

Universidade Federal Fluminense

ROBSON HILARIO DA SILVA

Uma implementação de um sistema adaptativo para
comunicação multimídia em redes sem fio
infra-estruturadas

NITERÓI

2006

ROBSON HILARIO DA SILVA

**Uma implementação de um sistema adaptativo para
comunicação multimídia em redes sem fio
infra-estruturadas**

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre em Computação. Área de concentração: Sistemas Paralelos e Distribuídos.

Orientador:

JULIUS C. B. LEITE

UNIVERSIDADE FEDERAL FLUMINENSE

NITERÓI

2006

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

S586 Silva, Robson Hilario da.

Uma implementação de um sistema adaptativo para comunicação multimídia em redes sem fio infra-estruturadas / Robson Hilario da Silva. - Niterói, RJ : [s.n.], 2006.

105 f.

Orientador: Julius C. B. Leite.

Dissertação (Mestrado em Ciência da Computação) - Universidade Federal Fluminense, 2006.

1. Multimídia (Sistema especial de computação). 2. Computação paralela e distribuída. 3. Qualidade de serviço. 4. Computação móvel. I. Título.

CDD 006.6

Uma implementação de um sistema adaptativo para comunicação
multimídia em redes sem fio infra-estruturadas

Robson Hilario da Silva

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre em Computação.

Aprovada por:

Prof. Julius C. B. Leite, PhD / IC-UFF (Presidente)

Profa. Anna Dolejsi Santos, D / IC-UFF

Prof. Joni da Silva Fraga, D / DAS-UFSC

Niterói, 23 de Outubro de 2006.

À minha esposa, aos meus pais, irmãos, sobrinhos e a todos os amigos e familiares, pois vocês são os responsáveis pelo meu sucesso!

Agradecimentos

Mais um obstáculo superado! Mais um sonho alcançado, ser mestre! Isso só foi possível graças a todos que, de uma forma ou de outra, participaram dessa conquista, e essa é a oportunidade de agradecer a cada um de vocês.

Gostaria de agradecer, primeiramente, àquele que é responsável por tudo e que dá um sentido pleno à minha vida, Deus! Obrigado por guiar meus passos e iluminar meus caminhos. Contigo terei sabedoria para levar o conhecimento aos meus alunos sem esquecer que ser mestre é vocação, e não simplesmente profissão.

Não existem palavras para expressar meus agradecimentos a você, meu amor. Como amiga, você soube me ouvir e incentivar sempre que precisei; como esposa, está ao meu lado em todos os momentos e, com muito carinho, me mostra os erros e acertos; profissionalmente, em muitos momentos, além de me mostrar os erros, ajudou a corrigi-los! A você Luciana, minha vida, muito obrigado por existir e me amar tanto.

Aos meus pais, Roberto e Maria Lúcia, obrigado por tudo, mas por tudo mesmo. Posso dizer que tudo o que sou devo a vocês; se cheguei até aqui, foi graças ao amor que dedicaram a mim. Com esse amor aprendi a amar. O mestrado é apenas um título, o mais importante na vida é ser feliz e, para isso, basta saber amar. Obrigado por terem sido meus primeiros, e os mais importantes, mestres!

Aos meus irmãos, Roberta e Dayvison, e aos irmãos emprestados, normalmente conhecidos como cunhados, Fabiano e Fabiana. Obrigado por tudo. A você, minha irmã e madrinha, e ao meu cunhado, obrigado pelas palavras de apoio e pelo incentivo. Embora distantes, sempre estiveram ao meu lado com suas orações. A você, meu irmão, obrigado por acreditar que, até mesmo eu, quem diria, chegaria um dia a ser mestre! Obrigado, por essas palavras e pelo seu exemplo, pois decidi ser mestre quando percebi que até mesmo você conseguiria ser médico, quem diria...

A você, minha cunhada e revisora predileta, obrigado pela paciência e pelas horas perdidas. Aos meus primos, emprestados, Priscilla, Rafa e Bruna e aos amigos Rafael

e Terezinha Mansano, obrigado pela torcida e pelo incentivo. A você, Dr. Francisco, obrigado por tudo, pelas orações, pelas conversas, por abrir a porta da sua casa para acolher dois netos e assim nos ajudar, e muito, nesse final de mestrado e, principalmente, por ser para mim, o avô que não tive a oportunidade de conhecer. Aos meus outros pais, emprestados, Antônio Carlos e Angela, obrigado por tudo e, principalmente, pela filha maravilhosa que, hoje, é minha esposa!

