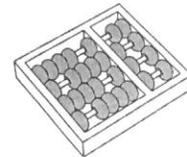


Jorge Lima de Oliveira Filho

**“Uma Arquitetura para Gerência de Mobilidade
Cognitiva em Redes sem Fio Heterogêneas”**

CAMPINAS
2013



Universidade Estadual de Campinas
Instituto de Computação

Jorge Lima de Oliveira Filho

“Uma Arquitetura para Gerência de Mobilidade Cognitiva em Redes sem Fio Heterogêneas ”

Orientador(a): Prof. Dr. Edmundo Roberto Mauro Madeira

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Doutor em Ciência da Computação.

ESTE EXEMPLAR CORRESPONDE À
VERSÃO FINAL DA TESE DEFENDIDA POR
JORGE LIMA DE OLIVEIRA FILHO, SOB
ORIENTAÇÃO DE PROF. DR. EDMUNDO
ROBERTO MAURO MADEIRA.

A handwritten signature in cursive script that reads "Edmundo Madeira".

Assinatura do Orientador(a)

CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

OL4a Oliveira Filho, Jorge Lima de, 1979-
Uma arquitetura para gerência de mobilidade cognitiva em redes sem fio heterogêneas / Jorge Lima de Oliveira Filho. – Campinas, SP : [s.n.], 2013.

Orientador: Edmundo Roberto Mauro Madeira.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Redes de computadores. 2. Computação móvel. 3. Computação ubíqua. 4. Sistemas de computação sem fio. 5. Sistemas e computação. I. Madeira, Edmundo Roberto Mauro, 1958-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: An architecture for cognitive mobility management in heterogeneous wireless networks

Palavras-chave em inglês:

Computer networks

Mobile computing

Ubiquitous computing

Wireless system computing

Systems and computing

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora:

Edmundo Roberto Mauro Madeira [Orientador]

José Ferreira de Rezende

Michele Nogueira Lima

Islene Calciolari Garcia

Nelson Luis Saldanha da Fonseca

Data de defesa: 21-06-2013

Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

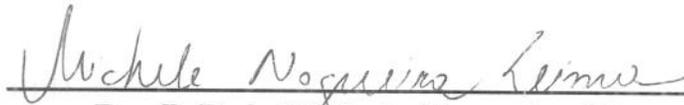
Tese Defendida e Aprovada em 21 de Junho de 2013, pela Banca
examinadora composta pelos Professores Doutores:



Prof. Dr. José Ferreira de Rezende
LAND / UFRJ



Prof^a. Dr^a. Islene Calciolari Garcia
IC / UNICAMP



Prof^a. Dr^a. Michele Nogueira Lima
DINF / UFPR



Prof. Dr. Nelson Luis Saldanha da Fonseca
IC / UNICAMP



Prof. Dr. Edmundo Roberto Mauro Madeira
IC / UNICAMP

Uma Arquitetura para Gerência de Mobilidade Cognitiva em Redes sem Fio Heterogêneas

Jorge Lima de Oliveira Filho¹

21 de junho de 2013

Banca Examinadora:

- Prof. Dr. Edmundo Roberto Mauro Madeira (Supervisor/*Orientador*)
- Prof. Dr. José Ferreira de Rezende
LAND - UFRJ
- Profa. Dra. Michele Nogueira Lima
DINF - UFPR
- Profa. Dra. Islene Calciolari Garcia
IC - UNICAMP
- Prof. Dr. Nelson Luis Saldanha da Fonseca
IC - UNICAMP

¹Suporte financeiro: Capes 2007-2008, Fapesp 2008-2011 (processo 2007/57336-0)

Abstract

The mobile nodes (MNs) have become popular nowadays, the rapid increase in the use of wireless technologies has changed the behavior of individuals and businesses in the way they communicate. Currently, there is a demand for people to be connected all time and everywhere by using their MNs to perform their tasks. It is desirable to have a ubiquitous Internet, available at any moment and everywhere. However, the integration of heterogeneous wireless networks introduces several challenges. One of the most challenging issues is service continuity during handovers. As a result of the high demand for mobility, IEEE has created a set of services and protocols in order to integrate heterogeneous networks, called IEEE 802.21. However, there is no mechanisms in the standard in order to support mobility in heterogeneous networks, such as decision algorithms and mechanisms to perform the handover. Moreover, the mobility demand requires a special management mechanism due to the elements' diversity and the dynamic environment of the heterogeneous wireless networks. In order to tackle these challenges, this work presents a cognitive mobility management architecture for heterogeneous wireless networks. This architecture is based on the IEEE 802.21 and is able to integrate different networks with different technologies and supporting mobility while maintaining service continuity when the MN performs a handover. A handover decision algorithm was developed for this architecture to select networks based on the previous experiences of MNs. This algorithm uses as decision metric the TCP loss in the MNs. The cognitive module in the architecture is an evolution of research results of our proposed algorithm for reconfiguration of cognitive parameters of the IEEE 802.11, called CogMAC. The CogMAC is a completely distributed proposal capable to monitor the MAC layer performance and react to changes to avoid performance degradation. The CogMAC is evaluated by simulations and real testbed and the results are compared with the first-WiFi algorithm. This algorithm prioritizes the connection of WiFi networks. The results also confirm the good performance of the proposed architecture.

Resumo

Atualmente há uma necessidade das pessoas estarem conectadas a todo momento e em qualquer lugar utilizando seus MNs (*Mobile Nodes*) para realizarem suas tarefas. É desejável uma Internet disponível a todo instante. Porém, a integração das redes sem fio de tecnologias diferentes apresenta vários desafios, como por exemplo, manter a continuidade dos serviços que rodam no MN quando há uma troca de rede. Percebendo a demanda crescente por mobilidade e de uma necessidade de uma Internet ubíqua, o IEEE padronizou um conjunto de serviços e protocolos com o intuito de integrar as redes de diferentes tecnologias denominado IEEE 802.21. Entretanto, não faz parte do padrão alguns mecanismos necessários para suportar mobilidade nas redes heterogêneas, como por exemplo, o algoritmo de decisão para a escolha da rede e o mecanismo de gerência de *handover*. Além disso, a demanda por mobilidade exige um mecanismo de gerência diferenciado devido à heterogeneidade dos elementos e à dinâmica do ambiente das redes sem fio heterogêneas. Para lidar com esses desafios, este trabalho apresenta uma arquitetura de gerência de mobilidade cognitiva para redes sem fio heterogêneas. Essa arquitetura é baseada no IEEE 802.21 e é capaz de integrar diversas redes com tecnologias diferentes e de suportar mobilidade mantendo a continuidade dos serviços quando o MN troca de uma rede para outra. Foi desenvolvido para esta arquitetura um algoritmo cognitivo de decisão de *handover* que seleciona as redes baseado nas experiências prévias dos MNs. Esse algoritmo utiliza como métrica de decisão as perdas dos fluxos TCP configurados nos MNs. O módulo cognitivo embutido na arquitetura foi uma evolução do resultado de pesquisa da nossa proposta de um algoritmo cognitivo para reconfiguração dos parâmetros do IEEE 802.11, denominado CogMAC. O CogMAC é uma proposta descentralizada, ou seja, os MNs que o implementam não possuem uma visão total da rede, cada MN individualmente é capaz de monitorar o desempenho e reconfigurar os parâmetros da subcamada MAC para evitar degradação do desempenho devido a mudanças nas condições da rede. O CogMAC é avaliado através de simulações e de um *testbed* real e os resultados comparados confirmam sua eficácia. Já a arquitetura de gerência cognitiva é avaliada através de simulações, e os resultados são comparados com o algoritmo que prioriza a conexão das redes WiFi. Os resultados obtidos confirmam também o bom desempenho da arquitetura proposta.

Agradecimentos

Prof. Edmundo Madeira. Muito obrigado por ter sido meu orientador, uma parceria de seis anos onde muito aprendi.

Minha família. Pelo apoio e suporte.

Minha namorada. Muito obrigado pela paciência e o apoio.

Amigos. Pelo apoio fundamental nesta longa jornada.

Fapesp (processo número: 2007/57336-0) e Capes. Pelo suporte financeiro.

Sumário

Abstract	ix
Resumo	xi
Agradecimentos	xiii
1 Introdução	1
2 Conceitos Fundamentais	5
2.1 Gerência de Mobilidade	5
2.2 Qualidade de Serviço - QoS	7
2.3 WiFi	8
2.4 WiMAX	9
2.5 Celular	11
2.6 IEEE 802.21	12
2.7 SAW - Simple Additive Weight	14
2.8 CogProt - Arcabouço Cognitivo	15
3 Trabalhos Relacionados	17
3.1 Redes cognitivas	17
3.2 Gerência de mobilidade	21
4 Arcabouço CogMAC	29
4.1 Descrição do CogMAC	29
4.2 Implementação no NS-2	33
4.2.1 Experimentos realizados no NS-2	34
4.2.2 Resultados obtidos a partir dos experimentos realizados no simulador	35
4.3 Implementação do CogMAC no <i>driver</i> de dispositivo real	38
4.3.1 Experimentos realizados no <i>testbed</i>	43
4.3.2 Resultados obtidos a partir dos experimentos realizados no <i>testbed</i> .	43

4.4	Comparação - experimento real com experimento no <i>Testbed</i>	47
5	Arquitetura de Gerência de Mobilidade Cognitiva	49
5.1	Algoritmo de Decisão	49
5.2	Simulador de <i>handover</i> (Handsim)	54
5.3	Experimentos de simulação realizados com o algoritmo de decisão	56
5.4	Resultados obtidos com o algoritmo de decisão	57
5.4.1	Taxa de Utilização	57
5.4.2	Taxa de Bloqueio	60
5.5	Módulo de <i>handover</i> cognitivo - CogHand	61
5.6	Arquitetura de gerência de mobilidade cognitiva	63
5.7	Experimentos realizados no NS-2	69
5.8	Resultados obtidos a partir dos experimentos realizados no NS-2	70
6	Conclusão	77
	Referências Bibliográficas	80

Lista de Tabelas

2.1	Tabela de decisão utilizada como entrada para o método SAW	14
3.1	Comparação das arquiteturas para gerência de mobilidade	25
3.2	Comparação dos algoritmos de decisão de <i>handover</i>	28
4.1	Armazenamento da métrica de qualidade na base de dados local associada com os parâmetros do IEEE 802.11	31
5.1	Taxa de bloqueio - 26MNs	60
5.2	Taxa de bloqueio - 50 MNs.	60
5.3	Taxa de bloqueio - 70 MNs.	60
5.4	Taxa de bloqueio - 100 MNs.	60
5.5	Pontuação das redes calculadas com o método SAW	63
5.6	Perda de pacotes - CogHand X algWiFi (%)	76

Lista de Figuras

1.1	Cenário com tecnologias de redes sem fio heterogêneas [1]	2
2.1	Sistema celular (baseada em [40])	12
2.2	Serviços MIH (baseada em [31])	13
2.3	Visão geral do CogProt[57]	15
4.1	Visão geral do CogMAC	30
4.2	Distribuição normal utilizada pelo gerador de números aleatórios com média $P_{\{def\}}$ para a criação do novo parâmetro $P_{\{new\}}$	33
4.3	Topologia de rede infraestruturada com os MNs distribuídos obedecendo uma distribuição uniforme dentro do alcance do AP.	34
4.4	MAC <i>access delay</i> médio em função do número de MNs numa rede infraestruturada utilizando tráfego FTP para cada parâmetro de <i>retry limit</i>	35
4.5	MAC <i>access delay</i> médio ao longo do tempo numa rede infraestruturada utilizando tráfego FTP.	36
4.6	MAC <i>throughput</i> médio em função do número de MNs numa rede infraestruturada utilizando tráfego FTP para cada parâmetro de <i>retry limit</i>	36
4.7	MAC <i>throughput</i> médio ao longo do tempo numa rede infra-estruturada utilizando tráfego FTP.	37
4.8	Frequência dos valores selecionados do parâmetro <i>retry limit</i> para um cenário com 50 MNs.	37
4.9	Visão conceitual do CogMAC no GNU/Linux <i>testbed</i>	38
4.10	Registro de duas variáveis em <i>/proc/sys/cogmac</i>	40
4.11	Principais modificações realizadas no <i>kernel</i> do GNU/Linux separadas por camadas TCP/IP para a implementação do arcabouço CogMAC.	42
4.12	Topologia de rede do <i>testbed</i>	44
4.13	Frequência de <i>retry limits</i> utilizados com um limite de <i>access delay</i> em 300 ms.	44
4.14	Frequência de <i>retry limits</i> utilizados com um limite de <i>access delay</i> em 400 ms.	45

4.15	Frequência de <i>retry limits</i> utilizados com um limite de <i>access delay</i> em 1000 ms.	45
4.16	MAC access delay médio para diferentes <i>thresholds</i>	46
4.17	Vazão com diferentes <i>access delay thresholds</i>	46
4.18	Vazão com o CogMAC ativo comparada com o CogMAC inativo	47
5.1	Diagrama de sequência do IEEE 802.21	51
5.2	<i>Thresholds</i> de prioridade	52
5.3	Arquitetura do simulador de eventos discretos HandSim.	54
5.4	Distribuição dos MNs na área de cobertura das redes Celular, WiMAX e WiFi	57
5.5	Taxa de utilização de rede comparando o ADH com o WiFi utilizando 26 MNs	58
5.6	Taxa de utilização de rede comparando o ADH com o WiFi utilizando 50 MNs	58
5.7	Taxa de utilização de rede comparando o ADH com o WiFi utilizando 70 MNs	59
5.8	Taxa de utilização de rede comparando o ADH com o WiFi utilizando 100 MNs	59
5.9	Funcionamento do CogHand no MN e no IS	62
5.10	Entidades participantes da arquitetura de mobilidade cognitiva	65
5.11	Arquitetura de gerência de mobilidade cognitiva	66
5.12	Diagrama de classes parcial do IEEE 802.21	68
5.13	Diagrama de sequência da arquitetura de mobilidade cognitiva.	69
5.14	Cenário utilizando uma rede WiFi e uma rede WiMAX	70
5.15	Vazão TCP - cenário contendo 1 MN	71
5.16	Vazão TCP - cenário contendo 10 MNs	71
5.17	Vazão TCP - cenário contendo 20 MNs	72
5.18	Vazão TCP - cenário contendo 30 MNs	72
5.19	Vazão TCP - cenário contendo 40 MNs	72
5.20	Vazão TCP - cenário contendo 50 MNs	73
5.21	Atraso fim a fim - 1MN	73
5.22	Atraso fim a fim - 10MNs	74
5.23	Atraso fim a fim - 20MNs	74
5.24	Atraso fim a fim - 30MNs	75
5.25	Atraso fim a fim - 40MNs	75
5.26	Atraso fim a fim - 50MNs	75

Capítulo 1

Introdução

A popularização da Internet mudou o comportamento dos indivíduos e das empresas na forma de comunicação. É crescente a necessidade das pessoas estarem sempre conectadas para realizar suas tarefas. É desejável uma Internet ubíqua, que esteja disponível a todo o momento. Novos serviços que surgem na Internet precisam estar sempre disponíveis e com algum requisito de Qualidade de Serviço (*Quality of Service* - QoS) como, por exemplo, rastreadores de carro, monitoramento remoto de pacientes com suporte a vídeo, etc. A tecnologia de comunicação sem fio surge como uma alternativa flexível e de baixo custo para redes de acesso, possibilitando a mobilidade de usuários.

Atualmente existe uma grande quantidade de dispositivos aptos a acessar redes sem fio, através de uma grande diversidade de tecnologias de acesso. Cada tecnologia tem suas vantagens e desvantagens de acordo com o cenário utilizado, por exemplo, o uso da rede WiFi (*Wireless Fidelity*) [26, 24] é ideal para ambientes internos e inadequado para o uso externo. Já a tecnologia WiMAX (*Worldwide Interoperability for Microwave Access*) [43] é adequada para ambientes externos com alcance relativamente longo. A Figura 1.1 exibe um cenário atual onde um nó móvel (*Mobile Node* - MN) precisa interagir com diversas tecnologias de redes sem fio.

Um dos desafios de pesquisa para a próxima geração de redes sem fio é o desenvolvimento de técnicas inteligentes de gerência de mobilidade que tirem vantagem da tecnologia baseada em IP para alcançar *roaming* global entre várias tecnologias de acesso sem fio [9]. Muitas questões sobre a integração (composição) de redes ainda são desafiadoras para a comunidade de pesquisa, principalmente o tempo gasto por um dispositivo ao trocar de rede (*handover*) mantendo a conexão ativa (continuidade de sessão) [33]. Alguns trabalhos propostos têm alta carga de sinalização e um atraso de *handover* alto [9]. As soluções propostas deveriam abordar novos projetos de protocolos e mecanismos que levam em consideração informações e funções de múltiplas camadas da arquitetura de rede (*cross-layer design*).

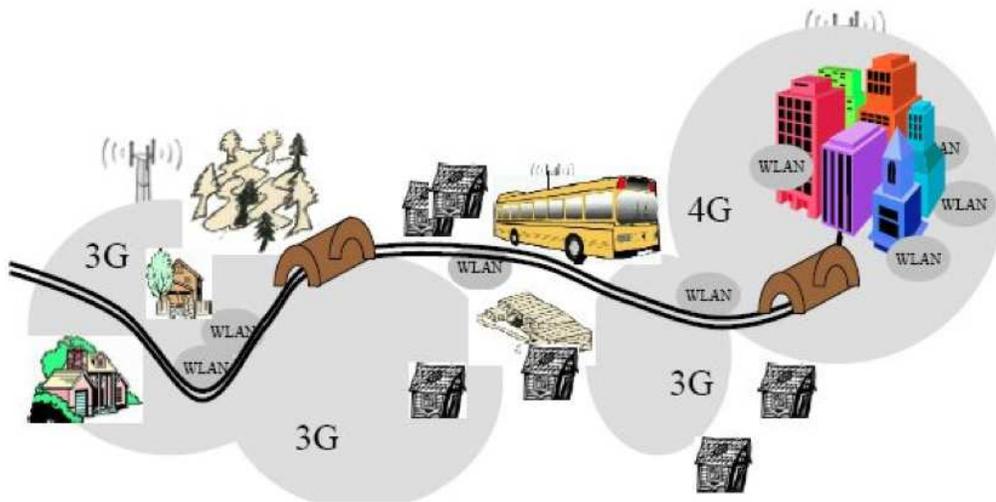


Figura 1.1: Cenário com tecnologias de redes sem fio heterogêneas [1]

A tendência mundial é que todas as redes sem fio sejam baseadas no protocolo IP e que estas novas redes suportem múltiplas tecnologias de acesso. As redes com estas características também são conhecidas como sistemas sem fio de próxima geração (*Next-Generation Wireless System* - NGWS) [9].

Com o objetivo de promover as redes de próxima geração, o IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) criou um padrão para auxiliar os MNs a estarem sempre conectados, sejam em redes sem fio ou cabeadas. O padrão conhecido como 802.21 [46] é adequado para redes padrão IEEE ou não. A proposta utiliza informações de múltiplas camadas das redes (*cross-layer*) para auxiliar o mecanismo de *handover* a executá-lo. Estas informações sobre as redes ficam armazenadas num servidor de informação (*Information Server* - IS). Existem muitos componentes que a proposta não contempla, dentre eles, o algoritmo de decisão do *handover* para a escolha da melhor rede que o dispositivo móvel irá se conectar. Com esta proposta o IEEE espera promover a mobilidade dos MNs entre redes de diferentes tecnologias de forma transparente.

A crescente demanda por mobilidade nos tempos atuais exige um mecanismo de gerência diferenciado devido à heterogeneidade dos elementos e à dinâmica do ambiente. As soluções de gerência tradicionais não são apropriadas para estes cenários devido ao seu modelo centralizado [57]. Para lidar com estes desafios de gerência foi proposto o conceito de redes autônomicas [17] a partir do paradigma de computação autônomicas. Este paradigma define princípios para o projeto de sistemas com capacidade de autogerência. Neste contexto, o conceito de redes autônomicas consiste em desenvolver mecanismos de autogerência para redes de comunicação, tornando-as capazes de se adaptarem ao ambiente com a finalidade de otimizar seu desempenho. O modelo de referência de um sistema au-

tonômico define que as funcionalidades de autogerência podem ser implementadas através de um laço de controle [81]. No entanto, as etapas desse laço são definidas de maneira abstrata. O projeto de soluções para efetivamente prover funcionalidades de autoconfiguração ainda é um grande desafio [28]. Dentre os principais desafios destacam-se: a representação do conhecimento, a obtenção e a gerência de informações de contexto e a tomada de decisão de forma descentralizada.

Identificada a necessidade de uma alta mobilidade e os desafios apresentados anteriormente, este trabalho propõe uma arquitetura para a gerência de mobilidade com princípios cognitivos para redes multiacesso sem fio que considera a necessidade das pessoas se deslocarem, enquanto fazem uso de suas aplicações, sem perder a conexão ao longo do percurso (continuidade de sessão), ainda que alternem entre diferentes redes de acesso. A arquitetura utiliza dentre outros critérios, os parâmetros de QoS para decidir efetuar uma troca de rede. Se por um lado as redes móveis atuais têm limitações de cobertura, por outro, as diversas tecnologias de acesso não se integram. A arquitetura proposta tem como objetivo resolver esses aspectos criando uma infraestrutura comum aos usuários fazendo uso de várias tecnologias de acesso de forma transparente.

Para construir a arquitetura de gerência de mobilidade cognitiva, foi necessário pesquisar na literatura sistemas autonômicos com funcionalidades de autogerência. Desta forma foi desenvolvido o CogMAC baseado no CogProt [52], um arcabouço cognitivo para autoconfiguração de protocolos de comunicação. O CogMAC é um arcabouço cognitivo que reconfigura os parâmetros do IEEE 802.11 de forma autonômica. Ele foi criado para estudar os conceitos cognitivos com o objetivo de viabilizar a construção da arquitetura proposta neste trabalho. O CogMAC foi implementado no ns-2 [53] e também num *driver* de uma placa de rede real [57, 65]. Os resultados positivos obtidos com estas implementações motivaram o uso do CogMAC na arquitetura de gerência de mobilidade cognitiva.

A arquitetura proposta usa informações de múltiplas camadas do TCP/IP para auxiliar o mecanismo de *handover* a executar uma troca de rede. Ela é composta do IEEE 802.21 juntamente com uma adaptação do CogMAC, denominada CogHand como arcabouço cognitivo. O IEEE 802.21 é usado para obter informações das redes disponíveis para os MNs e também coordenar o mecanismo de *handover* a realizar a troca de rede. Já o CogHand é usado no mecanismo de decisão do IEEE 802.21, e é responsável por decidir a qual rede se conectar baseado em parâmetros de QoS e experiências prévias de desempenho. Os conceitos do Algoritmo de Decisão de *Handover* (ADH) [66], criado para ser usado no IEEE 802.21, foram incorporados no desenvolvimento do CogHand.

As principais contribuições deste trabalho são:

- O detalhamento do estado da arte em algoritmos cognitivos com princípios autonômicos e algoritmos de decisão de *handover* em redes sem fio heterogêneas in-

cluindo uma revisão bibliográfica;

- Proposta do CogMAC, um mecanismo contendo um algoritmo cognitivo [53] que reconfigura os parâmetros da camada de enlace das redes sem fio IEEE 802.11 de forma autônoma;
- Implementação do CogMAC no simulador Network Simulator (ns-2)[5];
- Implementação real do CogMAC [65] num *driver* de placa de rede dentro do projeto Madwifi (*Multiband Atheros Driver for Wireless Fidelity*)[2], conhecido como *ath5k*, que funciona com *chips* Atheros;
- Avaliação de desempenho do CogMAC através de uma comparação entre os resultados obtidos nas simulações utilizando o ns-2 e utilizando resultados obtidos de medições reais realizadas num *testbed*;
- Proposta do algoritmo de decisão de *handover* [66] que utiliza os dados disponibilizados pelo servidor de informação (IS) do 802.21 para auxiliar o dispositivo móvel a escolher em qual rede se conectar baseado nas preferências do usuário.
- Implementação do simulador de eventos discretos (Handsim) e avaliação do desempenho do algoritmo de decisão com este simulador;
- Algoritmo CogHand e a arquitetura para o gerenciamento de mobilidade com princípios cognitivos;
- Implementação e avaliação desta arquitetura no simulador ns-2 utilizando o módulo *Seamless and Secure Mobility* do NIST [5].

Este trabalho está organizado como segue. No Capítulo 2 são apresentados os conceitos de gerência de mobilidade, IEEE 802.21 e gerência autônoma que são necessários para o entendimento da tese. O estado da arte nestas áreas, incluindo os trabalhos relacionados, é apresentado no Capítulo 3. No Capítulo 4 é apresentado o arcabouço CogMAC, detalhando os experimentos simulados e os experimentos reais num *testbed*. No Capítulo 5 a arquitetura proposta é apresentada incluindo os detalhes do algoritmo cognitivo CogHand e do simulador juntamente com os resultados obtidos. Por fim, o Capítulo 6 apresenta as conclusões.

Capítulo 2

Conceitos Fundamentais

Este capítulo aborda os conceitos fundamentais sobre os assuntos discutidos nesta tese. Na Seção 2.1, são apresentados os conceitos de gerência de mobilidade incluindo a descrição de um *handover* e suas particularidades. A Seção 2.2 apresenta os conceitos básicos de Qualidade de Serviço. As Seções 2.3 e 2.4 apresentam as redes sem fio regidas pelos padrões IEEE 802.11 e IEEE 802.16, respectivamente. Na Seção 2.5 é apresentada uma breve descrição de uma rede celular. Na Seção 2.6 é apresentado o padrão 802.21, desenvolvido pelo IEEE para facilitar a gerência de mobilidade e indicado para redes que seguem padrões do IEEE (redes IEEE) ou não. Este padrão é utilizado na composição da arquitetura proposta. Na Seção 2.7 o método SAW (*Simple Additive Weight*) de decisão multicritério é apresentado. Esse método é utilizado na composição do algoritmo de decisão proposto na Seção 5.1. E por último, a Seção 2.8 descreve o arcabouço cognitivo CogProt, uma solução descentralizada para reconfiguração dinâmica dos protocolos que foi utilizada como modelo para criação do CogMAC.

2.1 Gerência de Mobilidade

A gerência de mobilidade contém dois componentes principais [8]: gerência de localização e gerência de *handover*. A gerência de localização permite ao sistema rastrear a localização dos MNs entre comunicações consecutivas. A gerência de *handover* permite que MNs troquem de rede mantendo a conexão ativa.

Nos sistemas de próxima geração (*Next Generation - NG*) existem dois tipos de *roaming* para os MNs: Intrassistema (Intradomínio) e intersistema (Interdomínio). O *roaming* intrassistema acontece na movimentação de MNs entre células diferentes do mesmo sistema. Técnicas de gerência de mobilidade intrassistema são baseadas nos mesmos protocolos e interfaces de rede. Já o *roaming* intersistema ocorre na movimentação de MNs entre *backbones*, protocolos e tecnologias diferentes. Baseado no *roaming* intra ou intersis-

tema, gerências de *handover* e de localização podem ser classificadas como gerência intra ou intersistema de *handover* e gerência intra ou intersistema de localização [9].

Devido à heterogeneidade dos protocolos e das tecnologias envolvidas, a gerência intersistema de *handover* é muito mais difícil de se implementar em relação à gerência intrassistema. Diversos trabalhos propostos pela comunidade científica defendem a implementação em diferentes camadas da pilha de protocolos TCP/IP [76]. As possíveis alternativas são implementações na camada de aplicação, rede, enlace e em múltiplas camadas (*cross-layer*) [9].

Handover

Quando um dispositivo conectado em um ponto de acesso se afasta de sua área de cobertura, o nível do sinal sofre uma degradação. Ao se aproximar de outro ponto de acesso, com um nível de sinal mais forte, é necessário um mecanismo na rede para manter o estado da conexão, transferindo a responsabilidade pela comunicação para o novo ponto de acesso. O mecanismo que faz a transferência de conexão de um dispositivo de um ponto de acesso para outro é chamado mecanismo de *handover*.

Adicionalmente, a decisão de executar um *handover* pode ser feita por um agente situado no dispositivo do usuário, baseada em políticas bem definidas tal como largura de banda, custo, segurança, cobertura da rede, QoS ou até mesmo pelas preferências do usuário [76].

Existem dois tipos de *handovers* [19]. Os *handovers* que ocorrem entre pontos de acessos de mesma tecnologia são chamados de *handovers* horizontais. *Handovers* que ocorrem entre pontos de acessos pertencentes a redes com tecnologias diferentes (Ex. WiFi para WiMAX) são chamados de *handovers* verticais. Assim, um *handover* vertical é usado entre células heterogêneas de rede de acesso, que diferem em muitos aspectos, tais como largura de banda, frequência do sinal, etc. Essas características particulares de cada rede fazem a implementação de *handovers* verticais muito mais difícil se comparada com a dos *handovers* horizontais.

