralmente não possuem desempenho ótimo, mas podem apresentar características interessantes como robustez e escalabilidade. Nosso sistema multiagentes de autogerenciamento distribuído de redes virtuais utiliza os conceitos da computação autonômica para lidar com a complexidade. O sistema apresentou-se funcional no cenário estudado, cujo foco é a autocura das redes virtuais. Também pudemos analisar diferentes abordagens para a recuperação dos roteadores virtuais, como a restauração do estado da memória e o planejamento de ações pré-diagnóstico.

Os resultados se mostraram favoráveis à utilização de técnicas de backup de memória, em especial se tratando de roteadores virtuais para ganhar o tempo de configuração das rotas. Esse backup poderia ocorrer em momentos de ociosidade da rede para não impactar no funcionamento da mesma e os sistemas de virtualização poderiam oferecer recursos para manter a cópia da memória atualizada de maneira eficiente.

Para os trabalhos futuros pretendemos validar o sistema multiagentes de autogerenciamento de redes virtuais em um cenário maior. Também desejamos estudar algumas questões levantadas na discussão, como a utilização de técnicas de *ballooning* de memória e a manutenção dos *backups* do estado dos roteadores virtuais através do sistema multiagentes. Outro assunto de interesse é o autogerenciamento de enlaces virtuais. Alguns problemas poderiam ser corrigidos apenas com a migração do enlace virtual, evitando a migração de roteadores virtuais, que possui um custo maior.

Referências

Anderson, T., Peterson, L., Shenker, S., and Turner, J. (2005). Overcoming the internet impasse through virtualization. *Computer*, 38(4):34 – 41.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. (2003). Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, 37:164–177.

Boucadair, M., Georgatsos, P., Wang, N., Griffin, D., Pavlou, G., Howarth, M., and Elizondo, A. (2009). The agave approach for network virtualization: differentiated services delivery. *Annals of Telecommunications*, 64:277–288. 10.1007/s12243-009-0103-4.

Braga, T. R. M., Silva, F. A., Ruiz, L. B., and Assunção, H. P. (2006). Redes autonômicas. In *Minicursos do Simpósio Brasileiro de Redes de Computadores, SBRC'2006*.

Clark, D. D., Partridge, C., Ramming, J. C., and Wroclawski, J. T. (2003). A knowledge plane for the internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 3–10, New York, NY, USA. ACM.

Coulson, D., Berrange, D., Veillard, D., Lalancette, C., Stump, L., and Jorm, D. (2010). Libvirt 0.7.5: Application development guide. disponível em: http://libvirt.org/guide/pdf/application development guide.pdf.

Dobson, S., Denazis, S., Fernández, A., Gaïti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., and Zambonelli, F. (2006). A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1:223–259.

Fajjari, I., Ayari, M., and Pujolle, G. (2010). Vn-sla: A virtual network specification schema for virtual network provisioning. *International Conference on Networking*, 0:337–342.

Feamster, N., Gao, L., and Rexford, J. (2007). How to lease the internet in your spare time. *SIGCOMM Comput. Commun. Rev.*, 37:61–64.

Gaïti, D., Pujolle, G., Salaun, M., and Zimmermann, H. (2006). Autonomous network equipments. In *Autonomic Communication*, pages 177–185.

Ginkgo Networks (2008). Ginkgo distributed network piloting system. white paper.

Habib, I. (2008). Virtualization with kvm. *Linux J.*, 2008.

Horizon (2010). Horizon project: A new horizon to the internet. disponível em: http://www.gta.ufrj.br/horizon/.

Houidi, I., Louati, W., and Zeghlache, D. (2008). A distributed and autonomic virtual network mapping framework. *Autonomic and Autonomous Systems, International Conference on*, 0:241–247.

Houidi, I., Louati, W., Zeghlache, D., Papadimitriou, P., and Mathy, L. (2010). Adaptive virtual network provisioning. In *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*, VISA '10, pages 41–48, New York, NY, USA. ACM.

IBM (2001). Autonomic computing: Ibm's perspective on the state of information technology.

Ishiguro, K. (2006). Quagga: A routing software package for tcp/ip networks. disponível em: http://www.quagga.net/docs/quagga.pdf.

Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36:41–50.

Marquezan, C., Granville, L., Nunzi, G., and Brunner, M. (2010). Distributed autonomic resource management for network virtualization. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 463 –470.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38:69–74.

Strassner, J. (2007). *Cognitive networks: towards self-aware networks*, chapter The role of autonomic networking in cognitive networks, pages 23–52. John Wiley and Sons.

Turner, J. and Taylor, D. (2005). Diversifying the internet. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 2, pages 6 pp. –760.

Yu, M., Yi, Y., Rexford, J., and Chiang, M. (2008). Rethinking virtual network embedding: substrate support for path splitting and migration. *SIGCOMM Comput. Commun. Rev.*, 38:17–29.

Zhu, Y. and Ammar, M. (2006). Algorithms for assigning substrate network resources to virtual network components. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –12.

MeshAdmin: Plataforma Integrada de Gerência para Redes em Malha sem Fio

Rafael De Tommaso do Valle[1], **Débora Christina Muchaluat-Saade**[2]

[1]Departamento de Engenharia de Telecomunicações

[2]Instituto de Computação
Laboratório MídiaCom – Universidade Federal Fluminense (UFF)
Rua Passos da Pátria, 156, Bl. E, sl. 408 – 24210-240 – Niterói – RJ – Brasil

{rafael,debora}@midiacom.uff.br

Abstract. *There are several techniques and tools proposed in the literature for wireless mesh network management. Each of those techniques and tools usually includes just a few functionalities leading to the use of more than one together. This article discusses a set of requirements of an integrated platform for wireless mesh network management. This work also presents MeshAdmin, a management platform that fulfills a subset of the presented requirements. In order to evaluate MeshAdmin monitoring traffic overhead, this work presents performance tests done in a real mesh testbed.*

Resumo. *Existem diversas técnicas e ferramentas propostas na literatura para gerência de redes em malha sem fio. Porém, essas técnicas e ferramentas normalmente focam em um conjunto de requisitos específicos, fazendo-se necessária a utilização de mais de uma delas em conjunto. Este artigo discute os requisitos de uma plataforma de gerência integrada para redes em malha sem fio, que engloba diversas funcionalidades. Este trabalho também apresenta MeshAdmin, uma plataforma de gerência que satisfaz a um grupo dos requisitos apresentados. Ainda, para avaliar o quantidade de tráfego de monitoramento inserida pela ferramenta são apresentados testes de desempenho feitos em um testbed real de rede mesh.*

1. Introdução

Redes em malha sem fio (redes mesh) têm surgido como uma alternativa de baixo custo para promover acesso banda larga [Muchaluat-Saade et al. 2007]. Os nós mesh utilizam o padrão IEEE 802.11 no modo ad hoc e formam uma malha sem fio com transmissão através de múltiplos saltos. As principais vantagens na sua utilização são a facilidade de instalação e tolerância a falhas [Passos et al. 2006]. Para viabilizar a utilização em larga escala dessa tecnologia é necessária uma plataforma para gerência de redes que facilite a operação e manutenção da rede.

Gerência de redes mesh é uma tarefa mais complexa que gerência de redes cabeadas devido, principalmente, à limitação de recursos na rede e à grande variabilidade de qualidade nos enlaces sem fio [Duarte et al. 2007]. As soluções para monitoramento de redes cabeadas não apresentam desempenho satisfatório quando utilizadas em redes em malha sem fio [Sailhan et al. 2007]. Ainda, há grande diferença na gerência de redes

mesh em comparação à gerência de redes sem fio infraestruturadas, principalmente devido à maior confiabilidade e capacidade do *backbone* cabeado em relação ao *backbone* mesh [Nanda and Kotz 2008].

Soma-se a isso o fato de que, atualmente, existem poucas soluções para gerenciar redes mesh no mercado, sendo geralmente ferramentas desenvolvidas exclusivamente para soluções mesh proprietárias. Essas soluções proprietárias têm custo muito elevado, o que pode inviabilizar a utilização das mesmas em projetos que possuem recursos limitados, como projetos de inclusão digital e projetos de cidades digitais.

Na literatura, existem algumas técnicas e ferramentas desenvolvidas para auxiliar na gerência de redes em malha sem fio (algumas delas são apresentadas na Seção 2). Normalmente, cada uma delas individualmente não é suficiente para gerenciar uma rede em malha, tornando-se necessária a utilização de outras ferramentas em conjunto. Isso causa uma certa dispersão das informações, pois cada ferramenta tem sua própria base de dados e/ou interface, e sobreposição de funcionalidades, quando diferentes ferramentas tratam de alguns parâmetros em comum. Pode até ser interessante comparar informações de duas fontes diferentes, porém, essa sobreposição significa maior consumo de memória e CPU nos roteadores e maior tráfego de informação na rede. Considerando que usualmente o hardware de roteadores sem fio tem recursos limitados, é desejável que se utilize o mínimo de recursos possível.

Neste contexto, existe a necessidade de desenvolvimento de uma plataforma de gerência de redes em malha sem fio integrada, unificando as informações coletadas através dos roteadores em uma única base de dados e possibilitando a visualização das mesmas através de uma única interface. Este trabalho tem como objetivo definir um conjunto de requisitos para o desenvolvimento de uma plataforma integrada de gerência de redes em malha. A plataforma deve fornecer informações suficientes para que o(s) administrador(es) não precise(m) utilizar outras ferramentas na gerência da(s) rede(s). Este artigo também apresenta a implementação atual de MeshAdmin, uma plataforma que já atende a grande parte dos requisitos levantados.

O restante do texto está dividido da seguinte maneira. A Seção 2 discute diversas ferramentas e técnicas utilizadas para gerência de redes em malha sem fio. A Seção 3 aponta os principais desafios para gerência de redes em malha e propõe os requisitos de uma plataforma integrada. A Seção 4 apresenta a plataforma de gerência integrada MeshAdmin. A Seção 5 mostra alguns resultados da avaliação da plataforma em uma rede mesh real. Por fim, a Seção 6 traz as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Sailham et al. [Sailhan et al. 2007] propõem uma arquitetura que organiza os nós da rede em uma estrutura hierárquica baseada em cluster para disseminação das informações coletadas. Um nó, chamado de *cluster head*, é eleito para coordenar e publicar as informações referentes ao conjunto de nós do qual ele faz parte e aos outros nós também definidos como *cluster heads*. Para coleta de informações, a ferramenta aproveita informações disseminadas pelo protocolo de roteamento OLSR. O trabalho também propõe uma estrutura de diretórios para armazenamento das informações coletadas referentes aos nós e para mapear os nós de acordo com suas características.

Raghavendra et al. [Raghavendra et al. 2009] propõem o MeshMon, um *frame-*

work multinível para monitoramento de redes em malha sem fio. Nesse trabalho, é feita uma classificação hierárquica de métricas para redes mesh, voltadas para detecção automática de erros. Basicamente, MeshMon coleta um pequeno conjunto de dados capaz de identificar possíveis problemas na rede. Caso um problema seja detectado, a ferramenta passa a coletar novas informações relacionadas ao problema.

Nanda e Kotz [Nanda and Kotz 2008] propõem uma ferramenta de gerência de redes mesh proativa, desenvolvida para ambientes dinâmicos. Cada nó é responsável por monitorar seus recursos e de seus vizinhos a um determinado número de saltos de distância. Ainda, cada nó deve manter uma representação detalhada da rede ao seu redor e uma representação com menos detalhes da rede como um todo. A ferramenta possui uma unidade de análise da rede que é responsável por: monitorar o estado da rede, através de medidas ativas de parâmetros como perda de pacotes e RTT (*round trip time*); gerenciar os parâmetros configurados em cada nó, quando um parâmetro de um nó tem um valor discrepante comparado a um vizinho, um alerta é gerado; e analisar a topologia, de forma que seja possível prever e detectar possíveis partições na rede.

Huang et al. [Huang et al. 2007] definem em seu trabalho o *MeshFlow*, um *framework* para monitoramento baseado no padrão internacional da Cisco para monitoramento de tráfego, o Netflow [Netflow 2010]. O *framework* é composto de diversos componentes: estrutura de um registro do *MeshFlow*, criação de registro, manutenção de registro, difusão de registro, agregação de registros e análise. Um registro é um pacote que contém algumas propriedades relacionadas ao tráfego que passa por cada nó mesh. Esses registros são mantidos na memória dos nós e depois de um tempo predeterminado são enviados a um servidor dedicado responsável por armazenar e analisar os dados coletados.

Jardosh et al. [Jardosh et al. 2008] apresentam o SCUBA, uma ferramenta para monitoramento em tempo real de redes em malha sem fio. Possui três contextos de visualização da topologia: contexto da rota, que mostra informações referentes à vazão e RTT; contexto do enlace, que mostra informações sobre a qualidade dos enlaces baseado na métrica ETX [Couto et al. 2003]; e contexto do cliente, que mostra informações referentes aos clientes associados a cada nó mesh, como número de clientes por nó, porcentagem de utilização do canal por cliente e RSSI (*Received Signal Strength Indicator*) dos quadros recebidos.

A ferramenta MTV (*Mesh Topology Viewer*) [Valle et al. 2008] tem como objetivo mostrar em tempo real a topologia da rede e a qualidade de cada enlace. A partir de um arquivo de configuração XML, são passadas informações referentes à topologia da rede. Já as informações referentes à qualidade dos enlaces são coletadas no *gateway* da rede através da métrica de roteamento utilizada. Essas informações são processadas por um programa CGI em um servidor web e são apresentadas graficamente pelo navegador.

Pinheiro et al. [Pinheiro et al. 2010] propõem o *framework* Abaré, para implantação, monitoramento e gerência para redes em malha sem fio. O *framework* possui um módulo responsável pela interação com o administrador, um núcleo responsável pela parte lógica do sistema e pelo armazanamento de informações e uma camada para acesso aos nós da rede.

O WiFiDog [Lenczner 2005] é uma ferramenta aberta de *captive portal* que possibilita autenticação de usuários e monitoramento centralizado de uma rede sem fio.

O CoovaChilli [CoovaChilli 2010] também é uma ferramenta de controle de acesso de usuários para redes sem fio para *captive portal* e acesso via 802.1X. Uma das vantagens do CoovaChilli é a possibilidade da utilização de um servidor RADIUS para autenticação dos usuários. Assim, como o WiFiDog, o CoovaChilli também fornece diversas estatísticas sobre os usuários da rede.

3. Plataforma de Gerência Integrada

Como dito anteriormente, gerência de redes em malha é uma tarefa mais complexa que gerência de redes cabeadas ou redes sem fio infraestruturadas [Duarte et al. 2007, Sailhan et al. 2007, Nanda and Kotz 2008]. Os principais desafios encontrados na gerência de rede em malha são: limitação de recursos dos equipamentos, como espaço em disco e memória disponível; *backbone* formado por enlaces sem fio acarretando limitação de banda para mensagens de controle; grande variação da qualidade dos enlaces; e nós fora do alcance físico dos administradores (topo de edifícios, torres, postes etc). Uma discussão mais detalhada sobre os principais desafios da gerência de redes em malha sem fio pode ser encontrada em [Duarte et al. 2007].

De um modo geral, as funcionalidades de uma plataforma de gerência incluem:

- coletar e armazenar estatísticas da rede;
- monitoramento da rede, permitindo a visualização da topologia da rede, informando a qualidade dos enlaces e facilitando a verificação de falhas na rede;
- efetuar o controle de usuários que acessam a rede.

3.1. Coleta de Dados

Uma plataforma de gerência deve ser capaz de coletar dados referentes aos nós e enlaces da rede. Desta forma, será possível observar o comportamento da rede como um todo e com o auxílio das técnicas de monitoramento será possível identificar e reparar falhas.

Cada nó possui um conjunto de parâmetros que necessitam ser observados para que seja possível identificar falhas em seu funcionamento, tais como: *uptime* (tempo desde a última iniciação); uso de CPU; memória utilizada/disponível; bytes trafegados nas interfaces LAN, WLAN e WAN (apenas para os *gateways*); espaço em disco; informações do rádio, tais como taxa de transmissão e potência de transmissão.

A qualidade dos enlaces da rede também é um importante parâmetro a ser verificado, pois com essa informação pode-se identificar problemas na instalação dos pontos mesh. Na literatura relacionada, são encontradas diversas referências a métricas para redes em malha sem fio que indicam a qualidades dos enlaces [Campista et al. 2008].

3.2. Armazenamento de Dados

Uma plataforma deve, ainda, guardar os dados coletados por determinado período. Não é interessante que estes dados sejam armazenados nos roteadores, primeiro devido a sua limitação de recursos e, segundo, caso um nó fique inacessível, não seria possível observar as informações coletadas sobre o mesmo a fim de se obter um diagnóstico. Logo, é necessário utilizar um servidor com um banco de dados para armazenar as informações coletadas pela plataforma de gerência.

Porém, por um período de tempo reduzido, pode haver necessidade de armazenar as informações coletadas nos nós, para então serem transferidas para a base de dados.

Para tanto, cada nó deve possuir um espaço de memória reservado para o armazenamento desses dados. O tempo de armazenamento dos dados nos nós deve ser definido levando-se em consideração o espaço disponível em memória e a quantidade de tráfego que a transferência dessas informações irá gerar na rede. Isto implica um mecanismo inteligente de transferência de dados do nó para o servidor, como o proposto em [Huang et al. 2007].

Esse mecanismo não deve interferir no tráfego de dados dos usuários. Caso seja utilizado este tipo de abordagem, é interessante que se definam classes de tráfego distintas a fim de que o tráfego de dados coletados não interfira no tráfego dos usuários e vice-versa. Dessa forma, o diagnóstico da rede não será comprometido.

3.3. Monitoramento

Os dados coletados por uma plataforma de gerência devem ser constantemente monitorados para que seja possível a identificação e solução de problemas na rede. As ferramentas de monitoramento devem ser projetadas visando a facilitar ao máximo o trabalho do administrador da rede em se obter o diagnóstico da mesma.

Uma característica essencial para uma plataforma em termos de monitoramento é a possibilidade de visualização da topologia em tempo real [Valle et al. 2008, Jardosh et al. 2008]. A plataforma deve ser capaz de informar ao administrador a qualidade dos enlaces – de acordo com a métrica que lhe for mais conveniente – para que ele possa identificar possíveis falhas. É interessante que a visão da topologia informe a posição geográfica dos nós facilitando a identificação de um possível problema na rede.

Outra característica importante é a possibilidade de visualizar as informações coletadas dos elementos da rede. Com isso, em caso de alguma anormalidade, o administrador pode consultar séries históricas armazenadas na base de dados e identificar algum padrão de falha, como por exemplo aumento do consumo de memória, esgotamento de espaço em disco, taxa excessiva de perda de pacotes em uma interface etc.

As facilidades de visualização de topologia e dos dados coletados necessitam que o administrador acesse a plataforma para obter as informações desejadas. Além disso, a mesma deve ser capaz de enviar ao administrador da rede alertas sobre o (mau) funcionamento da rede. A plataforma deve observar os parâmetros coletados e compará-los com limiares definidos pelo administrador da rede e/ou valores anteriormente coletados. Caso haja alguma anormalidade, a plataforma pode gerar uma mensagem de alerta que pode ser enviada de diferentes maneiras (email, SMS, twitter etc) para o administrador.

3.4. Controle de Usuários

É esperado que uma ferramenta de gerência de rede sem fio efetue o controle de usuários que acessam a rede. Ainda é esperado que ela forneça informações sobre os usuários registrados e sobre o perfil de acesso dos mesmos. Essas informações podem ser divididas em dois grupos, informações sobre a conexão do usuário associado e sobre os fluxos gerados por cada usuário.

Sobre a conexão dos usuários, é desejável informar: usuários conectados; nó mesh a qual cada usuário está conectado; endereço IP de cada conexão; número de fluxos por usuário; data e hora de início de cada conexão; data e hora de fim de cada conexão (se já estiver terminada); quantidade de bytes trafegados por usuário.

Também devem ser coletadas as informações sobre os fluxos gerados por cada usuário, tais como: porta de origem; IP e porta de destino; protocolo de transporte; números de pacotes enviados e recebidos; bytes enviados e recebidos pelo fluxo; *gateway* utilizado pelo fluxo.

A análise de fluxos é importante também para o aspecto de segurança. Pois é possível identificar tráfego malicioso na rede e que usuário originou ou recebeu esse tráfego. Ainda, essas informações são importantes para identificar pontos de saturação da rede, ou seja, determinados nós que estejam sendo acessados por um número considerável de usuários. A partir desses dados é possível replanejar a rede, remanejando ou adicionando nós de forma a melhorar a qualidade da mesma [Huang et al. 2007].

Além desses requisitos que são atendidos pela maior parte das ferramentas de controle de usuários para rede sem fio infraestruturada, temos ainda requisitos que são específicos de redes em malha sem fio e que são desejáveis para uma plataforma integrada de gerência. Eles são descritos a seguir.

Suporte a Múltiplos Saltos: A comunicação em redes em malha se dá através de múltiplos saltos. É necessário que um usuário após se registrar na rede, possa se autenticar a partir de qualquer nó.

Suporte a Múltiplos *Gateways*: Em uma rede em malha sem fio, pode existir mais de um *gateway* [Duarte et al. 2008]. Isso implica que a ferramenta de autenticação possua uma base integrada, que seja consultada por todos os *gateways* no momento da autenticação. Dessa forma, o usuário que se registrar através de um determinado *gateway* poderá se conectar à Internet através de qualquer outro.

Autenticação em múltiplas interfaces: Em redes mesh é possível que o usuário se conecte à rede através de interface cabeada ou sem fio. Por isso, é necessário que a ferramenta seja capaz de autenticar usuários em múltiplas interfaces.

4. Plataforma MeshAdmin

Baseado nos requisitos discutidos, este trabalho apresenta MeshAdmin, uma plataforma integrada para gerência de redes em malha sem fio. A implementação atual da plataforma foi desenvolvida no escopo do projeto ReMoTE, do Laboratório MídiaCom em parceria com a empresa TBE (Transmissores Brasileiros de Energia) [Valle et al. 2009]. Ao longo do projeto, foi desenvolvida uma solução mesh de baixo custo para ser instalada no topo de torres de transmissão de energia. Essa solução foi desenvolvida de modo que fosse compatível com hardware de baixo custo como Linksys WRT54g e Ubiquiti Bullet que são facilmente encontrados no mercado, porém possuem recursos limitados no que diz respeito à capacidade de processamento, memória RAM e espaço em disco. O *firmware* desenvolvido para esses roteadores é uma personalização do sistema operacional para sistemas embarcados OpenWrt[1]. Cada nó mesh utiliza duas antenas direcionais, cada uma apontando para um sentido da linha de transmissão, para aumentar a capacidade de alcance do sinal sem fio. Como nas torres de alta tensão não há fonte de alimentação, os nós são alimentados por baterias que, por sua vez, são carregadas por painéis solares. Para chavear esses equipamentos interconectados no sistema de alimentação do roteador, é utilizado um controlador de carga, que evita descarga da bateria abaixo dos níveis aceitáveis

[1] http://www.openwrt.org

e evita que o roteador receba níveis indesejados de tensão.

MeshAdmin foi desenvolvida de forma modular, de modo que seja possível adicionar novas funcionalidades à ferramenta. A plataforma é composta dos seguintes módulos: Painel de Configuração da Ferramenta, Módulo de Coleta, Módulo de Armazenamento de Dados, Módulo de Alerta e Módulo de Exibição. A Figura 1 mostra como esses módulos se relacionam.

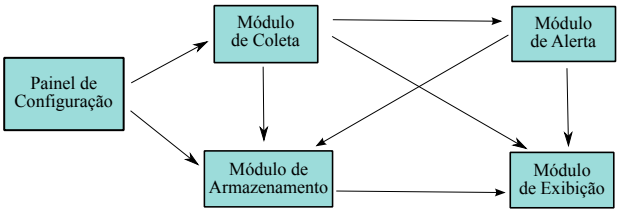


Figura 1. Módulos da plataforma MeshAdmin.

A plataforma MeshAdmin possui um Painel de Configuração que atualiza informações referentes à configuração das redes que serão monitoradas no Módulo de Armazenamento e configura alguns parâmetros do Módulo de Coleta, como interfaces e discos que serão monitorados em cada nó. O Módulo de Armazenamento, além das informações de configuração, também recebe informações do Módulo de Coleta, que reúne informações dos nós e enlaces da rede, e do Módulo de Alerta, a fim de que os registros (*logs*) gerados por este módulo possam ser acessados no futuro. O Módulo de Exibição consulta o Módulo de Armazenamento para obter informações sobre as redes que serão monitoradas, recebe informações do Módulo de Coleta a fim de atualizar as informações de topologia em tempo real e recebe as mensagens do Módulo de Alerta, que serão exibidas na tela. A seguir, cada um desses módulos será definido com detalhes.

4.1. Painel de Configuração da Ferramenta

O painel de configuração da ferramenta é utilizado pelo administrador para incluir as redes mesh que serão monitoradas, os nós de cada rede e outros parâmetros de configuração dos mesmos. O painel é dividido em quatro grupos de configuração: *Auth*, para criação de grupos e usuários e suas permissões; *Configuration*, para inclusão de redes – com nome, descrição e posição geográfica de cada rede –, nós *mesh* – com latitude e longitude, para posicionamento no mapa, identificador, endereço IP da interface WLAN, discos e interfaces que serão monitorados e rede mesh à qual o nó pertence – e *gateways* – com a definição de seu endereço WAN –; *Diagnosis*, para adição dos discos e interfaces que serão monitorados; e *Monitoring*, para adição da métrica de roteamento que será utilizada. Na implementação atual utiliza-se a métrica ML (*Minimum Losses*), utilizada pelo protocolo OLSR-ML [Passos et al. 2006]. Para adicionar novas métricas, possibilitando a utilização de outros protocolos, deverão ser feitas alterações no código da ferramenta.

4.2. Módulo de Coleta

O Módulo de Coleta realiza a coleta de informação dos nós e dos enlaces.

4.2.1. Coleta de Informação dos Nós

Um processo agente é instalado nos nós da rede e um processo gerente é integrado à ferramenta MeshAdmin. O processo gerente faz requisições em busca de informações

que auxiliem o administrador no diagnóstico de problemas nos nós.

Como o hardware utilizado na solução desenvolvida pelo projeto possui recursos limitados, há a necessidade de se utilizar uma implementação compacta de agente SNMP nos roteadores, de modo que não comprometa o uso de memória e CPU dos mesmos.

Foram testadas duas implementações do SNMP desenvolvidas para o OpenWrt: Net-SNMP[2] e Mini SNMP Daemon[3]. A primeira esgotou os recursos de memória, já a segunda obteve um bom desempenho, não comprometendo os recursos do roteador. Por isso, o Mini SNMP é usado nos roteadores mesh para comunicação com o servidor MeshAdmin. Entretanto, o Mini SNMP atende a um número limitado de MIBs, ainda que novas MIBs possam ser implementadas com alterações em seu código. Na implementação atual são usadas as MIBs HOST-RESOURCES-MIB, UCD-SNMP-MIB e IF-MIB.

Além da variáveis definidas pelas MIBs que são implementadas pelo agente SNMP escolhido, também há a necessidade de captar outras informações que não estão definidas em MIBs e que são utilizadas pelos administradores no diagnóstico de problemas na rede, como parâmetros referentes ao rádio dos roteadores e parâmetros do sistema de alimentação com energia solar. Para coletar essas informações, o agente SNMP foi modificado para que o gerente pudesse fazer a coleta desses parâmetros utilizando o protocolo, ao invés de utilizar outros mecanismos de coleta que pudessem não ser tão eficientes, como por exemplo *scripts* de coleta.

Em relação ao rádio, as seguintes variáveis são coletadas: nível de sinal para cada vizinho; taxa de transmissão para cada vizinho e potência de transmissão do rádio.

Em relação ao sistema de alimentação, as seguintes variáveis são coletadas: tensão no painel solar; corrente gerada pelo painel solar; tensão nas baterias; corrente que alimenta o roteador; temperatura interna da caixa que armazena o kit mesh; temperatura externa e luminosidade.

4.2.2. Coleta de Informação de Enlaces

O Módulo de Coleta de Informação de Enlaces é responsável por obter os endereços de origem e destino de todos os enlaces ativos na rede e a respectiva qualidade, indicada pela métrica de roteamento, de cada um desses enlaces. A plataforma de gerência se comunica periodicamente com o *gateway* da rede em busca das informações sobre a qualidade dos enlaces da rede. A periodicidade varia de acordo com as necessidades do administrador da rede e pode ser configurada através do Painel de Configuração.

Essas informações são obtidas através de consultas ao protocolo de roteamento OLSR-ML, que utiliza a métrica de roteamento ML (*Minimum Losses*). Em redes que utilizam OLSR, cada nó mantém uma visão completa da topologia da rede, ou seja, cada nó possui em sua tabela de roteamento todos os nós atingíveis – nós com quem o nó em questão tem conectividade – e a qualidade de cada enlace da malha conectada. Por isso, é possível obter as informações de qualidade dos enlaces consultando apenas um nó da rede. Na implementação de MeshAdmin, os *gateways* da rede são consultados. Dessa forma, para obtenção dessas informações, não há inserção de nenhum tráfego adicional

[2]http://www.net-snmp.org/

[3]http://freshmeat.net/projects/minisnmpd

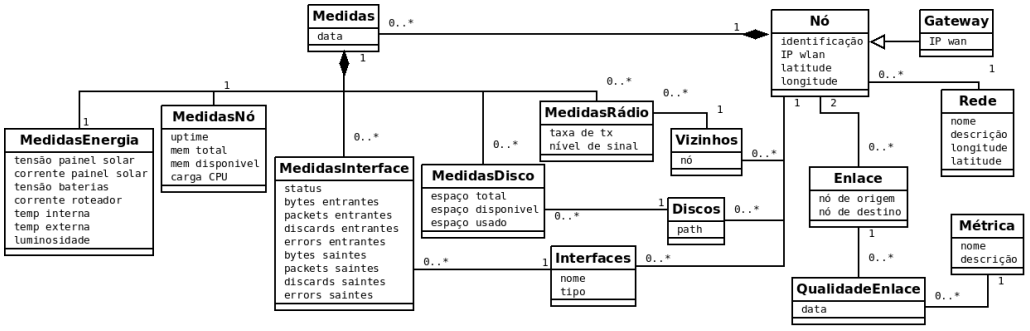


Figura 2. Organização do banco de dados de MeshAdmin.

na malha sem fio.

Os dados coletados são repassados ao módulo de armazenamento e ao módulo de visualização da topologia. Dependendo das necessidades do administrador, a periodicidade com que as informações são coletadas pode ser configurada para cada módulo.

4.3. Módulo de Alerta

O Módulo de Alerta tem como objetivo notificar o administrador da rede sobre eventuais problemas na rede no momento da coleta de informações. Com isso, o administrador pode ser alertado prontamente sobre alguma falha na rede que necessite de reparo imediato sem precisar observar as informações de cada nó.

Esses alertas estão divididos em 3 níveis: Crítico, Aviso e Informação. O nível Crítico é responsável por identificar falhas nos nós e enlaces, como nós desligados ou queda de enlaces. O nível Aviso alerta sobre falhas na configuração da ferramenta, como nós que estão na tabela de roteamento do *gateway*, mas não estão configurados na ferramenta ou alguma falha na coleta de dados. Já o nível Informação deve fornecer informações sobre a execução dos procedimentos da ferramenta, como a coleta de informações dos elementos ou aparecimento de novo enlace na topologia. Considerando que a topologia de redes em malha é dinâmica é interessante informar ao administrador o surgimento de novos enlaces.

4.4. Módulo de Armazenamento de Dados

As informações obtidas pelo Módulo de Coleta são reunidas no Módulo de Armazenamento de MeshAdmin. Ainda, este módulo recebe as informações adicionadas pelo administrador da rede pelo Painel de Configuração e armazena as mensagens geradas pelo Módulo de Alerta.

O Módulo de Armazenamento é composto basicamente por um banco de dados relacional (Postgres[4]), estruturado de forma que seja possível armazenar as informações coletadas e fornecê-las para os diversos módulos da ferramenta. A organização do banco de dados de MeshAdmin é exibida na Figura 2.

A base de dados da plataforma armazena as informações sobre os nós e redes que serão monitoradas. Para cada nó monitorado, serão armazenadas sua identificação, seu endereço IP da interface sem fio e suas coordenadas geográficas. Caso esse nó seja um

[4]http://www.postgresql.org

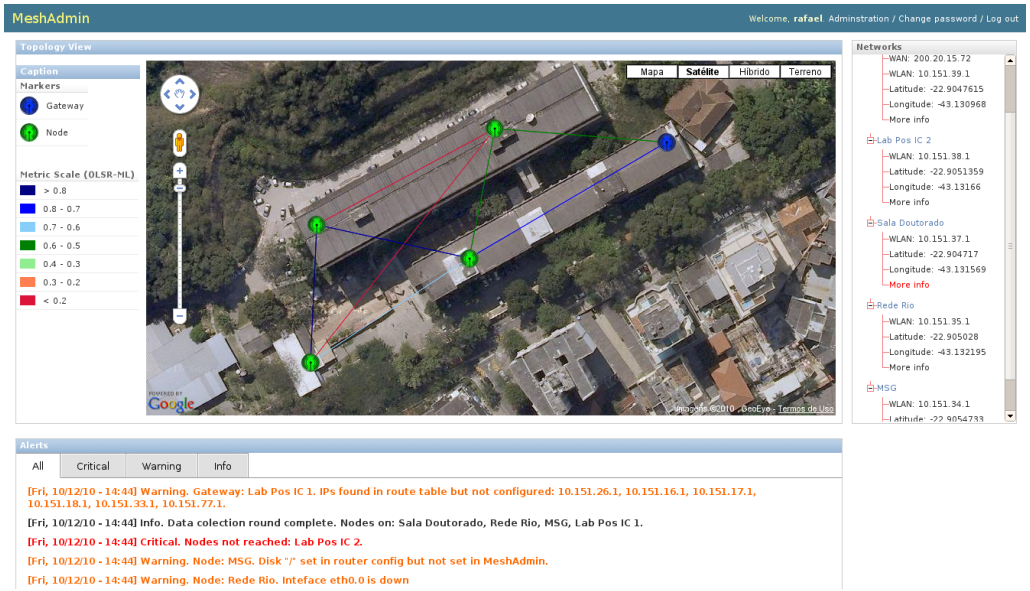


Figura 3. Tela inicial da ferramenta.

gateway, também será armazenado o endereço IP de sua interface WAN. Cada nó deve pertencer a uma rede, que por sua vez, possui um nome, uma descrição e coordenadas geográficas. Os nós mesh possuem interfaces e discos que serão monitorados. Ainda, serão associadas a cada nó, medidas referentes a diversos parâmetros dos mesmos, do sistema de alimentação, do rádio (para cada vizinho), de cada interface e de cada disco monitorados. Finalmente, os enlaces são formados por dois nós. Cada enlace possui uma qualidade de acordo com uma métrica, que também é armazenada na base de dados.

4.5. Módulo de Exibição

Para facilitar o monitoramento remoto, MeshAdmin oferece uma interface web, desenvolvida utilizando o Django, um *framework* de desenvolvimento em Python para web[5].

A tela inicial da ferramenta MeshAdmin possui três *frames*: visualização da topologia, informação de redes e nós e mensagens de alerta. Além disso, a tela inicial possui um link para o painel de configuração da ferramenta. A tela inicial e a interface de configuração só podem acessadas pelos administradores após autenticação na ferramenta. A Figura 3 mostra uma captura da tela inicial de MeshAdmin.

4.5.1. Visualização da Topologia

Um dos principais requisitos que devem ser atendidos por uma plataforma de gerência é a possibilidade de visualização da topologia da rede em tempo real. O módulo de visualização de MeshAdmin tem como principal objetivo mostrar a topologia da rede em tempo real, com os nós da rede posicionados geograficamente em um mapa e os enlaces da rede coloridos de acordo com sua qualidade (vide Figura 3).

As informações sobre cada enlace são coletadas pelo Módulo de Coleta. Então, elas são processadas por um programa em Python que recebe os endereços IPs de origem

[5]http://http://www.djangoproject.com/

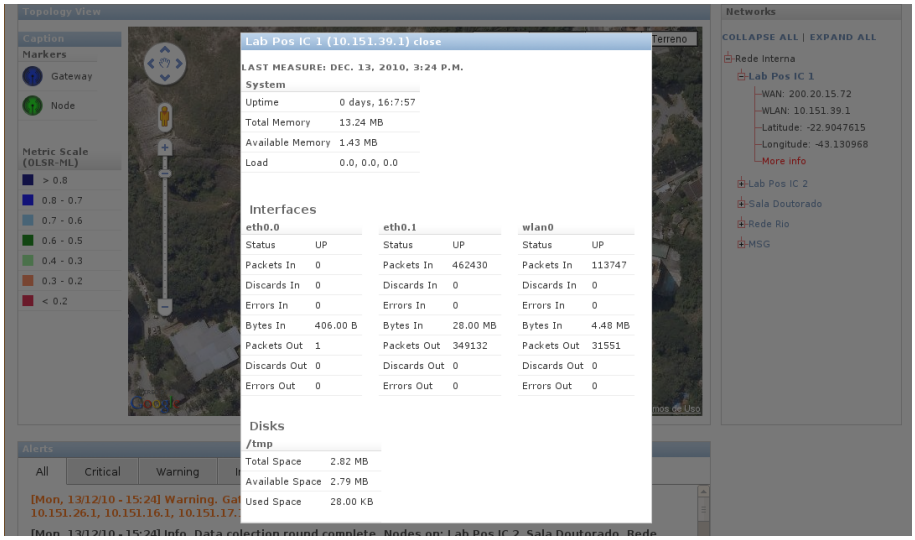


Figura 4. Captura de tela com a informação de um determinado nó.

e destino e busca as informações referentes à posição geográfica na base de dados. As informações de enlace e de posicionamento geográfico dos nós são então processadas e apresentadas utilizando a API Javascript do Google Maps[6]. Nessa interface, os enlaces são desenhados na tela e coloridos de acordo com sua qualidade, em correspondência a uma escala de cores que relaciona cor e qualidade.

4.5.2. Monitoramento dos Nós

No lado direito da tela principal (vide Figura 3), uma árvore tem como objetivo facilitar a navegação do administrador por redes e nós. A partir dessa árvore, o administrador da rede pode centralizar o nó que desejar na visão da topologia, apenas clicando em uma das entradas da árvore. A partir da árvore também é possível acessar as informações de latitude, longitude e endereço IP da interface WLAN de um nó específico. Ainda, há um link disponível para acessar as informações coletadas do nó.

A Figura 4 mostra uma captura da tela da ferramenta com as informações de um nó selecionado na árvore. Os dados que são coletados nos nós estão divididos em 3 grupos: Sistema, Interfaces e Discos. Os dados de Sistema são uptime, memória total, memória disponível e carga de CPU. Para cada interface configurada, são coletados os seguintes parâmetros: bytes trafegados, pacotes trafegados, perdidos e descartados, saintes e entrantes. Para cada disco configurado, são coletados o espaço total, disponível e utilizado. Na implementação atual, os dados são coletados a cada hora. Porém o administrador pode configurar o tempo de coleta de acordo com sua necessidade.

4.5.3. Exibição de Alertas

A tela inicial da plataforma MeshAdmin apresenta, na parte inferior, diversas mensagens de alerta geradas pela ferramenta (vide Figura 3). As mensagens são apresentadas em quatro abas diferentes. A primeira aba, denominada All, apresenta todas as mensagens de

[6]http://code.google.com/intl/pt-BR/apis/maps/documentation/javascript/

alerta. Já as demais, Critical, Warning e Info, apresentam os alertas correspondentes ao seu nível – Crítico, Aviso e Informação, respectivamente.

5. Avaliação da Plataforma

Com intuito de avaliar o desempenho da plataforma em relação ao *overhead* gerado pelo tráfego de monitoramento injetado na rede pelo Módulo de Coleta, foram realizados testes em um ambiente real, na rede em malha sem fio mantida pelo Laboratório MídiaCom no Instituto de Computação da UFF. Foram elaborados 5 cenários diferentes para a realização dos testes: com 5, 7, 9, 10 e 12 nós. Para cada cenário foram realizadas 30 medições a cada 20 minutos. Todas as rotinas de teste foram efetuadas com sucesso pela plataforma MeshAdmin, coletando informação de todos os nós conectados à rede em malha.

Como esperado, a quantidade de tráfego injetado cresce linearmente à medida que o número de nós da rede aumenta. Podemos perceber também, que a quantidade de bytes em cada cenário varia muito pouco. Essa pequena variação se deve há alguma retransmissão por conta de alguma informação incompleta recebida pelo processo gerente na plataforma de gerência. No cenário com 12 nós, trafega na rede uma média de 23 Kbytes de informação de monitoramento. Esse é um valor relativamente baixo considerando-se a capacidade de transmissão da rede.

O tempo de duração também aumenta com o número de nós. Mas seu aumento não se deve apenas à quantidade de nós adicionada, mas também devido à quantidade média de saltos de cada nó ao *gateway*. Na rede utilizada para os testes, o *gateway* fica em uma de suas extremidades. Dessa forma, à medida que nós foram sendo adicionados aos cenários de testes, a quantidade média de saltos de um nó até o *gateway* também aumentou. Com isso, aumenta também a latência média dos nós ao *gateway*, e por sua vez, ao servidor de gerência. Isso faz com que o tempo de coleta de dados seja ainda maior. Porém, mesmo em um cenário com 12 nós, estando os dois últimos nós a uma quantidade média de 7 saltos do *gateway*, o tempo total médio de coleta de informação de todos os nós da rede foi de apenas 4 segundos. Ou seja, para cada rodada de coleta de dados nesse cenário, a plataforma de gerência insere tráfego de monitoramento por apenas 4 segundos. Com base nos resultados, é possível afirmar que o administrador poderia realizar coletas a cada 5 segundos na rede com 12 nós, caso precise de uma granularidade alta de informações.

A Tabela 1 faz uma comparação de MeshAdmin com as ferramentas apresentadas.

6. Considerações Finais

Este trabalho apresentou os requisitos de uma plataforma integrada para gerência de redes em malha sem fio, de modo que o administrador da rede consiga identificar falhas e diagnosticá-las, sem precisar consultar outras ferramentas e necessite de um mínimo de interação com a plataforma. Ainda, descreveu a plataforma MeshAdmin, cuja implementação atual contempla um subconjunto desses requisitos. Essa ferramenta já está em operação na rede de testes mantida pelo Laboratório MídiaCom.

Das ferramentas citadas, MeshAdmin e Abaré são as únicas ferramentas que se preocupam com o monitoramento dos nós da rede. Enquanto as outras ferramentas coletam apenas informações de parâmetros da rede e de fluxos de usuários, MeshAdmin

Tabela 1. Requisitos atendidos pela implementação atual de MeshAdmin.

	Controle de Usuários	Coleta de Dados			Visualização da Topologia	Monitoramento de falhas	
		Usuários	Nós	Rede		Detecção	Notificação
[Sailham et al. 2007]				x		x	x
MeshMon				x		x	x
Mesh-Mon		x		x			
MeshFlow		x		x		x	
SCUBA				x	x	x	
MTV				x	x	x	
Abaré			x		x	x	
WiFiDog	x	x					
CoovaChilli	x	x					
MeshAdmin			x	x	x	x	x

monitora diversos parâmetros dos nós. Em sistemas embarcados com recursos de disco, processamento e memória limitados, é interessante observar o comportamento desses recursos e, com isso, prever possíveis problemas com esses equipamentos. Em relação ao *framework* Abaré, que apresenta grande contribuição na parte de configuração e resolução autonômica de problemas, MeshAdmin contribui ainda, com a possibilidade de visualização da topologia e dos diversos dados coletados de forma gráfica e em um ambiente web, podendo ser acessado a partir de qualquer estação de trabalho conectada à Internet ou à rede em malha.

Além disso, em sua implementação atual, a plataforma já atende boa parte dos requisitos necessários para gerência de redes em malha sem fio, como visualização da topologia e o próprio monitoramento de nós sem gerar tráfego que prejudique o funcionamento da rede. MeshAdmin também se mostrou escalável, não tendo queda de desempenho à medida que novos nós foram sendo adicionados à rede de teste.

Como trabalho futuro, será desenvolvido o módulo para controle de usuários que engloba o controle de acesso e monitoramento de fluxos gerados por cada conexão dos usuários. Ainda, é desejado que se integre à ferramenta de gerência um módulo de configuração da rede, onde alguns parâmetros de um grupo de nós da rede possam ser configurados de uma única vez, sem necessidade de se acessar um a um.

Agradecimentos

Ao programa de P&D ANEEL – com recursos das empresas EATE (Empresa Amazonense de Transmissão de Energia S.A.), LUMITRANS (Companhia Transmissora de Energia Elétrica) e STC (Sistema de Transmissão Catarinense) –, Faperj, CNPq, CAPES e INCT-MACC que possibilitaram este trabalho através do fomento à pesquisa.

Referências

Campista, M., Esposito, P., Moraes, I., Costa, L., Duarte, O., Passos, D., de Albuquerque, C., Muchaluat-Saade, D., and Rubinstein, M. (2008). Routing metrics and protocols for wireless mesh networks. *Network, IEEE*, 22(1):6–12.

CoovaChilli (2010). Open source captive portal access controller and radius software. http://coova.org/wiki/index.php/CoovaChilli. Acessado em Novembro de 2010.

Couto, D., Aguayo, D., Bicket, J., and Morris, R. (2003). A high-throughput path metric for multi-hop wireless routing. In *ACM MobiCom*, San Diego, CA, USA.

Duarte, J., Passos, D., and de Albuquerque, C. (2008). DynTun: Túneis dinâmicos e a escalabilidade de redes em malha. In *Simpósio Brasileiro de Redes de Computadores (SBRC 2008)*, Rio de Janeiro, RJ, Brasil.

Duarte, J., Passos, D., Valle, R., Oliveira, E., Muchaluat-Saade, D., and Albuquerque, C. (2007). Management issues on wireless mesh networks. In *5th Latin American Network Operations and Management Symposium (LANOMS 2007)*, Petrópolis.

Huang, F., Yang, Y., and He, L. (2007). A Flow-Based Network Monitoring Framework for Wireless Mesh Networks. *IEEE Wireless Communications*, 14(5).

Jardosh, A., Suwannatat, P., Hollerer, T., Belding, E., and Almeroth, K. (2008). SCUBA: Focus and Context for Real-Time Mesh Network Health Diagnosis. *Lecture Notes in Computer Science*, 4979:162.

Lenczner, M. (2005). Wireless portals with wifidog. *Linux J.*, 2005(140):8.

Muchaluat-Saade, D., Albuquerque, C., Magalhaes, L., Passos, D., Duarte, J., and Valle, R. (2007). Redes em malha: Solução de baixo custo para popularização do acesso a internet no Brasil. In *Simpósio Brasileiro de Telecomunicações (SBrT 2007)*, Recife.

Nanda, S. and Kotz, D. (2008). Mesh-Mon: A multi-radio mesh monitoring and management system. *Computer Communications*, 31(8):1588–1601.

Netflow (2010). Cisco IOS Netflow. www.cisco.com/web/go/netflow. Acessado em Novembro de 2010.

Passos, D., Teixeira, D., Muchaluat-Saade, D., Magalhães, L., and Albuquerque, C. (2006). Mesh network performance measurements. In *International Information and Telecommunications Technologies Symposium (I2TS)*, Cuiabá, MT, Brasil.

Pinheiro, B., de Brito Nascimento, V., Cerqueira, E., Abelém, A., and Neto, A. (2010). Abaré: Um Framework para Implantação, Monitoramento e Gerenciamento Coordenado e Autônomo para Redes em Malha sem Fio. In *XV Workshop de Gerência e Operação de Redes e Serviços - SBRC*, Gramado, RS, Brasil.

Raghavendra, R., Acharya, P., Belding, E., and Almeroth, K. (2009). MeshMon: a multi-tiered framework for wireless mesh networkmonitoring. In *Proceedings of the 2009 MobiHoc S 3 workshop*, New York, NY, USA. ACM.

Sailhan, F., Fallon, L., Quinn, K., Farrell, P., Collins, S., Parker, D., Ghamri-Doudane, S., and Huang, Y. (2007). Wireless mesh network monitoring: Design, implementation and experiments. In *IEEE DANMS workshop*, Washington, DC, USA.

Valle, R., Justen, A., Silva, J.O. Passos, D., Magalhaes, L., Albuquerque, C., and Muchaluat-Saade, D. (2009). Infraestrutura de Comunicação para Linhas de Transmissão de Energia através de Redes em Malha Sem Fio. In *8th International Information and Telecommunication Technologies Symposium (I2TS 2009)*, Florianópolis.

Valle, R., Passos, D., Albuquerque, C., and Muchaluat-Saade, D. (2008). Mesh Topology Viewer (MTV): an SVG-based interactive mesh network topology visualization tool. In *IEEE Symposium on Computers and Communications, 2008. ISCC 2008*, pages 292–297, Marrakesh, Marrocos.

XVI Workshop de Gerência e Operação de Redes e Serviços

♦

Sessão Técnica 2

Desempenho e Qualidade de Serviço, Aprovisionamento de Redes e Planejamento de Capacidade

Optimizing Server Storage Capacity on Content Distribution Networks

Felipe Uderman[1], **Tiago Neves**[1], **Célio Albuquerque**[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niteroi – RJ – Brazil

{uderman,tneves,celio}@ic.uff.br

Abstract. *This work addresses the Storage Capacity Allocation Problem (SCAP), which is closely related to Content Distribution Networks (CDNs) planning and management. The problem consists of determining optimally the fraction of the total storage that must be allocated to each CDN server. A survey of the relevant bibliography on the subject is provided. In this work we have designed and implemented an exact method in order to solve the SCAP problem, and have also evaluated the performance of a CDN before and after the storage capacity optimization. Our simulation results show that an optimally storage capacity configuration has a major impact on the costs related to the delivery of contents.*

1 Introduction

A Content Distribution Network (CDN) is an overlay network of collaborative servers where content replicas are placed and then delivered to its clients [Rajkumar Buyya (2008)]. Because of the increasing amount on the number of requests for contents on the Internet, CDNs have become a popular choice for content providers that want to reach its current and potential clients. This statement is valid especially for multimedia contents, which usually have more severe Quality of Service (QoS) constraints. A CDN is able to place servers, frequently called cache servers or surrogate servers, near its clients, to replicate contents on them and also to process clients' requests for contents, designating which server will handle each request. These functionalities imply on an improvement of content availability and, therefore, result on positives effects on the performance of the service from the perspective of clients, that will experience a reduction on network delay and an improvement on reception rate in most cases.

The current Internet architecture can not enforce acceptable levels of QoS to its users, as it operates on a best effort model. Despite the existence of protocols that are able to provide QoS guarantees on TCP/IP networks, the implementation of such protocols in global scale can be very challenging, due mostly to the distributed management of the Internet and to the ossification of its protocols and services. Still, there is a great demand for improvements on the QoS levels provided by the Internet that motivates the development of alternative solutions to accomplish this objective. Therefore, the main purpose of a CDN is to mitigate current Internet deficiencies and undesirable characteristics, in order to deliver contents to its users with an acceptable QoS. To accomplish this challenging task while still being able to profit, CDN providers must apply optimized mechanisms that aim to reduce its implementation and operational costs.

One important portion of a CDN dimensioning is to determine the storage capacity of its servers. The employment of an over dimensioned storage capacity on CDN servers

may contribute to an improvement on CDN performance. In an over dimensioned scenario, servers are able to hold more contents and therefore, they may handle requests of nearby clients more easily. However, an over dimensioned server storage is obviously not desirable due to the costs involved and to the high probability of under usage of the storage capacity. On the other hand, a sub dimensioned storage capacity will probably lead to lower average performance, since in such condition CDN servers can not hold enough contents to handle most clients' requests with an acceptable performance. In case a content requested by a client is not present on a nearby server, the client will have to wait until either the CDN forwards its request to another server, or until the contacted server retrieves the desired content from another server. As the CDN providers have a limited budget to employ in their physical structure, it is of major importance to develop efficient techniques to aid CDN providers to efficiently dimension the storage capacity of each server in order to achieve an acceptable level of QoS imposed by a set of geographically distributed clients. Therefore, the problem addressed in this paper, known as Storage Capacity Allocation Problem (SCAP), is very relevant for CDN providers that aim to employ efficient infrastructures of collaborative servers.

On this paper, we implement and evaluate a solution for SCAP by modeling it as a Integer Linear Programming Problem (ILPP). The ILPP implemented can obtain the optimal solution for the SCAP on CDNs where any server is able to handle requests from any client. Therefore, upon a request for a content, the server that can perform the delivery more efficiently will be utilised whenever possible. If this particular server does not hold the requested content, the request will be routed to another CDN server that has the desired content and the missing content will be copied to the originally chosen server. It is important to highlight that the ILPP implemented for this particular CDN model can easily be modified to suit other models. We evaluate the effectiveness of optimizing the storage capacity of a CDN, by comparing the performance of test instances with the storage capacity of the servers given by an uniform distribution against instances with optimized storage capacity allocation. Our simulation results demonstrate that a CDN with an optimized storage capacity allocation performs significantly better when compared to CDNs with non optimized storage capacity allocation.

The remaining of this paper is organised as follows: In Section 2 we analyse the SCAP, focusing on the necessary input data to solve this problem and also on possible variations regarding its mathematical modeling and expected results. In Section 3 we review relevant related works available on the literature. In Section 4 a mathematical model to solve the SCAP is presented and analysed. In Section 5 we detail the test scenarios, and also introduce the mechanism used to evaluate the effectiveness of the storage capacity allocation on CDNs. In Section 6 we present and analyse our simulation results. We conclude our work on Section 7, by consolidating the results obtained and proposing possible future works.

2 The Storage Capacity Allocation Problem (SCAP)

Many of the challenges faced by CDN providers can be modeled as optimization problems, whose solutions can be used to assist on the dimensioning of CDN resources and on the development of efficient algorithms and strategies to perform content replication and delivery. There are three fundamentals optimization problems related to CDNs dimensioning and operation:

- **Server Location Problem (SLP):** This problem consists on determining the optimal location of the available cache servers in order to better server the clients [Krishnan (2000)].
- **Request Distribution Problem (RDP):** This problem consists on determining optimally which server must be responsible for delivering the content to each client [Wang (2002), Shaikh (2001)].
- **Replica Placement Problem (RPP):** Due to capacity constraints, the cache servers available don't have enough storage capacity to hold all the contents of the CDN system thus, the RPP consists on determining optimally which contents must be stored at each server on each time period, and also on determining the servers that will be used as seeds during the replication process [Khan (2009)].

The problem stated on this work, known as the Storage Capacity Allocation Problem (SCAP), is a variant of the SLP. The SCAP consists on optimally determining the distribution of the total storage capacity available among the surrogate servers of the CDN. Solutions for the SCAP require input data that describes characteristics and limitations of the CDN, such as the total storage capacity available, the communication costs between each element of the CDN and the CDN topology and operational model. Besides that, information about the clients of the CDN are required in order to determine the relative popularity of each content among the clients and also how many requests for contents each client will perform over the time periods analysed.

Besides determining the storage capacity allocation for each server, some versions of the SCAP can also determine, in a static manner, which server should handle each request and also which contents should be placed on each server, characteristics related to the RDP and RPP, respectively. However, as dynamic approaches for the RDP and the RPP have been proved to be effective [Neves (2010)], these static additional results provided by the SCAP should only be utilised to determine the initial location of content replicas and request distribution, leaving to dynamic mechanisms the solution of the RDP and RPP during the CDN operation phase.

Although there is a decreasing tendency on the costs of storage for servers, the SCAP can become a major issue on future scenarios, because of the rapid growth of the demand for content on the Internet, especially for multimedia content. Some kinds of multimedia content, like high resolution videos, which still have a low availability on current Internet, have a considerable size when compared to ordinary contents, and can easily exhaust the storage capacity of CDN servers if they become popular and widespread. The small availability of high speed connections on the last mile of the Internet is still the main cause of the low popularity of services specialised in delivering high resolution multimedia content. But on scenarios where the majority of the end users have access to high speed connections, the bottleneck of those services will become the availability of contents on the servers. As centralized approaches to content delivery has severe scalability problems and P2P networks can not ensure QoS guarantees, we reinforce the importance of the SCAP on the planning of CDN infrastructures that aim to support an efficient delivery of widespread high storage and bandwidth demanding contents over the Internet.

An important issue of the SCAP is that it is essentially an offline problem, meaning that all the information required to solve it must be a priori available. Thus, some input parameters of the SCAP that have a dynamic behaviour, such as the communication costs between the servers, magnitude of contents demands and number of clients, must be ei-

ther accurately estimated or extracted from some data set that describes a CDN operation scenario. This work follows the second approach, by extracting the required information from test instances for the Replica Placement and Request Distribution Problem (RPRDP), a joint problem of the RDP and the RPP, which deals with many issues related to the operation of a CDN simultaneously. Thus, as the mathematical modeling of the SCAP benefits from this kind of information, the results obtained must be considered as theoretical bounds of performance, as in practice, the information required to compute the input parameters for the SCAP can not be exactly obtained.

Another limitation is that traditional solutions for the SCAP will distribute the available storage capacity in a very granular fashion [Laoutaris (2005)], ranging from a single store unit to content size increments. This means that, even with all the input parameters at hand, the SCAP will generate an optimal storage capacity distribution that can not be implemented in practice, as real devices storage capacity is available in few discrete sizes, usually with a considerable amount of storage units apart from each other. Even so, the optimal results generated by SCAP can easily be used to aid the choice of the best available server capacity for each location.

A realistic scenario, where is possible to use the SCAP results accurately, is the one where the CDN servers are implemented as virtual machines that shares computational resources with other applications and services. On this scenario, due to virtualization of the servers, it is easier to allocate storage capacity for each CDN content server with a high granularity. In addition, it would be possible to repeatedly use the static model of the SCAP to solve the dynamic version of the problem, on which the storage capacity of each virtual server could be adapted to provide the most suitable storage capacity distribution at each time period. Even on this dynamic scenario, the SCAP could be modeled as an offline ILPP that would provide an optimal solution based on the information available for all the time periods, providing a theoretical bound on the performance for this dynamic version of the problem. However, the most reasonable approach for this dynamic version is to implement an efficient method to solve the problem on an online fashion, where the storage capacity distribution must be determined, for each time period, based solely on information of the current and previous time periods.

3 Related Works

In this section, previous publications that have guided the development of this work are analysed. On [Laoutaris (2005)], the authors present the SCAP for CDNs with hierarchical topologies. Three mathematical models are proposed for different versions of the SCAP, along with greedy heuristics that have archived results close to optimum on the simulations performed. Due to the high computational complexity of the models, the LP relaxed version of the models are solved instead of the original ILPPs. The first model, suited for CDNs with a strong hierarchical relationship between its servers, is the simplest one as it does not consider additional features and limitations that could improve the model verisimilitude. The following models include a load balance constraint, that limits the maximum number of requests per unit time that may be serviced by each server, and a request peering capability, that allows CDN servers to redirect requests to other servers that belong to its same hierarchical level. The addition of the load balance constraint increases the total cost of the solution, but causes the usage of the different CDN servers be more uniform. The peer requesting feature reduces significantly the total cost of the solution,

specially when the communication cost between peer servers is low.

On [Li (1999)], the authors address the SLP as a dynamic programming problem, with the objective of determining the optimum location of web proxy servers on the Internet while minimising the delay experienced by clients when locating and accessing contents on the CDN. Although the SLP can solve the problem of the location of the surrogate servers on a CDN, no results can be obtaining on the dimensioning of each server storage capacity, which represents a drawback of this model. Besides, the presented scenario supposes that all network links have a homogeneous capacity and the existence of web proxy servers for a CDN with only one original server, characteristics that do not match with realistic scenarios.

On [Qiu (2001)], the authors have proposed several heuristic algorithms to solve the SLP, and have evaluated their performance over synthetic and real network topologies derived from BGP routing tables. Among the heuristic algorithms proposed, a greedy Algorithm that aims to individually determine which location is better suited to place a cache server on each interaction has archived the best results. The authors also comment about practical issues of obtaining the input data necessary to solve the SLP on realistic scenarios. Although it is relatively simple to obtain accurate information about the network topology being analysed, it is not so easy to obtain information of some parameters accurately, such as the performance of the communication links and the amount of clients requests for contents, as those parameters have a very dynamic nature. Nevertheless, simulation results show that the placement algorithms proposed are relatively insensitive to errors on the estimate of those parameters.

On [Wu (2009)], the authors also address the SLP, but their solution is based on a genetic algorithm approach. A numeric solution for the same network topology analysed on [Li (1998)] is presented, along with a performance comparison of the genetic algorithm approach with the greedy algorithm proposed on [Qiu (2001)]. The greedy algorithm was chosen for performance evaluation because it outperforms the other algorithm proposed on [Qiu (2001)] and the dynamic programming algorithm proposed on [Li (1998)]. The numeric results demonstrate that, for a number of available servers greater than two, the genetic algorithm can achieve better results then the greedy algorithm. However, like other similar works on this issue, no consideration about the dimensioning of the servers capacity is made. Beside, as the numeric results were computed only for one network topology with homogeneous network links capacity, it is not possible to conclude that the genetic algorithm proposed will have a good performance for other network topologies. Indeed, the authors shows that the performance gain of the genetic algorithm proposed, when compared to the greed solution of [Qiu (2001)], can vary significantly when different sets of request rates are assigned to clients.