Aos amigos, do laboratório da pós, Daniela e Rodrigo, Viviane e Joni, Jacques, Luis, Alessandro Copetti, Cris, Renathinha, Glauco, Bruno, Raphael, Alex, Luciana, Nilmax, Aline e Alexandre, Kennedy e a todos que estiveram sempre me acompanhando e apoiando durante todos esses meses de trabalho.

Aos amigos da época da faculdade, Alan, Heverton, Silvia e Marcos, Rodolfo, Fabrício, Eduardo, Paulinha, Michelle e Laura obrigado por entenderem todas as vezes que não compareci aos eventos, sempre por causa do mestrado.

Aos amigos e funcionário do Instituto Carlinhos, Izabela, Angela e Maria.

Aos membros da banca, pelos comentários e contribuições apresentadas.

À Capes, por financiar os meus estudos durante a maior parte do tempo.

Por fim, agradeço ao meu orientador e amigo Julius Leite. Obrigado pela paciência, pelas conversas, pelos conselhos. Durante esses anos de trabalho, que teve início em 2001, com a iniciação científica, aprendi muitas coisas e só tenho a lhe agradecer. Posso dizer que o seu fator "se vira" é fundamental para conseguirmos superar nossos limites e, principalmente, aprender a aprender.

Resumo

A comunicação sem fio é um importante avanço tecnológico que possibilita a mobilidade e a conectividade, independentemente do lugar e do tempo. Da união dessa tecnologia com técnicas de transferência multimídia podem surgir novas aplicações, como aquelas destinadas aos ambientes hospitalares. Nesses ambientes, por exemplo, é muito comum a necessidade de transferência de imagens em tempo real. Contudo, alguns problemas característicos das redes sem fio necessitam ser superados, a fim de fornecer qualidade de serviço na transmissão de dados multimídia.

Na tentativa de amenizar alguns desses problemas sobre as *streams* de mídia contínua será apresentado um aplicativo, desenvolvido para redes sem fio infra-estruturadas, que visa tratar as instabilidades inerentes a esses ambientes, de forma automática e transparente. Para esse fim, algumas políticas de adaptação foram propostas na tentativa de reduzir os efeitos nocivos de migrações abruptas e interferências nas comunicações.

Palavras-chave:

- (1) Redes infra-estruturadas;
- (2) Transmissão multimídia;
- (3) Qualidade de serviço;
- (4) Computação móvel.

Abstract

Wireless communication is a technological advance that enables device mobility and their interconnectivity, any time and anywhere. The union of this technology with multimedia transmission techniques can bring birth to new applications, like those applied to hospital environments. In these environments, for example, the need for the transfer of real-time images is very common. However, some typical wireless network problems need to be overcome to provide quality of service in multimedia transmissions.

This project presents some solutions to overcome the above referred problems of transmitting continuous media streams over an infrastructured wireless network. The developed solution aims at dealing with the inherent instabilities of the wireless environment, automatically and without the users help. For this purpose, some adaptation policies were proposed in the attempt to reduce the harmful effects of the sudden handoffs and the interferences in the communication.

Keywords:

- (1) Infrastructured networks;
- (2) Multimedia transmission;
- (3) Quality of service;
- (4) Mobile computing.

Sumário

Lista de Abreviaturas	12
Lista de Figuras	13
Lista de Tabelas	16
1 Introdução	17
1.1 Introdução	17
1.2 Redes Sem Fio	18
1.3 Aplicações	19
1.4 Alguns Problemas nas Redes Sem Fio	20
1.5 Objetivo	21
1.6 Roteiro	22
2 Qualidade de Serviço	23
2.1 Introdução	23
2.2 O que é Qualidade de Serviço	23
2.3 Requisitos de QoS em Redes Infra-estruturadas	26
2.4 Conclusão	30
3 Trabalhos Relacionados	32
3.1 Introdução	32
3.2 Redes Sem Fio	33
3.3 Transmissão Multimídia em Redes Sem Fio	37