As métricas de decisão de *handover* são usadas como critério para determinar quando é necessário executar um *handover*. Os *handovers* tradicionais eram apenas baseados na qualidade do sinal do rádio e na disponibilidade de canais. Nas novas gerações de redes sem fio heterogêneas, novas métricas de *handovers* são necessárias para uma alta mobilidade do usuário e para minimizar o atraso de *handover*. Em [60] são propostas algumas métricas de decisão de *handover*:

- **Tipo de serviço:** Diferentes tipos de serviço requerem diferentes tipos de confiabilidade, limites de atraso e taxa de transmissão.

- **Custo:** Custo é um fator importante para os usuários, como diferentes redes podem ter diferentes estratégias de cobrança, a escolha da rede baseada no custo influi na escolha do usuário na execução do *handover*.
- **Condições da rede:** Os parâmetros relacionados com o estado da rede como tráfego, disponibilidade de banda, atraso e congestionamento (perda de pacotes) precisam ser considerados para o uso eficaz da rede. O uso das informações relacionadas ao estado da rede para a escolha do *handover* pode ser útil para o balanceamento de carga entre diferentes redes possibilitando diminuir o congestionamento.
- **Desempenho do sistema:** Para garantir o desempenho do sistema, uma variedade de parâmetros pode ser empregada na decisão do *handover*. Por exemplo, características da propagação do canal, perda de caminho, interferência (*Signal-to-noise* - *SNR*) e erro de transmissão de *bit*. Além disso, a carga de bateria pode ser um fator importante para certos usuários. Quando o nível de bateria for mínimo, o usuário pode escolher fazer o *handover* baseado no consumo de bateria do MN, por exemplo, numa rede *bluetooth ad hoc*.
- **Condições do dispositivo móvel:** As condições do MN incluem fatores dinâmicos como velocidade, modelo de mobilidade, localização e histórico de movimentação.
- **Preferências do usuário:** As preferências do usuário podem ser usadas para fornecer pedidos especiais para a rede.

2.2 Qualidade de Serviço - QoS

Verifica-se hoje um crescimento do número de aplicações que utilizam o conjunto de protocolos TCP/IP, principalmente das aplicações multimídia, as quais têm requisitos mínimos para que apresentem um bom funcionamento. Como o projeto inicial das redes IP não previa essa evolução, foram desenvolvidos vários mecanismos para contornar este problema [64].

Esses novos requisitos diferenciados necessários para as aplicações multimídia em rede levam à noção e necessidade de garantias da QoS em redes TCP/IP. Com objetivo de atender às necessidades de QoS deste novo conjunto de aplicações, surgiram diversas arquiteturas como IntServ [16], DiffServ [15] e MPLS [68]

O termo Qualidade de Serviço pode ter várias definições, dependendo do autor. Segundo [30], QoS é o “conjunto de ferramentas disponível ao administrador da rede para assegurar certa garantia a qual um nível mínimo de serviço irá ser provido a certo tráfego. Já [59] define QoS como sendo “requisito de uma aplicação expresso por um conjunto de

parâmetros que devem ser providos pela rede sobre o fundamento fim-a-fim para preservar o comportamento adequado da operação de uma aplicação e uma “satisfação” do usuário final”.

Os parâmetros básicos de especificação da QoS são os seguintes [59]: Atraso, variação de atraso (*jitter*), largura de banda, perda de pacote e confiabilidade. O conjunto de parâmetros com seus respectivos valores especifica um conjunto de regras e um perfil de tráfego que ajudam a definir a QoS necessária para as aplicações.

2.3 WiFi

A rede local sem fio IEEE 802.11 tornou-se o padrão mais popular entre as tecnologias de acesso sem fio. Hoje já é possível acessá-las em aeroportos, cafés, instituições e esquinas. Conhecida também como WiFi (*Wireless Fidelity*), esta tecnologia já é uma das mais importantes, utilizada principalmente como rede de acesso para a Internet [54].

Há vários adendos para o padrão IEEE 802.11, entre eles o 802.11b, o 802.11a e o 802.11g. O adendo 802.11a opera na faixa de frequência desde 5,1 até 5,8 GHz e alcança uma velocidade de 54 Mbps. O adendo 802.11b alcança até 11 Mbps e opera na faixa de frequência desde 2,4 até 2,485 GHz. Por fim o adendo 802.11g, atualmente o mais utilizado, alcança até 54 Mbps e opera na faixa desde 2,4 até 2,485 GHz.

O componente principal do 802.11 é o conjunto básico de serviço (*Basic Service Set - BSS*). Um BSS contém uma ou mais estações sem fio e um ponto de acesso (*Access Point - AP*). As estações móveis são associadas com o AP. O AP conecta suas estações com uma infraestrutura. O protocolo MAC básico do IEEE 802.11 é a função de coordenação distribuída (*Distributed Coordination Function - DCF*). O DCF usa um esquema baseado no CSMA (*Carrier Sense Multiple Access*) que escuta o meio antes de transmitir, não existe nenhum controle central e as estações concorrem pelo tempo no ar [78]. O 802.11 também define um esquema (*Collision Avoidance - CA*) para diminuir a probabilidade de colisão.

As redes 802.11 em sua padronização inicial não ofereciam QoS. Para oferecer QoS, o IEEE criou o adendo 802.11e [25]. Este novo adendo define melhoramentos na subcamada MAC, introduzindo o EDCF (*Enhanced Distributed Coordination Function*) e a HCF (*Hybrid Coordinator Function*). Estações que operam sob 802.11e são agora chamadas de QSTAs (*QoS Stations*) e o BSS é chamado de QBSS (*QoS Basic Service Set*).

A HCF possui um coordenador híbrido (HC) responsável pela funcionalidade de QoS dentro do QBSS. O EDCF é utilizado no período de contenção (*Contention Period - CP*) enquanto o HCF é utilizado em ambos os períodos (com contenção e livre de contenção). A extensão EDCF tem a finalidade de complementar o modo de acesso DCF do padrão 802.11 para prover diferenciação dos serviços utilizando um controle distribuído. Tanto

o mecanismo de acesso baseado em contenção quanto o mecanismo livre de contenção são providos pelo HC no CP. Esta técnica permite disponibilizar até 8 classes de tráfego em uma estação. Cada QSTA pode ter até 8 categorias de acesso, ou seja, 8 filas na subcamada MAC para suportar até 8 prioridades de usuários [35].

Os parâmetros de QoS utilizados pelas estações são definidos pelo HC sempre com base em 8 prioridades de usuários. Um conjunto de parâmetros de QoS são usados para informar às QSTAs os parâmetros necessários para o estabelecimento da funcionalidade de QoS.

Uma descrição detalhada do padrão IEEE 802.11 pode ser encontrada em [24] e do adendo 802.11e em [25].

2.4 WiMAX

WiMAX ou Wireless MAN (*Worldwide Interoperability for Microwave Access*) é o nome popular para o padrão IEEE 802.16. O padrão oferece uma alternativa de acesso a banda larga tradicional, como sistemas coaxiais utilizando modem a cabo, ADSL (*Assymmetric Digital Subscriber Line*) e fibra ótica. O WiMAX oferece acesso sem fio em alta velocidade numa ampla área geográfica. Além disso, não há necessidade de investimento em uma infraestrutura de alto custo (como acontece numa rede de acesso banda larga cabeada). Por todos esses fatores, essa tecnologia, de fato, proporciona um acesso banda larga ubíquo com um custo mínimo [32].

O padrão define a interface aérea e o protocolo de acesso ao meio para redes metropolitanas sem fio, funciona na frequência de 10 a 66 GHz em sua versão inicial (2002) e é necessário que os pares estejam em linha de visada direta (*Line of Sight - LoS*). A segunda versão do padrão, o 802.16a (2003) não requer transmissão com visada direta (*Non Line of Sight - NLOS*) e funciona também nas frequências de 2 a 11 GHz. O alcance médio do WiMAX é cerca de 50 km e as taxas de transferência por volta de 72 Mbps.

Em Dezembro de 2005 o IEEE aprovou o padrão 802.16e [45]. Este padrão acrescenta características e atributos ao padrão original que são necessários para suportar mobilidade no WiMAX.

O WiMAX móvel oferece escalabilidade tanto na tecnologia de rádio utilizada quanto na arquitetura de rede escolhida, assim provendo um ótimo grau de flexibilidade na construção e nos serviços oferecidos pela rede. Algumas características oferecidas pelo WiMAX móvel são [47]:

- **Alta taxa de transmissão:** Implementa técnicas de MIMO (*Multiple-Input Multiple-Output*), as quais utilizam múltiplas antenas, alcançando taxas de até 63 Mbps.

- **Qualidade de Serviço:** Define classes de serviços que podem ser mapeadas para o código DSCP do DiffServ ou rótulos do MPLS possibilitando uma garantia de QoS fim a fim para as aplicações.
- **Escalabilidade:** o padrão é capaz de operar em frequências desde 2,4 GHz até 66 GHz, não significando que o mesmo rádio será capaz de operar em todas as frequências simultaneamente. Os fabricantes terão a opção de usar o mesmo protocolo de modulação. O WiMAX móvel é capaz de operar com diferentes canais desde 1,25 até 20 MHz para obedecer uma variedade de requisitos do espectro em várias partes do mundo.
- **Segurança:** suporta autenticação baseada em EAP, baseada em AES-CCM, e esquema de controle de mensagem baseado em CMAC e HMAC.
- **Mobilidade:** WiMAX móvel suporta *handover* com atrasos de menos de 50ms.

A QoS é oferecida na subcamada MAC do WiMAX via fluxo de serviço. Um fluxo de serviço é um fluxo unidirecional de pacotes associado com um conjunto particular de parâmetros de QoS. Os fluxos de serviço são definidos nas duas direções de transmissão (*uplink* e *downlink*). A estação do assinante (*Subscriber Station - SS*) e a estação base (*Base Station - BS*) garantem a QoS de acordo com o conjunto de parâmetros de QoS definidos para o fluxo. Antes de prover um certo tipo de serviço de dados, primeiramente a estação base e o terminal do usuário estabelecem um enlace lógico unidirecional entre suas subcamadas MAC, chamada de conexão. A saída MAC associa os pacotes que atravessam a interface MAC a um serviço de fluxo para então serem transmitidos. Os parâmetros de QoS associados com o serviço de fluxo definem a ordem de transmissão e o escalonamento na interface aérea [47].

Para atender aos requisitos das novas aplicações multimídia tais como videoconferência e voz sobre IP (VoIP), o padrão define mecanismos de sinalização entre a estação base e a estação cliente e 4 níveis de serviço [22]:

- **UGS (*Unsolicited Grant Service*):** Projetado para aplicações em tempo real que têm requisitos de baixo atraso, as quais geram pacotes de tamanho fixo em intervalos periódicos, ou seja tráfego CBR (*Constant Bit Rate*).
- **rtPS (*Real Time Packet Service*):** Indicado para aplicações de tempo real que geram pacotes de tamanho variável em intervalos periódicos, por exemplo, *streaming* de áudio ou vídeo, e VoIP com supressão de silêncio.
- **nrtPS (*Non Real-time Packet Service*):** Projetada para oferecer suporte às aplicações que não sejam de tempo real e que gerem pacotes de tamanho variável em intervalos periódicos, por exemplo, FTP e email.

- **BE (*Best Effort*)**: usado para aplicações que não têm nenhum requisito de QoS.

Embora esses serviços forneçam a base para empregar QoS, para que essa tarefa seja realizada de maneira eficiente, os fabricantes de equipamentos WiMAX devem projetar e implementar um conjunto de mecanismos não especificados no padrão. Tais mecanismos incluem policiamento e moldagem de tráfego (*traffic shaping*), controle de admissão e escalonamento. Trabalhos que tratam o escalonamento nas redes WiMAX podem ser encontrados em [37, 36].

2.5 Celular

Atualmente sistemas de celulares são extremamente lucrativos e populares ao redor do mundo. Eles provêm comunicação de voz e dados numa cobertura regional, nacional e internacional. Estes sistemas foram inicialmente projetados para funcionar em terminais móveis dentro de veículos com antenas montadas em seus tetos. Nos dias atuais estes sistemas são concebidos para funcionar em aparelhos portáteis e leves, funcionam dentro e fora de construções e dentro de veículos em velocidades elevadas [40].

Uma rede celular é formada a partir de células de rádio com alcance limitado e cada uma dessas células possui uma ou mais estação base (BS). A união de várias destas células de rádio permite a criação de um sistema com uma ampla cobertura permitindo que os MNs se movimentem entre estas células mantendo a continuidade da sessão. A movimentação dos MNs entre as células só é possível graças aos mecanismos de *handover*.

A premissa básica na concepção dos sistemas de celulares é o reuso da frequência que explora o fato do desvanecimento do sinal à medida que o MN se afasta da estação base [40]. A área de cobertura é dividida em células não sobrepostas onde um conjunto de canais é associado com cada célula. Este mesmo conjunto de canais é utilizado em uma outra célula, não vizinha, como ilustra a Figura 2.1, onde C_i é o conjunto de canais utilizados numa determinada célula. A operação dentro de uma célula é controlada por uma BS centralizada que recebe as conexões de todos os MNs presentes na célula.

O ITU (*International Telecommunication Union*) [48] é o órgão que define os principais padrões para os sistemas celulares, como por exemplo o IMT-2000 (*International Mobile Telecommunications-2000*) [49] também conhecido como sistema de terceira geração (3G), que inclui o GSM (*Global System for Mobile Communications*) e o UMTS (*Universal Mobile Telecommunications System*).

Sistemas celulares estão evoluindo cada vez mais para suportar altas taxas de transmissão de pacotes em suas redes, incluindo WiMAX e UMB (*Ultra Mobile Broadband*) [7]. Embora debates continuem tratando da necessidade do uso destas tecnologias de acesso, outros fatores como custo, problemas de compatibilidade com as tecnologias anteriores e

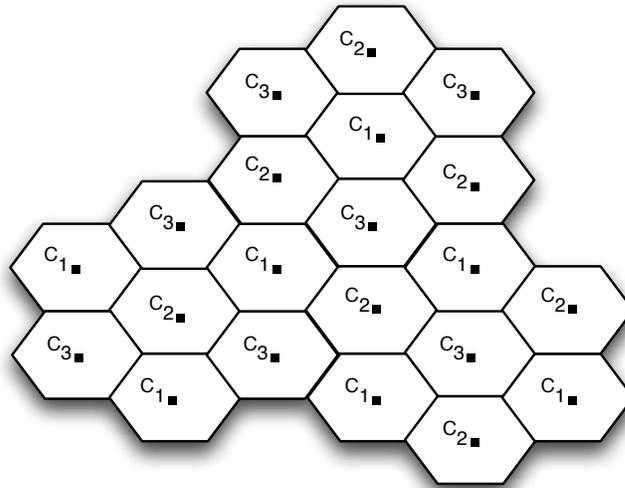


Figura 2.1: Sistema celular (baseada em [40])

interesses comerciais levam a crer que a indústria não adotará uma única tecnologia como padrão. Como consequência, o cenário para sistemas sem fio será muito heterogêneo, fazendo que mecanismos de interligações destas tecnologias sejam essenciais para um sistema ubíquo.

2.6 IEEE 802.21

Como já discutido anteriormente, devido à heterogeneidade dos protocolos e tecnologias envolvidas, um *handover* vertical é muito mais desafiador para ser executado em comparação ao *handover* horizontal. Para resolver esses aspectos de gerência heterogênea de mobilidade o IEEE padronizou o 802.21. Este padrão provê um arcabouço para auxiliar no processo de *handover* e também define um mecanismo denominado de *Media Independent Handover* (MIH) que suporta *handover* sem costuras (*seamless*) entre redes heterogêneas. O modelo projetado por este padrão é muito consistente com os cenários de redes móveis atuais e os cenários previstos para o futuro pelos especialistas [33].

O mecanismo de MIH oferece três serviços principais:

- Serviço de Evento (*Media-Independent Event Service - MIES*)
- Serviço de Comando (*Media-Independent Command Service - MICS*)
- Serviço de Informação (*Media-independent Information Service - MIIS*)

A Figura 2.2 ilustra a camada de serviços do IEEE 802.21 e sua localização na pilha de protocolo do TCP/IP. A especificação define uma camada de serviços entre as camadas de rede e de enlace. Esta camada é denominada de *Media Independent Handover Function*. A seguir é dada uma breve descrição dos serviços oferecidos pelo IEEE 802.21.

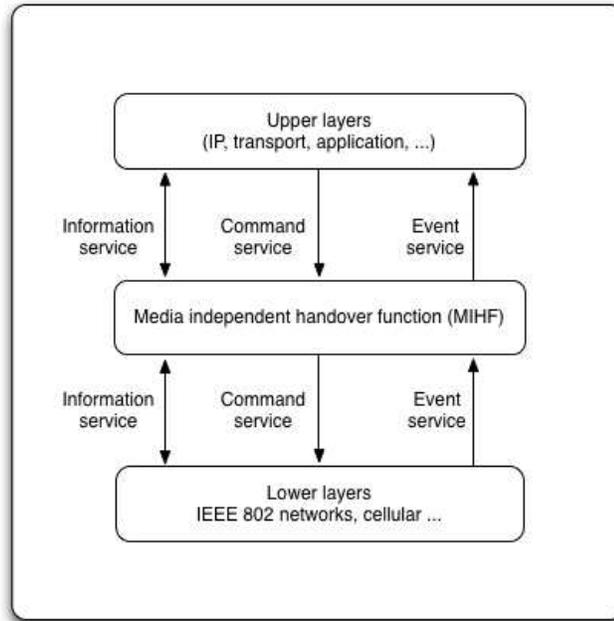


Figura 2.2: Serviços MIH (baseada em [31])

O principal papel do MIES é detectar eventos que ocorrem localmente ou nas interfaces remotas, em outras palavras, eventos que ocorrem nas camadas abaixo da camada MIHF são reportados para as camadas acima da camada MIHF. Exemplos de eventos desta natureza são a degradação do nível de sinal do rádio e a indisponibilidade do enlace sem fio, dentre outros.

O MICS é responsável por disponibilizar comandos para as camadas acima da camada MIHF. Estes comandos são responsáveis por controlar as camadas inferiores, mais especificamente, por controlar os procedimentos referentes ao *handover*.

O serviço de informação (MIIS) provê informações sobre as características e serviços das redes que os dispositivos têm disponíveis para se conectar. Estas informações podem ser, por exemplo, coordenadas GPS, informação do canal, estado atual da rede, etc.

Também é previsto um servidor de informação (*Information Server - IS*) onde são concentradas todas as informações referentes às redes e que vão auxiliar na seleção da melhor rede e conseqüentemente na execução do *handover*. Os dispositivos móveis terão acesso às informações armazenadas no IS através do serviço de informação. Como o IEEE

Tabela 2.1: Tabela de decisão utilizada como entrada para o método SAW

	<i>banda</i>	<i>peso</i>	<i>custo</i>	<i>peso</i>
WiFi	54	0,5	R\$ 0,70	0,5
WiMAX	70	0,4	R\$ 1,00	0,6
Celular	42	0,1	R\$ 5,00	0,9

802.21 não define onde o IS fica localizado na arquitetura, ele pode estar localizado, por exemplo, fora do domínio do dispositivo móvel. Também não é definida a estrutura de armazenamento dos dados referentes às redes disponíveis para os dispositivos móveis.

2.7 SAW - Simple Additive Weight

Os tomadores de decisões frequentemente lidam com problemas que envolvem múltiplos critérios. Geralmente estes critérios são conflitantes. Um exemplo é a compra de um carro [84]. Existem diferentes opções de cores, motores, número de ocupantes, etc. Como escolher um modelo que se adeque às necessidades do comprador? Este tipo de problema é similar ao problema estudado, onde o usuário tem várias opções de redes para se conectar e precisa decidir qual a melhor alternativa baseada em suas necessidades.

O método SAW é o método mais utilizado dos métodos de decisão multicritério [84]. Além do método SAW, existem outros métodos para resolver problemas de decisão multicritério como, por exemplo, AHP (*Analytical Hierarchy Process*) [74], MAHP (*Multiplicative Analytical Hierarchy Process*) [14] e TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) [44]. O método SAW foi escolhido para a construção do algoritmo devido à sua simplicidade em comparação aos outros métodos [84].

Para chegar a um valor, ou seja ao resultado gerado pelo método, uma tabela de decisão é montada baseada em cada alternativa e cada critério disponível. A Tabela 2.1 mostra os dados de uma tabela de decisão. Cada alternativa é uma linha e cada critério é uma coluna com seus respectivos pesos. Neste exemplo cada rede é uma alternativa com suas respectivas características e pesos associados. Todos os elementos da tabela de decisão precisam ser normalizados e então o método SAW pode ser usado para qualquer tipo e qualquer número de critérios.

Um peso que determina a importância de cada critério na composição da nota final precisa ser associado a cada critério. E a soma de todos os pesos precisa ser igual a 1 [71]. A equação resultante do método SAW terá a seguinte forma:

$$Score_i = \sum_{j=1}^M W_j(m_{ij})normal \quad (2.1)$$

onde $Score_i$ é a nota da alternativa i calculada pelo método SAW, W_j é o peso do critério j , M é o número de critérios disponíveis e m_{ij} é o valor do critério da alternativa i para o critério j .

2.8 CogProt - Arcabouço Cognitivo

Um arcabouço cognitivo para autoconfiguração de protocolos de comunicação é apresentado em [52]. Esse arcabouço, denominado CogProt, foi baseado no conceito de redes autônomicas e é uma solução descentralizada para reconfiguração dinâmica dos protocolos. Ele adiciona na arquitetura do elemento de rede um plano cognitivo *cross-layer*.

Os elementos que implementam o CogProt são capazes de monitorar o desempenho e reconfigurar parâmetros dos protocolos para evitar degradação do desempenho devido a mudanças nas condições da rede. Esse processo de adaptação é realizado através da reconfiguração dinâmica dos parâmetros de interesse. Os protocolos são reconfigurados periodicamente a partir de decisões de gerência tomadas de acordo com a experiência obtida pelo monitoramento do desempenho.

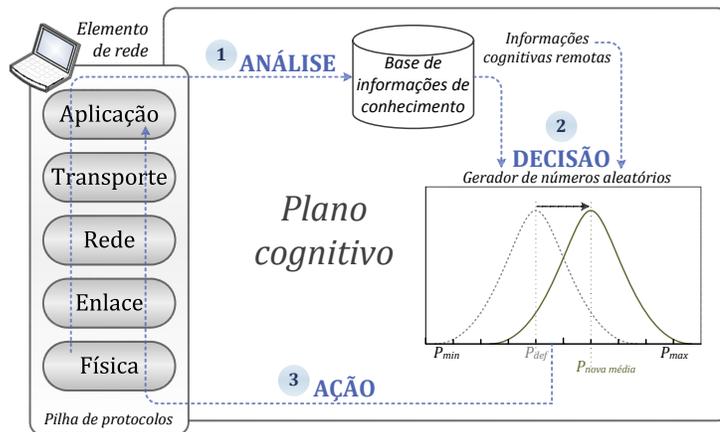


Figura 2.3: Visão geral do CogProt[57]

Para um parâmetro de interesse P , o CogProt introduz um laço de controle autônomico responsável por monitorar o desempenho e executar ações de reconfiguração do parâmetro. A Figura 2.3 exibe o funcionamento do CogProt. O laço de controle é dividido em três etapas: análise (1), decisão (2) e ação (3). Ao final de intervalos regulares, o mecanismo captura e armazena o desempenho obtido com o uso do valor de P , de acordo com métricas

de qualidade previamente definidas. A etapa de análise permite que o algoritmo construa uma base de conhecimento com informações de desempenho para os diferentes valores de P . Na etapa de decisão, o algoritmo procura na base de conhecimento pelo valor de P que apresente o melhor desempenho baseado na métrica de qualidade. Este valor é então usado como média para um gerador de números aleatórios que segue uma distribuição normal. Finalmente durante a etapa de ação, um novo número aleatório contido no intervalo operacional $[P_{min}, P_{max}]$ é obtido a partir desse gerador. Este novo número é atribuído ao parâmetro P e utilizado até a próxima execução do ciclo.

Este processo de otimização ajusta continuamente a média da distribuição normal para o valor de P que apresentar o melhor desempenho. A escolha dos valores em torno da média provê ao mecanismo a capacidade de adaptação. Se as mudanças na rede resultam em um novo valor ótimo para P , esse valor será eventualmente selecionado e seu desempenho será introduzido como a nova média para a etapa de decisão.

Capítulo 3

Trabalhos Relacionados

Este capítulo contém uma revisão bibliográfica dos assuntos abordados nesta tese. Na Seção 3.1 são apresentados os trabalhos relacionados com as redes cognitivas. Já na Seção 3.2 são apresentados os trabalhos que abordam gerência de mobilidade, incluindo os algoritmos de decisão de *handover*. E por fim, as Tabelas 3.1 e 3.2 apresentam um resumo das principais características dos trabalhos pesquisados sobre gerência de mobilidade e algoritmos de decisão de *handover*, respectivamente.

3.1 Redes cognitivas

O novo paradigma proposto por [80] para redes cognitivas é motivado principalmente pela necessidade de gerenciar de forma mais eficiente as redes de comunicação atuais devido ao aumento significativo e crescente de sua complexidade. A ideia principal desse novo paradigma é o processo cognitivo que pode perceber o estado da rede, planejar o futuro, tomar decisões e agir [34]. A ideia apresentada nos arcabouços CogMAC e CogHand, apresentados neste trabalho, é usar o conceito proposto por [80] e atuar nos parâmetros dos protocolos da subcamada MAC e de gerência de *handover*, observando, aprendendo e agindo de acordo com as mudanças da rede em tempo real. Nesta seção são apresentadas as principais soluções cognitivas para redes de comunicação encontradas na literatura.

A plataforma m@Angel [27] apresenta uma solução para resolver problemas de mobilidade em redes heterogêneas com princípios cognitivos. O processo cognitivo é implementado na rede, entre as estações bases e os MNs. A estrutura da rede de acesso é composta por dois planos: o plano de infraestrutura que inclui os elementos reconfiguráveis e o plano de gerência composto por entidades m@Angels. Cada entidade m@Angel é responsável pelo monitoramento da rede: uso dos recursos, metas e objetivos de gerência; e por funcionalidade de controle dos elementos reconfiguráveis.

Diferentemente do arcabouço CogMAC, apresentado neste trabalho, a plataforma

m@Angel exige um certo grau de cooperação entre seus elementos. No entanto, esta cooperação é geralmente executada dentro do escopo dos elementos da rede localizados em células vizinhas.

Uma plataforma reconfigurável para redes cognitivas é proposta por [77]. Esta plataforma apresenta uma arquitetura projetada para MNs dentro de uma rede cognitiva. Cada MN é dotado com uma habilidade de aprendizado. A solução é baseada no ciclo cognitivo proposto inicialmente por [61]. Os autores propõem a divisão deste ciclo em duas partes. A parte cognitiva e a parte do MN reconfigurável que possui a habilidade de observar e agir.

O raciocínio é baseado em observações feitas nas diferentes camadas do TCP/IP. Estas observações feitas pelos MNs são transmitidas para uma entidade central que executa um algoritmo de decisão e retorna uma mensagem contendo uma ação que deve ser executada pelo MN. A solução depende de uma entidade centralizada que é responsável pela decisão dos MNs, este modelo centralizado tem um ponto crítico de falha. Caso a entidade central falhe, ela compromete todo o sistema fazendo com que todos os MNs deixem de funcionar corretamente. Outro ponto negativo desta solução é a quantidade excessiva de sinalização entre os MNs e a entidade central que é responsável pelas decisões.

A ADMA [11] é uma arquitetura de gerência distribuída proposta para redes MANETs com princípios autônômicos. Ela permite que os MNs se auto-configurem em resposta às mudanças do ambiente. Além disso, a ADMA não necessita de uma entidade centralizada para executar funções de gerência, em vez disto, os MNs estabelecem conexões *peer-to-peer* entre si para trocar informações de gerência. O papel do administrador de rede nesta arquitetura é limitado a definir as políticas e introduzi-las em apenas um MN. A partir deste ponto os MNs precisam se conectar entre eles através de uma conexão *peer-to-peer* para disseminar as políticas de gerência. Uma vez que as políticas são repassadas, os MNs são capazes de tomar decisões baseadas em informações coletadas da rede.

A ADMA faz uma adaptação dos componentes propostos na RFC 3084 [20] e os associa com os elementos do ciclo cognitivo proposto por [61]. A principal desvantagem da arquitetura é o *overhead* introduzido na rede devido à troca de informações entre os MNs para disseminar as políticas de gerência. Além disso, para implementar os componentes de gerência propostos no MN é preciso um *hardware* robusto, alto processamento, levando a um consumo excessivo da bateria do MN.

O CogNet (*Cognitive Complete Knowledge Network*) [58] é um projeto de pesquisa que propõe uma arquitetura para redes cognitivas. A arquitetura faz uso de uma grande quantidade de informações obtidas por entidades da rede, como os MNs, os pontos de acesso e as estações base, com o objetivo de aumentar o desempenho geral da rede. A principal ideia por traz desta arquitetura é manter uma abstração de cada camada da pilha de protocolo TCP/IP. Cada camada do TCP/IP é estendida com um módulo de cognição,

também chamado módulo de cognição intracamada. Estes módulos são responsáveis pelo controle, coordenação e monitoramento entre as camadas. Os módulos se comunicam através de um barramento cognitivo (*Cognitive cross layer bus*) que faz parte do plano cognitivo da solução. Este barramento cognitivo é responsável pela coordenação dos módulos de cognição e são implementados em paralelo com a pilha de protocolo TCP/IP.