On [Li (2008)], the authors address the RPP on the context of hierarchical topologies. The optimal solution for the formulated RPP is obtained by solving a dynamic problem. The simulations performed compare the performance of several replica positioning protocols on scenarios with a variable total storage available and also with topologies with variable hierarchical levels. The results suggest that a correct dimensioning of the total storage capacity on a CDN is an important parameter for performance. It is also possible to conclude that the number of hierarchical levels of a CDN should not be superior to four because, despite the moderate increase of the cache hit ratio on hierarchical CDN

topologies with more levels, there is a huge increase on the average delay observed by the clients due to the increase on the difficulty for locating contents on servers. Furthermore, simulations performed with different values of the parameter of the distribution used to model the content popularity on the CDN show that scenarios in which clients' demands are concentrated on few contents are advantageous to the proposed solution.

By analysing the referred literature, it is possible to conclude that related works that addressed storage allocation issues on CDNs [Laoutaris (2005), Li (1998), Wu (2009)] have not extrapolated their results by evaluating the performance of a CDN operation when its total storage capacity is optimally distributed or when the network nodes that hold surrogate servers are optimally chosen. Indeed, those works limit the scope of their analysis to the performance of the SCAP or SLP solutions proposed, by analysing the computational complexity of the solution or by proposing efficient heuristics algorithms to solve those problems. We believe that, by evaluating the impact of allocating optimally the storage capacity of surrogate server on the performance of a CDN operation, we can provide a contribution to the scientific community and to CDN providers as well.

4 Mathematical Modeling

The SCAP can be formally defined as follows [Laoutaris (2005)]: let S be the total storage capacity available, φ the set of N unique unit sized contents, J the set of m clients, each client j having a request rate λ_j and a object demand distribution $p_j(k) \rightarrow [0, 1]$, V the set of n servers, $d_{j,v} : J \times V \rightarrow R^+$ the communication cost between the clients j and the node v, and C the set of all the possible node-objects pairs $(v, k), v \in V, k \in \varphi$. The SCAP consists on determining a subset $A \subset C$ with no more then S elements that maximize the gain obtained when a client of the CDN receives a content from a surrogate server instead of receiving it directly from the content server.

The mathematical modeling of SCAP must consider parameters related to the RPRDP to ensure that the storage capacity allocation is optimal to a given CDN model. For example, on a CDN operational model that bounds each client to a local server, it makes no sense to consider the possibility that other CDN servers might be able to deliver contents to then. Similarly, on a CDN with hierarchical topology, only a subset of servers are able to deliver contents to each client. The CDN model utilised to evaluate our results operates with a full-mesh topology, where every server of the CDN is able to deliver contents to a given client.

The SCAP can be modeled as ILPP in order to compute the optimal solution of the problem. Firstly, it is necessary to define two variables:

$$X_{j,v}(k) = \begin{cases} 1 & \text{, if the client } j \text{ receives the content } k \text{ from the server } v \\ 0 & \text{, otherwise} \end{cases} \quad (1)$$

$$\delta_v(k) = \begin{cases} 1 & \text{, if } \sum_{j \in J} X_{j,v}(k) > 0 \\ 0 & \text{, otherwise} \end{cases} \quad (2)$$

The variable $X_{j,v}(k)$, defined on Equation 1, implies on whether the content k will be delivery by the server v to the client j and the variable $\delta_v(k)$, defined on Equation 2,

implies on whether the content k will be present on the server j. By computing $\delta_v(k)$, it is possible to obtain the storage capacity allocated on each node of the CDN. The ILPP adapted for solving the SCAP for full-mesh CDN topologies is defined as:

Max:

$$\sum_{j \in J} \lambda_j \sum_{k \in \varphi} p_j(k) \sum_{v \in V} (d_{j,os(k)} - d_{j,v}) X_{j,v}(k) \tag{3}$$

S.A.:

$$\sum_{v \in V} X_{j,v}(k) \leq 1 \tag{4}$$

$$\sum_{j \in J} X_{j,v}(k) \leq U \cdot \delta_v(k) \tag{5}$$

$$\sum_{v \in V} \sum_{k \in \varphi} \delta_v(k) \leq S \tag{6}$$

$$\sum_{k \in \varphi} \delta_v(k) \geq |\{k : OS(k) = v\}| \tag{7}$$

The Equation 3 is the objective function of the ILPP and consists on maximizing the gain obtained when storage space for content k is allocated on server v. The gain is proportional to the difference of the network distances from client j to the content original server and from client j to server v. This means that more storage capacity will be allocated on servers that have a high potential of reducing the network load or improving the QoS perceived by the users, depending of the distance metrics applied. The next term of Equation 3 is relative to the request rate of client j. The request rate should represent the amount of requests originated on of client j. The number of users on a client site can be utilised to derive the request rate, but it is likely that different client sites will have users with different activity profiles. Finally, the request distribution component takes into consideration that each content will have a different popularity on different client sites.

Equations 4, 5 and 6 are the constrains that must be observed in order to keep the consistency of the solution. Equation 4 states that a client j must receive each content k from a single server v. This assumption is necessary to ensure that the storage capacity allocation will consider the best server to address each request, but it is the responsibility of the RPRDP method implemented to dynamically make this decision. Equation 5 states that a client j can only receive a content k from a server v if this server have storage capacity allocated for this content, and also that the number of clients that receives content k from server j must not exceed the total number of clients of the CDN. Equation 6 states that the amount of storage capacity allocated must not exceed the total storage capacity available. Equation 7 states that the amount of space allocated on a given server must be enough to place at least the contents that originate on that CDN server. This avoids inconsistencies while processing the RPRDP, as at its initial period, there must be enough storage space on each server to accommodate the contents.

Table 1. Relevant RPRDP parameters

Parameter	Description
Content size	Uniformed distributed between 250MB and 400MB
Content origin Server	Random
Server capacity	Uniformed distributed between 3000MB and 4000MB
Contents popularity	Zipf distributed, $\alpha = 0.7$[Adamic (2002)]
Requests local server	Zipf distributed, $\alpha = 0.7$
Simulation periods	25
Request arrival time periods	Time periods 0 to 14

5 Test Scenarios

The simulations performed aim at identifing improvements on the performance of a CDN with an optimised storage capacity allocation, in contrast with a CDN with an uniformly distributed storage capacity allocation. To accomplish this task, we have generated test instances for the SCAP by extracting the necessary information from test instances for the RPRDP [Neves (2010)]. With the SCAP results, we have generated new test instances for the RPRDP with an optimized storage capacity allocation, keeping all the other parameters unchanged. This means that differences in performance observed between the original and optimised instances of the RPRDP must be credited solely to the new storage scheme.

The RPRDP is a typical operation scenario of a CDN, where it is necessary to dynamically define which contents will be stored on each servers and also which server will attend each client request for contents. Therefore, it is possible to extract from these instances the necessary information necessary to model the SCAP, since the RPRDP instances describe the requests for contents events, which can be used to obtain $p_j(k)$ and λ_j, besides other necessary parameters such as total storage capacity available, amount of contents and communication costs between the clients and the servers. We have created a total of 140 instances for the RPRDP, segmented in groups of 10, 20, 30 and 50 servers. Some of the parameters of the original RPRDP instances that are relevant for the SCAP are described on Table 1.

The objective function of the RPRDP computes the costs associated with the delivery of contents to the CDN clients and with the replication of contents among the CDN servers, as it can be observed on Equation 8. The delivery cost is associated with the network delay between a client and a server, at the time period a content is being delivered. This parameter models the user perceived QoS, as a lower network delay will result on a faster and more reliable delivery. The replication cost is associated with how much data is being replicated among the CDN servers. This parameter models the CDN provider costs with their communication infrastructure. If less content needs to be replicated, it means that the CDN providers will spend less resources on their communication infrastructure. There is also a backlog term on the RPRDP objective function, that models situations where a requested content can not be delivered to a client upon a request. Therefore, the backlog represents the amount of data that could not be delivered to the client, being essentially a measurement of the client satisfaction to the service provided by the CDN. We have verified that the backlog is caused by lack of bandwidth with our RPRDP implementation. As

our test instances have enough bandwidth to attend the clients requests, no backlog cost was observed on our simulations. The RPRDP minimizes the following cost function:

$$Cost = \sum_{i \in R} \sum_{j \in S} \sum_{t \in T} c_{ijt} x_{ijt} + \sum_{i \in R} \sum_{t \in T} p_{it} b_{it}$$

$$+ \sum_{k \in C} \sum_{j \in S} \sum_{l \in S} \sum_{t \in T} L_k w_{kjlt} \quad (8)$$

Where:

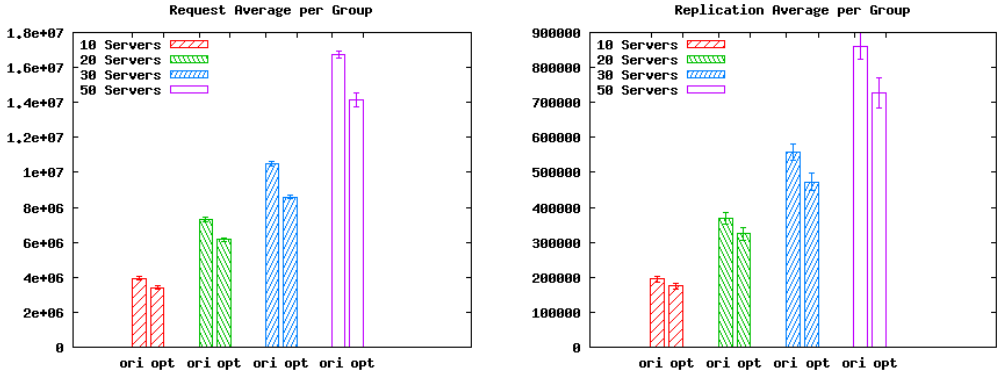
- c_{ijt}: Cost of delivery of the content k requested on i by the server j, on the time period t
- x_{ijt}: Fraction of the content requested on i delivered by server j, on the time period t
- p_{it}: Backlog penalty of request i on time period t
- b_{it}: Backlog amount of request i on time period t
- L_k: Size of content k
- w_{kjlt}: 1, if content k is replicated from server l to server k on the time period t; 0, otherwise
- $i \in R$: Set of requests
- $j, l \in S$: Set of servers
- $t \in T$: Set of time periods
- $k \in C$: Set of contents

6 Simulation Results

In this section the simulation results of evaluating the performance gain of a CDN with an optimally distributed storage allocation in contrast with a CDN with an uniform distributed storage allocation are presented. The results of the 35 instances of each group of 10, 20, 30 and 50 servers were summarised for better visualisation.

Figure 1 shows costs averages for each group of instances along with the 95% confidence interval. For all groups of instances, the optimization of the storage allocation have reduced significantly both the replication and the delivery costs of the RPRDP. This means that both the CDN provider and its clients can benefit from an optimized storage capacity allocation, even if this optimization is performed statically. As the request cost is related to the network distance between the client and the CDN server assigned for content delivery, a lower request cost average means that the client would experience a faster and more reliable delivery. The replication cost is related with how much data must transfer among the CDN servers, and it is related to the costs of the internal infrastructure of the CDN. Therefore, a lower replication cost represents savings for the CDN provider, as less resources would be spent on its infrastructure.

Another improvement achieved is the increase of the cache hit ratio on the local CDN server, as can be observed on Figure . For each CDN client, it is assigned a local server that is closer to it and therefore must be the first choice for content delivery. An improved cache hit ratio on the local server also benefits both the client experience and the CDN provider costs, as delivering a content from a remote server will deteriorate the user



(a) Request (b) Replication

Figure 1. Cost comparison between optimized and uniform distributed storages allocation

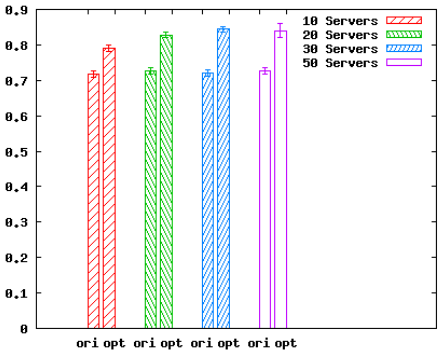
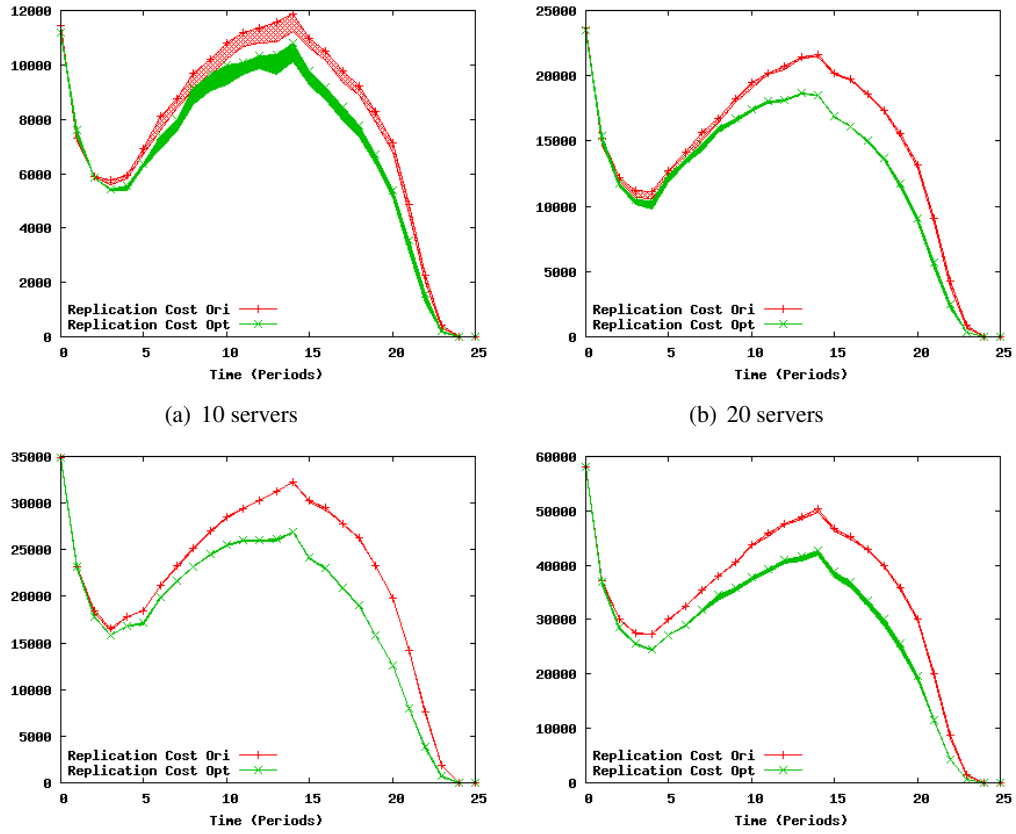


Figure 2. Cache Hit Ratio on the local server

perceived quality of delivery and also will represent more data transit on the CDN internal infrastructure.

Figure 3 shows the replication costs for each group of instances summarised by each time period of simulation. Both the original and optimised instances follows the same behaviour, but the optimized instances perform about 10% to 20% less replication over almost all the time periods of simulation. At the very begin of simulation, it is necessary to perform a high amount of replication, as the CDN servers are not yet populated with contents. Therefore, the replication cost is high at the very begin of the simulation and keeps dropping until around period 4. Beyond this period, as the number of active requests on the CDN keeps rising, more replication is performed in order to better address the requests. This rise is observed until the end of request generation on time period 14, where it is possible to observe the higher amount of replication performed. After the end of new requests generation, the number of active requests keeps droping, and less replication is necessary until the last request is finally completed around time period 24.

On Figure 4, it is possible to observe the delivery cost for each time period of simulation. At the begining of the simulation, the contents have not yet been replicated among the CDN servers. Therefore, the content delivery is performed from remote servers, increasing the delivery cost. After content replication occurs, the delivery cost drops for a few time periods, but the accumulation of active requests makes the delivery cost rise until around time period 14, when the request generation ends. After this time period, the



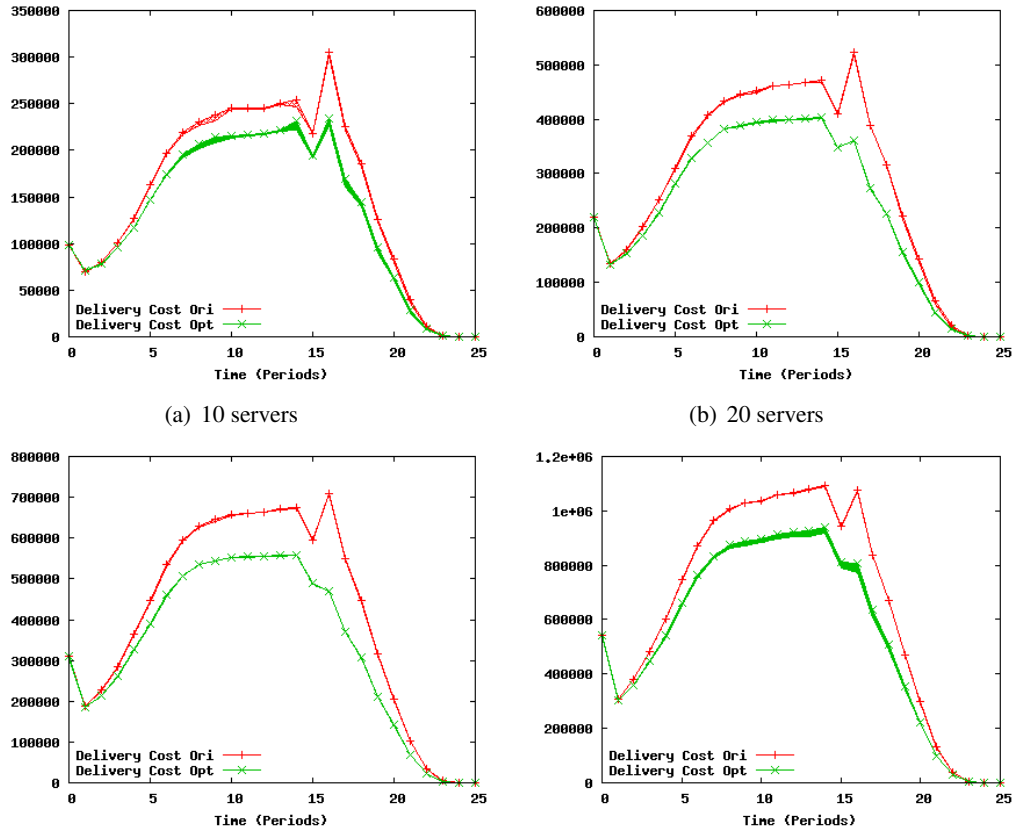
(a) 10 servers (b) 20 servers

(c) 30 servers (d) 50 servers

Figure 3. Replication Costs over time comparison

delivery request cost starts to drop, but around time period 17 there is a sudden increase on the delivery cost for the non optimized instances. On the optimized instances, even when this spike is not observed on all of the optimized instances groups, it is still possible to observe an attenuation on the dropping rate. As can be observed on Figure 5, this behaviour can be explained by a drop on the cache hit ratio. Less requests are addressed by the local servers, what impacts the delivery costs even with less active requests present on the CDN.

Figure 5 presents the cache hit ratio on the local server for each time period of simulation. At the initial time period of simulation, the cache hit ratio is very poor for both optimized and non-optimized instances, because the contents have not yet been replicated among the CDN servers. After replication occurs, the cache hit ratio improves fast, but as the number of active requests on the CDN rises, it is not possible to maintain the cache hit ratio on the local server at a high level. It is also possible to observe a drastic drop on the cache hit ratio around time period 17. This behaviour can be explained by the simplicity of the replication heuristic utilised to evaluate the storage capacity optimization performed. We have observed that, as the removal content policy of the replication heuristic does not consider the amount of requests for a given content, there are many requests for contents whose replicas have been removed on previous time periods. Nevertheless, this behaviour is much more evident on the non-optimized instances.



(a) 10 servers (b) 20 servers

(c) 30 servers (d) 50 servers

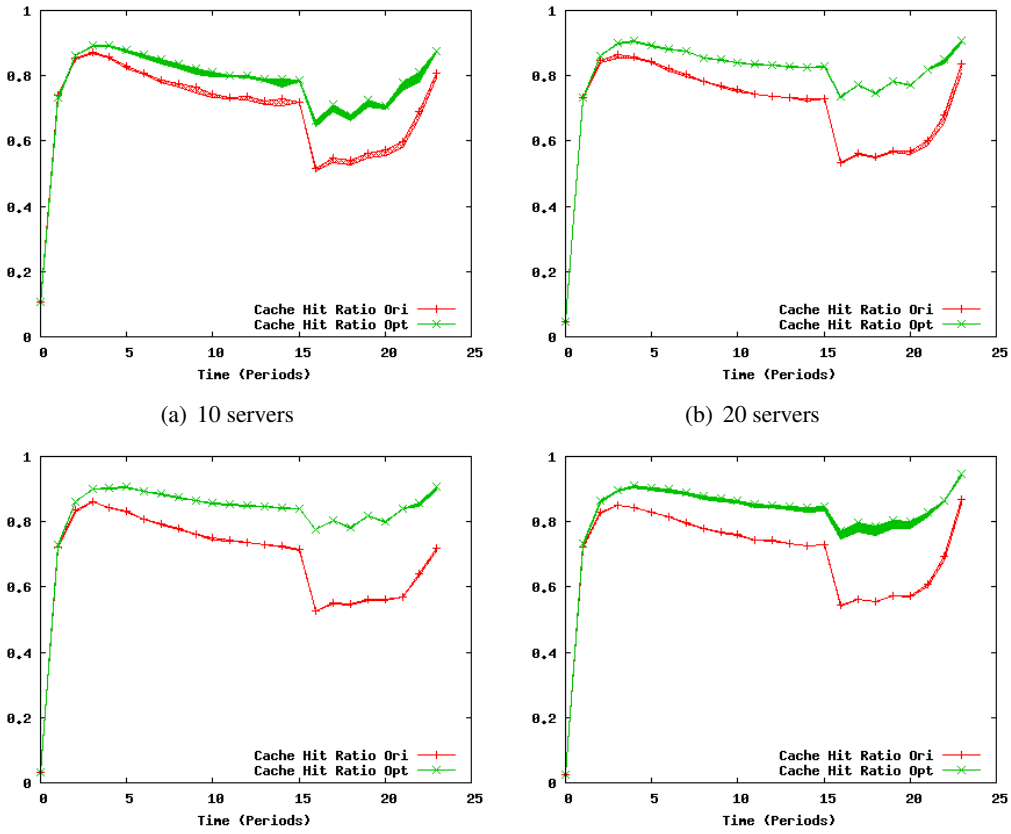
Figure 4. Delivery Costs over time comparison

7 Conclusion and Future Works

On this paper, we have formulated a model to solve the SCAP statically for CDNs with a full-mesh topology. We have presented results that show how a CDN can benefit from a optimized storage capacity allocation among it servers. Reducing its replication and delivery costs is of major importance for CDN providers, as this will reduce their operational costs and improve the satisfaction level of their clients. Also, an improved cache hit ratio on the local server avoids unnecessary transit of data on the networks and ensures that the CDN client will receive the requested content from the closer server.

A granular allocation unit model makes sense on a future Internet scenario where CDN servers are implemented as virtual machines. In such scenario, allocating the storage capacity with high granularity is not only possible, but desirable to avoid the waste of hardware resources. When working on virtual machines scenarios, it is also possible to implement a dynamic approach to storage allocation, where the storage capacity of each CDN server can be increased or decreased on demand. This approach would certainly lead to improved results, as it would be easier to accommodate a sudden increase on the demand for contents without increasing the delivery cost. It is also possible to easily use the high granularity storage capacity allocation results to select the most suited available servers.

As a future work, we plan to improve the SCAP mathematical model in order to contemplate economic costs and benefits related to the CDN business. On this work we have demonstrated that an statically optimized storage capacity allocation can reduce the



(a) 10 servers (b) 20 servers

(c) 30 servers (d) 50 servers

Figure 5. Cache Hit Ratio over time comparison

amount of data replicated on a CDN and also can reduce the client perceived network delay while retrieving contents. We believe that by analysing potential economic gains and costs of a CDN provider, we can build a mathematical model that could determine if it is worth for a CDN provider to build an infrastructure to deliver content to a remote site, while determining how much storage capacity must be allocated on each potential point of presence of the CDN.

Acknowledgments

The authors would like to thank Faperj, CNPq and Petrobras for their support on the development of this work.

References

Adamic, L. A. and Huberman, B. A. (2002). Zipf's law and the Internet. *Glottometrics*, 3:143–150.

Khan, S. U., Maciejewski, A. A., and Siegel, H. J. (2009). Robust cdn replica placement techniques. In *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, pages 1–8, Washington, DC, USA. IEEE Computer Society.

Krishnan, P., Raz, K. D., and Shavitt, Y. (2000). The cache location problem. *IEEE/ACM Transactions on Networking*, 8:568–582.

Laoutaris, N., Zissimopoulos, V., and Stavrakakis, I. (2005). On the optimization of storage capacity allocation for content distribution. *Comput. Netw.*, 47(3):409–428.

Li, B., Deng, X., Golin, M. J., and Sohraby, K. (1998). On the optimal placement of web proxies in the internet: The linear topology. In *HPN '98: Proceedings of the IFIP TC-6 Eigth International Conference on High Performance Networking*, pages 485–495, Deventer, The Netherlands, The Netherlands. Kluwer, B.V.

Li, B., Golin, M., Italiano, G., Deng, X., and Sohraby, K. (1999). On the optimal placement of web proxies in the internet. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1282 –1290 vol.3.

Li, W., Chan, E., Wang, Y., Chen, D., and Lu, S. (2008). Cache placement optimization in hierarchical networks: analysis and performance evaluation. In *NETWORKING'08: Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet*, pages 385–396, Berlin, Heidelberg. Springer-Verlag.

Neves, T. A., Drummond, L. M. A., Ochi, L. S., Albuquerque, C., and Uchoa, E. (2010). Solving replica placement and request distribution in content distribution networks. *Electronic Notes in Discrete Mathematics*, 36:89–96.

Qiu, L., Padmanabhan, V., and Voelker, G. (2001). On the placement of web server replicas. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.*

Rajkumar Buyya, Mukaddim Pathan, A. V. (2008). *Content Delivery Networks*. Springer.

Shaikh, A., Tewari, R., and Agrawal, M. (2001). On the effectiveness of dns-based server selection. In *In Proceedings of IEEE Infocom*.

Wang, L., Pai, V., and Peterson, L. (2002). The effectiveness of request redirection on cdn robustness. *SIGOPS Oper. Syst. Rev.*, 36:345–360.

Wu, J. and Ravindran, K. (2009). optimization algorithms for proxy server placement in content distribution networks. In *IM '09: Integrated Network Management-Workshops*, pages 193–198, New York, NY, USA.

Uma Função de Cálculo de Tamanho de *Frames* para o Protocolo DFSA em Sistemas RFID

Júlio D. de Andrade, Paulo André da S. Gonçalves

Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
50.740-560 – Recife – PE – Brasil

{jda, pasg}@cin.ufpe.br

Abstract. *DFSA (Dynamic Framed Slotted ALOHA) is a popular anti-collision protocol for RFID systems. In such protocol, the size of each frame succeeding the first frame is dynamically calculated based on the estimate of the number of competing tags in the previous frame. The main existing estimators for DFSA seek to increase their accuracy in order to minimize the total identification time of tags. However, such estimators do not consider the impact of the Early-End extension when calculating frame sizes. In this paper, we show that there are missing optimization opportunities in the total tag identification time when DFSA uses such an extension but its impact is ignored in the frame size calculus. Based on this fact, we propose a function to calculate frame sizes but taking into account the estimator's estimate and the impact of the Early-End. Performance evaluations of the proposed function with the Eom-Lee, Vogt, and Schoute estimators show that the total tag identification time is reduced, respectively, up to 24%, 20%, and 29% when compared to that observed by using the estimator itself.*

Resumo. *O DFSA (Dynamic Framed Slotted ALOHA) é um protocolo anticolisão popular para sistemas RFID. Nesse protocolo, o tamanho de cada frame subsequente ao frame inicial é calculado dinamicamente com base na estimativa da população de etiquetas que competiram no frame anterior. As principais propostas de estimadores para o DFSA procuram maximizar a acurácia de suas estimativas na busca da minimização do tempo total de identificação de etiquetas. Contudo, elas desconsideram o impacto do uso da extensão Early-End no cálculo do tamanho dos frames. Este artigo mostra que oportunidades de otimização do tempo total de identificação de etiquetas são perdidas ao se desconsiderar o impacto dessa extensão no cálculo de tamanho de frames quando a mesma é utilizada no DFSA. A partir disso, este artigo propõe uma função de cálculo de tamanho de frames que considera a estimativa do estimador usado e o impacto do uso da extensão Early-End. As avaliações de desempenho da função proposta com os estimadores Eom-Lee, Vogt e Schoute mostram que o tempo total de identificação de etiquetas é reduzido, respectivamente, em até 24%, 20% e 29% quando comparado ao tempo obtido com o uso do estimador isoladamente.*

1. Introdução

Os sistemas RFID (*Radio Frequency IDentification*) são os mais promissores para a identificação automática de objetos através de sinais de radiofrequência. Em geral, os

sistemas RFID mais básicos são compostos por um *leitor* e várias *etiquetas*. Cada etiqueta armazena um identificador (ID) único e é colada ou embutida em um objeto. No processo de identificação, o leitor requisita o ID das etiquetas que se encontram em seu alcance de comunicação. Contudo, é possível que duas ou mais etiquetas transmitam informações ao mesmo tempo ao longo desse processo. Nesse caso, ocorre uma colisão de sinais, impedindo que o leitor reconheça as informações enviadas. Assim sendo, se faz necessária a utilização de um protocolo anticolisão de etiquetas a fim de se resolver os conflitos de transmissão e permitir uma rápida identificação de todos os objetos.

O problema em questão é um caso especial do problema de controle de acesso ao meio em redes sem fio, pois introduz um novo desafio conforme explicitado a seguir: as etiquetas RFID geralmente possuem limitações importantes de poder computacional, de memória, de custo e de consumo de energia. Dessa forma, torna-se irreal assumir que elas poderiam trocar mensagens para a reserva do meio de comunicação, ou ainda, assumir que elas poderiam avaliar o uso do canal de comunicação antes de qualquer transmissão a fim de se minimizar as chances de colisão. Portanto, existe a necessidade de se desenvolver e adotar protocolos anticolisão de etiquetas específicos para sistemas RFID [Klair et al. 2010].

Dentre as diversas propostas de protocolos anticolisão de etiquetas, o protocolo DFSA (*Dynamic Framed Slotted ALOHA*) vem recebendo recentemente grande atenção na literatura [Eom and Lee 2010] [Tong et al. 2009] [Chen 2009]. Nesse protocolo, o leitor organiza o tempo em um ou mais *frames*, onde cada *frame* está subdividido em *slots*. As etiquetas são requisitadas a transmitir em um *slot* a cada *frame* até que sejam identificadas pelo leitor. O número total de *slots* de cada *frame* subsequente ao *frame* inicial é calculado dinamicamente com base na estimativa da população de etiquetas que competiram por *slots* no *frame* precedente. Um dos problemas existentes é como estimar tal população de etiquetas da forma mais precisa possível e relacionar a estimativa obtida em uma função de cálculo de tamanho de *frames*.

Em geral, as propostas de estimadores para o DFSA procuram maximizar a acurácia de suas estimativas na busca da minimização do tempo total de identificação de etiquetas. Contudo, elas desconsideram o impacto do uso da extensão *Early-End* no cálculo do tamanho dos *frames*. Tal extensão é comumente usada em sistemas RFID e permite que a duração de *slots* vazios seja menor do que a duração de *slots* em colisão e bem sucedidos. Este artigo mostra que oportunidades de otimização do tempo total de identificação de etiquetas são perdidas ao se desconsiderar o impacto dessa extensão no cálculo de tamanho de *frames* quando a mesma é utilizada no DFSA. A partir disso, este artigo propõe uma função de cálculo de tamanho de *frames* que considera a estimativa do estimador usado e o impacto do uso da extensão *Early-End*. As avaliações de desempenho da função proposta com os estimadores Eom-Lee [Eom and Lee 2010], Vogt [Vogt 2002] e Schoute [Schoute 1983] mostram que o tempo total de identificação de etiquetas é reduzido, respectivamente, em até 24%, 20% e 29% quando comparado ao tempo obtido com o uso do estimador de forma isolada.

O restante deste artigo está organizado como segue: a Seção 2 apresenta os trabalhos relacionados. O desempenho dos principais estimadores para o DFSA é avaliado e analisado na Seção 3. A Seção 4 apresenta a função de cálculo de tamanho de *frames* proposta neste artigo. As avaliações de desempenho do DFSA com a função proposta são

apresentadas na Seção 5. Finalmente, a Seção 6 apresenta as conclusões deste trabalho.

2. Trabalhos Relacionados

Atualmente, existem diversos estimadores para o DFSA propostos na literatura como: o *Lower Bound* [Vogt 2002], o Schoute [Schoute 1983], o Vogt [Vogt 2002], o Eom-Lee [Eom and Lee 2010], o Chen [Chen 2009] e o Tong [Tong et al. 2009]. Todos, exceto o *Lower Bound*, focam apenas na melhoria da acurácia das estimativas na busca da minimização do tempo total de identificação de etiquetas. Como será apresentado adiante, o objetivo do *Lower Bound* é obter apenas uma estimativa grosseira para análises de pior caso. Este artigo foca no estimador mais recente proposto na literatura – o Eom-Lee – e nos estimadores clássicos *Lower Bound*, Schoute e Vogt. As principais conclusões deste artigo também são aplicáveis aos estimadores Chen e Tong.

Os estimadores estudados neste artigo serão detalhados nas próximas seções. Antes disso, considere a Figura 1, na qual os *frames* de interesse no DFSA estão representados: o *frame* que acabou de ser finalizado (*frame finalizado*) e o seu *frame* imediatamente posterior (*frame posterior*), cujo tamanho precisa ser calculado. No *frame finalizado*, s_v, s_s e s_c representam, respectivamente, a quantidade *slots* vazios, a quantidade de *slots* bem sucedidos e a quantidade de *slots* em colisão. O tamanho do *frame posterior* é representado por $\hat{f}$. Para todos os estimadores descritos neste artigo, $\hat{n}$ representa a estimativa do número de etiquetas que competiram no *frame finalizado*.

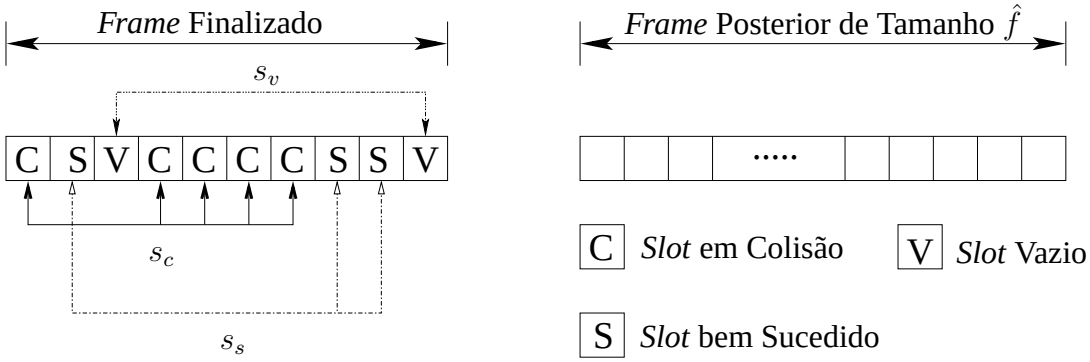


Figura 1. *Frames* de interesse no DFSA.

2.1. Lower Bound

O estimador *Lower Bound* foi inicialmente proposto em [Vogt 2002]. A ideia por trás desse estimador é simples: há ao menos duas etiquetas envolvidas em uma colisão. Então, a menor quantidade possível de etiquetas que competiram em um *frame finalizado* é igual a quantidade total de *slots* com transmissões bem sucessidas somada com o dobro da quantidade de *slots* em colisão. A quantidade total estimada de etiquetas que restam a ser identificadas (*backlog*) é simplesmente o dobro da quantidade de *slots* em colisão. Dessa forma, esse é o valor do tamanho do próximo *frame* a ser utilizado em busca da maximização da eficiência do sistema [Schoute 1983]. Portanto, o estimador *Lower Bound* define:

$$\hat{n} = s_s + 2 \cdot s_c \,, \tag{1}$$

$$\hat{f} = 2 \cdot s_c \, . \tag{2}$$

Teoricamente, bons estimadores não deveriam ter erros de estimação piores do que os do *Lower Bound*, pois os valores estimados por este último para a quantidade de etiquetas restantes são sempre os menores dentro de uma gama de possibilidades. Por causa disso, o estimador *Lower Bound* é comumente utilizado em avaliações de desempenho do DFSA para fins de comparação com a qualidade de estimação de outros estimadores.

2.2. Schoute

Schoute [Schoute 1983] computa o tamanho do *frame posterior* como sendo o resultado da multiplicação do número total de *slots* em colisão no *frame finalizado* por um fator igual a 2,39. Esse fator é o número esperado de etiquetas que transmitirão em cada *slot* em colisão no *frame posterior*. Assim sendo, o método de Schoute define:

$$\hat{f} = 2,39 \cdot s_c \, . \tag{3}$$

A Equação (3) é obtida considerando-se um processo de chegadas do tipo Poisson, sendo ela uma aproximação subótima caso o processo de chegada de pacotes respeite outras distribuições. Para se obter a estimativa $\hat{n}$, basta somar $\hat{f}$ com o número total de etiquetas identificadas com sucesso no *frame finalizado*. Logo,

$$\hat{n} = s_s + 2,39 \cdot s_c \, . \tag{4}$$

2.3. Vogt

O método de Vogt [Vogt 2002] considera que a alocação de transmissões de etiquetas em um *slot* é um problema de ocupação. Tal problema lida com a alocação aleatória de "bolas" em uma certa quantidade de "sacos", onde as "bolas" podem ser vistas como etiquetas e os "sacos" como *slots* em um *frame* de interesse. Dado que existem L *slots* em um *frame* e n etiquetas que competiram por *slots* nele, a probabilidade de haver r etiquetas em um *slot* é binomialmente distribuída com parâmetros n e $1/L$ conforme representa a seguinte equação:

$$B_{n,\frac{1}{L}}(r) = \binom{n}{r} \left(\frac{1}{L}\right)^r \left(1 - \frac{1}{L}\right)^{n-r} \, . \tag{5}$$

Adicionalmente, em um *frame* com L *slots*, a quantidade esperada de *slots* contendo transmissões de r etiquetas é dada por:

$$a_r^{L,n} = LB_{n,\frac{1}{L}}(r) = L\binom{n}{r} \left(\frac{1}{L}\right)^r \left(1 - \frac{1}{L}\right)^{n-r} \, . \tag{6}$$

O método de Vogt é baseado na desigualdade de Chebyshev, a qual afirma que o resultado de um experimento envolvendo uma variável aleatória X é, provavelmente, próximo ao valor esperado de X [Vogt 2002]. Utilizando esse conceito, Vogt propõe uma função de estimativa do número de etiquetas que busca a minimização da distância entre o

vetor $< s_v, s_s, s_c >$ e o vetor contendo os valores esperados para s_v, s_s e s_c. Essa função de estimativa é dada por:

$$\hat{n}(L, s_v, s_s, s_c) = \min_n \left| \begin{pmatrix} a_0^{L,n} \\ a_1^{L,n} \\ a_{\geq 2}^{L,n} \end{pmatrix} - \begin{pmatrix} s_v \\ s_s \\ s_c \end{pmatrix} \right| \ , \tag{7}$$

onde $\hat{n}$ é o valor de n que minimiza o módulo da diferença dos dois vetores representados nessa equação.

O estimador de Vogt também define uma função de cálculo do tamanho do próximo *frame* com base no número estimado $\hat{n}$ de etiquetas. Os resultados possíveis para a função proposta são apresentados na Tabela 1. Caso $\hat{n} \in [17, 27]$ tanto $\hat{f} = 32$ quanto $\hat{f} = 64$ são escolhas adequadas.

$\hat{f}$	$\hat{n} \in [x, y]$
16	$[1, 9]$
32	$[10, 27]$
64	$[17, 56]$
128	$[51, 129]$
256	$[112, \infty]$

Tabela 1. Tamanho de *frames* para o método de Vogt.

É importante ressaltar que de forma diferente de outras propostas apresentadas, o estimador de Vogt foi desenvolvido considerando-se limitações do sistema RFID *I-Code*. Nesse sistema, o tamanho dos *frames* estão limitados a uma potência de 2 com tamanho máximo possível igual a 256 *slots*. A maioria das pesquisas atuais envolvendo estimadores para o DFSA utilizam o método de Vogt sem tais limitações. Para isso, o tamanho do próximo *frame* no DFSA é calculado simplesmente como:

$$\hat{f} = \hat{n} - s_s \ . \tag{8}$$

2.4. Eom-Lee

O método Eom-Lee [Eom and Lee 2010] propõe o uso de um algoritmo iterativo para se estimar a quantidade de etiquetas competindo por *slots* em um *frame* e o tamanho $\hat{f}$ de seu próximo *frame*. Primeiramente, Eom-Lee define L como sendo o tamanho do *frame* que será analisado para se estimar o tamanho do *frame* imediatamente posterior. O valor de L é assumido ser igual ao número estimado de etiquetas que competiram no *frame* multiplicado por um fator β que deverá ser determinado. Assim sendo, o valor de L pode ser representado por:

$$L = \beta \cdot \hat{n} \ . \tag{9}$$

Também é assumido que o número de etiquetas competindo em um *slot* em colisão é igual a γ. Considerando o tamanho do próximo *frame* igual ao *backlog*, o valor de $\hat{f}$ pode ser calculado como:

$$\hat{f} = \hat{n} - s_s = \gamma \cdot s_c \, . \tag{10}$$

A fim de se obter a estimativa $\hat{n}$ do número de etiquetas, o problema em questão passa a ser a determinação do valor de γ ou de β. Em [Eom and Lee 2010], é demonstrado que γ e β podem ser relacionados através da Equação (11) ao se considerar que a probabilidade de r etiquetas, dentre o universo total de etiquetas, transmitirem em um mesmo *slot* pode ser aproximada por uma distribuição binomial e que a equação de cômputo de tal probabilidade pode ser aproximada por uma distribuição de Poisson com média n/L para L suficientemente grande.

$$\gamma = \frac{1 - e^{-\frac{1}{\beta}}}{\beta(1 - (1 + \frac{1}{\beta})e^{-\frac{1}{\beta}})} \, . \tag{11}$$

Encontrar uma solução fechada para se determinar os valores de γ e β a partir da Equação (11) é um desafio. Por causa disso, o método Eom-Lee utiliza um algoritmo iterativo para encontrar tais valores. Considere γ_k e β_k sendo, respectivamente, uma aproximação para o valor de γ e de β na k-ésima iteração do algoritmo. Essas aproximações são obtidas de acordo com as seguintes equações:

$$\beta_k = \frac{L}{\gamma_{k-1} \cdot s_c + s_s} \, , \tag{12}$$

$$\gamma_k = \frac{1 - e^{-\frac{1}{\beta_k}}}{\beta_k(1 - (1 + \frac{1}{\beta_k})e^{-\frac{1}{\beta_k}})} \, . \tag{13}$$

No primeiro passo, considera-se $\beta_1 = \infty$ e $\gamma_1 = 2$ e em cada passo k seguinte se determina uma nova aproximação para β e γ com o auxílio das Equações (12) e (13), respectivamente. Quando $|\gamma_{k^*-1} - \gamma_{k^*}|$ for menor que um limiar pré-definido $\epsilon_{threshold}$, o processo iterativo é interrompido. γ_{k^*-1} e γ_{k^*} representam, respectivamente, a aproximação anterior e atual para o valor de γ. A partir de então, o tamanho $\hat{f}$ do próximo *frame* e a quantidade estimada $\hat{n}$ de etiquetas são obtidos, respectivamente, pelas Equações (14) e (15), onde β_{k^*} é a aproximação mais recente para o valor de β.

$$\hat{f} = \gamma_{k^*} \cdot s_c \, . \tag{14}$$

$$\hat{n} = \frac{\hat{f}}{\beta_{k^*}} \, . \tag{15}$$

3. Avaliação e Análise dos Trabalhos Relacionados

Esta seção apresenta uma avaliação independente do desempenho de todos os estimadores apresentados na Seção 2 e analisa os resultados obtidos. Para isso, foi desenvolvido um simulador de eventos discretos na linguagem C++. Nas simulações apresentadas, considera-se um sistema RFID básico com um leitor e uma determinada quantidade de

etiquetas a serem identificadas utilizando-se o protocolo DFSA. Em todas as simulações apresentadas neste artigo, a taxa de transmissão é de 40 kbps e o identificador das etiquetas possui 128 *bits*.

Adicionalmente, as extensões *Early-End* e *Muting* também são utilizadas em todas as simulações neste artigo. A extensão *Muting* permite ao leitor silenciar, até o término do processo de identificação, uma etiqueta identificada com sucesso. Assim sendo, essa etiqueta deixa de competir por *slots*, reduzindo a quantidade de colisões e, por consequência, melhorando a eficiência do protocolo. A extensão *Early-End* permite que o leitor "encurte" a duração de um *slot* vazio. Como o leitor informa às etiquetas não só o início de um *frame* como também o início de cada *slot*, ele pode verificar rapidamente se *bits* de informações começaram a ser recebidos ou não após o envio de um comando de início de *slot*. Ao se passar um tempo pré-determinado após o início de um *slot*, se o leitor não detectar o início da transmissão de ao menos uma etiqueta, o *slot* é encerrado prematuramente. Com isso, o desperdício de tempo gerado por *slots* vazios ao longo do processo de identificação é reduzido.

As métricas de desempenho estudadas foram: o número total de *slots* utilizados no processo de identificação, o tempo total para a identificação de todas as etiquetas, o número total de *slots* vazios, o número total de *slots* em colisão e a média do erro absoluto de estimação por *frame* ao longo do processo de identificação. O erro absoluto de estimação é definido como o módulo da diferença entre o número real e o número estimado de etiquetas em um *frame* de interesse. Para cada estimador, os resultados apresentados foram obtidos a partir da média dos resultados de 2.000 simulações. A fim de se estudar apenas o impacto dos estimadores no desempenho do processo de identificação, considera-se um canal de comunicação livre de erros. Em particular ao estimador Eom-Lee, adota-se o parâmetro $\epsilon_{threshold}$ igual a 0,001. O tamanho do *frame* inicial é de 64 *slots*.

3.1. Modelagem do Canal de Comunicação

Neste trabalho, o canal de comunicação foi modelado a partir da dados fornecidos pela especificação EPCglobal Class-1 Gen-2 [EPC Global 2008]. Essa especificação define as temporizações para os diferentes tipos de *slots* no canal de comunicação e as regras de comunicação entre etiquetas e leitores. O leitor deve se referir à especificação usada para maiores informações sobre como calcular os tempos T_1 a T_5 explicitados nesta seção.

Considere a simbologia T =>R como sendo a representação de uma comunicação de etiqueta para leitor e a simbologia R =>T como sendo a representação de uma comunicação de leitor para etiqueta. Com base na especificação EPCglobal Class-1 Gen-2, modelou-se o canal de comunicação do seguinte modo: toda comunicação (*i.e.* comandos e respostas) é precedida de um período de sinalização, independente de seu sentido. Na comunicação R =>T, a sinalização pode ser de dois tipos: *preâmbulo* e *frame-sync*, onde o preâmbulo é utilizado apenas quando o leitor envia um comando de início de *frame* e o *frame-sync* é utilizado para qualquer outro tipo de comando. Na comunicação T =>R, a sinalização é feita através de um preâmbulo de tamanho igual a 6 *bits* conforme define a norma utilizada.

Imediatamente após o comando de início de *frame*, o leitor envia um comando de início de *slot*. A duração de um comando de início de um *slot* é igual a T_1. Este va-

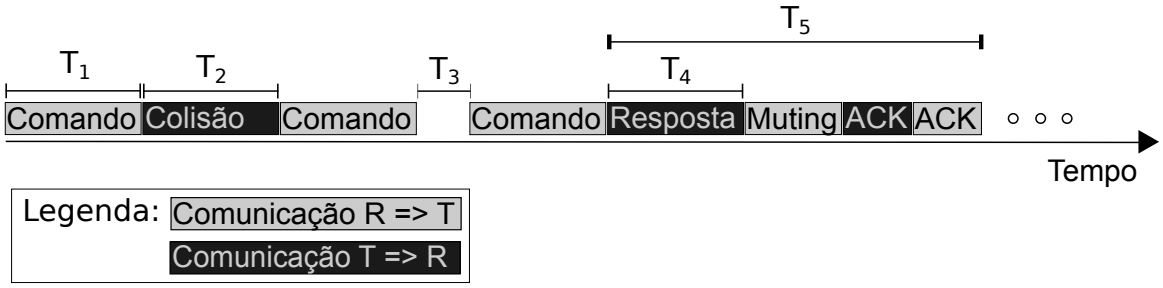


Figura 2. Modelo do canal de comunicação.

lor representa a soma da duração da sinalização com o tempo de transmissão e atraso de propagação. Após o envio desse comando, três situações podem acontecer: a ausência de respostas, respostas múltiplas ou uma única resposta. Um *slot* em colisão possui duração T_2. Essa duração é composta pelo tempo de propagação somado ao tempo de transmissão da mensagem das etiquetas e à duração do preâmbulo usado pelas mesmas. Caso o leitor transmita um comando de início de *slot*, se ele não começar a receber os *bits* iniciais de ao menos uma resposta, a duração do *slot* é encurtada. Um *slot* de tempo vazio, possui duração igual a $T_3 = max(RT_{Cal}, 10T_{pri})$, conforme parâmetros definidos na especificação [EPC Global 2008]. A duração de um *slot* bem sucedido é igual a T_4 e considera a soma dos atrasos de transmissão e propagação da mensagem da etiqueta. O tempo T_5 representa o tempo necessário para identificar uma etiqueta e silenciá-la.

A Figura 2 ilustra os três casos de *slots* possíveis em um determinado *frame* e os intervalos de tempo mais importantes. Nessa ilustração, o leitor envia um comando de início de *slot* para as etiquetas das quais as respostas colidem em um *slot*. Em seguida, o leitor envia um novo comando de início de *slot* para o qual não há respostas. Por fim, o leitor envia um comando de início de *slot* para o qual apenas uma etiqueta responde. Após a resposta da etiqueta, o leitor envia o comando *Muting*, a etiqueta responde com um *ACK* e o leitor confirma a recepção desse *ACK* com outro *ACK*. Nesse ponto, a etiqueta é considerada identificada.

3.2. Resultados e Análises

A Figura 3(a) mostra o erro absoluto médio de estimação para cada um dos estimadores quando o número total de etiquetas varia de 100 a 1.000. O erro absoluto médio do *Lower Bound* e do Schoute tem crescimento exponencial lento em função do aumento do número de etiquetas a serem identificadas. Dentre todos os estimadores avaliados, o Eom-Lee é o melhor de todos até pouco mais de 900 etiquetas. Em torno desse valor, o Vogt passa a ser melhor. É importante observar que no intervalo de 100 a 300 etiquetas, o erro absoluto médio do estimador Vogt é muito próximo ao observado para o estimador Eom-Lee. Já entre 300 e 500 etiquetas, o erro absoluto médio do estimador Vogt cresce rapidamente. Contudo, observa-se que a partir de 700 etiquetas até 900 etiquetas, o erro absoluto médio do Vogt volta a se aproximar cada vez mais do erro observado com o uso do estimador Eom-Lee.

A Figura 3(b) mostra o número total médio de *slots* usados em função da quantidade de etiquetas a serem identificadas. Até pouco menos de 900 etiquetas, o estimador Eom-Lee produziu os melhores resultados, embora seja verificado que os demais estimadores tenham produzido resultados próximos para determinadas quantidades de etiquetas.

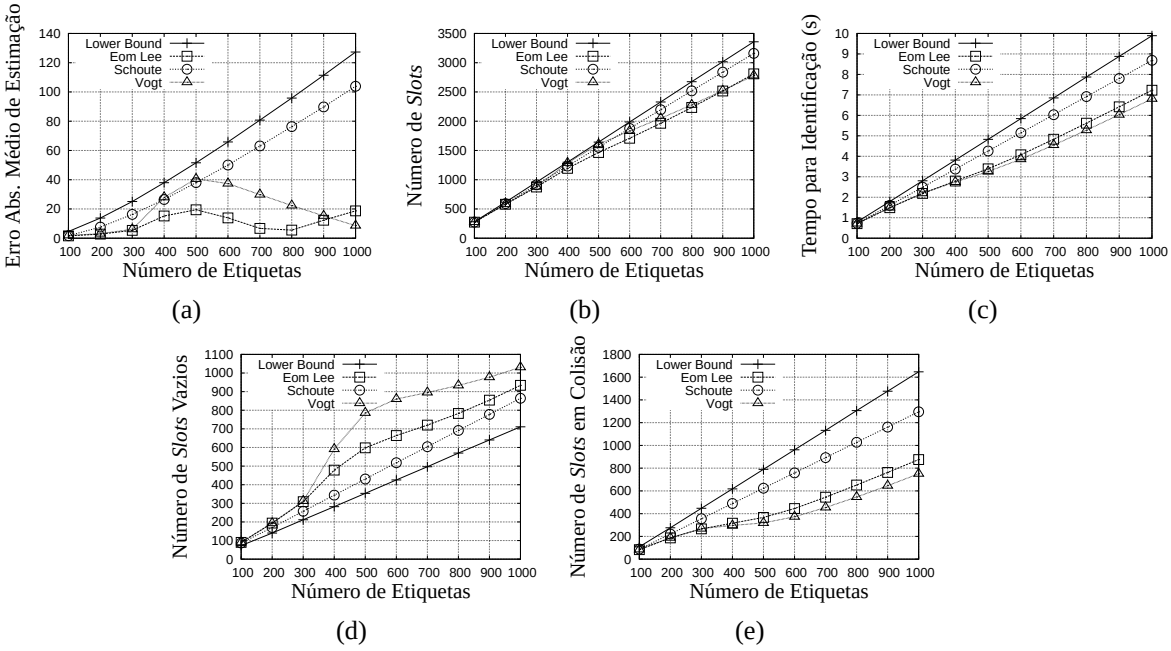


Figura 3. Desempenho dos Estimadores.

Note que a diferença do erro absoluto médio de estimação entre o pior e o melhor estimador não é tão grande, principalmente, até 500 etiquetas. Essa diferença não trouxe impacto significativo no desempenho dos estimadores em relação à quantidade total de *slots* usados ao longo do processo de identificação até 500 etiquetas. A partir de 500 etiquetas, a diferença no erro absoluto médio de estimação entre o pior e o melhor estimador aumenta, impactando, cada vez mais, na quantidade média total de *slots* usados no processo de identificação. Em particular, observa-se que os resultados com o estimador Vogt começam a convergir para os resultados obtidos com o estimador Eom-Lee a partir de 500 etiquetas. Já entre 800 e 1.000 etiquetas, esses dois estimadores possuem desempenho similar em termos do número total médio de *slots* usados.

A Figura 3(c) mostra a média do tempo total de identificação em função do número total de etiquetas. Em particular, observa-se que sob o ponto de vista dessa métrica, os estimadores Vogt e Eom-Lee são os melhores e permitem um tempo de identificação comparável até 400 etiquetas. A partir desse valor, o Vogt leva uma ligeira vantagem sobre o Eom-Lee. Isso é explicado pelo impacto da extensão *Early-End* associado à quantidade de *slots* vazios e em colisão gerada no processo de identificação. Note que o Vogt permite a ocorrência de mais *slots* vazios (Figura 3(d)) e, portanto, a ocorrência de menos colisões (Figura 3(e)) em relação ao Eom-Lee a partir de 300 etiquetas. Note também que o custo "temporal" de um *slot* em colisão é significativamente maior em relação ao custo "temporal" de um *slot* vazio no DFSA com a extensão *Early-End*. Assim, a partir de 400 etiquetas, o Vogt consegue produzir uma melhor relação custo/benefício entre a quantidade de *slots* vazios e em colisão, permitindo um menor tempo total de identificação.

Pelo exposto, algumas conclusões importantes podem ser tiradas quando o DFSA é utilizado em conjunto com a extensão *Early-End*, mas não se considera o impacto da mesma no cálculo do tamanho dos *frames*: 1) uma melhoria na quantidade total de *slots*

utilizados pode não implicar necessariamente em uma redução no tempo total do processo de identificação; 2) buscar apenas um menor erro de estimação não é uma condição suficiente para se minimizar o tempo total de identificação; 3) uma redução adequada na quantidade de *slots* em colisão com um aumento adequado na quantidade de *slots* vazios em cada *frame* gerado contribui para a minimização do tempo total de identificação; 4) superestimar de forma adequada o *backlog* contribui para a minimização do tempo total de identificação; e 5) a métrica de avaliação de desempenho mais importante a ser considerada é o tempo total de identificação de etiquetas.

4. A Função de Cálculo de Tamanho de *Frames* Proposta

De acordo com as conclusões da seção anterior, percebe-se que nenhuma das funções de cálculo de tamanho de *frames*, até então, explora explicitamente o impacto da extensão *Early-End* a fim de se buscar uma minimização do tempo total de identificação de etiquetas. Assim sendo, esta seção apresenta uma proposta de função de cálculo de tamanho de *frames* que explora o impacto dessa extensão. A estratégia proposta consiste em superestimar adequadamente o *backlog* para o cálculo do tamanho de cada novo *frame* gerado, permitindo uma quantidade "razoável" de *slots* vazios e, uma consequente redução de *slots* em colisão. Como o tamanho do próximo *frame* no DFSA é proporcional ao *backlog* estimado, é proposto que o tamanho $\hat{f}$ do próximo *frame* calculado pelo estimador seja ajustado por um fator multiplicativo δ_i. Onde i representa o tamanho do *frame* inicial. Assim sendo, é proposta a seguinte função que relaciona a função de cálculo do estimador com esse fator de ajuste:

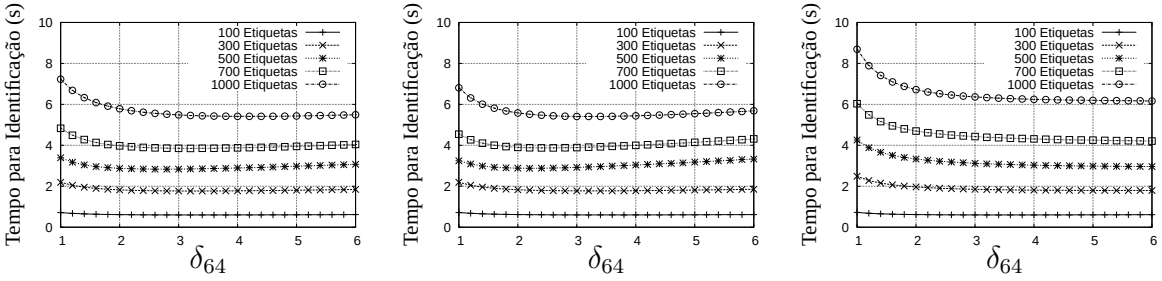
$$F(\delta_i, \hat{f}) = \delta_i \hat{f} \ . \tag{16}$$

O valor de δ_i a ser utilizado na Equação (16) precisa ser determinado para cada estimador, buscando-se a minimização do tempo total do processo de identificação de etiquetas. Um estudo sobre o impacto do valor de δ_i no tempo total de identificação de etiquetas é apresentado a seguir na Seção 4.1.

4.1. Determinação do Parâmetro δ_i

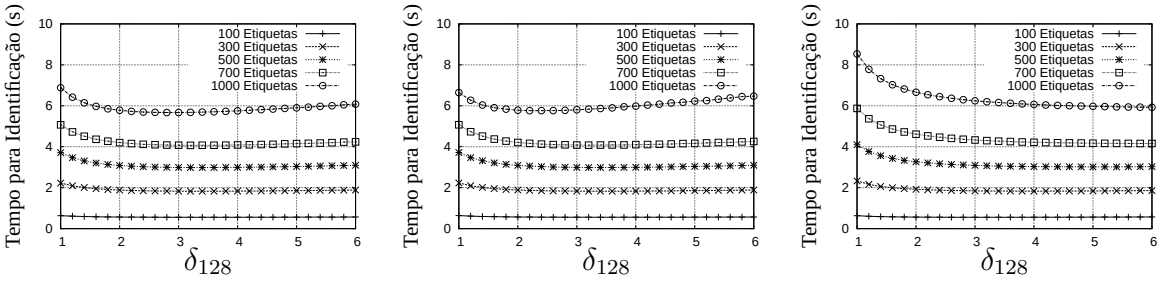
Alguns valores adequados de δ_i para uso na função de cálculo proposta foram determinados através de simulações. Nessas simulações, foi avaliado o tempo total de identificação de etiquetas em função de δ_i considerando-se os métodos Eom-Lee, Vogt e Schoute para um *frame* inicial de 64 *slots* e para um *frame* inicial de 128 *slots*. O valor de δ_i foi variado de 1 a 6 em passos de 0,2. Todos os métodos foram avaliados considerando-se um total de 100, 300, 500, 700 e 1.000 etiquetas a serem identificadas. Os resultados apresentados são médias obtidas a partir de 2.000 simulações. Os parâmetros do canal de comunicação são idênticos aos descritos na Seção 3.