3.3.1	Formato Multimídia	38
3.3.2	Adaptação Multimídia	39
3.4	Ambiente Hospitalar	40
3.5	Conclusão	40
4	Mecanismos de Adaptação Propostos	42
4.1	Introdução	42
4.2	Caracterização do Ambiente do Projeto	43
4.3	Gerenciamento de <i>Buffers</i>	43
4.4	Políticas de Migração	44
4.4.1	Média Simples	45
4.4.2	Média Simples com Limite Inicial	45
4.4.3	Diferentes Níveis	46
4.5	Indicação da Qualidade do Sinal	47
4.6	Conclusão	48
5	Implementação do Sistema Proposto	49
5.1	Introdução	49
5.2	Proposta do Sistema	49
5.3	Interface e Descrição do Sistema	52
5.3.1	Comunicação	56
5.3.2	Codificação	56
5.3.3	Implementação	58
5.4	Ferramentas	59
5.5	Bibliotecas	60
5.6	Configuração	61
5.7	Conclusão	61

Sumário	10
6 Avaliação do Sistema	63
6.1 Introdução	63
6.2 Cenário	63
6.3 Métricas	65
6.4 Resultados Obtidos	66
6.4.1 Disposição dos Pontos de Acesso com Grande Sobreposição dos Sinais	67
6.4.1.1 Sem Política de Migração	68
6.4.1.2 Política de Migração - Média Simples	68
6.4.1.3 Política de Migração - Média Simples com Limite Inicial	70
6.4.1.4 Política de Migração - Diferentes Níveis	72
6.4.2 Disposição dos Pontos de Acesso com Baixa Sobreposição dos Sinais	74
6.4.2.1 Sem Política de Migração	76
6.4.2.2 Política de Migração - Média Simples	77
6.4.2.3 Política de Migração - Média Simples com Limite Inicial	79
6.4.2.4 Política de Migração - Diferentes Níveis	81
6.5 Conclusão	83
7 Conclusão	85
7.1 Introdução	85
7.2 Trabalhos Futuros	86
Referências	88
Apêndice A – Documentação Técnica do Sistema	90
A.1 Apresentação	90
A.2 Aplicativo Servidor	90
A.2.1 Métodos da Biblioteca DecoderH263	94
A.2.2 Métodos da Biblioteca WRAPI	94

A.3	Aplicativo Cliente	95
A.3.1	Métodos da Biblioteca DecoderH263DLL_ARM	97
A.3.2	Métodos da Biblioteca WiFiConnection	97
A.4	Estruturas das Mensagens do Aplicativo	98
A.5	Principais Eventos do Aplicativo	98

Lista de Abreviaturas

ACK	: <i>Acknowledgement</i>
API	: <i>Application Programming Interface</i>
ARC	: <i>Automatic Rate Control</i>
dB	: <i>decibel</i>
dBm	: <i>dB miliwatt</i>
DHCP	: <i>Dynamic Host Configuration Protocol</i>
DLL	: <i>Dinamic Link Language</i>
GUI	: <i>Graphical user interface</i>
IEEE	: <i>Institute of Electrical and Electronics Engineers</i>
IP	: <i>Internet Protocol</i>
ISM	: <i>Industrial, Scientific and Medical</i>
LAN	: <i>Local Area Network</i>
MAC	: <i>Media Access Control</i>
MTU	: <i>Maximum Trasmit Unit</i>
NIC	: <i>Network Interface Card</i>
NDIS	: <i>Network Driver Interface Specification</i>
QoS	: <i>Quality of Service</i>
WZC	: <i>Wireless Zero Configuration</i>

Lista de Figuras

1.1	Configurações de redes sem fio.	20
1.2	Migração entre pontos de acesso.	21
4.1	Limites de cada faixa da política de migração.	46
4.2	Indicação por cores da qualidade do sinal do ponto de acesso.	48
5.1	Resumo das tarefas do aplicativo.	50
5.2	Conexões possíveis em uma videoconferência.	51
5.3	Transferência dos quadros de vídeo.	52
5.4	Componente para monitorar a rede sem fio.	52
5.5	Iniciando o aplicativo servidor: (a) Iniciar a espera por uma conexão; (b) Conectando com o computador remoto.	53
5.6	Iniciando uma conexão com o servidor: (a) Aceitar ou rejeitar uma conexão; (b) Aceitar ou rejeitar uma conexão de vídeo; (c) Formulário que representa a videoconferência.	54
5.7	Transferência de Vídeo: (a) Vídeo capturado localmente; (b) Vídeo rece- bido pela máquina remota.	55
5.8	Imagens do aplicativo cliente, no <i>Pocket PC</i> : (a) <i>Menu File</i> ; (b) Formulário de conexão com o aplicativo servidor; (c) Vídeo no dispositivo.	55
6.1	Disposição dos sinais dos pontos de acesso: (a) Com muita sobreposição; (b) Com sobreposição reduzida.	64
6.2	Esquema aproximado do trajeto percorrido durante os testes.	67
6.3	Sem política de migração e grande sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	69