Na arquitetura proposta, as funções cognitivas são implementadas nos elementos cognitivos que estão distribuídos entre os protocolos das camadas. Esta forma de implementação simplifica o processo cognitivo executado na rede, uma vez que reduz o *overhead* causado pela sinalização excessiva. No entanto, o desempenho da arquitetura proposta parece ser altamente dependente da operação do plano cognitivo que é responsável pela tradução dos objetivos fim a fim para as configurações dos parâmetros em cada camada. Conseqüentemente, a falta de coordenação pode levar a um resultado de desempenho imprevisível.

Um esquema cognitivo para redes de acesso é proposto por Baldo e Zorzi [12] (CNA). O objetivo principal deste esquema é auxiliar o MN a escolher uma rede de acesso baseado nas preferências dos usuários provendo qualidade de serviço para as aplicações. Porções de *software* localizados nos MNs monitoram as conexões ativas e obtêm dados que serão encaminhados para um módulo responsável pela decisão. Este módulo de decisão usa técnicas de lógica *fuzzy* para decidir quais redes de acesso são mais adequadas para o usuário se conectar. O módulo de decisão é na realidade um arcabouço para representação do conhecimento e decisão baseado em lógica *fuzzy*. Ele auxilia a implementação e o funcionamento do processo cognitivo. Todo esse esquema é construído levando em consideração informações de diversas camadas da pilha de protocolo TCP/IP (*cross-layer*) armazenadas numa base de dados em cada MN que representa o conhecimento das redes.

À medida que a base de dados é incrementada, os MNs compartilham estas informações entre si com o propósito de aumentar a confiabilidade na escolha das redes e conseqüentemente aumentando a satisfação dos usuários. Nesta solução, o *overhead* causado pela troca de mensagens entre os MNs pode comprometer o desempenho geral da rede. Além disso, a composição da decisão final pelo MN é formada por informações obtidas por outros MN. Um problema com a troca de mensagens entre os MNs ou mesmo um atraso devido a um congestionamento da rede pode comprometer a qualidade da decisão, não refletindo nas escolhas dos usuários.

Uma arquitetura de rede com pacotes cognitivos é proposta por [39]. A ideia do trabalho é mudar a forma tradicional como as redes TCP/IP fazem o roteamento e o controle de fluxos. Estas funções que são executadas nas camadas do TCP/IP, nos MNs da rede, são inseridas nos pacotes de dados. Estes pacotes especiais são chamados pelos autores de pacotes cognitivos e são responsáveis pelo próprio roteamento. Eles escolhem rotas baseadas no prévio conhecimento para evitar caminhos congestionados e assim evitar

de serem destruídos. Cada pacote cognitivo contém um mapa cognitivo e uma porção de código que é executado cada vez que um pacote chega num roteador. As decisões de roteamento são tomadas baseadas no mapa cognitivo contido no pacote, mensagens deixadas por outros pacotes ou também pelo MN da rede.

A ideia dos pacotes cognitivos é baseada no conceito das redes ativas [18]. Em ambas as soluções os roteadores presentes nas redes executam códigos que estão contidos nos pacotes. Uma característica ímpar da solução proposta por [39] é que os pacotes cognitivos têm a habilidade de mudar seu comportamento baseado no estado da rede. Eles coletam informações da rede e montam uma base de conhecimento que será usada para tomar futuras decisões. Uma desvantagem desta arquitetura que deve ser levada em consideração é o *overhead* introduzido nos roteadores ao executar porções de código que são carregados pelos pacotes cognitivos.

Um protocolo cognitivo para camada de transporte é proposto por [75] chamado TCPE (*TCP Everglades*). Além do protocolo de transporte, os autores propõem uma arquitetura de rede cognitiva genérica que possibilita a implementação de uma família de protocolos cognitivos para a camada de transporte. Muitas ideias propostas por [58] são adotadas na arquitetura cognitiva deste trabalho. Por exemplo, a arquitetura prevê um módulo cognitivo em cada camada da pilha de protocolo do TCP/IP num modelo *cross-layer*. Além do módulo cognitivo, a solução possui um módulo adicional chamado módulo de conhecimento (*knowledge module*) que é responsável por armazenar as informações necessárias para alimentar as funções cognitivas do módulo cognitivo posicionado em cada camada. As informações armazenadas no módulo de conhecimento podem ser obtidas de quatro fontes: da camada onde está o módulo do conhecimento, da camada superior, da camada inferior e do módulo cognitivo da mesma camada. O módulo cognitivo possibilita implementar funções cognitivas em todas as camadas. A solução prevê que estes módulos podem estar posicionados de forma centralizada ou distribuída ao longo da rede.

No protocolo TCPE, o TCP *westwood* foi estendido tendo sua função responsável por estimar a largura de banda do enlace substituída por uma função cognitiva que leva em consideração informações da camada de enlace. O desempenho deste protocolo pode sofrer degradação, pois não leva em consideração informações da camada corrente para a configuração de parâmetros das camadas inferiores visto que ele confia em serviços oferecidos por estas camadas.

O arcabouço de rede cognitiva proposto por [81] é composto por três camadas horizontais. A camada superior é responsável pela especificação e tradução dos requisitos requeridos pelas aplicações e usuários em ações que podem ser entendidas pelo processo cognitivo. Esta especificação é realizada utilizando uma linguagem chamada linguagem cognitiva de especificação CSL (*Cognitive Specification Language*). Vários processos cognitivos podem ser executados na camada intermediária podendo ser distribuídos entre

vários MNs da rede. Estes processos podem se comunicar com seus pares com o objetivo de compartilhar e obter informações. O processo cognitivo é responsável pelo aprendizado, conhecimento e decisão podendo ser executado quando as informações sobre a rede são limitadas, geralmente no início do processo. Todas as informações obtidas pela interação de todas as camadas do protocolo TCP/IP têm o objetivo de aumento de desempenho fim a fim.

A camada inferior do modelo corresponde ao SAN (*Software Adaptable Network*), consistindo de elementos de rede modificados que trabalham em conjunto com sensores. A comunicação entre os elementos de rede modificados e o plano cognitivo é feita utilizando a SAN API. Esta arquitetura traz modularidade e flexibilidade ao modelo dos elementos de rede modificados podendo ser adaptada a uma série de protocolos que precisam ser modificados com o objetivo de se tornarem protocolos cognitivos.

Um outro trabalho que trata de redes cognitivas é proposto por [67]. Neste trabalho os autores propõem uma arquitetura de rede cognitiva colaborativa (*Collaborative and Cognitive Network Platform - CCNP*) com o objetivo de resolver problemas de gerência de rede de forma colaborativa. A arquitetura é uma plataforma distribuída que emprega cognição nos seus nós para melhorar as habilidades de adaptação e prover cooperação entre eles. Os nós podem cooperar entre si para aumentar a precisão de suas ações.

A arquitetura prevê um plano cognitivo que interliga todas as camadas da pilha do protocolo TCP/IP formando também um modelo *cross-layer*. Este plano cognitivo fornece decisões inteligentes para controlar os recursos compartilhados, coletar e gerenciar as transições de estados durante todo o ciclo desses recursos. O objetivo principal do plano cognitivo é garantir que a tradução dos objetivos de alto nível fornecidos pelos usuários sejam de fato aplicados na rede. Como a proposta é projetada para ser aplicada em elementos de rede heterogêneos, estes elementos precisam ser representados como dispositivos virtuais e mapeados para recursos físicos com o objetivo de oferecer portabilidade para as aplicações e melhorar a utilização dos recursos sem afetar as aplicações que já foram desenvolvidas.

O trabalho propõe ideias para uma arquitetura de rede cognitiva colaborativa. Mas como não foi apresentado um exemplo de implementação ou simulação para fins de validação é muito difícil mensurar a eficiência da arquitetura, principalmente no que diz respeito à cooperação entre os elementos de rede com o objetivo de resolver problemas de gerência.

3.2 Gerência de mobilidade

Os trabalhos que tratam da gerência de mobilidade tentam suprir as limitações do TCP/IP em lidar com mobilidade, que basicamente é o termo usado quando o MN baseado em IP se

movimenta entre pontos de conexão em diferentes redes. Quando esta mobilidade ocorre surgem diversos problemas que não foram pensados no projeto original da arquitetura do TCP/IP, como por exemplo, o que fazer com as conexões ativas quando o MN troca seu IP original? Estas e outras questões surgem e precisam ser resolvidas para implementar mobilidade dos MNs nas redes TCP/IP.

Vários trabalhos propostos pela comunidade científica tratam da gerência de mobilidade nas redes de acesso sem fio heterogêneas. Muitos trabalhos são modificações do IP móvel [70] e são implementados na camada de rede. Alguns trabalhos também se baseiam no protocolo SIP (*Session Initiation Protocol*) [73] e são implementados na camada de aplicação. Além disso, foram encontrados trabalhos que foram implementados nas camadas de transporte, enlace como também propostas de mecanismos de gerência numa nova camada, por exemplo, uma nova camada entre as camadas de transporte e de rede ou uma nova camada entre as camadas de rede e de enlace.

Algumas arquiteturas implementam o IEEE 802.21 e estão aptas a realizarem tanto *handover* vertical quanto *handover* horizontal. Outras arquiteturas combinam estas características com um processo cognitivo que é capaz de analisar, aprender e atuar nas redes de forma autônoma. Nesta seção são apresentadas propostas de mecanismos, arquiteturas e algoritmos de decisão relacionados com o trabalho desta tese que propõem a mobilidade dos MN e a integração das redes sem fio com tecnologias diferentes.

Em [70] é proposto um conjunto de protocolos e mecanismos que proporcionam mobilidade na Internet, conhecido como IP móvel (MIPv4). O IP móvel permite roteamento de datagramas de forma transparente para MNs na Internet. Cada MN é associado com um IP permanente que é um identificador (*Home Address*) independentemente da rede que estiver conectado. Enquanto estiver situado fora de sua rede nativa (*Home Network*), um MN é também associado a um “endereço aos cuidados” (*Care-of-Address - CoA*) que provê informações sobre sua rede atual.

No IP móvel, quando um MN está em sua rede nativa, ele atua como um nó fixo tradicional. Cada vez que ele se move de sua rede nativa e acessa outra rede (*Foreign network - FA*) ele obtém um CoA e informa sua rede nativa sobre o novo endereço mandando uma mensagem *registration request*. A arquitetura prevê um agente que associa o CoA com o agente nativo (*Home Agent*). Ao receber datagramas destinados ao MN, vindos de um nó correspondente (*Correspondent Node - CN*), o agente nativo reenvia-os através de um túnel para o CoA do MN que está na FA. Quando o FA recebe os datagramas ela os entrega para o MN. Quando o MN deseja enviar datagramas de volta para o CN, eles são roteados diretamente pelo roteador da FA como rota padrão.

O protocolo HOPOVER [29] (*HandOff Protocol for OVERlay networks*), compatível com o IP móvel, foi projetado especialmente para redes sobrepostas (*overlay networks*). Pode realizar tanto o *handover* horizontal quanto o vertical. O atraso do *handover* é

um pouco inferior ao do IP móvel. Este protocolo tem a desvantagem da necessidade de utilização de estações base para manter um grande número de informações sobre os MNs. Estas informações incluem: uma lista de estações base para o encaminhamento dos pacotes, filas de *buffers* para os pacotes em trânsito, parâmetros de reserva de recursos, roteamento e informações específicas a respeito das características das redes.

Uma arquitetura para o redirecionamento das conexões ativas do TCP foi desenvolvido por [38] chamado TCP *Redirection* (TCP-R). O TCP-R redireciona as conexões ativas do TCP quando é detectada uma mudança de IP na camada de rede. A ideia do TCP-R é inspecionar periodicamente o par de endereços IP da conexão TCP ativa e quando um endereço IP associado a essa conexão é alterado, um mecanismo de redirecionamento avisa aos pares comunicantes sobre esta mudança.

Quando o MN inicia uma nova conexão, ele assegura que o nó correspondente é ou não compatível com o TCP-R. Quando o MN troca de rede e obtém um novo IP, ele envia uma mensagem RD REQ para o nó correspondente avisando-o da mudança. Como este mecanismo não foi pensado de uma forma *cross-layer*, ou seja, a camada de transporte não está ciente das camadas de rede e de enlace, o que implica que o TCP-R não sabe diferenciar os tipos de perdas. Por exemplo, uma perda de pacote detectada pelo TCP que foi ocasionada por um *handover* demorado pode impactar na diminuição de sua janela fazendo-o baixar o envio de pacotes erroneamente.

Uma arquitetura para comunicação móvel ubíqua (*Architecture for Ubiquitous Mobile Communications - AMC*) proposta por [10] integra diversas tecnologias de redes sem fio. A arquitetura AMC elimina a necessidade de se fazer diretamente um acordo de nível de serviço (SLA) entre provedores de serviço. Para isso é usado um agente de interconexão. Em vez de criar uma estrutura totalmente nova, a arquitetura estende a estrutura existente para integrar diversas tecnologias de rede sem fio. O AMC usa o IP como protocolo de interconexão. Usando o IP como protocolo, transparência na interconexão das redes é alcançada. O AMC também prevê um sistema de autenticação e um sistema de bilhetagem.

Uma outra alternativa é o *handover* baseado no protocolo SIP [73, 13]. É uma solução implementada na camada de aplicação para gerência de mobilidade em redes heterogêneas. O SIP é um protocolo baseado em mensagens, escalável, extensível e que oferece controle de chamada/sessão. Os principais componentes do SIP são o agente de usuário (*User agent*), servidor de proxy (*Proxy server*) e servidor de redirecionamento (*Redirect server*). Um usuário é identificado pelo endereço de email *usuario@dominiodousuario*, onde *usuario* é o nome do usuário e o *dominiodousuario* é o seu domínio. Pode-se usar também o endereço IP do domínio.

Quando um nó correspondente inicia uma sessão com o MN, ele envia uma mensagem *INVITE* para a rede nativa do MN. O servidor de redirecionamento SIP da rede nativa

tem a informação atualizada sobre a localização do MN naquele momento, então ele redireciona a mensagem para o MN. A partir deste momento o procedimento normal do protocolo SIP é seguido para os pares estabelecerem uma sessão.

A desvantagem de usar arquiteturas de gerência de mobilidade baseadas no SIP é a excessiva troca de mensagens entre seus pares e o alto atraso proporcionados por eles, o que é intolerável por algumas aplicações, principalmente as de voz sobre IP. Como o SIP é uma solução de camada de aplicação para mobilidade, o desempenho dos protocolos de transporte é a principal causa dos atrasos excessivos nas técnicas de *handover* [33].

Em [62] é proposto um protocolo de gerência de *handover* denominado CHMP, implementado nas camadas de rede e de enlace (*cross-layer*). O protocolo usa informações de velocidade do dispositivo móvel e a qualidade do sinal do rádio como critério de decisão de *handover*. Informações específicas da camada de enlace são passadas para a camada de rede, melhorando o desempenho no tempo de *handover*.

O M3S e o SASHA (*Smooth Adaptive Soft Handover Algorithm*) são soluções propostas por [23]. O M3S é uma arquitetura de gerência de mobilidade que faz um balanceamento de carga entre as redes móveis utilizando um modelo *cross-layer*. Os resultados apresentados pelos autores mostram que o M3S tem um desempenho um pouco melhor do que o IP Móvel [70]. Sua principal característica é usar múltiplas conexões simultâneas para maximizar o QoS e a qualidade percebida pelo usuário.

Uma arquitetura de gerenciamento de mobilidade utilizando o SIP e o IEEE 802.21 foi proposta por [21] denominada de MxN. Foram criados para esta arquitetura gerenciadores de conexão e de adaptador (interfaces de rede, ex: WiFi, WiMAX) que integram com os serviços do IEEE 802.21. Assim, o SIP, que não é apto a trocar de interfaces quando há uma mudança de rede pelo MN, usa o gerenciador de conexão da arquitetura para lidar com essa mudança. Gerenciadores de conexão de outras redes podem usar o MIH (*Media Independent Handover*) do IEEE 802.21 e o gerenciador de adaptador da arquitetura para poder manipular o gerenciador de conexão.

A solução apresentada pelos autores parece promissora, porém por empregar o protocolo SIP há um excesso de sinalização, principalmente quando há um grande número de MNs no sistema, comprometendo o desempenho geral da rede. Talvez com a evolução das tecnologias de rádio e com o aumento da largura de banda, esta solução se torne viável.

A Tabela 3.1 apresenta uma comparação resumida das principais propostas de arquitetura de gerência de mobilidade IP citadas previamente. A primeira coluna indica o nome da arquitetura, a próxima coluna, **Camada(s) do TCP/IP**, indica em qual camada a arquitetura de gerência fica localizada na pilha do TCP/IP. A terceira coluna indica se a arquitetura possui um processo cognitivo em sua composição. A próxima coluna indica o tipo de *handover* que a arquitetura suporta. A última coluna indica o uso do IEEE 802.21.

Tabela 3.1: Comparação das arquiteturas para gerência de mobilidade

Arquitetura	Camada(s) do TCP/IP	Processo cognitivo	Tipo de <i>handover</i>	IEEE 802.21
MIPv4 [70]	Rede	Não	Horizontal, vertical	Não
HOPOVER [29]	Rede	Não	Horizontal, vertical	Não
TCP-R [38]	Transporte	Não	Vertical	Não
Mobilidade (SIP) [13]	Transporte	Não	Horizontal, vertical	Não
AMC [10]	Rede	Não	Horizontal, vertical	Não
CHMP [62]	Rede, Enlace	Não	Horizontal, vertical	Não
CNA[12]	Transporte, Enlace	Sim	Horizontal, vertical	Não
MxN [21]	Aplicação	Não	Horizontal, vertical	Sim
Arquitetura proposta	Rede, enlace	Sim	Horizontal, vertical	Sim

Algoritmos de decisão de *handover*

Na época da criação das redes celulares, os *handovers* tradicionais eram apenas baseados na qualidade do sinal do rádio e na disponibilidade de canais, ou seja, no momento da troca de redes, o mecanismo de gerência de mobilidade do MN usava para compor o algoritmo de decisão apenas estas duas métricas. Como citado anteriormente na Seção 2.1, nas novas gerações de redes sem fio heterogêneas, novas métricas de *handovers* são necessárias para uma alta mobilidade do MN e conseqüentemente acompanhadas de novos algoritmos de decisão. Como resultado da pesquisa deste trabalho, foi criado um algoritmo de decisão de *handover* que usa as métricas de banda e custo para auxiliar o MN a se conectar quando há mais de uma rede disponível. Nesta seção os principais algoritmos de decisão de *handover* encontrados na literatura são apresentados e são expostos numa tabela resumo (Tabela 3.2) para fins de comparação com o algoritmo de *handover* proposto.

Um algoritmo de decisão para o auxílio do *handover* de uma maneira distribuída usando o método SAW é proposto por [79]. Os autores usam como métricas a largura

de banda, a probabilidade de perda e o custo (em dinheiro) das redes. No algoritmo proposto, para cada métrica é associado um peso pelo usuário que significa a importância da métrica na composição da nota da rede. A soma dos pesos tem que ser igual a 1. Para cada rede disponível o usuário envia suas preferências associadas com os pesos e cada rede calcula um valor de qualidade através do método SAW que é enviado para o dispositivo móvel. De posse de todos esses valores de qualidade o dispositivo móvel escolhe a rede com o maior valor para fazer o *handover*.

Esta proposta tem desvantagens que são relevantes para o funcionamento do mecanismo de *handover*, os autores não levam em consideração a fonte em que o MN vai obter as informações para compor as métricas que serão usadas para ranquear as redes pelo mecanismo de gerência. Dependendo da solução adotada, pode levar muito tempo, comprometendo assim o tempo de total de *handover*. Além disto, num conjunto de redes ao alcance do MN, apenas uma rede pode ficar extremamente carregada enquanto as outras subutilizadas. Por exemplo, considerando um cenário em que a maioria dos usuários querem muita largura de banda em suas conexões, a rede que oferecer maior largura de banda para o usuário recebe o maior número de solicitações de conexões pelos MNs, fazendo-a ficar extremamente congestionada.

Um algoritmo de decisão usando o IEEE 802.21 entre redes WiFi e WiMAX é proposto por [83]. O algoritmo proposto é baseado em provisionamento de qualidade de serviço (QoS). Os autores definem uma função de QoS que fornece uma nota para ranquear as redes baseada nos métodos AHP e SAW. O método AHP é utilizado para calcular o peso das métricas de tráfego enquanto que o método SAW é usado para calcular a função de QoS. Os MNs dos usuários são associados com quatro classes de QoS. Cada classe de serviço tem diferentes pesos para as seguintes métricas: custo, largura de banda, atraso, taxa de erro dos pacotes e *jitter*. Estes pesos diferentes têm influência na função de QoS que é calculada pelo método SAW.

O algoritmo sobrecarrega os MNs, todos os cálculos necessários para a decisão do *handover* executados pelos métodos AHP e SAW são realizados pelos processadores dos MNs fazendo-os diminuir a autonomia de suas baterias, o que é vital para seu funcionamento. Esta proposta diminui a importância das preferências dos usuários na escolha das redes pois faz um mapeamento entre os serviços oferecidos pelas redes com as classes de serviço de QoS de cada tecnologia sem fio, além disso, a decisão de *handover* é baseada em informação dinâmica de QoS o que pode comprometer o desempenho das aplicações, pois pode ocasionar trocas desnecessárias de redes pelos MNs, um efeito conhecido como efeito *ping pong* [72].

Um algoritmo de decisão baseado numa função utilidade para satisfazer a QoS das redes sem fio é proposto por [56]. Esta função utilidade considera como métricas de decisão, o *Signal to Interference plus Noise Ratio* (SINR), a largura de banda, a carga de

tráfego e a mobilidade do usuário. O objetivo é escolher a melhor rede para maximizar a vazão. Como no trabalho proposto por [83], este algoritmo sobrecarrega os MNs processando os cálculos de decisão, além disso é difícil mensurar a eficiência do *handover*, pois o algoritmo propõe usar a métrica de mobilidade do usuário em sua função utilidade porém não descreve o modelo de mobilidade usado nos experimentos.

Um algoritmo de decisão utilizando o IEEE 802.21 é proposto por [51]. O dispositivo móvel periodicamente faz uma medição do *Receive Signal Strength* (RSS) que representa a quantidade de potência presente em um sinal de rádio. Para realizar o *handover*, o dispositivo móvel faz uma média dos RSS obtidos das redes para saber qual o melhor momento de realizar o *handover*.

Basicamente a solução tem o mesmo princípio do *handover* tradicional que faz a decisão baseada apenas na potência do sinal, não levando em consideração as preferências dos usuários. Os experimentos obtidos com esse trabalho demonstra uma pequena melhora no tempo do *handover* se comparado com o algoritmo tradicional, isso se deve ao uso do IEEE 802.21 em conjunto com o *Mobile IP* na realização do *handover*.

Em [42] são discutidos os diferentes fatores e métricas de qualidade que indicam se um *handover* é ou não necessário e propõem um algoritmo de decisão para *handover* vertical que habilita os dispositivos móveis a associar pesos às diferentes métricas das redes como, por exemplo, custo. Basicamente o algoritmo calcula um valor de qualidade Q . A função é $Q = f(\text{peso1} * \text{custo}, \text{peso2} * \text{larguradebanda}, \dots)$. Assim os dispositivos móveis escolhem as redes que possuem o maior valor Q .

O algoritmo proposto é muito genérico, pode ser adaptado para diversas métricas que podem ser escolhidas pelos usuários em seus MNs. Como não foi apresentada uma implementação ou simulação, é difícil avaliar a eficiência do algoritmo nas redes sem fio heterogêneas. Devido à natureza da solução proposta parece que o algoritmo sofre de excesso de sinalização para alimentar sua função de qualidade que decide a melhor rede para o MN se conectar. Além disso os autores não explicam como obtêm as métricas no cenário proposto, uma vez que a agilidade em obter essas informações impacta diretamente no tempo total de *handover*.

Todos os algoritmos de decisão citados anteriormente, exceto o algoritmo proposto por [79], executam todos os cálculos de decisão de *handover* no próprio MN. Como os MNs são limitados e dependem de bateria para funcionar, os cálculos realizados no MN requerem um processamento considerável que pode causar degradação de desempenho além de aumentar excessivamente o seu consumo de bateria. Outra característica importante que não foi observada nos algoritmos pesquisados foi o uso de prioridade para privilegiar os MNs que já estão conectados na rede. Por exemplo, se um MN está realizando uma chamada de voz e precisa fazer um *handover* para outra rede, é preferível priorizar esta conexão e manter o serviço do que derrubar o usuário caso a rede destino esteja lotada.

Tabela 3.2: Comparação dos algoritmos de decisão de *handover*

Algoritmo	IEEE 802.21	Preferências do usuário	Balanceamento de carga	Prioridade
Tawill et all. [79]	Não	Sim	Não	Não
Yang et all. [83]	Sim	Em parte	Não	Não
Lee et all. [56]	Não	Sim	Sim	Não
Kim and Jang [51]	Sim	Não	Não	Não
Hass et all. [42]	Não	Sim	Não	Não
ADH [66]	Sim	Sim	Sim	Sim

A Tabela 3.2 apresenta uma comparação entre as propostas de algoritmos de decisão de *handover* citados previamente. Além dos algoritmos pesquisados na literatura a tabela lista o ADH [66] para fins de comparação. Como mencionado anteriormente, o ADH é o algoritmo de decisão de *handover* que foi resultado da pesquisa deste trabalho. A descrição detalhada do ADH pode ser encontrada na Seção 5.1.

A Tabela 3.2 contém cinco colunas. A primeira coluna indica o nome do algoritmo, a segunda coluna é o **IEEE 802.21**, que indica se o algoritmo utiliza o IEEE 802.21 para o auxílio do *handover*. A próxima coluna é a **Preferências do usuário** que indica se o algoritmo leva em consideração as preferências dos usuários na decisão do *handover*. O quarta coluna é o **Balanceamento de carga** que indica se o algoritmo leva em consideração o estado da rede antes de direcionar os MNS para elas. E por fim, a última coluna indica se o algoritmo usa algum tipo de prioridade de MNS no momento da decisão.

Capítulo 4

Arcabouço CogMAC

Um algoritmo cognitivo para a reconfiguração automática dos parâmetros da subcamada MAC do IEEE 802.11 é fundamental para melhorar o desempenho das redes. Num ambiente dinâmico, suscetível à interferência, um algoritmo que se adapte à variação do meio, reconfigurando seus parâmetros, é de vital importância. Nesse contexto, este capítulo apresenta o CogMAC, um arcabouço cognitivo para autoconfiguração dos parâmetros da subcamada MAC do IEEE 802.11. A Seção 4.1 introduz o arcabouço cognitivo CogMAC. A Seção 4.2 descreve a implementação do CogMAC no simulador ns-2 e a avaliação dos resultados. E por fim, na Seção 4.3, é apresentada a implementação do CogMAC no *driver* de dispositivo *ath5k* configurado no *kernel* do GNU/Linux juntamente com a avaliação dos resultados.

4.1 Descrição do CogMAC

O arcabouço CogMAC foi baseado no CogProt [52] apresentado anteriormente na Seção 2.8. Este arcabouço foi projetado para funcionar na subcamada MAC do IEEE 802.11. O CogMAC é um arcabouço descentralizado, ou seja, os MNs que o implementam não possuem uma visão total da rede. Cada MN individualmente é capaz de monitorar o desempenho e reconfigurar os parâmetros da subcamada MAC para evitar degradação do desempenho devido a mudanças nas condições da rede. Esse processo de adaptação é realizado através da reconfiguração dinâmica dos parâmetros *retry limit*, *contention window* e *RTS/CTS threshold*, com o objetivo de melhorar o desempenho de acordo com as condições da rede.

Como no CogProt (Veja Seção 2.8), o CogMAC implementa um laço de controle autônomo que é executado a cada intervalo de tempo. A cada iteração deste laço de controle, o CogMAC executa três fases para a realização das tarefas de monitoração e otimização de desempenho e reconfiguração de parâmetros. Estas fases estão listadas nos

Algoritmos 1, 2 e 3, respectivamente.

Algoritmo 1 Monitor de Desempenho

1. **Obtenha** o número de bytes recebidos no intervalo *SampleInterval*
 2. **Obtenha** o intervalo de tempo entre a última interrupção do timer e o *SampleInterval*
 3. **Calcule** a vazão média R_i como $nBytes/SampleInterval$
 4. **Calcule** a vazão ponderada $R_i = (R_{i-1}) * (p) + R_i * (1-p)$, onde p é o peso informado na variável *historySignificance*
 5. **Obtenha** o access delay médio D_i
 6. **Calcule** o access delay ponderado $D_i = (D_{i-1}) * (p) + D_i * (1-p)$, onde p é o peso informado na variável *historySignificance*
 7. **Armazene** os valores ponderados de R_i e D_i na matriz na posição ocupada pelo atual *retry limit*, *contention window* e *RTS/CTS*
-

A Figura 4.1 exibe o funcionamento do CogMAC. Cada número na figura corresponde a uma fase do CogMAC, representadas pelos Algoritmos 1, 2 e 3. Estas fases compõem o laço de controle autônomo que é executado num intervalo de tempo definido pela variável *SampleInterval*.