As Figuras 4 e 5 mostram a influência do fator de ajuste no tempo total de identificação de etiquetas para um *frame* inicial de 64 *slots* e de 128 *slots*, respectivamente. Em particular, observa-se que a minimização do tempo total de identificação depende do valor do fator de ajuste, o qual depende também da quantidade total de etiquetas a serem identificadas. O problema em questão é encontrar um valor adequado para o fator de ajuste que permita a obtenção de ganhos no tempo total de identificação para qualquer quantidade de etiquetas no intervalo $[100, 1.000]$.



(a) Eom-Lee (b) Vogt (c) Schoute

Figura 4. Influência do fator δ no desempenho do DFSA para cada estimador.



(a) Eom-Lee (b) Vogt (c) Schoute

Figura 5. Influência do fator δ no desempenho do DFSA para cada estimador.

No caso do Eom-Lee, o valor de δ_{64} (Figura 4(a)) que minimiza o tempo total de identificação para 300 e 500 etiquetas é igual a 3,0. Esse valor também permite uma redução no tempo total de identificação para o caso de haver 100, 700 ou 1.000 etiquetas a serem identificadas, apesar do tempo total não ser minimizado. Contudo, observa-se que para $\delta_{64} > 3{,}0$ ou não há ganhos significativos de desempenho ou o desempenho passa a ser pior do que quando $\delta_{64} = 3{,}0$. No caso do *frame* inicial ser de 128 *slots*, o valor do fator de ajuste (Figura 5(a)) que minimiza o tempo total de identificação para 500 e 700 etiquetas é igual a 3,2. Esse valor também permite uma redução no tempo total de identificação para 100, 300 e 1.000 etiquetas, embora o tempo total de identificação não seja minimizado. Apesar de $\delta_{128} = 3{,}2$ não ser um valor ótimo para 100, 300 e 1.000 etiquetas, o módulo da diferença do tempo total de identificação com tal valor e o valor do δ_{128} ótimo para para tais quantidades de etiquetas é, respectivamente, igual a 0,47 ms, 1,1 ms e 6,72 ms. Logo, a diferença não é significativa. De acordo com as análises feitas, $\delta_{64} = 3{,}0$ e $\delta_{128} = 3{,}2$ são valores adequados para o fator de ajuste.

Com relação ao Vogt, o valor de δ_{64} (Figura 4(b)) que minimiza o tempo total de identificação para 100 e 300 etiquetas é igual a 3,2. Com esse valor, o tempo total de identificação também é reduzido quando há 500, 700 ou 1.000 etiquetas a serem identificadas, embora o tempo total de identificação não seja minimizado. O módulo da diferença do tempo total de identificação entre o caso de $\delta_{64} = 3{,}2$ e os fatores ótimos para as quantidades de etiquetas de 100, 300 e 1.000 é, respectivamente, igual a 63,7 ms, 354,66 ms e 0,3 ms. Mais uma vez, as diferenças não são significativas. No caso do *frame* inicial ser de 128 *slots*, o valor do fator de ajuste (Figura 5(b)) que minimiza o tempo total de identificação para 300 e 700 etiquetas também é igual a 3,2. Esse valor também permite uma redução no tempo total de identificação para o caso de haver 100,

500 ou 1.000 etiquetas a serem identificadas, mesmo não minimizando o tempo total. O módulo da diferença do tempo total de identificação obtido ao se utilizar $\delta_{128} = 3{,}2$ e os fatores ótimos para quantidades de etiquetas de 100, 500 e 1.000 é, respectivamente, igual a 0,472 ms, 1,062 ms e 75,018 ms. Com base nas análises apresentadas, $\delta_{64} = 3{,}2$ e $\delta_{128} = 3{,}2$ são valores adequados para o fator de ajuste.

No caso do Schoute, $\delta_{64} = 6{,}0$ (Figura 4(c)) produz o menor tempo total de identificação para 300, 500, 700 e 1.000 etiquetas. Para uma quantidade de etiquetas igual a 100, o valor de δ_{64} que minimiza o tempo total de identificação é igual a 3,6. Contudo, para essa quantidade de etiquetas, um $\delta_{64} = 6{,}0$ ainda produz ganhos de desempenho em relação ao caso de se utilizar um valor de $\delta_{64} = 1{,}0$. No caso do *frame* inicial ser de 128 *slots*, $\delta_{128} = 5{,}8$ minimiza o tempo total de identificação para 700 etiquetas. O módulo da diferença do tempo total de identificação entre o caso de $\delta_{128} = 5{,}8$ e os fatores ótimos para 100, 300, 500 e 1.000 etiquetas é, respectivamente, igual a 14,591 ms, 23,405 ms, 5,996 ms e 8,313 ms. Mais uma vez, a diferença de tempo observada não é significativa. Com base no exposto, $\delta_{64} = 6{,}0$ e $\delta_{128} = 5{,}8$ são valores adequados para o fator de ajuste.

5. Avaliação de Desempenho com a Função Proposta

Esta seção apresenta uma avaliação de desempenho do DFSA com e sem o uso da função de cálculo proposta. Mais uma vez são empregadas as extensões *Early-End* e *Mutting*. As métricas de desempenho avaliadas em função do número total de etiquetas foram: o número total de *slots* vazios, o número total de *slots* em colisão e o tempo total para a identificação das etiquetas. Os parâmetros do canal de comunicação são idênticos aos descritos na Seção 3. Os resultados apresentados são médias obtidas a partir de 2.000 simulações.

As Figuras 6, 7 e 8 mostram os resultados obtidos para um *frame* inicial de 64 *slots* utilizando-se os estimadores Eom-Lee, Vogt e Schoute, respectivamente. Note que para todos os casos avaliados, a função proposta permite uma melhoria no tempo total de identificação de etiquetas já que ela explora melhor o impacto da extensão *Early-End*. Tal melhoria em relação ao Eom-Lee, ao Vogt e ao Schoute é, respectivamente, de até 24%, 20% e 29%. A correta exploração da extensão se traduz em um aumento adequado da quantidade de *slots* vazios e, em consequência, em uma redução adequada da quantidade de *slots* em colisão por *frame* estimado. Note também que embora a quantidade total de *slots* utilizados no processo de identificação aumente, não há impacto negativo no tempo total de identificação das etiquetas. Os resultados obtidos para um *frame* inicial de 128

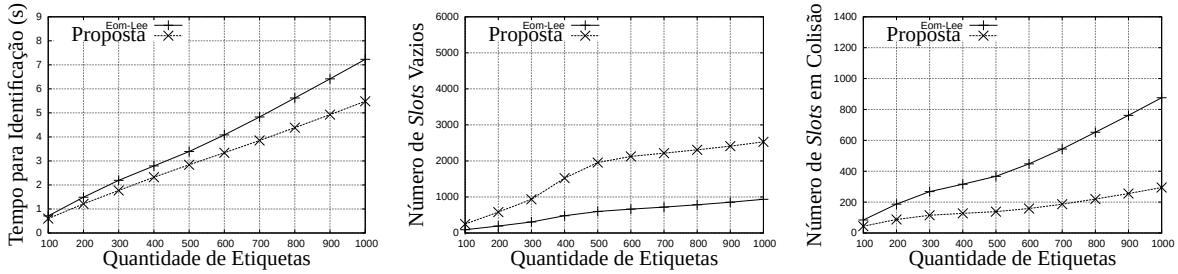


Figura 6. Estimador Eom-Lee e *frame* inicial de 64 *slots*.

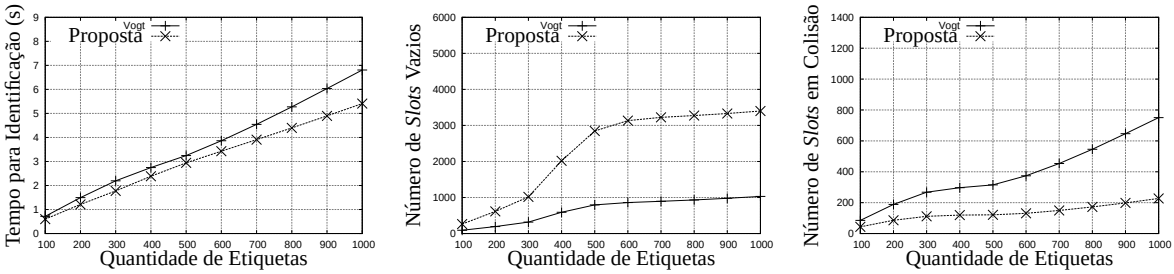


Figura 7. Estimador Vogt e *frame* inicial de 64 *slots*.

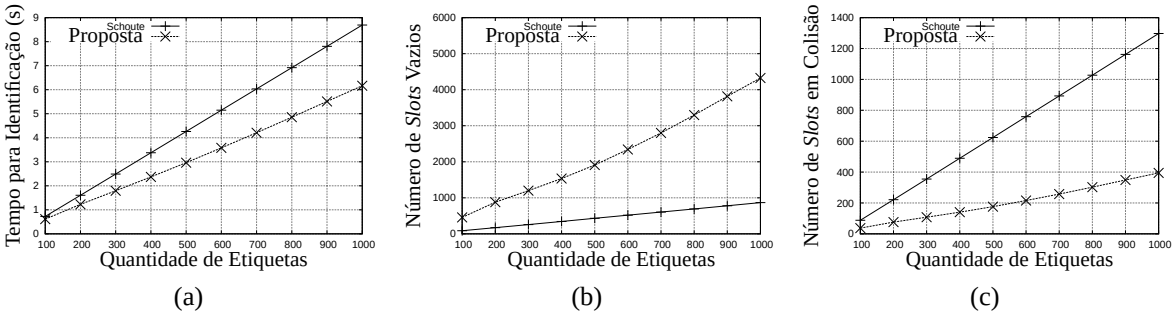


Figura 8. Estimador Schoute e *frame* inicial de 64 *slots*.

slots foram omitidos por limitações de espaço, porém a abordagem proposta continua trazendo melhorias significativas no tempo total de identificação de etiquetas.

Os fatores de ajuste encontrados na Seção 4.1 para cada estimador não diferiram significativamente ao se mudar o tamanho do *frame* inicial de 64 para 128 *slots*. Além disso, os valores de δ_{64} e δ_{128} encontrados são subótimos para várias quantidades de etiquetas. Mesmo assim, o tempo total de identificação ao se utilizar um fator ótimo não se mostrou significativamente diferente do tempo obtido com os fatores subótimos indicados. Tudo isso sugere que, para ambos os tamanhos de *frame* inicial estudados, um mesmo fator de ajuste pode ser utilizado sem prejuízo perceptível no tempo total de identificação. A avaliação dessa hipótese é apresentada na Figura 9. Ela mostra o tempo total de identificação para o Eom-Lee e para o Schoute com a função proposta quando se considera um *frame* inicial igual a 128 *slots* e os fatores de ajuste δ_{64} e δ_{128}. Note que não há diferenças perceptíveis no tempo total de identificação.

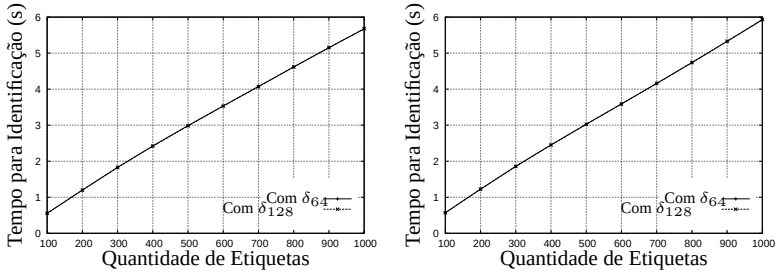


Figura 9. Impacto dos fatores de ajuste para um *frame* inicial de 128 *slots*.

6. Conclusões

Esse artigo apresentou um estudo do protocolo DFSA com os principais estimadores de tamanho de *frame* propostos na literatura. Dentre as principais contribuições encontram-se conclusões importantes tiradas ao se analisar o DFSA com a extensão *Early-End*, mas sem considerar o impacto da mesma no cálculo do tamanho dos *frames*: 1) uma melhoria na quantidade total de *slots* utilizados pode não implicar necessariamente em uma redução no tempo total do processo de identificação; 2) um menor erro de estimação não é uma condição suficiente para se minimizar o tempo total de identificação; 3) uma redução adequada na quantidade de *slots* em colisão com um aumento adequado na quantidade de *slots* vazios em cada *frame* gerado contribui para a minimização do tempo total de identificação; e 4) superestimar de forma adequada o *backlog* contribui para a minimização do tempo total de identificação.

A partir dessas conclusões, foi apresentada mais uma contribuição, sendo ela a proposta de uma função de cálculo de tamanho de *frames* para o DFSA que explora o impacto da extensão *Early-End*. A função proposta relacionou a função de cálculo de um estimador escolhido com um fator de ajuste. Valores adequados para esse fator foram determinados de acordo com os estimadores e com os tamanhos de *frame* inicial estudados. Os resultados mostraram que a abordagem proposta permitiu que o tempo total de identificação de etiquetas no DFSA com *Early-End* fosse melhorado quando comparado com o tempo total obtido ao se utilizar a função original do estimador isoladamente. Em particular, a melhora observada no tempo total de identificação de etiquetas no caso de uso da função proposta em conjunto com o Eom-Lee, o Vogt e o Schoute foi de até 24%, 20% e 29%, respectivamente.

Referências

Chen, W.-T. (2009). An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length ALOHA. *IEEE Transactions on Automation Science and Engineering*, 6(1):9 –15.

Eom, J.-B. and Lee, T.-J. (2010). Accurate Tag Estimation for Dynamic Framed-slotted ALOHA in RFID Systems. *IEEE Communications Letters*, 14:60–62.

EPC Global, I. (2008). *EPC Radio-Frequency Identity Protocols Class-1 Generation 2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz*. Version 1.2.0.

Klair, D., Chin, K.-W., and Raad, R. (2010). A Survey and Tutorial of RFID Anti-Collision Protocols. *IEEE Communications Surveys Tutorials*, 12(3):400 –421.

Schoute, F. C. (1983). Dynamic Frame Length ALOHA. *IEEE Transactions on Communications*, 31:565–568.

Tong, Q., Zou, X., and Tong, H. (2009). Dynamic Framed Slotted ALOHA Algorithm Based on Bayesian Estimation in RFID System. In *Proc. of the WRI World Congress on Computer Science and Information Engineering*, pages 384–388.

Vogt, H. (2002). Efficient Object Identification with Passive RFID Tags. In *Proceedings of the First International Conference on Pervasive Computing*, pages 98–113, London, UK. Springer-Verlag.

Aprovisionamento de QoS e QoE em Redes Sem Fio Heterogêneas com Suporte a Balanceamento de Carga

Warley M. V. Junior[1], José Jailton[1], Tassio Carvalho[1], Kelvin Lopes Dias[2]

[1] Pós-Graduação em Engenharia Elétrica – Universidade Federal do Pará (UFPA)
Caixa Postal 479 – 66.075-110 – Belém – PA – Brasil

[2] Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50.670-901 – Recife – PE – Brasil

{warleyjunior, jjj, tassio}@ufpa.br, kld@cin.ufpe.br

Abstract. *This paper proposes a framework for QoS and QoE provisioning in heterogeneous wireless networks formed by WiMAX/Wi-Fi networks, in order to promote the QoS mapping and ensure the quality of video received by user. The proposal adopts the new IEEE 802.21 standard to allow the vertical handover, the target network detection, and to facilitate a seamless integration between technologies. Furthermore, we propose an algorithm for vertical handover decision that takes into account the classes of services of WiMAX, the access categories of Wi-Fi, and the aggregate throughput from both the current and the target networks. The simulations demonstrate the effectiveness of the framework in ensuring QoS/QoE and load balancing.*

Resumo. *Este artigo propõe um arcabouço para provisionamento de QoS e QoE em redes sem fio heterogêneas WiMAX/Wi-Fi, de modo a promover o mapeamento de QoS e garantir a qualidade do vídeo recebido pelo usuário. A proposta adota o novo padrão IEEE 802.21 para permitir o handover vertical, detecção da rede alvo e facilitar a integração e mobilidade transparente entre as tecnologias. Além disso, desenvolvemos um algoritmo de handover vertical que leva em conta as classes de serviços do WiMAX, categorias de acesso do Wi-Fi e vazão agregada tanto da rede atual, quanto da rede alvo. As simulações realizadas demonstram a eficácia do arcabouço na garantia de QoS/QoE e balanceamento de carga.*

1. Introdução

Recentemente a sociedade está vivendo em meio a um aglomerado de dispositivos portáteis com múltiplas interfaces sem fio que permitem que clientes domésticos e corporativos solucionem problemas rotineiros no menor espaço de tempo e independente de sua localização. Além disso, a demanda por *smartphones*, *tablets*, laptops e notebooks está com um índice de crescimento nas vendas acima do esperado pelas indústrias. Consequentemente, a exigência por serviços de qualidade prestados por operadores de rede também está aumentando. Por outro lado, a disponibilidade de diversas redes sem fio como: Wi-Fi (*Wireless Fidelity*), WiMAX (*Worldwide Interoperability for Microwave Access*), UMTS (*Universal Mobile Telecommunication System*) e LTE (*Long Term Evolution*), proporcionam um cenário heterogêneo com diversas oportunidades de conectividade para o usuário de dispositivos com múltiplas interfaces.

Particularmente, duas tecnologias de acesso sem fio, amplamente difundidas, podem prover suporte à qualidade de serviço (QoS - *Quality of Service*) para o acesso à internet móvel. O IEEE 802.11e [IEEE 802.11e 2005], um padrão para garantia de QoS em redes Wi-Fi, permite cobertura sem fio local. Por outro lado, o IEEE 802.16e [IEEE 802.16e 2005], suporta QoS no âmbito de redes metropolitanas WiMAX. Mesmo com suporte à QoS presente em cada uma dessas tecnologias, não há garantias que em um ambiente integrado e heterogêneo seja possível manter a qualidade e continuidade dos serviços à medida que o usuário muda, por exemplo, de uma estação base (BS - *Base Station)* WiMAX para um ponto de acesso (AP – *Access Point*) Wi-Fi.

A emenda IEEE 802.11e, inclui o HCF (*Hybrid Coordination Function*) que introduz dois modos de operação, o EDCA (*Enhanced Distributed Coordinated Access*), um mecanismo baseado em contenção e o HCCA (*HCF Controlled Channel Access*), um mecanismo livre de contenção. O EDCA [IEEE draft p802.11e d13.0 2005] define basicamente quatro categorias de acesso AC (*Access Categories*) na camada MAC, conhecidos como (AC_VO) para tráfego de voz, (AC_VI) para tráfego de vídeo, (AC_BE) para tráfego de melhor esforço, tal como HTTP e (AC_BK) para tráfego de fundo, tal como FTP, classificados da maior para menor prioridade, respectivamente. Cada categoria de acesso possui uma única fila de transmissão e parâmetros particulares, tais como limiares superiores e inferiores da janela de contenção, (CW_{Max} e CW_{Min}, respectivamente), espaçamento inter-quadros arbitrário AIFS (*Arbitrary Inter-Frame Spacing*) e oportunidade de transmissão TxOP (*Transmission Opportunity*).

O WiMAX, implementa suporte à QoS na camada MAC para facilitar a interação com o gerenciamento de recursos de rádio e camada física. Seu *framework* de QoS adota cinco classes de serviços ou CoS (*Class of Service*): UGS (*Unsolicited Grant Service*), rtPS (*real-time Polling Service*), ertPS (*extended real-time Polling Service*), nrtPS (*non real-time Polling Service*) e BE (*Best Effort*). Cada CoS possui um conjunto de parâmetros de QoS que devem ser inclusos na definição do fluxo de serviço quando a classe de serviço está habilitada para um fluxo de serviço. Os principais parâmetros são: prioridade de tráfego, latência máxima, jitter, máxima e mínima taxa de dados e máximo atraso [Sekercioglu *et al* 2009].

Alinhado com a necessidade de permitir conectividade transparente para usuários em movimento, servidos por diversas tecnologias sem fio, o IEEE desenvolveu e aprovou o novo padrão IEEE 802.21 ou MIHS (*Media Independent Handover Services*) [IEEE draft p802.21 d11.0 2008]. O MIHS ou simplesmente MIH, foi projetado para aperfeiçoar a integração e mobilidade entre redes sem fio de diferentes tecnologias, bem como para permitir o *handover* horizontal e vertical, isto é, a troca de PoA (*Point of Attachment*) entre tecnologias similares e distintas, respectivamente. Para realizar estes objetivos, o MIH conta com um conjunto de eventos de sinalização, gatilhos e serviços, unificados para qualquer tecnologia, que disponibilizam informação de camadas inferiores (MAC - *Media Access Control* e Física) para as camadas superiores (Camada de Aplicação) da pilha de protocolos.

Este artigo propõe um arcabouço para o aprovisionamento de QoS e QoE (*Quality of Experience*) em redes sem fio heterogêneas formada por redes WiMAX e Wi-Fi. Especificamente, nossa solução provê mapeamento de QoS entre as classes de serviços WiMAX e categorias de acesso Wi-Fi. Além disso, nossa proposta também combina funcionalidades de balanceamento de carga com a solução de mapeamento, a

fim de alcançar um bom compromisso tanto para o operador da rede, quanto para o usuário, através de um novo algoritmo de decisão de *handover* vertical VHD (*Vertical Handover Decision*). Além de utilizarmos o MIH, como tradicionalmente usado na literatura, para facilitar o *handover*, também definimos neste artigo, uma nova metodologia baseada no MIH para a obtenção de vazão agregada tanto da rede atual, quanto da rede alvo, bem como para auxiliar o algoritmo VHD no processo de decisão de *handover*. A proposta é avaliada via simulação, através de métricas de QoS (vazão) e QoE (PSNR, SSIM e VQM).

O artigo está organizado da seguinte forma: Na seção 2, os trabalhos relacionados são analisados. A seção 3 discute o esquema de mapeamento de QoS, algoritmo VHD e a metodologia de obtenção de vazão agregada através do MIH. A avaliação e resultados finais são apresentados na seção 4. A seção 5 apresenta as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Esta seção discute trabalhos relacionados à QoS, QoE, redes sem fio heterogêneas, MIH e políticas de decisão de *handover* presentes na literatura e esclarece a necessidade de um novo arcabouço para provisionamento de QoS e QoE com suporte a balanceamento de carga.

Em [Andi *et al* 2010], os autores propõem um *framework* de gerenciamento de interface com suporte a QoS em terminais *multi-interface* sem fio. O *framework* se baseia no MIH e utiliza métricas da camada de enlace para avaliar as condições da rede e assim auxiliar na decisão de *handover*. Mesmo com informações da camada de enlace de ambas as redes, os autores não propõem um esquema de balanceamento de carga. Este artigo não avalia o gerenciamento de QoS considerando cenários móveis, ou seja, os usuários estão estáticos em uma área de sobreposição entre as redes Wi-Fi e WiMAX. Além disso, a prioridade dos fluxos são estabelecidas por meio de reserva de canal ao invés de um framework de QoS heterogêneo.

A proposta apresentada em [Chen *et al* 2008], desenvolveu uma arquitetura denominada VHTC (*Vertical Handoff Translation Center Architecture*) para garantir QoS durante a fase de *handover* em redes heterogêneas. No entanto, os autores não simularam ou avaliaram qualquer cenário com mobilidade, tampouco o suporte à QoS durante o procedimento de *handover*. O termo *handover* mencionado no artigo é referente ao tráfego de dados e não a mobilidade de nodos. Os nodos estão estáticos e comunicam entre si via enlace cabeado, cujas extremidades da rede cabeada é formada pelas redes WiMAX e Wi-Fi. Este estudo usou uma versão modificada de módulos no ns-2, porém o módulo WiMAX utilizado, não implementa classes de serviços para a tecnologia WiMAX, ou seja, neste artigo não há, verdadeiramente, uma implementação de QoS para o WiMAX.

Em [Tarng *et al* 2010], os autores propõem um ambiente heterogêneo integrado das redes IEEE 802.11 e IEEE 802.16, bem como o desenvolvimento de um mecanismo de mapeamento de QoS afim de atender os requisitos de aplicações de tempo real por meio da alocação de largura de banda para a estação assinante. Os autores desenvolveram também dois algoritmos de QoS, um para a BS e outro para a SS (*Subscriber Station*). Vale ressaltar que o módulo WiMAX utilizado neste artigo, não implementa escalonamento de classes de serviços e no cenário avaliado, os nodos não

realizam mobilidade e não adotam nenhum algoritmo que possa auxiliar inteligentemente a decisão de *handover*.

Os autores em [Cerqueira *et al* 2008], propõem um controlador de sessões multi-usuários em redes heterogêneas sem fio e cabeada denominado *QUALITIS*, através da qual coordena o mapeamento de QoS e mecanismos de adaptação de QoS, juntamente com mecanismo de alocação de recurso e mobilidade. Apesar de a proposta apresentar diversos mecanismos e funcionalidades capazes de trabalharem em conjunto para prover o melhor nível de qualidade de uma sessão em andamento, o mesmo somente avalia dispositivos com interfaces Wi-Fi e, consequentemente, o mapeamento de QoS é realizado entre redes Wi-Fi, além de não assegurar *handover* transparente.

Alguns artigos na literatura implementaram classes de serviços no WiMAX e categorias de acesso no Wi-Fi, mas as análises realizadas foram independentes. Estes trabalhos somente incluem o MIH com parâmetros da camada de enlace para auxiliar o processo de decisório da rede ou desenvolvem algum mecanismo para realizar mapeamento de tráfego. Assim, até onde sabemos, as propostas não contemplam estudos que integrem a mobilidade em ambiente heterogêneo, o mapeamento entre classes de serviços e categorias de acesso com auxílio do MIH, bem como não propõem decisões de *handover* inteligentes baseadas em estratégias de balanceamento de carga entre redes Wi-Fi e WiMAX.

3. Arcabouço, Algoritmo VHD e Obtenção de Vazão Agregada via MIH

Esta seção descreve a funcionalidade integrada das três principais contribuições deste artigo: arcabouço para mapeamento de QoS, algoritmo VHD e mecanismo baseado no MIH para a medição e coleta das vazões agregadas das redes atual e alvo pela qual o nodo móvel MN (*Mobile Node*) pretende realizar *handover*.

3.1. Arcabouço para Mapeamento de QoS

A Figura 1 apresenta o diagrama lógico da integração do MIH com os padrões 802.11e/802.16e. Ela mostra a arquitetura interna do MN, rede 802, rede 3GPP e do núcleo da rede. Como podemos notar, todos os nodos e PoAs com suporte ao MIH, tem uma estrutura em comum em torno de uma entidade central, denominada MIHF (*Media Independent Handover Function*). O MIHF atua como uma camada intermediária entre as camadas inferiores e superiores, de modo que sua principal função é coordenar e trocar informações e comandos entre diferentes dispositivos que desejam tomar decisões e realizar *handover*s. Cada MN e PoA pode ter um conjunto de usuários MIH, protocolos de gerenciamento de mobilidade, que utilizam a funcionalidade do MIHF para controlar e obter informações relacionadas ao *handover*. O MIHF se comunica com os usuários MIH e camadas inferiores, com base em um número de primitivas de serviço definidas que são agrupadas em SAPs (*Service Access Points*). Conforme a figura, os três SAPs definidos estão listados a seguir: MIH_SAP, MIH_NET_SAP e MIH_LINK_SAP. O MIH_SAP é a interface que permite a comunicação entre o MIHF e as camadas superiores. O MIH_NET_SAP é a interface responsável pela troca de informações entre as entidades MIHF remota. O MIH_LINK_SAP é a interface de comunicação entre o MIHF e as camadas inferiores [Oliva *et al* 2011].

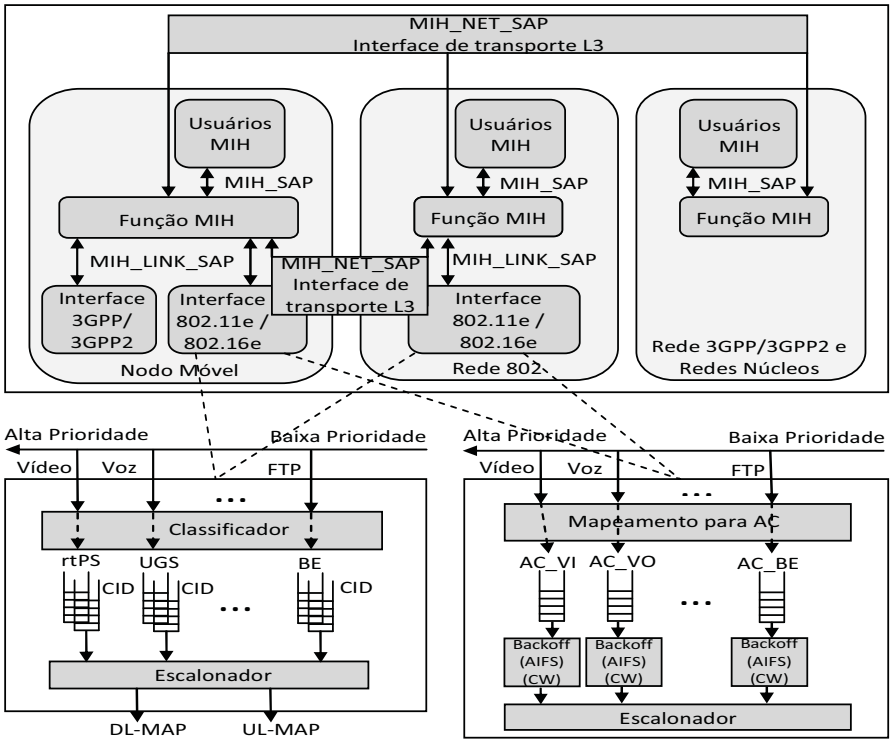


Figura 1. Arquitetura geral para mapeamento de QoS.

É através do MIH_LINK_SAP que os parâmetros de QoS da camada MAC são passados para as camadas superiores, em ambas as tecnologias Wi-Fi e WiMAX. Assim, com as adaptações e integração das funcionalidades que permitam a classificação e escalonamento dos fluxos provenientes das camadas superiores e vice-versa, é possível garantir QoS conforme os padrões IEEE 802.11e quando o usuário está conectado com a rede Wi-Fi e IEEE 802.16e quando o usuário está conectado com a rede WiMAX.

A estratégia de mapeamento está ilustrada na Tabela 1. Aplicações de Voz, *streaming* de vídeo e FTP são classificados e escalonados conforme o padrão IEEE 802.16e, utilizando CoS WiMAX, tais como UGS, rtPS e BE, ou classificados conforme o padrão IEEE 802.11e (EDCA) em AC Wi-Fi, tais como AC_VO, AC_VI e AC_BE.

Tabela 1. Mapeamento de QoS entre as redes WiMAX e Wi-Fi.

Aplicação	Exemplo	IEEE 802.16e	IEEE 802.11e
Voz	VoIP sem supressão de silêncio (Tráfego CBR)	UGS	AC_VO
Vídeo	MPEG, VoIP com supressão de silêncio(Tráfego VBR)	rtPS	AC_VI
Melhor Esforço	FTP (*File Transfer Protocol*)	BE	AC_BE

3.2. Suporte ao Balanceamento de Carga

Apesar de o MIH auxiliar no processo de *handover* transparente entre diferentes tecnologias, adoção do protocolo MIPV6, além de outras funcionalidades primárias e secundárias, o mesmo não possui um sistema ou algoritmo inteligente de decisão de *handover*, capaz de considerar as condições do meio em que o MN está inserido.

Assim, propomos um algoritmo VHD capaz de auxiliar o MIH no processo decisório. O VHD levará em conta CoS e AC combinado com a vazão agregada atual da rede que está servindo e da rede alvo, afim de garantir continuidade de serviço e uma distribuição satisfatória do tráfego dentro de redes heterogêneas.

A Figura 2 ilustra o algoritmo VHD. Os valores dos limiares são baseados em resultados empíricos obtidos através de medições depois da execução de várias simulações para cenários similares aos avaliados em nosso estudo. É importante ressaltar que estes valores podem mudar dependendo do cenário de estudo (diferentes tecnologias, quantidade de usuários, modelo de tráfego, entre outros). Seja qual for o caso, presume-se que as medições poderão ocorrer a fim de definir novos limiares para diferentes cenários.

AlgoritmoVHD( )

1: **if** (*RedeAtual* = *"WiMAX"*) **then**
2: **if** (*CoS* = *"rtPS"*) **then**
3: **if** ($VazãoAgregada_{WiMAX} >= TH_{VI}^{totWM}$) **and** ($VazãoAgregada_{WiFi} < TH_{VI}^{totWF}$) **then**
4: Inicia handover para rede alvo Wi-Fi
5: **else if** (*CoS* = *"UGS"*) **then**
6: **if** $VazãoAgregada_{WiMAX}$ ($>= TH_{VO}^{totWM}$) **and** ($VazãoAgregada_{WiFi} < TH_{VO}^{totWF}$) **then**
7: Inicia handover para rede alvo Wi-Fi
8: **else if** (*CoS* = *"BE"*) **then**
9: **if** ($VazãoAgregada_{WiMAX} >= TH_{BE}^{totWM}$) **and** ($VazãoAgregada_{WiFi} < TH_{BE}^{totWF}$) **then**
10: Inicia handover para rede alvo Wi-Fi
11: **end if**
12: **else**
13: **if** (*RedeAtual* = *"Wi-Fi"*) **then**
14: **if** (*AC* = *"AC_VI"*) **then**
15: **if** ($VazãoAgregada_{WiFi} > TH_{VI}^{totWF}$) **and** ($VazãoAgregada_{WiMAX} < TH_{VI}^{totWM}$) **then**
16: Inicia handover para rede alvo WiMAX
17: **else if** (*AC* = *"AC_VO"*) **then**
18: **if** ($VazãoAgregada_{WiFi} > TH_{VO}^{totWF}$) **and** ($VazãoAgregada_{WiMAX} <= TH_{VO}^{totWM}$) **then**
19: Inicia handover para rede alvo WiMAX
20: **else if** (*AC* = *"AC_BE"*) **then**
21: **if** ($VazãoAgregada_{WiFi} > TH_{BE}^{totWF}$) **and** ($VazãoAgregada_{WiMAX} <= TH_{BE}^{totWM}$) **then**
22: Inicia handover para rede alvo WiMAX
23: **end if**
24: **end if**

Figura 2. Pseudocódigo do Algoritmo VHD.

TH_{VI}^{totWM}(10Mbps): Limiar máximo ou mínimo de vazão agregada WiMAX para vídeo, se a rede atual é WiMAX ou Wi-Fi respectivamente.

TH_{VO}^{totWM}(8Mbps): Limiar máximo ou mínimo de vazão agregada WiMAX para voz, se a rede atual é WiMAX ou Wi-Fi respectivamente.

TH_{BE}^{totWM}(6Mbps): Limiar máximo ou mínimo de vazão agregada WiMAX para melhor esforço, se a rede atual é WiMAX ou Wi-Fi respectivamente.

TH_{VI}^{totWF}(6Mbps): Limiar máximo ou mínimo de vazão agregada Wi-Fi para vídeo, se a rede atual é Wi-Fi ou WiMAX respectivamente.

TH_{VO}^{totWF}(4Mbps): Limiar máximo ou mínimo de vazão agregada Wi-Fi para voz, se a rede atual é Wi-Fi ou WiMAX respectivamente.

TH_{BE}^{totWF}(2Mbps): Limiar máximo ou mínimo de vazão agregada Wi-Fi para melhor esforço, se a rede atual é Wi-Fi ou WiMAX respectivamente.

Independente de qual seja a rede candidata a ser selecionada para a realização de *handover* (Wi-Fi ou WiMAX), considera-se que o MN pode iniciar sua sessão em qualquer cobertura WiMAX ou Wi-Fi. Assim, o estudo considera duas direções de mapeamento de QoS. Em primeiro lugar, o algoritmo verifica o AC/CoS do MN, dependendo de sua rede atual e alvo. Em seguida, a comparação da vazão agregada da rede atual com o limiar máximo pré-definido, bem como a vazão agregada da rede alvo em comparação com o limiar mínimo pré-definido, para o mapeamento de QoS correspondente.

Uma questão interessante a notar, é que um MN com fluxo de menor prioridade é mais propício a realizar *handover* do que um MN de maior prioridade. Isto beneficia o balanceamento de carga, pois permite que em caso de congestionamento, por exemplo, o tráfego BE desocupe a célula atual a fim de melhorar seu desempenho, pois em geral, esta classe será a primeira a ser degradada de acordo com as prioridades dos fluxos. Desta forma, em nosso cenário, uma rede WiMAX com vazão agregada de 6 Mbps, é o suficiente para prejudicar um fluxo BE que esteja competindo pelo acesso ao meio com fluxos de maior prioridade. Em resumo, nossa proposta leva em conta o mapeamento de QoS afim de manter a continuidade do serviço e também evita que fluxos de baixa prioridade sofram degradação em células sobrecarregadas, portanto, promovendo também balanceamento de carga.

A Figura 3 mostra as mensagens de sinalização do MIH em conjunto com o algoritmo VHD no MN, que é responsável pela decisão de *handover*. A figura ilustra o cenário onde o MN com classe de serviço BE está se movendo a partir da área de cobertura WiMAX, que já está saturada, e então decide realizar *handover* para uma cobertura sobreposta Wi-Fi. A sequência de sinalização é descrita a seguir.

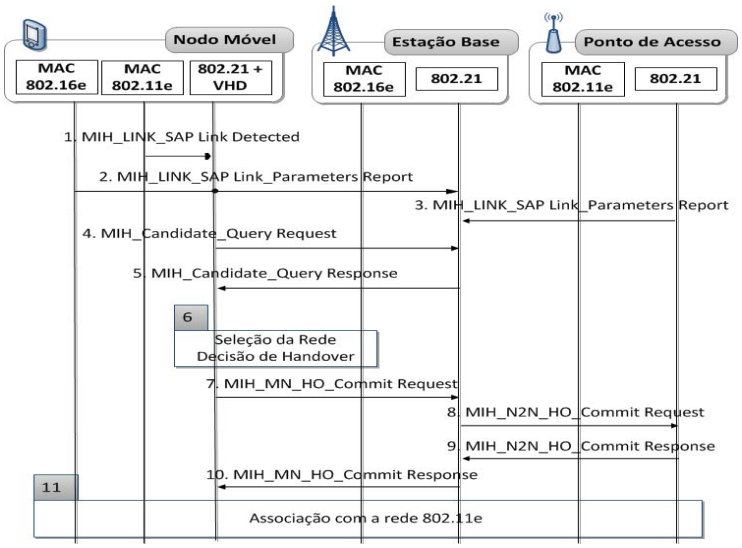


Figura 3. Sinalização durante o *handover* vertical.

1. Em primeiro lugar, o MN detecta uma rede vizinha Wi-Fi por meio da mensagem MIH_LINK_SAP Link_Detected.

2. A interface WiMAX envia para o MIH sua vazão atual (MIH_LINK_SAP Link_Parameters_Report). O MIH repassa o valor da vazão pela rede WiMAX para a BS atual.

3. O MIH da rede alvo (Wi-Fi) também envia a vazão atual para a BS (MIH_LINK_SAP Link_Parameters_Report). Supõe-se que, se esta informação existe, ela foi enviada anteriormente para o AP Wi-Fi por meio de um dos seus MNs.

4. Apesar de o MN já ter listado a sua rede alvo, ele envia para a BS uma requisição de consulta de redes candidatas disponíveis (MIH_Candidate_Query Request). A BS realiza sucessivas trocas de mensagens com o AP a fim de requisitar informações de recursos.

5. O resultado da consulta é enviado para o MN (MIH_Candidate_Query Response), juntamente com o resultado da soma da vazão de todos os nodos, tanto dos que estão no WiMAX, quanto dos que estão no Wi-Fi.

6. Neste ponto, o MN tem informação suficiente sobre a rede alvo, para então tomar a decisão de realizar *handover* ou não. Como o AP é o único disponível, o mesmo é selecionado e a decisão final fica por conta do resultado das vazões agregadas em função das classes de serviço, que neste caso é BE. Como a vazão agregada no WiMAX é maior que o limiar máximo (Sum_WiMAX > 6Mbps) e a vazão agregada no Wi-Fi é menor que o limiar mínimo (Sum_Wi-Fi < 2Mbps), então o MN irá iniciar o processo de associação com a rede Wi-Fi.

7. O MN envia uma mensagem de notificação para a BS com informações sobre o AP alvo (MIH_MN_HO_Commit Request).

8. A BS então informa ao AP alvo (MIH_N2N_HO_Commit Request) que o MN irá se mover para sua área de cobertura.

9. O AP alvo responde para a BS, autorizando o início do *handover* (MIH_N2N_HO_Commit Response).

10. A BS repassa a autorização para o MN (MIH_MN_HO_Commit Response).

11. A interface 802.11e associa-se ao AP alvo.

Com base nas mensagens de sinalização descritas acima, é importante mencionar que em nossa proposta, o MIH foi estendido para ter a capacidade de coletar a vazão atual no instante em que uma rede vizinha é detectada, bem como enviar esta informação para a PoA atual, responsável pelo calculo da vazão agregada (rede atual e alvo) em um dado instante.

4. Resultados das Simulações

Nesta seção, os resultados das simulações para a proposta do esquema de mapeamento de QoS e algoritmo de balanceamento de carga são apresentados. No ns-2 [Network Simulator NS-2 2005], o módulo NIST Mobility [Nist Mobility 2009] foi modificado para adaptação e integração dos módulos de QoS para WiMAX [Belghith and Nuaymi 2009] e Wi-Fi [TKN Group 2006], coleta de vazão agregada via MIH e inclusão do algoritmo VHD.

A topologia utilizada nas simulações está ilustrada na Figura 4. Para todas as simulações, a rede infra-estruturada é formada por um servidor web, quatro roteadores,

uma BS (802.16e), um AP (802.11e) e MNs equipados com MIH e duas interfaces (802.16e/802.11e).

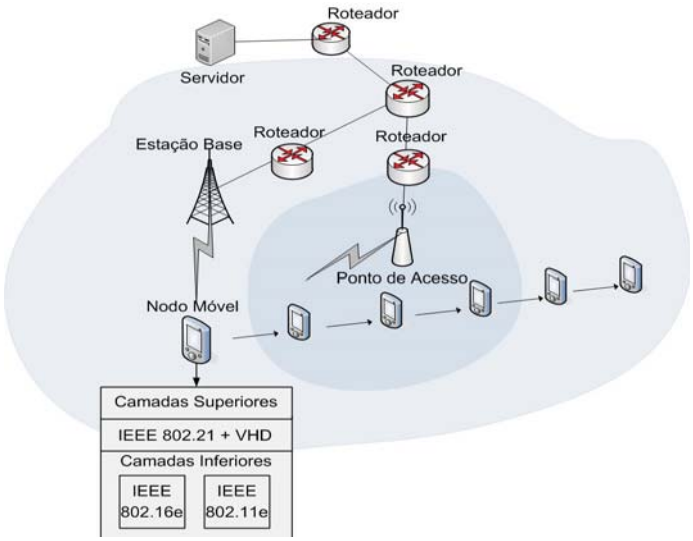


Figura 4. Topologia da Rede.

Os principais parâmetros utilizados nas simulações são apresentados na Tabela 2 a seguir.

Tabela 2. Parâmetros usados nas simulações.

	802.11e	**802.16e**	**Rede Cabeada**
Taxa de transmissão	11 Mbps	75 Mbps	10 Mbps
Raio de Cobertura	50 m	1000 m	-
Escalonador	-	Round Robin (RR)	-
Tipo de Vídeo	Resolução: 176 x 144 Taxa de Frames: 30 frames/s Modo de cores: Y, U, V		
Tipo de Fila	Drop Tail (40 ms de atraso)		
Tamanho de pacotes	1052 bytes		
Fragmentação máxima de pacotes	1024 bytes		
Tempo de cada Simulação	50 segundos		
Número de simulações para cada cenário	100		
Intervalo de Confiança	95 %		

4.1. Mapeamento de QoS

Neste primeiro cenário, três MNs estão equipados com duas interfaces (WiMAX e Wi-Fi), cada um recebe do servidor (direção *downlink*) um diferente tipo de tráfego (*streaming* de vídeo, voz e FTP). Todos os fluxos estão configurados com taxa de 3 Mb/s. Todos os MNs estão, inicialmente, em uma única cobertura WiMAX. Como os MNs se movem a 5 metros por segundo (18 km/h), eles entrarão em uma área sobreposta (cobertura Wi-Fi e WiMAX). À medida que os MNs continuam se movendo, eles retornam para uma única cobertura pertencente à célula WiMAX.

Na Figura 5, como não há suporte à QoS para ambas as tecnologias, os fluxos não são classificados ou escalonados para CoSs ou ACs. Durante a simulação, apesar de os três fluxos atingirem a vazão máxima na rede WiMAX, não existe diferenciação de

tráfego e, depois do *handover* para o Wi-Fi, a vazão diminui de forma acentuada para todos os fluxos, prejudicando tráfegos sensíveis ao atraso e perda, como voz e vídeo. No momento em que os MNs retornam para a célula WiMAX, a vazão permanece a mesma para todos os fluxos, ou seja, não existe prioridade entre eles para o acesso ao meio.

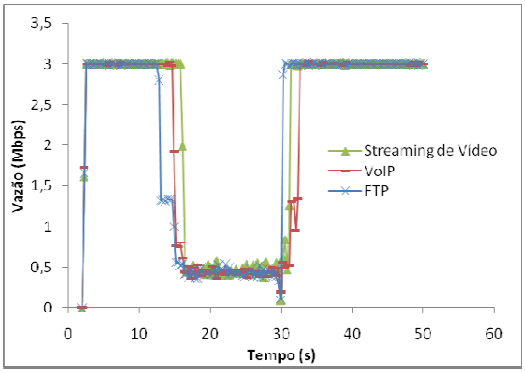


Figura 5. Vazão do três fluxos sem o mapeamento de QoS.

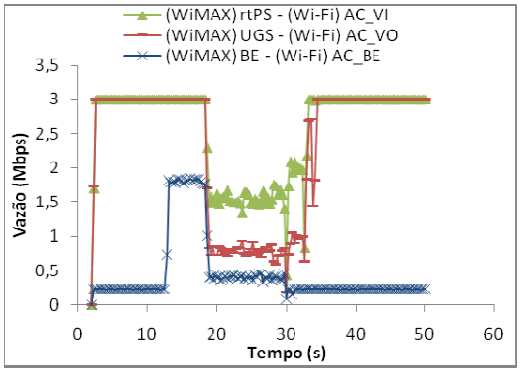


Figura 6. Vazão do três fluxos com o mapeamento de QoS.

Conforme a Figura 6, os MNs configurados com fluxos UGS e rtPS tem vazão máxima até o instante 19s quando, então, eles realizam *handover* e seus fluxos são mapeados para AC_CO e AC_VI, respectivamente na rede Wi-Fi. O MN com fluxo BE na cobertura WiMAX, tem a menor vazão até o instante 13s quando realiza *handover* e seu fluxo é mapeado para AC_BE na rede Wi-Fi. Como mostrado na Figura 6, o MN com fluxo BE chega a ter um aumento repentino da vazão para 1,8 Mbps, pois encontra o canal livre até o instante 19s. Os MNs realizam mais um *handover* de volta para a rede WiMAX, de modo que os fluxos com categorias de acesso AC_VO, AC_VI e AC_BE são mapeados para UGS, rtPS e BE, respectivamente.

4.1.1 Avaliação das Métricas de QoE

A métrica mais tradicional de QoE é o PSNR (*Peak Signal to Noise Ratio*) [Winkler 2005], que estima a qualidade do vídeo em decibéis, comparando o vídeo original com o vídeo recebido pelo usuário considerando o aspecto de luminosidade. Para cada faixa de valores de PSNR, há uma qualificação para o vídeo que foi recebido pelo usuário, conforme descrito na Tabela 3.

Tabela 3. Valores de Classificação do PSNR.

PSNR (db)	> 37	31 – 37	25 – 31	20 – 25	< 20
Qualidade	Excelente	Bom	Aceitável	Pobre	Péssimo

A métrica SSIM (*Structural Similarity Index*) [Wang *et al* 2004] faz a comparação quadro a quadro do vídeo original com o vídeo recebido pelo usuário, considerando os seguintes aspectos: contraste, luminosidade e estrutura. O SSIM é expresso como um valor decimal entre 0 e 1. Quanto mais próximo do valor 1, melhor é a qualidade do vídeo.

A métrica VQM (*Video Quality Metric*) [Xiao 2000] também compara o vídeo original em relação ao vídeo recebido pelo usuário, considerando os seguintes aspectos:

embasamento, ruído, distorção dos frames e cor. Quanto mais próximo o valor for de 0, melhor será a qualidade do vídeo.

O cenário para a avaliação do vídeo é similar ao primeiro cenário. A ferramenta Evalvid [Evalvid 2007], que permite a transmissão real de vídeo no ns-2, foi utilizada para avaliar a qualidade do vídeo recebido. Uma sequência de frames do vídeo "*Highway*" [YUF CIF 2007] foi utilizada nas simulações para o tráfego de vídeo. O vídeo consiste de 2000 quadros com o formato YUV, amostragem 4:2:0, dimensão de 176x144, comprimido através de um codec MPEG-4 e enviado a uma taxa de 30 quadros por segundo. O objetivo da avaliação do vídeo, é validar a eficácia do mapeamento de QoS proposto no ponto de vista do usuário.

Conforme a Tabela 4, o vídeo recebido pelo usuário sem qualquer garantia de QoS em ambas a redes WiMAX e Wi-Fi, obteve média PSNR, SSIM e VQM igual 22,17 dB, 0,68 e 6,47 respectivamente, o que qualifica o vídeo como POBRE. O vídeo recebido com o mapeamento de QoS proposto obteve média PSNR, SSIM e VQM igual a 32,1 dB, 0,91 e 1,96 respectivamente, qualificando o vídeo como BOM.

Tabela 4. Resultados das métricas de QoE para o tráfego de vídeo.

	PSNR [dB]	**SSIM**	**VQM**
Sem Mapeamento de QoS	22,17	0,68	6,47
Com Mapeamento de QoS	32,1	0,91	1,96

A Tabela 5 ilustra a sequência de três quadros de frames de vídeo. Verifica-se que o vídeo recebido sem o mapeamento de QoS foi bastante degradado no momento em que o MN estava na célula WiMAX, e após o *handover* para a rede Wi-Fi, a qualidade do vídeo piora ainda mais. O mesmo acontece quando o MN retorna para o WiMAX. Com o mapeamento de QoS, a continuidade da boa qualidade do vídeo é a mesma independente se o MN está no WiMAX ou Wi-Fi.

Tabela 5. Frames do vídeo "*Highway*".

Qualidade do Vídeo sem Mapeamento de QoS		
FRAME [850] - (WiMAX)	**FRAME [1309] – (Wi-Fi)**	**FRAME [1806] – (WiMAX)**
		
Qualidade do Vídeo com Mapeamento de QoS		
FRAME [850] – (WiMAX)	**FRAME [1309] - (Wi-Fi)**	**FRAME [1806] – (WiMAX)**
		

4.2. Suporte ao Balanceamento de Carga

No segundo cenário, nove MNs estão equipados com duas interface (WiMAX e Wi-Fi). Todos os MNs recebem do servidor (direção *downlink*) diferentes tipos de tráfegos, ou seja, um grupo de três MNs recebe *streaming* de vídeo, outro grupo de três MNs recebe voz e os três restantes recebem dados pelo protocolo FTP. Todos os fluxos estão configurados com taxa de 1,5 Mb/s. Nesta análise, os nove MNs estão inicialmente em uma única cobertura WiMAX. Três MNs com CoS BE e três MNs com CoS rtPS, se movem a 5 metros por segundo (18 Km/h) em direção a uma área sobreposta (cobertura WiMAX e Wi-Fi) e as três SSs com CoS UGS estão estáticos gerando tráfego de fundo.

A Figura 10, ilustra a vazão média dos seis MNs a serem analisados antes e depois do *handover* para a rede Wi-Fi. Quando todos os MNs estão na célula WiMAX, os MNs com fluxo BE não possuem vazão suficiente para transmitir dados. Isso acontece porque a rede está saturada por MNs com fluxos rtPS e UGS. Como não existe nenhum controle de *handover* por um sistema inteligente ou algoritmo, os seis MNs com fluxos BE e rtPS realizam *handover* para a rede Wi-Fi. Podemos notar que a vazão dos fluxos AC_BE teve uma pequena melhora. Por outro lado, um dos fluxos AC_VI teve um decremento da vazão (1,2 Mbps) comprometendo a qualidade do vídeo.

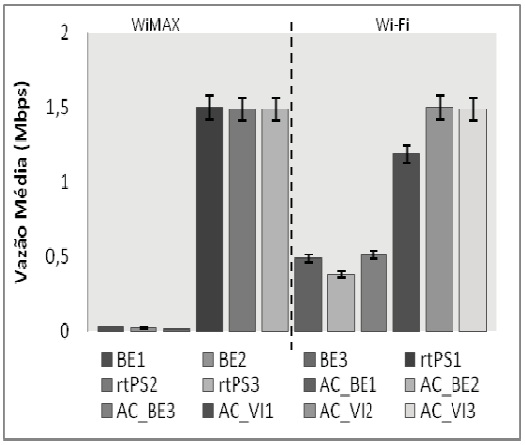


Figura 10. Vazão Média sem o algoritmo VHD.

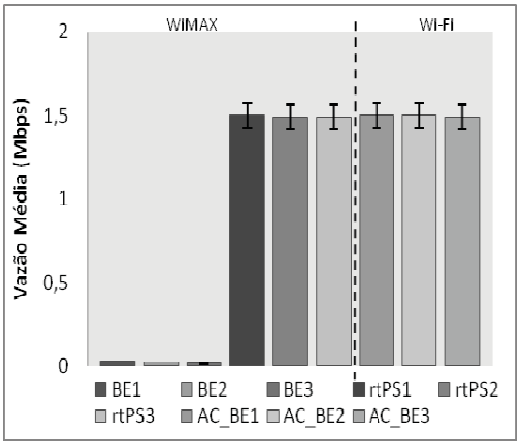


Figura 11. Vazão Média com o algoritmo VHD.

Como pode ser visto na Figura 11, a implementação do algoritmo VHD promove uma melhor distribuição de carga entre as células da rede heterogênea. Neste caso, antes mesmo de realizar *handover* para o Wi-Fi, os MNs com fluxo BE, verificam se a vazão agregada da rede WiMAX, cujo o valor é 9,2 Mbps, é maior do que 6 Mbps. O valor da vazão agregada da rede Wi-Fi alvo é de 0 Mbps, ou seja, até o momento não há nenhum MN gerando tráfego. Este valor é comparado com o limiar mínimo que é de 2 Mbps. Então, com as duas condições satisfeitas, os três MNs com fluxo BE realizam *handover*. Os MNs com fluxo rtPS, também comparam a vazão agregada atual (9,2 Mbps) com o limiar máximo que é de 10 Mbps, bem como a vazão agregada atual da rede Wi-Fi alvo, cujo o valor é de 4,5 Mbps (os três MNs com AC_BE já estão no Wi-Fi com vazão máxima de 1,5 Mbps), comparado ao limiar mínimo de 6 Mbps. Como a primeira condição não foi satisfeita, os três MNs com fluxo rtPS não realizam *handover* para o Wi-Fi.

O resultado do uso do algoritmo VHD juntamente com o esquema de mapeamento de QoS é apresentado na Figura 11. Os três fluxos rtPS não realizam *handover* e, portanto, permanecem com CoS rtPS e vazão máxima de 1,5 Mbps. Os três fluxos BE realizam *handover* e são mapeados para AC_BE na rede Wi-Fi, além de atingirem uma vazão máxima de 1,5 Mbps.

5. Conclusões e Trabalhos Futuros

Este artigo descreve o desenvolvimento de um arcabouço para mapeamento de QoS e balanceamento de carga em redes sem fio heterogêneas WiMAX/Wi-Fi de modo a garantir QoS e QoE, bem como mobilidade transparente em ambiente heterogêneo. Além disso, um algoritmo VHD foi incluso com o objetivo de controlar o *handover* dos nodos móveis e prover balanceamento da carga entre redes heterogêneas.

Conforme os resultados de varias simulações, a QoS e QoE podem ser mantidas durante a mobilidade do MN, independente da tecnologia de acesso ser Wi-Fi ou WiMAX. A mobilidade transparente e controle na decisão de *handover* têm sido garantidos com o auxílio do MIH. Diferentemente de nossa proposta, muitos artigos da literatura permitem a deterioração de tráfego não prioritário com o intuito de dar garantias ao tráfego de classes/categorias com requisitos estritos de QoS. Nossa proposta permitiu que tanto nodos com fluxos de maior quanto de menor prioridade tivessem seus requisitos de QoS satisfeitos, mesmo em situações em que a rede está congestionada.

Para trabalhos futuros, propõem-se adicionar a classe de serviço nrtPS e a categoria de acesso AC_BK para as redes WiMAX e Wi-Fi, respectivamente. Além disso, iremos propor o mapeamento dinâmico, com base nas mudanças dos parâmetros de rede e experiência do usuário.

Agradecimentos

Os autores agradecem a FAPESPA (processo 2009/185122) e CNPq (processos 475814/2008-8 e 309142/2008-3), pelo apoio ao desenvolvimento deste trabalho.

Referências

Andi, W. C., Yeo, C. K. and Lee, B. S. (2010) “Environment-aware QoS framework for multi-interface terminal”. Elsevier Journal on Computer Communications, vol. 33, pp. 1049-1055, January 2010.

Belghith, A. and Nuaymi, L. (2009) “Design and Implementation of a QoS-included WiMAX Module for NS-2 Simulator”. http://perso.telecom-bretagne.eu/aymenbelghith/tools/.

Cerqueira, E., Veloso, L., Curado, M., Monteiro, E. and Mendes, P. (2008) “Quality Level Control for Multi-user Sessions in Future Generation Networks”. Global Telecommunications Conference. IEEE Globecom 2008. University of Coimbra, Coimbra.

Chen, Y. C., Hsia J. H. and Liao, Y. J. (2008) “Advanced seamless vertical handoff architecture for WiMAX and WiFi heterogeneous networks with QoS guarantees”. Elsevier Journal on Computer Communications, vol. 32, pp. 281-293, November 2008.

Evalvid Tool. http://www.tkn.tu-berlin.de/research/evalvid/.

IEEE 802.11e (2005), "Status of Project IEEE 802.11e, MAC Enhancements for Quality of Service," IEEE Standard, 802.11e.

IEEE 802.16e-2005 (2005), "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, Corrigendum 1".

IEEE 802.21-2008 (2008), "Draft Standard for Local and Metropolitan Area Networks: Media Independent *Handover* Services". IEEE Standard, P802.21/D11.0.

IEEE P802.11e/D13.0 (2005), "Medium Access Control (MAC) Quality of Service (QoS) Enhancements", IEEE Standard.

Oliva, A., Banchs, A., Soto, I., Lessmann, J., Niephaus, C. and Melia, T. (2011) "IEEE 802.21: Media Independence beyond Handover". Computer Standards & Interfaces, March 2011.

Seamless and Secure Mobility Tool Suite (Nist Mobility). http:// http://www.nist.gov/itl/antd/emntg/ssm_seamlessandsecure.cfm

Sekercioglu, Y. A., Ivanovich, M. and Yegin, A. (2009) "A Survey of MAC based QoS implementation for WiMAX networks". Computer Networks Journal, May 2009.

Tarng, W., Chen, N. W., Deng, L. Z., Ou, K. L., Hsie, K. R. and Chen, M. (2010) "The Integration of Heterogeneous Wireless Networks (IEEE 802.11/IEEE 802.16) and its QoS Analysis". International Journal of Communication Networks and Information Security, vol. 2, No. 3, December 2010.

Technical University Berlin Telecommunication Networks Group TKN. http://www.tkn.tu-berlin.de/research/802.11e_ns2/.

The Network Simulator NS-2. http://www.isi.edu/nsnam/ns/.

Wang, Z., Lu, L. and Bovik, A.C. (2004) "Video quality assessment based on structural distortion measurement, Signal Processing: Image Communication". vol. 19, no. 2.

Winkler, S. (2005) "Perceptual video quality metrics – a review, in Digital Video Image Quality and Perceptual Coding", eds. H. R. Wu, K. R. Rao, cha 5, CRC Press.

Xiao, F. (2000) "DCT-based Video Quality Evaluation MSU Video Quality Metric". http://compression.ru/video/quality_measure/vqm.pdf.

YUF CIF. DOI= http://www.tkn.tu-berlin.de/research/evalvid/cif.html.