6.4	Política de migração Média Simples e grande sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	71
6.5	Política de migração Média Simples com Limite Inicial e grande sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	73
6.6	Política de migração Diferentes Níveis e grande sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	75
6.7	Sem política de migração e baixa sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	77
6.8	Política de migração Média Simples e baixa sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	78
6.9	Política de migração Média Simples com Limite Inicial e baixa sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	80
6.10	Política de migração Diferentes Níveis e baixa sobreposição dos sinais: (a) Ponto de acesso str ; (b) Ponto de acesso compmov ; (c) Ponto de acesso compmov2 ; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.	82
A.1	Classes pertencentes ao pacote principal do aplicativo servidor.	91
A.2	Classes pertencentes ao pacote <i>VideoConference</i> do aplicativo servidor.	92
A.3	Classes pertencentes ao pacote <i>Connection</i> do aplicativo servidor.	93
A.4	Classes pertencentes ao pacote <i>WirelessEnvironment</i> do aplicativo servidor.	94

A.5	Classes pertencentes ao pacote <i>ControlerH263</i> do aplicativo servidor.	95
A.6	Classes pertencentes ao pacote <i>BufferCapturingVideo</i> do aplicativo servidor. .	95
A.7	Classes pertencentes ao pacote <i>CaptureImage</i> do aplicativo servidor.	96
A.8	Relações entre os pacotes do aplicativo servidor.	96
A.9	Classes pertencentes ao pacote principal do aplicativo cliente.	97
A.10	Classes pertencentes ao pacote <i>VideoConference</i> do aplicativo cliente.	98
A.11	Classes pertencentes ao pacote <i>WirelessEnvironment</i> do aplicativo cliente. . .	98
A.12	Relações entre os pacotes do aplicativo cliente.	99

Lista de Tabelas

6.1	Com grande sobreposição dos sinais	83
6.2	Praticamente sem sobreposição dos sinais	83
A.1	Funções da biblioteca DecoderH263.	97
A.2	Estrutura de algumas mensagens do aplicativo.	101
A.3	Funções da biblioteca WRAPI.	104
A.4	Funções da biblioteca WiFiConnection.	105

Capítulo 1

Introdução

1.1 Introdução

A possibilidade de mobilidade e conectividade, independentemente do lugar e do tempo, é uma das vantagens das redes sem fio. O uso dessa tecnologia, juntamente com técnicas de transferência multimídia, pode permitir a implementação de novas aplicações, como a comunicação interna em um hospital, a transferência de imagens do local de um desastre ou ainda operações militares.

Nos hospitais, a comunicação entre as equipes médicas e dispositivos que monitoram os pacientes (como controladores cardíacos e aparelhos de raio-X) poderá agilizar o atendimento em casos de emergência. Para isso, cada médico, através de um dispositivo portátil (*pocket PC* ou celular), pode receber dados multimídia gerados por aparelhos conectados diretamente ao paciente, ou ainda, obter informações eletrônicas sobre um determinado paciente quando estiver próximo ao leito.

É uma tendência crescente a busca por transferência de dados, de forma segura e rápida, através da tecnologia sem fio, que permite mobilidade e flexibilidade. As redes locais tradicionais, as chamadas LANs, transferem dados em altas taxas, o que favorece as aplicações como videoconferências e transferências de *streaming* de vídeo, entre outras. Com o avanço da tecnologia das redes sem fio tornou-se possível a transferência de dados a taxas mais elevadas. Esse avanço faz com que as aplicações que buscam oferecer troca de informações multimídia possam ser utilizadas nessas redes, proporcionando conforto e mobilidade aos usuários. Porém, diversos problemas característicos das redes sem fio podem dificultar a transferência de dados e provocar uma redução considerável na qualidade do serviço.

Alguns desses problemas podem ser: grande instabilidade nos enlaces, o que pode gerar momentos de desconexão; alta variação nas taxas de transferência, ou seja, redução da capacidade de vazão da rede por diversos motivos durante a conexão; alta variação do *jitter*; e, perda de pacotes. Diante desses percalços, as aplicações desenvolvidas sobre as redes sem fio devem ser capazes de se adaptar rapidamente e de forma transparente ao usuário.