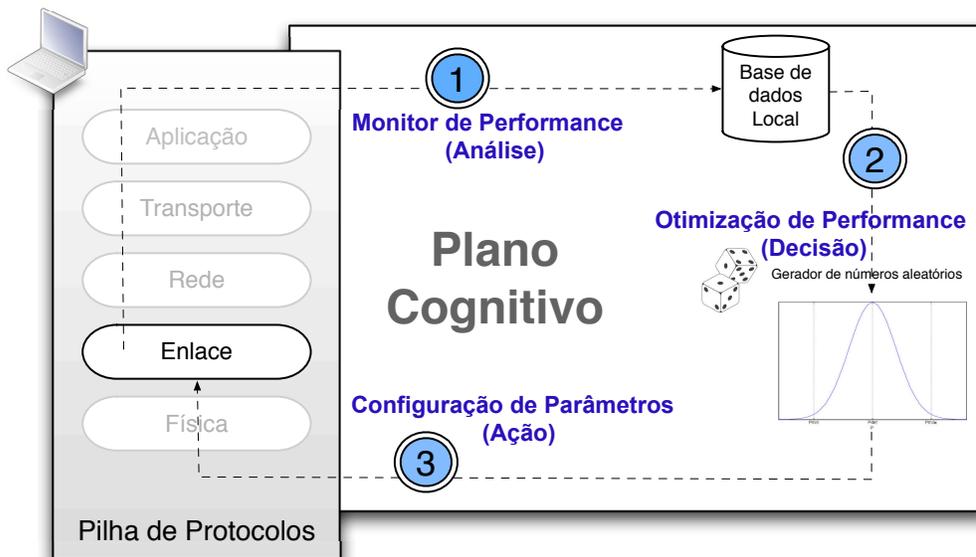


Figura 4.1: Visão geral do CogMAC

A primeira fase é a de obtenção dos dados. Nessa fase, o CogMAC analisa a rede para obter valores para a métrica de qualidade. Assim que a métrica de qualidade é composta, o CogMAC armazena-a numa base de dados local. O valor desta métrica está associada

com os valores dos parâmetros que estão configurados atualmente na interface de rede do dispositivo. A Tabela 4.1 exibe um exemplo destes dados. Considere a linha 1 da tabela, a métrica de qualidade é composta por uma vazão de 800 kbps e um *access delay* de 100 ms. Esta métrica de qualidade está associada aos valores 3, 8 e 0 que representam valores do *retry limit*, *contention window* e *RTS/CTS threshold*, respectivamente. O CogMAC faz uma fotografia da rede num intervalo de tempo e a associa com os valores dos parâmetros, configurados na interface de rede naquele intervalo.

Tabela 4.1: Armazenamento da métrica de qualidade na base de dados local associada com os parâmetros do IEEE 802.11

	<i>vazão</i>	<i>access delay</i>	<i>retry limit</i>	<i>cw</i>	<i>rts/cts</i>
1	800	100	3	8	0
2	2300	120	7	64	1500
3	500	30	1	16	300
4	700	50	2	16	300

A seguir o CogMAC executa a fase 2. Nesta fase ele analisa os números armazenados na base de dados e decide quais valores (ótimos) serão utilizados como média para compor a entrada do gerador de números aleatórios. Na terceira e última fase, o CogMAC obtém os valores através do gerador de números aleatórios e os atribuem aos parâmetros.

Como já mencionado anteriormente na fase 1, o algoritmo que a representa, monitor de desempenho (Algoritmo 1), é o responsável por obter a métrica de qualidade. Nesta implementação do CogMAC, a métrica de qualidade é a maior vazão da subcamada MAC para um certo *access delay* máximo (*AD Threshold*) que é definido pelo usuário. Para calcular a vazão da subcamada MAC (linha 3), o algoritmo acumula o número de *bytes* recebidos no intervalo *sampleInterval* (linha 1). O *access delay* médio (linha 5) é calculado como a diferença entre o momento em que um pacote entra na fila de transmissão da camada de rede para a camada de enlace e o momento em que ele é transmitido com sucesso, incluindo as retransmissões, pelo hardware.

Antes de armazenar a métrica de qualidade na base de dados (linha 7), o *Exponential Weighted Moving Average* (EWMA) é calculado:

$$P_i = P_c * w + (1 - w) * P_{i-1} \quad (4.1)$$

Onde P_c é o valor atual da métrica, P_{i-1} corresponde ao último valor do EWMA, e w é a distribuição do peso entre o valor atual e o último valor. Variando o valor de w , pode-se ajustar qual valor será mais significativo na composição do P_i . O valor de w é configurado através da variável *historySignificance*.

Algoritmo 2 Otimização de Desempenho:

1. **Configure** a vazão máxima $R_{\{max\}}$ para zero.
 2. **Para** cada elemento da matriz com $R_{\{i\}}$ e $D_{\{i\}}$ **faça**
 3. **If** $R_{\{iAtual\}} > R_{\{max\}}$ **Then**
 4. **If** $D_{\{iAtual\}} \leq AD\ Threshold$ **Then**
 5. $retryLimitOtimo = retryLimit$
 6. $contentionWindowOtimo = ContentionWindow$
 7. $rtsctsOtimo = rtscts$
 8. $R_{\{max\}} = R_{\{iAtual\}}$
 9. **EndIf**
 10. **EndIf**
 11. **EndPara**
-

Já o algoritmo de otimização de desempenho (Algoritmo 2) é responsável por analisar os valores obtidos na fase de monitoração. Os valores obtidos são analisados e uma configuração ótima para cada parâmetro é escolhida. O algoritmo faz uma consulta na base de dados local (linha 2) e recupera os valores dos parâmetros associados com a maior vazão (linha 3). Caso o *access delay* seja menor ou igual ao *AD Threshold* (linha 4), os valores recuperados são atribuídos aos parâmetros correspondentes como valores ótimos (linhas 5, 6 e 7).

A configuração ótima para cada parâmetro é representada por $P_{\{def\}}$. Cada parâmetro tem seu $P_{\{def\}}$ que será usado como média no gerador de números aleatórios. Como exemplo, considere a Tabela 4.1. Para um *AD Threshold* de 50 ms, a linha 4 apresenta os valores ótimos para os parâmetros *retry limit*, *contention window* e *RTS/CTS threshold*, 2, 16 e 300 respectivamente. O algoritmo seleciona na base de dados as linhas com os *access delays* menores que o *AD Threshold* de 50 ms. Dentre estas linhas, ele seleciona a linha com a maior vazão e define o $P_{\{def\}}$ de cada parâmetro.

Quando o algoritmo começa a executar, as medições ainda estão no início, então os valores de configuração do parâmetro escolhido podem não corresponder aos valores ótimos devido ao curto tempo de execução. No entanto, à medida que o tempo avança e mais medições ficam disponíveis, o sistema se estabiliza e os valores escolhidos para a configuração do parâmetro tendem a se tornar ótimos.

Por último, o algoritmo de configuração de parâmetro (Algoritmo 3) é responsável por selecionar novos valores obtidos a partir da distribuição normal (linhas 2, 4 e 6) e configurar estes valores nos parâmetros na interface de rede do MN.

Para cada parâmetro o CogMAC seleciona um valor mínimo ($P_{\{min\}}$), um valor máximo ($P_{\{max\}}$) e um valor padrão ($P_{\{def\}}$). Inicialmente, todos os valores de $P_{\{def\}}$ são baseados nos valores padrões do 802.11 que são configurados estaticamente pelo fabricante do equipamento. A partir de $P_{\{def\}}$, são atribuídos valores para

$P_{\{min\}}$ e $P_{\{max\}}$ que são baseados na faixa de valores de cada parâmetro recomendada pelo fabricante. Um valor para uma nova atribuição a um parâmetro é obtido por meio de um gerador de número aleatório no intervalo $[P_{\{min\}}, P_{\{max\}}]$, de acordo com uma distribuição normal usando como média o $P_{\{def\}}$, como ilustrado na Figura 4.2.

Algoritmo 3 Configuração de Parâmetros:

1. **Configure** a média da distribuição normal para *retryLimitOtimo*
 2. **Obtenha** novo *retryLimit* da distribuição
 3. **Configure** a média da distribuição normal para *contentionWindowOtimo*
 4. **Obtenha** novo *ContentionWindow* da distribuição
 5. **Configure** a média da distribuição normal para *rtsetsOtimo*
 6. **Obtenha** novo *rtsets* da distribuição
-

A média da distribuição $P_{\{def\}}$ é considerada um valor ótimo. Valores próximos de $P_{\{def\}}$ também são escolhidos de tempos em tempos, se um outro valor $P_{\{new\}}$ for escolhido e levar a um melhor desempenho, a média da distribuição normal é trocada de $P_{\{def\}}$ por $P_{\{new\}}$.

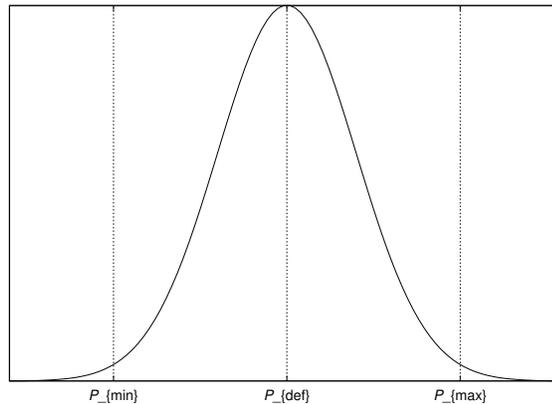


Figura 4.2: Distribuição normal utilizada pelo gerador de números aleatórios com média $P_{\{def\}}$ para a criação do novo parâmetro $P_{\{new\}}$.

4.2 Implementação no NS-2

O arcabouço CogMAC foi implementado no simulador de rede ns-2. Todo o arcabouço foi codificado na classe Mac802_11 representada pelos arquivos (mac-802_11.h, mac-802.11.cc) do simulador. Variáveis e métodos foram adicionados para acomodar as funcionalidades do monitor de desempenho, otimização de desempenho e configuração de parâmetros.

Uma matriz (Veja Tabela 4.1) foi alocada para armazenar numa base de dados local a vazão e o *access delay* médio associados com os parâmetros *retry limit*, *contention window* e *RTS/CTS threshold*.

Para calcular o *access delay* médio a estrutura do cabeçalho do pacote do simulador foi estendida. Foi adicionado ao cabeçalho um campo nomeado de *timestamp* preenchido cada vez que o pacote é enviado para a camada de enlace para a transmissão. Tendo o *timestamp*, o *access delay* pode ser facilmente calculado assim que o pacote for transmitido, diminuindo o tempo atual pelo tempo armazenado no *timestamp*.

4.2.1 Experimentos realizados no NS-2

A Figura 4.3 exibe a topologia utilizada nos experimentos. Uma fonte de tráfego S gera um tráfego FTP para cada MN da rede sem fio. O MN S é interligado ao AP através de dois roteadores R1 e R2. O enlace entre o nó S e o roteador R1 tem uma velocidade de 100Mbps e um atraso de 1ms. O enlace entre o roteador R1 e o roteador R2 tem uma velocidade de 100Mbps e um atraso de 10ms. O enlace entre o roteador R2 e o AP tem velocidade de 10Mbps e um atraso de 340ms.

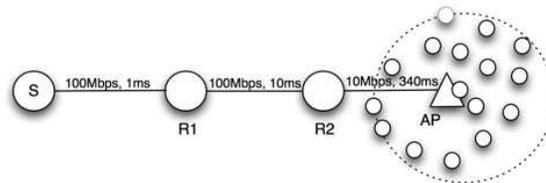


Figura 4.3: Topologia de rede infraestruturada com os MNs distribuídos obedecendo uma distribuição uniforme dentro do alcance do AP.

A rede sem fio é configurada seguindo a especificação do IEEE 802.11 com uma taxa de enlace de 11Mbps e taxa básica de 1Mbps. O modelo de propagação utilizado nas simulações foi o *two ray ground*. Foi introduzido nos experimentos um modelo de erro uniformemente distribuído entre os MNs, variando de 0,01 até 0,31 com incremento de 0,05 a cada 10 segundos.

O número de MNs variou de 1 a 50 com apenas um fluxo de tráfego por MN. Os MNs foram distribuídos obedecendo a uma distribuição uniforme dentro do alcance do AP. Tal configuração permitiu que todos os MNs se comunicassem com o AP mas os MNs localizados em lados opostos do AP se tornassem escondidos uns dos outros.

A implementação do algoritmo cognitivo variou dinamicamente o parâmetro de *retry limit* no intervalo entre 1 e 7. A janela de contenção variou entre 8 e 64 com incrementos de 8. Já o parâmetro *RTS/CTS threshold* variou no intervalo de 0 a 1500 com incrementos

de 300 *bytes*. Quando o *RTS/CTS threshold* é configurado para zero, ele é desligado, e quando é configurado para 1500 *bytes*, ele é habilitado para todos os pacotes que serão transmitidos.

4.2.2 Resultados obtidos a partir dos experimentos realizados no simulador

A Figura 4.4 apresenta o *MAC access delay* médio para diferentes valores (1 a 7) do parâmetro de *retry limit*. É possível observar que para valores de *retry limit* baixos, o *MAC access delay* permanece baixo. Nos cenários com maiores números de dispositivos móveis, o *MAC access delay* vai aumentando gradativamente. A curva CogMAC, que indica o algoritmo cognitivo, mantém o *MAC access delay* no limite configurado, *AD Threshold* de 0,004 segundos, valor aceitável para aplicações de Voz sobre IP e aplicações multimídia [41].

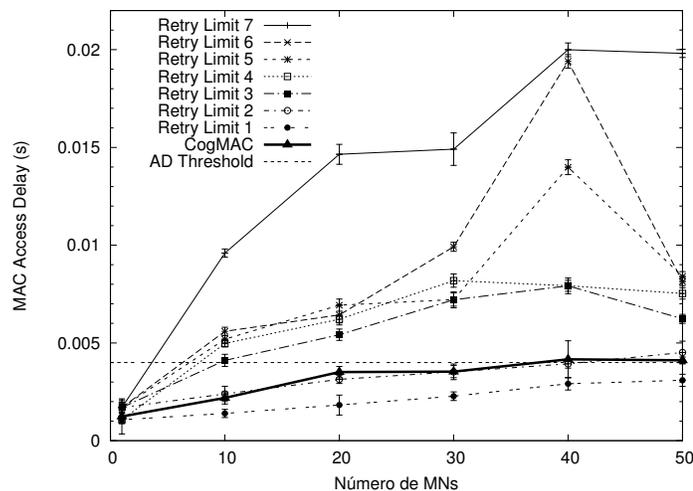


Figura 4.4: *MAC access delay* médio em função do número de MNs numa rede infraestruturada utilizando tráfego FTP para cada parâmetro de *retry limit*.

A Figura 4.5 exibe o *MAC access delay* ao longo do tempo. Os parâmetros *retry limit*, janela de contenção e *RTS/CTS Threshold* são configurados dinamicamente. O *access delay* permanece baixo quando o número de dispositivos móveis é baixo (25 segundos, 11 nós). À medida que o número de dispositivos aumenta, o *access delay* também aumenta e estabiliza (100 segundos, 50 dispositivos) mantendo o *access delay* no limite estabelecido (*AD Threshold* = 0,03s). Novamente, é possível observar a habilidade do CogMAC de se adaptar num ambiente altamente dinâmico como a rede sem fio satisfazendo os requisitos de atraso.

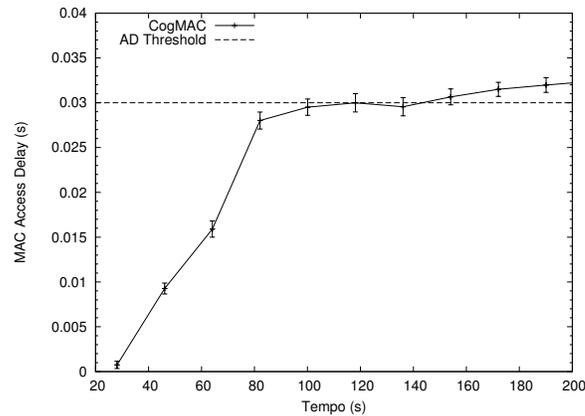


Figura 4.5: MAC *access delay* médio ao longo do tempo numa rede infraestruturada utilizando tráfego FTP.

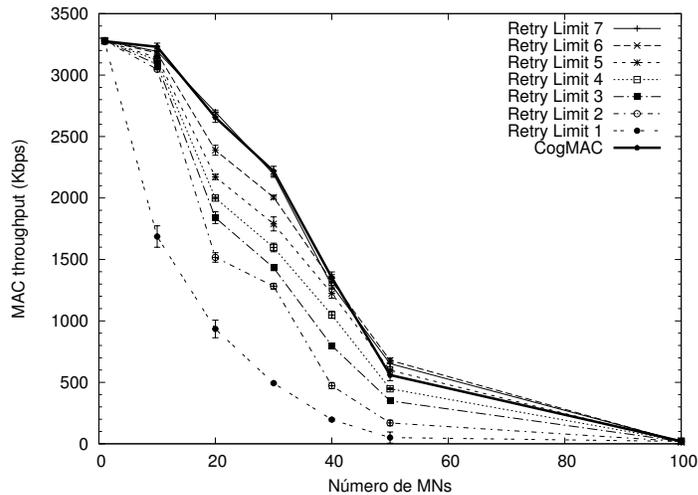


Figura 4.6: MAC *throughput* médio em função do número de MNs numa rede infraestruturada utilizando tráfego FTP para cada parâmetro de *retry limit*.

A Figura 4.6 exhibe o MAC *throughput* para diferentes valores (1 a 7) de *retry limit*. O CogMAC alcança bom desempenho para qualquer número de nós inseridos na rede garantindo o MAC *access delay* médio. Porém existem poucos casos onde alguns valores de *retry limit* leva a um MAC *throughput* alto. No entanto, o *AD Threshold* foi violado para estes casos como mostra a Figura 4.4.

A Figura 4.7 ilustra o MAC *throughput* médio ao longo do tempo com os parâmetros *retry limit*, janela de contenção e RTS/CTS *threshold* configurados dinamicamente. O MAC *throughput* diminui à medida que mais dispositivos entram na rede devido ao aumento do número colisões dos MNs. Apesar de alcançar o número de MNs máximo (50

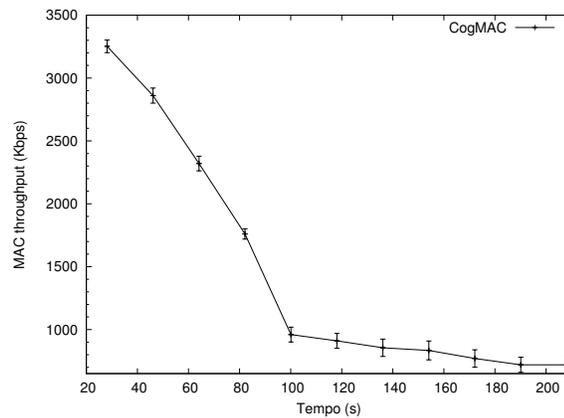


Figura 4.7: MAC *throughput* médio ao longo do tempo numa rede infra-estruturada utilizando tráfego FTP.

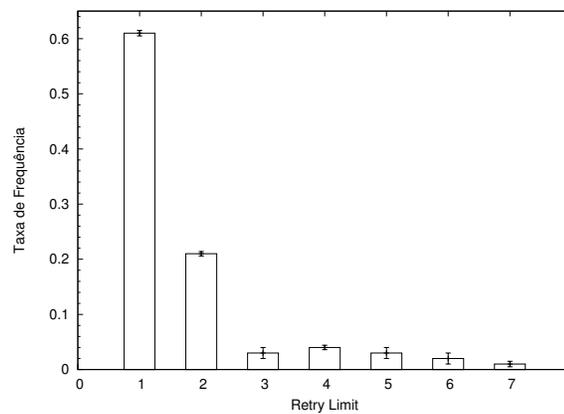


Figura 4.8: Frequência dos valores selecionados do parâmetro *retry limit* para um cenário com 50 MNs.

MNs, 1250 Kbps) depois de 91 segundos de simulação, o MAC *throughput* continua a diminuir devido ao aumento do tamanho do pacote.

A Figura 4.8 exhibe o número de vezes que cada valor de *retry limit* foi selecionado pelo CogMAC (valores normalizados) para um cenário com um grande número de dispositivos móveis. Como esperado, valores pequenos de *retry limit* implicam na configuração ótima como consequência do limite do *access delay* estabelecido pelo *AD Threshold*. Valores adjacentes dos valores ótimos são também selecionados ocasionalmente.

4.3 Implementação do CogMAC no *driver* de dispositivo real

Nesta seção são detalhadas as modificações realizadas no *kernel* [82] do GNU/Linux como também as alterações feitas no *driver ath5k* do dispositivo de rede sem fio com a finalidade de implementar o arcabouço CogMAC.

A maior parte da implementação do CogMAC foi realizada no *driver* de dispositivo, configurado como módulo do *kernel*. A Figura 4.9 apresenta uma visão geral da implementação e a localização dos componentes do CogMAC no GNU/Linux. Foi usado um *notebook* contendo uma placa PCMCIA da marca Proxim com *chip* Atheros.

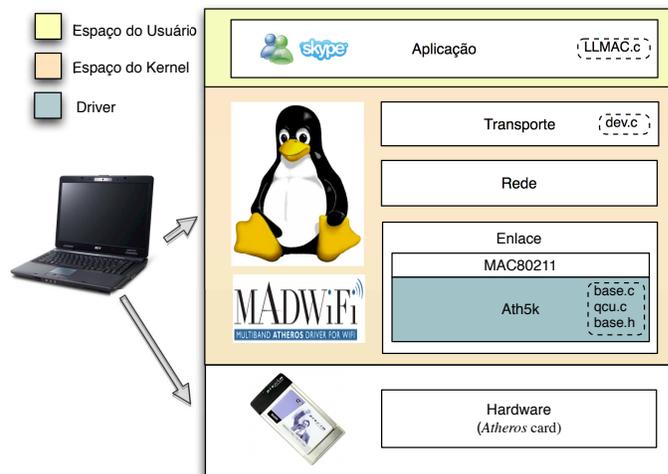


Figura 4.9: Visão conceitual do CogMAC no GNU/Linux *testbed*

O *driver ath5k*, que é ilustrado na Figura 4.9 pelo arquivo “*base.c*”, foi alterado com o objetivo de implementar o arcabouço CogMAC. Este *driver* é um dispositivo *soft-MAC* e é compatível com a maioria das placas com *chips* Atheros. O fato deste *driver* ser *soft-MAC* significa que todas as funções de gerência da subcamada MAC são implementadas em *software*. A API *MAC80211* [3] implementa estas funções de gerência que são usadas pelas placas de redes sem fio *soft-MAC*, como a placa de rede usada no *testbed* com *chip* Atheros.

A arquitetura do GNU/Linux separa a memória virtual em duas partes: espaço do usuário e espaço do *kernel*. O espaço do *kernel* é reservado para executar as extensões do *kernel*, os *drivers* de dispositivos dentre outros. Já o espaço do usuário é a parte da memória onde residem as bibliotecas do sistema, as aplicações de entrada e saída e as aplicações do usuário, dentre outras.

Estas duas partes de memórias são implementadas em espaços de endereços separados. Uma aplicação que reside em espaço de usuário não pode acessar diretamente os dados residentes no espaço do *kernel*. Para uma variável residente no espaço do *kernel* ser alterada, é necessário alterar o código fonte do *kernel*, compilá-lo e carregá-lo. Uma tarefa que dependendo das funcionalidades adicionadas ao *kernel* pode levar muito tempo. Para a movimentação de dados entre as duas partes da memória, sem ter que recompilar todo o *kernel*, o GNU/Linux oferece interfaces que podem ser usadas tanto no espaço do *kernel* quanto no espaço do usuário. Dentre várias interfaces disponibilizadas (*procfs*, *sysfs*, *configfs*, *debugfs* e *sysctl*) a interface *sysctl* foi escolhida pela facilidade de modificação.

O comando *sysctl*, quando utilizado no espaço do usuário, possibilita modificar e alterar variáveis do *kernel* em tempo de execução, ou seja, sem precisar compilar o módulo ou todo o *kernel* e carregá-los novamente. Todas as variáveis que o *kernel* disponibiliza, que podem ser modificadas com este comando, ficam localizadas no diretório */proc/sys*.

Cada arquivo ou pasta no diretório */proc/sys* é representada por uma entrada numa tabela mantida pelo *kernel* disposta numa hierarquia. Uma pasta é representada por uma entrada apontando para uma subtabela. Um arquivo é representado por uma entrada do tipo *struct ctl_table*. Esta entrada consiste em dados representados por este arquivo junto com algumas regras de acesso.

Foram adicionados uma pasta e dois arquivos no diretório */proc/sys* para configuração do CogMAC. Desta forma é possível configurar as variáveis CogMAC residentes no espaço do *kernel* a partir do espaço do usuário pela interface *sysctl*.

Algoritmo 4 Estrutura *ctl_table*

```

1.  static struct ctl_table cogmac_dir_table[] = {{
2.      .procname = "cogmac",
3.      .mode = 0555,
4.      .child = cogmac_table }};
5.  static struct ctl_table cogmac_table[] = {
6.      {.procname = "sample_interval",
7.       .data = &loop_interval,
8.       .maxlen = sizeof(int),
9.       .mode = 0666,
10.      .proc_handler = proc_dointvec},
11.     {.procname = "access_delay_threshold",
12.      .data = &access_delay_threshold,
13.      .maxlen = sizeof(int),
14.      .mode = 0666,
15.      .proc_handler = proc_dointvec}};

```

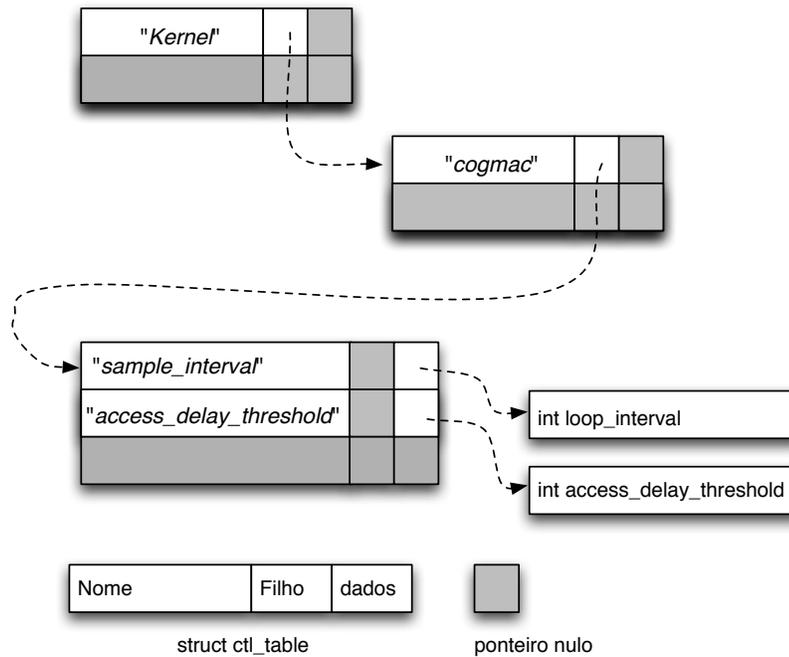


Figura 4.10: Registro de duas variáveis em `/proc/sys/cogmac`

O Algoritmo 4 exibe a estrutura `ctl_table`, que foi adicionada como entrada no `kernel`. A variável `procname` (linha 2) significa que a pasta `cogmac` será criada no diretório `/proc/sys`. A variável `mode` (linha 3) significa a permissão da pasta `/proc/sys/cogmac`, neste caso 0555. E por último a variável `child` (linha 4) indica o filho desta entrada, a subtabela `cogmac_table`. As linhas 6 e 11 indicam a criação das variáveis `sample_interval` e `access_delay_threshold` na pasta `/proc/sys/cogmac`. A variável `data` (linhas 7 e 12) representa as variáveis dentro do espaço do `kernel` que estão associadas às variáveis `sample_interval` e `access_delay_threshold`, respectivamente. Assim quando o usuário atribuir um valor à variável `sample_interval`, do espaço do usuário, estará na realidade atribuindo um valor à variável `loop_interval`. A variável `mode` (linhas 9 e 14) representa a permissão das variáveis `sample_interval` e `access_delay_threshold`, respectivamente. Por fim, a variável `maxlen` (linhas 8 e 13) indica o tamanho da variável, neste caso, um tamanho de um tipo inteiro. A Figura 4.10 exibe uma representação gráfica da estrutura `ctl_table`.

Neste exemplo as variáveis `sample_interval` e `access_delay_threshold` podem ser acessíveis no diretório `/proc/sys/cogmac` no espaço do usuário quando o módulo do `driver ath5k` estiver carregado.