XVI Workshop de Gerência e Operação de Redes e Serviços

♦

Sessão Técnica 3

Gerenciamento de Serviços e Aplicações

Monitoramento Colaborativo de Trânsito utilizando Redes IEEE 802.11 em Cidades Inteligentes

José Geraldo Ribeiro Júnior[1,2], **Miguel Elias Mitre Campista** [1], **Luís Henrique Maciel Kosmalski Costa**[1]

[1]Grupo de Teleinformática e Automação – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

[2]Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG
Leopoldina – MG – Brasil

{jgrjunior,miguel,luish}@gta.ufrj.br

Abstract. *The increase in popularity of the IEEE 802.11 wireless technology and schemes that offer Internet access to whole cities, it has become possible to use these networks to monitor the traffic. This paper propose a strategy to monitor traffic, in a collaborative form, using these networks whithout the necessity of new investiment or modifications. Our strategy does not alter the network infrastructure to obtain sufficient information about vehicle movement. To test this proposal, we collect and analyze the data in a real scenery and the results obtained are discribed in this paper.*

Resumo. *Com a popularização da tecnologia sem-fio IEEE 802.11 e a realização de projetos que disponibilizam Internet a cidades inteiras, surge a possibilidade de utilizar estas estruturas para fazer o monitoramento do trânsito. Este artigo propõe uma estratégia para monitorar o trânsito, de forma colaborativa, utilizando estas redes sem a necessidade de novos investimentos ou modificações. Ao usuário, basta possuir um equipamento com interface IEEE 802.11. A estratégia proposta não altera a infraestrutura da rede para obter informações sobre a movimentação do veículo. Para testar a viabilidade da proposta, foram coletados e analisados dados de um cenário real e os resultados obtidos são descritos no artigo.*

1. Introdução

O aumento constante do número de veículos em circulação, especialmente em grandes cidades, tem como consequência direta o aumento dos engarramentos. Grandes capitais como a cidade do Rio de Janeiro convivem diariamente com engarrafamentos que se estendem por quilômetros, exigindo que ações preventivas sejam vistas como prioridade. Situações rotineiras como pequenas colisões, reparos na via ou até reduções bruscas de velocidade podem ser suficientes para gerar grandes engarrafamentos. Uma consequência direta deste problema é o aumento da emissão de gás carbônico (CO_2) e outros poluentes na atmosfera, pois o ritmo lento exige constantes reacelerações. Segundo o INEA (Instituto Estadual do Ambiente) do Rio de Janeiro, os veículos são responsáveis por 77% do total de poluentes emitidos para a atmosfera [INEA 2004].

Diferentes propostas para minimizar os problemas no trânsito necessitam de um sistema de monitoramento eficiente e automatizado. No entanto, a forma mais

utilizada para monitorar grandes vias nas principais cidades do Brasil ainda é o uso de câmeras de vídeo, operadas por pessoas, onde todo o controle do tráfego é feito de forma visual. Outras propostas têm surgido utilizando sensores nos carros ou nas vias [Edelmayer et al. 2010] ou combinando o uso de GPS (*Global Positioning System*) com GPRS (*General Packet Radio Service*) ou IEEE 802.11 [Ott and Kutscher 2011, Akkihebbal et al. 2011]. No entanto, o uso de sensores exige alto investimento pois ainda é pouco utilizado. Outro inconveniente é que exigiria um tempo para instalação de toda a rede até ela se tornar operacional. Esse tempo pode inviabilizar o uso de sensores pois o problema é urgente. Já o GPS, que conta com um bom número de usuários no Brasil, é usado apenas para a definição da localização do motorista, pois para enviar estes dados via satélite para uma central de processamento o custo seria proibitivo devido à alta potência necessária. Algumas propostas utilizam GPRS ou IEEE 802.11 para envio dos dados. O uso do GPRS possui como vantagem uma maior cobertura pois utiliza a rede de telefonia celular. Porém, a taxa de *upload* é muito baixa quando comparada ao IEEE 802.11. Por outro lado, apesar de 11% do total de acessos ser via rede 3G (banda larga móvel), segundo o relatório de 03/2011 da Anatel (Agência Nacional de Telecomunicações), esta opção ainda apresenta um alto custo de utilização.

É fato que a popularização da tecnologia sem-fio IEEE 802.11 fez crescer muito o número de APs (*Access Point*) existentes nas cidades. Vários projetos têm iluminado[1], por exemplo, determinados lugares da cidade do Rio de Janeiro, como a orla, bairros e avenidas [UFRJ 2008]. Essa realidade possibilita uma expansão no número de aplicações e serviços que podem ser disponibilizados através dessas redes. Por meio de estruturas assim é possível captar, analisar e integrar dados de diferentes serviços, permitindo responder com inteligência a certas necessidades do usuário. Entre estes serviços está o auxílio no monitoramento de trânsito. Um exemplo de via nestas condições é a Avenida Brasil, na cidade do Rio de Janeiro, que desde 2009 está totalmente iluminada.

Este artigo tem como objetivo propor uma estratégia para realizar o monitoramento de trânsito, de forma colaborativa, utilizando estruturas já existentes de redes IEEE 802.11. Para isso os nós clientes, neste caso os veículos, utilizam informações obtidas nos *beacons* recebidos e as enviam para uma central de gerenciamento, que é responsável por tratar os dados e divulgar informações que sejam úteis para os motoristas. Por meio destas informações deverá ser possível estimar a localização do veículo na via, além da direção e velocidade em que o veículo se desloca. O nó cliente portanto, é fundamental no processo uma vez que é ele quem envia todas as informações que serão utilizadas pela central. O diferencial desta proposta está na utilização de uma estrutura de rede já em funcionamento, sem a necessidade de novos investimentos ou modificações. Ao usuário, basta possuir um equipamento com interface IEEE 802.11, e a aplicação que será desenvolvida, para usufruir dos serviços da rede.

Experimentos realizados na Avenida Brasil mostram que é possível detectar a aproximação e o distanciamento do veículo com relação aos APs, utilizando apenas as informações obtidas por meio dos *beacons* recebidos, em diferentes faixas de velocidade. Embora os *beacons* tenham como papel principal anunciar os APs disponíveis redes IEEE 802.11, essas mensagens também podem ser usadas para extrair informações adicionais para conhecimento das condições do trânsito na via em tempo real. A proposta é baseada

[1] *Uma área iluminada corresponde a uma área com cobertura Wi-fi.*

neste cenário mas se estende a todos os cenários com características semelhantes. Outra característica desta proposta é que não é necessário que 100% dos veículos que circulam pela via enviem informações, sendo possível uma coleta de dados por amostragem. Os resultados obtidos em experimentos práticos com o sistema proposto confirmam a possibilidade da obtenção das condições do trânsito em tempo real.

O restante deste artigo está estruturado da seguinte forma: a Seção 2 apresenta alguns projetos relacionados enquanto a Seção 3 apresenta detalhes do sistema proposto neste artigo. A Seção 4 descreve os experimentos realizados, bem como seus resultados. Finalmente a Seção 5 conclui o artigo e apresenta as direções futuras.

2. Projetos Relacionados

A proposta descrita neste artigo faz uso de uma estrutura existente na Avenida Brasil, na cidade do Rio de Janeiro. Graças ao projeto Avenida Brasil Digital, realizado pela SECT (Secretaria Estadual de Ciência e Tecnologia), com recursos da FAPERJ (Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro) e coordenação técnica de pesquisadores da UERJ (Universidade do Estado do Rio de Janeiro), são 58km de avenida totalmente cobertos por 163 APs, dispostos na avenida a cada 400m [UERJ 2010]. A conexão com os usuários utiliza a frequência 2,4 GHz e o padrão IEEE 802.11b/g. Segundo o governo do estado, a largura da banda é de 10Mb/s para compartilhamento entre os usuários. Todos os APs estão configurados no mesmo ESSID (*Extended Service Set ID*), o que significa que durante a movimentação do veículo só acontece transição entre BSS (*Basic Service Set*) do mesmo ESSID. Desta forma, o processo de reassociação entre AP e STA (*station*) é transparente para o usuário, não havendo perda de conexão durante o *handoff* [IEEE 2003].

O projeto CarTel [Bychkovsky et al. 2006], desenvolvido pelo MIT (*Massachusetts Institute of Technology*), é um sistema projetado para coletar, processar, distribuir e visualizar dados de sensores localizados em nós móveis, como carros. O objetivo é modelar o tráfego de uma área, permitindo definir rotas alternativas caso problemas sejam detectados. Para isso, cada nó possui um computador de bordo, semelhante a um celular, acoplado a um conjunto de sensores. Cada nó recolhe e processa as leituras localmente antes de entregá-los para uma central, por meio de redes IEEE 802.11, onde estes dados são armazenados em um banco de dados para posterior análise e visualização. CarTel foi implantado inicialmente em seis carros mas atualmente conta com cerca de 50, sendo 40 taxis, em Boston - EUA [Bychkovsky et al. 2011].

O projeto DieselNet [Corner et al. 2011] da Universidade de Massachusetts Amherst, nos Estados Unidos, consiste em uma rede formada por 40 ônibus da cidade de Amherst, cobrindo uma área de aproximadamente 240 km^2. Os ônibus ligam-se uns aos outros de forma intermitente e são equipados com computadores com recursos limitados conectados a três rádios IEEE 802.11b: (1) um ponto de acesso para prover acesso DHCP (*Dynamic Host Configuration Protocol*) aos passageiros e transeuntes, (2) uma interface que constantemente procura outros ônibus e anúncios DHCP e (3) um rádio de longo alcance que permite a comunicação com as estações receptoras das informações coletadas. Além disso, um dispositivo GPS registra periodicamente o horário e a localização de cada ônibus. O software embarcado permite realizar atualizações das aplicações e capturar informações sobre a mobilidade dos ônibus, a conectividade entre AP e ônibus

e vazão da rede. Para aumentar a conectividade entre os nós móveis da DieselNet, foram projetados *throwboxes* que funcionam como roteadores estacionários, armazenando e enviando dados entre os ônibus. Os *throwboxes* são colocados em prédios ao longo das rotas dos ônibus e utilizam baterias ou energia solar para funcionar.

O projeto "DTN for Urban Environment"[Akkihebbal et al. 2011], desenvolvido na NUS (*National University of Singapore*), tem por objetivo aplicar conceitos de DTN (*Delay-tolerant networking*) [McMahon and Farrell 2009] em ambientes urbanos, onde os nós móveis são pedestres e veículos. Até então quatorze ônibus foram equipados com roteadores LinkSys WRT54GL, ou superiores, para fazer o papel de clientes móveis. Estes clientes utilizam o sinal da rede IEEE 802.11 NUSOPEN para enviar sua localização. O projeto disponibiliza uma interface Web para visualizar a localização dos ônibus. Assim como nesta proposta, o projeto utiliza uma infraestrutura pública já disponível para o envio de dados de localização por parte dos clientes. No entanto, não há garantias de continuidade de conexão pois os APs estão distantes uns dos outros e cada um possui uma conexão independente com a Internet.

O projeto Drive-thru Internet [Ott and Kutscher 2011], desenvolvido pela Universidade de Tecnologia de Helsinki, tem como objetivo prover acesso à Internet a usuários em veículos trafegando dentro de uma cidade, em vias de alta velocidade. Pontos fixos de acesso à Internet, utilizando a tecnologia IEEE 802.11, são espalhados de forma que estejam interconectados provendo tanto serviços locais quanto acesso à Internet. Em função do acesso intermitente obtido pela passagem pelos pontos de acesso, Ott e Kutscher [Ott and Kutscher 2005] desenvolveram o protocolo PCMP (*Persistent Connection Management Protocol*), que habilita sessões de comunicação de longa duração na presença de conectividade intermitente. Desta forma, as conexões entre os APs e entre APs e veículos podem ser perdidas sem afetar a persistência da sessão. A sessão pode ser restabelecida no momento em que houver conectividade. O Drive-thru atua nas camadas de aplicação e transporte e não na camada de rede, o que permite suportar a falta de conexão por períodos mais longos.

Diferente das propostas descritas nesta seção, esta proposta utiliza uma estrutura de rede já em funcionamento, não dedicada e sem a necessidade de nenhuma configuração ou alteração em sua estrutura. O nó cliente também não depende de nenhum hardware específico, sendo o único requisito, possuir um equipamento com interface de rede IEEE 802.11. Além disso, mesmo em cenários onde a conexão contínua não é garantida, não é necessário ao cliente suportar protocolos como DTN, por exemplo.

3. Sistema Proposto

A Figura 1 mostra a arquitetura do sistema proposto, que é composto pelo nó cliente (veículo), pontos de acesso e central de gerenciamento de dados. O objetivo do sistema proposto é detectar a movimentação dos veículos utilizando apenas informações já disponíveis nas redes. Para isso optou-se por utilizar os *beacons* que são enviados periodicamente pelos APs (por padrão, o *beacon* é transmitido a cada 100 ms). Entre as principais justificativas em utilizar *beacons* estão: (1) o fato destes já possuírem todas as informações necessários para o monitoramento proposto, como: ESSID do AP, endereço MAC (*Media Access Control*) do AP (BSSID), potência de transmissão e hora de geração do pacote e (2) ser possível capturá-los mesmo sem estar associado ao AP. Com a combi-

nação destes dados obtidos com as informações sobre a via é possível gerar informações em tempo real sobre as condições de tráfego. As informações extraídas com a recepção dos *beacons* são enviadas, posteriormente, à central para processamento. Dessa forma, o sistema é dito colaborativo, pois quanto maior a quantidade de dados recebidos na central, mais completo se torna o mapeamento das condições de trânsito da cidade.

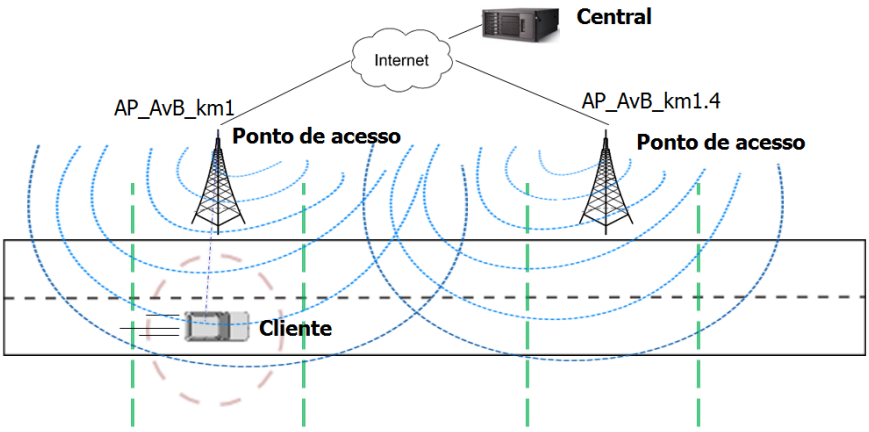


Figura 1. Arquitetura do sistema.

No cenário tratado existem pelo menos três formas de detectar a movimentação do veículo, etapa fundamental para verificar as condições do trânsito:

I) baseado no tempo gasto pelo pacote entre o envio (no AP) e o recebimento (no veículo);
II) baseado no tempo gasto entre associação e reassociação do cliente entre APs;
III) baseado no tempo gasto para percorrer distâncias entre os APs, com base nas variações na potência do sinal.

Na primeira forma, a confiabilidade e a precisão ficam seriamente comprometidas se houver uma falha na sincronização entre os relógios do AP e do cliente. Como não é possível garantir este sincronismo de forma eficiente, esta opção foi descartada.

As duas outras opções podem ser implementadas no sistema proposto. A opção II, mais simples de implementar, apresenta como desvantagem um tempo maior entre as atualizações das informações que serão apresentadas para os clientes já que o tempo de transição entre os APs pode ser alto dependendo das condições da via. A opção III, adotada nesta proposta, calcula o tempo gasto para percorrer trechos entre os APs, usando como base a potência de transmissão recebida pelo cliente. Combinando estas informações obtidas com informações sobre a via, como a velocidade por exemplo, é possível conhecer as condições do trânsito. Uma vez que esta opção foi adotada, apenas ela será mencionada a partir daqui.

Para ilustrar o funcionamento da opção adotada para detectar a movimentação do veículo, a Figura 2 apresenta um cenário próximo ao existente na Avenida Brasil. Com os APs distantes 400m uns dos outros e com raio de transmissão de cerca de 200 a 300m[2] há alta probabilidade de continuidade de conexão. Com isso, a proposta consiste em dividir

[2]*Como não há nenhuma informação disponível sobre o hardware utilizado, optou-se por considerar o raio de 200m para os cálculos, igual a metade da distância entre os APs na Avenida Brasil.*

a área coberta por cada AP em três zonas (ilustradas como áreas de A a F (2 APs)). Cada zona equivale a um intervalo de potência de transmissão, medido no cliente. Por exemplo, o raio de transmissão do AP AP_AvB_km1 corresponde às zonas A, B e C. A zona A corresponde ao intervalo -70dBm e -90dBm, enquanto a zona B corresponde ao intervalo -20dBm e -69dBm e a zona C ao intervalo -70dBm e -90dBm. Desta forma, a medida que o carro se movimenta há uma alteração entre estes intervalos. Os limites que definem as zonas foram escolhidos com base no resultado apresentado nos experimentos práticos, que serão apresentados na Seção 4. Nos resultados obtidos, a potência -70 dBm aparece como um valor médio entre os valores superiores e inferiores obtidos. A Figura 3 ilustra a representação esperada do sinal do AP, tendo como base que o tamanho da área coberta pelo AP (2*raio) corresponde aos períodos em destaque na figura.

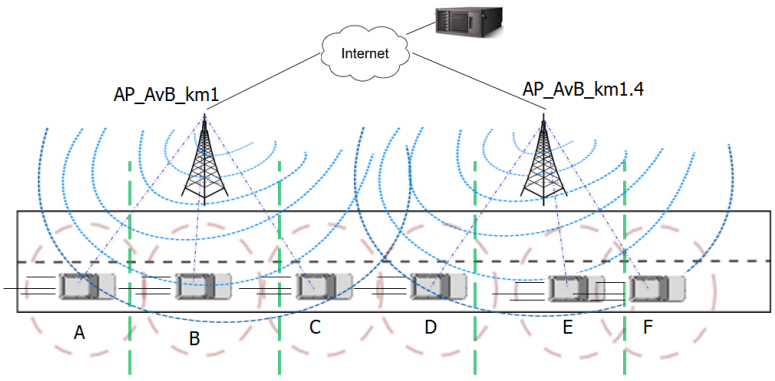


Figura 2. Divisão da área de cobertura em zonas.

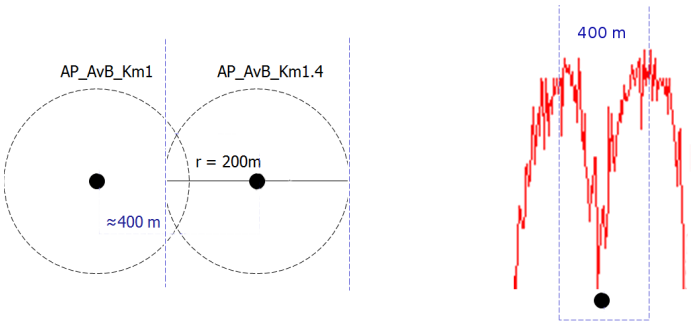


Figura 3. Representação da área de cobertura do sinal.

A aplicação que roda no nó cliente possui um cadastro de ESSID das redes conhecidas. Ao iniciar a aplicação cliente, uma interface de rede é criada no modo monitor (mon0) fazendo com que o cliente passe a monitorar as redes até que receba algum *beacon* de uma rede conhecida. A razão de criar uma interface no modo monitor é que assim será possível capturar os pacotes mesmo sem estar associado a algum AP. Ao receber um *beacon* de algum ESSID conhecido o nó cliente tentará se associar ao AP que possui o MAC enviado pelo *beacon* (BSSID), utilizando a interface de rede wireless (wlan0) que não está funcionando no modo monitor. Caso o cliente não consiga se associar, ele tentará novamente até ter sucesso. Em cada ciclo de captura o nó cliente obtém informações atualizadas sobre o AP ao qual está associado e sobre a potência de transmissão. O MAC do AP, o MAC do cliente, a potência de transmissão e o horário de geração do *beacon* são enviados para a central de dados por meio de um formulário simples que chama um método

POST e são armazenados em um banco de dados. A central de dados irá cruzar as informações recebidas com as informações existentes sobre a via e sobre *beacons* anteriores para obter informações relevantes sobre o trânsito naquele momento. Estas informações serão divulgadas via Internet por meio de uma interface que ainda será desenvolvida. A Figura 4 ilustra o processo de troca de informação entre cliente e servidor.

É importante observar que, no caso de haver outras redes com o mesmo ESSID de uma das redes conhecidas, o cliente tentará se associar. No entanto, mesmo que o cliente se associe e transmita dados para a central, os dados não serão armazenados pois o MAC do AP ao qual o nó cliente está associado não é conhecido pela central. Por isso é importante ter um cadastro dos equipamentos da via, incluindo o MAC e a posição geográfica de cada AP. A seção a seguir apresenta outros requisitos do sistema.

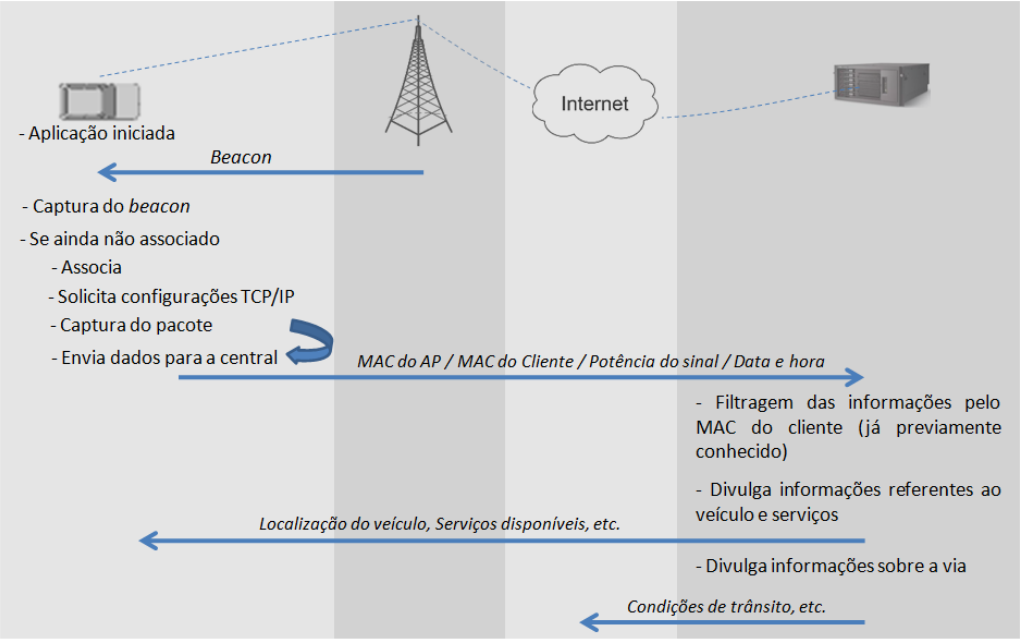


Figura 4. Funcionamento do sistema proposto.

3.1. Requisitos do Sistema Proposto

A proposta apresentada neste artigo é baseada no cenário existente na Avenida Brasil, na cidade do Rio de Janeiro, mas pode ser aplicada em qualquer cenário com características semelhantes. Os requisitos são:

- cobertura IEEE 802.11 em toda extensão da via ou em partes dela;
- disseminação de pacotes *beacons* pelos APs;
- clientes como *smartphone* ou *laptop* utilizando um equipamento com interface 802.11 e rodando a aplicação proposta;
- mapeamento da posição geográfica dos APs.

Ter cobertura em toda extensão da via é interessante para outros fins como acessar a Internet mas, para o monitoramento do trânsito, não é fundamental. É possível obter dados suficientes mesmo que os APs estejam distantes, não garantindo conectividade o tempo todo. Neste caso, uma consequência direta seria o atraso na atualização das informações visto que os dados só seriam enviados quando houver conectividade. Além disso,

em cenários onde apenas partes da via possuam cobertura seria preciso prever uma estrutura mínima que possibilite que o usuário envie os dados para a central. Essa estrutura poderia, por exemplo, ser baseada no uso de protocolos de redes DTN.

A opção por utilizar *beacons* tanto na detecção da rede quanto para o funcionamento do sistema e não utilizar mecanismos ativos como o envio de quadros *Probe Request* foi, além de conter todas as informações necessárias e ser possível obter os dados sem a necessidade de estar associado ao AP, influenciar o mínimo possível no funcionamento normal da rede, aproveitando pacotes que já são frequentemente enviados pelos APs. O *beacon* é um *frame* de sincronismo enviado pelo AP para informar ao cliente que está ativo, além de avisar sobre quadros armazenados no *buffer* do AP aguardando transmissão e também sincronizar a transmissão dos dados. Apesar do *beacon* consumir tempo podendo reduzir a taxa de transmissão da rede, para o monitoramento do trânsito, quanto maior o número de *beacons* mais informação o cliente terá para enviar para a central. Outra característica importante é que um intervalo curto pode ajudar a melhorar a estabilidade em ambientes com muito ruído, que pode ser o caso neste cenário devido ao grande número de redes sem-fio ativas em determinadas regiões das cidades.

É fundamental que o usuário disponibilize as informações para a central de dados, pois o sistema funciona de forma colaborativa. Quem fornece todas as informações para a central são os clientes. Para incentivar o uso por parte do cliente, é possível disponibilizar, além das informações sobre o trânsito, informações sobre um conjunto de serviços no entorno da via, de acordo com a posição do veículo. O condutor pode receber informações no formato texto ou mesmo, em equipamentos com mais recursos, através de mapas, como nos serviços disponibilizados nos navegadores GPS comerciais.

O último requisito para o funcionamento do sistema proposto é o mapeamento da posição geográfica dos APs. Com os APs mapeados é possível saber, por exemplo, a direção do veículo na via bastando saber a qual AP o cliente estava conectado. Uma forma de fazer o mapeamento dos APs é por meio da técnica de *wardriving*, onde uma busca por redes sem fio é feita utilizando um automóvel e um equipamento com interface de rede sem fio. Desta forma, para definir a localização geográfica, considera-se que o AP está localizado onde for detectado o sinal com maior potência para aquele AP. Apesar de existirem vários algoritmos para fazer este mapeamento, como apresentado em [Cheng et al. 2005], o uso deste método é considerado suficiente já que todos os APs estarão, em algum momento, próximos ao equipamento que faz a varredura. Com os APs mapeados, o próximo passo é coletar as informações dos APs e enviar para uma central responsável pelo tratamento e divulgação dos dados de forma útil para o motorista.

3.2. Gerenciamento das Informações

Conhecendo a distância entre os APs, o mapeamento da via e a velocidade máxima permitida é possível calcular o tempo médio gasto para percorrer cada trecho. Por exemplo, considerando que o alcance dos APs é de 400m e que a área de cobertura de cada AP está dividida em três zonas (cerca de 133 metros cada zona), um veículo a 90km/h vai gastar cerca de 5 segundos para percorrer cada zona e 15 segundos pra trocar de AP. Um veículo a 60 km/h vai gastar cerca de 8 segundos para percorrer cada zona e 24 segundos para trocar de AP. Desta forma, é possível definir algumas faixas para saber quais as condições do trânsito (Tabela 2). No exemplo da Avenida Brasil, é possível dizer que o trânsito está

"muito bom"se o carro demorar até 24 segundos para mudar de um AP para o outro já que uma velocidade mínima de 60km/h é aceitável. É possível dizer também que o trânsito está muito lento se o tempo for acima de 1 minuto e 10 segundos.

Tabela 1. *Faixas para definição de Condição de Trânsito.*

Velocidade máxima	Tempo por AP	Tempo por zona	Condição do Trânsito
60 km/h a 90 km/h	até 24 segundos	até 8 segundos	MUITO BOM
30 km/h a 59 km/h	até 36 segundos	até 12 segundos	BOM
10 hm/h a 39 km/h	até 72 segundos	até 24 segundos	LENTO

A partir do momento que os dados estão no servidor é preciso tratá-los de acordo com a aplicação desejada. Além da detecção de movimento do fluxo de veículos é possível citar alguns exemplos de aplicações como: previsão de hora de chegada de determinado veículo, tempo médio por percurso (de acordo com a categoria do veículo), previsão de engarrafamento de acordo com horários e dias definidos (histórico), entre outros. É fundamental fazer esta separação de veículos em categorias para que seja possível distinguir veículos de emergência, veículos que usam a faixa seletiva, usuários em menor ritmo, como ciclistas, ou qualquer outro nó móvel que tenha um perfil diferenciado.

Na proposta atual, a central deve conter um registro dos endereços MACs dos APs pois a eles estão associadas as localizações de cada um. Desta forma, além de servirem para identificar a direção que o veículo segue, é possível saber em que parte da via o veículo está quando se sabe a qual AP o cliente está associado.

3.3. Tratamento das variações de Potência de Transmissão

Na maioria das interfaces de rede 802.11 encontradas em *laptops* e celulares é necessária uma potência de -90dBm para que um quadro seja corretamente recebido quando for transmitido a 1Mb/s (taxa na qual os *beacons* são enviados). Essa potência é inferior à necessária para receber o mesmo quadro se transmitido a 54Mb/s [Vilela et al. 2007]. Considerando a propagação do sinal na frequência 2,4GHz, as transmissões estão sujeitas a interferências e desvanecimento, sendo assim, o sinal recebido pode variar mesmo se o veículo estiver parado.

Desta forma, caso a potência fosse o único parâmetro considerado, o sistema poderia interpretar que o carro está em movimento. Para minimizar este problema são adotadas duas estratégias. Na primeira, o cliente possui um *buffer* local. Sendo assim, o cliente envia as informações de tempos em tempos para a central, enviando apenas a mediana entre os valores da potência. Ao enviar apenas o valor da mediana, acontece o primeiro filtro para minimizar a variação de potências. Além disso, caso o cliente receba algum *beacon* com potência de transmissão que esteja discrepante devido a alguma interferência ou ruído, este será descartado. A Tabela 2 ilustra um *buffer* de dois segundos com três valores de potência discrepantes. Com o valor da mediana obtida é possível concluir que o veículo está na zona A do AP AP_AvB_km1.

Outro mecanismo é verificar se, enquanto há alterações na potência, há reassociações para novos APs. Se a potência informada pelo veículo varia por um longo período de tempo, estando o veículo associado ao mesmo AP, significa que, ou o trânsito está ruim ou há alguma interferência no sinal.

Tabela 2. *Exemplo de buffer de 2 segundos.*

	AP_AvB_km1	
Buffer	Segundo 1	Segundo 2
1	-80 dBm	-90 dBm
2	-62 dBm	-59 dBm
3	-62 dBm	-80 dBm
4	-61 dBm	-59 dBm
5	-90 dBm	-59 dBm
6	-90 dBm	-58 dBm
7	-61 dBm	-56 dBm
8	-60 dBm	-55 dBm
9	-60 dBm	-54 dBm
10	-60 dBm	-52 dBm
Mediana: -60 dBm | Zona A		

O tamanho do *buffer* no cliente pode variar de acordo com as características da via. Para uma via com menor velocidade, por exemplo, pode ser mais interessante possuir um *buffer* maior já que usando um *buffer* pequeno serão enviados muitos valores relativos à mesma zona. Quanto maior o *buffer* menor o tráfego gerado na rede, no entanto, maior o tempo entre as atualizações de localização.

4. Experimentos

Com o objetivo de validar a proposta de monitoramento apresentada, foram criadas duas rotinas para mapear a posição geográfica dos APs de parte da Avenida Brasil e coletar *beacons*. Neste experimento, as informações foram armazenadas localmente para análise posterior. Foram percorridos 16 dos 58 quilômetros da Avenida Brasil, em destaque na Figura 5.

Na tentativa de mapear a posição dos APs utilizou-se a técnica de *wardriving*. Duas rotinas foram executas em paralelo: a primeira armazenava a hora, a data, a velocidade, a direção e as coordenadas geográficas obtidas via GPS e a segunda armazenava as informações da rede (MAC do AP, MAC do cliente, potência, data e hora de geração do pacote) obtidas via *beacon* durante o percurso. Para mapeamento utilizou-se um GPS modelo "u-blox 5", que informava a posição quatro vezes por segundo. O cliente foi executado em um *laptop* Intel Atom N450, com 2GB de RAM e 250GB de HD. Não foi utilizada antena externa.

Foi possível perceber que em determinados pontos da Avenida Brasil nenhum *beacon* é recebido da rede "Avenida Brasil Digital". Acredita-se que nestes locais há algum problema com os APs. Outra situação detectada durante os experimentos foi que por algumas vezes foram capturados *beacons* de apenas um mesmo AP (mesmo MAC) por cerca de 1,4 km. Isso representa receber dados do mesmo MAC por mais de dois minutos estando numa velocidade de 30 a 50 km/h, como mostra a Figura 6. Como esta situação esteve presente em mais de um momento do experimento, pretende-se investigar mais a respeito visto que influencia diretamente, por exemplo, no cálculo da velocidade do veículo na via, como mostrado na Seção 3.2. No entanto, a curva de sinal apresentou

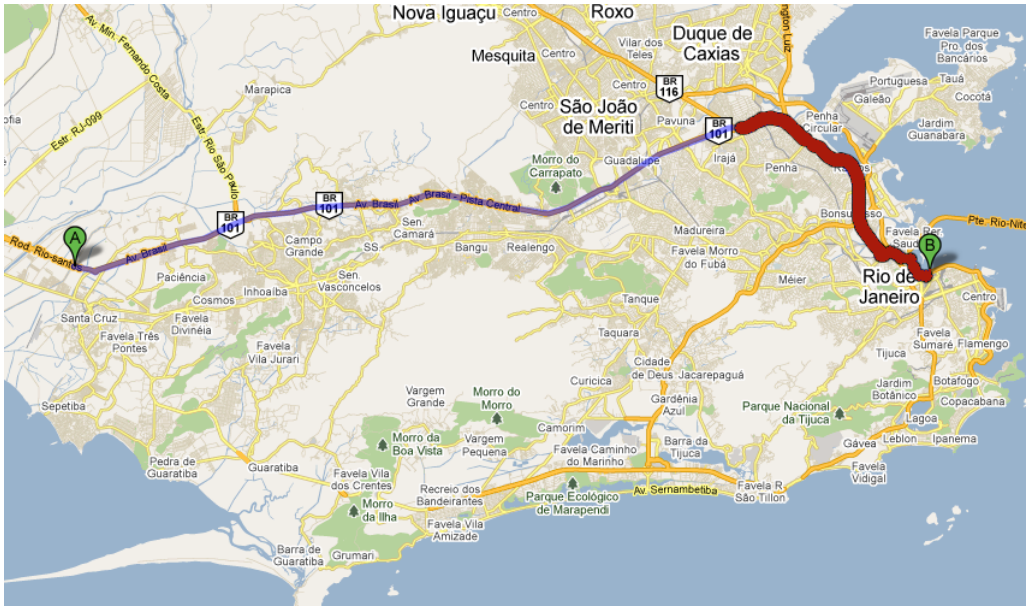
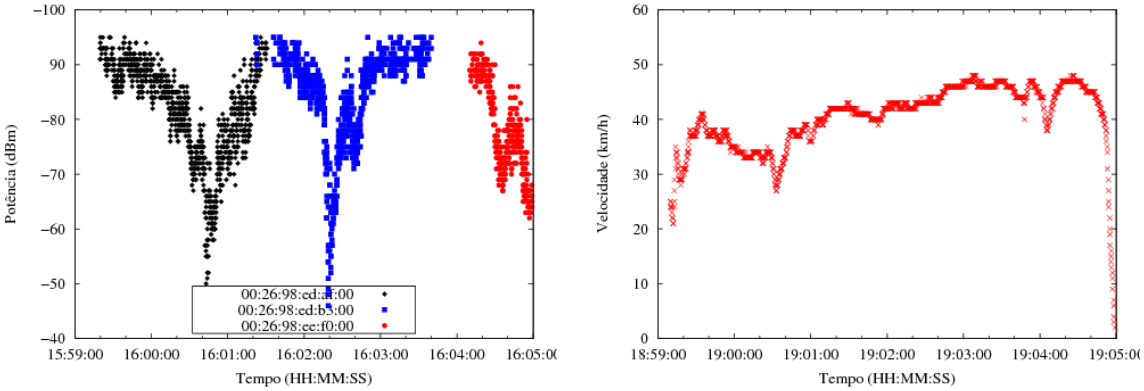


Figura 5. Cenário de teste - Avenida Brasil.

uma grande similaridade à curva esperada, ilustrada na Figura 3



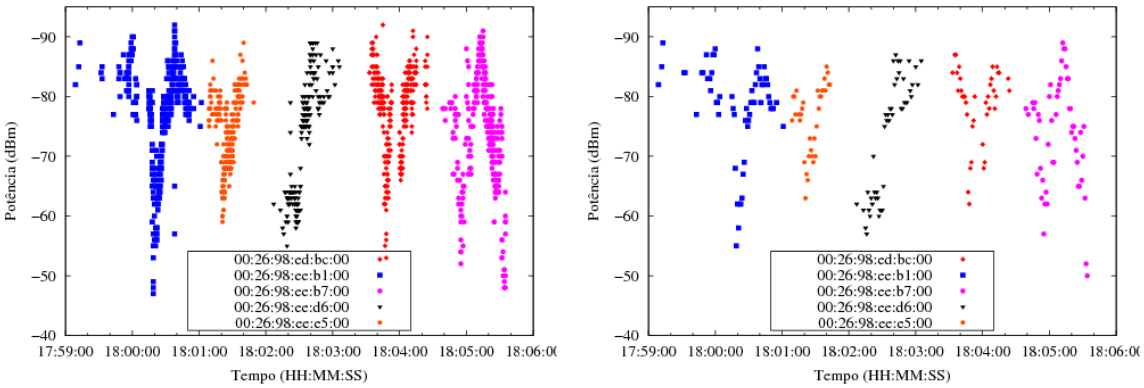
(a) Variação das potências de transmissão recebidas. (b) Velocidade média entre 30 e 50km/h.

Figura 6. *Captura de beacons na faixa de velocidade entre 30 e 50 km/h.*

A Figura 7 apresenta um comparativo da variação da potência de transmissão recebida pelo nó cliente entre o total de *beacons* e a mediana de cada segundo. Como pode ser visto, o uso da mediana retrata a variação da potência mesmo possuindo cerca de sete a oito vezes menos valores. Além das vantagens apresentadas na Seção 3.3, a opção por fazer o cálculo da mediana no cliente diminui o tráfego na rede e o processamento da central de gerenciamento de dados.

Durante os seis dias de coleta de dados, foram percorridos cerca de 61 quilômetros (dentro do trecho de 16 quilômetros destacados na Figura 5). Devido ao grande volume de tráfego a velocidade[3] variou entre 0 e 80km/h durante o percurso. A Figura 8(a) apresenta

[3] *A velocidade máxima atingida no marcador analógico do veículo foi de 90km/h. No entanto, a velocidade registrada durante o experimento foi a velocidade do GPS, cerca de 10km/h a menos na média.*



(a) *Beacons* coletados no período. (b) Mediana - um valor por segundo.

Figura 7. *Representação dos beacons recebidos utilizando apenas a mediana de cada segundo.*

os valores das potências de transmissão recebidas pelo nó cliente em um dos trechos, enquanto a Figura 8(b) apresenta a velocidade. É possível perceber que no minuto 14 o veículo enfrentou um engarrafamento. O resultado é que o nó cliente recebeu pacotes do mesmo AP (MAC 00:26:98:ef:da:00) por mais tempo. No trecho onde o veículo manteve uma velocidade média entre 40 e 50km/h o resultado foi o esperado, uma vez que foi possível detectar quando o veículo se aproximava e quando se distanciava de cada AP, baseado apenas na potência de transmissão recebida por meio do *beacon*. Observou-se também que potências superiores a -80dBm devem receber um peso menor na definição das zonas propostas na Seção 3, uma vez que é o trecho onde acontece a fase de reassociação entre APs. Este limite é representado por uma linha na faixa da potência -80 dBm na Figura 8(a).

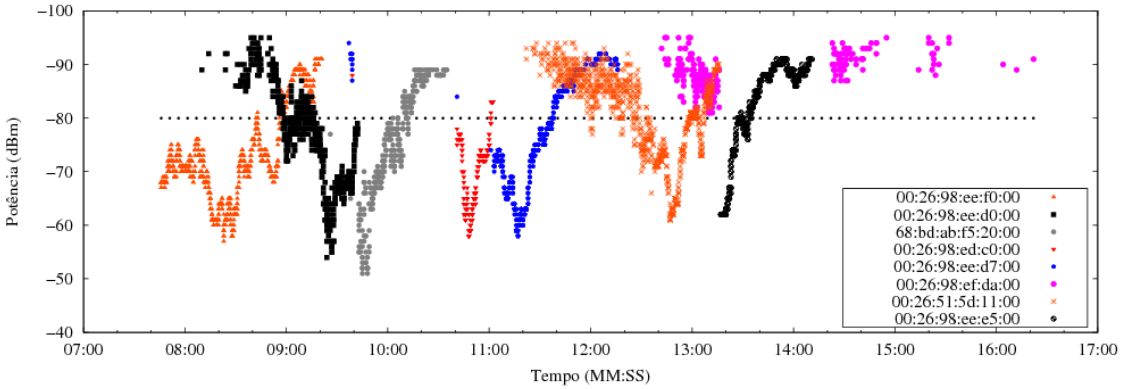
Com os resultados obtidos conclui-se que é possível detectar a movimentação de veículos utilizando apenas os *beacons* recebidos de uma rede sem-fio IEEE 802.11. É importante destacar que este é um cenário de uso cada vez mais frequente em cidades que possuem projetos que visam iluminar comunidades inteiras, disponibilizando o acesso a Internet por meio de tecnologia sem-fio.

5. Conclusão

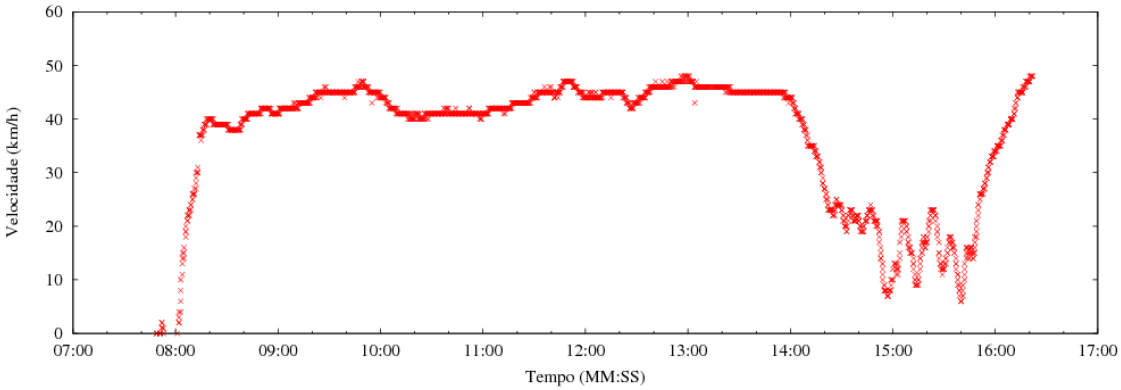
O artigo apresentou uma proposta de monitoramento colaborativo de trânsito utilizando redes IEEE 802.11 já existentes. Para utilização do modelo proposto, não é necessário nenhuma alteração na rede, desde que esta possua os seguintes requisitos: cobertura IEEE 802.11 em toda extensão da via ou em partes dela, disseminação de pacotes *beacons* pelos APs, clientes como *smartphone* ou *laptop* utilizando um equipamento com interface 802.11 e rodando a aplicação proposta e pontos de acesso mapeados.

O sistema proposto se baseia na potência informada no *beacon* recebido para calcular a movimentação dos veículos. Além da potência são utilizadas informações como o: ESSID do AP, endereço MAC (*Media Access Control*) do AP (BSSID) e hora de geração do pacote. Essas informações são enviadas para uma central responsável por consolidar os dados e oferecê-los para os usuários.

Experimentos realizados em um cenário real comprovaram que é possível detectar



(a) Variação das potências de transmissão recebidas.



(b) Velocidade durante o percurso.

Figura 8. *Captura de beacons.*

a movimentação de veículos utilizando apenas os *beacons* recebidos de uma rede sem-fio IEEE 802.11, sem a necessidade de interferir no funcionamento rede ou alterar a configuração dos roteadores. Concluiu-se também que o envio da mediana dos *beacons* recebidos a cada segundo retrata a variação da potência recebida. O uso da mediana diminui o tráfego na rede e o processamento da central de gerenciamento de dados uma vez que os APs estão configurados para enviar cerca de 8 *beacons* por segundo.

Como sugestão de trabalho futuro está a disponibilização de serviços como: previsão de horário de chegada do veículo, sinalização inteligente e prioridade para serviços de emergência. Pretende-se ainda investigar o funcionamento da proposta comparando diversos veículos em velocidades diferentes. Vislumbra-se também o desenvolvimento de um equipamento dedicado para coleta e envio de informações para a central. Este equipamento poderia ser instalado em veículos que passam diariamente pela via, como ônibus ou táxis.

Referências

Akkihebbal, A. L., Choon, C. M., Bin, C. B., Venkatagiri, P., Cheong, C. F., Fa, G. X., Zongyao, M., Full, L. C., Praveen, K. S., and Wong, J. (2011). DTN for Urban Environment. Disponível em

http://mobtorrent.ddns.comp.nus.edu.sg/wiki/index.php/Main_Page. Acessado em fevereiro de 2011.

Bychkovsky, V., Chen, K., Goraczko, M., Hu, H., Hull, B., Miu, A., Shih, E., Zhang, Y., Balakrishnan, H., and Madden, S. (2006). The CarTel Mobile Sensor Computing System. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys '06, pages 383–384, New York, NY, USA. ACM.

Bychkovsky, V., Chen, K., Goraczko, M., Hu, H., Hull, B., Miu, A., Shih, E., Zhang, Y., Balakrishnan, H., and Madden, S. (2011). CarTel. Disponível em http://cartel.csail.mit.edu/doku.php. Acessado em fevereiro de 2011.

Cheng, Y.-C., Chawathe, Y., LaMarca, A., and Krumm, J. (2005). Accuracy characterization for metropolitan-scale wi-fi localization. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, MobiSys'05, pages 233–245, New York, NY, USA. ACM.

Corner, M., Levine, B., and Brian, L. (2011). A Mobility Testbed - UMass DOME. Disponível em http://prisms.cs.umass.edu/dome/. Acessado em janeiro de 2011.

Edelmayer, A., Miranda, M., and Nebehaj, V. (2010). Cooperative federated filtering approach for enhanced position estimation and sensor fault tolerance in ad-hoc vehicle networks. *Intelligent Transport Systems, IET*, 4(1):82 –92.

IEEE (2003). IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11, 1999*, pages i –513.

INEA (2004). Qualidade do Ar. Instituto Estadual do Ambiente. Disponível em http://www.inea.rj.gov.br/fma/qualidade-ar.asp. Acessado em abril de 2011.

McMahon, A. and Farrell, S. (2009). Delay- and disruption-tolerant networking. *IEEE Internet Computing*, 13:82–87.

Ott, J. and Kutscher, D. (2005). A disconnection-tolerant transport for drive-thru internet environments. In *IN PROCEEDINGS OF IEEE INFOCOM*, pages 1849–1862.

Ott, J. and Kutscher, D. (2011). Drive-thru Internet. Disponível em http://www.drive-thru-internet.org. Acessado em janeiro de 2011.

UERJ (2010). UERJemdia. Disponível em http://www.uerj.br/publicacoes/uerj_emdia/535/. Boletim Semanal 24 a 30 de maio de 2010 Ano XIII - No 535 - Acessado em janeiro de 2011.

UFRJ (2008). Projeto Orla Digital. Disponível em http://www.orladigital.coppe.ufrj.br/. Acessado em abril de 2011.

Vilela, U. C., Cardoso, K. V., and de Rezende, J. F. (2007). Redes 802.11 em centros urbanos: Varredura, estatísticas e aplicações. *in VI Workshop em Desempenho de Sistemas Computacionais e de Comunicação - WPerformance'2007 (XXVII SBC). Rio de Janeiro*, pages 703–718.

Planejamento do Posicionamento de Leitores e Etiquetas de Referência em Sistemas de Localização RFID

Bruno Almeida da Silva e Paulo André da S. Gonçalves

Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
50.740-560 – Recife – PE – Brasil

{bas2,pasg}@cin.ufpe.br

Abstract. *Active RFID-based location systems are the most promising to provide automatic indoor location services of objects. Such systems rely on both received signal strength measurements and an infrastructure composed of readers and reference tags in order to report position estimates of target objects. This paper studies the impact of the positioning of this infrastructure on the location efficiency. For doing so, we consider 27 different infrastructure positioning and different signal propagation conditions in an indoor environment that is composed of 9 rooms or areas. This paper focuses on the following location systems: LANDMARC and LANDMARC+. Both are configured to estimate the room or area where the target object is placed. Simulation results show that the infrastructure positioning impacts significantly on the location efficiency. Based on this fact, we present some guidelines for proper infraestructure positioning while seeking to maximize the location efficiency.*

Resumo. *Os sistemas de localização RFID com etiquetas ativas são os mais promissores para o provimento de serviços de localização automática de objetos em ambientes internos. Esses sistemas contam com medidas de força de sinais e com uma infraestrutura de leitores e etiquetas de referência para a obtenção de estimativas de localização de objetos de interesse. Este artigo estuda a influência do posicionamento dessa infraestrutura na eficiência de localização. Para isso, considera-se 27 configurações de posicionamento da infraestrutura de localização e diversas condições de propagação de sinais em um ambiente interno com 9 salas ou áreas. O foco deste artigo está nos sistemas LANDMARC e LANDMARC+, onde ambos estão configurados para estimar a sala ou área na qual se encontra um objeto de interesse. Os resultados de simulação mostram que o posicionamento da infraestrutura dos sistemas estudados tem impacto significativo na eficiência de localização. A partir disso, são apresentadas orientações para um posicionamento mais adequado dessa infraestrutura, buscando a maximização da eficiência de localização.*

1. Introdução

Atualmente, os sistemas RFID (*Radio Frequency IDentification*) são os mais promissores para a identificação automática de objetos através de sinais de radiofrequência (RF). Os sistemas RFID mais básicos são compostos por um *leitor* e várias *etiquetas*. Cada etiqueta armazena um identificador (ID) único e é colada ou embutida em um objeto de interesse.

O processo de identificação é realizado pelo leitor, o qual requisita o ID das etiquetas que estão em seu alcance de comunicação. De acordo com um mapeamento prévio entre os IDs e os objetos, é possível descobrir automaticamente quais objetos estão no alcance de comunicação do leitor.

Assim sendo, os sistemas RFID possuem duas características importantes: a capacidade de identificar objetos e a capacidade de comunicação através de sinais de RF. Essas características em conjunto com a proliferação de sistemas RFID vêm estimulando o desenvolvimento de sistemas de localização RFID que utilizam medidas de força de sinal recebido (RSS - *Received Signal Strength*) para se obter automaticamente uma estimativa do posicionamento de objetos de interesse em ambientes internos [Ni et al. 2004], [Zhao et al. 2007], [Silva and Gonçalves 2009a], [Silva and Gonçalves 2009b], [Zhu et al. 2009], [Zhang et al. 2010], [Shi et al. 2010]. Para isso, esses sistemas contam com uma infraestrutura fixa de leitores e etiquetas de referência montada no ambiente desejado.

Diversos estudos vêm sendo realizados com foco na melhoria da eficiência de localização desses sistemas. Contudo, não foram encontrados estudos que demonstrem o impacto do posicionamento da infraestrutura de leitores e etiquetas de referência na eficiência de localização de aplicações que precisam informar a sala ou área na qual se encontra um objeto de interesse. Com base nisso, este artigo propõe um estudo do impacto do posicionamento da infraestrutura de localização na eficiência do sistemas LANDMARC [Ni et al. 2004] e LANDMARC+ [Silva and Gonçalves 2009a], [Silva and Gonçalves 2009b] com o tipo de aplicação descrito.

Em particular, este trabalho considera um ambiente interno dividido em áreas que podem estar separadas por paredes, divisórias ou marcações no chão. Os objetos estão espalhados pelo ambiente e um usuário deseja conhecer a área ou sala específica na qual um objeto de interesse se encontra fisicamente. O estudo apresentado neste artigo considera 27 configurações de posicionamento da infraestrutura de localização e diversas condições de propagação de sinais em um ambiente interno com 9 áreas ou salas. Os resultados de simulação mostram que o posicionamento da infraestrutura dos sistemas estudados tem impacto significativo na eficiência de localização. A partir disso, são apresentadas orientações para um posicionamento mais adequado dessa infraestrutura ao se buscar a maximização da eficiência de localização.

O restante deste artigo está organizado como segue: a Seção 2 detalha o funcionamento dos sistemas de localização LANDMARC e LANDMARC+. A Seção 3 detalha o *layout* de ambiente interno estudado e as 27 configurações de posicionamento da infraestrutura de localização nesse ambiente. A Seção 4 apresenta uma avaliação de desempenho dos sistemas estudados em diversos cenários. A Seção 5 apresenta as considerações finais deste trabalho.

2. Sistemas de Localização Estudados

2.1. LANDMARC

O LANDMARC (*LocAtioN iDentification based on dynaMic Active Rfid Calibration*) utiliza uma infraestrutura composta por leitores e etiquetas de referências para prover o serviço de localização de objetos, onde esses objetos são rotulados com uma etiqueta

RFID ativa. O processo de localização do LANDMARC consiste na construção de dois mapas de força de sinais ou RSS. O primeiro mapa de RSS construído é representado pelo vetor $\vec{S}(j) = (S_1^j, S_2^j, ..., S_n^j)$, onde S_i^j denota o RSS medido no leitor $i \in [1, n]$ com relação ao sinal transmitido pelo objeto j a ser localizado. Um segundo mapa é construído para cada etiqueta de referência $r \in [1, m]$ e é representado por $\vec{\theta} = (\theta_1^r, \theta_2^r, ..., \theta_n^r)$, onde θ_i^r é o RSS medido pelo leitor $i \in [1, n]$.

O LANDMARC define

$$E_r^j = \sqrt{\sum_{i=1}^{n}(\theta_i^r - S_i^j)^2} \quad (1)$$

como a distância Euclideana, em níveis de potência, entre o objeto a ser localizado e uma etiqueta de referência r. Conceitualmente, quanto menor for o valor de E_r^j, menor será a distância Euclideana entre a etiqueta de referência r e o objeto j a ser localizado.

O processo de obtenção da estimativa de localização de um objeto j com o LANDMARC ocorre em três etapas. Na primeira etapa, para cada etiqueta de referência $r \in [1, m]$ e o objeto j, ele calcula a distância Euclideana e armazena os valores no vetor $\vec{E}(j) = (E_1^j, E_2^j, ..., E_m^j)$. Na segunda etapa, o LANDMARC usa o vetor $\vec{E}(j)$ como entrada para o algoritmo KNN (*K-Nearest Neighbors*) [Bahl and Padmanabhan 2000] que permite encontrar as k etiquetas de referência com a menor distância para o objeto a ser localizado. Para $k = 1$, a estimativa de localização do objeto a ser localizado é igual ao posicionamento da etiqueta de referência mais próxima dele. Para $k > 1$, o LANDMARC estabelece pesos em função da distância entre o objeto a ser localizado e sua l-ésima etiqueta de referência vizinha mais próxima. O peso é dado por:

$$W_l^j = \frac{1/(E_l^j)^2}{\sum_{l=1}^{k}(1/(E_l^j)^2)} \quad . \quad (2)$$

Note que o maior peso será produzido pela l-ésima etiqueta de referência mais próxima do objeto a ser identificado e com menor valor E_l^j.

Na última etapa, o LANDMARC estima as coordenadas (x_j, y_j) do objeto j de acordo com a equação a seguir:

$$(x_j, y_j) = \sum_{l=1}^{j} W_l^j \times (x_l, y_l) \quad , \quad (3)$$

onde (x_l, y_l) são as coordenadas conhecidas da l-ésima etiqueta de referência.

Para estimar a sala ou área na qual se encontra um objeto, é necessário mapear a localização fornecida pelo LANDMARC no ambiente. Adicionalmente, é conhecido que o valor de k influencia no desempenho do LANDMARC e que o melhor valor para ele é 4 [Ni et al. 2004]. Assim sendo, em todas as simulações neste artigo é utilizado $k = 4$.

2.2. LANDMARC+

A abordagem LANDMARC+ visa evitar que usuários realizem buscas cegas, pelo ambiente, de objetos cujas localizações foram erroneamente estimadas. Para isso, o LAND-

MARC+ fornece automaticamente e simultaneamente duas estimativas de localização: uma primária proveniente da execução do LANDMARC e uma secundária. O cálculo da estimativa secundária leva em consideração a distância média Euclideana, em RSS, entre o objeto j a ser localizado e as etiquetas de referência em cada sala ou área a do ambiente. Essa distância é dada por:

$$M(j,a) \; = \; \frac{\sum_{t(a)=1}^{\rho} \Phi_{t(a)}^{j}}{\rho} \quad , \tag{4}$$

onde ρ representa o número de etiquetas de referência por sala ou área a e $\Phi_{t(a)}^{j}$ é a distância Euclideana entre o objeto j e uma etiqueta de referência $t(a)$ pertencente à área ou sala a.

Após executar o processo de localização conforme o LANDMARC, o LANDMARC+ constrói o vetor de distâncias médias Euclideanas $\vec{M}(j,A) = [M(j,a_1), M(j,a_2), ..., M(j,a_q)]$, onde q é o número de áreas ou salas presentes no ambiente interno A. A área a com menor valor $M(j,a)$ é a estimativa secundária de localização do objeto j a ser localizado.

3. Ambiente Interno e Configurações de Posicionamento Estudados

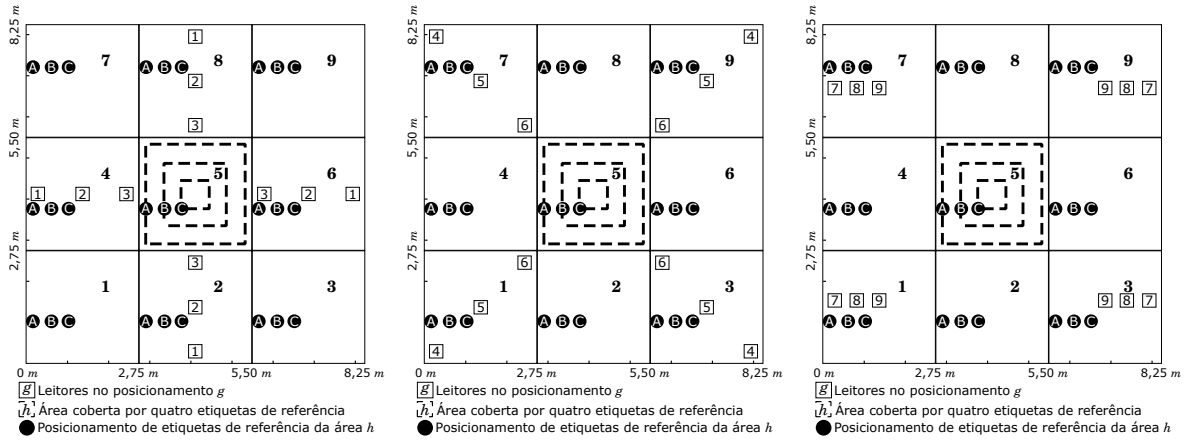
A Figura 1 apresenta o *layout* do ambiente e das diferentes configurações de posicionamento de leitores e etiquetas de referência. O ambiente é composto por 9 salas ou áreas quadradas (2,75 m x 2,75 m) e é coberto por uma infraestrutura com 4 leitores e 36 etiquetas de referência, havendo 4 etiquetas de referência por sala. As áreas que compõem o ambiente podem ser separadas fisicamente por paredes/divisórias ou por marcações no chão, formando ambientes fechados ou ambientes abertos, respectivamente.

Este trabalho estuda 9 configurações de posicionamento dos leitores. Cada configuração é identificada pelos pequenos quadrados numerados na Figura 1 ($1, 2, 3$ na Figura 1(a); $4, 5, 6$ na Figura 1(b); $7, 8, 9$ na Figura 1(c)). Em relação às etiquetas de referência, 3 configurações de posicionamento são estudadas. Elas são identificadas pelos pequenos círculos com letras A, B e C na Figura 1. O posicionamento das etiquetas de referência em cada uma das configurações é exemplificado pelos quadrados tracejados na área 5. Em todas as áreas, as etiquetas de referências estão posicionadas a 0,1 m, 0,6 m e 1,1 m da borda de cada área, respectivamente, para as configurações A, B e C.

4. Avaliações de Desempenho

Esta seção apresenta um estudo detalhado do desempenho do LANDMARC e do LANDMARC+ com as 27 combinações de configurações de posicionamento de leitores e de etiquetas de referência apresentadas na seção anterior. Neste artigo, a notação X_Y representa uma infraestrutura montada de acordo com a configuração X de etiquetas de referência e a configuração Y de posicionamento de leitores[1]. A seguir, serão apresentados os modelos de propagação de sinais utilizados nesse estudo, os parâmetros de simulação, as métricas de avaliação de desempenho e os resultados obtidos.