Em (4.2) e (4.3) são exibidos exemplos de uso do comando `sysctl` para ler e escrever

na variável *sample_interval*, respectivamente.

$$\text{sysctl cogmac.sample_interval} \quad (4.2)$$

$$\text{sysctl cogmac.sample_interval} = 300 \quad (4.3)$$

A implementação do CogMAC foi dividida em duas partes devido a estas imposições da arquitetura do GNU/Linux. Uma parte do mecanismo que implementa as funcionalidades das fases de monitoração de desempenho, otimização de desempenho e configuração de parâmetros (Algoritmos 1, 2 e 3, respectivamente) foi codificada no arquivo “*base.c*”, como exibido na Figura 4.9, que reside no espaço de *kernel*. Outra parte do mecanismo que compreende as variáveis de configuração, que precisam ser modificadas constantemente, foi implementada como um programa no espaço de usuário no arquivo “*LLMAC.c*”, também mostrado na Figura 4.9.

Outras alterações foram feitas nos arquivos-fonte do *kernel* para adicionar algumas funcionalidades no CogMAC relacionadas com a métrica de qualidade. A Figura 4.11 faz um resumo de alto nível dessas funcionalidades do CogMAC exibindo onde foram implementadas, separadas por camadas da pilha de protocolo TCP/IP.

Na arquitetura do GNU/Linux, o movimento dos dados entre as camadas da pilha de protocolo TCP/IP ocorre através de uma estrutura chamada *socket buffer* (*sk_buff*). Um *sk_buff* contém os dados e os cabeçalhos dos pacotes que abrangem todas as camadas da pilha de protocolo. Todo pacote enviado e recebido pelas camadas do *kernel* é representado por um *sk_buff*. A estrutura *sk_buff* é definida no arquivo *skbuff.h* localizado no diretório */linux/include/linux*.

Nesta implementação do CogMAC, a estrutura *sk_buff* foi modificada para a criação da variável *instant_t1* que armazena o instante em que o pacote entra na fila, na camada de rede, para ser transferido para a camada de enlace. A variável *instant_t1* ajuda a compor o *access delay* que será um dos componentes da métrica de qualidade usada no CogMAC. A Figura 4.11 exibe a modificação da estrutura *sk_buff* (1) e a função em que o tempo é atribuído na variável *instant_t1* (2) na camada de rede. O *access delay* é o tempo em que o pacote permanece na subcamada MAC e é calculado pelo instante em que o pacote é transmitido com sucesso pelo *hardware* menos o *instant_t1*, como mostrado em (5) na Figura 4.11.

Ainda na Figura 4.11, pode-se observar a criação das métricas de qualidade, *MAC throughput*, a ser maximizada, e o *access delay*, a ser minimizada, limitada por um *access delay threshold* (*AD Threshold*) mostradas em (4) na Figura 4.11. O *MAC throughput* é calculado acumulando o total de *bytes* durante o intervalo *sample_interval* e dividindo-o por *sample_interval*.

Como discutido anteriormente, o CogMAC usa um gerador de números aleatórios para gerar novos valores para os parâmetros da subcamada MAC a cada intervalo de tempo

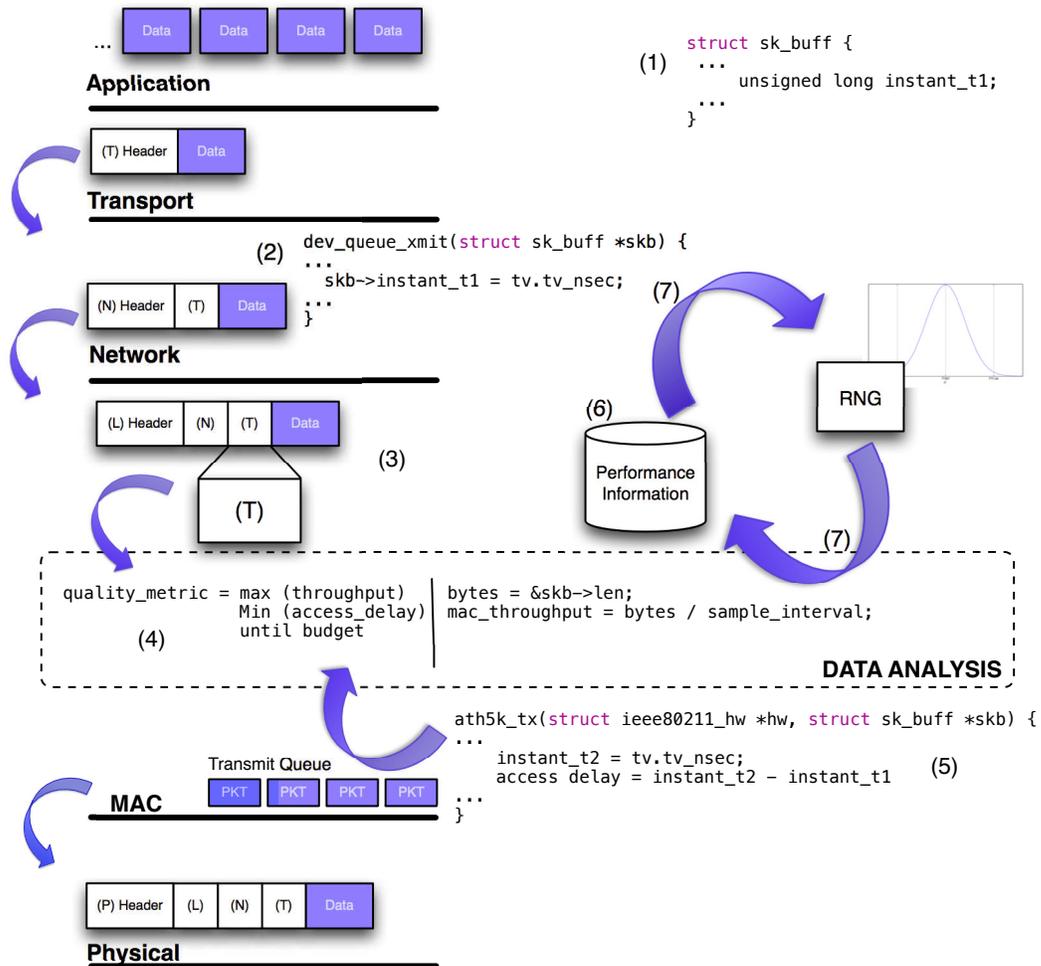


Figura 4.11: Principais modificações realizadas no *kernel* do GNU/Linux separadas por camadas TCP/IP para a implementação do arcabouço CogMAC.

que é definido pela variável *sample_interval*. No espaço do *kernel* não existe um gerador de números aleatórios com uma distribuição normal, também não existe a possibilidade de utilizar as bibliotecas padrões do C e nem de lidar diretamente com números reais. Devido às limitações do *kernel*, foi criada uma distribuição normal a partir de tipos inteiros dentro do código do *driver ath5k* com base na fórmula proposta em [50]. A seguir é exibida a fórmula usada:

$$N = \frac{(\sum_{i=1}^n u_i) - n/2}{(n/12)^{1/2}}, \quad (4.4)$$

onde n é a quantidade de números aleatórios uniformemente distribuídos, u_i é a soma de i números aleatórios uniformemente distribuídos fornecidos pela função da API do *kernel* “get_random_bytes”. Foi usado $n = 12$ como sugerido por [50].

4.3.1 Experimentos realizados no *testbed*

Para validar a implementação proposta, um *testbed* real foi montado. A Figura 4.12 ilustra a topologia utilizada nos experimentos. Dois *notebooks* e um *access point* foram utilizados para a montagem da rede infraestruturada IEEE 802.11g. O *notebook* A utiliza a placa PCMCIA com o *driver ath5k* modificado com o CogMAC habilitado. O *notebook* B utiliza a placa de rede sem fio padrão. Os experimentos foram realizados com a rede configurada no canal 6. No ambiente onde se encontra o *testbed* existem mais 5 redes sem fio, sendo que 4 delas causam interferência diretamente na rede do *testbed* por estarem configuradas nos canais 5, 6, 6 e 8 respectivamente. Topologia muito parecida com o cenário utilizado pela maioria dos edifícios e condomínios residenciais hoje em dia, devido à popularidade das redes 802.11. Um mecanismo cognitivo de reconfiguração dos parâmetros das redes sem fio, como o CogMAC, que detecte as deficiências da rede e aja em tempo real para corrigi-las é de grande importância para o desenvolvimento das redes.

Uma fonte de tráfego no *notebook* A gera um fluxo TCP para o *notebook* B. O tráfego é gerado com o programa *Iperf* [4] em modo cliente instalado no *notebook* A e em modo servidor instalado no *notebook* B. Cada experimento teve a duração de 5 minutos e foi repetido por 10 vezes, com as figuras mostrando um intervalo de confiança de 95% apresentadas na próxima subseção.

4.3.2 Resultados obtidos a partir dos experimentos realizados no *testbed*

A implementação do algoritmo cognitivo variou dinamicamente o parâmetro de *retry limit* no intervalo entre 1 e 10. A Figura 4.16 exibe o MAC *access delay* médio para os seguintes

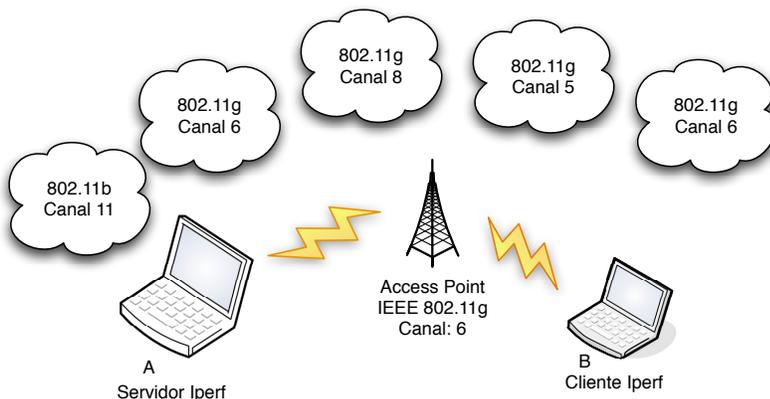


Figura 4.12: Topologia de rede do *testbed*

AD thresholds: 300, 400 e 1000 ms. O *retry limit* escolhido pelo algoritmo tem influência nestes valores como mostrado a seguir.

A Figura 4.13 exibe a quantidade de vezes que o parâmetro *retry limit* é escolhido pelo CogMAC quando o *AD threshold* é configurado em 300 ms. Note que o valor 2 é escolhido mais vezes, enquanto que outros valores também são testados. Isto acontece devido à limitação imposta pelo *AD threshold*. Com o *retry limit* igual a 2, o *access delay* não ultrapassa 300 ms, como ilustrado na Figura 4.16. Esta situação é requerida por algumas aplicações sensíveis ao atraso onde a vazão não é a métrica mais importante.

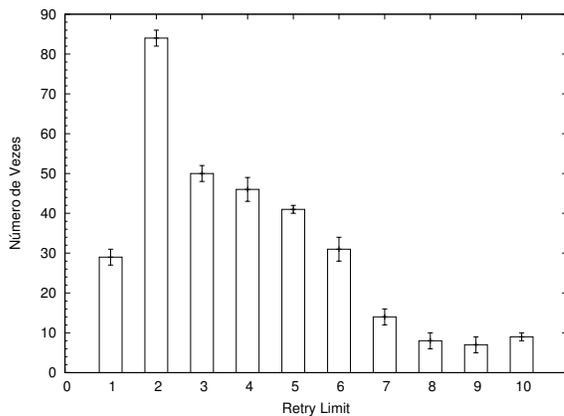


Figura 4.13: Frequência de *retry limits* utilizados com um limite de *access delay* em 300 ms.

A Figura 4.14 exibe a quantidade de vezes que o parâmetro *retry limit* é escolhido pelo CogMAC quando o *AD threshold* é configurado em 400 ms. O valor 3 é escolhido mais vezes, limitando o *access delay* em 400 ms como pode ser visto na Figura 4.16,

confirmando o bom funcionamento do algoritmo cognitivo.

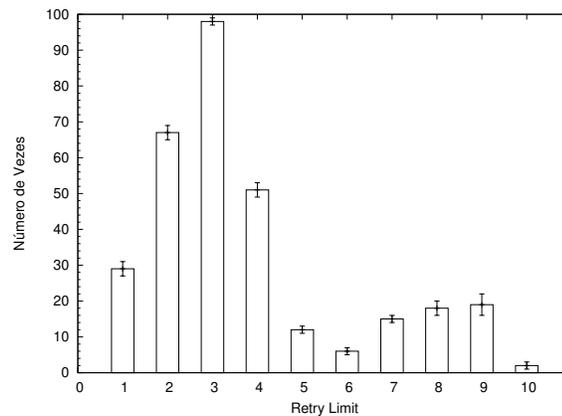


Figura 4.14: Frequência de *retry limits* utilizados com um limite de *access delay* em 400 ms.

Na Figura 4.15 o *access delay* é limitado em 1000 ms. Neste caso o valor 10 é escolhido mais vezes. Note que apesar do *AD threshold* ser configurado em 1000 ms, o *access delay* não ultrapassa os 700 ms como ilustrado na Figura 4.16.

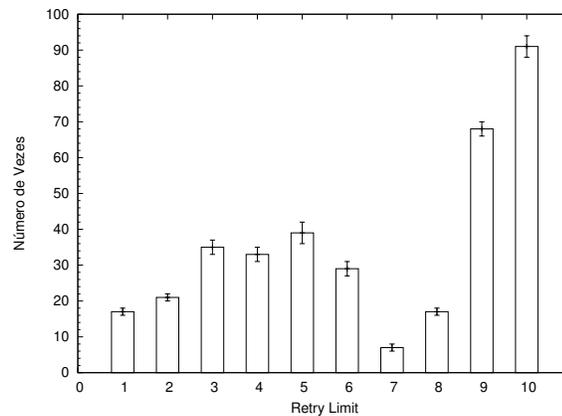


Figura 4.15: Frequência de *retry limits* utilizados com um limite de *access delay* em 1000 ms.

Podemos observar a vazão para diferentes *AD threshold* (300 ms, 400 ms e 1000 ms) na Figura 4.17. Note que quando o *AD threshold* é configurado em 300 ms, a vazão permanece num nível menor, entre 10 e 11 Mbps, se comparada com a vazão com os *AD thresholds* configurados em 400 e 1000 ms. Isto acontece, pois o *retry limit* escolhido dinamicamente pelo CogMAC permanece na maior parte do tempo em 2, como mostrado na Figura 4.13. Para um meio físico com interferências, como é o caso do meio testado nos

experimentos, a probabilidade de ocorrer erros nos pacotes é elevada. Com um número de *retry limit* baixo, a chance da perda de pacotes é maior que um *retry limit* alto, para um meio com interferências. Caso haja perdas de pacotes, o TCP diminui sua janela de transmissão, e conseqüentemente a taxa de transmissão.

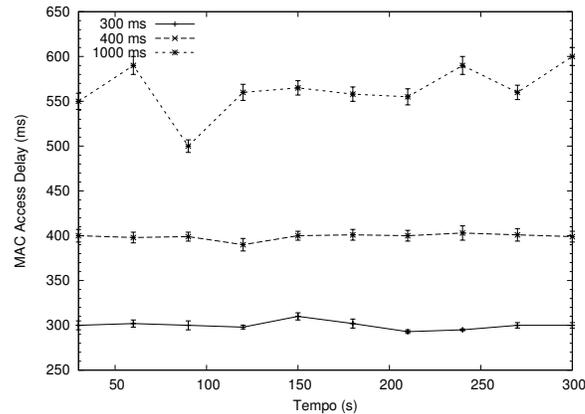


Figura 4.16: MAC access delay médio para diferentes *thresholds*

Quando o *AD threshold* é configurado em 400 ms a vazão aumenta um pouco, entre 11 e 12 Mbps como exibido na Figura 4.17. E por fim quando o *AD threshold* é configurado em 1000 ms a vazão aumenta um pouco mais e alcança em média 13 Mbps, mas o *access delay* médio observado ficou em torno de 600 ms como mostra a Figura 4.16.

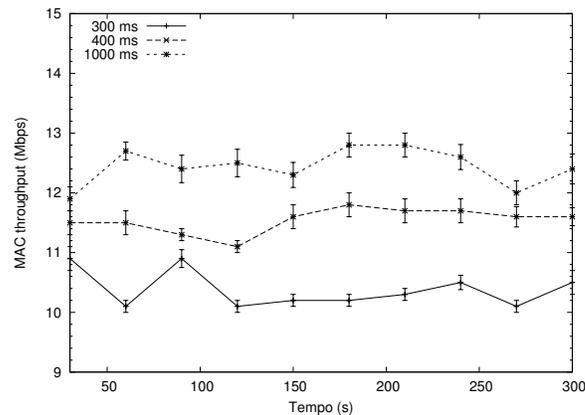


Figura 4.17: Vazão com diferentes *access delay thresholds*

A Figura 4.18 exibe uma comparação da vazão do CogMAC ativo versus CogMAC inativo usando a configuração original da placa PCMCIA, com o *retry limit* configurado estaticamente em 3. Pode-se observar um ganho de aproximadamente 25% quando o CogMAC está ativado. Uma outra característica observada do algoritmo cognitivo é a

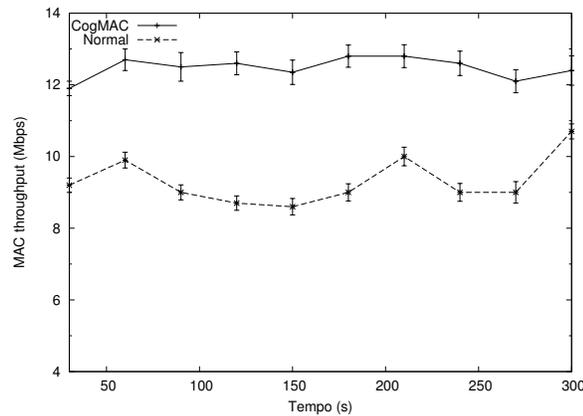


Figura 4.18: Vazão com o CogMAC ativo comparada com o CogMAC inativo

suavidade da curva da vazão. Com a retransmissão dos quadros na camada de enlace, o TCP não reduz sua janela de transmissão com frequência, fazendo com que o tráfego seja mais constante.

4.4 Comparação - experimento real com experimento no *Testbed*

Nem todos os cenários dos experimentos simulados foram reproduzidos nos experimentos reais, portanto somente alguns resultados foram comparados. O CogMAC consegue respeitar o *access delay* determinado (*AD threshold*) em ambos os casos como mostram as Figuras 4.16 e 4.4.

O valores configurados dinamicamente pelo CogMAC do *retry limit* refletem no MAC *throughput* em ambos os experimentos (simulação e *testbed*). Por exemplo, a Figura 4.16 exibe os valores dos *access delays* para diferentes *thresholds*, para esses resultados o CogMAC escolhe os valores de *retry limit* 2, 3 and 10, respectivamente (Figuras 4.13-4.15) na maioria das vezes. Para ambos os experimentos valores baixos de *retry limit* limitam o MAC *throughput* nos ambientes com interferência. Esse resultado indica que o CogMAC implementado no simulador apresenta comportamento similar ao do CogMAC implementado na placa de rede.

Capítulo 5

Arquitetura de Gerência de Mobilidade Cognitiva

Este capítulo apresenta a arquitetura de gerência de mobilidade cognitiva e seus componentes principais. Na Seção 5.1, é apresentado um algoritmo de decisão de *handover*. Este algoritmo usa as preferências do usuário num método de decisão multicritério para montar um *ranking* das redes disponíveis. A Seção 5.2 apresenta o Handsim, um simulador de eventos discretos, escrito em Java, criado com o objetivo de avaliar o algoritmo de decisão de *handover* proposto. Já na Seção 5.3, são descritos os experimentos realizados utilizando o algoritmo de decisão. A Seção 5.4 exibe os resultados obtidos com o Handsim utilizando o algoritmo de decisão. A parte cognitiva da arquitetura está detalhada na Seção 5.5. Ela faz uso do CogHand, um arcabouço cognitivo baseado no CogMAC que auxilia o MN a decidir em qual rede se conectar baseado em experiências prévias de desempenho armazenadas numa base de dados local. Na Seção 5.6 é detalhada a arquitetura de mobilidade cognitiva. Esta arquitetura utiliza o IEEE 802.21 para obter informações das redes disponíveis para os MNs, auxiliando-os a se manterem conectados. Foram realizados experimentos no ns-2, descritos na Seção 5.7. Os resultados obtidos dos experimentos são exibidos na Seção 5.8.

5.1 Algoritmo de Decisão

Como mencionado anteriormente (veja Capítulo 1), o IEEE 802.21 não padroniza o algoritmo de decisão de *handover*, ou seja, cada fabricante fica livre para desenvolver e implementar o seu próprio algoritmo. Esta seção apresenta um novo algoritmo baseado nas preferências dos usuários que é usado em conjunto com o IEEE 802.21 para a decisão de *handover*. A largura de banda requerida pelo usuário e o custo que ele está disposto a pagar pelo acesso à rede são levados em consideração como métricas de entrada para

o algoritmo. Estas métricas são utilizadas para criar uma pontuação (*score*) de cada rede ao alcance do dispositivo móvel utilizando o método de decisão multicritério MADM denominado SAW.

Com o objetivo de incluir as métricas no algoritmo, algumas modificações nas mensagens do IEEE 802.21 foram efetuadas. A mensagem *MIH_Get_Information_request* (MGIR) é utilizada pelo MN para consultar os dados das redes em seu alcance no servidor de informações (IS). Neste caso específico, as informações consultadas são largura de banda e custo. Foram incluídos nesta mensagem cinco campos. Os primeiros quatro campos que são preenchidos pelo usuário representam as métricas banda e custo com seus respectivos pesos. O quinto campo, denominado *type*, que é preenchido pelo mecanismo de gerência do MN, informa se o usuário está iniciando uma nova conexão ou está iniciando um *handover*.

Ao receber a mensagem “MGIR” vinda do MN, o IS consulta sua base de dados e envia uma mensagem *MIH_N2N_HO_Query_Resource_request* (MNHQR) para as redes candidatas aptas a aceitar a conexão solicitada pelo MN. Inicialmente as redes candidatas são aquelas que estão ao alcance do MN naquele instante. Uma alteração foi realizada na mensagem MNHQR para comportar o campo *flag*. Este campo é preenchido pelo IS, e informa para as redes candidatas se o MN tem mais de uma opção de rede ao seu alcance. Esta informação será usada pelas redes na decisão de aceitação do MN.

A mensagem *MIH_N2N_HO_Query_Resources_response* é enviada pelas redes candidatas para o IS como resposta ao pedido da mensagem MNHQR. Foi incluído nesta mensagem o campo *load*, que é preenchido pelas redes candidatas e indica a carga atual da rede.

Assim que o IS calcula as notas das redes candidatas, ele prepara uma mensagem *MIH_Get_Information_response* (MGIRResp) e a envia para o MN em resposta da mensagem MGIR. Foi incluída nesta mensagem uma estrutura de dados para armazenar as informações das notas das redes. Ao receber a mensagem MGIRResp, o MN seleciona a rede com a maior pontuação e lança um pedido de conexão diretamente para ela.

Protocolo de *Handover*

A Figura 5.1 exibe o diagrama de caso de uso do protocolo de *handover* do IEEE 802.21 adaptado para ser usado na proposta da arquitetura. Primeiramente, o usuário preenche seus requisitos de qualidade de serviço, neste caso, banda e custo com seus respectivos pesos (1) que vão ser as métricas utilizadas pelo algoritmo. De posse destas informações o usuário MIH envia uma mensagem para o MIHF (2) que por sua vez reenvia

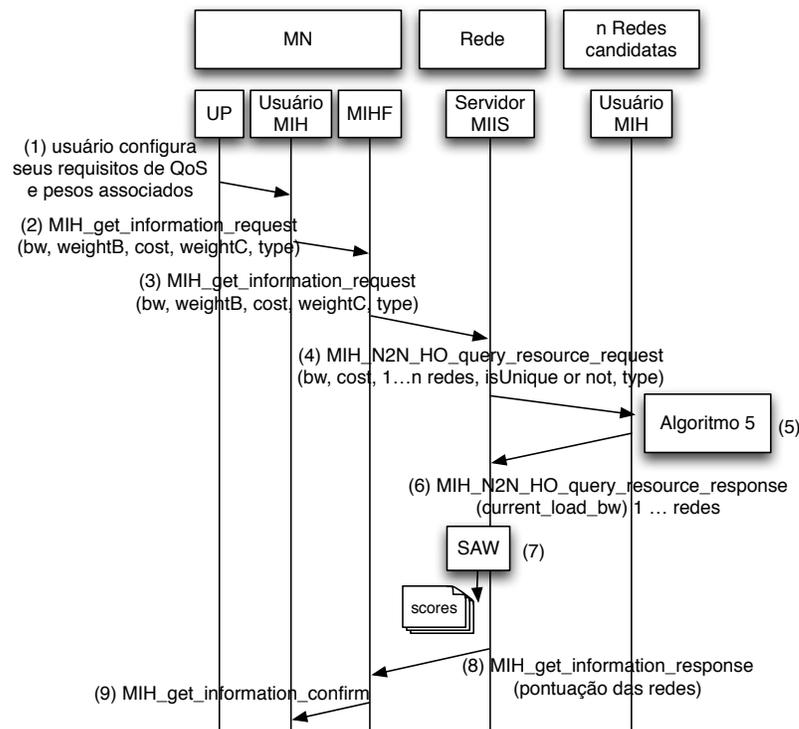


Figura 5.1: Diagrama de sequência do IEEE 802.21

esta mensagem para o IS (3). Baseado na localização do MN, o IS carrega uma lista com todas as redes que estão ao alcance do MN que solicitou a conexão, preenche o campo *flag* informando se o MN tem mais de uma opção de rede disponível e envia a mensagem *MIH_N2N_HO_Query_Resources* para todas estas redes (4). Cada rede ao receber a mensagem *MIH_N2N_HO_Query_Resources* executa o Algoritmo 5 (5) que decidirá se aceita ou não a conexão do MN. Toda rede que aceitar o pedido de conexão do MN prepara uma mensagem *MIH_N2N_HO_Query_Resources response* preenchendo o campo *load* com sua carga atual e envia esta mensagem para o IS (6). Assim que a mensagem é enviada, a rede reserva o recurso solicitado (banda) e dispara um *timeout*. Caso o *timeout* expire sem ter recebido a conexão do MN, a rede libera o recurso reservado. Ao receber as respostas de todas as redes que concordaram em aceitar a conexão do MN, o IS monta uma tabela de decisão como ilustrada na Tabela 2.1 e calcula a nota de cada rede, através do método SAW. Para compor a nota final, o IS divide a pontuação gerada pelo método SAW pela carga de cada rede e então responde com a mensagem *MIH_Get_Information_response* para o MIHF contendo as notas finais (8). Quando o MN recebe a resposta do IS (9), ele escolhe a rede com a maior pontuação e faz um pedido de conexão diretamente para ela.

A seguir, o Algoritmo 5, citado no passo (5) da Figura 5.1, é detalhado. Este algoritmo

de decisão de *handover* auxilia a rede a aceitar ou rejeitar um pedido de conexão do MN.

Na linha 1 é testado se o custo oferecido pela rede (*networkCost*) é menor ou igual ao custo (*cost*) solicitado e a banda oferecida pela rede (*networkBandwidth*) é maior ou igual a banda solicitada. Caso as preferências dos usuários sejam satisfeitas, o Algoritmo 5 continua, caso contrário ele rejeita o pedido de conexão (linha 20).

O algoritmo faz uso de dois *thresholds* distintos (*threshold1* e *threshold2*) que influenciarão na aceitação ou no bloqueio dos MNs. O bloqueio considerado neste caso é efetuado pela rede e não pelo sistema formado pelas redes. Por exemplo, uma rede do tipo WiFi pode bloquear um pedido de conexão de um MN mas ele pode ser aceito por outra rede, por exemplo, do tipo WiMAX. A utilização de dois *thresholds* no Algoritmo 5 possibilita a operadora escolher uma prioridade para os MNs ao serem aceitos na rede, uma vez que o algoritmo consegue diferenciar se o MN está iniciando uma nova conexão ou está tentando iniciar um *handover*.

Caso o MN já esteja conectado numa rede, é preferível manter uma continuidade de conexão ao longo das redes que ele se movimenta do que interromper seu serviço. Configurando os *thresholds*, o algoritmo reserva um recurso extra para os MNs que querem fazer um *handover*. A Figura 5.2 ilustra os dois *thresholds*. O *threshold1* é utilizado por todos os MNs que solicitam uma conexão, a partir do *threshold1* até o *threshold2* apenas os MNs que estão realizando um *handover* podem conectar.

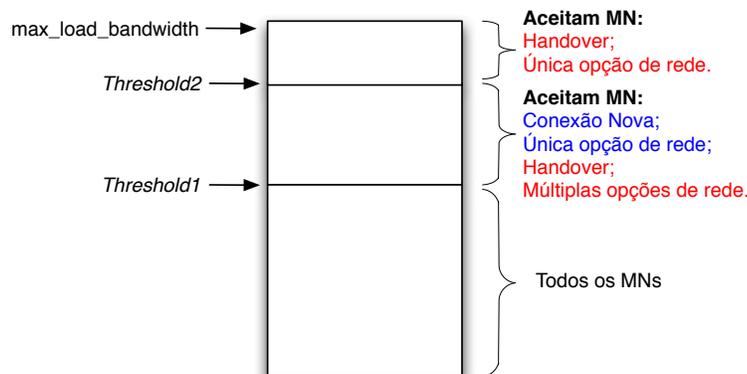


Figura 5.2: *Thresholds* de prioridade

Uma vez confrontadas as métricas solicitadas pelos MNs com as opções oferecidas pelas redes, o algoritmo reconhece três casos: O primeiro caso ocorre quando o MN está iniciando uma nova conexão com a rede (*newConnection*) e a opção de conexão com a rede não é única (*not isUnique*) (linha 2), ou seja, existe outra rede na qual o MN pode se conectar. Neste caso o algoritmo verifica se a carga atual (*actual.load*) da rede mais a banda que foi solicitada (*bandwidth*) é menor ou igual ao *threshold1* e menor ou igual a

carga máxima suportada pela rede *max_load_bandwidth* (linhas 3 e 4). Neste caso a rede aceita a conexão (linha 6).

O segundo caso ocorre quando o MN também está iniciando uma nova conexão e esta conexão é a única disponível (*isUnique*), ou o MN está tentando fazer um *handover* (*fromHandover*) e existe mais de uma opção de rede para se conectar (linha 8). Neste caso o algoritmo verifica se a carga da rede atual mais a banda que foi solicitada é menor ou igual ao *threshold2* e menor ou igual a carga máxima suportada pela rede (linhas 9 e 10). Neste caso a rede aceita o pedido de conexão (linha 12).

O terceiro e último caso ocorre quando o MN está tentando fazer um *handover* e tem apenas uma opção de rede disponível (linha 14). Neste caso o algoritmo aceita o MN (linha 17) se a carga atual da rede mais a banda solicitada for menor ou igual à capacidade máxima suportada pela rede (linha 15).

Algoritmo 5 Algoritmo de Decisão de Handover

```

1. if( (cost >= networkCost) and (bandwidth <= networkBandwidth) ) {
2.     if (newCon) and (not isUnique) {
3.         if (((actual_load + bandwidth) <= threshold1) and
4.             ((actual_load + bandwidth) <= max_load_bandwidth) ) {
5.             actual_load = actual_load + bandwidth;
6.             connection accepted; }
7.         }else connection refused; }
8.     else if((newCon)and(isUnique))or((fromHandover)and(not isUnique)){
9.         if (((actual_load + bandwidth) <= threshold2) and
10.            ((actual_load + bandwidth) <= max_load_bandwidth) ) {
11.             actual_load = actual_load + bandwidth;
12.             connection accepted;
13.         }else connection refused; }
14.     else if (fromHandover) and (isUnique) {
15.         if (((actual_load + bandwidth) <= max_load_bandwidth) {
16.             actual_load = actual_load + bandwidth;
17.             connection accepted; }
18.         else connection refused; }
19.     else connection refused;
20. } else connection refused;

```

Em todas as três situações em que o algoritmo aceita o pedido de conexão, a banda solicitada é reservada (linhas 5, 11 e 16) e o pedido de conexão é armazenado com um identificador único até um certo *timeout*. Caso o MN não se conecte, o recurso é liberado

quando o *timeout* expirar.

O algoritmo proposto tem uma vantagem em relação a outras soluções encontradas na literatura [79, 83]. O cálculo da decisão principal não é feito pelo MN, deste modo exige menos processamento no MN aumentando a autonomia da bateria do dispositivo. Parte do cálculo é feito pelas redes (Figura 5.1, passo 5) e a outra parte, que exige mais processamento, é feito pelo IS no módulo SAW (Figura 5.1, passo 7). O MN é quem fica com a decisão final da escolha da rede, porém esta escolha exige pouco processamento, pois o MN escolhe, dentre uma lista, a rede que apresenta a maior pontuação.

Os procedimentos que o MN irá executar a partir do momento em que se decide em qual rede irá se conectar estão fora do escopo desta tese.

5.2 Simulador de *handover* (Handsim)

Para avaliar o algoritmo de decisão de *handover* apresentado na seção anterior, foi desenvolvido um simulador de eventos discretos na linguagem de programação Java, o Handsim. Com este simulador foram realizados experimentos que simulam a taxa de bloqueio dos MNs pelas redes e a taxa de utilização das redes sem fio com tecnologias WiFi, WiMAX e Celular.

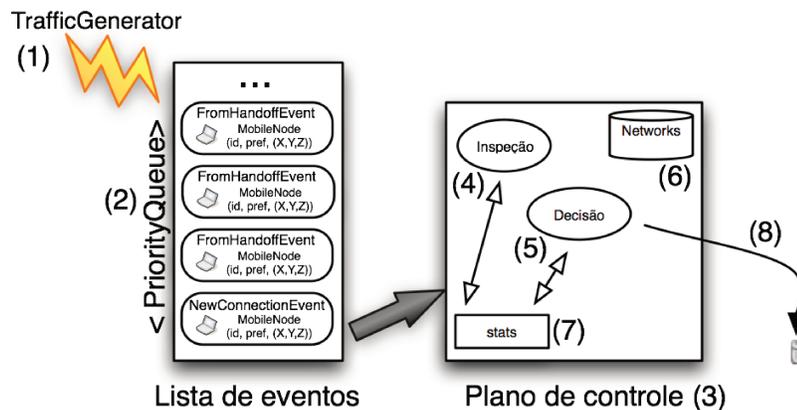


Figura 5.3: Arquitetura do simulador de eventos discretos Handsim.

A Figura 5.3 exibe a arquitetura do simulador. Inicialmente, o módulo *TrafficGenerator* (1) alimenta uma lista de eventos. Esta lista de eventos simula a chegada dos MNs numa área geográfica. Ao chegar nesta área, o MN pede ao IS uma conexão com uma rede sem fio ao seu alcance. O tempo de chegada dos dispositivos é gerado a partir de uma distribuição exponencial com média definida pela variável *MEANARRIVALTIME*.

A lista de eventos é uma estrutura do Java denominada *PriorityQueue* (2) e sempre é ordenada pelo tempo de chegada. O tamanho da lista de eventos é definida pela variável *EVENTSNUMBER*.

Na estrutura *PriorityQueue* existem 2 tipos de eventos: *FromHandoverEvent* que representa um pedido de conexão quando o MN está tentando fazer um *handover*, e *NewConnectionEvent* que representa um pedido de conexão quando o MN está tentando fazer uma nova conexão.

O módulo *TrafficGenerator* cria os eventos para a lista de eventos seguindo uma distribuição uniforme. Dentro de cada evento existe um tipo *MobileNode*, que contém as informações do MN. O *MobileNode* contém um *id*, as preferências dos usuários (banda e custo, com seus respectivos pesos), e as coordenadas em que o dispositivo móvel se encontra no espaço. O *id* é uma informação sequencial.

Para cada evento de chegada criado pelo *TrafficGenerator* outro evento de saída é criado, o evento *FlowDepartureEvent*. Este evento contém apenas o *id* do evento de chegada e o tempo. Este tempo vai determinar quanto tempo o dispositivo móvel permanece no sistema (*holding time*) e é gerado seguindo uma distribuição exponencial com média definida pela variável *MEANDEPARTURETIME*.

Depois de criada a lista de eventos pelo módulo *TrafficGenerator* (1), o plano de controle (3) retira estes eventos da lista e então os envia para o módulo de inspeção (4) que verifica o tipo de evento e envia um pedido de conexão para o módulo de decisão (5). Este módulo de decisão simula o IS e as requisições para as redes disponíveis que são previamente cadastradas no sistema e fica localizado no módulo *Networks* (6). Neste módulo são cadastradas as informações referentes às redes sem fio. Estas informações incluem, por exemplo, o modelo de propagação, a capacidade da rede, as coordenadas da estação base e a potência do sinal. Inicialmente foi implementado redes do tipo *WiFi*, *WiMAX* e Celular.

Para definir quais redes estão disponíveis para o MN, o módulo de decisão verifica qual modelo de propagação está configurado para ser utilizado no simulador. Por exemplo, para o modelo de propagação *TwoRayground*, o módulo de decisão utiliza a equação (5.1).

$$Pr = \frac{P_t * G_t * G_r * (ht^2 * hr^2)}{d^4 * L}, \quad (5.1)$$

onde Pr é a potência recebida, P_t é a potência do transmissor, G_t é o ganho da antena do transmissor, G_r é o ganho da antena do receptor, ht é o tamanho da antena do transmissor, hr é o tamanho da antena do receptor, d é a distância entre a estação base e o dispositivo móvel, e L é a perda do sistema.

Devido às muitas variações no meio, este modelo utilizado para calcular a potência do sinal é bastante simplista, porém é o modelo utilizado no simulador de rede ns-2 e também adotado no Handsim. Os valores utilizados na equação (5.1) foram retirados do simulador

ns-2. Para $P_t = 0.281838$, G_t e $G_r = 1$, h_t e $h_r = 1,5$, $L = 1$, d é calculado usando as coordenadas fornecidas dentro do evento e as coordenadas da estação base obtidas da rede cadastrada no módulo *Networks* (6).

Depois de obtido o valor de P_r , o módulo de decisão (5) compara essa potência com a variável *thresholdSignal*, caso P_r seja maior ou igual ao valor desta variável a rede se torna candidata e o IS fica apto para fazer pedidos para estas redes.

Todos os módulos dentro do simulador utilizam o módulo *stat* (7) que alimenta as estatísticas. Neste módulo são computados os MNs aceitos e bloqueados pelas redes e a utilização dos recursos da rede.

Na próximas subseções serão detalhados os experimentos realizados utilizando o Hand-sim como também os resultados obtidos.

5.3 Experimentos de simulação realizados com o algoritmo de decisão

Esta seção descreve os experimentos de simulação realizados para avaliar o algoritmo de decisão de *handover* em termos de utilização das redes e taxa de bloqueio dos MNs pelas redes. Para a simulação foi utilizado o simulador HandSim que foi detalhado na seção anterior.

A Figura 5.4 exibe o cenário usado nos experimentos. Ele simula a situação de três redes sem fio com diferentes tecnologias. Um *access point* (AP) e duas estações bases (BS) estão localizadas na mesma posição. Os MNs estão posicionados uniformemente em torno delas. Os MNs estão na mesma área de cobertura das 3 redes. A primeira rede sem fio tem tecnologia WiFi, com a potência de transmissão do AP configurada em 0.281838 dBm e a largura de banda de 25.6 Mbps. A segunda rede sem fio tem tecnologia WiMAX, com a potência de transmissão da BS configurada em 0.481838 dBm e a largura de banda de 65 Mbps. A terceira rede sem fio é do tipo celular, com a potência de transmissão da BS configurada em 4.982838 dBm e a largura de banda de 41 Mbps.

Não foi considerada a existência de interferência entre os MNs e os AP/BSs. Foi implementado um mecanismo de controle de admissão nas redes. Este controle vale para as três tecnologias utilizadas nos experimentos. Quando as redes recebem um pedido de conexão do MN vindo do IS, elas verificam se podem atender a esse pedido comparando sua carga atual mais o pedido de banda do MN é menor ou igual a sua largura de banda máxima suportada (veja o Algoritmo 5, linhas 4, 11 e 17).

Todos os experimentos realizados com o HandSim foram executados 20 vezes e foi calculado um intervalo de confiança de 95%.

Foi feito um estudo comparativo onde foi confrontado o algoritmo de decisão de *han-*

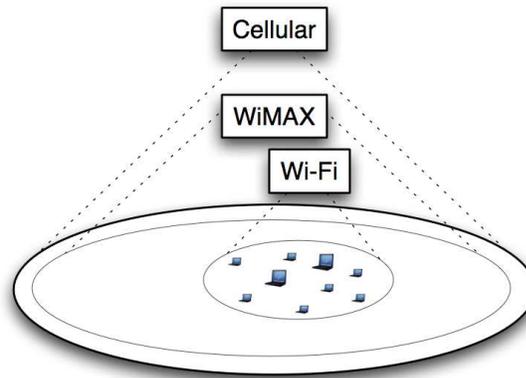


Figura 5.4: Distribuição dos MNs na área de cobertura das redes Celular, WiMAX e Wi-Fi

dover (ADH) proposto com o algoritmo de decisão usado pela maioria dos fabricantes de redes Wi-Fi (algWi-Fi), onde o MN escolhe uma rede, dentre as possíveis candidatas, avaliando sua RSSI (*Received Signal Strength Indicator*) recebida localmente.

A grande desvantagem do algWi-Fi é que ele não leva em consideração as preferências dos usuários ao escolher uma rede, e sempre escolhe uma rede baseado no valor do RSSI, mesmo que exista uma rede com RSSI um pouco mais baixo porém com uma carga menor.

5.4 Resultados obtidos com o algoritmo de decisão

Nesta seção os resultados obtidos com o ADH utilizando o simulador HandSim são apresentados. É feita comparação entre o ADH e o algWi-Fi onde a taxa de utilização e a taxa de bloqueio das redes são confrontadas. Os resultados mostram que o ADH apresenta uma melhor utilização das redes e uma melhor distribuição de MNs nas redes quando comparado com o algoritmo algWi-Fi

5.4.1 Taxa de Utilização

As taxas de utilização das redes são exibidas nas Figuras 5.5-5.8. Todos os experimentos são realizados com os MNs posicionados numa área com cobertura das redes Wi-Fi, WiMAX e Celular. É possível perceber a influência do algoritmo ADH quanto a distribuição dos MNs entre as redes Wi-Fi, WiMAX e Celular. Em todos os experimentos, o ADH distribui melhor os MNs entre as redes disponíveis, satisfazendo as preferências dos usuários.

A Figura 5.5 exibe um cenário onde 26 MNs fazem um pedido de conexão ao IS.

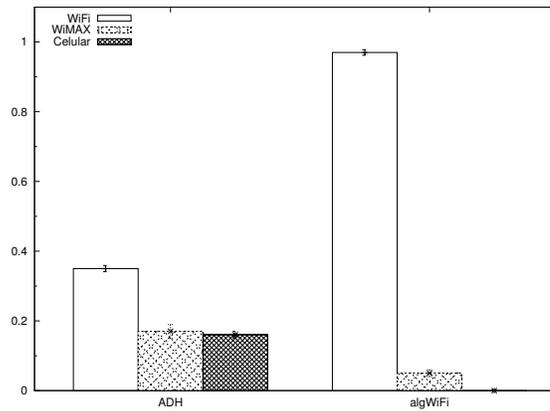


Figura 5.5: Taxa de utilização de rede comparando o ADH com o WiFi utilizando 26 MNs

Comparando o ADH com o algWiFi, nota-se que o algoritmo proposto distribui os MNs entre as redes de forma mais equilibrada. Utilizando o algWiFi, os MNs tendem a se conectar com mais frequência na rede WiFi mesmo havendo outras redes disponíveis que não estão sendo usadas, ou seja sem carga. Note que os MNs se conectam na rede WiFi até sua utilização chegar a quase 100% enquanto as redes WiMAX e Celular estão com uma utilização menor que 5%. Já utilizando o ADH, a rede com a maior carga (WiFi) chega a 40% enquanto que a carga das redes WiMAX e Celular não ultrapassa os 20%.

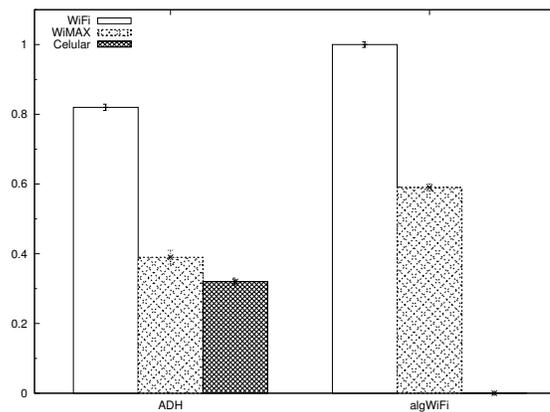


Figura 5.6: Taxa de utilização de rede comparando o ADH com o WiFi utilizando 50 MNs

A Figura 5.6 exibe um cenário onde 50 MNs fazem um pedido de conexão ao IS. Utilizando o ADH, 80% dos MNs se conectaram na rede WiFi enquanto que as redes WiMAX e Celular tiveram uma utilização de no máximo 40%. Utilizando o algWiFi, a rede WiFi fica sobrecarregada, 100% de utilização, enquanto que a utilização da rede

WiMAX chega a 60% e a da rede Celular fica em 0%.

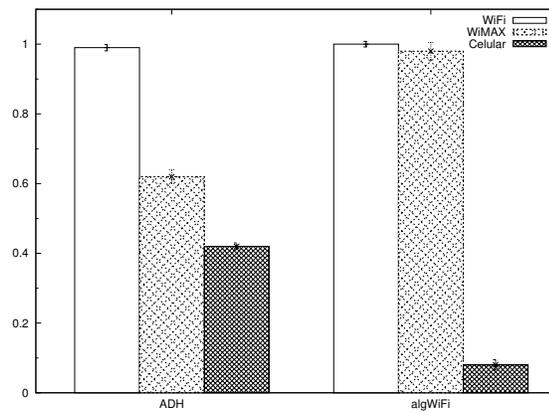


Figura 5.7: Taxa de utilização de rede comparando o ADH com o WiFi utilizando 70 MNs

A Figura 5.7 exibe um cenário onde 70 MNs fazem um pedido de conexão ao IS. Novamente, num cenário com 70MNs, o ADH distribui melhor os MNs entre as redes, a carga da rede WiFi chega a 100% enquanto que a carga da rede WiMAX chega a 60% e a carga da rede Celular chega a 40%. Utilizando o algWiFi, as redes WiFi e WiMAX chegam a uma carga de 100% enquanto que a rede Celular fica subutilizada, com carga de 10%.

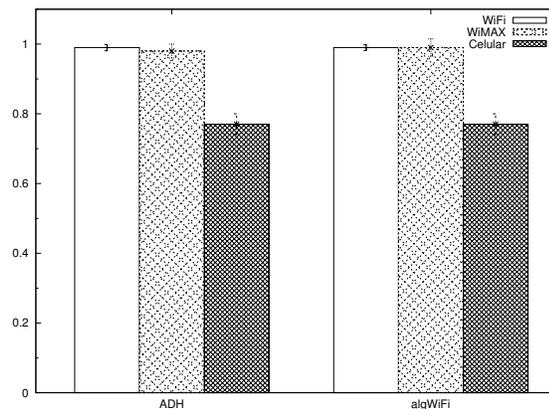


Figura 5.8: Taxa de utilização de rede comparando o ADH com o WiFi utilizando 100 MNs

Por fim, o experimento exibido na Figura 5.8, representa um cenário onde 100 MNs fazem um pedido de conexão ao IS. Neste caso, ambos os algoritmos apresentam o mesmo comportamento, distribuindo igualmente os MNs entre as redes disponíveis.

Com base nos resultados obtidos, pode-se concluir que o ADH dispõe melhor os MNs entre as redes se comparado com o algWiFi. Esta situação ocorre justamente por o algWiFi sempre priorizar as redes do tipo WiFi, quando elas estão disponíveis. Já o ADH leva em consideração as preferências dos usuários (banda e custo) na escolha da rede, além disso, leva em consideração a carga da rede, fazendo um balanceamento de carga entre as redes disponíveis.

5.4.2 Taxa de Bloqueio

As Tabelas 5.1-5.4 exibem os experimentos onde foram contabilizadas as taxas de bloqueio utilizando os algoritmos ADH e algWiFi. Como mencionado anteriormente, a taxa de bloqueio considerada é o bloqueio efetuado pela rede sendo que o MN pode ser bloqueado por uma rede mas pode ser atendido por outra rede. Quando a carga das redes é baixa, a taxa de bloqueio das redes permanece baixa em ambos os algoritmos. À medida que a carga das redes aumenta, ou seja, mais MNs se conectam nas redes, o algWiFi apresenta uma taxa de bloqueio maior que o ADH.

Tabela 5.1: Taxa de bloqueio - 26MNs

	WiFi	WiMAX	Celular
ADH	0	0	0
algWiFi	0.05	0	0

Tabela 5.2: Taxa de bloqueio - 50 MNs.

	WiFi	WiMAX	Celular
ADH	0	0	0
algWiFi	0.43	0	0

A Tabela 5.1 e a Tabela 5.2 exibem os experimentos onde foram usados 26 e 50 MNs respectivamente. Em ambos os casos não foram registrados bloqueios usando o ADH. Já utilizando o algWiFi foi observado um bloqueio de 5% para um cenário com 26 MNs e 43% para um cenário com 50 MNs realizado pela rede WiFi.

Tabela 5.3: Taxa de bloqueio - 70 MNs.

	WiFi	WiMAX	Cellular
ADH	0.1	0	0
algWiFi	0.59	0.07	0

Tabela 5.4: Taxa de bloqueio - 100 MNs.

	WiFi	WiMAX	Cellular
ADH	0.36	0.06	0
algWiFi	0.71	0.43	0

A Tabela 5.3 e a Tabela 5.4 exibem os experimentos onde foram usados 70 e 100 MNs respectivamente. Quando o cenário tem 70 MNs (Tabela 5.3), utilizando o ADH, é observado apenas 10% de bloqueio realizado pela rede WiFi. Já utilizando o algWiFi, este bloqueio aumenta para 59% realizado pela rede WiFi e 7% realizado pela rede WiMAX. Para um cenário com 100 MNs (Tabela 5.4), o desempenho do algWiFi piora ao ser comparado com o ADH. A rede WiFi bloqueia 36% dos MNs e a rede WiMAX bloqueia 6% dos MNs utilizando o ADH. Utilizando o algWiFi esse bloqueio aumenta para 71%

realizado pela rede WiFi e 43% realizado pela rede WiMAX. Não foi observado bloqueio realizado pela rede Celular por ambos os algoritmos.

Quando se compara o ADH com o algWiFi nota-se uma melhor distribuição dos MNs entre as redes, pois o algWiFi sempre tenta se conectar numa rede WiFi quando disponível, mesmo quando as outras redes têm uma carga mínima.

5.5 Módulo de *handover* cognitivo - CogHand

O módulo cognitivo de *handover*, denominado CogHand, foi criado com a intenção de ser utilizado na arquitetura de gerência de mobilidade cognitiva para atuar na decisão de *handover*, podendo ser usado tanto sozinho quanto em conjunto com um algoritmo de decisão, como por exemplo, com o ADH (ver Seção 5.1).

Inicialmente o usuário pode escolher métricas que serão coletadas localmente no MN e enviadas para o servidor de informações através de mensagens IEEE 802.21. Estas métricas comporão notas para ranquear as redes ao alcance do MN. De tempos em tempos o CogHand atualiza as métricas na base de dados do IS e também usa essa base de dados para verificar se existe uma rede ao alcance do MN que ofereça uma pontuação maior do que a rede corrente, instruindo o MN a fazer um *handover* para essa rede.

O CogHand foi baseado no CogMAC e foi adaptado para ser usado com o IEEE 802.21. Além disso ele faz uso de um modelo *cross-layer* para alimentar o servidor de informações. Para fins experimentais foi criado um protótipo em que a medida de perda vinda da camada de transporte foi considerada. O CogHand pode ser adaptado para coletar dados de diferentes camadas da pilha de protocolos do TCP/IP.

A Figura 5.9 apresenta o esquema de funcionamento do CogHand. Assim como no CogMAC, o CogHand implementa um laço de controle autonômico que é executado a cada intervalo de tempo. A cada iteração, o CogHand executa três fases: monitoração, otimização e tomada de ação, representadas pelos Algoritmos 6, 7 e 8.

Na fase de monitoração de desempenho (1), o CogHand coleta a informação de perda da camada de transporte do MN e a armazena no IS em sua base de dados local. Estas informações são enviadas do MN para o IS através de mensagens do IEEE 802.21. Junto com as informações da camada de transporte, o CogHand envia a localização do MN. A localização do MN é associada com a informação de perda e a identificação (*id*) da rede sem fio.

Logo depois da fase de monitoração de desempenho é executada a fase de otimização de desempenho (2), nesta fase o CogHand vai consultar a base de dados local e procurar na localização do MN uma rede que ofereça maior pontuação. Esta fase é responsável pela decisão do *handover* e pode ser combinada com um algoritmo de decisão.

Por último é executada a fase de ação (3), com o envio de uma mensagem para o MN

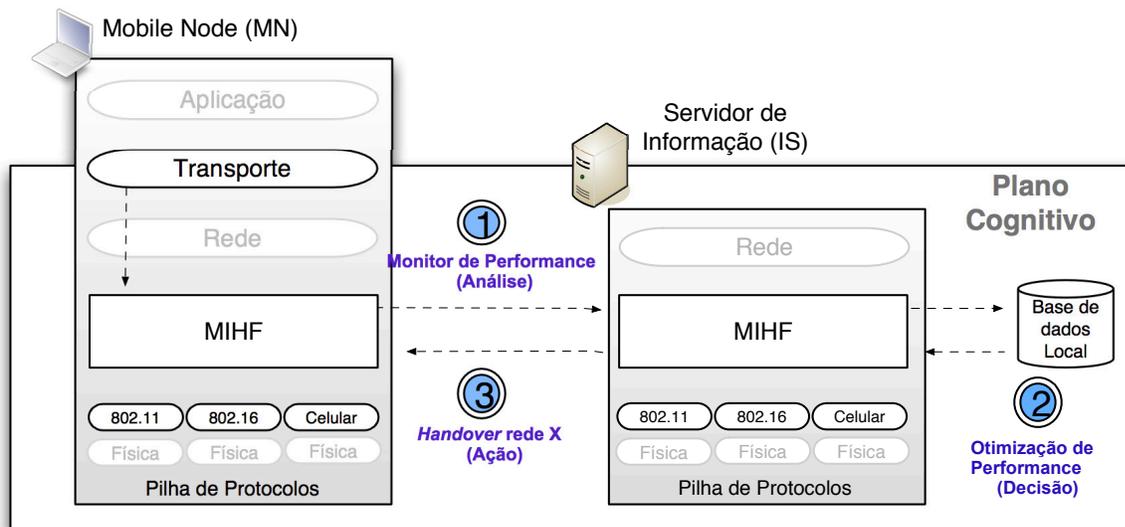


Figura 5.9: Funcionamento do CogHand no MN e no IS

Algoritmo 6 Monitoração de Desempenho

1. **Obtenha** as informações das métricas na camada de transporte no intervalo *SampleInterval*
 2. **Envie** as métricas para a camada MIHF
 3. **Obtenha** o *id* da rede, a localização do MN
 4. **Envie** o *id* da rede, a localização do MN e a perda para o IS
 5. **Obtenha** através do método SAW uma pontuação
 6. **Armazene** a pontuação na base de dados local
-

Algoritmo 7 Otimização de Desempenho

1. **Procure** em todos os elementos da base de dados uma rede ao alcance do MN e armazene em X
 2. **Procure** em X uma rede com a maior pontuação e armazene em Y
-

indicando a rede com a maior pontuação. A partir desse ponto o MN envia um pedido de conexão para a rede escolhida. Caso a conexão entre o MN e o IS se torne indisponível, o MN usa o algoritmo tradicional para seleção de redes que escolhe a rede baseado na qualidade do sinal (ver Seção 5.3) dando prioridade para redes do tipo WiFi.

Algoritmo 8 Ação

1. **Caso** a rede corrente seja diferente da rede armazenada em Y faça
 2. **Envie** uma mensagem para o MN indicando a rede em que se conectar
-

À medida que o algoritmo é executado, a base de dados do IS vai sendo atualizada com as informações das métricas escolhidas, localização do MN e *id* da rede fazendo com que futuras decisões de *handover* sejam mais precisas. O CogHand permite ao MN procurar redes no seu alcance que ofereçam melhor qualidade de serviço em tempo real. Por exemplo, um MN que esteja fazendo uso de uma aplicação de Voz sobre IP precisa selecionar uma rede que tenha pouco atraso, já um MN que esteja fazendo uso de uma aplicação FTP precisa selecionar uma rede que ofereça mais largura de banda, e assim por diante.

A pontuação das redes que estão armazenadas na base de dados do IS foi criada a partir do método SAW (ver Seção 2.7), que foi usado no algoritmo de decisão de *handover* ADH (ver Seção 5.1). A distribuição dos pesos é igual para todas as métricas, por exemplo, para os valores de perda e atraso é associado 0,5 de peso para cada métrica.

A Tabela 5.5 mostra um exemplo de uma medição de perda e atraso num determinado intervalo de tempo, no qual foi aplicado o método SAW para pontuar as redes. A maior nota foi 1 para a rede (1)802.11. As demais redes ficaram com a nota 0,25.

Tabela 5.5: Pontuação das redes calculadas com o método SAW

Rede	Perda	Atraso	Nota SAW
(1) 802.11	0,25	0,1	1
(2) 802.16	0,30	0,2	0,25
(3) 802.11	0,35	0,15	0,25

5.6 Arquitetura de gerência de mobilidade cognitiva

A característica mais importante que uma arquitetura de gerência deve ter para suportar mobilidade nas redes TCP/IP é manter a conexão ativa quando um MN faz um *handover*. Para implementar a função de *handover* nas redes TCP/IP é preciso implementar um módulo de gerência de *handover* no núcleo das redes, assim a principal tarefa desse módulo é evitar a ruptura da conexão do MN durante a troca de rede.