[1]As coordenadas de cada etiqueta de referência e de cada leitor nas diversas configurações estudadas podem ser obtidas em http://www.cin.ufpe.br/∼pasg/gpublications/bas2-rel2011a.pdf



(a) (b) (c)

Figura 1. *Layout* dos cenários utilizados considerando 9 conjuntos de posicionamento de leitores e 3 conjuntos de posicionamento de etiquetas de referência.

4.1. Modelos de Propagação de Sinais

Para avaliar, através de simulações, o desempenho de sistemas de localização baseados em medidas de RSS, é necessário primeiramente modelar a propagação de sinais. Neste artigo, dois modelos de propagação de sinais foram utilizados: o modelo *Path Loss* e o modelo para canais com desvanecimento (*fading*).

4.1.1. *Path Loss*

O modelo de propagação *Path Loss* (PL) é um modelo empírico para ambientes internos proposto em [Seidel and Rappaport 1992] e expressado por

$$PL = PL(d_0) + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + \sum_{q=1}^{Q} FAF(q) + \sum_{p=1}^{P} WAF(p) \quad , \tag{5}$$

onde PL é a atenuação sofrida (em dB) pelo sinal entre o transmissor e o receptor; $PL(d_0)$ é a atenuação sofrida pelo sinal a uma distância de referência d_0 do transmissor; d é a distância, em metros, entre o transmissor e o receptor; e γ é o expoente de atenuação e representa a taxa de aumento da atenuação em relação à distância. Os valores típicos do expoente de atenuação variam de 2,0 a 4,0. Os fatores de atenuação de pisos e paredes no ambiente são representados respectivamente por $FAF(q)$ e $WAF(p)$. As variáveis q e p representam, respectivamente, um piso (de Q existentes) e uma parede (de P existentes) entre o transmissor e o receptor.

Tipicamente, d_0 é igual a $1\ m$ em ambientes internos e a atenuação do sinal de um transmissor a uma distância de referência de $1\ m$ é dada por

$$PL\ (1metro)\ dB = 20 \log_{10}\left(\frac{4\pi}{\lambda}\right)\ dB \quad , \tag{6}$$

onde λ é o comprimento da onda eletromagnética.

Para um mesmo par transmissor-receptor estático, a Equação (5) permite obter apenas um valor fixo de RSS. Entretanto, em ambientes reais, as medidas de RSS variam ao longo do tempo para um mesmo par transmissor-receptor estático. Isso ocorre por causa de efeitos de *multipath*, *shadowing*, propagação sem linha de visada direta, e interferências de outros dispositivos RF. Na prática, tais efeitos fazem com que os valores de RSS pareçam randômicos e imprevisíveis ao longo do tempo. Assim sendo, é importante considerar variações nos valores de RSS de forma a tornar as simulacões mais realísticas. Por causa disso, este trabalho usa um modelo de propagação *Path Loss* estendido e definido por

$$PL^* = \Gamma \quad , \tag{7}$$

onde Γ denota uma variável randômica Gaussiana com variância σ^2 e média PL, sendo tal média a atenuação computada através da Equação (5).

4.1.2. Modelagem de Canal com Desvanecimento

Uma onda de rádio, ao se propagar, está sujeita a reflexões que provocam alterações em sua amplitude e em seu caminho percorrido. Isso leva a variações na potência do sinal recebido. Tais variações são chamadas de desvanecimento ou *fading*. O desvanecimento pode ser causado também por obstáculos na linha de visada direta entre o transmissor e o receptor. Esses obstáculos causam o fenômeno de propagação do sinal por múltiplos caminhos. As alterações na força de sinal causadas por desvanecimento são modeladas como processos randômicos. As duas distribuições mais utilizadas para a modelagem de canais com desvanecimento são as distribuições de Rayleigh e Rice.

A distribuição Rayleigh é voltada para situações onde o receptor obtém toda a energia do sinal por múltiplos caminhos. Por outro lado, quando toda a energia do sinal é obtida apenas por um caminho, ou seja, com visada direta, a distribuição que representa esse tipo de canal é a de Rice. A função densidade de probabilidade para a potência recebida (p) em um canal do tipo Rice é dada por [Rappaport 2001], [Sanchez-Garcia and Smith 2002]:

$$f_p(p \mid \bar{p}, K) = \frac{1+K}{\bar{p}} e^{-K} e^{\frac{p(1+K)}{\bar{p}}} I_0\left(\sqrt{\frac{4K(1+K)p}{\bar{p}}}\right) \quad , \tag{8}$$

onde $\bar{p}$ é a potência média recebida, I_0 é a função de Bessel modificada de ordem zero e tipo um e K é o fator de Rice. O fator K é definido por meio da relação entre a potência recebida pela visada direta (p_d) e a potência recebida por múltiplos caminhos ($\bar{p}_s$). Essa relação é dada por:

$$K = \frac{p_d}{\bar{p}_s} \quad . \tag{9}$$

Quanto maior for o fator K de um canal de comunicação, maior será a influência

da visada direta. Para $K = 0$, o canal é do tipo Rayleigh. Diante disso, observa-se que a distribuição Rayleigh é um caso especial da distribuição de Rice e a função densidade de probabilidade da potência recebida é obtida por

$$f_p(p \mid \bar{p}) = \frac{1}{\bar{p}} e^{\frac{p}{\bar{p}}} \quad . \tag{10}$$

4.2. Cenários de Estudo

Os cenários de simulações estudados neste artigo são descritos a seguir:

Cenário 1 - Neste cenário, todas as áreas do ambiente interno estão separadas por divisórias, formando salas. Assume-se que um sinal atravessando uma divisória é atenuado em 2,5 dB. O número total de divisórias pelas quais um sinal passa é obtido contando-se o número de divisórias na visada direta entre o transmissor e o receptor. Assim sendo, esse número depende do posicionamento do transmissor e do receptor. O modelo de propagação para canais com desvanecimento foi utilizado neste cenário. Os transmissores operam na frequência de 915 *MHz*.

Cenário 2 - Neste cenário, todas as áreas do ambiente interno estão separadas por marcações no chão. O modelo de propagação para canais com desvanecimento também foi utilizado neste cenário. A frequência de operação dos transmissores é também de 915 *MHz*.

Cenário 3 - Neste cenário, todas as áreas do ambiente interno estão separadas por divisórias, havendo a formação de salas. O canal de comunicação segue modelo *Path Loss* estendido. Nesse caso, o valor de γ depende do posicionamento do transmissor e do receptor. Se ambos estiverem na mesma sala, utiliza-se $\gamma =$ 2,4, caso contrário, utiliza-se $\gamma =$ 2,8. Os valores de γ estão de acordo com as diretrizes empíricas para escritórios convencionais simples com transmissores operando na frequência de 915 *MHz* [Sarkar et al. 2003]. Neste cenário, é assumido que um sinal atravessando uma divisória é atenuado em 2,5 dB ($WAF(p)$). O número total de divisórias (P) pelas quais um sinal passa é obtido em função do número de divisórias na visada direta entre o transmissor e o receptor. Assim sendo, esse número depende do posicionamento do transmissor e do receptor.

Cenário 4 - Neste cenário, todas as áreas do ambiente interno estão separadas por marcações no chão. O canal de comunicação segue o modelo *Path Loss* estendido. Nesse caso, um valor de γ igual a 2,4 foi adotado de forma a seguir as diretrizes empíricas para escritórios convencionais de plano aberto com transmissores operando na frequência de 915 *MHz* [Sarkar et al. 2003].

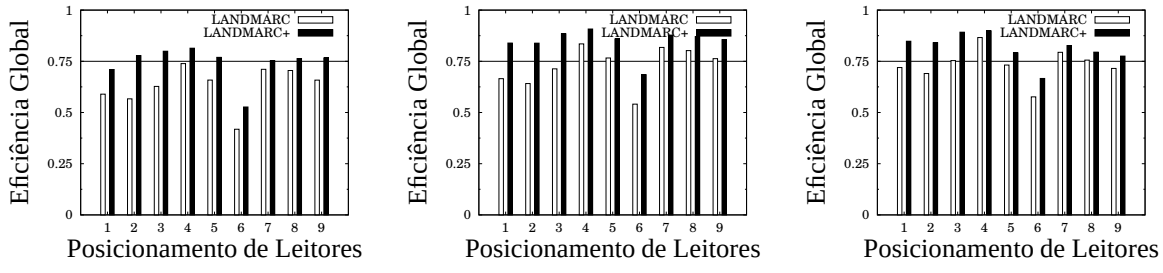
4.3. Desempenho com as Diversas Configurações de Posicionamento

Na execução de cada cenário descrito na Seção 4.2 combinou-se cada uma das configurações de posicionamento de leitores com cada uma das configurações de etiquetas de referência apresentadas na Seção 3. No interior de cada uma das salas do ambiente interno estudado, os objetos a serem localizados foram posicionados aleatoriamente. Para estimar a localização de cada objeto, foram utilizados os sistemas de localização LANDMARC e LANDMARC+. Nos cenários com desvanecimento foram estudados ambientes para cada fator K no intervalo $[0, 6]$ em passos de 0,5. Nos cenários que utilizam o modelo de propagação *Path Loss* estendido, foram estudados ambientes para cada σ^2 no intervalo

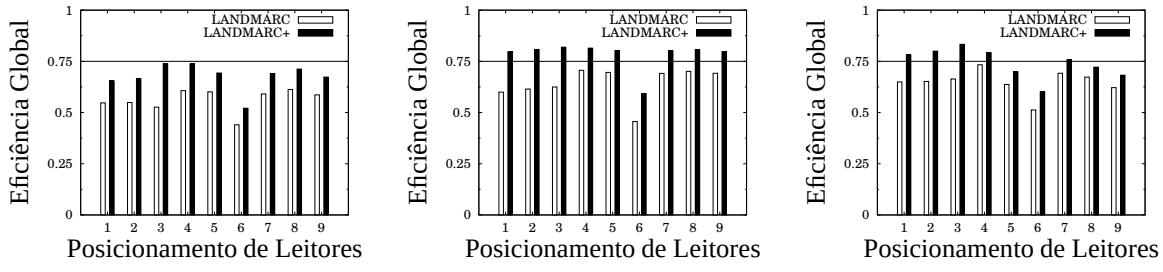
$[0, 20]$ em passos de 2. Para cada passo de variação de K e de σ^2, foram posicionados aleatoriamente 2.000 objetos a serem identificados em cada sala.

A métrica de avaliação de desempenho definida para esse estudado é a *eficiência global*. A eficiência global do LANDMARC é definida como a probabilidade dele informar corretamente, com sua única estimativa de localização, a área na qual se encontra o objeto a ser localizado. A eficiência global do LANDMARC+ é a probabilidade de ao menos uma de suas duas estimativas de localização representar a área real na qual se encontra o objeto a ser localizado. Todos os resultados apresentados nesta seção são médias obtidas a partir de todas as simulações feitas.

A Figura 2 apresenta os resultados dos **cenários 1** e **2** para cada uma das 27 combinações possíveis de configuração de posicionamento de leitores e etiquetas de referência. Observa-se que os piores desempenhos[2] de localização ocorrem sempre com o posicionamento 6 de leitores em ambos os cenários. Isso ocorre independentemente do ambiente ser aberto ou fechado e independentemente do posicionamento de etiquetas de referência estudado (A, B, C). Contudo, observa-se que nesse caso específico de posicionamento de leitores, o posicionamento C de etiquetas de referência provê os melhores resultados.



(a) Posicionamento **A** de etiquetas (b) Posicionamento **B** de etiquetas (c) Posicionamento **C** de etiquetas



(d) Posicionamento **A** de etiquetas (e) Posicionamento **B** de etiquetas (f) Posicionamento **C** de etiquetas

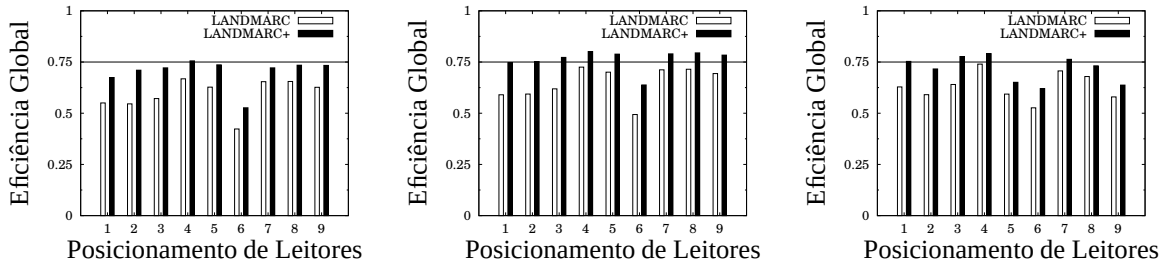
Figura 2. (a), (b) e (c): Cenário 1 (fechado/Desvanecimento). (d), (e), (f): Cenário 2 (aberto/Desvanecimento).

No **cenário 1** é observado que a melhor eficiência de localização para cada posicionamento de etiquetas de referência estudado (A, B, C) ocorre sempre para o posicionamento 4 de leitores. Em relação a tal posicionamento de leitores, o posicionamento C de etiquetas de referência provê o melhor resultado. No **cenário 2**, nota-se que o melhor desempenho de localização ocorre quando é utilizado o posicionamento C_4.

[2]Para se classificar um desempenho como melhor ou pior na análise das configurações, considera-se apenas o desempenho do LANDMARC, ou seja, a eficiência primária.

Pelo exposto, conclui-se que para os cenários avaliados, é melhor utilizar o posicionamento 4 de leitores associado ao posicionamento C de etiquetas de referência. Para o posicionamento em questão de leitores, apenas o posicionamento C de etiquetas de referência produz uma melhor eficiência de localização.

A Figura 3 apresenta os resultados dos **cenários 3** e **4** para cada uma das 27 configurações estudadas de posicionamento da infraestrutura de localização. Observa-se que em ambos os cenários, os piores desempenhos de localização também ocorrem sempre com o posicionamento 6 de leitores. Isso também ocorre independentemente do ambiente ser aberto ou fechado e independentemente do posicionamento de etiquetas de referência avaliado (A, B, C). Como no caso do ambiente ser fechado, observa-se que o posicionamento C de etiquetas de referência provê os melhores resultados quando o posicionamento 6 de leitores é utilizado.



(a) Posicionamento **A** de etiquetas (b) Posicionamento **B** de etiquetas (c) Posicionamento **C** de etiquetas

(d) Posicionamento **A** de etiquetas (e) Posicionamento **B** de etiquetas (f) Posicionamento **C** de etiquetas

Figura 3. (a), (b) e (c): Cenário 3 (fechado/*Path Loss* estendido). (d), (e) e (f): Cenário 4 (aberto/*Path Loss* estendido).

No **cenário 3**, nota-se que a melhor eficiência de localização para cada posicionamento de etiquetas de referência estudado (A, B, C) ocorre sempre para o posicionamento 4 de leitores. Em relação a tal posicionamento de leitores, o posicionamento C de etiquetas de referência também provê o melhor resultado. Contudo, a análise dos resultados do **cenário 4** mostra que o melhor posicionamento de leitores varia em função do posicionamento das etiquetas de referência. Para o posicionamento A de etiquetas de referência, o posicionamento 8 de leitores produz o melhor resultado. Já para o posicionamento B de etiquetas de referência, o melhor resultado é obtido com o posicionamento 5 de leitores. No caso do posicionamento C de etiquetas de referência, o posicionamento 4 de leitores produz o melhor resultado. Dentre os posicionamentos citados, o posicionamento B de etiquetas de referência associado ao posicionamento 5 de leitores produz a melhor eficiência ($59,82\%$ com o LANDMARC). Entretanto, note que a melhor eficiência para o posicionamento C de etiquetas de referência ocorre quando o posicionamento 4 de leitores é utilizado. Nesse caso, a eficiência é de $57,93\%$ com o LANDMARC.

A análise dos resultados obtidos para os **cenários 1**, **2** e **3** mostra que o posicionamento C_4 é o mais adequado por produzir os melhores resultados. Para o **cenário 4**, o posicionamento B_5 é o melhor, embora o posicionamento C_4 produza uma eficiência global apenas ligeiramente inferior. Pelo exposto, o posicionamento C_4 pode ser escolhido como o planejamento de posicionamento mais adequado já que produz quase sempre os melhores resultados sob diversas condições de propagação de sinais e independentemente do ambiente ser aberto ou fechado. Corrobora para tal escolha, o fato de que em um ambiente real, é importante utilizar um *layout* único de posicionamento de leitores e etiquetas de referência. Isso ocorre, pois a infraestrutura de localização precisa ser fixa e com posicionamento previamente mapeado. Além disso, deve-se considerar que, na prática, um canal de comunicação sem fio está sujeito a condições de propagação de sinais que variam ao longo do tempo. Tais condições são desconhecidas *a priori* e seria inviável readequar o posicionamento da infraestrutura de leitores e etiquetas de referência toda vez que as condições de propagação de sinais mudassem.

4.4. Eficiência por Sala com o Posicionamento C_4

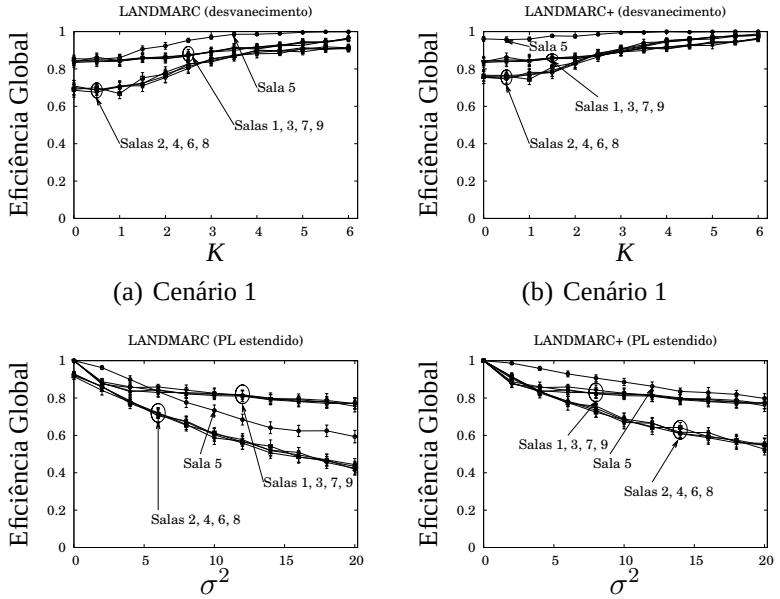
Esta seção apresenta um estudo da eficiência global dos sistemas de localização em cada uma das salas do ambiente interno e para cada um dos 4 cenários definidos na Seção 4.2. Nesse novo estudo, apenas o posicionamento C_4 é utilizado. Em todos os cenários, 2.000 objetos a serem localizados foram posicionados aleatoriamente dentro de cada uma das 9 salas do ambiente interno. O LANDMARC e o LANDMARC+ foram executados para localizar cada um dos objetos. Nos cenários com desvanecimento e para cada objeto a ser localizado, obteve-se a resposta dos sistemas de localização para cada fator K no intervalo $[0, 6]$ em passos de 0,5. Nos cenários que utilizam o modelo de propagação *Path Loss* estendido e para cada objeto a ser localizado, obteve-se a resposta dos sistemas de localização para cada σ^2 no intervalo $[0, 20]$ em passos de 2. Todos os resultados apresentados possuem intervalo de confiança de 99%. Tal intervalo é representado por barras de erro nos gráficos.

4.4.1. Ambientes Fechados e Abertos

A Figura 4 mostra a eficiência global por sala com o LANDMARC e o LANDMARC+ sob os **cenários 1** e **3** (ambientes fechados). A eficiência global por sala com o LANDMARC e o LANDMARC+ sob os **cenários 2** e **4** (ambientes abertos) é apresentada na Figura 5.

Ao se comparar os resultados obtidos com ambientes abertos e fechados, observa-se que uma maior eficiência de localização nos ambientes fechados. Isso é explicado pelo fato da presença de divisórias aumentar a probabilidade do KNN escolher como etiquetas de referência vizinhas mais próximas do objeto a ser localizado, as etiquetas que estão posicionadas na mesma sala onde se encontra tal objeto. Em todos os resultados, observa-se também um aumento da eficiência com o aumento de K e com a diminuição de σ^2. Isso é esperado, pois se reduz cada vez mais a variabilidade das medidas de RSS. Em particular, observa-se que a estratégia adotada pelo LANDMARC+ contribui de forma mais significativa para a melhoria da eficiência global em salas que não possuem leitores instalados.

Note que a eficiência de localização em algumas salas é próxima à eficiência de localização em outras. Assim, é possível identificar 3 conjuntos distintos de salas de acordo com os resultados apresentados: o conjunto formado pelas *salas 1, 3, 7* e *9*; o conjunto formado pelas *salas 2, 4, 6* e *8*; e o conjunto unitário formado pela *sala 5*. A formação desses conjuntos é explicada pela simetria das salas em relação à infraestrutura de localização que também está disposta de forma simétrica no ambiente.



(a) Cenário 1 (b) Cenário 1

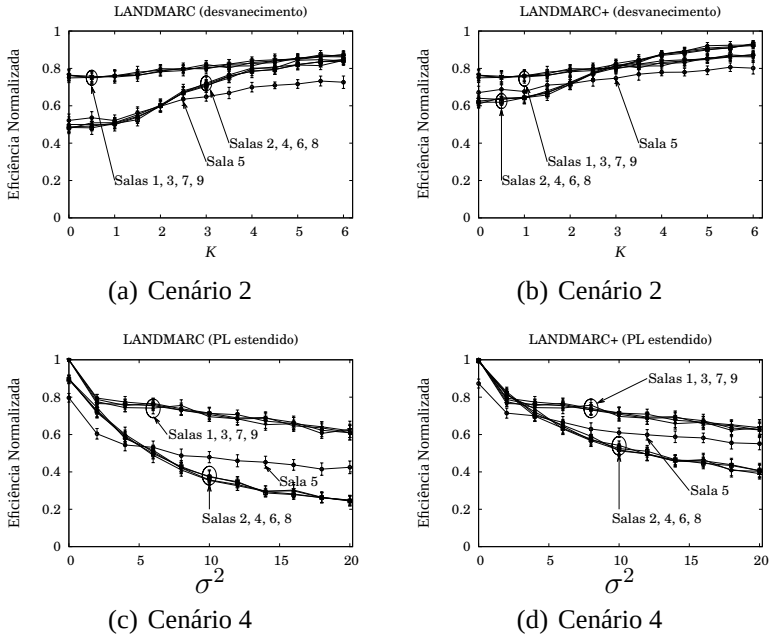
(c) Cenário 3 (d) Cenário 3

Figura 4. Eficiência Global com posicionamento aleatório de objetos em ambientes fechados.

4.4.2. Posicionamento Restrito de Objetos em Ambientes Abertos e Fechados

Em [Silva and Gonçalves 2009b], são apresentados estudos preliminares que sugerem que a eficiência global de localização do LANDMARC e do LANDMARC+ aumenta quando os objetos a serem localizados estão dentro da região interna formada pelas etiquetas de referência da sala ou área onde se situam. Essa sugestão é baseada na análise da eficiência de localização em apenas uma das 6 salas do ambiente interno estudado. Esta seção apresenta uma avaliação da eficiência de localização do LANDMARC e do LANDMARC+ seguindo tal sugestão, porém a avaliação é feita para cada uma das 9 salas do ambiente interno estudado neste artigo. As avaliações descritas na Seção 4.4.1 são repetidas aqui, alterando-se apenas a localização dos 2.000 objetos em cada sala. Para cada sala, os objetos estão agora posicionados aleatoriamente dentro da região interna formada pelas etiquetas de referências.

Ao se comparar os resultados apresentados nas Figuras 6 e 4, observa-se uma melhora significativa na eficiência global de localização. A mesma observação é válida ao se comparar os resultados apresentados nas Figuras 7 e 5. Por exemplo, houve melhorias de até 20% na eficiência global de localização nas *salas 1, 3, 7* e *9* ao se comparar o desempenho do LANDMARC nas Figuras 7(c) e 5(c) para $\sigma^2 = 2$. Logo, os

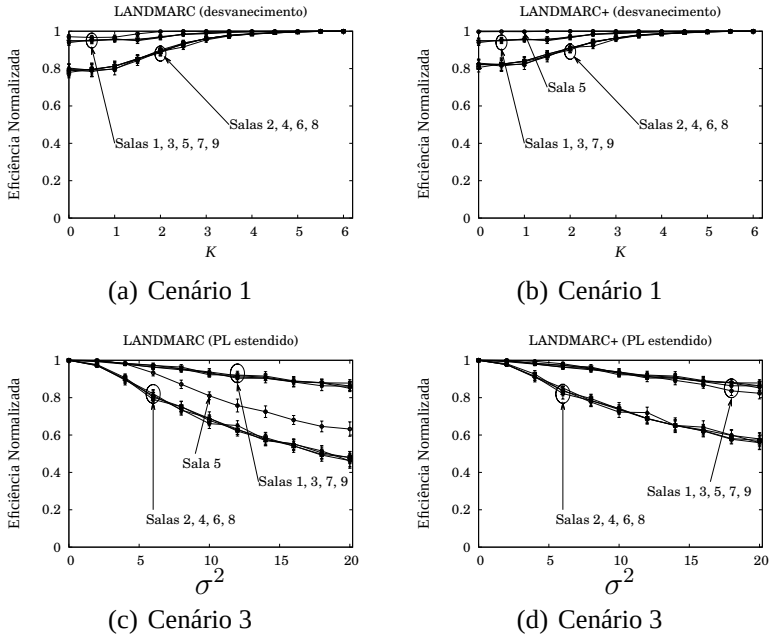


(a) Cenário 2 (b) Cenário 2

(c) Cenário 4 (d) Cenário 4

Figura 5. Eficiência Global com posicionamento aleatório de objetos em ambientes abertos.

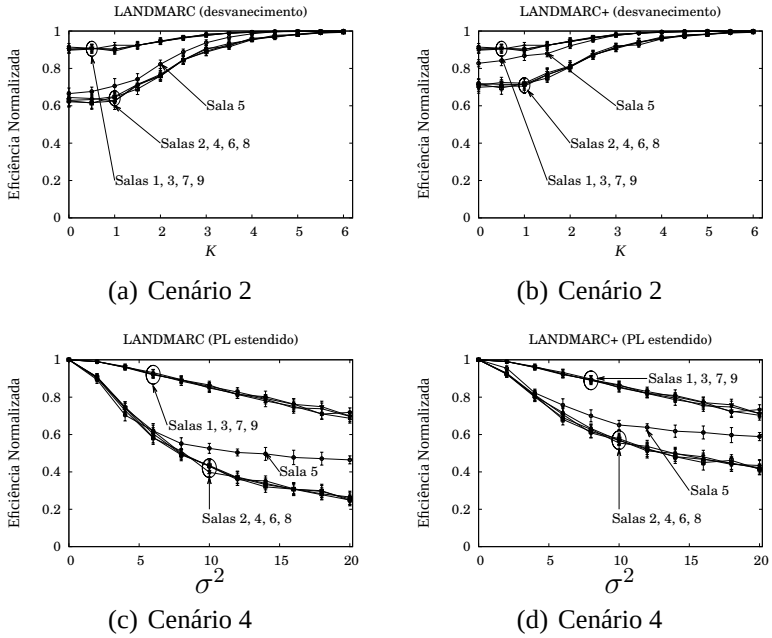
resultados apresentados confirmam a sugestão de posicionamento de objetos apresentada em [Silva and Gonçalves 2009b].



(a) Cenário 1 (b) Cenário 1

(c) Cenário 3 (d) Cenário 3

Figura 6. Eficiência Global com posicionamento restrito dos objetos em ambientes fechados.



(a) Cenário 2 (b) Cenário 2

(c) Cenário 4 (d) Cenário 4

Figura 7. Eficiência Global com posicionamento restrito de objetos em ambientes abertos.

5. Considerações Finais

Esse artigo demonstrou a importância de se planejar o posicionamento da infraestrutura de leitores e etiquetas de referência dos sistemas de localização LANDMARC e LANDMARC+. Dependendo de como essa infraestrutura é posicionada, a eficiência de localização dos objetos etiquetados com RFID pode ser significativamente afetada para aplicações que precisam informar a sala ou área onde um objeto está fisicamente posicionado.

Através do estudo de 27 configurações de posicionamento dessa infraestrutura, foi demonstrado que a configuração de posicionamento C_4 se mostrou a mais adequada sob diversas condições de propagação de sinais no ambiente interno estudado. Isso indica que em ambientes semelhantes ao estudado, é preferível posicionar as etiquetas de referência mais próximas do centro das salas ou áreas do ambiente do que mais próximas das fronteiras dessas salas ou áreas. Por outro lado, os resultados também indicam que o melhor posicionamento dos leitores no ambiente é aquele onde eles estão próximos ao limite do ambiente, simetricamente nos cantos, mas respeitando a condição de haver comunicação com todas as etiquetas.

Referências

Bahl, P. and Padmanabhan, V. (2000). RADAR: An In-building RF-based User Location and Tracking System. In *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, volume 2, pages 775–784.

Ni, L., Liu, Y., Lau, Y., and Patil, A. (2004). LANDMARC: Indoor Location Sensing using Active RFID. *Wireless Networks*, 10(6):701–710.

Rappaport, T. S. (2001). *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2nd edition.

Sanchez-Garcia, J. and Smith, D. R. (2002). Capture Probability in Ricean Fading Channels with Power Control in the Transmitters. *IEEE Transactions on Communications*, 50(12).

Sarkar, T., Ji, Z., Kim, K., Medouri, A., and Salazar-Palma, M. (2003). A Survey of Various Propagation Models for Mobile Communication. *IEEE Antennas and Propagation Magazine*, 45(3):51–82.

Seidel, S. and Rappaport, T. (1992). 914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings. *IEEE Transactions on Antennas and Propagation*, 40(2):207–217.

Shi, W., Liu, K., Ju, Y., and Yan, G. (2010). An Efficient Indoor Location Algorithm based on RFID Technology. In *Proc. of International Conference on Wireless Communications, Networking, and Mobile Computing (WiCOM)*, pages 1–5.

Silva, R. A. and Gonçalves, P. A. S. (2009a). Enhancing the Efficiency of Active RFID-based Indoor Location Systems. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6.

Silva, R. A. and Gonçalves, P. A. S. (2009b). Um Novo Algoritmo de Auxílio à Localização de Etiquetas RFID Ativas em Ambientes Internos. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 539–552.

Zhang, X., Peng, J., and Cao, X. (2010). RFID Indoor Localization Algorithm Based on Dynamic Netting. In *Proc. of International Conference on Computational and Information Sciences (ICCIS)*, pages 428–431.

Zhao, Y., Liu, Y., and Ni, L. (2007). VIRE: Active RFID-based Localization Using Virtual Reference Elimination. In *Proc. of IEEE International Conference on Parallel Processing*, pages 5–12.

Zhu, F.-J., Wei, Z.-H., Hu, B.-J., Chen, J.-G., and Guo, Z.-M. (2009). Analysis of Indoor Positioning Approaches Based on Active RFID. In *Proc. of International Conference on Wireless Communications, Networking, and Mobile Computing (WiCOM)*, pages 1–4.

Uma Linguagem para Especificação de SLA para a Negociação de Redes Virtualizadas na Internet do Futuro

Rafael Lopes Gomes [1] **Edmundo Madeira** [1]

[1]Instituto de Computação - IC
Universidade Estadual de Campinas - UNICAMP
Campinas, SP, Brasil

rafaellgom@lrc.ic.unicamp.br, edmundo@ic.unicamp.br

Abstract. *Over the years the Internet became the primary means of communication, where many companies use it as basis for its services. In most cases these companies use applications with different requirements. However, the current Internet does not guarantee Quality of Service (QoS), so emerging the concept of network virtualization as the basis for the Future Internet. A common strategy used by companies is to define a Service Level Agreement (SLA) with their respective Internet Service Providers (ISP). Within this context, this paper proposes a language specification of SLA for the Future Internet which is based on classes, allowing the negotiation of the traditional aspects of QoS, and also the network protocols to be used in the defined classes. So, now the companies can deploy different parameters with the ISPs for the desired classes.*

Resumo. *Ao longo dos anos a Internet se tornou o principal meio de comunicação, onde muitas empresas usam a Internet como base para os seus serviços. Onde na maioria dos casos, estas empresas fazem uso de aplicações com diferentes requisitos. Entretanto, a Internet atual não garante Qualidade de Serviço (Quality of Service – QoS), surgindo assim o conceito de virtualização de redes como base para a Internet do Futuro. Uma estratégia usada pelas empresas é definir um Acordo de Nível de Serviços (Service Level Agreement – SLA) com os seus respectivos provedores de Internet (Internet Service Providers - ISP). Dentro deste contexto, este trabalho propõe uma linguagem de especificação de SLA para a Internet do Futuro baseada em classes, permitindo a negociação não somente dos recursos tradicionais de QoS, mas também dos protocolos de redes a serem utilizados. Assim, as empresas poderão definir parâmetros diferentes com os ISPs para cada uma das classes desejadas.*

1. Introdução

A Internet tem crescido e o seu uso está cada vez mais diversificado. A Internet foi projetada dando ênfase à generalidade e heterogeneidade na camada de rede. Além disso, é baseada na premissa de ser descentralizada e dividida em múltiplas regiões administrativas autônomas, os chamados ASs (*Autonomous Systems*).

Vários ASs fornecem serviços de acesso à Internet aos usuários, chamados de provedores de Internet (*Internet Service Providers* - ISP). Os ISP fornecem serviços através de Acordos de Nível de Serviços (*Service Level Agreement* - SLA), que são usados como uma base contratual para definir propriedades não funcionais.

Atualmente, a Internet como um todo funciona sobre um aspecto de melhor esforço (*Best Effort* - BE), ou seja, não há garantias de nível de serviço. Devido a isso, as redes de melhor esforço são inadequadas para os serviços de nova geração (*Next Generation Services* - NGS), os quais necessitam de altos níveis de QoS.

Com a estrutura atual da Internet, os ISPs não conseguem, em muitos casos, atender as demandas de recursos exigidas pelas novas aplicações [Papadimitriou et al. 2009]. Devido às dificuldades encontradas recentemente, existe um consenso de que a Internet atual precisa ser reformulada, criando assim a chamada "Internet do Futuro".

Junto a esse cenário, surge a Virtualização de Redes (*Network Virtualization* – NV). NV é a tecnologia que permite a operação simultânea de múltiplas redes lógicas, onde além de redes virtuais pode-se ter roteadores e enlaces virtuais, e consequentemente pilhas de protocolos específicas. Espera-se que seja uma das mais importantes tecnologias para a Internet do Futuro [Papadimitriou et al. 2009].

Com o surgimento das redes virtualizadas, a figura do ISP é dividida em dois novos papéis: o *Infrastructure Provider* (InP) e o *Virtual Network Provider* (VNP). Sendo o usuário da rede chamado de *Virtual Network User* (VNU) [Fajjari et al. 2010].

O InP é o dono e gerencia os recursos físicos, oferecendo os recursos ao VNP, que é o seu cliente direto. Sendo assim, o InP não oferece serviços aos VNU. O VNP é o responsável pela criação e implantação das redes virtuais, alugando recursos de um ou mais InPs para oferecer um serviço fim-a-fim ao VNU. Portanto, o VNP é o responsável por implantar os protocolos, serviços e aplicações da rede virtual, atendendo assim os requisitos contratados pelos VNUs. Onde esses requisitos são especificados em um SLA entre as partes envolvidas (cliente e provedor, VNU e VNP).

Com a flexibilização decorrente da virtualização de redes, os VNUs e os VNPs podem definir diversas redes virtualizadas, com as mais variadas características. Portanto, um VNU pode definir classes que serão atendidas por diferentes redes virtualizadas. Onde essas redes podem ser moldadas para atender os requisitos específicos das classes definidas. Um exemplo pode ser visualizado na Figura 1.

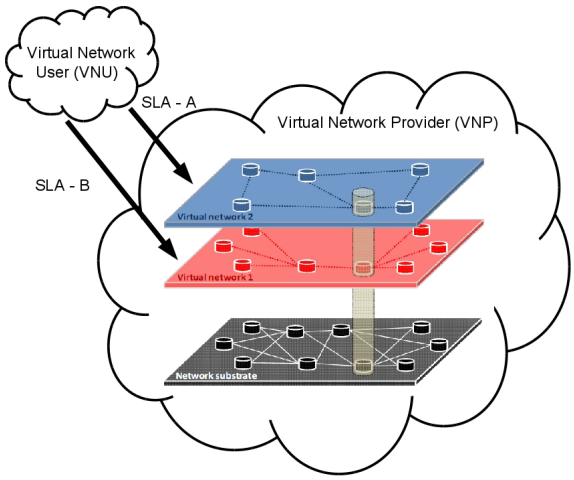


Figura 1. Exemplo de SLA baseado em Classes

Dentro deste contexto, este trabalho propõe uma linguagem de especificação de

SLA para a Internet do Futuro baseada em classes, permitindo a negociação não somente dos recursos tradicionais de QoS, mas também os protocolos de rede a serem utilizados. Neste caso, as classes representam tipos de tráfegos distintos, que consequentemente têm requisitos diferentes.

O objetivo é permitir a negociação completa da rede entre o VNU e o VNP, podendo-se personalizar a pilha de protocolo utilizada para cada classe definida. Assim, cada rede virtual negociada atende uma das classes definidas, onde as mesmas possuem características próprias, como os parâmetros de QoS (atraso, *jitter*, perda e outros), pilha de protocolo, obrigações, tempo de duração do contrato e preço a ser pago.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta alguns dos trabalhos relacionados ao tema abordado, a Seção 3 descreve a linguagem de especificação de redes virtualizadas proposta, a Seção 4 mostra um estudo de caso utilizando a linguagem proposta e a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Esta seção apresenta os trabalhos encontrados na literatura que mais se relacionam com a proposta deste trabalho, apresentando os principais aspectos abordados relacionados à definição de linguagens de especificação de SLAs.

O Projeto AQUILA [Koch and Hussmann 2003] define modelos de SLS (*Service Level Specification*) afim de padronizar as requisições de QoS entre os clientes e os provedores de serviço, para assim prover suporte à QoS em redes IP. A idéia era definir modelos de SLS para simplificar o processo de tradução de definições de SLS para as configurações dos dispositivos. O objetivo de padronizar os modelos de SLS é evitar erros e excesso de complexidade nas requisições de QoS. Em geral, o foco do projeto AQUILA era definir modelos para negociação entre clientes e provedores de serviços a partir de conceitos de SLS e SLA.

O Projeto TEQUILA [Trimintzios et al. 2001] é focado no contexto intra domínio onde os serviços IP oferecidos são implantados em toda a Internet. Esse projeto apresenta uma especificação DiffServ em um modelo de camadas e aborda tópicos como SLA e SLS, definindo um modelo de SLS. De modo geral, o projeto aborda uma modelagem de redes IP com DiffServ para o aprovisionamento e controle de admissão na Internet. O modelo de SLS definido pelo projeto TEQUILA foi um dos pontos iniciais para se usar um SLS expresso com parâmetros relacionados às redes de computadores.

WSLA (*Web Service Level Agreement Language*) [Keller and Ludwig 2003] é uma linguagem para definição de SLAs baseada em Web Services e XML, onde cria-se um *XML Schema* que engloba a definição das partes envolvidas, as garantias de serviços e a descrição do serviço. WSLA tem os seguintes componentes principais: *Parties*, *Service Definiton* e *Obligations*. *Parties* descreve as partes envolvidas no serviço (cliente ou provedor). *Service Definition* descreve os serviços ligados ao SLA, representando o entendimento de ambas as partes sobre os parâmetros do serviço descrito. Finalmente, *Obligations* define o nível de serviço que deve ser garantido com relação aos parâmetros definidos no *Service Definition*. WSLA pode ser aplicado para gerenciamento inter domínio em cenários orientados a negócios, visto que é baseado em descrições de serviços como WSDL, SOAP e UDDI, que facilitam a descrição e indexação dos serviços.

Lamanna et al [Lamanna et al. 2003] propõem SLAng, uma linguagem baseada em XML que é implantada sobre a WSDL (*Web Service Definition Language*) e um servidor de aplicações. Inicialmente dividi-se o SLA em duas categorias principais: Horizontal e Vertical. No SLA Horizontal, o contrato é feito entre duas entidades com o mesmo nível de arquitetura, ou seja, que atuam com funções semelhantes. No SLA Vertical, o contrato ocorre entre as entidades de camadas diferentes, ou seja, com funções distintas. SLAng introduz o conceito de responsabilidades, tanto do cliente quanto do provedor, além da definição de penalidades, descrevendo assim as obrigações de cada parte.

Tebbani et al [Tebbani and Aib 2006] propõem GXLA, que é a implementação de uma linguagem genérica para especificação de SLA. GXLA é definida como um *XML Schema* orientado a papéis com suporte a múltiplas partes envolvidas no contrato. O objetivo é modelar uma especificação formal, onde cada papel especificado inclui um conjunto de regras que caracteriza o comportamnento do SLA como um todo. Assim, tentando automatizar o gerenciamento em arquiteturas orientadas a serviços.

Fajjari et al [Fajjari et al. 2010] definem uma especificação de recursos em redes virtualizadas, chamado de VN-SLA. A especificação define os recursos virtuais oferecidos pelo provedor de infra-estrutura e o acordo cumprido entre as partes do contrato. O VN-SLA foca principalmente na especificação dos recursos de infra-estrutura, como nós, interfaces de rede, topologia, etc. Sendo pouco específico com relação à pilha de protocolo utilizada, tratando esta a partir de restrições estáticas, impossibilitando a negociação e personalização da pilha de protocolo a ser implantada na rede virtual negociada.

Nenhum dos trabalhos mostrados aborda o objetivo deste trabalho: desenvolver uma linguagem de SLA para a negociação completa de redes virtualizadas baseada em classes. Onde a negociação em questão, além dos tradicionais parâmetros de QoS (atraso, perda, etc), leva em consideração a negociação da pilha de protocolo utilizada na rede.

3. Linguagem de Especificação Desenvolvida

Atualmente, as empresas visam cada vez mais aumentar suas opções de contratação de serviços. Dentre os serviços existentes, o acesso à Internet é um deles. Devido a isso, as empresas cada vez mais adotam uma política de se ter mais de um provedor de Internet (ISP), onde para cada um dos mesmos se implanta um SLA.

Nem sempre os ISPs possuem a mesma qualidade em sua infraestrutura, e consequentemente o mesmo custo. ISPs que oferecem uma infraestrutura mais qualificada cobram mais por seus serviços e garantias dos mesmos. Mas nem sempre todas as aplicações necessitam de uma infraestrutura tão rebuscada assim, o que acaba gerando custos desnecessários às empresas.

Da mesma forma, existem aplicações que tem uma necessidade de parâmetros de QoS para um bom desempenho. Uma estratégia comum para se garantir esses parâmetros de QoS é a criação de um SLA entre as empresas e seus ISPs delimitando os requisitos desejados.

Com a eminente utilização da virtualização de redes para se tornar base da Internet do Futuro [Chowdhury and Boutaba 2009], surge a possibilidade de se adequar a infraestrutura de rede para as diversas necessidades dos clientes. Assim, pode-se definir classes que representam os mais distintos requisitos das aplicações a serem utilizadas.

Dentro deste contexto, propõe-se uma linguagem de especificação de SLA para a Internet do Futuro baseada em classes, permitindo a negociação não somente dos recursos tradicionais de QoS, mas também os protocolos de rede a serem utilizados. Onde as classes definidas representam tipos de tráfego com diferentes requisitos.

A linguagem de especificação desenvolvida para descrever o SLA entre as partes foi baseada na linguagem XML (*Extensible Markup Language*). A linguagem XML tem força expressiva para descrever as especificações de serviços e definições em geral.

De fato, a linguagem XML tem várias características que a transformam em uma boa escolha para a definição de contratos SLA. A linguagem XML é extensível, caso novos requisitos sejam identificados, os mesmos podem ser facilmente adaptados. A linguagem XML usa arquivos de texto e é baseada em *tags*, portanto, pode ser processada em qualquer plataforma e ser transportada sobre qualquer tipo de rede [Sun et al. 2005].

Antes de ser analisado o arquivo XML, este necessita ser validado para garantir sua integridade. Embora existam diversos esquemas de validação de arquivos XML, adotou-se o *XML Schema*, pois é uma linguagem de especificação amplamente utilizada, permitindo a descrição da estrutura do documento, elementos e tipos. De forma geral, o *XML Schema* pode ser usado para descrever a estrutura de um documento XML e definir a semântica dos elementos [Sun et al. 2005].

Os SLAs devem possuir necessariamente alguns elementos em sua descrição: as partes envolvidas, parâmetros do SLA, as métricas a serem avaliadas para descrever os serviços, as obrigações de cada parte e o custo dos serviços [Sun et al. 2005].

Além dos elementos tradicionais dos contratos SLA, este trabalho define os seguintes componentes adicionais para o contexto de redes virtualizadas: a descrição das classes definidas e a pilha de protocolo de rede desejada para certa classe.

A seguir a linguagem de especificação de SLA desenvolvida será detalhada, onde serão descritos os seus componentes e as determinadas funções. Uma visão geral da linguagem de especificação pode ser visualizada na Figura 2.

3.1. Componentes *SLA*, *Parties* e *Actor*

O componente *SLA* é o componente raiz, contendo os componentes *Traffic_Class* e *Parties*, além do identificador (*ID*) do contrato. Podem ser declarados diversos componentes *Traffic_Class*, permitindo a negociação de quantas classes forem necessárias. A seguir é mostrada a parte do arquivo *XML Schema* que representa o componente SLA:

```
1 <complexType name="SLA">
2   <sequence>
3     <element name="ID" type="xsd:int" minOccurs="1" maxOccurs="1"/>
4     <element name="parties" type="Parties" minOccurs="1" maxOccurs="1" nillable="false"
      />
5     <element name="classes" type="Traffic-Class" minOccurs="1" maxOccurs="unbounded"/>
6   </sequence>
7 </complexType>
```

O componente *Parties* define as partes envolvidas no contrato e os seus determinados papéis (VNP e VNU). Uma instância do componente *Parties* se relaciona com duas instâncias do componente *Actor*, que define as características de cada uma das partes envolvidas, além de possibilitar a execução de mecanismo de monitoramento e segurança. O atributo *URI_Digital_Certification* representa o endereço para se analisar o certificado

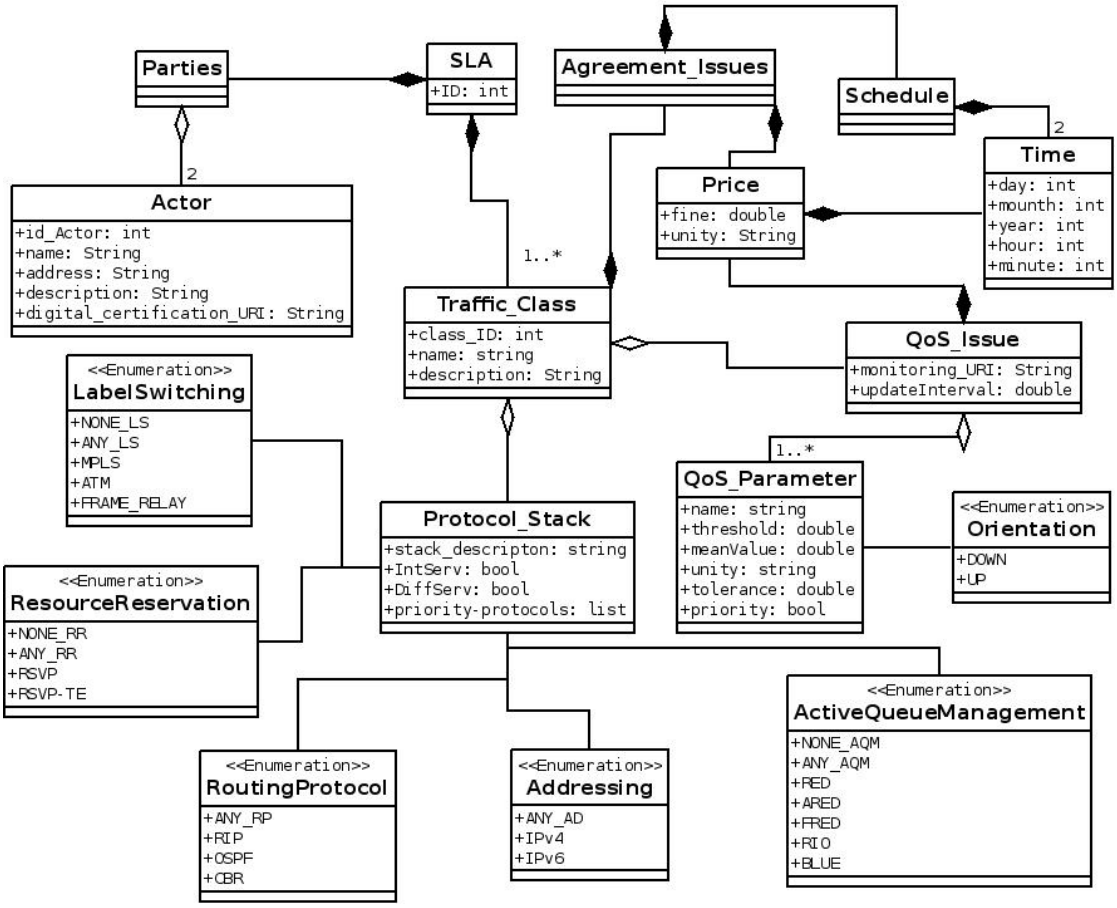


Figura 2. Diagrama que representa a linguagem de especificação desenvolvida

digital do ator, visando garantir a integridade do mesmo. A seguir é mostrada a parte do arquivo *XML Schema* que representa os componentes *Parties* e *Actor*:

```
1  <complexType name="Actor">
2     <sequence>
3        <element name="ID-Actor" type="xsd:int" minOccurs="1" maxOccurs="1"/>
4        <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
5        <element name="address" type="xsd:string" minOccurs="1" maxOccurs="1"/>
6        <element name="description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
7        <element name="URI-Digital-Certification" type="xsd:string" minOccurs="1" maxOccurs
         ="1"/>
8     </sequence>
9  </complexType>
10 <complexType name="Parties">
11    <sequence>
12       <element name="VN-User" type="Actor" minOccurs="1" maxOccurs="1" nillable="false"/>
13       <element name="VN-Provider" type="Actor" minOccurs="1" maxOccurs="1" nillable="
         false"/>
14    </sequence>
15 </complexType>
```

3.2. Componentes *Traffic_Class* e *Agreement_Issues*

O componente *Traffic_Class* representa uma classe definida no SLA, onde pelo menos uma classe deve ser definida. O *Traffic_Class* é composto basicamente de três subcomponentes: *Agreement_Issues*, *QoS_Issues* e *Protocol_Stack*. A idéia é fazer com que cada classe possua pilha de protocolo, métricas de QoS e definições gerais de forma específica,

assim possibilitando que cada classe trate de determinados tipos de tráfego e aplicações. Além disso, os aspectos citados são diferenciados visando modularizar o SLA, facilitando uma possível adequação da linguagem quando necessário. A seguir é mostrada a parte do arquivo *XML Schema* que representa o componente *Traffic_Class*:

```
<complexType name="Traffic-Class">
  <sequence>
    <element name="ID-Class" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <element name="description" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <element name="qos-issues" type="QoS-Issues" minOccurs="1" maxOccurs="1" nillable="
    false"/>
    <element name="protocol-stack" type="ProtocolStack" minOccurs="1" maxOccurs="1"
    nillable="false"/>
    <element name="agreement-issues" type="Agreement-Issues" minOccurs="1" maxOccurs="1
    " nillable="false"/>
  </sequence>
</complexType>
```

O componente *Agreement_Issues* trata dos aspectos ligados ao contrato em relação a classe, como o tempo de duração do contrato (componente *Schedule*) e o preço relacionado ao mesmo (componente *Price*). A seguir é mostrada a parte do arquivo *XML Schema* que representa os componentes *Agreement_Issues*, *Schedule* e *Price*:

```
<complexType name="Time">
  <sequence>
    <element name="day" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    <element name="mounth" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    <element name="year" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    <element name="hour" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    <element name="minute" type="xsd:int" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
<complexType name="Price">
  <sequence>
    <element name="price" type="xsd:double" minOccurs="1" maxOccurs="1"/>
    <element name="unity" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <element name="payment-deadline" type="Time" minOccurs="1" maxOccurs="1" nillable="
    false"/>
  </sequence>
</complexType>
<complexType name="Schedule">
  <sequence>
    <element name="begin" type="Time" minOccurs="1" maxOccurs="1" nillable="false"/>
    <element name="end" type="Time" minOccurs="1" maxOccurs="1" nillable="false"/>
  </sequence>
</complexType>
<complexType name="Agreement-Issues">
  <sequence>
    <element name="schedule" type="Schedule" minOccurs="1" maxOccurs="1" nillable="
    false"/>
    <element name="price" type="Price" minOccurs="1" maxOccurs="1" nillable="false"/>
  </sequence>
</complexType>
```

3.3. Componentes *QoS_Issues* e *QoS_Parameters*

O componente *QoS_Issues* define os parâmetros relacionados à QoS para classe em questão (componente *QoS_Parameters*), além de definir o endereço para monitoramento dos parâmetros (atributo *Monitoring_URI*), o intervalo de tempo de atualização das medições (atributo *timeUpdate*) e o valor da multa caso ocorra violação dos parâmetros definidos no conjunto de *QoS_Parameters*. O esquema definido permite a declaração de diversas métricas de QoS, fazendo com que classes mais complexas possam ser delimi-

tadas mais criteriosamente. A seguir é mostrada a parte do arquivo *XML Schema* referente aos elementos citados:

```
<complexType name="QoS-Issues">
   <sequence>
     <element name="Monitoring-URI" type="xsd:string" minOccurs="1" maxOccurs="1"/>
     <element name="timeUpdate" type="xsd:double" minOccurs="1" maxOccurs="1"/>
     <element name="qos-parameters" type="QoS-Parameter" minOccurs="1" maxOccurs="
     unbounded"/>
     <element name="violation" type="Price" minOccurs="1" maxOccurs="1"/>
   </sequence>
</complexType>
```

O componente *QoS_Parameters* define os atributos referentes às métricas de QoS como atraso, *jitter*, perda, etc. O componente possui atributos para definir o valor médio desejado (*threshold*), o limite máximo/mínimo (*meanValue*), a unidade de medida (*unity*), a porcentagem de vezes que esses parâmetros podem ser quebrados durante o período de monitoramento (*tolerance*) e a prioridade da métrica (*priority*).

A prioridade é definida para indicar se esta métrica é prioritária ou não num processo de negociação dos parâmetros de QoS do SLA. Além desses atributos, é definida uma orientação para a métrica (atributo *orientation*), o objetivo é indicar se a métrica deve ter os valores minimizados (*DOWN*) ou maximizados (*UP*). Por exemplo, as definições para uma métrica de atraso visam uma minimização visto que quanto menor o atraso do tráfego mais benéfico é para a aplicação, no caso de uma métrica de vazão ocorre o inverso, a mesma tende a ser maximizada.

O fato de se declarar não somente a média (*meanValue*) mas também um limiar (*threshold*) faz com que o VNP tenha que garantir uma maior estabilidade à rede, evitando assim que raros problemas proporcionem a quebra do SLA sem um motivo real, onde a idéia de "raros" é definida pela tolerância definida. A seguir é mostrada a parte do arquivo *XML Schema* referente aos elementos citados anteriormente:

```
<simpleType name="Orientation">
   <restriction base="xsd:string">
     <enumeration value="DOWN"/><!-- enum const = 0 -->
     <enumeration value="UP"/><!-- enum const = 1 -->
   </restriction>
</simpleType>
<complexType name="QoS-Parameter">
   <sequence>
     <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
     <element name="unity" type="xsd:string" minOccurs="1" maxOccurs="1"/>
     <element name="threshold" type="xsd:double" minOccurs="1" maxOccurs="1"/>
     <element name="meanValue" type="xsd:double" minOccurs="1" maxOccurs="1"/>
     <element name="tolerance" type="xsd:double" minOccurs="1" maxOccurs="1"/>
     <element name="priority" type="xsd:bool" minOccurs="1" maxOccurs="1"/>
     <element name="orientation" type="Orientation" minOccurs="1" maxOccurs="1"/>
   </sequence>
</complexType>
```

3.4. Componente *Protocol_Stack*

O componente *Protocol_Stack* descreve a pilha de protocolo desejada para a classe em questão. A idéia é de personalizar a pilha de protocolo de acordo com os protocolos definidos a partir das funções listadas. Onde se tem a vantagem da utilização da linguagem XML, pois essa lista de protocolos pode ser modificada facilmente, de acordo com a disponibilidade do VNP em questão. Onde há a possibilidade de se informar se um certo protocolo é considerado prioritário (*priority-protocols*).

Algumas funcionalidades específicas da pilha de protocolo podem ser abdicadas (como por exemplo usar reserva de recurso), e outras podem ser deixadas em aberto, ou seja, há a necessidade de se ter a funcionalidade mas não é determinado o protocolo específico (como por exemplo o tipo de endereçamento). Além disso, possibilita suporte aos mecanismos de *DiffServ* e/ou *IntServ*.

Em geral, o SLA considera as seguintes informações referente à pilha de protocolo: encaminhamento por rótulo (*Label Switching*), gerenciamento ativo de filas (*Active Queue Management*), endereçamento (*Addressing*), protocolo de roteamento (*Routing Protocol*), reserva de recursos (*Resource Reservation*), IntServ e DiffServ. A seguir é mostrada a parte do arquivo *XML Schema* que representa o componente *Protocol_Stack*:

```
1  <complexType name="ProtocolStack">
2     <sequence>
3       <element name="labelSwitching" type="LabelSwitching" minOccurs="1" maxOccurs="1"/>
4       <element name="activeQueueManagement" type="ActiveQueueManagement" minOccurs="1"
        maxOccurs="1"/>
5       <element name="addressing" type="Addressing" minOccurs="1" maxOccurs="1"/>
6       <element name="routingProtocol" type="RoutingProtocol" minOccurs="1" maxOccurs="1"/
        >
7       <element name="resourceReservation" type="ResourceReservation" minOccurs="1"
        maxOccurs="1"/>
8       <element name="Intserv" type="xsd:bool" minOccurs="1" maxOccurs="1"/>
9       <element name="DiffServ" type="xsd:bool" minOccurs="1" maxOccurs="1"/>
10      <element name="descripton" type="xsd:string" minOccurs="1" maxOccurs="1"/>
11      <element name="priority-protocols" type="xsd:string" minOccurs="1" maxOccurs="
        unbounded"/>
12    </sequence>
13 </complexType>
14 <simpleType name="Addressing">
15    <restriction base="xsd:string">
16      <enumeration value="ANY-AD"/><!-- enum const = 0 -->
17      <enumeration value="IPv4"/><!-- enum const = 1 -->
18      <enumeration value="IPv6"/><!-- enum const = 2 -->
19    </restriction>
20 </simpleType>
21 <simpleType name="RoutingProtocol">
22    <restriction base="xsd:string">
23      <enumeration value="ANY-RP"/><!-- enum const = 0 -->
24      <enumeration value="RIP"/><!-- enum const = 1 -->
25      <enumeration value="OSPF"/><!-- enum const = 2 -->
26      <enumeration value="CBR"/><!-- enum const = 3 -->
27    </restriction>
28 </simpleType>
29 <simpleType name="ResourceReservation">
30    <restriction base="xsd:string">
31      <enumeration value="NONE-RR"/><!-- enum const = 0 -->
32      <enumeration value="ANY-RR"/><!-- enum const = 1 -->
33      <enumeration value="RSVP"/><!-- enum const = 2 -->
34      <enumeration value="RSVP-TE"/><!-- enum const = 3 -->
35    </restriction>
36 </simpleType>
37 <simpleType name="ActiveQueueManagement">
38    <restriction base="xsd:string">
39      <enumeration value="NONE-AQM"/><!-- enum const = 0 -->
40      <enumeration value="ANY-AQM"/><!-- enum const = 1 -->
41      <enumeration value="RED"/><!-- enum const = 2 -->
42      <enumeration value="ARED"/><!-- enum const = 3 -->
43      <enumeration value="FRED"/><!-- enum const = 4 -->
44      <enumeration value="RIO"/><!-- enum const = 5 -->
45      <enumeration value="BLUE"/><!-- enum const = 6 -->
46    </restriction>
47 </simpleType>
48 <simpleType name="LabelSwitching">
49    <restriction base="xsd:string">
50      <enumeration value="NONE-LS"/><!-- enum const = 0 -->
51      <enumeration value="ANY-LS"/><!-- enum const = 1 -->
```

```
52      <enumeration value="MPLS"/><!-- enum const = 2 -->
53      <enumeration value="ATM"/><!-- enum const = 3 -->
54      <enumeration value="FRAME-RELAY"/><!-- enum const = 4 -->
55    </restriction>
56  </simpleType>
```

A partir da linguagem definida os VNUs podem definir contratos SLA com os VNPs, onde há a possibilidade de se determinar diversas classes, cada qual com suas particularidades (parâmetros de QoS, pilha de protocolo, duração do contrato, custo, etc).