Num ambiente móvel, o endereço IP desempenha o papel de localizador e identificador do MN. Como o IP precisa ser trocado quando o MN troca de rede, ou seja, troca seu ponto de conexão, fica impossível para outros MNs estabelecerem uma comunicação utilizando o seu IP fixo. E mesmo que o MN se associe com o novo endereço IP da nova rede através do protocolo DHCP, as conexões da camada de transporte já estabelecidas com endereço IP anterior são interrompidas. Diante destes fatos, é essencial considerar numa arquitetura de gerência de mobilidade uma funcionalidade conhecida como gerência de

localização. Essa funcionalidade informa a localização corrente de um MN para outros MNs que desejem estabelecer uma conexão. Assim quando o MN troca de rede todas as conexões ativas são notificadas sobre a troca do endereço IP com o objetivo de manter a continuidade da sessão.

Algumas considerações também devem ser feitas ao projetar uma arquitetura de gerência de mobilidade. As aplicações que fazem uso da arquitetura precisam funcionar sem nenhuma modificação. Além disso devem ser levadas em consideração as características dos enlaces sem fio. Dependendo da distância entre o MN e os APs/BSs, da quantidade de rádios próximos dos MN e APs, etc, a qualidade do sinal pode ser ruim, ter baixa largura de banda e apresentar altas taxas de erro. Um mecanismo de decisão de *handover* que aprenda o estado da rede e tome decisões baseadas neste aprendizado em tempo real é desejável para melhorar o desempenho das aplicações. Deste modo o MN do usuário pode estar sempre conectado numa rede que lhe ofereça uma melhor qualidade de serviço.

Outro fator importante no projeto de uma arquitetura de gerência de mobilidade é a facilidade do MN coletar informações a respeito das redes. Durante o processo de *handover* o mecanismo de gerência executa 3 fases: coleta de informações, decisão de *handover* e execução de *handover*. Um mecanismo para auxiliar o processo de coleta de informações e conseqüentemente no algoritmo de decisão é fundamental para a agilidade do funcionamento da arquitetura, e portanto para o tempo de *handover*.

Elencadas as funcionalidades essenciais para a construção de uma arquitetura, esta seção apresenta uma arquitetura de gerência de mobilidade cognitiva com o objetivo de prover mobilidade para os MN nas redes TCP/IP de forma a levar em consideração as preferências dos usuários ao trocar de rede.

A arquitetura usa soluções e protocolos já conhecidos na literatura, mas que sozinhos não tem o poder de compor uma arquitetura de gerência de mobilidade completa. O grande desafio de trabalho ao criar esta arquitetura foi adaptar e integrar todas estas soluções de forma transparente para os usuários.

A Figura 5.10 apresenta os participantes da arquitetura com suas respectivas funcionalidades. O MN usa o IP móvel para gerência de *handover* e localização. Também usa o IEEE 802.21 para alimentar a base de dados do IS, receber notificações das camadas de enlace, e receber comandos do IS que serão enviados para o IP móvel para a execução do *handover*. Por último utiliza o CogHand, que auxilia na decisão cognitiva de *handover*.

Em seguida o IS, previsto pelo IEEE 802.21 para auxiliar os MNs com as informações das redes, usa o CogHand que vai trabalhar em conjunto com o método SAW para prover a decisão de *handover* cognitiva e a base de dados para armazenar as métricas enviadas pelos MNs com suas respectivas posições.

Por último, o AP/BS que implementa o IEEE 802.21 e o IP móvel. O AP/BS repre-

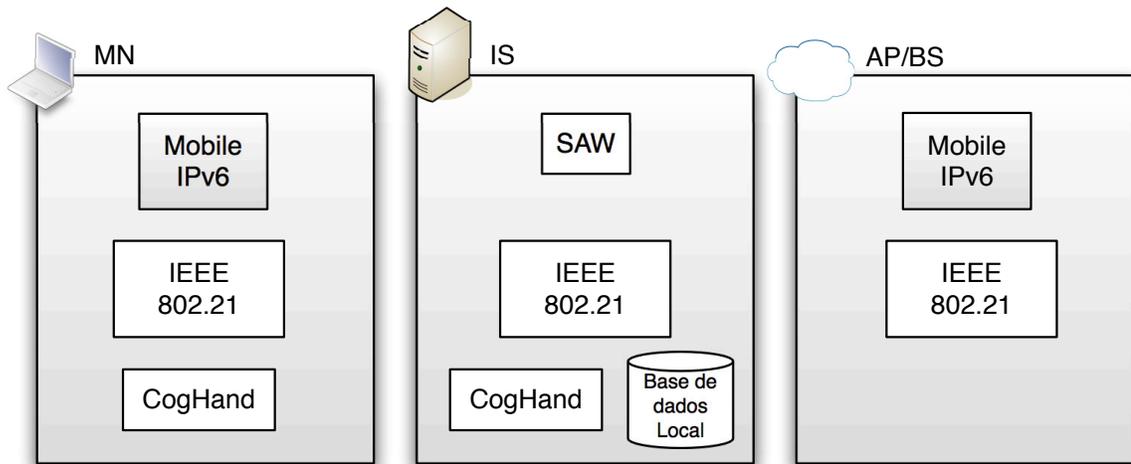


Figura 5.10: Entidades participantes da arquitetura de mobilidade cognitiva

senta a rede em que o MN está conectado. Na implementação do IP móvel essa rede tem dois agentes: O agente nativo (*Home Agent - HA*) e o agente estrangeiro (*Foreign Agent - FA*). Num primeiro momento, o FA é usado pelo HA para encaminhar pacotes para o MN quando ele não está em sua rede nativa.

A Figura 5.11 detalha a arquitetura de gerência de mobilidade cognitiva. Todos os protocolos necessários para prover mobilidade em redes IP estão reunidos e integrados na arquitetura apresentada.

A arquitetura apresentada anteriormente foi implementada no ns-2 utilizando o módulo *Seamless and Secure Mobility* do NIST [5] para o ns-2. Nos próximos parágrafos serão explicadas as modificações realizadas neste módulo para integrar todos os componentes da arquitetura.

Modificações na extensão do ns-2

O módulo do NIST foi implementado no simulador de rede ns-2 versão 2.29. Foi implementada uma versão do IEEE 802.21 baseada no *draft 3*. Foram feitas modificações no ns-2 original pelo NIST para poder adaptar esse módulo ao simulador [6]. Muitas funcionalidades definidas no padrão 802.21 não estão contempladas nesta implementação de forma que algumas funcionalidades propostas originalmente na arquitetura não se refletem nesta implementação.

Usuários MIH são entidades que usam as funcionalidades da camada MIHF para melhorar o desempenho do usuário otimizando os *handovers* [6]. Foi implementado um usuário MIH localizado na camada de transporte. O usuário MIH usa a camada MIHF

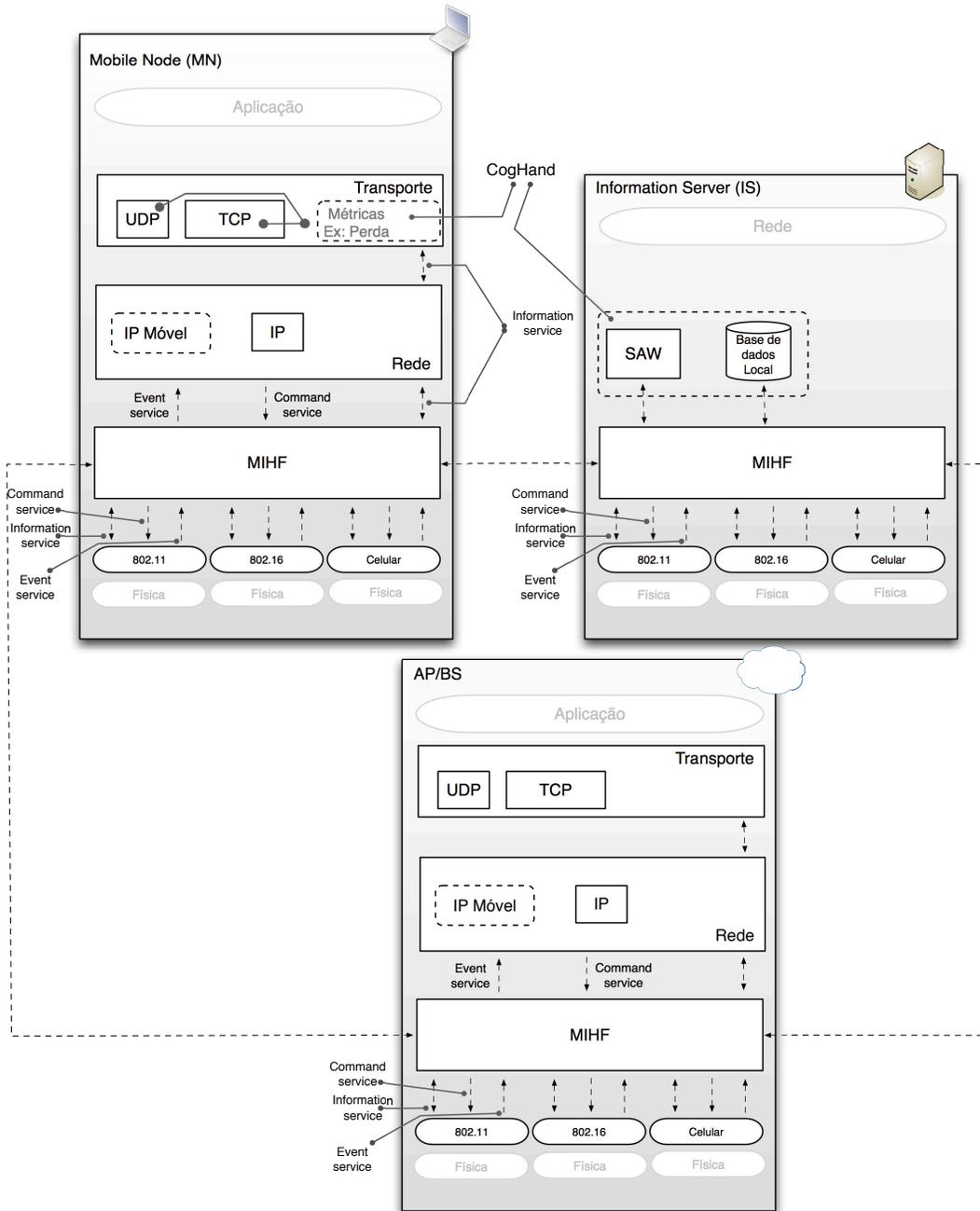


Figura 5.11: Arquitetura de gerência de mobilidade cognitiva

do MN para enviar as informações das métricas de qualidade para o IS remoto. Para fins de experimentação a métrica de perda foi utilizada. Para representar o usuário MIH, foi criada uma classe chamada *MIHUserCogHand* que estende a classe *MIHUserAgent* e foi registrada com o MIHF para receber eventos e poder enviar mensagens para as interfaces locais e remotas.

Além da classe *MIHUserCogHand*, foi criada a classe *Coghand* que foi estendida da classe *Handover*. A classe *Coghand* é responsável pela decisão do *handover*. Todas as funcionalidades do *CogHand* explicadas na Seção 5.5 são implementadas nesta classe. Tanto o IS quanto o MN usam esta classe, entretanto quando é usada uma instância *Coghand* no IS, ela é responsável por implementar o processo cognitivo do *CogHand* ao receber as métricas do MN e armazená-las numa base de dados, além de verificar qual rede fornece melhor pontuação e decidir a qual rede se conectar, enviando uma mensagem para o MN informando esta rede. Quando instanciada no MN, é responsável por receber a mensagem vinda do IS e enviar um comando para a camada MIHF. Então a camada MIHF invoca um comando para a camada de enlace instruindo-a a realizar um *handover* para a rede indicada.

A Figura 5.12 apresenta o diagrama de classes parcial do IEEE 802.21 com seus respectivos arquivos fontes. A figura ilustra a classe *Coghand* que é uma extensão da classe *Handover*. A classe *Handover* é abstrata e é responsável por especificar as funcionalidades que são comuns a todas as implementações *Handover*, como é o caso da classe *Coghand*. A classe *Handover* é uma extensão da classe *MIPV6Agent*. A classe *MIPV6Agent* implementa as funções equivalentes ao agente do IP Móvel.

A classe *MIPV6Agent* é uma extensão da classe *IFMNGMTAgent*. A classe *IFMNGMTAgent* provê as funções de gerência de fluxos, assim os usuários podem registrar os fluxos que estão sendo usados nos MNs. No momento do *handover* os fluxos registrados são direcionados para o IP da nova rede. Por fim, a classe abstrata *MIHUserAgent* trata das mensagens e dos eventos que serão enviados para a camada MIHF.

O funcionamento das tecnologias participantes da arquitetura já foi explicado nas seções anteriores de forma que será detalhada nos próximos parágrafos a interação entre um MN, IS e as redes sem fio.

A Figura 5.13 apresenta o diagrama de sequencia da arquitetura de mobilidade cognitiva. Inicialmente o MN está conectado em um AP/BS, e de tempos em tempos o usuário MIH envia a métrica de perda (1) para o MIHF (2) para ser enviado para o IS. Esse intervalo de tempo é controlado pela variável *sample_interval*. Além da perda, a localização do MN e o id da rede em que ele está conectado também são enviados. A mensagem do tipo *MIH.Get.Information.request* (MGIR) foi adaptada para fazer essa atualização no IS devido à ausência de implementação da mensagem correta no módulo do NIST.

Ao receber a mensagem vinda do MIHF do MN, o IS aplica o método SAW nas

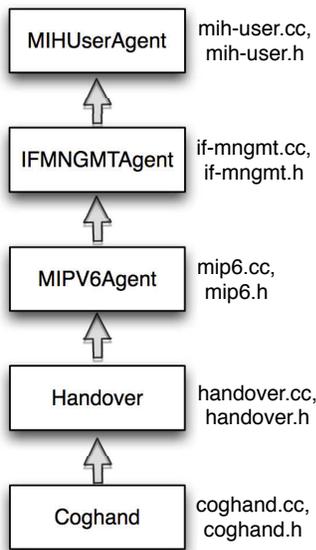


Figura 5.12: Diagrama de classes parcial do IEEE 802.21

métricas para obter uma pontuação (4). Depois disto, o IS armazena essa pontuação juntamente com o id da rede e a localização do MN em sua base de dados local (5). Logo depois, ele verifica na base de dados qual rede apresenta a maior pontuação naquela localização do MN. Caso o MN esteja conectado numa rede diferente da rede com maior nota, o IS envia uma mensagem *MIH_Get_Information_response* (7) para o MIHF do MN indicando a rede para ele se conectar. O MN consulta sua variável *handover_interval* para decidir se faz ou não um novo *handover*. A variável *handover_interval* evita que o MN faça *handovers* desnecessários e com muita frequência criando o efeito *ping pong*. Então o MIHF atualiza o usuário MIH (8). Neste ponto a extensão do módulo gera um evento *LINK_DETECT* (9) juntamente com o id da rede que foi escolhida. Ao gerar um *LINK_DETECT* a implementação do módulo de mobilidade consulta a classe *Coghand* com o objetivo de descobrir se deve ou não se conectar na rede detectada. O evento *LINK_DETECT* foi modificado de forma a incluir o id da rede detectada indicando a rede em que o MN deve se conectar. Ao decidir fazer o *handover* para a nova rede, o CogHand envia um comando *LINK_CONNECT* (10) para o MIHF, que por sua vez faz o pedido de conexão para a nova rede.

Ao fazer o *handover* de uma rede para outra, o módulo de mobilidade do NIST redireciona os fluxos que estão registrados na classe *IFMNGMTAgent*. Como esta parte do código não foi modificada, não foi incluída no diagrama de sequencia.

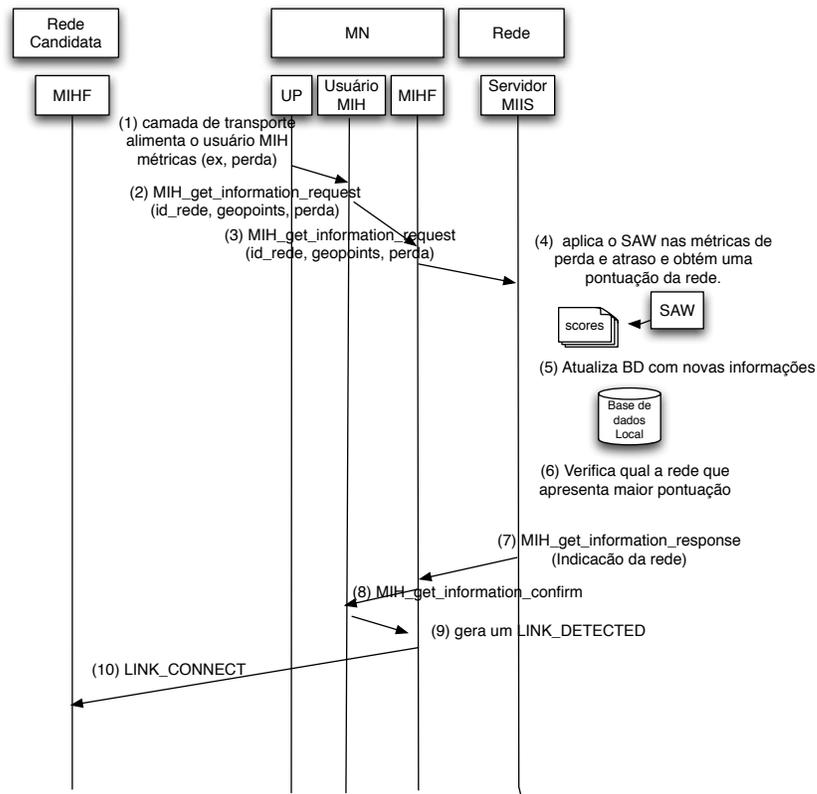


Figura 5.13: Diagrama de sequência da arquitetura de mobilidade cognitiva.

5.7 Experimentos realizados no NS-2

Nesta seção são detalhados os experimentos realizados com a arquitetura de gerência de mobilidade cognitiva. Como relatado anteriormente, foi usado o simulador ns-2 com a extensão *Seamless and Secure Mobility* modificada.

A Figura 5.14 apresenta o cenário utilizado nos experimentos. Um MN com duas interfaces de rede sem fio com tecnologias diferentes (WiFi e WiMAX) está posicionado num local onde há cobertura de duas redes sem fio. O ponto de acesso AP1 é de tecnologia WiFi com alcance de 20 metros de raio a partir de sua localização. A estação base BS1 é de tecnologia WiMAX e tem um alcance de 1000 metros de raio a partir de sua localização. Tanto o AP1 quanto o BS1 estão ligados num nó (Roteador) por um enlace *ethernet duplex* com 100 Mbps de velocidade e 10 ms de atraso. O nó Roteador está ligado no nó correspondente (CN) por um enlace *ethernet duplex* com 100 Mbps de velocidade e 10 ms de atraso. Entre o MN e o nó correspondente foi configurada uma aplicação FTP que roda sobre TCP. No nó correspondente é gerado o tráfego FTP e o MN é configurado como *sink*. Foram realizadas seis simulações de 2000 segundos. A cada simulação foi

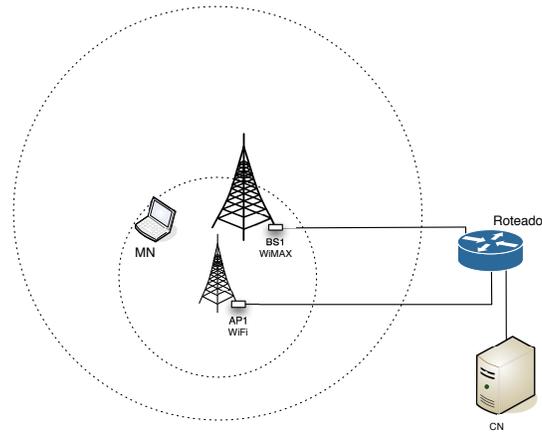


Figura 5.14: Cenário utilizando uma rede WiFi e uma rede WiMAX

alterado o número de MNs no sistema para 5, 10, 20, 30, 40 e 50. Os resultados obtidos são apresentados na próxima seção.

5.8 Resultados obtidos a partir dos experimentos realizados no NS-2

A Figura 5.15 apresenta a vazão TCP com um cenário contendo um MN no sistema. Os algoritmos WiFi e o CogHand são executados, e as vazões TCP resultantes são plotadas no gráfico. Ambos os algoritmos apresentam vazões semelhantes, em torno de 1650 e 1700 kbps.

A Figura 5.16 apresenta o mesmo cenário da figura anterior com dez MNs no sistema. O algoritmo CogHand tem um ganho no valor da vazão em torno de 20% em comparação com o algoritmo algWiFi. Aparentemente ambos os algoritmos apresentam valores de vazões com poucas variações.

A Figura 5.17 apresenta o mesmo cenário da figura anterior com vinte MNs no sistema. Neste cenário o CogHand também apresenta uma vazão superior em comparação com o algoritmo algWiFi. No começo da simulação, no instante 30s ambos os algoritmos apresentam a mesma vazão na média, mas à medida que o tempo avança o CogHand vai melhorando a vazão se estabilizando no instante 900s.

A Figura 5.18 apresenta o mesmo cenário da figura anterior com trinta MNs no sistema. No começo da simulação, entre os instantes 30s e 200s, o CogHand tem um desempenho inferior em relação ao algWiFi. À medida que o algoritmo vai se estabilizando e alimen-

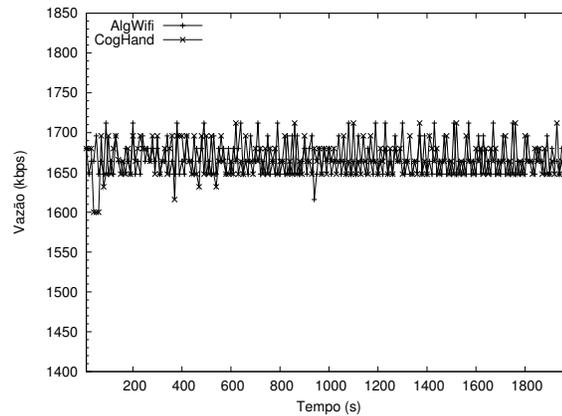


Figura 5.15: Vazão TCP - cenário contendo 1 MN

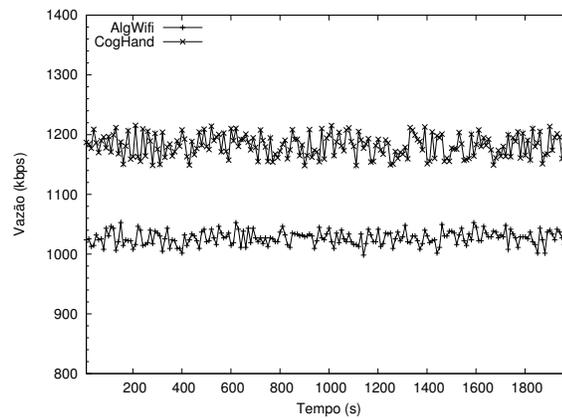


Figura 5.16: Vazão TCP - cenário contendo 10 MNs

tando a base de conhecimento no IS ele vai melhorando seu desempenho. Nos instantes entre 230s e 900s ele tem um desempenho superior em média de 7%, e a partir do instante 1000s o desempenho aumenta para 20% em comparação ao algWiFi.

A Figura 5.19 apresenta o mesmo cenário da figura anterior com quarenta MNs no sistema. Ambos os algoritmos têm desempenho parecido até o instante 300s da simulação. A partir disso o CogHand tem uma leve melhora até o instante 1580s. Entre os instantes 1585s e 1700s, o CogHand tem decréscimo da vazão permanecendo ainda melhor que o algWiFi, e a partir do instante 1700s ele volta a aumentar sua vazão novamente.

Por fim a Figura 5.20 apresenta um cenário com cinquenta MNs. O CogHand apresenta um leve ganho de vazão em relação ao algWiFi até o instante 1300s de simulação. A partir desse instante os dois algoritmos apresentam vazão similar. Isso acontece devido à sobrecarga no sistema. Não importa a escolha dos MNs, pois as redes estão sobrecarregadas. Além das redes sem fio, existe o gargalo dos enlaces entre o roteador e o nó

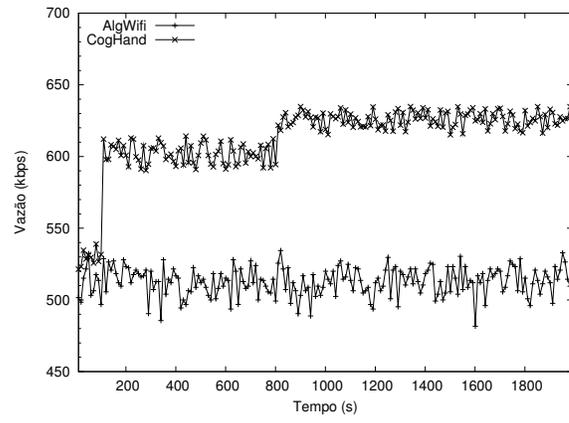


Figura 5.17: Vazão TCP - cenário contendo 20 MNs

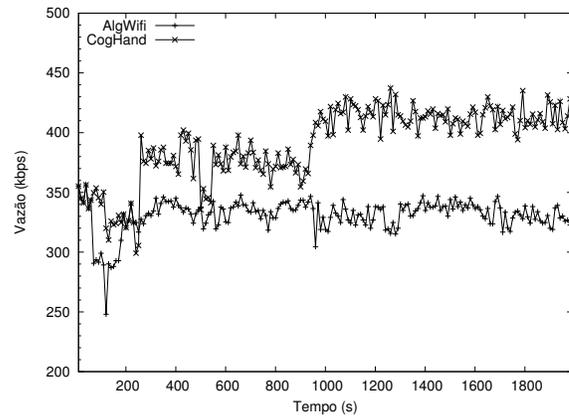


Figura 5.18: Vazão TCP - cenário contendo 30 MNs

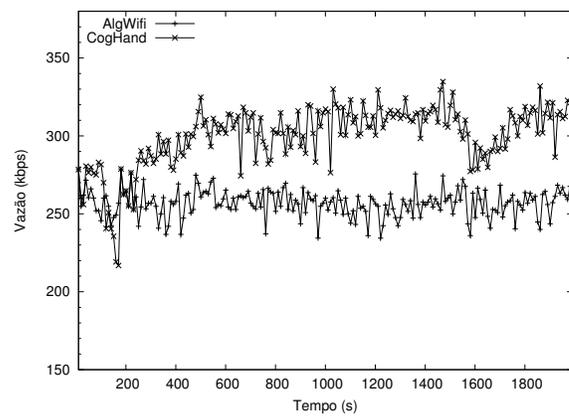


Figura 5.19: Vazão TCP - cenário contendo 40 MNs

correspondente.

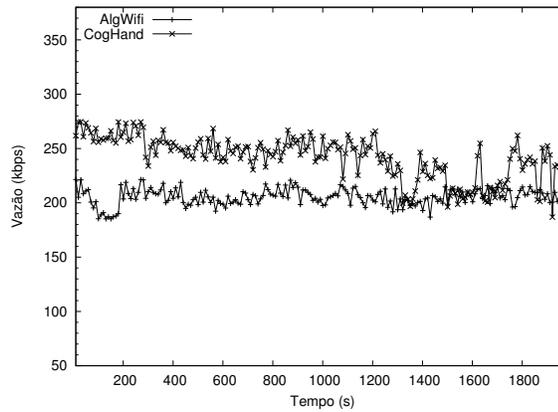


Figura 5.20: Vazão TCP - cenário contendo 50 MNs

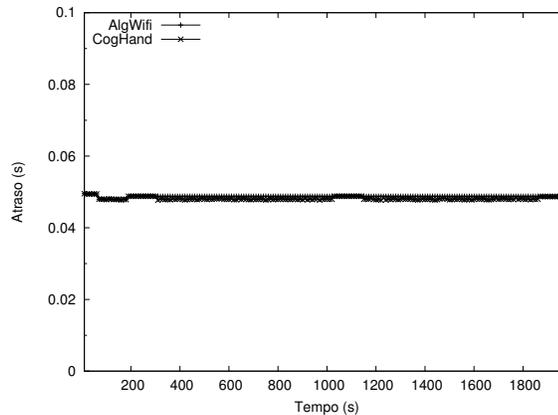


Figura 5.21: Atraso fim a fim - 1MN

O atraso fim a fim entre o MN e o nó correspondente é apresentado na Figura 5.21. Apenas um MN participa deste cenário. Não há diferença no valor do atraso entre os dois algoritmos, que fica em torno de 0,05s.

A Figura 5.22 exibe o atraso fim a fim com um cenário contendo 10 MNs. Ambos os algoritmos aumentaram o valor do atraso para 0,09s. Vale enfatizar que o CogHand teve duas alterações no valor do atraso, superando o algWiFi ao longo da simulação. Entre os instantes 890s e 1010s, e 1720s e 1830s, o atraso aumentou aproximadamente para 0,12s. Novamente aqui pode ser visto que não houve diferença significativa no valor do atraso apesar do CogHand apresentar uma vazão visivelmente maior como mostra a Figura 5.16.