Desta forma, a negociação dos parâmetros de SLA pode ocorrer para as diversas classes, onde em um contexto multi provedor, um VNU pode definir, por exemplo, um SLA para uma certa classe "A" com um VNP, e um outro SLA "B" com outro VNP, que seria mais adequado para a classe em questão.

4. Estudo de Caso

Esta Seção tem por objetivo mostrar um exemplo de utilização da linguagem de especificação de SLA desenvolvida. No caso, o VNU determina duas classes para o VNP, uma rede mais adequada para tráfego multimídia, e outra rede para tráfego de dados.

A classe multimídia é composta de uma rede mais robusta: uma rede MPLS, com suporte a RSVP e DiffServ, utilizando RIP como protocolo de roteamento. Onde são definidos parâmetros de atraso (*Delay*), perda (*Loss*) e largura de banda (*Bandwidth*), sendo que o atraso é um parâmetro prioritário. Definiu-se os seguintes valores para as métricas: um limiar de 150 ms e um valor médio de 100 ms para atraso; um limiar de 10% de perda com valor médio de 5%; e uma largura de banda de 1000 Mbps.

A classe de dados é composta de uma rede mais simples, com requisitos inferiores e menor custo. A rede em questão define parâmetros de perda e largura de banda, onde nenhum dos parâmetros é prioritário. A pilha de protocolo em questão usa IPv6 para o endereçamento e OSPF como protocolo de roteamento. Os valores configurados para as métricas definidas foram: um limiar de 15% e um valor médio de 5% para perda; e uma largura de banda de 1500 Mbps. Consequentemente, o custo para essa rede é pequeno em relação à rede multimídia, sendo cerca de 5 vezes menor.

A seguir é mostrado o arquivo XML que representa o exemplo de SLA citado acima. O arquivo foi divido em partes, para ser detalhado. A parte abaixo trata do componente *Parties* do contrato SLA, onde os dois componentes *Actor* exigidos são definidos, um referente ao VNP e o outro ao VNU. As informações colocadas são exemplos para demonstrar o uso dos componentes no caso em questão.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <sla xmlSOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlSOAP-ENC="http://schemas
       .xmlsoap.org/soap/encoding/" xmlxsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlxsd="http://www.w3.org/2001/XMLSchema" xmlns="urn:sla">
3      <ID>1</ID>
4      <parties>
5          <VN-User>
6              <ID-Actor>1</ID-Actor>
7              <name>User</name>
8              <address>Street 1</address>
9              <description>user...</description>
10             <URI-Digital-Certification>uri_user</URI-Digital-Certification>
11         </VN-User>
12         <VN-Provider>
13             <ID-Actor>2</ID-Actor>
```

```
14              <name>VNP</name>
15              <address>Street 2</address>
16              <description>provider...</description>
17              <URI-Digital-Certification>uri_vnp</URI-Digital-Certification>
18          </VN-Provider>
19      </parties>
```

A seguir serão mostrados no SLA exemplos das duas classes descritas anteriormente: a classe multimídia e a classe de dados. A classe multimídia será detalhada componente por componente, enquanto que será mostrada uma visão geral da classe de dados.

O segmento abaixo representa a definição base da classe multimídia (nome, identificador e descrição), a multa em caso de quebra de contrato ("violation", um componente do tipo *Price*) e o componente *QoS_Issues* da classe em questão. No componente *QoS_Issues* além das três métricas de QoS declaradas (*Loss*, *Delay* e *Bandwidth*), são definidos o tempo de atualização e a URI para monitoramento da rede criada. A *tag* "payment-deadline" possui um identificador (id), pois as informações da mesma serão reutilizadas posteriormente no SLA.

```
1       <classes>
2           <ID-Class>1</ID-Class>
3           <name>Multimedia</name>
4           <description>Multimedia Traffic Class Example</description>
5           <qos-issues>
6               <Monitoring-URI>monitoring_uri_multimedia</Monitoring-URI>
7               <timeUpdate>1</timeUpdate>
8               <qos-parameters>
9                   <name>Delay</name>
10                  <unity>ms</unity>
11                  <threshold>150</threshold>
12                  <meanValue>100</meanValue>
13                  <tolerance>0.01</tolerance>
14                  <priority>true</priority>
15                  <orientation>DOWN</orientation>
16              </qos-parameters>
17              <qos-parameters>
18                  <name>Loss</name>
19                  <unity>%</unity>
20                  <threshold>10</threshold>
21                  <meanValue>5</meanValue>
22                  <tolerance>0.01</tolerance>
23                  <priority>false</priority>
24                  <orientation>DOWN</orientation>
25              </qos-parameters>
26              <qos-parameters>
27                  <name>Bandwidth</name>
28                  <unity>Mbps</unity>
29                  <threshold>1000</threshold>
30                  <meanValue>1000</meanValue>
31                  <tolerance>0</tolerance>
32                  <priority>false</priority>
33                  <orientation>UP</orientation>
34              </qos-parameters>
35              <violation>
36                  <price>10000</price>
37                  <unity>$$$</unity>
38                  <payment-deadline id="_5">
39                      <day>10</day>
40                      <mounth>2</mounth>
41                      <year>2011</year>
42                      <hour>12</hour>
43                      <minute>0</minute>
44                  </payment-deadline>
45              </violation>
46          </qos-issues>
```

A seguir é mostrado o trecho do SLA referente à definição da pilha de protocolo utilizada para a classe multimídia. Percebe-se que a classe em questão não faz uso dos recursos de gerenciamento ativo de fila e do IntServ.

```
<protocol-stack>
    <labelSwitching>MPLS</labelSwitching>
    <activeQueueManegment>NONE-AQM</activeQueueManegment>
    <adressing>IPv4</adressing>
    <routingProtocol>RIP</routingProtocol>
    <resourceReservation>RSVP</resourceReservation>
    <Intserv>false</Intserv>
    <DiffServ>true</DiffServ>
    <descripton>protocol stack example for multimedia traffic</descripton>
</protocol-stack>
```

O trecho seguinte trata das informações base do contrato para a classe multimídia: o tempo de duração (*tags* "begin" e "end") e o preço a ser pago pelo contrato (*tag* "price").

```
<agreement-issues>
    <schedule id="_8">
        <begin>
            <day>1</day>
            <mounth>1</mounth>
            <year>2011</year>
            <hour>12</hour>
            <minute>0</minute>
        </begin>
        <end>
            <day>1</day>
            <mounth>2</mounth>
            <year>2011</year>
            <hour>12</hour>
            <minute>0</minute>
        </end>
    </schedule>
    <price>
        <price>100000</price>
        <unity>$$$</unity>
        <payment-deadline id="_12">
            <day>1</day>
            <mounth>1</mounth>
            <year>2011</year>
            <hour>11</hour>
            <minute>59</minute>
        </payment-deadline>
    </price>
</agreement-issues>
</classes>
```

Abaixo é mostrada por completo a declaração da classe de dados explicada anteriormente no início desta seção. Os atributos "href" representam as referências feitas às *tags* anteriores declaradas com o atributo "id". Como exemplo, a *tag* "payment-deadline" desta classe faz referência às informações contidas na *tag* com "id" 5, ou seja, faz referência às informações de prazo de pagamento da multa contratual da classe anterior.

```
<classes>
    <ID-Class>2</ID-Class>
    <name>Data</name>
    <description>Data Traffic Class Example</description>
    <qos-issues>
        <Monitoring-URI>monitoring_uri_data</Monitoring-URI>
        <timeUpdate>1</timeUpdate>
        <qos-parameters>
            <name>Loss</name>
            <unity>%</unity>
            <threshold>15</threshold>
            <meanValue>5</meanValue>
```

```
13                  <tolerance>0.01</tolerance>
14                  <priority>false</priority>
15                  <orientation>DOWN</orientation>
16              </qos-parameters>
17              <qos-parameters>
18                  <name>Bandwidth</name>
19                  <unity>Mbps</unity>
20                  <threshold>1500</threshold>
21                  <meanValue>1500</meanValue>
22                  <tolerance>0</tolerance>
23                  <priority>false</priority>
24                  <orientation>UP</orientation>
25              </qos-parameters>
26              <violation>
27                  <price>2000</price>
28                  <unity>$$$</unity>
29                  <payment-deadline href="#_5"/>
30              </violation>
31          </qos-issues>
32          <protocol-stack>
33              <labelSwitching>NONE-LS</labelSwitching>
34              <activeQueueManegment>NONE-AQM</activeQueueManegment>
35              <adressing>IPv6</adressing>
36              <routingProtocol>OSPF</routingProtocol>
37              <resourceReservation>NONE-RR</resourceReservation>
38              <Intserv>false</Intserv>
39              <DiffServ>false</DiffServ>
40              <descripton>protocol stack example for data traffic</descripton>
41          </protocol-stack>
42          <agreement-issues>
43              <schedule href="#_8"/>
44              <price>
45                  <price>20000</price>
46                  <unity>$$$</unity>
47                  <payment-deadline href="#_12"/>
48              </price>
49          </agreement-issues>
50      </classes>
51  </sla>
```

A partir do XML mostrado, o cliente (VNU) e o provedor (VNP) estão aptos a negociar os recursos e os protocolos definidos para cada uma das classes. A flexibilidade existente com a linguagem desenvolvida permite a definição de diversas classes, até para um mesmo tipo de tráfego.

Comparando-se as duas classes declaradas no SLA é evidente a diferença de tecnologias e recursos exigidos, e consequentemente o suporte necessário para as mesmas. Da mesma forma, é bastante distinto o custo de cada uma delas. Sendo assim, o cliente se torna apto a negociar as características que são realmente necessárias a cada uma das classes definidas pelo mesmo.

De modo geral, a linguagem proposta habilita a negociação de SLAs para o novo paradigma da Internet do Futuro, baseda em virtualização de redes [Chowdhury and Boutaba 2009]. Trazendo benefícios para as empresas fazendo com que as mesmas consigam adequar os custos às necessidades de cada aplicação existente, ou seja, as empresas conseguem garantir a QoS necessária pagando o valor referente ao nível de infraestrutura necessário.

5. Conclusão e Trabalhos Futuros

A virtualização de redes surge como uma tecnologia promissora para superar a "ossificação" da Internet, provendo uma infraestrutura compartilhada para uma maior

variedade de serviços de rede e arquiteturas.

Junto a esse novo paradigma, surge a necessidade de se adaptar os mecanismos existentes à nova realidade que surge. Dentro desse contexto, este trabalho apresentou uma linguagem de especificação de SLA para a Internet do Futuro baseada em classes, permitindo a negociação completa da rede virtualizada entre o VNU e o VNP, onde as mesmas possuem características próprias.

Como trabalhos futuros pretende-se estender a linguagem proposta para englobar parâmetros mais específicos para o monitoramento das redes virtualizadas e dos parâmetros definidos no contrato. Além de se adicionar parâmetros de especificação de segurança, como criptografia, certificação digital e outros.

Agradecimento
Os autores agradecem ao CNPq pelo apoio financeiro.

Referências

Chowdhury, N. M. M. K. and Boutaba, R. (2009). Network virtualization: state of the art and research challenges. *Communication Magazine*, 47(7):20–26.

Fajjari, I., Ayari, M., and Pujolle, G. (2010). Vn-sla: A virtual network specification schema for virtual network provisioning. In *2010 Ninth International Conference on Networks (ICN)*, pages 337–342.

Keller, A. and Ludwig, H. (2003). The wsla framework: Specifying and monitoring service level agreements for web services. *J. Netw. Syst. Manage.*, 11(1):57–81.

Koch, B. F. and Hussmann, H. (2003). Overview of the project aquila. In *Proceedings of the 2003 international conference on Architectures for quality of service in the internet*, pages 154–164.

Lamanna, D., Skene, J., and Emmerich, W. (2003). Slang: a language for defining service level agreements. In *The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003. FTDCS 2003. Proceedings.*, pages 100 – 106.

Papadimitriou, P., Maennel, O., Greenhalgh, A., Feldmann, A., and Mathy, L. (2009). Implementing network virtualization for a future internet. 20th ITC Specialist Seminar on Network Virtualization - Concept and Performance Aspects.

Sun, W., Xu, Y., and Liu, F. (2005). The role of xml in service level agreements management. In *Proceedings of International Conference on Services Systems and Services Management.*, volume 2, pages 1118 – 1120 Vol. 2.

Tebbani, B. and Aib, I. (2006). Gxla a language for the specification of service level agreements. In *Autonomic Networking*, volume 4195 of *Lecture Notes in Computer Science*, pages 201–214. Springer Berlin / Heidelberg.

Trimintzios, P., Andrikopoulos, I., Pavlou, G., Cavalcanti, C., Goderis, D., T'Joens, Y., Georgatsos, P., Georgiadis, L., Griffin, D., Jacquenet, C., Egan, R., and Memenios, G. (2001). An architectural framework for providing qos in ip differentiated services networks. *7th IFIP/IEEE International Symposium on Integrated Network Management*.

XVI Workshop de Gerência e Operação de Redes e Serviços

♦

Sessão Técnica 4

Gerenciamento de Redes Móveis, Sem Fio e de Sensores

Modelo de handover vertical suave entre redes WiMAX e UMTS

Werley P. Santos[1], **Suéllen O. Reis**[1], **Rafael S. Nogueira** [1]
Fátima de L. P. Duarte-Figueiredo [1]

[1]Pontifícia Universidade Católica de Minas Gerais (PUC - Minas)
Rua Walter Ianni, 255 – Belo Horizonte –
Minas Gerais – 31980-110 – Brasil

{werleypsantos,suellen.oliveira}@gmail.com,sandernogueira@hotmail.com

fatimafig@pucminas.br

Abstract. *This work presents a soft vertical handover between WiMAX and UMTS networks. The model uses Mobile IP and Media Independent Handover (IEEE 802.21). To evaluate the model simulations were conducted through the Network Simulator version 2 (NS-2). Various WiMAX and UMTS integrated scenarios were simulated with multiple users in handover processes. The handover time was measured. In all situations evaluated, the model was efficient. The Quality of Service(QoS) evaluated showed that the application simulated had good performance during and after the handover, for all simulated scenarios.*

Resumo. *Este trabalho apresenta um modelo de handover vertical suave entre redes WiMAX e UMTS. O modelo utiliza o IP móvel e o Media Independent Handover (IEEE 802.21). Para avaliar o modelo, simulações foram desenvolvidas através do Network Simulator version 2 (NS-2). Foram simulados vários cenários de integração das redes WiMAX e UMTS, com múltiplos usuários em ambas as redes e em processo de handover. O tempo de handover foi medido e, em todas as situações avaliadas, o modelo mostrou-se eficaz. A Qualidade de Serviço (QoS) avaliada mostrou que os aplicativos simulados mantiveram bom desempenho durante e após o handover, em todos os cenários simulados.*

1. Introdução

Há alguns anos, existe uma forte tendência de integração de redes de telefonia celular com redes Wi-Fi (*Wireless Fidelity*) e WiMAX (*Worldwide Interoperability for Microwave Access*). Uma nova geração de redes conhecida como NGN (*Next Generation Network*) se baseia nessa integração [Kibria R. and Jamalipour 2005].A idéia é que os dispositivos móveis transmitem em ambientes heterogêneos sem perda de conexão.Com o intuito de fornecer um modelo para padronização de integração entre redes sem fio, o IEEE (*Institute of Electrical and Electronics Engineers*) especificou o padrão IEEE 802.21 denominado de MIH (*Media Independent Handover*) [STEIN 2006].

Este trabalho foi desenvolvido com o objetivo de avaliar um modelo de *handover* vertical suave entre redes WiMAX e UMTS. O modelo avaliado se baseia na utilização do IP Móvel (*Mobile IP - MIP*) [Nguyen et al. 2006] e do *Media Independent Handover*, proposto no padrão IEEE 802.21 [STEIN 2006]. Para avaliar o modelo, simulações

foram desenvolvidas com a elaboração de cenários integrados entre as redes WiMAX e UMTS. Os resultados mostraram que o modelo garante o *handover* vertical suave. Pode ser usado entre operadoras distintas, mediante um acordo ou na mesma operadora que possua redes de tecnologias diferentes. Algumas medidas foram apresentadas para ilustrar o desempenho das diferentes classes de serviços nas duas redes, distribuídas antes e após o procedimento de *handover*.

Este trabalho está organizado da seguinte maneira: a Seção 2 descreve os principais conceitos e trabalhos relacionados, onde são abordados: a proposta de padronização de integração especificada pelo IEEE [STEIN 2006] e o modelo de integração definido por [Nguyen et al. 2006], ambos utilizados no modelo proposto neste trabalho. A Seção 3 aborda o modelo de *handover* proposto. A Seção 4 apresenta as simulações, resultados e análises. Por fim, a Seção 5 apresenta conclusões e trabalhos futuros.

2. Principais Conceitos e Trabalhos Relacionados

[Kassar et al. 2008] apresenta estratégias de decisão de handover baseadas em contexto e uma política de implementação de *handover* utilizando MIP. O trabalho apresentado é teórico e não contém resultados de implementação ou simulação. Além disso, sequer faz referência ao IEEE 802.21. [Baek et al. 2008] Apresenta uma proposta de *handover* iniciado pela rede, baseado no IEEE 802.21 e em continuidade de serviço de QoS. Esse trabalho deixa em aberto o problema de endereçamento IP quando o nó móvel sai da rede de origem e entra em uma rede estrangeira.

Jakimoski e Janevski propuseram em [Jakimoski and Janevski 2009] um modelo de *handover* vertical semelhante ao deste trabalho, porém o trabalho deles não apresentou uma análise de QoS para múltiplos usuários em processos de handover, diferentemente deste trabalho que apresenta os efeitos do handover em medida de QoS.

2.1. Proposta de Padronização IEEE 802.21

A proposta do padrão IEEE 802.21 é descrever um *framework* contendo métodos e procedimentos que facilitem o *handover* entre redes com tecnologias homogêneas ou heterogêneas, ou seja, total transparência no *handover* horizontal e vertical. O padrão não aborta questões de decisões de *handover*, nem políticas de segurança ou qualidade de serviço. Ele oferece, simplesmente, uma funcionalidade denominada MIHF (MIH *Function*, capaz de fornecer informações e serviços para auxiliar nas tomadas de decisões dentro das etapas de inicialização, preparação e execução do procedimento de *handover*. Essa funcionalidade fica posicionada entre as camadas L2 (Enlace) e L3 (Rede) e permite que o MIH possa se comunicar com vários protocolos IP's. Como exemplos, podem ser citados o SIP (*Session Initiantion Protocol*), o IP Móvel (*Mobile IP - MIP*), além dos protocolos *DiffServ* e *IntServ* para tratamento de QoS (*Quality of Service*) e outros. Os serviços oferecidos pela funcionalidade MIH são: (i) Serviço de Eventos, através do MIES (*Media Independent Event Service*) ; (ii) Serviço de Comandos, através do MICS (*Media Independent Command Service*); e (iii) Serviço de Informação, através do MIIS (*Media Independent Information Service*), [STEIN 2006].

Promover os serviços viabiliza a comunicação entre as camadas rede e enlace, fazendo com que seja possível integrar as responsabilidades de cada uma dessas camadas. Assim, coopera, reciprocamente, a gerência de mobilidade, provida pela camada de rede, e a gerência do meio físico, provida pela camada de enlace.

2.2. Modelo de Integração proposto por Nguyen-Vuong

O trabalho apresentado por [Nguyen et al. 2006] propõe uma arquitetura de integração entre as redes WiMAX e UMTS. A movimentação do usuário envolve dois cenários de *handover*: *handover* WiMAX para UTRAN (*UMTS Terrestrial Radio Access Network*) e *handover* UTRAN para WiMAX. Em ambos os cenários, o procedimento consiste, basicamente, em uma preparação do equipamento móvel para conectar-se à rede alvo antes que a conexão com a rede corrente seja desfeita. Além disso, é descrito um procedimento comum para a redução de perda de pacotes, onde os pacotes são armazenados temporariamente durante o processo de *handover* até que a conexão com a rede alvo seja restabelecida e o equipamento móvel possa recebê-los.

No primeiro cenário descrito por [Nguyen et al. 2006], que envolve o procedimento de *handover* do WiMAX para a UTRAN, o equipamento do usuário está conectado em uma rede WiMAX e passa a ser atendido por uma rede UMTS. Ao entrar na zona de intercessão, o equipamento do usuário realiza uma mensuração da qualidade de sinal da rede UMTS. Caso as condições para o *handover* vertical sejam satisfatórias, a decisão de *handover* é tomada. A rede UMTS é notificada pela rede WiMAX através de mensagens de requisição de *handover*. O equipamento realiza a conexão com a rede UMTS através da UTRAN, o IP móvel registra a movimentação do equipamento na rede de origem (WiMAX) e um novo contexto PDP (*Packet Data Protocol*) é ativado entre o GGSN (*Gateway GPRS Support Node*) e o equipamento do usuário.

No segundo cenário, que envolve o procedimento de *handover* do UTRAN para o WiMAX, o equipamento do usuário está conectado em uma rede UMTS e passa a ser atendido por uma rede WiMAX. Ao entrar na zona de intercessão, o equipamento do usuário realiza uma mensuração da qualidade de sinal da rede WiMAX. Caso as condições para o *handover* vertical sejam satisfatórias, a decisão de *handover* é tomada. A rede WiMAX é notificada pela rede UMTS através de mensagens de requisição de *handover*. O equipamento realiza a conexão com a rede WiMAX, o IP móvel registra a movimentação do equipamento na rede de origem (UMTS) e a comunicação pode ser restabelecida pelo equipamento do usuário através da rede WiMAX.

O modelo por [Nguyen et al. 2006] se baseia na utilização do IP móvel na camada de rede, o que garante a compatibilidade do protocolo em ambas as redes, assim como a gerencia de mobilidade. Além disso, a proposta não requer muitas mudanças nas infraestruturas existentes das redes o que é uma grande vantagem.

O modelo considera que existe total cooperação entre as operadoras de redes UMTS e WiMAX. O que, na prática, nem sempre é verdade. Assim, pode ser apontada uma falha bastante grave encontrada no modelo, no que se refere à etapa de iniciação do procedimento de *handover*. Em ambos os cenários, a rede corrente é a responsável por iniciar esse procedimento e não o equipamento do usuário. Caso as redes WiMAX e UMTS sejam de operadoras distintas, pode não ser interessante para a operadora da rede corrente que o usuário faça um *handover* para outra rede. Um exemplo disso é o fator custo de transmissão, onde o custo de transmissão no ponto de acesso corrente pode ser maior que o custo de transmissão em um ponto de acesso de outra rede. Assim, esse seria um fator que não entraria na política de *handover* da operadora, já que seria interessante apenas para o usuário.

3. Modelo de *Handover* Avaliado

A proposta apresentada por [Nguyen et al. 2006], utiliza o IP móvel para gerir a mobilidade e promover a integração entre as redes WiMAX e UMTS. O IP móvel, por si só, abrange apenas a camada de rede e depende de uma contextualização de eventos ocorridos na rede, como por exemplo, eventos de descoberta de topologia, notificações de movimentação e pedidos de *handover*. Esses eventos são descritos com demasiada abstração pelo modelo de [Nguyen et al. 2006] deixando em aberto questões relevantes de implementação. O modelo de [Nguyen et al. 2006] deixa evidente que é necessária total colaboração entre operadoras, principalmente para a iniciação do procedimento de *handover*. Esse pré-suposto torna-se um fator crítico para o modelo, comprometendo sua eficiência.

O modelo de integração avaliado neste trabalho, [Santos 2009] considera não só a mobilidade, mas também a gerência do meio físico. Assim, a solução baseia-se na utilização do modelo de [Nguyen et al. 2006] atrelado à proposta de padronização apresentada pela IEEE [STEIN 2006], o 802.21 . A contextualização dos eventos ocorridos na rede e as formas de interação desses eventos com os vários elementos da rede foram descritos no modelo considerados mecanismos que permitem maior autonomia do usuário em relação às redes, minimizando a dependência de colaboração entre as operadoras.

3.1. Descrição da Integração

Existem, pelo menos, três elementos básicos na arquitetura proposta de redes integradas: o nó móvel MN (*Mobile Node*), a rede UMTS e a rede WiMAX. Além desses, outros dois elementos foram adicionados ao modelo do trabalho: um roteador MIP e um servidor WEB. A figura 1 dá uma visão geral dos elementos que compõem o cenário de integração modelado.

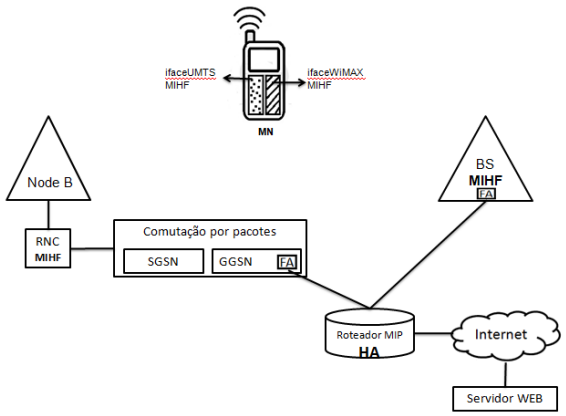


Figura 1. Visão geral dos elementos do cenário de integração

Na figura 1, observa-se um modelo de nó móvel composto por duas interfaces de conexão. Cada interface é provida de uma MIHF (MIH *Function*, sendo uma compatível com a rede UMTS (*ifaceUMTS*) e outra com a rede WiMAX (*ifaceWiMAX*). A rede UMTS [Eastwood et al. 2008] é simplificada e são ilustrados apenas os componentes que provêm a comutação por pacotes, SGSN (*Service GPRS Support Node*) e GGSN (*Gateway GPRS Support Node*). O MIH *Function* foi adicionado ao RNC (*Radio Network*

Control). O RNC é o responsável pelo gerenciamento dos recursos de rádio do NodeB, sinalização da interface aérea,processamento de chamadas e outros. Ao GGSN, foram adicionadas as funcionalidades inerentes ao FA (*Foreign Agent*) [Nagamuta 2006]. Essas funcionalidades são utilizadas pela rede UMTS, quando for essa a rede visitada pelo nó móvel.

Na rede WiMAX [Eastwood et al. 2008], a MIH *Function* foi adicionada à Estação Base – BS (*Base Station*). Além disso, foram adicionadas, também, as funcionalidades do FA – (*Foreign Agent*) à BS. Essas funcionalidades são utilizadas pela rede WiMAX, quando for essa a rede visitada pelo nó móvel.

O RoteadorMIP é o elemento responsável pelo roteamento correto dos pacotes. Ele tem a função de gerenciar a mobilidade do usuário, tanto na rede de origem quanto na rede visitada. Portanto, o IP móvel é atrelado a esse elemento. Dessa forma, ele age como um HA (*Home Agent*) [Nagamuta 2006], às vezes para a rede UMTS e às vezes para a rede WiMAX, dependendo, obviamente, de qual delas é a rede de origem. O servidor WEB, por sua vez, representa o nó com o qual o nó móvel se comunica. A esse elemento compete atender às requisições do nó móvel solicitadas ao mundo externo (Internet).

Além das interfaces de conexão e MIHF atreladas ao nó móvel, é necessário, ainda, que sejam incluídas outras funcionalidades. A figura 2 ilustra essas funcionalidades em camadas e como elas são distribuídas no nó. Pode-se observar uma macro-camada, denominada Gerência, que representa funcionalidades gerenciais do nó móvel, incluindo a parte que cabe às definições de política de *handover*. A gestão do *handover* pelo nó móvel, faz com que a responsabilidade de iniciação de *handover* seja retirada da rede. Além disso, existe uma camada intermediária, denominada Interfaces de Conexão, onde estão representadas as interfaces *ifaceUMTS* e *ifaceWiMAX*, cada qual com sua respectiva MIH *Function*. Cada MIHF, por sua vez, provê serviços de eventos (MIES), comando (MICS) e de informação (MIIS). Por fim, temos uma camada final, denominada Camada Física, que representa as conexões físicas de cada uma das interfaces, sendo uma para a rede UMTS (PHY UMTS) e uma para a rede WiMAX (PHY WiMAX).

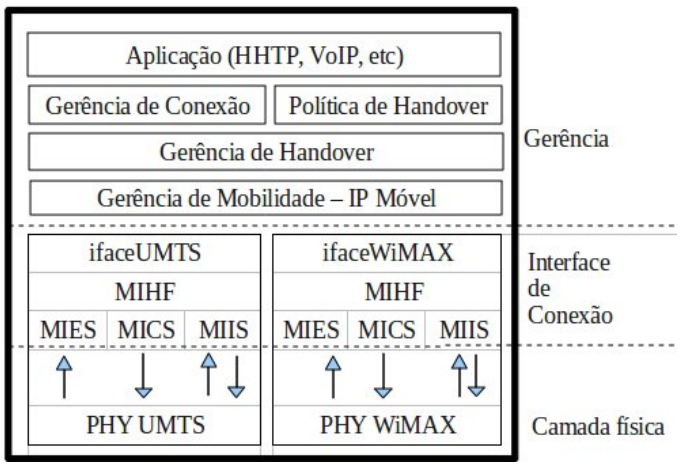


Figura 2. Composição do nó móvel distribuída em camadas.

3.2. Procedimento de *Handover*

O modelo de integração aqui avaliado, baseado no modelo de [Nguyen et al. 2006], envolve dois cenários de *handover*: *handover* WiMAX para UMTS *Terrestrial Radio Access Network* (UTRAN) e *handover* UTRAN para WiMAX.O modelo usa MIP, MIH e um *buffer* que armazena os dados diferenciados ao usuário em *handover*. Este *buffer* minimiza atrasos e perdas. Partindo do pré-suposto que o nó móvel encontra-se em operação na rede de origem, é premissa que o procedimento de registro de eventos do MIH *Function* ocorra tanto local quanto remotamente. Dessa forma, os eventos ocorridos no enlace do nó móvel podem ser reportados às camadas superiores locais e remotas. Essa etapa pode ser provida pelo padrão IEEE 802.21 [STEIN 2006] , através do evento MIH *Event Register* do *Media Independent Event Service*.

A partir dessa premissa, o procedimento de *handover* consiste em três macro- etapas: inicialização, preparação e execução do procedimento de *handover*. Além disso,o procedimento comum definido por [Nguyen et al. 2006] para redução de perda de pacotes, é mantido. Assim, os pacotes são armazenados, temporariamente, durante o processo de *handover*, até que a conexão com a rede alvo seja restabelecida e o equipamento móvel possa recebê-los.

3.2.1. *Handover* WiMAX para UMTS

No primeiro cenário, que envolve o procedimento de *handover* do WiMAX para a UTRAN, o equipamento do usuário está conectado a uma rede WiMAX e passa a ser atendido por uma rede UMTS. Ao entrar na zona de intercessão, o equipamento do usuário realiza uma mensuração da qualidade de sinal da rede UMTS. Caso as condições que atendam às políticas de *handover* estabelecidas pelo equipamento sejam satisfatórias, a decisão de *handover* é tomada e o equipamento informa o início do procedimento à sua rede de origem, WiMAX. A rede UMTS, que é a rede visitante, é notificada pela rede WiMAX através de comandos de requisição e resposta providos pelo *Media Independent Command Service* [STEIN 2006]. Durante essa etapa, notificações MIP, que indicam a movimentação do equipamento, são realizadas. Assim, os pacotes são retidos em *buffer*, para evitar a perda de dados. Após a finalização da conexão do equipamento com a rede UMTS, o equipamento notifica a finalização do *handover*. A comunicação é restabelecida e os dados mantidos em *buffer* são encaminhados ao equipamento, pela rede UMTS.

Para detalhar esse cenário, foram definidos dezesseis passos que descrevem desde a mensuração até a finalização do handover. Resumidamente, os passos são:

1. O nó móvel faz uma varredura dos canais, enviando para rede a mensagem *MIH_Get_Information_REQ*, para obter da topologia vizinha. O nó móvel também pode ser ”forçado”a executar esse procedimento, caso receba de sua rede de origem uma notificação do evento *MIH_Link_Going_Down*, de eminência de rompimento do link.
2. O nó móvel executa mensuração de informações recebidas da rede (*MIH_Get_Information_RESP*).
3. O nó móvel seleciona NodeB's que podem lhe servir e envia essa lista e o comando *MIH_Handover_Initiate* para sua Estação Base.

4. A Estação Base seleciona um NodeB na lista recebida e envia o comando *MIH_Handover_Prepare_REQ* para o RNC associado a ele.
5. O RNC verifica se o NodeB alvo suporta a configuração de link requerida e quais outros NodeB's associados a ele também podem atender à solicitação. O RNC, então, retorna uma resposta (*MIH_Handover_Prepare_RESP*)indicando os parâmetros apropriados para iniciar a configuração do novo link.
6. A Estação Base, descobre o endereço do RoteadorMIP (*Home Agent*).
7. O RoteadorMIP, ao receber a notificação, inicia o procedimento MIP de criação da tabela de mobilidade (*Mobility Binding List*) do *Home Agent*.
8. A Estação Base envia o comando *MIH_Configure* para o nó móvel indicando os parâmetros para configuração do novo link.
9. O nó móvel, ao receber o comando de configuração, dispara o evento *MIH_Link_Down* para a Estação Base.
10. O nó móvel, inicia o procedimento de conexão GPRS com a rede UMTS, trocando mensagens de autenticação, autorização e contabilidade AAA –(*Authentication, Authorization and Accounting*).
11. Após o registro no núcleo GPRS, o nó móvel executa o procedimento de ativação de contexto PDP junto à rede UMTS.
12. O nó móvel dispara o evento *MIH_Link_Up* para o RNC, indicando que a conexão UMTS está estabelecida e o link está disponível para uso.
13. O nó móvel envia o comando *MIH_Handover_Complete* para o RNC.
14. O RNC, então, envia o comando *MIH_Network_Address_Information* para a rede WiMAX (Estação Base)informando que o endereço IP antigo (*home address*) deve ser reconfigurado para o novo endereço IP (*Care-of address*) e qual o ponto de acesso (GGSN/FA) será utilizado.
15. A Estação Base envia ao RoteadorMIP (*Home Agent*) uma notificação para registro do *Care-of address*.
16. O RoteadorMIP, finaliza o procedimento MIP de criação da tabela de mobilidade do *Home Agent* e o tempo de vida da tabela (*LifeTime*) é reiniciado. A partir de então, os dados armazenados em *buffer* podem ser encaminhados para o endereço de tratamento (*Care-of address*), por meio de tunelamento. A comunicação é restabelecida.

A figura 3 ilustra os dezesseis passos acima definidos para o procedimento de handover do WiMAX para o UMTS.

3.2.2. Handover UMTS para WiMAX

No segundo cenário, que envolve o procedimento de *handover* da UTRAN para o WiMAX, o equipamento do usuário está conectado a uma rede UMTS e passa a ser atendido por uma rede WiMAX. Ao entrar na zona de intercessão, o equipamento do usuário realiza uma mensuração da qualidade de sinal da rede WiMAX. Caso as condições que atendam às políticas de *handover* estabelecidas pelo equipamento sejam satisfatórias, a decisão de *handover* é tomada e o equipamento informa o início do procedimento à sua rede de origem, UMTS. A rede WiMAX, que é a rede visitante, é notificada pela rede UMTS através de comandos de requisição e resposta providos pelo *Media Independent Command Service*. Durante essa etapa, notificações MIP, que indicam a movimentação

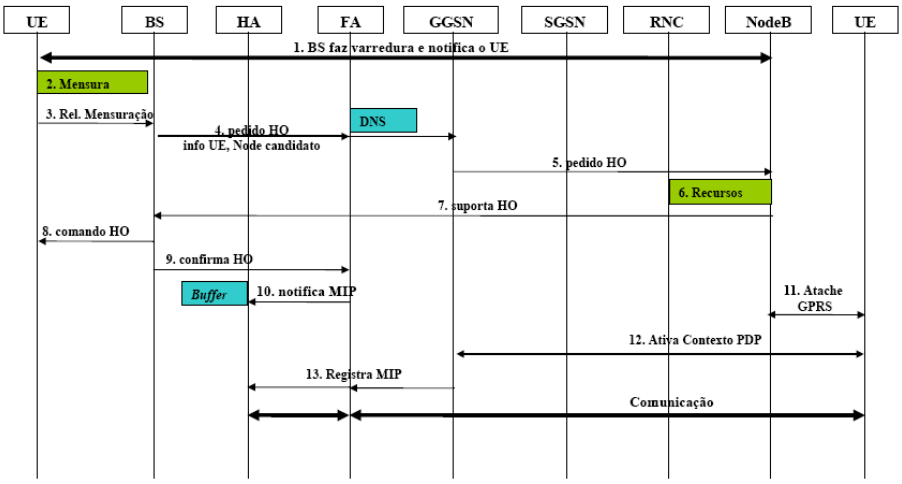


Figura 3. Procedimento de Handover WiMAX-UMTS

do equipamento, são realizadas. Assim, os pacotes são retidos em *buffer* para evitar a perda de dados. Após a finalização da conexão do equipamento com a rede WiMAX, o equipamento notifica a finalização do *handover*. A comunicação é restabelecida e os dados mantidos em *buffer* são encaminhados ao equipamento pela rede WiMAX.

Diante da grande similaridade entre os procedimentos de *handover* WiMAX-UMTS e UMTS-WiMAX, os passos que descrevem o processo de *handover* UMTS-WiMAX serão suprimidos. A única particularidade desse cenário refere-se à etapa de conexão do nó móvel com sua rede destino, no caso, a rede WiMAX. Os passos 10, 11 e 12 descrevem essa etapa com detalhes:

10. O nó móvel inicia o procedimento de conexão com a Estação Base,trocando mensagens de *Registration Request* e *Registration Response*.
11. A Estação Base, disponibiliza o canal de *uplink* e *downlink* para o nó.
12. A Estação Base disponibiliza um endereço IP local para o nó móvel, que será usado como *Care-of address*. Esse endereço de IP é associado à interface de conexão com a rede WiMAX (*ifaceWiMAX*).

A figura 4 ilustra todos os passos do procedimento de handover do UMTS para o WiMAX.

4. Simulações, Resultados e Análises

As simulações foram realizadas com o objetivo de avaliar o modelo proposto. Para o modelo descrito, foi criado um ambiente de simulação com um cenário integrado contendo as redes WiMAX e UMTS,os módulo das mesmas foram devidamente validados por seus autores. Dois tipos de análises dos resultados foram feitas: análise de garantia do processo de handover suave e análise do desempenho do usuário nas duas redes, antes, durante e após a execução handover. Na primeira análise, observou-se que as conexões foram mantidas nos dois sentidos de handover: UMTS para WIMAX e WIMAX para UMTS. Na segunda análise, foram avaliados duração do *handover*, vazão média, atraso médio e *jitter*. As simulações são descritas a seguir, juntamente com seus resultados e análises. O intervalo de confiança definido para validação dos resultados foi de 95%.

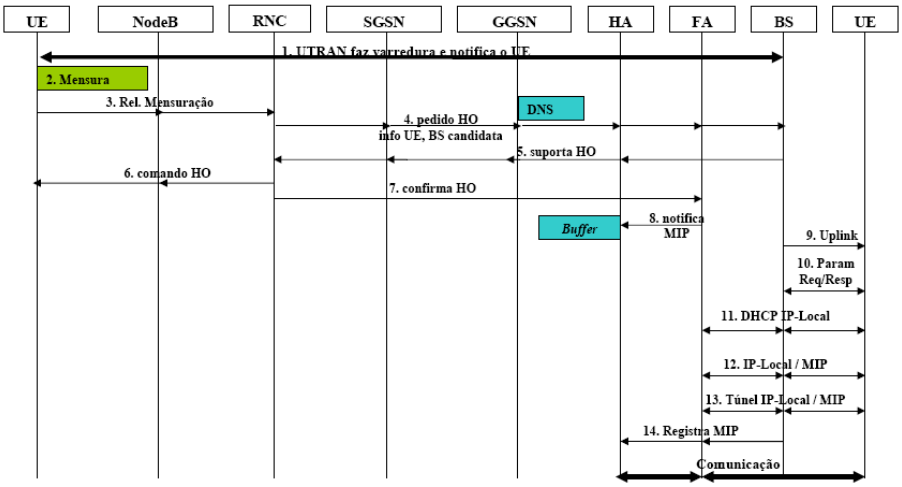


Figura 4. Procedimento de Handover UMTS-WiMAX

4.1. Simulações

As classes de serviço foram definidas para a simulação: *Conversational*, *Background*, *Interactive* e *Streaming*. Essas classes foram definidas em [3GPP 2005] e cada uma corresponde a um tipo de tráfego específico, com características próprias largura de banda e fluxo de pacotes. Para a simulação, a modelagem dessas classes foi feita seguindo a proposta de [Antoniou 2004]. O modelo de simulação foi definido pelos parâmetros número de usuários, tempo total da simulação e serviço atribuído a cada usuário. O momento em que cada usuário faz o *handover* é variável. A Tabela 1 descreve um exemplo de valores que definem um cenário de simulação. O NS2 [NS2 2004] foi utilizado.

Tabela 1. Exemplo de Parâmetros de Simulação

Distribuição das Classes	Usuários Conversacional	15%
	Usuários Interactive	40%
	Usuários Streaming	30%
	Usuários Background	15%
Redes Iniciais	Usuários UMTS	30%
	Usuários WiMAX	70%
	Tempo de Simulação	300s

Os parâmetros de QoS, tais como atraso, *jitter* e vazão são obtidos a partir de ferramentas que foram desenvolvidas para processamento de *trace files* gerados pelo NS2. A análise do comportamento da rede pode ser realizada, a partir dos parâmetros de QoS antes, durante e após o procedimento de *handover* proposto. Com isso, consegue-se afirmar que o mesmo é eficaz.

4.2. Resultados

As simulações foram realizadas inicialmente com apenas um usuário realizando o *handover* e, posteriormente, variações no número de usuários (30,70,100,500,700,1000). Cada simulação foi repetida 33 vezes, para validação estatística com 95% de intervalo de confiança.

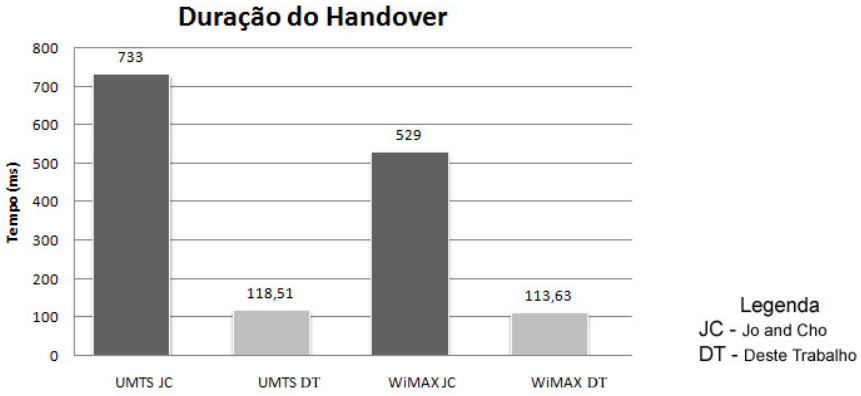
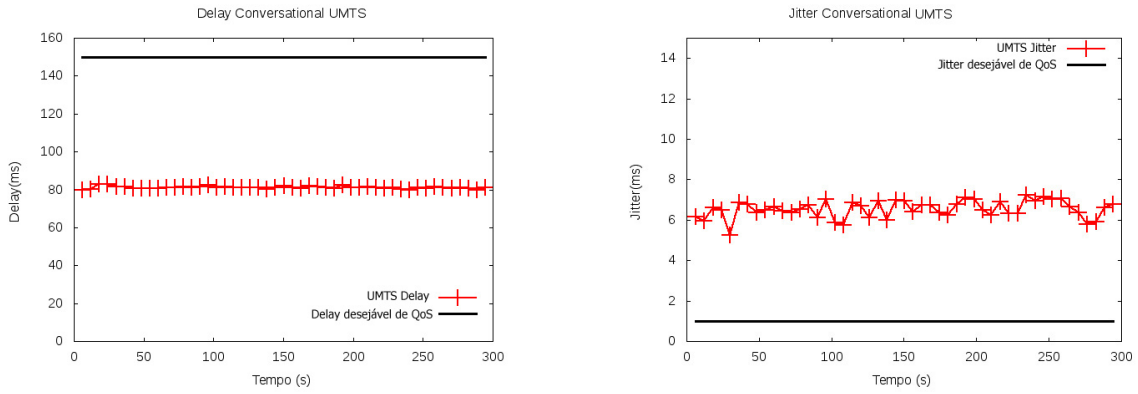


Figura 5. Comparação da duração do handover

Santos em [Santos 2009] mostra que seu modelo funciona para 1 usuário, provando que a conexão se mantém nos dois sentidos: UMTS x WiMAX, WiMAX x UMTS .

O gráfico da Figura 5 mostra a duração do *handover* para o modelo de integração de [Jo and Cho 2008] e para o modelo apresentado neste trabalho. O tempo de duração do modelo aqui apresentado foi menor do que o de [Jo and Cho 2008]. O maior tempo gasto no processo de *handover* foi de, aproximadamente, 120 ms, em contrapartida aos 733 ms do modelo de [Jo and Cho 2008]. Desse modo, o tempo do modelo proposto neste trabalho é perfeitamente aceitável. As simulações foram realizadas considerando-se que, neste trabalho, os usuários têm sempre à sua disposição as redes UMTS e WiMAX. UMTS comparada à rede WiMAX, é uma rede que possui menor largura de banda, está mais sujeita a gargalos do que a rede WiMAX. A rede WiMAX suporta muitos usuários utilizando aplicações que requerem alta largura de banda, como aplicações de streaming, por exemplo. Desse modo, os gráficos apresentados neste trabalho são da rede UMTS, que poderia ter a QoS prejudicada ao receber usuários vindos da rede WiMAX.



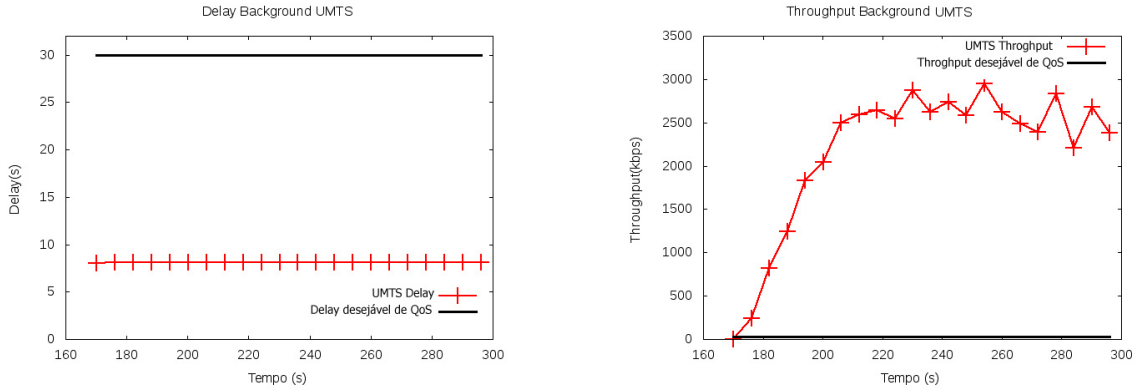
(a) Delay (b) Jitter

Figura 6. Delay e Jitter da classe de Serviço Conversational

Os resultados mostrados nos gráfico das Figuras de 6 a 9 representam a média de cada parâmetro de desempenho para um usuário, obtido na última simulação, que foi realizada com 1000 usuários. Os parâmetros medidos, na simulação, foram comparados

aos valores considerados desejáveis, conforme recomenda o 3GPP. Para cada classe de serviço, as avaliações ocorreram observando-se o parâmetro de maior importância para determinada classe. Neste artigo são mostradas apenas os resultados para UMTS que é a rede de menos recursos e, portanto, a mais vulnerável a oscilações de tráfego. Em [Reis 2010], há a análise completa de QoS, para as duas redes envolvidas no *handover*.

Para a classe *conversational*, os parâmetros avaliados foram atraso e *jitter*. No gráfico representado pela Figura 6(a), pode-se perceber que o atraso médio de *conversational* está sendo atendido quanto aos requisitos de QoS. O *jitter* variou entre 5 e 7 ms durante toda a simulação, acima do desejável. Os dois gráficos mostram que, mesmo na rede com menos recursos, UMTS, a QoS mantêm-se boa.



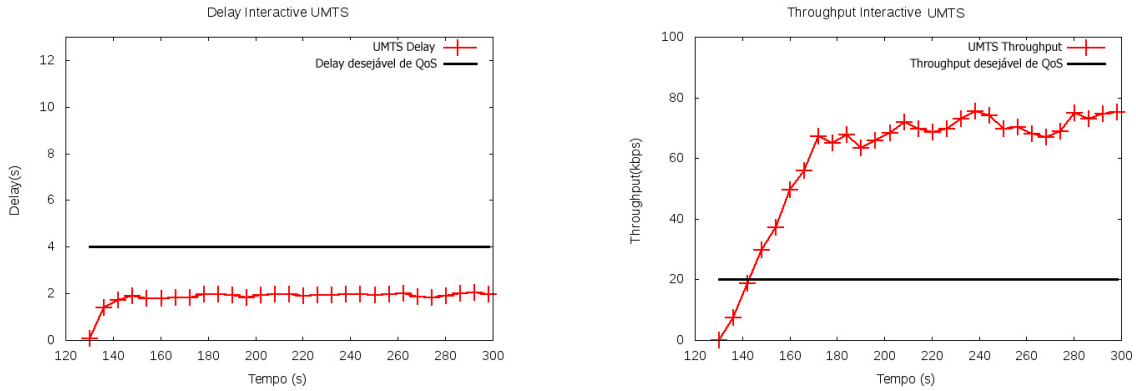
(a) Delay (b) Throughput

Figura 7. Delay e Throughput da classe de Serviço Background]

Para as aplicações que utilizam serviços de *background*, os parâmetros de QoS não são rigorosos, por se tratar de uma classe de serviço que não é de tempo real. Nessa classe, o *delay* das aplicações deve ser menor que 30 s e o *throughput* deve alcançar taxas mínimas de 2.8 kbps. Pelos gráficos da figura 7(a) e 7(b), percebe-se que dificilmente esse tipo de aplicação irá ultrapassar as métricas de QoS. Para as aplicações que utilizaram os serviços *interactive* e *streaming*,os parâmetros de QoS foram atendidos e os serviços não ficaram comprometidos, conforme mostram os gráficos das figuras 8 e 9. *Jitter* não foi avaliado por não ser um parâmetro importante para a chave *background*.

A classe de serviço *interactive* é extremamente sensível a *delay* e a *throughput*. Os gráficos das Figuras 8(a) e 8(b) mostram que o *delay* ficou em torno de 2s durante toda a simulação, representando um valor aceitável para essa classe, que requer *delay* máximo de 4s. O *throughput* mostrado no gráfico da Figura 8(b) alcançou e ultrapassou o *throughput* mínimo desejável, de 20 kbps para se obter QoS e durante todo o tempo da simulação desejável.

Para avaliar a classe de serviços *streaming*, foram analisados três parâmetros de QoS: atraso, *jitter* e vazão. *Streaming* é a classe que, se ao menos um dos três parâmetros de QoS não for atendido, o serviço fica comprometido. O gráfico da Figura 9(a) mostra que o *jitter* está variando até 1 s, sendo que 2 s é o limite superior para se obter QoS. O gráfico da Figura 9(b), mostrou que a vazão, no início da simulação, começou a subir. No instante 70s, a vazão atingiu o valor mínimo necessário de apenas 38 kbps, permanecendo



(a) Delay (b) Throughput

Figura 8. Delay e Throughput da classe de Serviço Interactive

acima do limite até o final da simulação. O gráfico 9(c), do atraso, mostrou que o atraso médio dessa aplicação, não ultrapassou 10 s, se mantendo entre 2 e 4s, em média.

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou um modelo de *handover* vertical suave baseado em MIP e no padrão IEEE 802.21. O modelo foi avaliado para os dois sentidos de *handovers*: de UMTS para WiMAX e de WiMAX para UMTS.

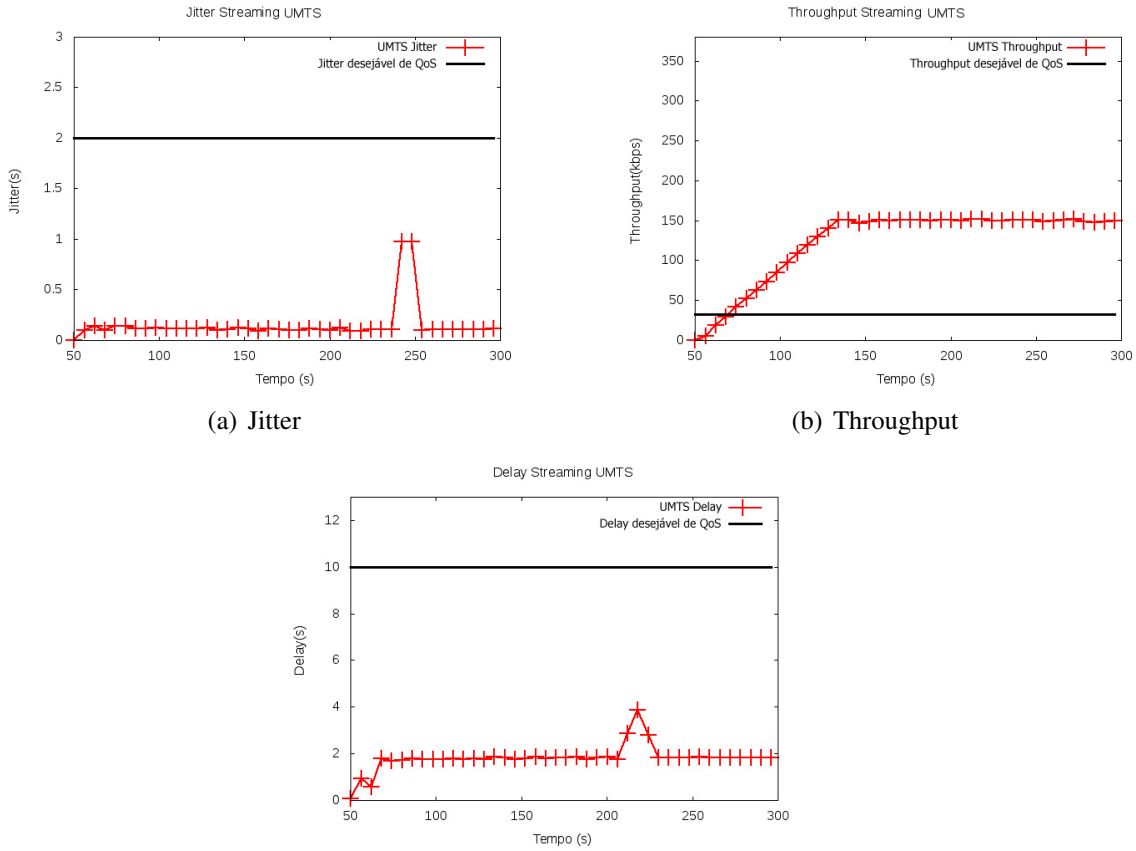
Como a QoS deve ser mantida em qualquer rede que o usuário esteja, em ambientes heterogêneos integrados, o modelo foi avaliado em alguns parâmetros de QoS. Neste trabalho foram mostrados os resultados da análise de QoS para UMTS, que é a rede mais vulnerável a gargalos sob forte demanda de tráfego.

Foram avaliadas as medidas de desempenho vazão, atraso e o *jitter* médio. As medidas dos resultados de todas as classes de serviços foram comparadas com os requisitos de QoS especificados pelo 3GPP. Em todas as classes, os requisitos de QoS foram atendidos, exceto o *jitter* de conversational que ficou acima do desejado. Com isso, foi possível mostrar que o modelo de handover vertical suave baseado em MIP e em MIH é eficaz, tanto do ponto de vista da garantia de disponibilidade ininterrupta, quanto de bom desempenho.

Como trabalhos futuros, podem ser destacadas algumas linhas de pesquisa,como por exemplo: implementação e análise do gerenciamento de mobilidade em extensões do IPv6 para minimizar o tempo de mudança de ponto de acesso. Isso poderá reduzir o tempo em que os pacotes permanecem em *buffer* durante o procedimento de *handover* e, conseqüentemente, o pico de variação de atraso médio será menor; implementação de um repositório que contenha informações sobre a topologia da rede. Isso poderá evitar o processo de varredura da rede, em busca dessas informações e, conseqüentemente, diminuirá o fluxo de informações na rede. Extensão da proposta, abrangendo outros padrões de redes sem fio como *Bluetooth* e Wi-Fi, também é um trabalho futuro.

Referências

3GPP (2005). End-to-end Quality of Service (QoS) concept and architecture. TS 23.207, 3rd Generation Partnership Project (3GPP).



(a) Jitter

(b) Throughput

(c) Delay

Figura 9. Jitter, Throughput e Delay da classe de Serviço Streaming

Antoniou, J. (2004). A system level simulator for enhanced umts coverage and capacity planning, Masthers Thesis, Department of Computer Science – University of Cyprys.

Baek, J.-Y., Kim, D.-J., Suh, Y.-J., Hwang, E.-S., and Chung, Y.-D. (2008). Network-initiated handover based on ieee 802.21 framework for qos service continuity in umts/802.16e networks. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2157 –2161.

Eastwood, L., Migaldi, S., Xie, Q., and Gupta, V. (2008). Mobility using ieee 802.21 in a heterogeneous ieee 802.16/802.11-based, imt-advanced (4g) network. *Wireless Communications, IEEE*, 15(2):26 –34.

Jakimoski, K. and Janevski, T. (2009). Performances of vertical handovers for multimedia traffic between wlan, wimax and 3g mobile networks. In *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, Mobimedia '09, pages 11:1–11:4, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Jo, J. and Cho, J. (2008). Cross-layer optimized vertical handover schemes between mobile wimax and 3g networks. *TIIS*, 2(4):171–183.

Kassar, M., Kervella, B., and Pujolle, G. (2008). An overview of vertical handover decision strategies in heterogeneous wireless networks. *Comput. Commun.*, 31:2607–2620.

Kibria R., V. M. and Jamalipour, A. (2005). A consolidated architecture for 4g/b3g networks. In *Proceedings of the International Conference on Wireless communications, networking and mobile computing*, WiCOM'05, pages 86–89. IEEE Press.

Nagamuta, V. (2006). Um arcabouço para composição, teste e sumulação de protocolos de handover suave. Tese de doutorado, disponível http://www.ime.usp.br/ song/papers/tese-vera.pdf.

Nguyen, Q., Fiat, L., and Agoulmine, N. (2006). An architecture for umts-wimax interworking. *Broadband Convergence Networks, 2006. BcN 2006. The 1st International Workshop on*, pages 1–10.

NS2 (2004). The network simulator. Disponível em http://www.isi.edu/nsnam/ns. Acessado em 27 de abril de 2007.

Reis, S. O. (2010). Avaliação de qualidade de serviço em redes wimax e umts integradas para múltiplos usuários, Dissertação de Mestrado, Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-Graduação em Informática, Brasil.

Santos, W. P. (2009). Especificação e validação de um modelo para handover vertical entre as redes wimax e umts utilizando ip móvel e media independet handover, Dissertação de Mestrado, Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-Graduação em Informática, Brasil.

STEIN, J. (2006). Survey of ieee802.21 media independent handover services. Disponível em: `http://www.cs.wustl.edu/~jain/cse574-06/ftp/handover/index.html`.

Uma Métrica de Roteamento Baseada na Taxa da Fila Aplicada às *Wireless Mesh Networks* com Tráfego VoIP

Cleverton Juliano Alves Vicentini[1], Mauro Sergio Pereira Fonseca[1], Roberson Cesar Alves de Araujo[2]

[1]Programa de Pós-Graduação em Informática Aplicada - (PPGia),
Pontíficia Universidade Católica do Paraná (PUCPR)
Caixa Postal 17.315 – 80.215-901 – Curitiba – PR – Brasil

[2]Instituto de Tecnologia do Paraná (TECPAR)
80.350-010 – Curitiba – PR – Brasil

{cleverton,mauro.fonseca}@ppgia.pucpr.br, roberson.araujo@tecpar.br

Abstract. *Nowadays a lot of research has been made to reach the best routing protocols and metrics in order to improve wireless mesh networks. Although achieving better routes in a wireless mesh network is not a trivial work. Many researches show generic scheme protocols aiming to work in any topology case, but they do not work in any scenario. This study shows a new router metric named FQ (Factor-Q), which aims to work in wireless mesh network scenarios optimized for a low mobility and VoIP traffic. The FQ's performance was compared and evaluated with ML (Minimum Loss), MD (Minimum Delay) and ETX (Estimated Transmission Count) metrics. Simulations reveal that the proposed FQ metric provides a better set of results than ML, MD and ETX metrics.*

Resumo. *Muita pesquisa é realizada a fim de encontrar melhores protocolos e métricas de roteamento para que se adequem a redes Wireless do tipo Mesh. Porém, buscar melhores rotas em uma rede sem fio não é uma tarefa trivial. Grande parte dos estudos objetivam a busca de soluções que se adaptem a qualquer topologia Wireless Mesh, mas nem sempre um protocolo ou métrica de roteamento irá operar de forma satisfatória em diversos ambientes. O presente artigo demonstra uma nova métrica de roteamento intitulada FQ (Factor-Q), que tem como objetivo atuar em redes Wireless Mesh com cenários de baixa mobilidade e com tráfego VoIP. Seu desempenho foi comparado e avaliado com a métrica ML (Minimum Loss), MD (Minimum Delay) e ETX (Estimated Transmission Count). Simulações realizadas com o NS-2 demonstraram que a métrica FQ propicia um melhor desempenho no modelo de rede utilizado com tráfego VoIP.*

1. Introdução

As redes locais sem fio definidas pelo padrão IEEE 802.11, que fornecem acesso sem fio a computadores e dispositivos móveis através de ondas de rádio, propiciam uma fácil implantação por não haver necessidade de uma estrutura de rede cabeada, sendo assim esta tecnologia tende a ser cada vez mais explorada por instituições e comunidades. Seguindo este mesmo princípio recentemente surgiu uma nova tecnologia intitulada: redes em malha sem fio (*Wireless Mesh Networks)* que tem como principal atrativo seu

custo reduzido para cobertura de áreas relativamente grandes onde é financeiramente inviável a instalação de uma infra-estrura de rede cabeada. Instituições de Ensino têm utilizado a plataforma *Wireless Mesh* de forma a prover serviços de Internet a seus alunos e funcionários, além de interligar prédios com a comunicação sem fio [ReMesh 2005], [Saade et al. 2007], [Tsarmpopoulos et al. 2005].

As redes *Wireless Mesh* têm como principal característica a utilização de roteadores sem fio, geralmente fixos e com maior poder de processamento em relação aos roteadores móveis que geralmente apresentam o suprimento de energia limitado. Os roteadores são responsáveis por encaminhar tráfego para os demais roteadores que compõem a rota de destino e também por receber e encaminhar tráfego para os nós clientes, estes nós podem estar conectados tanto via cabo como sem fio.

Características que devem estar presentes nas redes *Mesh* são auto-organização e auto-configuração, tais características possibilitam a manutenção das conexões dos roteadores presentes na rede de forma automática, visando a inclusão de novos roteadores na rede para o aumento da área de cobertura da rede *Mesh* [Akyildiz et al. 2005]. De uma forma geral, os protocolos de roteamento utilizados em redes *Wireless Mesh* são adaptações de protocolos de roteamento para redes *ad hoc*. Porém, o fato dos protocolos de roteamento *ad hoc* serem desenvolvidos para redes onde nós são móveis, podem não trazer bons resultados se utilizados nas redes *Wireless Mesh*.

O desempenho de uma rede *Wireless Mesh* bem como seu comportamento recaem geralmente na combinação do protocolo de roteamento com a métrica utilizada. Pesquisas envolvendo a utilização de protocolos de roteamento e métricas para redes *Mesh* têm dado prioridade à diminuição do atraso e ao aumento da vazão [Akyildiz et al. 2005].

A importância de cada métrica de roteamento é reflexo do que se espera de determinadas redes *Mesh*. Algumas métricas avaliam a taxa de perda de pacotes, outras utilizam múltiplos canais para melhor utilização do meio físico, outras fazem a utilização da largura de banda como medida. Este artigo propõe uma métrica de roteamento que utiliza as informações referentes a fila de um nodo. Seu objetivo é obter menores taxas de perda, menor atraso e melhor vazão nas redes *Mesh* estacionárias com tráfego *VoIP*. Seguindo este critério, rotas alternativas serão definidas com o objetivo de oferecer melhor desempenho a rede. A métrica apresentada neste documento é intitulada *Factor-Q* (FQ).

A motivação que leva ao estudo e desenvolvimento de uma métrica de roteamento que se adapte a redes *Wireless Mesh* com tráfego de Voz, deve-se ao crescimento e popularização da tecnologia *Voice over Internet Protocol* (VoIP), bem como pela crescente demanda de serviços multimídia. Este crescimento justifica-se pelas reduções significativas com os custos em telefonia. O cenário escolhido para os testes foi o Campus da Pontifícia Universidade Católica do Paraná (PUC-PR), por ser um cenário viável de implantação destas tecnologias.

A implementação da métrica de roteamento *Factor-Queue* (FQ) foi baseada na concatenação das informações referentes a taxa de utilização de fila dos nós vizinhos. Desta maneira quando um determinado enlace apresenta altas taxas de utilização de fila esta informação é repassada ao protocolo de roteamento forçando a escolha de rotas alternativas.

Com o objetivo de avaliar o comportamento da métrica FQ (*Factor-Queue*), foram

realizadas simulações no *Network Simulator 2* [NS2 2010], utilizando-se extensões para o OLSR e as métricas ML (*Minimum loss*), MD (*Minimum Delay*) e ETX (*Expected Transmission Count*), desenvolvidas para o NS-2 [Cordeiro et al. 2007]. Para interpretação dos resultados foi utilizada a ferramenta *Gnuplot* [Gnuplot 2010].

Este documento está organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados. A seção 3 descreve a métrica *Factor-Q*. A seção 4 apresenta o cenário de simulação e parâmetros utilizados na simulação, seção 5 apresenta os resultados obtidos e por fim seção 6 contém a conclusão deste artigo.

2. Trabalhos Relacionados

Esta seção descreve o comportamento dos protocolos de roteamento *wireless* e aborda algumas métricas de roteamento utilizadas em redes *Mesh*.

2.1. Protocolos de Roteamento para Redes *Wireless*

A classificação dos protocolos de roteamento *Wireless* é realizada da seguinte forma: pró-ativos, reativos e híbridos. Os protocolos pró-ativos realizam o processo de descoberta da rota de forma constante, desta maneira quando ocorrer a necessidade de tranferir dados a rota já é conhecida para utilização imediata. Os protocolos pró-ativos podem ser adequados para redes *Mesh* de baixa mobilidade, onde não existe a limitação de energia pois os roteadores são geralmente fixos e com alimentação contínua. Os protocolos reativos realizam o processo de descoberta da rota somente quando necessitam enviar dados, desta forma economizando processamento dos nós, sendo adequados para redes de alta mobilidade, onde economia de energia é um fator prioritário. A metodologia utilizada nos protocolos de roteamento híbridos é a concatenação dos conceitos pró-ativos e reativos, dividindo a rede em zonas de roteamento de forma que em deteminados localidades o princípio pró-ativo é utilizado e em outros momentos o princípio reativo é aplicado.

As redes *Wireless Mesh* podem oferecer como recurso o poder de processamento e a não limitação de energia [Passos and Albuquerque 2007], mesmo não aproveitando estas características alguns protocolos de roteamento desenvolvidos para redes *ad-hoc* foram implantados em redes *Wireless Mesh*. São exemplos de protocolos *ad-hoc* utilizados em redes *Wireless Mesh*: DSR (*Dynamic Source Routing*)[Johnson et al. 2003], AODV *(Ad Hoc On-Demand Distance Vector)* [Perkins et al. 2003] e OLSR (*Optimized Link State Routing Protocol*) [Clausen and Jacquet 2003].

Nas redes *Wireless* o grande número de mensagens de controle disparadas pelos nodos podem vir a comprometer a estabilidade da rede, alguns protocolos de roteamento pró-ativos têm como princípio diminuir esta sobrecarga de mensagens na rede. Um exemplo é o protocolo de roteamento OLSR (*Optimized Link State Routing Protocol*) [Clausen and Jacquet 2003], que utiliza a abordagem de *Multipoint Relays* (MPR). Os MPR são um conjunto de vizinhos selecionados por um determinado nó que terão a tarefa de retransmitir mensagens de controle pela rede. A utilização da abordagem MPR evita a inundação de *broadcasts*, auxiliando na estabilidade da rede.

2.2. Métricas de Roteamento

A métrica quantidade de saltos é normalmente utilizada em Redes *Ad Hoc* como padrão. Tal métrica é adequada a redes *Ad Hoc* pelo fato que novas rotas de uma rede devem

ser encontradas de forma rápida [Campista et al. 2008]. Porém, as redes *Wireless Mesh*, por possuírem uma topologia onde os nós formadores do *backbone* são normalmente fixos, uma rota com menor número de saltos pode não ser a melhor escolha. Partindo deste princípio, foram desenvolvidos algumas métricas de roteamento como alternativas da métrica de quantidade de saltos, tais implementações podem ser integradas aos protocolos de roteamento utilizados nas *Wireless Mesh Networks*.

A primeira alternativa à métrica quantidade de saltos proposta para as *Wireless Mesh Networks* (WMN) é a *Expected Transmission Count* (ETX) [Campista et al. 2008]. A métrica ETX mede de forma contínua a taxa de perda de ambos os sentidos entre cada nó e seus respectivos vizinhos, monitorando as taxas de perda dos enlaces através de troca de mensagens periódicas, assim como em enlaces alternativos para garantir o uso da melhor rota. Esta métrica calcula o peso da rota através da soma dos ETX's de cada enlace, que será utilizado pelo protocolo de roteamento para o cálculo da melhor rota.

A métrica *Expected Transmission Time* (ETT) [Bicket et al. 2005], foi desenvolvida como uma extensão da métrica ETX. A ETT considera a taxa de transmissão utilizada para realizar com precisão a qualidade dos enlaces. Seu objetivo é estimar o valor do atraso do canal, realizando a concatenação do ETX do enlace com a taxa de transmissão do nó.

Duas métricas alternativas a métrica ETX são: ML (*Minimum Loss*) [Passos and Albuquerque 2007] e MD (*Minimum Delay*) [Cordeiro et al. 2007]. A métrica ML objetiva a busca de caminhos com menores probabilidades de perda de pacotes mesmo que necessite utilizar um número maior de saltos que a métrica ETX. Já a métrica MD tem como critério o menor atraso de transmissão total para a construção de rotas entre pares comunicantes e seleção dos MPR's. Sendo assim, considera a menor soma dos atrasos originados de todos os enlaces envolvidos na rota.

Ambas as métricas ML e MD demonstraram um melhor desempenho e menores taxas de perda de pacotes quando comparadas a métrica ETX. É interessante destacar que a maioria das métricas de roteamento utilizam a métrica ETX, ou pequenas variações da mesma, para cálculo das tabelas de roteamento [Passos and Albuquerque 2007]. Seguindo este mesmo paradigma, a métrica FQ descrita na sub-seção 3.2 será uma variação da métrica ETX.

3. Fundamentação da Proposta

Esta seção fundamenta como a métrica *Factor-Q* foi desenvolvida. Descreve o comportamento da métrica ETX em sua forma original, após isto descreve as alterações realizadas na ETX para desenvolver a métrica *Factor-Q*, proposta e implementada neste documento.

3.1. A Métrica ETX

Para o cálculo da qualidade do enlace a métrica ETX utiliza o inverso do resultado gerado pelo produto do *Link* Direto (*forward delivery ratio(df)*) pelo *Link* Reverso (*reverse delivery ratio (dr)*), onde o *Link* Direto é responsável pelo envio dos pacotes *hello* e o *Link* Reverso é responsável pelos reconhecimentos positivos (ACKs) [Albuquerque et al. 2006]. Assim o ETX de um enlace $a \rightarrow b$ é definido como o inverso da probabilidade de transmissão com sucesso de um pacote através deste enlace como ilustra a equação 1.

$$ETX_{ab} = \frac{1}{P_{ab}} \tag{1}$$

O cálculo de uma rota com múltiplos saltos com a métrica ETX se dá pela soma do valor de ETX de cada salto. Sendo assim: em uma rota $a \rightarrow c$, será feita a soma do ETX do enlace $a \rightarrow b$ com ETX do enlace $b \rightarrow c$, como citado em [Passos and Albuquerque 2007], o ETX de uma rota $a \rightarrow n$ é definida por:

$$ETX_n = \sum_{i=0}^{n-1} \frac{1}{P_{a_i a_{i+1}}} \tag{2}$$

Onde $P_{a_i a_{i+1}}$ ilustra a probabilidade de transmissão com sucesso de um pacote entre os nós $a_i a_{i+1}$.

3.2. Métrica Proposta: *Factor-Q*

Para a criação da métrica *Factor-Q*, foi realizada uma alteração no cálculo original da métrica ETX. A métrica ETX como descrito na sub-seção 3.1 utiliza pacotes *hello* e reconhecimentos ACK's para o cálculo da qualidade do enlace. A *Factor-Q* segue o mesmo princípio da ETX, porém é adicionado ao Enlace Reverso (dr), a taxa de utilização da fila do enlace correspondente. A taxa de utilização da fila é cálculada através da divisão do tamanho da fila pela capacidade máxima da fila. Os valores do Enlace Direto (df), permanecem em sua forma natural. A equação 3 demonstra como a taxa de utilização da fila é calculada.

$$TxQ = queue\_ > length()/queue\_ > limit(); \tag{3}$$

Onde, $queue\_ > length()$ siginifica fila atual e $queue\_ > limit()$ corresponde ao total da fila. Feito o cálculo da taxa de utilização da fila, é necessário incluir o resultado deste cálculo na métrica ETX, formando assim a nova métrica *Factor-Q*. Ao atribuir o TxQ (Taxa da Fila) ao *Link* Reverso de ETX obtém-se a nova métrica *Factor-Q* como ilustra a expressão 4, onde P significa probabilidade.

$$FQ = \frac{1}{P_{(df \times (dr + TxQ))}} \tag{4}$$

Seguindo o mesmo princípio da métrica ETX, a equação 5 ilustra o cáculo da métrica em uma rota $A \rightarrow B$. Onde $P_{(ab)}$ denota a probabilidade de uma transmissão de um nodo origem a um nodo destino.

$$FQ_{ab} = \frac{1}{P_{(ab)}} \tag{5}$$

Para o cálculo de uma rota com múltiplos saltos a métrica *Factor-Q* realiza o somatório dos valores de FQ de cada enlace a fim de obter o custo total de cada rota. A equação 6 denota este cálculo.

$$FQ_n = \sum_{i=0}^{n-1} \frac{1}{P_{(a_i a_{i+1})}} \quad (6)$$

A métrica FQ demonstra um bom desempenho nas simulações realizadas, estes melhores resultados são decorrentes da utilização da taxa da fila, pois quando o enlace apresentar altas taxas de utilização de fila a métrica FQ irá retornar um peso maior para o atual enlace, forçando o protocolo de roteamento a escolha de uma rota alternativa.

Esta seção descreveu a metodologia utilizada para a implementação da métrica *Factor-Queue*, que além dos pacotes *hello default* utilizados na métrica ETX, também adiciona a seu cálculo, a taxa de utilização de fila dos nós vizinhos, visando retornar uma estimativa mais apurada para que o protocolo de roteamento direcione o tráfego de melhor forma possível.

4. Cenário de Simulação

Para as simulações foi adotado um cenário real, o campus da PUC-PR (Figura 1), que é composto por vários blocos acadêmicos e áreas de estacionamento entre os blocos. Com o objetivo de avaliar o comportamento da métrica FQ, as simulações foram executadas no *Network Simulator 2* [NS2 2010], utilizando extensões para o OLSR desenvolvidas para o NS-2 [Cordeiro et al. 2007].

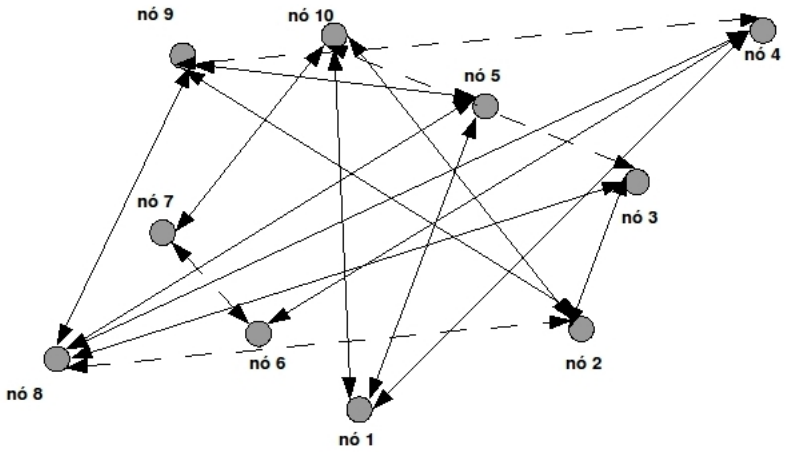


Figura 1. Campus PUC-PR

O tráfego foi gerado através de transmissões *VoIP* (UDP) e FTP (TCP), para extração dos dados foram realizadas 1000 simulações com diferentes sementes. As simulações foram compostas por 15 chamadas *VoIP* que representam 30 fluxos *VoIP*, juntamente com tráfego de *background* FTP. O tempo de simulação foi definido em 50 segundos seguindo o mesmo paradigma demonstrado em [Aguiar et al. 2007] e [Cordeiro et al. 2007].

Foi utilizado como protocolo de roteamento para as simulações o OLSR, amplamente utilizado em redes *Mesh*. Para comparar os resultados obtidos pela métrica *Factor-Q*, apresentada neste documento, foi utilizada a métrica *Minimum Loss* (ML) [Passos and Albuquerque 2007], *Minimum Delay* (MD) [Cordeiro et al. 2007] e *Expected Transmission Count* (ETX) [Campista et al. 2008] no mesmo cenário de simulação.

Tabela 1. Parâmetros de Simulação

Parâmetros	Valores
Métricas	FQ, ML, MD, ETX
Tempo de Simulação	50 Segundos
Padrão Utilizado	IEEE 802.11g
Modelo de Propagação	Shadowing
Modelo das Antenas	Omnidirecional, 18dB de ganho
Path Loss Exponent	2,7
Shadowing deviation	4.0dB
Área de Simulação	1000m x 1000m

Tabela 2. Distribuição dos Nós pelo Cenário

Nome do Nó	**Eixo** x	**Eixo** y	**Nome do Nó**	**Eixo** x	**Eixo** y
1. CTHC	160,00	485,00	6. CCET	628,00	320,00
2. Biblioteca Central	305,00	277,00	7. CCBS	570,00	440,00
3. Administração Central	340,00	226,00	8. CCJS	780,00	480,00
4. Quadras Poliesportivas	270,00	32,00	9. Parque Tecnológico	918,00	597,00
5. Bloco Acadêmico	476,00	200,00	10. PPGIA	968,00	550,00

A Figura 1 ilustra o campus da PUC-PR com os roteadores *Mesh* sendo representados pelos círculos cinzas, as linhas contínuas indicam as chamadas *VoIP* e as linhas tracejadas indicam o tráfego de *background*. Os blocos foram numerados de 1 a 10, ficando: 1-CTHC, 2-Biblioteca Central, 3-Administração Central, 4-Quadras Poliesportivas, 5-Bloco Acadêmico, 6-CCET, 7-CCBS, 8-CCJS, 9-Parque Tecnológico e 10-PPGIA. A tabela 2 descreve as localizações dos nós pelo cenário de simulação conforme figura 1, a área de simulação é de $1000m^2$ conforme tabela 1.

Cada chamada *VoIP* é composta por dois fluxos, pois a aplicação tem fluxo bidirecional, sendo assim os fluxos de ida e volta não trafegam pelos mesmos pontos. O tráfego de *background* (FTP) foi gerado através do Modelo de Pareto [NS2 2010], para caracterizar tráfego em rajadas, com valores *default*. O *codec* utilizado para as simulações foi o G.729, pois seu consumo de banda é de 8 Kbps, desta forma é o mais utilizado nas redes sem fio [Cordeiro et al. 2007]. A tabela 1 demonstra os parâmetros da simulação.

Para análise dos resultados foi utilizado o intervalo de confiança de 90% calculado conforme [Jain 1991]. Os valores escolhidos para avaliação dos resultados foram: *jitter*, atraso, vazão e probabilidade de bloqueio.

Esta seção descreveu o cenário utilizado para realização das simulações, a escolha pelo Campus da Pontíficia Unversidade Católica do Paraná - Curitiba, se dá pelo fato de ser um ambiente acadêmico, onde é possível a implementação de uma rede *Wireless Mesh* real para utilização das tecnologias abordadas neste documento.

5. Resultados Obtidos

Para análise dos resultados a métrica FQ foi comparada com as seguintes métricas: ETX, ML e MD, no mesmo cenário de simulação. Foram realizados 30 Fluxos VoIP ao total representando 15 chamadas *VoIP* (UDP), e simultâneamente foram realizadas transmissões de dados (TCP). Os resultados são demonstrados a seguir. A figura 2 ilustra os resultados de atraso para os 30 fluxos *VoIP* (2 fluxos por chamada) obtidos nas simulações.

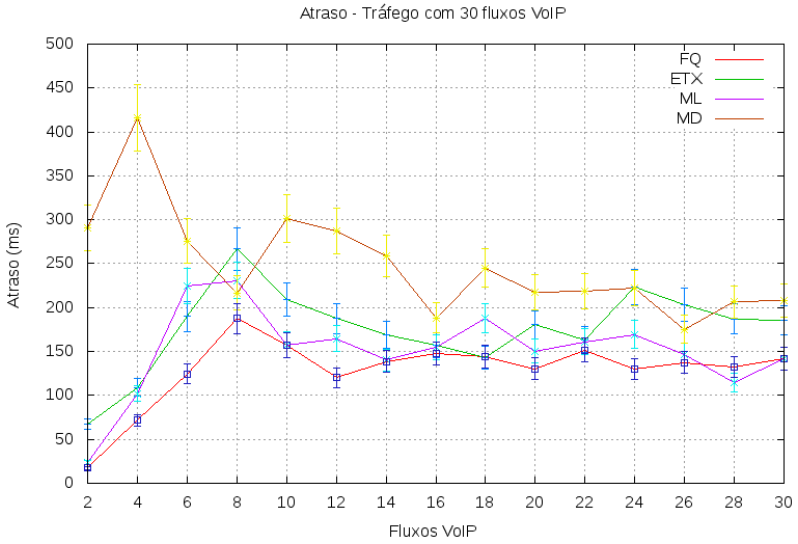


Figura 2. Atraso *Factor-Q*

A métrica *Factor-Q* demonstra manter as taxas de atraso mais baixas que as métricas concorrentes, demostrando um melhor desempenho durante o período de simulação. Manter o atraso na entrega dos dados é de suma importância em qualquer rede, porém quando o tráfego em questão é de voz, essa tarefa torna-se mais desafiadora e a métrica FQ supriu este desafio não ultrapassando os 187ms, quando comparada as demais métricas concorrentes.

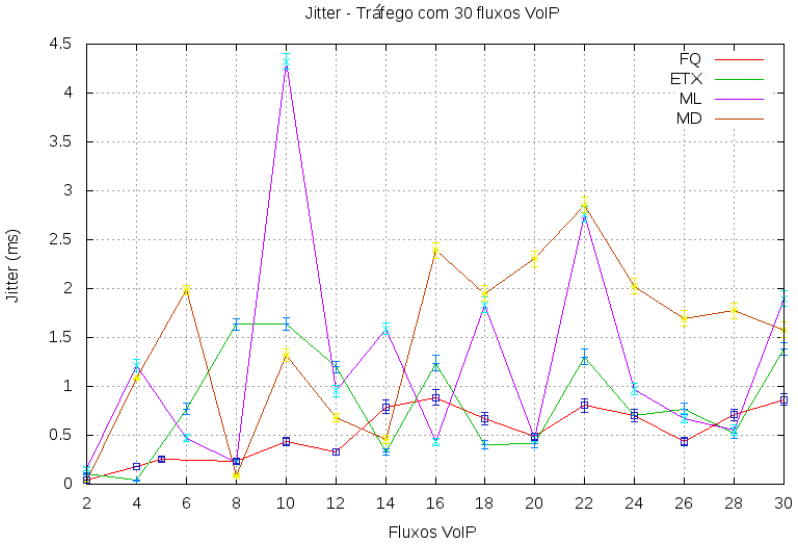


Figura 3. *Jitter Factor-Q*

Os resultados para *jitter* ilustrados na figura 3, demonstram o comportamento estável da métrica FQ, consequência dos menores atrasos obtidos com esta métrica. Tanto o atraso quanto o *jitter* diferem para fluxos da mesma chamada, isto ocorre pelo fato dos fluxos tomarem rotas diferentes, devido à interferências dos outros nós.

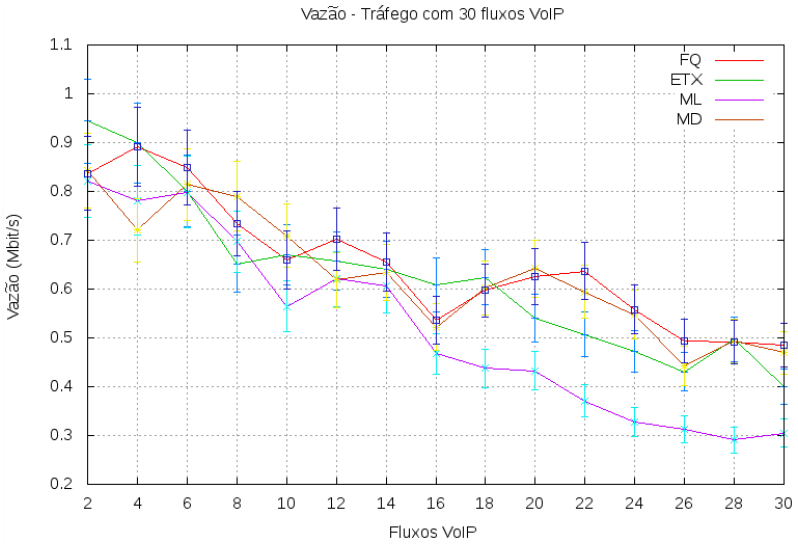


Figura 4. Vazão *Factor-Q*

A figura 4 apresenta os resultados de vazão. Observou-se que a métrica FQ obteve constantemente melhor comportamento perante a métrica ML, e com comportamento muito próximo das demais métricas analizadas. A distância entre os nós influencia na vazão dos dados, pois quanto menor a distância entre os nós vizinhos maior é a vazão. Os resultados relacionados à perda de pacotes ilustrados na figura 5, a métrica FQ demostrou novamente menores resultados em relação a ML, e manteve um bom comportamento no quadro geral de simulação.

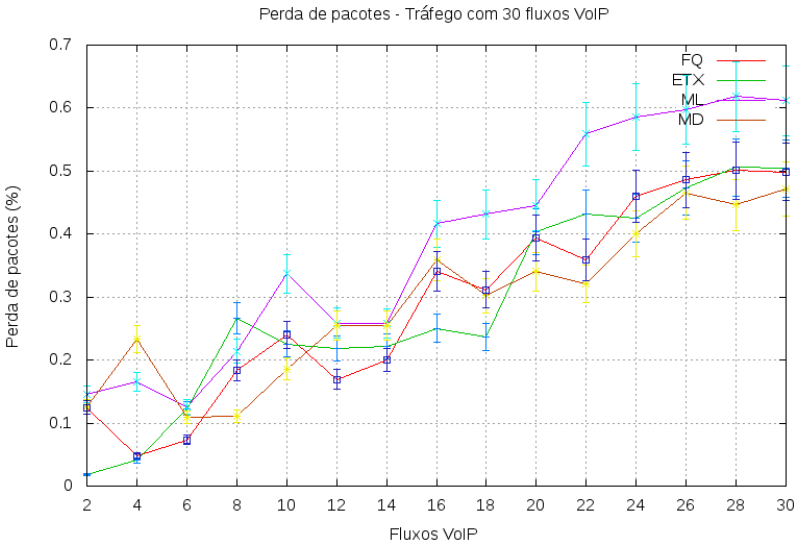


Figura 5. Perda de Pacotes *Factor-Q*

Através da análise dos gráficos apresentados, conclui-se que o comportamento

da métrica FQ atingiu melhores resultados que as métricas ML, MD, ETX, com relação ao atraso. E no geral o comportamento da FQ foi relevante, apresentando resultados significativos no cenário utilizado. Estes resultados decorrem do fato da métrica *Factor-Q* utilizar a taxa de utilização da fila juntamente com os pacotes *hello* para o cálculo do peso de cada enlace da rede. Desta forma a métrica FQ demonstra ser uma alternativa viável de utilização em redes *Wireless Mesh* de baixa mobilidade, com alto tráfego de dados *VoIP*.

6. Conclusão do Artigo

O crescimento das redes sem fio do tipo *Wireless Mesh Networks*, desencadeia a necessidade de novas tecnologias para estes tipo de rede. O tema *Wireless Mesh Networks* é complexo, desta forma encontra-se em processo constante de pesquisa. Desta maneira as redes sem fio do tipo *Mesh* podem desenvolver um maior potencial com relação a serviços oferecidos e desempenho.

Este artigo objetivou abordar e discutir as métricas de roteamento que constituem uma das diversas áreas de pesquisa dentro do tema redes *Mesh*. As métricas de roteamento são fundamentais em redes *Mesh* e *Ad-Hoc*, pois, seus enlaces e rotas necessitam estar em processo constante de avaliação, porém, interferindo o mínimo possível no desempenho da rede. Quando a rede *Mesh* dispõem de tráfego *VoIP* juntamente com tráfego TCP, o tema métricas de roteamento torna-se ainda mais desafiador.

Este trabalho apresentou uma nova métrica de roteamento denominada FQ (*Factor-Q*), que aprimora o cálculo de rotas nas redes *Mesh* com tráfego *VoIP*. A métrica FQ utiliza como base o cálculo de probabilidades de transmissões feito pela métrica ETX, porém seu grande diferencial é adicionar ao cálculo original do ETX a taxa de utilização da fila para cálculo das rotas. Desta forma quando o enlace estiver com a taxa de utilização da fila elevada, a métrica FQ irá retornar ao protocolo de roteamento um peso maior para esta rota, forçando a busca de rotas alternativas, promovendo uma distribuição da carga de comunicação entre rotas alternativas e menos congestionadas por utilização ou por contenção de enlaces vizinhos.

As simulações demonstraram que a métrica FQ obteve melhores resultados perante as métricas ML, MD, ETX, com relação a delay, mantendo baixa a taxa de atraso (Delay) durante todo o período de simulação não ultrapassando os 187ms. Menores taxas de atraso (delay) são indispensáveis quando a rede possui tráfego VoIP. As demais medidas da métrica FQ demonstraram a estabilidade da métrica. Sendo assim a FQ é uma alternativa viável para utilização em redes *Wireless Mesh* com tráfego *VoIP*.

Na sequência das pesquisas a métrica FQ poderá ser utilizada em redes com maior número de nós e tráfego mais intenso, com intuito a testar seu comportamento perante outras métricas de roteamento existentes na literatura em diferentes circunstâncias.

Referências

Aguiar, E., Bittencourt, P., Moreira, W., and Abelém, A. (2007). Estudo comparativo de protocolos de roteamento para redes mesh na regiao amazônica. *XXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC - Sessão de Artigos Curtos II*.

Akyildiz, I., Wang, X., and Wang, W. (2005). Wireless Mesh Networks: a survey. In *Computer Networks and ISDN Systems*, pages 445–487.

Albuquerque, C. V. N., Saade, D. C. M., Passos, D. G., Teixeira, D. V., Leite, J., Neves, L. E., and Magalhães, L. C. S. (2006). Gt-Mesh - Rede Mesh de Acesso Universitário Faixa Larga Sem Fio - Relatório Técnico 3. (RT-3 1-118).

Bicket, J., Aguayo, D., Biswas, S., and Morris, R. (2005). Architecture and evaluation of an unplanned 802.11 b mesh network. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 31–42. ACM New York, NY, USA.

Campista, M., Esposito, P., Moraes, I., Costa, L., Duarte, O., Passos, D., de Albuquerque, C., Saade, D., and Rubinstein, M. (2008). Routing metrics and protocols for wireless mesh networks. *IEEE network*, 22(1):6.

Clausen, T. and Jacquet, P. (2003). RFC3626: Optimized Link State Routing Protocol (OLSR). *RFC Editor United States*.

Cordeiro, W., Aguiar, E., Abélem, A., and Stanton, M. (2007). Providing Quality of Service for Mesh Networks Using Link Delay Measurements. *Proceedings of 16th International Conference on Computer Communications and Networks, p.991-996*.

Gnuplot (2010). Gnuplot, Home Page, http://www.gnuplot.info/.

Jain, R. (1991). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley New York.

Johnson, D., Maltz, D., Hu, Y., and Jetcheva, J. (2003). The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). *IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress*, 15.

NS2 (2010). Network Simulator-NS2, Home Page, http://www.isi.edu/nsnam/ns.

Passos, D. and Albuquerque, C. (2007). Proposta, Implementaçao e Analise de uma Metrica de Roteamento Multiplicativa para Redes em Malha Sem Fio. *Anais do XXVII Congresso da SBC*, pages 1935–1944.

Perkins, C., Belding-Royer, E., and Das, S. (2003). IETF RFC 3561, Ad hoc ondemand distance vector (AODV) routing [S].

ReMesh (2005). Universidade Federal de Fluminense. 2005. Disponível em: http://mesh.ic.uff.br.

Saade, D., Albuquerque, C., Magalhães, L., Passos, D., Duarte, J., and Valle, R. (2007). Redes em Malha: Solução de Baixo Custo para Popularização do Acesso à Internet no Brasil. *XXV SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÓES - SBrT*.

Tsarmpopoulos, N., Kalavros, I., and Lalis, S. (2005). A low-cost and simple-to-deploy peer-to-peer wireless network based on open source linux routers. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, pages 92–97.

Avaliando a Eficácia das Técnicas de Estimativa de Capacidade de Caminho em Redes com Enlaces WiMAX

Alex A. de Oliveira[1], Sidney C. de Lucena[1], Carlos A. V. Campos[1], Antônio A. de A. Rocha[2]

[1]Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

[2]Departamento de Ciência da Computação
Universidade Federal Fluminense (UFF)

{alex.oliveira,sidney,beto}@uniriotec.br, arocha@ic.uff.br

Resumo. *A métrica "capacidade máxima de transmissão de um caminho" representa a menor taxa de transmissão alcançada dentre todos os enlaces do caminho de rede medido. Este artigo analisa a eficácia das técnicas mais utilizadas para a estimativa desta métrica quando há no caminho medido pelo menos um enlace WiMAX. Para tal, ferramentas que usam o estado da arte das técnicas de estimativa de capacidade foram avaliadas em um ambiente experimental com enlace WiMAX. Os resultados obtidos demonstram que tais técnicas podem severamente falhar nas suas estimativas, especialmente se algumas propriedades específicas do padrão são conjuntamente habilitadas no enlace sem fio.*

Abstract. *The metric called "maximum transmission capacity of a path" represents the narrowest transmission rate available among all the links in a measured network path. This paper analyzes the effectiveness of some existing techniques for estimating this metric when applied to a path doted by at least one WiMAX link. For the evaluation, we consider state-of-art techniques for capacity estimation applied to an experimental environment with a WiMAX link. Obtained results show that those techniques can fail severally in their estimates, especially if some specific properties of the standard are jointly enabled in the wireless link.*

1. Introdução

O padrão IEEE 802.16 [IEEE Std 802.16-2003], também conhecido como WiMAX (do termo em inglês, *Worldwide Interoperability for Microwave Access*), consiste numa especificação para redes sem fio em grandes áreas (*Wireless Metropolitan Area Network* - WMAN). Ele surge como uma solução para o problema de acesso banda larga sem fio de última milha, oferecendo conexão a provedores de serviço de comunicação através de ligações ponto-a-ponto ou ponto-multiponto para curtas e longas distâncias. Trata-se de um modelo de transmissão para enlaces sem fio, com ou sem mobilidade, para a conexão de residências, corporações e outros tipos de redes, sendo também utilizado como solução para enlaces de rádio ponto-a-ponto em redes de núcleo. Esta tecnologia permite o atendimento de uma grande quantidade de usuários e o fornecimento de

alguns dos principais requisitos para as aplicações atuais, tais como elevada largura de banda, maior segurança e garantias de qualidade de serviço. Comparada a outras tecnologias de acesso (fibra óptica, *cable-modem* ou linhas DSL), o padrão WiMAX pode ser uma alternativa de menor custo uma vez que não se torna necessária infraestrutura de cabeamento e sua característica de cobertura o torna, em alguns casos, uma solução mais rápida para a implantação de acesso banda larga em larga escala. Devido a estas facilidades, o WiMAX surge cada vez mais como uma alternativa para centros urbanos densamente povoados.

Conhecer as características de desempenho dos diversos caminhos de rede por onde trafegam os fluxos de dados pode ser fundamental para o desenvolvimento de mecanismos que possibilitem melhorar a qualidade do funcionamento de aplicações distribuídas. Medir tais características para os caminhos de rede é igualmente importante para as atividades de gerenciamento, como o diagnóstico de problemas de desempenho, a verificação de contratos de qualidade de serviço e o planejamento de capacidade. Dentre as medidas de desempenho mais relevantes, como perda, atraso e *jitter*, a capacidade de transmissão efetiva do caminho [Prasad 1997] é de especial importância para certas aplicações distribuídas, como no caso da seleção de *peers* em redes P2P. No entanto, existem alguns fatores importantes que devem ser considerados quanto à obtenção dessas medidas, tais como: (i) assumir a inexistência de ajuda dos equipamentos das redes ao longo do caminho, assim como o desconhecimento das características dos enlaces e da topologia; (ii) utilizar o mínimo de tráfego e processamento adicional para a obtenção das medidas (i.e., ter um baixo custo operacional); e (iii) haver rapidez na obtenção das características desejadas, de forma que a aplicação possa usufruir dos resultados.

O presente trabalho faz uma análise das técnicas mais utilizadas para estimativa da capacidade de transmissão efetiva em caminhos de rede que possuam um ou mais enlaces WiMAX. A capacidade de transmissão efetiva, também chamada de capacidade ou *throughput* máximo do caminho, representa a menor taxa de transmissão alcançável dentre todos os enlaces de um caminho de rede medido. A análise aqui apresentada permite avaliar o quão (in)eficaz é a estimativa de capacidade de alguns métodos considerados estado da arte, como o Pares de Pacotes e o Trens de Pacotes [Johnsson 2004], quando aplicados a caminhos com enlaces sem fio IEEE 802.16. Os resultados apresentados demonstram os problemas e limitações das técnicas utilizadas quando aplicadas a este cenário. No limite do nosso conhecimento, este é o primeiro trabalho que investiga a eficácia destas técnicas quando aplicadas a um caminho de rede que possua ao menos um enlace WiMAX e, como contribuição, são identificados os respectivos problemas ao se usar as ferramentas mais populares para estimativa de capacidade efetiva.

A continuação deste trabalho está organizada da seguinte forma: na Seção 2 é feita uma breve revisão do padrão IEEE 802.16; na Seção 3 são descritos os métodos tradicionais de pares de pacotes e trem de pacotes, além de alguns trabalhos relacionados aos problemas gerais do uso do mecanismo de pares (e trens) de pacotes para obtenção de medidas em enlaces sem fio; na Seção 4 é apresentado o ambiente desenvolvido e detalhes da avaliação experimental desenvolvida neste trabalho; a Seção 5 é dedicada à apresentação dos resultados obtidos; e, finalmente, na Seção 6 são apresentadas as conclusões e os trabalhos futuros.

2. O Padrão IEEE 802.16

Conforme descrito na introdução, o padrão IEEE 802.16 foi desenvolvido para o acesso a banda larga sem fio em regiões metropolitanas [IEEE Std 802.16-2003]. Para esta razão, a especificação da pilha de protocolos a divide em duas camadas: a camada física e a camada de acesso ao meio (MAC). A camada MAC, por sua vez, é dividida em três subcamadas: (i) a subcamada de segurança (*Privacy Sublayer*), (ii) a subcamada de convergência comum MAC (*MAC CPS – Common Part Sublayer*) e (iii) a subcamada de convergência de serviços específicos (*CS – Service Specific Convergence Sublayer*). A camada física define funções específicas de transmissão, como o tipo de modulação, a codificação e a faixa de frequência, enquanto que a camada MAC controla e distribui os recursos do espectro de frequência entre a estação base (*Base Station,* ou BS) e as estações clientes (*Subscriber Station*, ou SS).