A Figura 5.23 apresenta um cenário com 20 MNs. Neste caso o CogHand tem um atraso maior em relação ao algWiFi na maior parte da simulação. Entre os instantes 50s

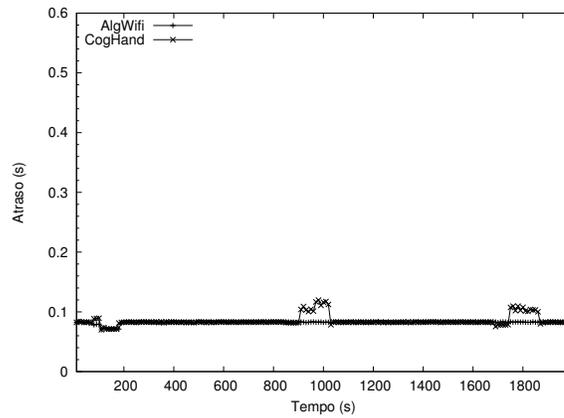


Figura 5.22: Atraso fim a fim - 10MNs

e 300s o CogHand apresenta um atraso similar ao algWiFi, mas a partir deste instante a diferença aumenta para 0,14s.

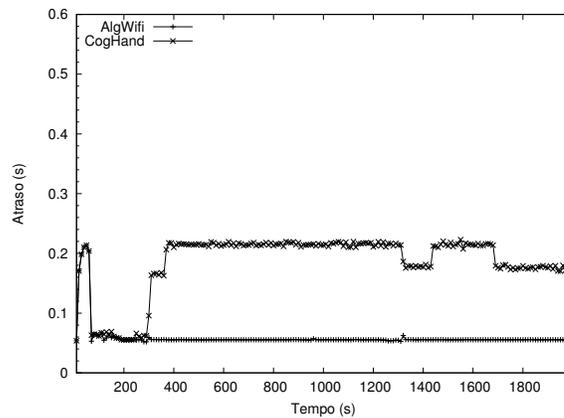


Figura 5.23: Atraso fim a fim - 20MNs

A Figura 5.24 exibe um cenário com 30 MNs no sistema. Como no cenário apresentado pela figura anterior, o atraso do CogHand supera o do algWiFi. Neste caso a diferença é maior, em torno de 0,22, na maior parte da simulação. Há uma variação significativa nos valores do atraso usando o CogHand, isto pode ser explicado pelos sucessivos *handovers* efetuados pelo algoritmo na busca de uma rede que apresente uma melhor vazão, podendo ser confirmado no gráfico apresentado pela Figura 5.18.

A Figura 5.25 exibe um cenário com 40 MNs no sistema. Neste cenário o atraso médio apresentado pelo algoritmo algWiFi é superior ao do CogHand. Neste caso ambas as redes estão sobrecarregadas. Novamente como aconteceu com o cenário com 30 MNs, há uma variação no valor do atraso médio utilizando o CogHand.

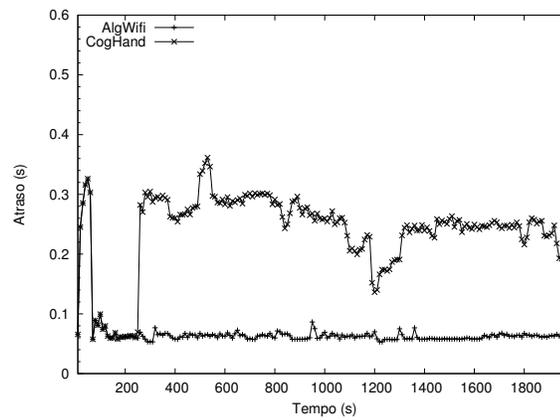


Figura 5.24: Atraso fim a fim - 30MNs

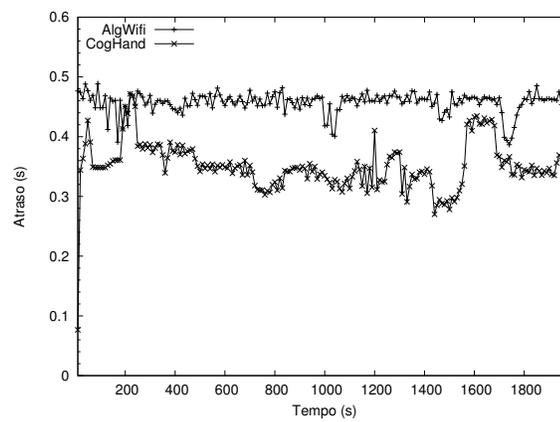


Figura 5.25: Atraso fim a fim - 40MNs

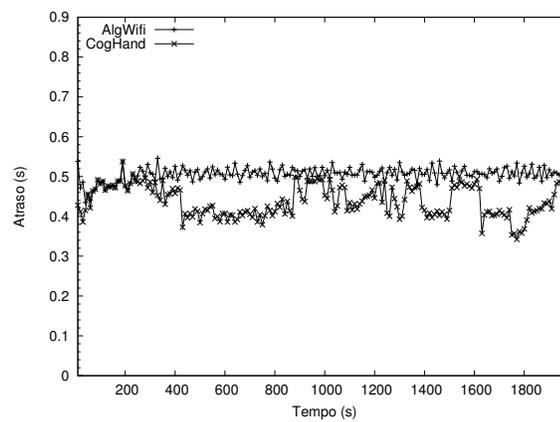


Figura 5.26: Atraso fim a fim - 50MNs

Por fim, a Figura 5.26 apresenta um cenário com 50 MNs. Ambos os algoritmos apresentam atrasos médios parecidos ficando o algWiFi com um atraso um pouco superior. Por mais que o CogHand tente escolher uma rede melhor há uma limitação, pois ambas as redes estão sobrecarregadas e oferecem um desempenho similar, como confirmado pela Figura 5.20.

Tabela 5.6: Perda de pacotes - CogHand X algWiFi (%)

	1MN	5MNs	10MNs	20MNs	30MNs	40MNs	50MNs
CogHand	0	0,058	0,131	0,234	0,229	0,339	0,382
algWiFi	0	0,088	0,211	0,288	0,346	0,451	0,483

A Tabela 5.6 apresenta a perda de pacotes para o CogHand e o algWiFi. Com apenas um MN, ambos os algoritmos não apresentam perdas. À medida que aumenta o número de MNs na rede, o CogHand apresenta uma perda menor em relação ao algWiFi em todos os cenários, para 5, 10, 20, 30, 40 e 50 MNs. Esse ganho do CogHand na perda de pacotes em relação ao algWiFi pode ser explicado pelo fato do CogHand escolher redes para se conectar baseado na menor perda de pacotes.

Como apresentado nos gráficos anteriores, o CogHand ajuda os MNs a escolherem redes de acordo com suas preferências. Neste caso específico foram realizados experimentos onde a perda de pacotes foi levada em consideração na composição da decisão de *handover* baseada em experiências prévias num algoritmo cognitivo. O CogHand escolheu as melhores redes para o MN se conectar em termos de perda obtendo uma melhor vazão, porém algumas vezes teve desempenho inferior ao algWiFi em relação ao atraso médio fim a fim para 20 e 30 MNs como mostrado nas Figuras 5.23 e 5.24. Mesmo nestes casos, o atraso médio observado utilizando o CogHand não foi significativo, no máximo 0,23s para 20 MNs e 0,34s para 30 MNs. Fica evidente que o modelo atual da escolha da rede utilizado por dispositivos, priorizando redes WiFi, não é a melhor decisão a ser adotada, e a tendência é que cada vez mais a decisão de *handover* seja motivada pelas preferências e necessidades dos usuários, fazendo-os participar ativamente na escolha das redes, configurando parâmetros como custo, perda, atraso etc, com uma cooperação cada vez mais presente entre operadoras, como previsto por [9].

Capítulo 6

Conclusão

Como já mencionado por vários autores [55, 33], ainda não existe uma solução perfeita para resolver o problema de mobilidade na Internet. Cada solução proposta pela comunidade científica tem seus prós e contras e de alguma forma contribui para preencher as lacunas dos problemas apontados por diversas pesquisas que tratam sobre o tema [69, 63, 9, 27].

Pensando em contribuir com uma solução para resolver o problema de mobilidade, o foco deste trabalho foi o desenvolvimento de uma arquitetura de gerência de mobilidade cognitiva que levasse em consideração a necessidade de integração das novas tecnologias de redes sem fio e que de alguma forma desse importância às preferências dos usuários ao selecionar um recurso de rede. Além disso, foram explicadas as noções básicas de qualidade de serviço em redes IP com um breve resumo das tecnologias disponíveis e já estudadas. Depois foram apresentadas as principais tecnologias de redes sem fio mais utilizadas pelo mercado, incluindo WiFi e Celular. Na sequência foi descrito o padrão 802.21 criado pelo IEEE com o intuito de integrar e padronizar as informações das diversas tecnologias de redes visando agilizar o processo de *handover* dos mecanismos de gerência existentes. Em seguida foi apresentado o método de decisão multicritério chamado SAW. E por fim foi descrito o CogProt, arcabouço cognitivo para autoconfiguração de protocolos de comunicação com princípios cognitivos. Logo em seguida foi descrito o conceito de redes cognitivas seguido de uma revisão bibliográfica sobre o assunto comentando os trabalhos mais relevantes. A ideia das redes cognitivas é permitir que os MNs sejam capazes de observar as condições da rede e depois planejar e agir de acordo com estas condições. Em sequência foi apresentado o problema da gerência de mobilidade e seus principais desafios. Foram classificados os principais trabalhos por camada do TCP/IP, processo cognitivo, tipo de *handover* suportado e suporte ao IEEE 802.21. Uma parte dos trabalhos pesquisados relacionados à gerência de mobilidade foi dedicada aos algoritmos de decisão de *handovers* que tratam da decisão do MN na hora de escolher uma rede. Foram discutidas

as vantagens e as limitações de cada algoritmo, as quais serviram para a construção de uma tabela comparativa.

A pesquisa sobre como projetar um mecanismo cognitivo para a arquitetura de gerência de mobilidade resultou na criação de um arcabouço cognitivo para a reconfiguração automática dos parâmetros da subcamada MAC do IEEE 802.11 denominado CogMAC. Os parâmetros *retry limit*, *contention window* e *RTS/CTS threshold* da subcamada MAC são configurados dinamicamente pelo arcabouço. Primeiramente o arcabouço CogMAC foi implementado no ns-2. Verificada a eficácia do mecanismo, foi implementada uma versão do CogMAC num *driver* real para uma placa de rede sem fio. O *driver* de dispositivo foi implementado como um módulo do *kernel* do GNU/Linux e testado num *testbed* com resultados muito próximos dos resultados obtidos com a implementação do CogMAC no ns-2.

A pesquisa realizada fomentou a criação de um algoritmo de decisão de *handover* baseado no método multicritério SAW. A ideia era preencher uma lacuna do padrão IEEE 802.21 que não especifica o seu algoritmo de decisão. Foram consideradas as preferências dos usuários na composição de uma nota de rede através do método SAW. Foi observado que o algoritmo proposto distribui melhor os MNs entre as redes disponíveis ao seu alcance fazendo com que as redes participantes tenham uma melhor taxa de utilização.

Além da criação do algoritmo de decisão de *handover*, foi criado um mecanismo cognitivo para o uso na arquitetura de gerência denominado CogHand. O CogHand, que foi baseado no CogMAC, utiliza um laço autonômico em que observa as condições da rede na camada de transporte, armazena estes dados numa base de conhecimento, analisa estes dados e depois toma decisões baseadas nestas análises. Vale ressaltar que as métricas utilizadas no CogHand são flexíveis, ou seja, podem ser substituídas com poucas modificações na implementação.

Por fim, foi proposta a arquitetura de gerência de mobilidade cognitiva. A construção desta arquitetura foi baseada nos conhecimentos adquiridos no decorrer deste trabalho através da pesquisa científica. Foi usado o padrão IEEE 802.21 para obter as informações sobre as redes e assim melhorar a qualidade da decisão dos usuários e agilizar o processo de *handover*. Também foi utilizado o IP móvel, que é responsável pelo mecanismo de *handover* redirecionando os fluxos das aplicações quando ocorre uma troca de IP. Ele também é encarregado pela gerência de localização. Também foi usado o método SAW de decisão multicritério para ranquear as redes disponíveis para os usuários. Um modelo *cross-layer* foi adotado com o objetivo de enviar as informações da camada de transporte do MN para a base de dados no IS, alimentando a base de conhecimento que é usada pelo CogHand. Resultados obtidos utilizando a arquitetura de mobilidade num simulador indicam que é possível implantá-la num ambiente real de produção com ganhos reais para os usuários móveis, satisfazendo suas necessidades.

Desta forma, as principais contribuições deste trabalho foram: o detalhamento do estado da arte dos trabalhos que abordam redes cognitivas, dos mecanismos de gerência de mobilidade e dos algoritmos de decisão de *handover*; a proposta do CogMAC desenvolvida a partir do CogProt; a implementação do CogMAC no ns-2 e num *driver* de dispositivo de rede sem fio; um algoritmo de decisão de *handover* e a implementação de um simulador para sua avaliação; a proposta do CogHand desenvolvida a partir do CogMAC, e a implementação e avaliação da arquitetura de gerência de mobilidade cognitiva e do CogHand no ns-2.

Um trabalho futuro é a criação de um *testbed* real para a implantação da arquitetura de gerência de mobilidade cognitiva. A ideia é reproduzir em um ambiente real todos os componentes da arquitetura de gerência. Para isso é preciso ter acesso ao código fonte dos *drivers* das interfaces de rede. O *driver Atheros* para dispositivos IEEE 802.11 é de código fonte aberto e já foi modificado no trabalho publicado em [65].

Além deste, ainda planeja-se alguns outros trabalhos:

- Implementar e avaliar o comportamento do ADH [66] utilizando o ns-2 com diferentes tipos de modelos de mobilidade.
- Implementar o ADH no *testbed* real.
- Estudar os diferentes modelos de banco de dados para serem usados pelo IS em sua base de conhecimento local.
- Testar outras métricas para serem usadas com o CogHand na arquitetura de gerência cognitiva.

Referências Bibliográficas

- [1] Ambient networks. Disponível em <http://www.ambient-networks.org>. Acessado em 12 de Junho de 2008.
- [2] Ath5k linux wireless device driver. available at <http://linuxwireless.org/en/users/Drivers/ath5k>.
- [3] Framework mac80211. Disponível em <http://linuxwireless.org/>. Acessado em 24 de Abril de 2011.
- [4] Iperf. Disponível em <http://iperf.sourceforge.net/>. Acessado em 13 de Abril de 2011.
- [5] Nist project - seamless and secure mobility tool suits. Disponível em <http://www.antd.nist.gov/seamlessandsecure/pubtool.shtml>. Acessado em 06 de Junho de 2012.
- [6] The network simulator ns-2 nist add-on, 2007. Disponível em http://www.nist.gov/itl/antd/emntg/upload/MIH_module.pdf. Acessado em 12 de Dezembro de 2012.
- [7] 3gpp. Umb. Disponível em <http://www.3gpp2.org/Public.html/specs/tsgc.cfm>. Acessado em 13 de Setembro de 2011.
- [8] I.F. Akyildiz. Mobility management in next generation wireless systems. In *Proceedings of the IEEE*, volume 87, pages 1347–1384. IEEE, Agosto 1998.
- [9] I.F. Akyildiz, X. Jiang, and S. Mohanty. A survey of mobility management in next-generation all-ip-based wireless systems. *IEEE Wireless Communications*, 11(4):16 – 28, Agosto 2004.
- [10] I.F. Akyildiz, S. Mohanty, and Jiang Xie. A ubiquitous mobile communication architecture for next-generation heterogeneous wireless systems. *IEEE Communications Magazine*, 43(6):29–36, Junho 2005.

- [11] M. Ayari, Z. Movahedi, G. Pujolle, and F. Kamoun. Adma: autonomous decentralized management architecture for manets: a simple self-configuring case study. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 132–137. ACM, 2009.
- [12] N. Baldo and M. Zorzi. Cognitive network access using fuzzy decision making. *IEEE Transactions on Wireless Communications*, 8(7):3523–3535, 2009.
- [13] N. Banerjee, A. Acharya, and S.K. Das. Seamless SIP-based mobility for multimedia applications. *IEEE Network*, 20(2):6–13, 2006.
- [14] J. Barzilai and FA Lootsma. Power relations and group aggregation in the multiplicative AHP and SMART. *Journal of multi-criteria decision analysis*, 6(3), 1997.
- [15] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, 1998.
- [16] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. RFC 1633, 1994.
- [17] T. Braga, F. Silva, L. Ruiz, and H. Assunção. Redes autonômicas. *Anais dos minicursos do XXVI Simpósio Brasileiro de Redes de Computadores, SBC*, pages 159–208, 2006.
- [18] S.F. Bush and A.B. Kulkarni. *Active networks and active network management: a proactive management framework*. Kluwer Academic/Plenum Publishers, 2001.
- [19] R. Chakravorty, P. Vidales, L. Patanapongpibul, K. Subramanian, I. Pratt, and J. Crowcroft. On Inter-network Handover Performance using Mobile IPv6. Technical report, University of Cambridge, 2003.
- [20] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith. Cops usage for policy provisioning (cops-pr), 2001.
- [21] KN Choong, VS Kesavan, SL Ng, F. de Carvalho, A.L.Y. Low, and C. Maciocco. SIP-based IEEE 802.21 media independent handover - a BT Intel collaboration. *BT Technology Journal*, 25(2):219–230, 2007.
- [22] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund. Quality of service support in IEEE 802.16 networks. *IEEE Network*, 20(2):50–55, 2006.
- [23] B. Ciubotaru and G.M. Muntean. Sasha - a quality-oriented handover algorithm for multimedia content delivery to mobile users. *IEEE Transactions on Broadcasting*, 55(2):437–450, 2009.

- [24] IEEE Computer Society LAN MAN Standards Committee. IEEE 802.11: Wireless LAN medium access control and physical layer specifications. <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>. Acessado em 05 de Fevereiro de 2012.
- [25] IEEE Computer Society LAN MAN Standards Committee. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. <http://standards.ieee.org/getieee802/download/802.11e-2005.pdf>. Acessado em 01 de Julho de 2012.
- [26] B. P. Crow, I. Widjaja, J. G. Kim, and P.T. Sakai. IEEE 802.11: Wireless local area networks. *IEEE Communications Magazine*, 35(9):116–126, 09 1997.
- [27] P. Demestichas, V. Stavroulaki, D. Bosovic, A. Lee, and J. Strassner. m@ angel: autonomic management platform for seamless cognitive connectivity to the mobile internet. *IEEE Communications Magazine*, 44(6):118–127, 2006.
- [28] S. Dobson, R. Sterritt, P. Nixon, and M. Hinchey. Fulfilling the vision of autonomic computing. *Computer*, 43(1):35–41, 2010.
- [29] Fan Du, Ni L.M, and A.H. Esfahanian. Hopover: a new handoff protocol for overlay networks. *IEEE International Conference on Communications*, 5:3234–3239, 2002.
- [30] B. Durand, J. Sommerville, M. Buchmann, and R. Fuller. *Administering QoS in IP Networks*. Syngress, 2001.
- [31] L. Eastwood, S. Migaldi, Q. Xie, and V. Gupta. Mobility using IEEE 802.21 in a heterogeneous IEEE 802.16/802.11-based, IMT-advanced (4g) network. *IEEE Personal Communications Wireless Communications*, 15(2):26–34, 2008.
- [32] C. Eklund, R.B. Marks, K.L. Stanwood, and S. Wang. IEEE standard 802.16: a technical overview of the Wireless MAN TM air interface for broadband wireless access. *IEEE Communications Magazine*, 40:98–107, 2002.
- [33] S. Fernandes and A. Karmouch. Vertical mobility management architectures in wireless networks: A comprehensive survey and future directions. *Communications Surveys & Tutorials, IEEE*, 14(1):45–63, 2012.
- [34] C. Fortuna and M. Mohorcic. Trends in the development of communication networks: Cognitive networks. *Computer Networks*, 53(9):1354–1376, 2009.

- [35] J. Freitag. Provisão de Qualidade de Serviço em Redes IEEE 802.11. Master's thesis, Universidade Estadual de Campinas (Unicamp), 2004.
- [36] J. Freitag and N. L. S. da Fonseca. Scheduler for IEEE 802.16 networks. *IEEE Communications Letters*, 12(4), 2008.
- [37] J. Freitag and N. L. S. Fonseca. Escalonamento com qualidade de serviço em redes ieee 802.16. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 723–736. Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2007.
- [38] D. Funato, K. Yasuda, and H. Tokuda. TCP-R: TCP mobility support for continuous operation. In *Proceedings of International Conference on Network Protocols*, pages 229–236. IEEE, 1997.
- [39] E. Gelenbe, Z. Xu, and E. Seref. Cognitive packet networks. In *Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence*, pages 47–54. IEEE, 1999.
- [40] A. Goldsmith. *Wireless communications*. Cambridge University Press, 2005.
- [41] B. Goode. Voice over Internet Protocol (VoIP). *Proceedings of the IEEE*, 90(9):1495–1517, 2002.
- [42] A. Hasswa, N. Nasser, and H. Hossanein. Generic vertical handoff decision function for heterogeneous wireless. In *Proceedings of Second IFIP International Conference on Wireless and Optical Communications Networks, 2005*, pages 239–243, 2005.
- [43] C. Hoymann and B. Walke. *IEEE 802.16 Wireless Metropolitan Area Networks*, pages 147–195. John Wiley & Sons, Chichester, West Sussex, UK, Novembro 2006.
- [44] C.L. Hwang and K. Yoon. *Multiple attribute decision making: methods and applications: a state-of-the-art survey*. Springer-Verlag New York, 1981.
- [45] IEEE. Air interface for fixed and mobile broadband wireless access systems. <http://ieeexplore.ieee.org/iel5/10676/33683/01603394.pdf?arnumber=1603394>. Acessado em 14 de maio de 2012.
- [46] IEEE. *Standard for Local and Metropolitan Area Network - Part 21: Media Independent Handover Services*. IEEE, 2009.
- [47] Intel. Mobile wimax part i: A technical overview and performance evaluation. <http://www.wimaxforum.org/technology/downloads/>. Acessado em 14 de Julho de 2007.

- [48] ITU. International telecommunication union. Disponível em <http://www.itu.int/net/home/index.aspx>. Acessado em 01 de Fevereiro de 2012.
- [49] ITU. Itu activities on imt-2000. Disponível em <http://www.itu.int/home/imt.html>. Acessado em 01 de Janeiro de 2011.
- [50] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley New York, 1991.
- [51] J. Kim and J. Jang. Low Latency Vertical Handover Using MIH L2-Trigger Algorithm in Mobile IP Networks. *Lecture Notes in Computer Science*, 4742:707, 2007.
- [52] D. Kliazovich, N. C. Malheiros, N. Fonseca, F. Granelli, R. Piesiewicz, and E. R. M. Madeira. Cogprot: A framework for cognitive configuration and optimization of communication protocols. *The 2nd International ICST Conference on Mobile Lightweight Wireless Systems (Mobilight)*, 2010.
- [53] D. Kliazovich, J. L. Oliveira Filho, N. L. S. Fonseca, F. Granelli, and E. R. M. Madeira. Cognitive link layer for wireless local area networks. *IEEE LATINCOM*, 2009.
- [54] J. F. Kurose and K. W. Ross. *Redes de computadores e a Internet uma abordagem top-down*. Addison-Wesley, Novembro 2006.
- [55] D. Le, X. Fu, D. Hogrefe, et al. A review of mobility support paradigms for the internet. *IEEE Communications Surveys & Tutorials*, 8(1):38–51, 2006.
- [56] D. Lee, Y. Han, and J. Hwang. QoS-Based vertical handoff decision algorithm in heterogeneous systems. In *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, 2006*, pages 1–5, 2006.
- [57] N. C. Malheiros, J. L. Oliveira Filho, L. J. Chaves, E. R. M. Madeira, N. L. S. Fonseca, and I. C. Garcia. Uma abordagem cognitiva para redes autoconfiguráveis. In *Workshop de Sistemas Distribuídos Autônomicos, Ouro Preto. Anais do XXVIII Simpósio Brasileiro de Redes de Computadores*, 2012.
- [58] BS Manoj, R.R. Rao, and M. Zorzi. Cognet: a cognitive complete knowledge network system. *IEEE Wireless Communications*, 15(6):81–88, 2008.
- [59] J. Martins, P. Levine, B. Stiller, H.M. Sherif, A. Fumagalli, J. Aracil, and L Valcarenghi. *Managing IP Networks: Challenges and Opportunities*. Wiley-IEEE Computer Society, 2003.

- [60] J. McNair and Fang Zhu. Vertical handoffs in fourth-generation multinet network environments. *IEEE Wireless Communications*, 11(3):8–15, 2004.
- [61] J. Mitola III. Cognitive radio for flexible mobile multimedia communications. *Mobile Networks and Applications*, 6(5):435–441, 2001.
- [62] S. Mohanty and I.F. Akyildiz. A cross-layer (layer 2 + 3) handoff management protocol for next-generation wireless systems. *IEEE Transactions on Mobile Computing*, 5(10):1347–1360, 2006.
- [63] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle. A survey of autonomic network architectures and evaluation criteria. *IEEE Communications Surveys e Tutorials*, (99):1–27, 2011.
- [64] J.L. Oliveira Filho. Tecnologias diffserv como suporte para a qualidade de serviço (qos) de aplicações multimídia: Aspectos de configuração e integração. Master's thesis, Universidade Salvador (Unifacs), 2006.
- [65] J.L. Oliveira Filho, D. Kliazovich, F. Granelli, Nelson L. S. da Fonseca, and E. Madeira. CogMAC - A Cognitive Link Layer for Wireless Area Networks. *Wireless Networks*, 19(6):1337–1347, 2013.
- [66] J.L. Oliveira Filho and E. R. M. Madeira. A mechanism for vertical handover based on saw using iee 802.21. *Second Internacional ICST Conference on Mobile Networks and Management (Monami), Santander, Spain*, 2010.
- [67] E. Onur, Y. Durmus, M.G. Hawas, S.M. Heemstra De Groot, and I.G.M.M. Niemegeers. Collaborative and cognitive network platforms: vision and research challenges. *Wireless Personal Communications*, 58(1):71–93, 2011.
- [68] E. Osborne and A. Simha. *Engenharia de Tráfego com MPLS*. Editora Campus, 2002.
- [69] S. Paul, J. Pan, and R. Jain. Architectures for the future networks and the next generation internet: A survey. *Computer Communications*, 34(1):2–42, 2011.
- [70] C. Perkins. Ip mobility support for ipv4. RFC 3220, 2002.
- [71] R.V. Rao. *Decision making in the manufacturing environment: using graph theory and fuzzy multiple attribute decision making methods*. Springer, 2007.
- [72] L.K. Rasmussen and I.J. Oppermann. Ping-pong effects in linear parallel interference cancellation for CDMA. *IEEE Transactions on Wireless Communications*, 2(2):357–363, 2003.

- [73] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, 2002.
- [74] T.L. Saaty. *Decision making for leaders: the analytical hierarchy process for decisions in a complex world*. Lifetime Learning Pub, 1982.
- [75] D. Sarkar and H. Narayan. Transport layer protocols for cognitive networks. In *Proceedings of the Conference on Computer Communications Workshops (INFOCOM)*, pages 1–6. IEEE, 2010.
- [76] F. Siddiqui and S. Zeadally. Mobility management across hybrid wireless networks: Trends and challenges. *Computer Communications*, 29(9):1363–1385, 2006.
- [77] P. Sutton, L.E. Doyle, and K.E. Nolan. A reconfigurable platform for cognitive networks. In *1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 1–5. Ieee, 2006.
- [78] Andrew S. Tanenbaum. *Redes de Computadores*. Editora Campus, 2003.
- [79] R. Tawil, O. Salazar, and G. Pujolle. Vertical Handoff Decision Scheme Using MADM for Wireless Networks. In *Proceeding of the IEEE Wireless Communications and Networking Conference*, 2008.
- [80] R.W. Thomas, L.A. DaSilva, and A.B. MacKenzie. Cognitive networks. In *Proceedings of the New Frontiers in Dynamic Spectrum Access Networks*, pages 352–360. IEEE, 2005.
- [81] R.W. Thomas, D.H. Friend, L.A. DaSilva, and A.B. MacKenzie. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine*, 44(12):51–57, 2007.
- [82] L. Torvalds. The linux kernel. Disponível em <http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.34.7.tar.bz2>. Acessado em 10 de Abril de 2012.
- [83] S.F. Yang, J.S. Wu, and H.H. Huang. A vertical Media-Independent Handover decision algorithm across Wi-Fi and WiMAX networks. In *5th IFIP International Conference on Wireless and Optical Communications Networks, 2008. WOCN08.*, pages 1–5, 2008.
- [84] K. Yoon and C.L. Hwang. *Multiple attribute decision making: an introduction*. Sage, 1995.