Na camada física, o padrão IEEE 802.16 define o uso de OFDM (*Ortogonal Frequency Division Multiplexing*) como técnica de multiplexação, operando na faixa de 2 a 11 GHz. Para contornar problemas durante a transmissão do sinal (como ruído, interferência, desvanecimento ou perdas), o padrão prevê o uso de modulação adaptativa com o intuito de adequar a taxa de dados conforme as características do meio. Por exemplo, é possível usar a modulação 64 QAM (*Quadrature Amplitude Modulation*) para transmissão de altas taxas, porém mais sensível a interferências, ou diminuir a taxa usando esquemas de modulação menos sensíveis, como a 16 QAM.

Duas importantes características da camada MAC, de caráter opcional, são as funções de Concatenação e Burst. Ambas têm como objetivo elevar o *throughput* da interface aérea do rádio. A função de Concatenação, quando habilitada, permite a junção do *payload* de vários quadros pequenos no *buffer* de transmissão num único quadro maior para transmissão via rádio, de acordo com um limite de tamanho máximo pré-estabelecido. O quadro concatenado, ao chegar no receptor, tem os *payloads* originais novamente restaurados para serem roteados. Além de aumentar o *troughput* devido à remoção do intervalo de tempo entre os quadros originais, chamado de intervalo de contenção, esta função também reduz o *overhead* pela eliminação dos cabeçalhos dos quadros originais, ficando apenas o do quadro resultante. Na BS, o processo de concatenação ocorre de forma separada e independente para cada SS de destino. A função Burst, quando habilitada, apenas minimiza o intervalo de contenção entre os quadros no *buffer* de transmissão ao serem enviados, fazendo com que os mesmos sejam transmitidos em rajada, praticamente sem intervalo de tempo entre eles. Esta função se aplica apenas a quadros *unicast* e é limitada pela configuração da duração máxima de rajada. Ambas as funções, Concatenação e Burst, podem ser simultaneamente ativadas.

A diferença básica entre rádios Pré-WiMAX e WiMAX é que o primeiro opera em frequências não licenciadas. Além disso, rádios Pré-WiMAX de alguns fabricantes não implementam algumas características optativas, como a correção de erros (FEC) e a confirmação (ACK). Neste artigo, foram utilizados rádios do tipo Pré-WiMAX.

3. Estimativa de Capacidade Máxima de Caminho

Diversos métodos foram propostos na literatura para a estimativa de métricas relacionadas com capacidade. Dentre os métodos mais conhecidos estão: (i) *One-packet*, implementado pelas ferramentas Pathchar e Clink, que tem como objetivo estimar a taxa

de transmissão de todos os enlaces presentes no caminho de rede medido [Downey 1999]; (ii) *Mult-packet*, uma variação da técnica One-packet desenvolvida por Lai e Baker em [Lai e Baker 2000], que também tem como finalidade estimar a capacidade de transmissão dos enlaces de um caminho; (iii) Pares de pacotes (ou *packet-pairs*), que é amplamente utilizado na literatura para estimar, dentre outras medidas, a capacidade de contenção (ou capacidade máxima de transmissão) do caminho, e é implementada em ferramentas como CapProbe [Kapoor 2004] e Tangram-II [Rocha 2009]; e, (iv) Trem de pacotes (ou *packet-train*), que é uma extensão da técnica de Pares de pacotes, desenvolvida por Dovrolis et al. em [Dovrolis 2001], e é utilizada por ferramentas como Pathrate e Pathload para medir, respectivamente, a capacidade de contenção e a largura de banda disponível em um caminho de rede [Jacob 2003].

Descrições mais detalhadas sobre o funcionamento de cada um dos métodos citados acima, assim como de algumas das métricas citadas, podem ser encontrados em diversos trabalhos da literatura (ver [Ziviani 2005]). O foco do presente trabalho será apenas para o método de pares de pacotes e para a sua variação, o método de trem de pacotes, pois são as técnicas apropriadas para a obtenção da medida de interesse avaliada neste trabalho: capacidade máxima de transmissão de um caminho de rede.

3.1. Pares de Pacotes

O método de pares de pacotes consiste na emissão de dois pacotes de mesmo tamanho e de uma mesma origem, separados por um intervalo de tempo bem próximo de zero. Os pacotes atravessam o mesmo caminho na rede até chegarem a um único destino, onde são coletados. A partir da coleta destes pacotes é possível identificar algumas características do caminho de rede atravessado pelo par, como a capacidade de contenção.

A suposição principal da técnica é que a dispersão entre os pacotes do par, identificada na coleta, é causada pela menor capacidade de transmissão ao longo do caminho (denominada, capacidade de contenção). Os pacotes, que são gerados de uma mesma origem e separados por intervalos de tempo bem próximos de zero, possuem o espaçamento entre eles mantido até que passem por um enlace com capacidade de transmissão inferior à do emissor. Essa dispersão, causada pelo tempo de transmissão deste enlace (superior aos tempos experimentados nos enlaces anteriores) é mantida até o destino dos pacotes, a menos que seja encontrado, ao longo do restante do caminho, outro enlace com uma capacidade ainda menor. Seja T o intervalo de tempo entre as chegadas dos dois pacotes dado em segundos, e seja B o tamanho dos pacotes dado em bits, capacidade de contenção $C = B/T$ bits/s.

Avaliações feitas do método de pares de pacotes, como os apresentados em [Dovrolis 2001, Carter 1996, Roesler 2003, Augusto 2003 e Rocha 2004], demonstram que uma situação de tráfego concorrente alto pode influenciar as estimativas e ocasionar erros nos resultados obtidos. A influência causada pelo tráfego concorrente pode ser caracterizada de duas formas: (i) a presença de pacotes em frente aos pares na fila dos roteadores, após já terem passado pelo nó de contenção do caminho, pode ocasionar uma redução na dispersão existente entre os pacotes. Como conseqüência, a capacidade de contenção é superestimada; (ii) a inserção de tráfego concorrente entre os dois

pacotes do par. Este evento pode resultar em um acréscimo da dispersão dos pacotes e causar uma estimativa inferior à capacidade real de transmissão do enlace de contenção.

Para reduzir a influência do tráfego concorrente, uma série de pares de pacotes pode ser utilizada e, então, gerado um histograma das capacidades estimadas por todos os pares. No caso, a capacidade de contenção estimada equivalerá àquela de maior frequência no histograma resultante. No entanto, avaliações apresentadas em [Rocha 2004] e em [Kapoor 2004] demonstram que, mesmo com o uso de diversos pares de pacotes, a precisão das medidas ainda sofre grande influência da condição da rede.

Em [Rocha 2004] é apresentada uma variação da técnica de pares de pacotes, na qual apenas pares selecionados são utilizados para computar a capacidade de contenção. A ferramenta CapProbe, apresentada em [Kapoor 2004], também propõe uma seleção dos pares de pacotes utilizados para computar a capacidade de contenção baseada no atraso sofrido por estas sondas. Uma diferença fundamental entre as duas técnicas é que a primeira refere-se à respectiva métrica de caminho em apenas um sentido, enquanto que a segunda mede a capacidade de contenção no caminho de ida-e-volta.

3.2. Trem de Pacotes

O método baseado em trens de pacotes é derivado da técnica de pares de pacotes. Ao invés de apenas sequências de pares, neste caso são usadas sequências com rajadas de pacotes. A idéia é enviar uma rajada de $L > 2$ pacotes, todos de tamanho igual a S, e determinar a dispersão na recepção desses pacotes. Essa dispersão, denotada por t, é determinada pelo intervalo de tempo entre o último bit do primeiro pacote e o último bit do último pacote.

A partir de uma sequência de pacotes coletados é possível determinar uma medida de taxa de dispersão assintótica. Denotada por R, essa taxa de dispersão assintótica é dada por $R = (L - 1) S / t$, onde L é o número de pacotes da rajada, S é o tamanho desses pacotes e t é o intervalo de tempo entre a chegada do primeiro e do último pacote do trem. Se a taxa de transmissão do trem de pacotes for superior à largura de banda disponível do caminho medido, então o valor dessa medida será equivalente ao valor computado pela taxa de dispersão assintótica. No entanto, se a taxa de transmissão do trem de pacotes for inferior à largura de banda disponível, então a medida obtida refere-se à capacidade de contenção do caminho medido.

Comparada à técnica de pares de pacotes, a técnica de trem de pacotes é mais robusta e menos sensível a erros. No entanto, ela é mais intrusiva, pois exige uma quantidade maior de pacotes a serem enviados pela rede. Além disso, a precisão da métrica ainda é influenciada pelo tráfego concorrente existente no caminho medido. Esta técnica é utilizada por ferramentas como Pathload e Pathrate [Jacobson 1999] para estimar, respectivamente, as métricas largura de banda disponível e capacidade de contenção.

3.3. Problemas com a Técnica de Pares de Pacotes no Padrão 802.16

Em trabalhos anteriores já foram utilizadas técnicas de medições com pares de pacotes para estimar algumas métricas relacionadas a capacidades em caminhos onde existem enlaces sem fio. Por exemplo, em [Kapor 2004] foram executadas medições para estimar a capacidade de contenção de caminho nas quais o enlace de menor

capacidade estava numa rede local sem fio. Já em [Rocha 2007], foi apresentada uma técnica para estimar a taxa de transmissão em uma rede local sem fio.

Apesar da existência de trabalhos na literatura relacionados à medição de capacidade de enlaces sem fio, ainda pouco se conhece do desempenho destas técnicas em redes com enlaces de tipo WiMAX. Para isto, alguns aspectos relacionados com características próprias do padrão 802.16 devem ser considerados para a estimativa de métricas de capacidade.

As funcionalidades de Concatenação e Burst da camada MAC do padrão IEEE 802.16, geralmente ativadas por *default* em rádios WiMAX e pré-WiMAX por elevarem o *throughput*, podem interferir severamente em estimativas de capacidade máxima de caminho que usem a técnica de pares de pacotes. Isto ocorre porque tais funcionalidades acabam por eliminar o intervalo de contenção entre alguns quadros transmitidos, de acordo com o distanciamento no tempo e tamanho dos mesmos. Caso isto ocorra com o intervalo entre dois pacotes de medição (por exemplo, um par de pacotes de sonda), elimina-se a dispersão de tempo entre os quadros que até então caracterizaria a menor capacidade encontrada no caminho.

A concatenação tem influência sobre quadros de tamanho pequeno no *buffer* de transmissão, uma vez que estes conseguem ser concatenados em quadros maiores e enviados pelo meio sem que haja intervalo de tempo entre eles. Conforme já mencionado, o tamanho de um quadro concatenado (ou seja, resultante de concatenação) é limitado a um valor máximo, que pode ser configurável ou definido pelo fabricante do rádio. O Burst tem o mesmo tipo de influência, independente do tamanho do quadro, uma vez que remove os intervalos de contenção entre todos os quadros no *buffer* de transmissão que conseguem ser enviados até um tempo limite, denominado duração máxima da rajada. Este parâmetro é configurável e pode assumir valores que geralmente são múltiplos de 1 ms até um limite máximo.

Para quaisquer dessas funções que estejam ativadas, juntas ou não, a consequência para uma sequência de pacotes de medição que se enquadre nas situações descritas é um aumento no valor estimado pela técnica de pares de pacotes, ou mesmo pela de trem de pacotes.

4. Avaliação Experimental

De maneira a possibilitar uma avaliação experimental das limitações impostas pelo padrão IEEE 802.16 para a estimativa de capacidade máxima de caminho, foi montado um cenário contendo um enlace de rádio WiMAX interconectando duas redes. Apesar de se tratar de uma topologia ponto-a-ponto, os rádios se comportavam como se estivessem numa topologia ponto-multiponto, com BS e SS operando em modo *half-duplex* usando TDD (*Time Division Duplexing*).

4.1. Especificações

Foram usados dois rádios pré-WIMAX BreezeAccess VL do fabricante Alvarion, sendo o modelo AU-E-AS-5.8VL usado para estação base (BS) e o modelo SU-A-5.8-24BD-VL para estação do assinante (SS). Ambos suportam uma taxa de transmissão no nível físico de até 54 Mbps na faixa de frequência de 5.8GHz. O software de gerência proprietário da Alvarion, chamado BreezeConfig [Breeze 2010],

foi utilizado na configuração e no monitoramento da quantidade de pacotes que atravessam o enlace. Os rádios foram colocados em ambiente fechado e foram usadas antenas direcionais dispostas em linha de visada, com uma distância de 4 metros entre elas. A potência dos rádios foi regulada de maneira a evitar problemas com reflexão e a modulação adaptativa foi desabilitada (fixada em 64 QAM) para evitar mudanças de capacidade durante os experimentos.

O cenário de experimentação usado é mostrado na Figura 1. Dois computadores foram ligados nas portas Ethernet de cada rádio, totalizando quatro. Um computador servia para alteração de configuração dos rádios e verificação dos contadores de pacotes no enlace sem fio, através do BreezeConfig. No segundo computador foram colocadas as ferramentas usadas para a estimativa de capacidade.

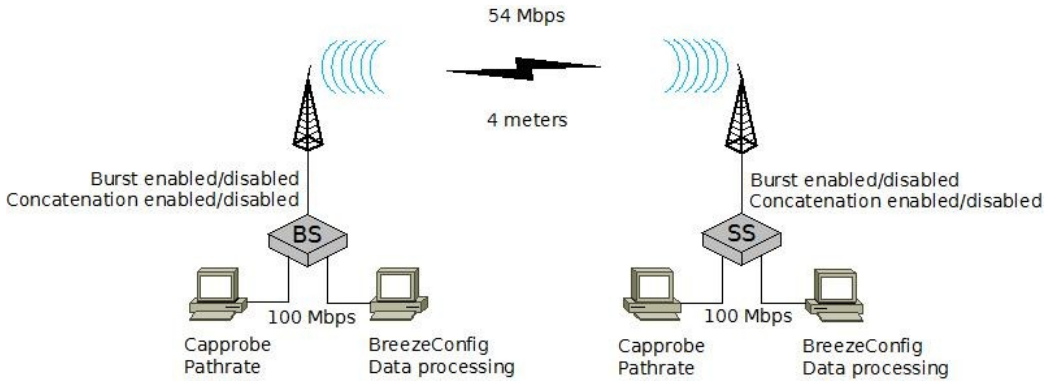


Figura 1. Cenário dos experimentos

4.2. Ferramentas Usadas

Para se obter a estimativa de capacidade máxima de caminho e avaliar o uso da técnica de pares de pacotes em redes com enlaces IEEE 802.16, foram utilizadas as ferramentas Pathrate [Jacobson 1999] e CapProbe [Kapoor 2004]. Ambas as ferramentas são amplamente conhecidas e baseadas no uso de pares de pacotes, porém cada qual com uma abordagem diferente.

O funcionamento do Pathrate pode ser dividido em três fases distintas. Na primeira, ele detecta o maior tamanho que um trem de pacotes pode ter de forma a ser transferido sem perdas. O coeficiente de variação da dispersão entre os pares de pacotes é medido e, se este valor for menor que um dado limiar, o Pathrate termina a primeira fase já com o resultado final. Quando isto não ocorre, o Pathrate entra na segunda fase e envia 1000 pares de pacotes com tamanhos variados, entre 550 bytes e o maior MTU encontrado na fase inicial. Ao final da segunda fase, as estimativas de capacidade obtidas são agrupadas em faixas para que, na terceira e última fase, o Pathrate realize uma análise estatística e envie 500 trens de pacotes com o máximo MTU encontrado e o maior número de pacotes possíveis em cada trem. O Pathrate fornece a estimativa de capacidade máxima após o término da terceira fase, indicando uma faixa de valores para esta medida.

Diferentemente do Pathrate, que usa uma abordagem cliente-servidor para realizar estimativas unidirecionais, o CapProbe é uma ferramenta não-cooperativa que se vale de *echo requests* e *responses* do ICMP, direcionados a um IP de destino, para

enviar pares de pacotes e monitorar a dispersão de ida e volta no caminho. Isto significa que o CapProbe fornece estimativas bidirecionais, informando a menor capacidade "unidirecional" dentre todas ao longo do caminho de ida e volta. De maneira a evitar distorções na medida causadas pela presença de tráfego cruzado e enfileiramentos ao longo do caminho (ida e volta), o CapProbe monitora os atrasos dos pacotes e usa o chamado *minimum delay sum*, que é a menor soma dos atrasos dos pacotes de um par, para selecionar o par que não sofreu ou que menos sofreu influência de tráfego cruzado. A dispersão deste par é então usada para a estimativa de capacidade.

4.3. Influência da subcamada MAC CPS nos Resultados das Ferramentas

Conforme mostrado a seguir na Seção 5, os resultados do Pathrate sofrem influência da função Concatenação, que costuma ser habilitada por *default* em rádios WiMAX ou Pré-WiMAX. Isto ocorre devido aos trens de pacotes de tamanho pequeno que o Pathrate gera durante sua fase inicial. A função Burst, que também costuma ser habilitada por *default*, pode também prejudicar a medição caso haja um enlace 802.16 após o ponto de "gargalo" no caminho com tenha esta função ativada. Dependendo do tamanho dos trens de pacotes enviados, as dispersões causadas pelo enlace de menor capacidade podem ser removidas ao passar pelo enlace 802.16.

O tamanho dos pacotes de medição usados pelo CapProbe é parametrizado pelo usuário, mantendo-se o mesmo até o final de sua execução. Dependendo do valor usado, os pares de pacotes poderão ou não sofrer influência da Concatenação habilitada em um rádio 802.16 no caminho. No caso da função Burst estar habilitada, esta quase sempre afetará a estimativa do CapProbe. Isto ocorre por conta da assimetria inerente a estes enlaces e pela característica bidirecional das medidas extraídas. Caso o sentido de "ida" tenha uma capacidade de contenção menor que o sentido de "volta", e caso o Burst esteja ativado em um rádio WiMAX no sentido de "volta", a capacidade registrada pela dispersão dos pacotes na ida será removida.

Devido às características do cenário usado neste trabalho para avaliação experimental, que tem apenas um salto entre rádios, e pelo aspecto unidirecional da medida realizada pelo Pathrate, apenas a influência da Concatenação foi comprovada para esta ferramenta. Já para as estimativas obtidas pelo CapProbe, tanto a influência da Concatenação como a do Burst foram verificadas nos experimentos.

Nos rádios usados no experimento, o tamanho máximo do quadro concatenado foi igual a 2200 bytes e a duração máxima da rajada foi igual a 5 ms, sendo ambos valores *default*.

4.4. Procedimentos Usados nos Experimentos

De acordo com o fabricante [Breeze 2010], as taxas médias de *throughput* variam conforme o sentido e a modulação usada. Além disso, caso as funções de Concatenação e/ou Burst estejam desativadas, a capacidade do enlace pode ser reduzida em até 10% no respectivo sentido. As Tabelas 1 e 2 apresentam os valores de capacidade, conforme sentido e configuração dos rádios, para a modulação de nível oito, que é a que permite maiores taxas. Estes valores foram usados como referência para validar os resultados obtidos com o Pathrate e o CapProbe. Vale notar que, apesar da capacidade de

transmissão na camada física ser de 54 Mbps, a capacidade efetiva de transmissão é substancialmente inferior devido às características de controle da camada MAC.

Tabela 1 - Capacidades esperadas no sentido BS para SS

Configuração	Vazão Média
Default	31,1 Mbps
Burst e/ou Concatenação desativados no BS	28 – 31,1Mbps

Tabela 2 - Capacidades esperadas no sentido SS para BS

Configuração	Vazão Média
Default	26,4 Mbps
Burst e/ou Concatenação desativados no BS	23,7 – 26,4 Mbps

Em linhas gerais, o procedimento adotado nos experimentos se consistiu da aplicação das ferramentas citadas para estimar as capacidades do enlace de rádio montado, e da comparação dos valores obtidos com os valores de *throughput* médio fornecidos pelo fabricante. As medidas para cada ferramenta foram extraídas nas duas direções possíveis: com sondas partindo do lado do BS para o lado SS e vice-versa. Para cada direção, foram variadas as configurações de Burst e Concatenação dos rádios.

No caso do Pathrate, por se tratar de uma estimativa unidirecional, apenas as funcionalidades de Burst e Concatenação do rádio no lado de onde partia as sondas foram variadas. Já no caso do CapProbe, pelas medições serem bidirecionais, para cada direção testada variou-se as funcionalidades de cada rádio separadamente. O tamanho dos pacotes usados pelo CapProbe assumiu valores de 500 e 1500 bytes em experimentos distintos.

5. Resultados

Os itens a seguir mostram uma comparação entre os valores obtidos pelas ferramentas e aqueles informados pelo fabricante, de acordo com a configuração usada para cada experimento. Todas as capacidades estão em Mbps.

5.1. Influência do Burst e da Concatenação nos Resultados do CapProbe

Conforme mencionado na Seção 4.3, a avaliação das estimativas do CapProbe deve considerar o ponto onde os pacotes de medição foram disparados: lado BS ou lado SS. Portanto, os resultados foram agrupados em função deste fator. No que diz respeito aos valores obtidos, como a ferramenta também fornece todas as dispersões medidas para cada par de pacotes enviado, foram também computadas a média, a mediana e a moda amostral destas dispersões, para cada experimento, assim como o desvio-padrão.

Vale observar que, idealmente, no cenário usado, o CapProbe deveria sempre indicar a menor capacidade no caminho de ida e volta, que é sempre a capacidade no sentido SS para BS. Entretanto, dada a influência da função Burst quando habilitada no BS, se pares de pacotes forem enviados a partir do lado SS com destino no lado BS, o CapProbe irá medir a capacidade no sentido de volta (BS para SS) devido à remoção da dispersão na BS.

As Tabelas 3, 4, 5 e 6 trazem os resultados obtidos para os experimentos com CapProbe. Cada tabela agrupa os experimentos realizados com um tamanho específico de pacote, 500 ou 1500 bytes, com pares de pacotes disparados de um lado específico (BS ou SS). Cada linha nas tabelas mostra os resultados para uma dada configuração em cada rádio, sendo que a primeira linha sempre traz os resultados para a configuração *default*, ou seja, Burst e Concatenação ativos em ambos os rádios (BS e SS). As demais linhas mostram as modificações em relação ao *default*: (i) Burst desabilitado na BS; (ii) Burst desabilitado na SS; (iii) Concatenação desabilitada em ambos os rádios; e (iv) Burst e Concatenação desabilitados em ambos os rádios. Para cada configuração, é mostrado nas colunas: o valor esperado conforme informação do fabricante, o valor fornecido pelo *minimum delay sum* (MDS) usado no CapProbe, a mediana, a média, a moda e o desvio-padrão das capacidades extraídas para cada par de pacotes enviado. O número de pares de pacotes enviado em cada experimento variou de 200 a 300.

5.1.1. Estimativas a partir do lado BS

As Tabelas 3 e 4 apresentam um quadro comparativo para os valores obtidos pelo CapProbe, nas diferentes configurações, para pacotes de medição com tamanhos de 500 e 1500 bytes, respectivamente, quando disparados a partir do computador ligado à BS. As capacidades expressas nas tabelas estão em Mbps.

Os valores informados pelo fabricante para a capacidade efetiva foram usados como referência, porém também foram consideradas as configuração de Burst no lado SS para identificar qual resultado o CapProbe deveria fornecer. No caso, a eliminação da dispersão dos pacotes pela SS não impacta a estimativa, já que o enlace de volta é o de menor valor. A faixa de valores de 23,7 a 26,4 Mbps está de acordo com as informações da Tabela 2.

Tabela 3 - CapProbe sentido BS-SS com pacotes de 500 bytes

Configuração	Referência	MDS	Mediana	Média	Moda	Desvio
Default	26.4	235	250	251.5	250	8.430
Burst desab. (BS)	26.4	38	38	37.7	38	2.897
Burst desab. (SS)	23.7 – 26.4	42	27	28	32	5.410
Concat. desab. (BS e SS)	23.7 – 26.4	26.2	26	26.3	26	1.740
Ambos desab. (BS e SS)	23.7 – 26.4	25.2	25	26.4	27	5.380

Pelos resultados da Tabela 3, é possível verificar o impacto negativo causado pela Concatenação quando se usa pacotes de medição com tamanho de 500 bytes. No caso de configuração *default*, com Burst e Concatenação habilitados nos rádios, a estimativa do CapProbe falhou de forma muito acentuada. Entretanto, ao se retirar a Concatenação dos rádios, o CapProbe foi capaz de fornecer uma estimativa dentro do esperado, medida esta que foi muito próxima da média, da mediana e da moda das amostras de dispersão. O Burst habilitado no SS não provoca erro na medida esperada uma vez que o enlace de menor capacidade é o do sentido de volta (SS-BS).

Já nos resultados da Tabela 4, é possível verificar que a Concatenação não surte o mesmo efeito negativo para as estimativas quando os pacotes de medição têm tamanho de 1500 bytes. Isto porque a concatenação de pacotes de 1500 bytes resultaria

num quadro concatenado que ultrapassaria o tamanho máximo permitido (2200 bytes, conforme consta na Seção 4.3), o que faz com que eles não sejam concatenados. Em todos estes casos, o CapProbe foi capaz de fornecer estimativas próximas dos valores esperados (valores nominais). Entretanto, é possível verificar que os valores obtidos para a média, a moda e, em especial, a mediana se aproximaram melhor do que o resultado final do CapProbe.

Tabela 4 - CapProbe sentido BS-SS com pacotes de 1500 bytes

Configuração	Referência	MDS	Mediana	Média	Moda	Desvio
Default	26.4	24.3	25	23.8	26	2.458
Burst desab. (BS)	26.4	31.2	27	28.8	26	2.179
Burst desab. (SS)	23.7 – 26.4	29.8	26	26.3	25	2.514
Concat. desab. (BS e SS)	23.7 – 26.4	26.3	26	26.4	26	2.470
Ambos desab. (BS e SS)	23.7 – 26.4	25.9	26	26.6	29	2.893

5.1.2. Estimativas a partir do lado SS

Assim como para as duas tabelas anteriores, as Tabelas 5 e 6 apresentam um quadro comparativo para os valores obtidos pelo CapProbe com pacotes de medição possuindo 500 e 1500 bytes de tamanho, respectivamente, quando disparados a partir do computador ligado à SS. As capacidades expressas nas tabelas estão em Mbps.

Os valores mostrados na coluna *Referência*, assim como nas Tabelas 3 e 4, consideraram a situação configurada no sentido de volta dos pacotes de medição, ou seja, o lado da BS. Sempre que houver eliminação das dispersões na BS, a capacidade medida será a do sentido BS para SS. Desta forma, sempre que o Burst estiver desativado, a capacidade estimada deverá ser de fato a do sentido SS para BS, que é a de menor valor. Por outro lado, se o Burst estiver habilitado, espera-se que a capacidade estimada seja a do sentido BS para SS. Os valores para as capacidades efetivas de referência estão de acordo com a Tabela 1.

Tabela 5 - CapProbe sentido SS-BS com pacotes de 500 bytes

Configuração	Referência	MDS	Mediana	Média	Moda	Desvio
Default	31.1	41	41	41.2	41	1,074
Burst desab. (BS)	26.4	42	42	41.8	42	0,478
Burst desab. (SS)	28.0 – 31.1	42	42	41.9	42	1,362
Concat. desab. (BS e SS)	28.0 – 31.1	29.4	29.4	29.4	29.4	2,126
Ambos desab. (BS e SS)	23.7 – 26.4	23.8	23	23.4	22	1,820

Pelos resultados da Tabela 5, é possível novamente verificar o impacto negativo causado pela Concatenação quando os pacotes de medição possuem tamanho igual a 500 bytes. Entretanto, o erro introduzido não é tão acentuado quanto o mostrado na Tabela 3. Apenas quando a Concatenação é desligada que o CapProbe torna-se capaz de fornecer uma estimativa dentro dos valores esperados. Nestes casos, a mediana, a média e a moda apresentaram valores muito próximos.

A explicação para a diferença entre os erros mostrados nas Tabelas 3 e 5, causados pela ação conjunta de Burst e Concatenação habilitados no sentido de volta, reside nos tempos que a SS e a BS levam para transmitir os quadros. Como a SS precisa efetuar a reserva de capacidade, e a BS não, os quadros levam mais tempo no *buffer* de transmissão da SS aguardando a liberação do meio. Consequentemente, mais quadros são enfileirados e, portanto, mais quadros são concatenados e enviados em rajada, causando um erro maior na dispersão medida.

Tabela 6 - CapProbe sentido SS-BS com pacotes de 1500 bytes

Configuração	Referência	MDS	Mediana	Média	Moda	Desvio
Default	31.1	29.4	30	29.2	30	1,747
Burst desab. (BS)	26.4	29.2	27	28	27	2,707
Burst desab. (SS)	28.0 – 31.1	24.9	30	28.4	31	3,130
Concat. desab. (BS e SS)	28.0 – 31.1	29.4	31	32.5	31	4,424
Ambos desab. (BS e SS)	23.7 – 26.4	24.7	30	31.2	27	6,163

Os resultados da Tabela 6 mostram que, conforme esperado, para pacotes de medição com tamanho igual a 1500 bytes a Concatenação não impacta na estimativa de capacidade. O CapProbe foi capaz de fornecer estimativas próximas do esperado em todos os casos, entretanto, as modas obtidas das amostras de dispersão se aproximaram melhor dos valores esperados.

Os resultados apresentados pelas Tabelas 4 e 6 sugerem que a técnica estatística usada pelo CapProbe pode não ser a mais adequada para casos de caminhos contendo enlaces no padrão IEEE 802.16, haja visto que medidas estatísticas simples extraídas a partir das amostras de dispersão, como a moda ou a média, apresentaram resultados mais próximos do esperado.

Conforme mencionado anteriormente, vale notar que os valores de referência informados não necessariamente são os valores que deveriam ser corretamente estimados como capacidade de contenção. No ambiente testado, considerando-se uma avaliação bidirecional, este valor deveria ser sempre a capacidade no sentido SS para BS. De forma resumida, isto significa que o Burst, quando habilitado na BS, sempre causará erro nesta medida caso os pacotes de medição sejam disparados do lado SS.

5.2. Influência do Burst e da Concatenação nos Resultados do Pathrate

No caso do Pathrate, como as estimativas são unidirecionais, as configurações do rádio do lado receptor não influenciam as medidas e, diferente do CapProbe, a ferramenta não provê controle do tamanho dos pacotes enviados. Portanto, os resultados foram agrupados em apenas duas tabelas. A Tabela 7 apresenta um quadro comparativo dos resultados relativos a pacotes de medição partindo do lado BS. A Tabela 8 apresenta o mesmo quadro, porém relativo a pacotes de medição partindo do lado SS. As capacidades expressas nas tabelas estão em Mbps.

Os resultados observados nas Tabelas 7 e 8 comprovam que as estimativas do Pathrate falham sempre que a Concatenação está ativa no lado cliente da ferramenta. No caso do cenário experimental usado, a configuração de Burst no lado cliente em nada afeta as estimativas. Todavia, num caso genérico de caminho com diversos enlaces,

sendo um deles no padrão IEEE 802.16, a presença de Burst neste enlace pode remover a informação de "gargalo" obtida num salto anterior através da dispersão dos pares de pacotes.

Tabela 7 - Pathrate sentido BS - SS

Configuração	Referência	Pathrate
Default	31,1	38 – 39
Burst desabilitado	28 – 31,1	35 – 39
Concatenação desabilitada	28 – 31,1	29 – 29
Ambos desabilitados	28 – 31,1	28 – 29

Tabela 8 - Pathrate sentido SS - BS

Situação	Referência	Pathrate
Default	26,4	30 – 30
Burst desabilitado	23,7 – 26,4	28 – 29
Concatenação desabilitada	23,7 – 26,4	25 – 27
Ambos desabilitados	23,7 – 26,4	24 – 26

6. Conclusão e Trabalhos Futuros

Neste trabalho foi possível verificar a imprecisão de métodos considerados estado da arte para estimativa de capacidade quando aplicados a caminhos com enlaces sem fio IEEE 802.16. No caso, os métodos verificados foram baseados na técnica de Pares de Pacotes. Sempre que determinadas características opcionais da camada MAC estão habilitadas, especificamente Concatenação e Burst, o intervalo de contenção entre pacotes é removido fazendo com que a dispersão inserida pelos canais de menor capacidade nas sondas enviadas seja perdida. A Concatenação atua sobre pacotes de tamanho menor, enquanto que o Burst atua sobre pacotes que chegam dentro de um período de tempo configurável chamado duração máxima da rajada.

Um cenário de experimentação usando dois rádios Pré-WiMAX foi montado para verificar as limitações das ferramentas usadas para estimativa de capacidade. Os resultados apresentados comprovam a influência do Burst e da Concatenação no resultado das ferramentas Pathrate e CapProbe, mostrando como as mesmas falham mediante determinadas configurações de enlaces 802.16 no caminho.

Como perspectiva de trabalhos futuros, pretende-se investigar a adoção de determinados parâmetros para o método de trem de pacotes e de pares de pacotes de maneira a torná-los mais imunes às características de Burst e Concatenação dos enlaces 802.16, normalmente habilitadas por *default* nos rádios WiMAX e Pré-WiMAX.

Referências

Augusto, M.; Murta, C. (2003) *Avaliação Experimental de Ferramentas para Medição de Capacidade em Redes de Computadores*. Anais do SBC WPerformance.

Breeze (2010) *Manual do radio Alvarion BreezeConfig for BreezeACCESS VL versão 4.5.0.9*

Carter, R., Crovella, M. (1996) *Measuring Bottleneck Link Speed in Packet Switched Networks*. ACM Performance Evaluation, Volume 27-28.

Dovrolis, C., Ramanathan, P., Moore D. (2001) *What do packet dipersion technique measure*. Proceding of IEEE INFOCOM.

Downey, A. (1999) *Using Pathchar to Estimate Internet Link Characteristics.* ACM SIGCOMM.

IEEE Std 802.16-2003. Standard for Local and Metropolitan area networs. Part 16: AirInterface for Fixed Broadband Wireless Access Systems – Amendment 2: Medium Access Control Modifications and Additional Physical Layer Specifications for 2-11 GHz (Amendment to IEEE Std 802.16-2001). April 01, 2003.

Jacob, S., Katabi, D., Kaashoek, F. (2003) *A Measurement Study of Available Bandwidth Estimation* - Proceedings of the ACM IMC '03.

Jacobson V. (1999) *Pathchar: A Tool to Infer Characteristics of Internet Paths*. Proceedings of the ACM SIGCOMM '99.

Johnsson, A. (2004) *On the Comparison of Packet Pair and Packet Train Measurement.* Swedish National Computer Networking Workshop, Arlandastad.

Kapoor, R., et al. (2004) *Capprobe: A Simple and Accurate Capacity Estimation Technique.* Proceedings of the ACM SIGCOMM.

Lai, D.; Baker, M. (1999) *Measuring bandwidth.* Proceedings of the IEEE INFOCOM.

Prasad, R.; Dovrolis, C.; Nurray M. (1997) *Bandwidth Estimation: Metrics, Measurement Techniques and Tools* , IEEE Network.

Rocha, A. A. A., Leão., R. M. M., Silva E. S. (2004) *Proposta de uma técnica de seleção dos pares de pacotes para estimar a capacidade de contenção*. Anais do SBC WPerformance.

Rocha, A. A. A., Leão., R. M. M., Silva E. S. (2007) *An End-toEnd Technique to Estimate the Transmission Rate of IEEE 802.11 WLAN*. Proceedings of the IEEE ICC.

Rocha, A., et al. (2009) *Novas evoluções integradas à ferramenta Tangram-II v3.1*. Anais do SBRC (Salão de ferramentas).

Roesler. V., Finzsch. P., Andrade. M., Lima. J. V. (2007) Análise do mecanismo de pares de pacotes visando estimar a banda da rede via UDP. Anais do SBRC.

So-In, C., Jain, R., Tamimi, A. A. (2009) *Capacity Estimation of IEEE 802.16e Mobile WiMAX Networks*

Ziviani, A.; Duarte, O. (2005) *Metrologia na Internet*, Anais de Minicursos SBRC.

AntRoP - Protocolo de Roteamento Bio-inspirado em Colônia de Formiga Tolerante a Falhas e Desconexões aplicado às Redes Emergenciais

Luiz H. A. Correia[1], Daniel F. Macedo[2], Michel A. S. Ribeiro[1],Tales Heimfarth[1]

[1] Departamento de Ciência da Computação
Universidade Federal de Lavras
Lavras-MG, Brasil

[2]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte-MG, Brasil

lcorreia,tales{@dcc.ufla.br},damacedo@dcc.ufmg.br, michelsra@gmail.com

Abstract. *In emergency situations, communication is essential for decision making in rescue operations. In the event of natural disasters, generally the communication infrastructure is damaged, hence being necessary to deploy alternative networks. Emergency networks arise as an option for the restoration of communication, being composed of mobile devices like PDAs, smartphones, fixed and mobile stations, interconnected in an ad hoc manner. Such networks are susceptible to delays and disconnections during message delivery, hence are characterized as DTN networks (Delay Tolerant Network). This paper proposes and evaluates a bio-inspired routing protocol based on ant colony for emergency networks, called AntRoP (Ant Routing Protocol). Simulations assessed the performance of the proposed routing protocol in disaster scenarios. The results show that AntRoP has a delivery rate 41% higher than the Epidemic protocol and 9.5% higher than Prophet's, which are two of the most important routing protocols for DTNs.*

Resumo. *Em situações de emergência a comunicação é essencial para a tomada de decisão das equipes de resgate. Na ocorrência de desastres, geralmente a infraestrutura das redes de comunicação é danificada, sendo necessário implantar redes alternativas. As redes emergenciais surgem como uma opção para criar uma estrutura de comunicação, sendo compostas por dispositivos móveis interligados em modo ad hoc. Essas redes são susceptíveis a atrasos e desconexões na entrega de mensagens, o que as caracteriza como redes DTN (Delay Tolerant Network). Um dos maiores desafios de tais redes é o roteamento. Este artigo propõe e avalia um protocolo de roteamento bio-inspirado em colônia de formigas, tolerante a falhas e desconexões de nós e enlaces, para redes emergenciais, denominado AntRoP (*Ant Routing Protocol*). Simulações de desempenho avaliaram três protocolos de roteamento em cenários de desastre. Os resultados mostraram que o AntRoP possui, em média, uma taxa de entrega 41% maior que o protocolo Epidêmico e 9,5% maior que o PRoPHET, dois dos principais protocolos de roteamento DTN da literatura.*

1. Introdução

Os desastres, tanto naturais, humanos ou tecnológicos, trazem transtornos e prejuízos à sociedade. Em desastres, as redes de comunicação de voz e dados podem ser totalmente destruídas ou danificadas, ou não podem ser utilizadas de forma confiável, pois se encontram congestionadas. A falta de uma infraestrutura de comunicação pode tornar a situação de desastre ainda mais caótica, dificultando a coordenação das equipes de resgate e o pronto atendimento às vítimas. Nesse caso, uma rede emergencial deve ser rapidamente implantada para restabelecer a comunicação e dar suporte aos serviços de voz e transmissão de dados [Schmitt et al. 2007]. Muitos governos estão cada vez mais conscientes da importância e dos benefícios que as redes emergenciais podem oferecer ao serem utilizadas em situações críticas e de crise [Hinton et al. 2005].

As redes emergenciais de comunicação não necessitam de uma infraestrutura fixa. Essas redes são compostas de nós, que são dispositivos como PDAs, *smartphones* e estações móveis, trazidos pela equipe de resgate que podem ser interligados em modo ad hoc. Esses nós tendem a se movimentarem em grupos e, em geral, existe uma maior densidade de nós em pontos de interesse, como o centro de controle de incidentes (que eventualmente pode possuir uma estação fixa) e as regiões de busca e resgate.

Um grande desafio nos cenários de desastre é a baixa capacidade de armazenamento de energia dos dispositivos móveis. Neste caso, o problema não está em somente transmitir os dados necessários de forma correta, mas também em economizar energia dos dispositivos móveis. Além das restrições de recursos, a conexão entre os nós está sujeita a severas atenuações dos sinais transmitidos, seja por obstáculos, interferências ou pela distância entre os nós. Finalmente, a mobilidade dos nós torna a topologia da rede dinâmica e altamente esparsa. As frequentes desconexões impostas pela dinamicidade da topologia da rede impedem que os protocolos de roteamento tradicionais possam ser aplicados nas redes de emergência. Essas redes possuem características de atrasos e desconexões similares às redes DTN (*Delay Tolerant Networks*). Os protocolos de roteamento empregados nas redes DTN são baseados em replicação ou encaminhamento de mensagens, e têm sido testados nas redes emergenciais.

As redes DTN suportam desconexões frequentes e longos atrasos na entrega de mensagens, que podem ser de horas e até mesmo dias. As desconexões podem ocorrer pela alta mobilidade dos nós, por péssimas condições de comunicação e/ou por economia de recursos. Esses eventos podem resultar em uma conectividade intermitente da rede durante um período, ou ainda, pode ser que um caminho entre a origem e o destino nunca chegue a ficar completamente estabelecido [de Oliveira 2008]. Dessa forma, os protocolos de roteamento empregados nas redes móveis ad hoc (MANET - *Mobile Ad Hoc Networks*) são ineficientes nas redes DTN, já que necessitam estabelecer um caminho fim-a-fim para comunicação entre os nós [Jain et al. 2004].

As heurísticas baseadas em colônia de formigas têm sido usadas para encontrar soluções ótimas em problemas computacionais, como o do caixeiro viajante, buscas em grafos e roteamento em redes móveis [Stutzle et al. 1999, Dorigo et al. 2006, Caro et al. 2004]. Neste trabalho propomos um protocolo de roteamento bio-inspirado em colônia de formigas denominado AntRoP (*Ant Routing Protocol*) para as redes emergenciais. Esse protocolo é baseado no comportamento de auto-organização observado em colônias de formigas, utilizando a descoberta de caminhos mais curtos por meio de rastros

de feromônio. O objetivo é encontrar uma maneira eficiente de estabelecer uma conexão entre os envolvidos e rotear as mensagens em uma rede móvel sem fio em cenários de emergência, visando reduzir a latência e a quantidade de mensagens enviadas, bem como melhorar a taxa de entrega.

O desempenho do protocolo AntRoP foi avaliado e comparado por simulação com duas soluções clássicas da literatura, os protocolos Epidêmico [Vahdat and Becker 2000] e PRoPHET [Lindgren et al. 2003]. Foram avaliadas as métricas de taxa de entrega, latência e overhead de comunicação. Os resultados mostram que o AntRoP possui, em média, uma taxa de entrega 41% maior que o protocolo Epidêmico e 9,5% maior que o PRoPHET.

Este trabalho está organizado como descrito a seguir. A Seção 2 apresenta os trabalhos relacionados. O algoritmo proposto é descrito na Seção 3. A Seção 4 apresenta a avaliação experimental e os resultados encontrados. Finalmente, a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Segundo [Schmitt et al. 2007] as redes emergenciais são redes construídas sobre cenários de desastres e tem a propriedade de estabelecer uma comunicação robusta. Elas não são necessariamente infraestruturadas e oferecem comunicação de dados e de voz. As redes emergenciais são uma das várias aplicações típicas de redes ad hoc [Boukerche 2008]. A comunicação durante o incidente é estabelecida através de dispositivos móveis sem fio, trazidos pelas equipes de resgate (policiais, bombeiros, defesa civil). Esses dispositivos são conectados de forma ad hoc e formam uma rede que visa a comunicação e a transmissão de dados entre as equipes de resgate.

Na literatura encontramos diversas redes emergenciais que empregam redes MANETs, entre elas os projetos Sahana e DUMBO. O Sahana é um sistema Web de código livre que visa ajudar em vários problemas de coordenação que ocorrem após e durante um desastre [Careem et al. 2007]. Essa ferramenta disponibiliza na Web informações que auxiliam a encontrar pessoas desaparecidas, gerenciar a ajuda e acompanhar a situação de acampamentos para desabrigados. O sistema foi utilizado no Tsunami da Ásia em 2004 e também no terremoto do Haiti no início de 2010. Apesar disso, o Sahana não investiga a intercomunicação entre os dispositivos móveis empregados no local de desastre.

O projeto DUMBO [Kanchanasut et al. 2008] (*Digital Ubiquitous Mobile Broadband OLSR*) implanta uma MANET para situações pós-desastre, na qual a infraestrutura de rede fixa não está disponível, assim uma rede de emergência deve ser instalada com urgência. Redes DUMBO suportam dispositivos heterogêneos e possibilitam a transmissão de *streaming* de vídeo, VoIP e mensagens curtas. Apesar disso, neste projeto é assumido que a infraestrutura de comunicação pode ser parcialmente restaurada e que existe acesso à Internet. Além disso, somente o protocolo de roteamento OLSR (*Optimized Link State Routing Protocol*) foi considerado no projeto.

As redes emergenciais, consideradas um caso particular de uma rede DTN (*Delay Tolerant Network*), podem ser classificadas em determinísticas ou estocásticas [Zhang 2006]. Um cenário é dito determinístico quando a informação dos instantes de contato entre os nós da rede e a capacidade de armazenamento dos mesmos é previsível, caso contrário, esse cenário é considerado estocástico. Dessa forma, os protocolos de ro-

teamento nas redes DTN são divididos em dois grupos: os baseados em replicação, que mantêm a mensagem em *buffer* durante a transmissão (empregados em redes estocásticas) e os baseados em encaminhamento (utilizados em redes determinísticas), que apagam a mensagem após encaminhá-la. Redes de emergência são normalmente estocásticas.

O protocolo de roteamento Epidêmico visa maximizar a taxa de entrega e a latência por meio de repasses de mensagens a cada nó contatado que possua espaço em *buffer*. O funcionamento desse protocolo é semelhante a uma doença epidêmica: enquanto houver espaço em *buffer* as mensagens são repassadas a cada contato entre os nós. Cada nó da rede possui uma lista com as mensagens que ele armazena. Essa lista é trocada entre vizinhos que estão no mesmo alcance de transmissão para determinar quais mensagens serão solicitadas. Esse processo se repete sempre que um nó entra em contato com um novo vizinho. Na Figura 1 é mostrado o funcionamento do Epidêmico para uma rede com 20 nós. O protocolo Epidêmico possui um tempo ótimo de propagação de dados, pois explora todos os caminhos ao mesmo tempo. Além disso, apesar do protocolo de roteamento Epidêmico não detectar falhas, a redundância e aleatoriedade na disseminação de mensagens contornam potenciais falhas de nós ou enlaces [Gupta et al. 2002]. Entretanto, o protocolo Epidêmico consome uma alta quantidade de mensagens e rapidamente ocupa todo o *buffer* dos nós.

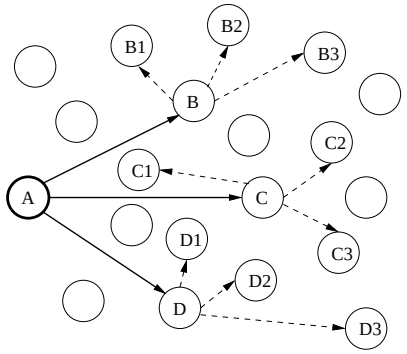


Figura 1. Protocolo Epidêmico.

Devido ao alto custo do Epidêmico, novos protocolos foram propostos tentando minimizar a quantidade de mensagens enquanto mantendo uma entrega rápida e confiável. Em [Spyropoulos 2007] foi proposta uma família de protocolos de múltiplas cópias chamada de *Spray*. A ideia central é que a origem gere um pequeno número de cópias das mensagens, assegurando que o número de transmissões seja reduzido e controlado. Um dos esquemas propostos é chamado de *Spray and Wait*, no qual o nó origem repassa todas as cópias para os primeiros N nós distintos que ele encontra. Uma vez que as cópias são distribuídas é realizada a transmissão direta. Segundo os autores o roteamento *Spray and Wait* não tem um bom desempenho para cenários onde a mobilidade é pequena e localizada, pois depende fortemente da mobilidade para que a mensagem seja entregue.

Outra abordagem é o uso de métricas de redes sociais, tais como histórico de contatos, mobilidade e número de nós na vizinhança. O protocolo MV (*Meeting Visit*) mantém um modelo de movimento dos participantes em uma rede DTN e usa essa informação para executar o encaminhamento de mensagens na rede. Este protocolo aprende a frequência de contatos entre pares de nós e suas visitas em certas regiões geo-

gráficas. Essas frequências de contato são usadas para classificar cada mensagem de acordo com sua probabilidade de entrega através de um caminho específico.

O PRoPHET (*Probabilistic Routing Protocol using History of Encounters and Transitivity*), também utiliza métodos de análise de conectividade da rede para entregar de forma eficiente uma mensagem ao destinatário [Lindgren et al. 2003]. O PRoPHET assume a não aleatoriedade de movimento dos nós da rede para melhorar a entrega das mensagens. Assim, o protocolo procura identificar os nós que mais se encontram com outros nós, para que sejam escolhidos prioritariamente para o repasse de mensagem. Dessa forma, quando dois nós se encontram, uma mensagem é transmitida para o outro nó se a probabilidade de entrega ao destino for mais alta no outro nó. Porém, como o nó que repassou a mensagem pode encontrar em seu caminho outro nó que possua melhor probabilidade que o anterior, ou até mesmo encontrar o nó destino, a mensagem não é removida imediatamente de seu *buffer* se este ainda possui espaço disponível.

No PRoPHET, quando um nó encontra outro nó, eles trocam informações sobre a probabilidade de encontrar um certo destino e decidem se trocam ou não mensagens entre eles. Essa probabilidade de encontrar um determinado destino é dado por $P \epsilon [0, 1]$. A probabilidade $P(a, b)$ de um nó a encontrar um nó b, aumenta sempre que esses nós se encontram, e é reduzida periodicamente, de forma a evitar que nós que se encontravam frequentemente, mas que agora não se encontram mais, continuem possuindo uma alta probabilidade. A entrega de mensagens se baseia na transitividade: se um nó a encontra frequentemente um nó b, e o nó b encontra frequentemente um nó c, logo o nó c possui uma grande probabilidade de entregar mensagens destinadas ao nó a. Os resultados das simulações demonstram que o PROPHET apresenta um bom desempenho em redes com alta mobilidade ou que possuem nós com grandes alcances de comunicação, já que estes fatores permitem um maior número de encontros de nós, o que conseqüentemente permite que mais informações sobre a rede sejam trocadas.

O protocolo de roteamento AntHocNet (*Ant Colony Optimization*), baseado em sistemas biológicos, foi desenvolvido para redes MANET [Caro et al. 2004]. Esse protocolo é baseado no comportamento de auto-organização observado em colônias de formigas, sendo que a ideia central é fornecer componentes reativos e pró-ativos para o roteamento. As formigas, ou agentes, amostram continuamente possíveis caminhos e registram a qualidade do enlace em variáveis que representam o rastro de feromônio deixado pelas formigas. Dessa forma, múltiplos caminhos são descobertos e armazenados em tabelas dos nós da rede. Esses caminhos são classificados de acordo com a quantidade de feromônio registrado. Para encontrar uma rota entre um nó de origem e outro de destino, os pacotes de dados são distribuídos estocasticamente sobre os nós da rede. A escolha do próximo salto é baseada em uma probabilidade proporcional à quantidade de feromônio, o que limita a quantidade de pacotes enviados em difusão na rede. Apesar disso, esse protocolo não considera as características de falhas e desconexões comuns nas redes emergenciais.

O protocolo de roteamento AntRoP, proposto neste trabalho, também é bio-inspirado em colônia de formigas, com a diferença de ser desenvolvido para redes de emergência. Dessa forma, o AntRoP considera as falhas e desconexões da rede, sendo comparado e avaliado com outros protocolos desenvolvidos para redes DTN.

Recentemente outros protocolos de roteamento para redes de emergência baseados

em contexto têm sido desenvolvidos, como os protocolo *3D routing* [Jacinto et al. 2010] e *Modified CAR*. Esses protocolos não foram avaliados neste trabalho, sendo que o AntRoP foi comparado somente aos protocolos de roteamento clássicos Epidêmico e PRoPHET.

3. Protocolo de Roteamento AntRoP

O protocolo de roteamento bio-inspirado AntRoP (Ant Routing Protocol) foi desenvolvido baseado no comportamento de auto-organização observado em colônias de formigas. Segundo [Caro et al. 2004] as heurísticas baseadas em colônia de formigas têm sido usadas com sucesso no roteamento de redes móveis. O AntRoP foi desenvolvido para ser usado em redes DTN, visando reduzir a latência e melhorar a taxa de entrega. Essas melhorias são baseadas na hipótese de encontrar um caminho mais curto entre a origem e o destino para a entrega de mensagens.

As formigas andam de forma aleatória até encontrar o alimento e, quando retornam ao formigueiro, deixam nesse caminho rastros de uma substância química volátil conhecida como feromônio. Caso não encontrem alimento, as formigas, ao retornarem ao formigueiro, informam que naquele caminho não existe alimento. Nas próximas buscas as formigas seguirão o rastro de feromônio de maior intensidade que possui alimento, ao invés de saírem em buscas aleatórias.

A trilha de feromônio pode evaporar com o tempo se nenhuma formiga seguir esse caminho novamente, ou pode ser reforçada quando outras formigas seguem o mesmo rastro. A ideia é que se o rastro de feromônio tem alta intensidade, as formigas irão seguir por este caminho mais rapidamente e este será o menor caminho entre o formigueiro (origem) e o alimento (destino). A redução da intensidade de feromônio no tempo tem como vantagem a convergência para encontrar o caminho mais curto, sendo uma solução ótima local. Se o feromônio não evaporasse, todos os caminhos encontrados pelas primeiras formigas seriam considerados os de menor distância. Como o feromônio é volátil e evapora, as formigas ao saírem do formigueiro irão escolher o caminho com o rastro de feromônio mais forte, que indica o menor caminho.

De forma análoga, os nós da rede de emergência ao se movimentarem buscam um caminho para entregar mensagens de uma origem até um destino. Os nós seguem diferentes caminhos, que são armazenados em suas tabelas de rotas, e transportam as mensagens que são repassadas entre os nós da origem até o ponto de interesse no destino.

Se um nó s tem uma mensagem para ser enviada a um destino d, então ele verifica na sua tabela de rotas se este nó pode ser alcançado diretamente e, em caso afirmativo, entrega a mensagem para ele. Caso contrário, o nó s deve repassar a mensagem para algum nó que conheça um caminho até o destino. Se os nós vizinhos de s não possuem uma rota para o destino d, a mensagem é enviada em difusão para a rede até que encontre algum nó que possua uma rota válida ou que encontre o nó de destino d. Os nós que recebem a mensagem armazenam o caminho de volta, o número de saltos e o atraso fim-a-fim, para que no futuro possam repassar mensagens para s. As rotas são armazenadas somente por um período de tempo, já que os nós dessa rede se movimentam e a rota pode ficar defasada.

O nó s ao encontrar um vizinho b, que tenha uma rota válida para d, repassa a mensagem para esse nó, atualiza a sua tabela de rotas e descarta a mensagem de seu buffer. O nó b que transporta a mensagem, ao contatar outros nós no caminho, compara

as rotas de suas tabelas, a validade dessas tabelas (intensidade de feromônio) e calcula a probabilidade de entrega da mensagem para cada um dos nós vizinhos. Se a probabilidade de entrega de um outro nó é maior que a de b, e se tabela é válida, a mensagem é repassada para o outro nó. A troca dessas informações de rotas entre os nós reduz o número de saltos para a entrega da mensagem. Caso o nó b não encontre no caminho outros nós que possuem rotas para d, ou com probabilidade de entrega maior, a mensagem é enviada novamente em difusão para a rede. Isso evita aumento do atraso na entrega da mensagem e prováveis falhas e desconexões de enlaces e nós.

Um nó, ao encontrar o destino d, entrega a mensagem e atualiza a tabela de d informando a rota seguida. O nó d, ao enviar uma mensagem de resposta para o nó s, possui uma rota de retorno válida por um curto período de tempo e com probabilidade de entrega proporcional ao número de saltos e à velocidade de movimentação dos nós. Dessa forma, quanto maior o número de saltos necessários para a entrega da mensagem, maior será a probabilidade das rotas armazenadas nos nós serem inválidas. Isso também pode ocorrer se os nós da rede se movimentam rapidamente e a mensagem não encontrar mais os vizinhos que estão na sua tabela de rotas. Nos dois casos, para entregar a mensagem repete-se o processo anterior.

A quantidade de feromônio indica a valorização dos caminhos utilizados recentemente e a desvalorização dos demais caminhos. As atualizações de feromônio são responsáveis pela manutenção das rotas durante o movimento dos nós. A equação 1 apresenta o cálculo da evaporação do feromônio em cada nó [Gunes et al. 2002]. O parâmetro τ indica a atualização do feromônio sobre o enlace i, j para o destino d, tal que $0 \leq \tau \leq 1$. O cálculo é efetuado considerando a quantidade anterior de feromônio armazenada pelo nó $\tau_{i,j(t-1)}$ e o intervalo de tempo t transcorrido na transferência da mensagem.

$$\tau_{i,j}^{d} = \frac{2}{\tau_{i,j(t-1)} + t} \tag{1}$$

A probabilidade de entrega das mensagens é calculada a cada interação entre os nós, comparando as rotas e as suas validades pela intensidade de feromônio. O nó que possui maior quantidade de feromônio terá maior probabilidade de entregar as mensagens. A equação 2 mostra o cálculo da probabilidade de uma mensagem ser transferida do nó i para o nó j com destino ao nó d, na qual N_i representa o conjunto de vizinhos e k é o expoente de seleção de rotas e determina a sensibilidade da mensagem encontrar um caminho e alterar o valor do feromônio:

$$P_{i,j}^{d} = \begin{cases} \frac{\tau_{i,j}^{k}}{\sum_{j \epsilon N_i} \tau_{i,j}^{k}}, & se\ j \in N_i \\ 0 \qquad\quad , & se\ j \notin N_i \end{cases} \tag{2}$$

O incremento do feromônio é dado por um parâmetro $\Delta\tau$ quando a mensagem encontra uma rota para o destino d sobre o enlace i, j, mostrado pela equação 3.

$$\tau_{i,j}^{d} = \tau_{i,j}^{d} + \Delta\tau,\ para\ 0 \leq \tau \leq 1 \tag{3}$$

O funcionamento do AntRoP é apresentado pelo Algoritmo 1. Inicialmente um nó de origem cria uma mensagem que deve ser enviada para um nó destino d. Em cada nó i que recebe a mensagem durante a sua propagação, os parâmetros do protocolo são ajustados, como o valor do do tempo t e a sensibilidade da mensagem encontrar uma rota e alterar o valor do feromônio k (linhas 1 a 3). O nó i verifica em sua vizinhança se d pode

ser alcançado diretamente, caso seja possível, i repassa a mensagem para d, atualiza o valor do feromônio e retira a mensagem de seu buffer (linhas 7 a 9). Se d não é alcançável diretamente, a probabilidade de entrega da mensagem é calculada e comparada para todos os nós vizinhos de i (linha 11). O nó com maior probabilidade de entregar a mensagem é registrado (linhas 12 a 14). Se a probabilidade de entrega calculada é zero, a mensagem será enviada em difusão para a rede (linha 16). Caso contrário, se a probabilidade é maior que zero para o nó registrado, i repassa a mensagem para o nó j, atualiza a sua tabela de rota e descarta a mensagem de seu buffer (linhas 18 a 20). O feromônio decai com o tempo t transcorrido, assim as rotas de i para j devem ser atualizadas, de forma a emular a evaporação do feromônio (linha 22).

Algoritmo 1 Pseudo-código do AntRoP.

1: $k \leftarrow 3$; $t \leftarrow 0$; $\Delta\tau \leftarrow 0.1$ // Setup inicial dos parâmetros
2: **for** $\{(i,j) \mid j \in N_i\}$ **do**
3: $\tau_{i,j} \leftarrow 0$ // Inicialização do feromônio
Require: $Send(Message)\ to\ N_d$ // Nó i envia mensagem para d
4: $Maior \leftarrow null$
5: $P^d \leftarrow 0$
6: **for** $\{j \mid i \in N_i\}$ **do**
7: **if** $(N_j = N_d)$ **then** // se d é alcançado diretamente por i
8: $Maior \leftarrow N_d$
9: $P^d \leftarrow 1$
10: **else**
11: $P^d_{i,j} = \frac{\tau^k_{i,j}}{\sum_{i \epsilon N_i} \tau^k_{i,j}}$ // calcula a probalidade de entrega para os nós vizinhos
12: **if** $(P^d_{i,j} > P^d)$ **then**
13: $Maior \leftarrow N_j$ // identifica a maior probabilidade de entrega
14: $P^d \leftarrow P^d_{i,j}$
15: **if** $(Maior = null)$ **then** // probabilidade igual a zero, uso de difusão
16: $Broadcast(Message)$
17: **else**
18: $Send(Message)\ to\ Maior$
19: $\tau_{i,Maior} \leftarrow \tau_{i,Maior} + \Delta\tau$ // atualiza valor do feromônio
20: $N_i.Message \leftarrow null$ // i retira mensagem do buffer
21: **for** $\{(i,j) \mid j \in N_i\}$ **do**
22: $\tau^d_{i,j} = \frac{2}{\tau_{i,j(t\ -\ 1)} + t}$ // evaporação do feromônio

Os protocolos de roteamento Epidêmico, PRoPHET e AntRoP foram avaliados e comparados em um cenário de desastre fictício. Os resultados são apresentados e discutidos a seguir.

4. Resultados e Discussão

Os protocolos de roteamento Epidêmico, PRoPHET e AntRoP foram avaliados em um simulador para redes DTN chamado *The One* [Keränen et al. 2009]. Esse simulador tem como características: gerar concomitantemente diferentes modelos de movimento para diferentes nós da rede, rotear mensagens entre nós heterogêneos e fornecer uma

interface de visualização gráfica de cenários que apresentam o movimento dos nós e a entrega de mensagens em tempo real.

Para a simulação foi adicionado um mapa de representação dos pontos de interesse de um cenário real. O mapa foi criado no *OpenJUMP*, e representa ruas e avenidas da cidade de Lavras-MG, incluindo a UFLA (Universidade Federal de Lavras) [Vivid Solutions 2009]. Foram considerados um ponto de desastre na Reitoria e dois pontos de interesse, um centralizado na polícia militar e outro em um Hospital, no qual se concentra o atendimento às vítimas, como mostra a Figura 2. Os parâmetros utilizados são os mesmos empregados nos artigos originais dos protocolos Epidêmico e PRoPHET. Os parâmetros de inicialização para os protocolos de roteamento são ajustados com os valores da Tabela 1.

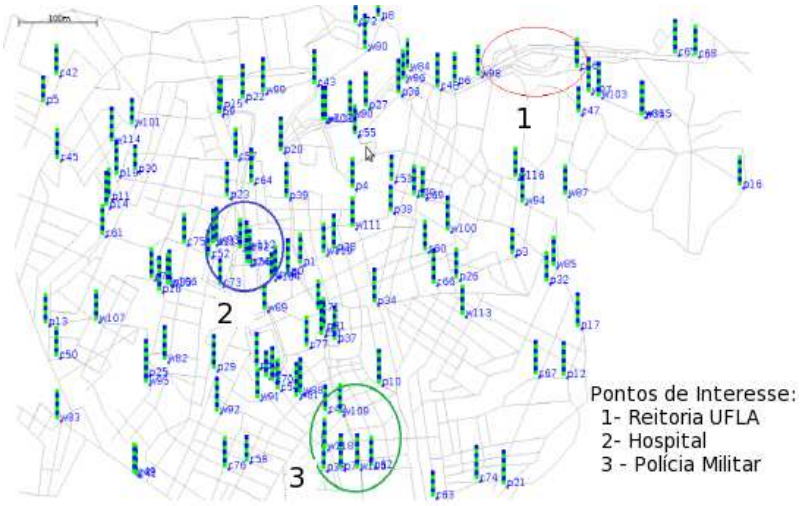


Figura 2. Cenário de simulação.

Tabela 1. Parâmetros iniciais dos protocolos de roteamento.

Parâmetros	Epidêmico	PRoPHET	AntRoP
TTL	∞	—	—
Fanout	∞	—	—
Buffer	Variável/simulação	Variável/simulação	Variável/simulação
K (desvanecimento)	—	Tempo da simulação	—
P_{init}	—	0,75	—
γ	—	0,98	—
$\tau_{i,j}$	—	—	0
$\Delta\tau$	—	—	0.1

Os parâmetros dos dispositivos móveis nas simulações foram ajustados próximos de suas especificações reais. Para a simulação foi considerado o modelo de movimento baseado em mapa (*Shortest Path Map Based Movement*), uma adaptação do algoritmo de Dijkstra para encontrar o caminho mais curto na área do mapa. Esse modelo de movimento pode conter pontos de interesse (POI), onde os nós costumam movimentar com mais freqüência. Os parâmetros utilizados na simulação estão sumariados na Tabela 2.

Tabela 2. Parâmetros da simulação.

Modelo de Movimento	SPMBM
Tamanho do buffer	Variável
Velocidade dos pedestres (agentes)	1,8 – 5,4 Km/h
Velocidade dos carros (viaturas)	10 – 50 Km/h
Tempo de simulação	$\sim$ 6 horas
Mensagens criadas	678
Velocidade de Transmissão	2 Mbps
Tamanho das mensagens	500 KB – 1 MB
Área da simulação	4500 x 3400 m

As métricas escolhidas para avaliar a simulação no *The One* no cenário proposto foram: Taxa de entrega, Latência, *Overhead* de Transmissão e Número de saltos. O cálculo do *Overhead* de transmissão é dado pela equação 4.

$$Overhead\,de\,transmissão\ =\ \frac{Mensagens\,Transmitidas\ -\ Mensagens\,Entregues}{Total\,de\,Mensagens} \tag{4}$$

As métricas dos protocolos foram avaliadas considerando-se alcance de transmissão, escalabilidade da rede e capacidade de armazenamento dos nós. O consumo de energia não foi considerado neste trabalho, mas apesar disso, pode-se inferir a redução do consumo de energia pela redução de mensagens transmitidas ou retransmitidas na rede.

4.1. Alcance de Transmissão

O alcance de transmissão dos dispositivos móveis usados pela equipe de resgate e por veículos em movimento foi variado de 5 a 50m. Nesta análise a quantidade de nós foi fixada em 90, representando por três grupos: 30 pedestres, 30 veículos e mais 30 pedestres com velocidade diferenciada. Foram avaliadas as métricas de *Overhead* de transmissão e Latência. O *Overhead* de transmissão é uma métrica que fornece a percepção de quantas mensagens estão sendo retransmitidas para cada mensagem entregue ao seu destino. A Figura 3 (a) e (b) apresentam o *Overhead* de transmissão e a latência quando o alcance transmissão é variado.

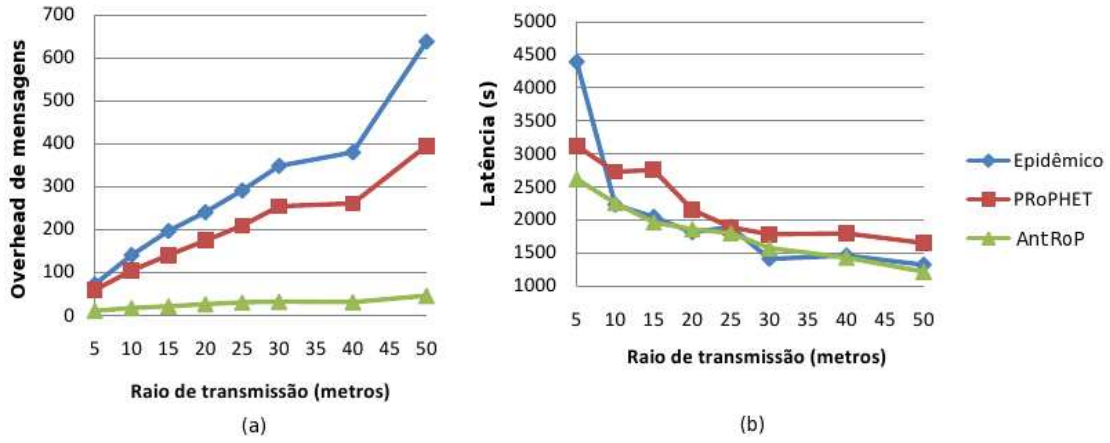


Figura 3. Alcance de transmissão: (a) *Overhead* de transmissão (b) Latência.

Os resultados mostram que o protocolo AntRoP obteve o menor *Overhead* de

transmissão para todas as variações do raio de transmissão. Em média o protocolo AntRoP obteve resultados 10 vezes menor que o protocolo Epidêmico e 6 vezes menor que o protocolo PRoPHET. Esses resultados demonstram que o AntRoP entrega as mensagens com menos retransmissões. O protocolo Epidêmico transmite as mensagens em *flooding* para muitos nós da rede, o que gera uma maior quantidade de mensagens retransmitidas mesmo quando o alcance de transmissão é expandido. A expansão do alcance de transmissão para o protocolo PRoPHET reduz o seu *Overhead* de transmissão, mas mesmo assim este ainda utiliza um mecanismo baseado em probabilidades que não garante a entrega da mensagem. O protocolo AntRoP não foi sensível à variação do alcance de transmissão para a entrega das mensagens, já que na atualização dos rastros de feromônio deixado pelos nós as mensagens são removidas do buffers dos nós.

A latência foi semelhante entre os protocolos Epidêmico e AntRoP, sendo a maior diferença no alcance de 5 m. Para os alcances de 10m a 50m esses dois protocolos possuem uma ligeira diferença nos tempos, chegando o AntRoP a ser melhor em 12%. O protocolo PRoPHET foi o que apresentou a maior latência, sendo inferior em 21% em relação ao AntRoP. O desempenho do AntRoP se justifica pela escolha da rota mais curta devido à intensidade de feromônio.

4.2. Escalabilidade da rede

A quantidade de agentes de resgate que estão inseridos em uma região de desastre varia em função do tipo de incidente, da quantidade de vítimas e da sua área de abrangência. Portanto, é essencial verificar o comportamento dos protocolos de roteamento quando a quantidade de nós é incrementada. Para a simulação da escalabilidade da rede, foi considerado que os dispositivos móveis dos agentes têm um alcance de transmissão fixado em 50 m. A escalabilidade da rede é simulada de 20 a 140 nós considerando as métricas: taxa de entrega, *Overhead* de transmissão, latência e número de saltos. A Figura 4 apresenta os resultados das métricas considerando a variação do número de nós. A Figura 4(a) apresenta a taxa de entrega, e o protocolo AntRoP obteve a melhor Taxa de Entrega, cerca de 40,3% e 9,5% superior aos protocolos Epidêmico e PRoPHET respectivamente. O protocolo Epidêmico obteve o pior *Overhead* de transmissão cerca de 6379 retransmissões, como mostra a Figura 4(b). Isso é porque o protocolo Epidêmico envia de forma aleatória as mensagens para todos os seus nós vizinhos, e os vizinhos também as reenviam para os próximos nós. O protocolo PRoPHET obteve o valor médio de 863,25 mensagens retransmitidas, próximo do AntRoP, que foi de 679. O protocolo PRoPHET limita as retransmissões dos vizinhos, enquanto o AntRoP baseia-se na intensidade de feromônio.

O comportamento dos três protocolos demonstra que a latência diminui à medida que a quantidade de nós aumenta, como mostra a Figura 4(c). Os protocolos Epidêmico e AntRoP obtiveram em média uma latência similar, enquanto o PRoPHET foi 44% mais lento. Vale salientar que o protocolo Epidêmico apresenta a latência próxima da ótima, pois emprega o método de inundação para a disseminação das suas mensagens. Isso mostra que a latência do AntRoP se aproxima da latência ótima.

O protocolo AntRoP encontra o destino em média com apenas dois saltos, enquanto os protocolos Epidêmico e PRoPHET gastam cerca de 5 e 4 saltos respectivamente (Figura 4(b)). O desvanecimento do feromônio permite ao protocolo AntRoP atualizar as tabelas de rotas e encontrar o menor caminho até o destino.

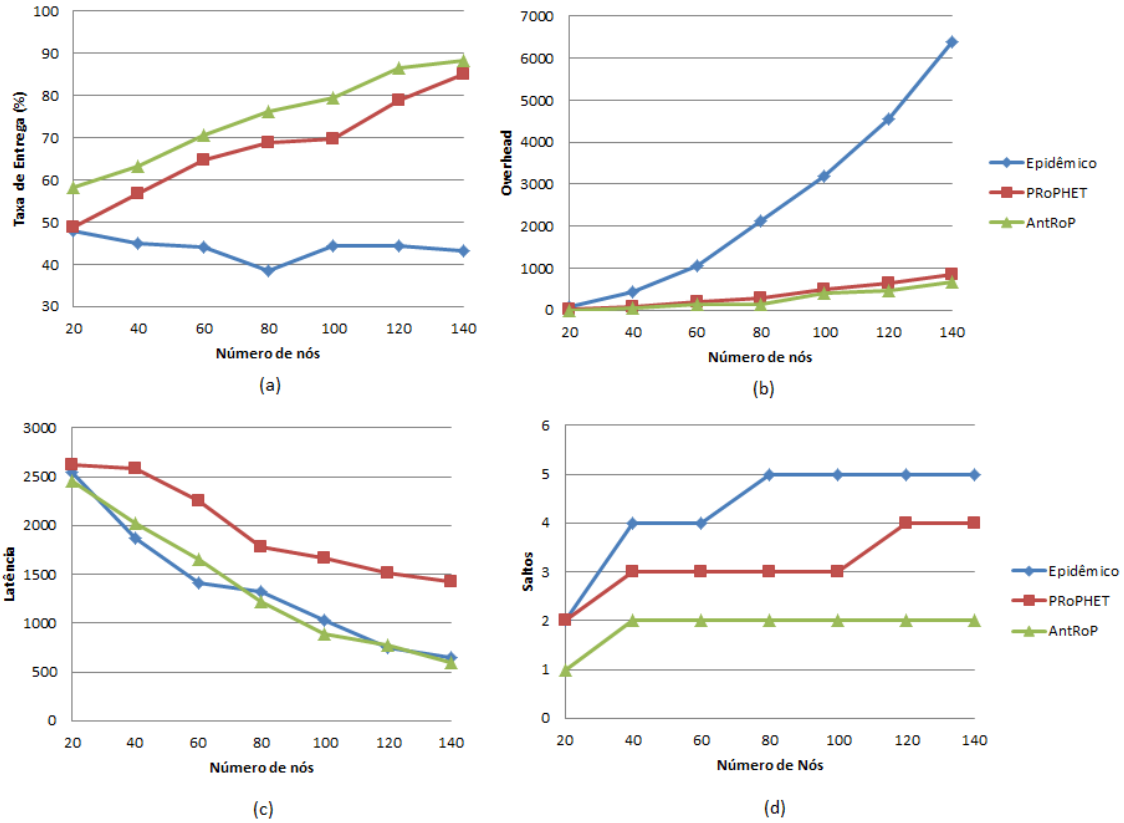


Figura 4. Escalabilidade: (a) Taxa de entrega (b) *Overhead* de transmissão (c) Latência (d) Número de Saltos

4.3. Tamanho do buffer

A capacidade de armazenamento dos dispositivos móveis interfere no envio das mensagens e na avaliação dos protocolos utilizados. Na simulação a capacidade de buffer dos dispositivos móveis utilizados foi variada em 25, 50, 75 e 100 Mbytes. Essa variação se aplica tanto para os pedestres quanto para as viaturas, e o alcance de transmissão foi fixado em 50 metros. Os outros parâmetros de simulação foram os mesmos mostrados na Tabela 2. A Figura 5 apresenta os resultados para o *Overhead* de transmissão e a taxa de entrega ao variarmos o tamanho de buffer.

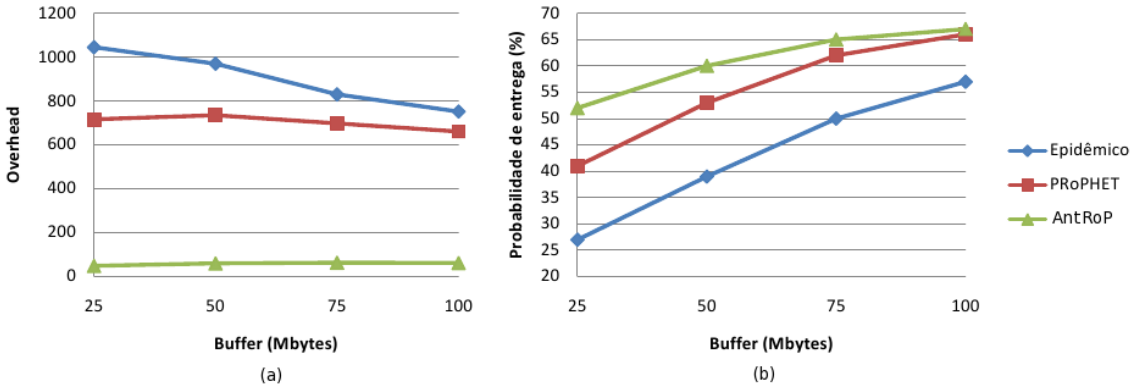


Figura 5. Tamanho do buffer: (a) *Overhead* de transmissão (b) Taxa de entrega.

O protocolo Epidêmico reduziu o *Overhead* de transmissão com um buffer maior, porque pode transportar mais mensagens e trocá-las com seus vizinhos. Isso incrementou a sua taxa de entrega em cerca de 11,5%. O protocolo PRoPHET manteve o mesmo *Overhead* de transmissão porque, ao contatar seus vizinhos, calcula a probabilidade de encontrar o destino e retira a mensagem trocada de seu buffer. Apesar disso, com o aumento do buffer, os nós conseguem ter uma lista de entrega de mensagens maior e por isso incrementa sua taxa de entrega. O protocolo AntRoP manteve o *Overhead* de transmissão pelo mesmo motivo do PRoPHET, mas sua taxa de entrega foi a melhor, cerca de 1% acima do PRoPHET e 19% melhor do que o protocolo Epidêmico.

5. Conclusões e Trabalhos Futuros

Este artigo analisou protocolos de roteamento empregados em redes de emergência. Foi proposto a utilização de um algoritmo de roteamento bio-inspirado em colônia de formigas, chamado AntRoP. O protocolo AntRoP mostrou-se tolerante a falhas e desconexões de nós e de enlaces por utilizar uma marcação de caminhos baseada em feromônio. Foram avaliadas as métricas *Overhead* de transmissão, Taxa de Entrega, Latência e Número de Saltos, considerando-se a variação do alcance de transmissão, a escalabilidade da rede e o tamanho do buffer dos nós. O protocolo AntRoP apresentou um *O*verhead de transmissão até 10 vezes menor ao obtido por protocolos de referência da literatura, enquanto ao mesmo tempo apresentou latência próxima da latência ótima. Isso mostra que a utilização de rastros de feromônio obtém uma otimização das rotas, entregando mais mensagens e com uma latência reduzida. Nas próximas etapas desse trabalho pretende-se avaliar outros protocolos de roteamento empregados em redes DTN, considerar cenários maiores, analisar o consumo de energia, utilizar outros modelos de movimento e de tráfego, além de investigar como melhorar a latência do protocolo AntRoP.

Agradecimentos

Os autores agradecem o apoio financeiro das agências FAPEMIG e CNPq.

Referências

Boukerche, A. (2008). *Algorithms and protocols for wireless and mobile ad hoc networks.* Wiley - IEEE Press.

Careem, M., de Silva, C., Silva, R. D., Raschid, L., and Weerawarana, S. (2007). Demonstration of sahana: free and open source disaster management. In *DG.O*, pages 266–267.

Caro, G. D., Ducatelle, F., and Gambardella, L. M. (2004). Anthocnet: an ant-based hybrid routing algorithm for mobile ad hoc networks. In *In Proceedings of Parallel Problem Solving from Nature (PPSN) VIII*, pages 461–470. Springer-Verlag.

de Oliveira, C. T. (2008). Uma proposta de roteamento probabilístico para redes tolerantes a atrasos e desconexões. Master's thesis, Universidade Federal do Rio de Janeiro - COPPE.

Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization – artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, 1:28–39.

Gunes, M., Sorges, U., and Bouazizi, I. (2002). ARA - The Ant-Colony Based Routing Algorithm for MANETs.

Gupta, I., Birman, K. P., and van Renesse, R. (2002). Fighting fire with fire: using randomized gossip to combat stochastic scalability limits. *Quality and Reliability Engineering International*, 18(3):165–184.

Hinton, D., Klein, T. E., and Haner, M. (2005). An architectural proposal for future wireless emergency response networks with broadband services. *Bell Labs Technical Journal*, 10(2):121–138.

Jacinto, B., Vilaça, L., Kelner, J., Sadok, D., and Souto, E. (2010). 3d routing: a protocol for emergency scenarios. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, IWCMC '10, pages 519–523, New York, NY, USA. ACM.

Jain, S., Fall, K., and Patra, R. (2004). Routing in a delay tolerant network. *SIGCOMM Comput. Commun. Rev.*, 34:145–158.

Kanchanasut, K., Wongsaardsakul, T., Chansutthirangkool, M., Laouiti, A., Tazaki, H., and Arefin, K. R. (2008). Dumbo ii: a v-2-i emergency network. In *Proceedings of the 4th Asian Conference on Internet Engineering*, AINTEC '08, pages 37–38, New York, NY, USA. ACM.

Keränen, A., Ott, J., and Kärkkäinen, T. (2009). The one simulator for dtn protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Simutools '09, pages 55:1–55:10, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Lindgren, A., Doria, A., and Schelén, O. (2003). Poster: Probabilistic routing in intermittently connected networks. In *SIGMOBILE Mobile Computing and Communication Review*, page 2003.

Schmitt, T., , R.Rao, R., and Eisenberg, J. (2007). *Improving Disaster Management: The Role of IT in Mitigation, Preparedness, Response, and Recovery*. National Academy Press, Washington, DC, USA.

Spyropoulos, T. (2007). Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *In Proceedings of IEEE PerCom Workshop on Intermittently Connected Mobile Ad Hoc Networks*.

Stutzle, T., Utzle, T. S., and Dorigo, M. (1999). Aco algorithms for the traveling salesman problem.

Vahdat, A. and Becker, D. (2000). Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University.

Vivid Solutions (2009). OpenJump - The free, Java based and open source Geographic Information System for the World. http://www.openjump.org, Data de acesso 01/04/2011.

Zhang, Z. (2006). Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. *IEEE Communications Surveys & Tutorials*, 8(1):24–37.

Índice por Autor

ISSN 2177-496X

Organização:

FACOM-UFMS

UFMS

Realização:

SBC

LARC

Patrocínios:

egi.br nic.br

CNPq

CAPES

Fundect

VALSPE GROUP

SERPRO

POLYCOM

DIGITALNETBR

Google

Netware Enterprise

SICREDI Gente que coopera cresce.