1.2 Redes Sem Fio

Uma rede sem fio tem como base uma arquitetura subdividida em células, em que cada célula (chamada de *Basic Service Set* ou simplesmente BSS) é controlada por uma estação base (chamada de Ponto de Acesso ou AP), como citado em [Chandramouli 2004]. Existem algumas soluções disponíveis para protocolos de acesso ao meio em redes, que contêm diversas diferenças em termos de padronização e interoperabilidade, tais como IEEE 802.11, *HyperLAN/4*, *Bluetooth* e *HomeRF*.

O padrão IEEE 802.11, criado em junho de 1997, opera na banda ISM (*Industrial, Scientific and Medical*) 2,4 GHz, com taxas de transferência de 1 e 2 Mbps. Com o passar dos anos, esse padrão sofreu modificações, e versões como 802.11a e 802.11b foram definidas. A primeira versão opera em 5 GHz e taxas de até 54 Mbps, e a versão 802.11b disponibiliza uma taxa de transmissão entre 1 Mbps e 11 Mbps na banda de 2,4 GHz. Entre outros padrões, o 802.11g transfere dados a uma taxa de até 54 Mbps, sendo este o mais utilizado nos dias atuais. O padrão 802.11e tem como objetivo oferecer qualidade às transmissões, implementando mecanismo para reserva de banda na camada MAC. Porém, ainda não existem muitos dispositivos sem fio no mercado que utilizem este padrão.

A solução *HiperLAN/1* foi definida em 1996 pelo ETSI (*European Telecommunications Standards Institute*) e fornece uma taxa de transferência de dados velocidade nas comunicações de 23,5 Mbps entre dispositivos portáteis. No padrão *HiperLAN/4*, a taxa de transmissão atinge 155 Mbps.

A tecnologia *Bluetooth*, como descrito em [Schiller 2003], estabelece uma conexão entre dois dispositivos a uma pequena distância. Geralmente os dois aparelhos devem estar próximos, e não deve haver nenhum obstáculo entre um e outro.

Por fim, a solução *HomeRF* [Schiller 2003] foi desenvolvida pela *Home Radio Frequency Working Group* e define a forma como dispositivos eletrônicos - PCs, telefones sem fio e outros que comunicam através de voz, dados e *streaming* de mídia - realizam a comunicação

no interior e ao redor de uma casa.

As redes sem fio podem ser classificadas em duas categorias:

- **Ad Hoc** - Alguns nós, através de conexão sem fio, se unem para estabelecer uma comunicação ponto-a-ponto. Nessa união, cada nó deve cooperar com seus vizinhos para o encaminhamento das mensagens. Quando o nó destino não se encontrar no raio de alcance do nó origem, uma rota deverá ser encontrada através de um protocolo de roteamento, utilizando para isso os outros participantes da rede e assim garantir a comunicação. Dessa forma, cada nó poderá transmitir e receber dados ou simplesmente viabilizar a transferência entre dois nós distantes fazendo parte de uma determinada rota. Sendo assim, nessas redes os nós compartilham seus recursos sem a necessidade de um servidor central.
- **Infra-estruturada** - Dispositivos móveis se comunicam com estações de trabalho - conectadas em uma rede LAN - através de roteadores sem fio, também conhecidos como pontos de acesso. Esses pontos de acesso permitem que diversos nós móveis compartilhem os recursos disponíveis na rede, de forma eficiente. Antes de iniciar uma comunicação com uma das estações de trabalho ou com outros nós móveis, os dispositivos sem fio e os pontos de acesso devem estabelecer uma associação. Uma característica dos pontos de acesso é a de não ter um longo raio de alcance. Obstáculos como paredes reduzem ainda mais esse alcance, o que leva à necessidade de diversos pontos de acesso em uma mesma rede LAN, para garantir a conectividade em uma área maior.

Essas duas configurações para as redes sem fio são utilizadas por diversas aplicações em diferentes áreas.

1.3 Aplicações

Com o aprimoramento da tecnologia das redes sem fio, principalmente o aumento considerável das taxas de transmissão, surgem muitas aplicações para explorar o potencial de mobilidade proporcionado por essas redes.

No âmbito militar, diversas aplicações são possíveis, como exemplo, aquelas em que objetos de tamanho bastante reduzido podem ser espalhados em território inimigo, formando uma rede de sensores capazes de capturar e transmitir informações. Nesse caso, os sensores formam uma rede *Ad Hoc* com milhares, ou até, milhões de nós.

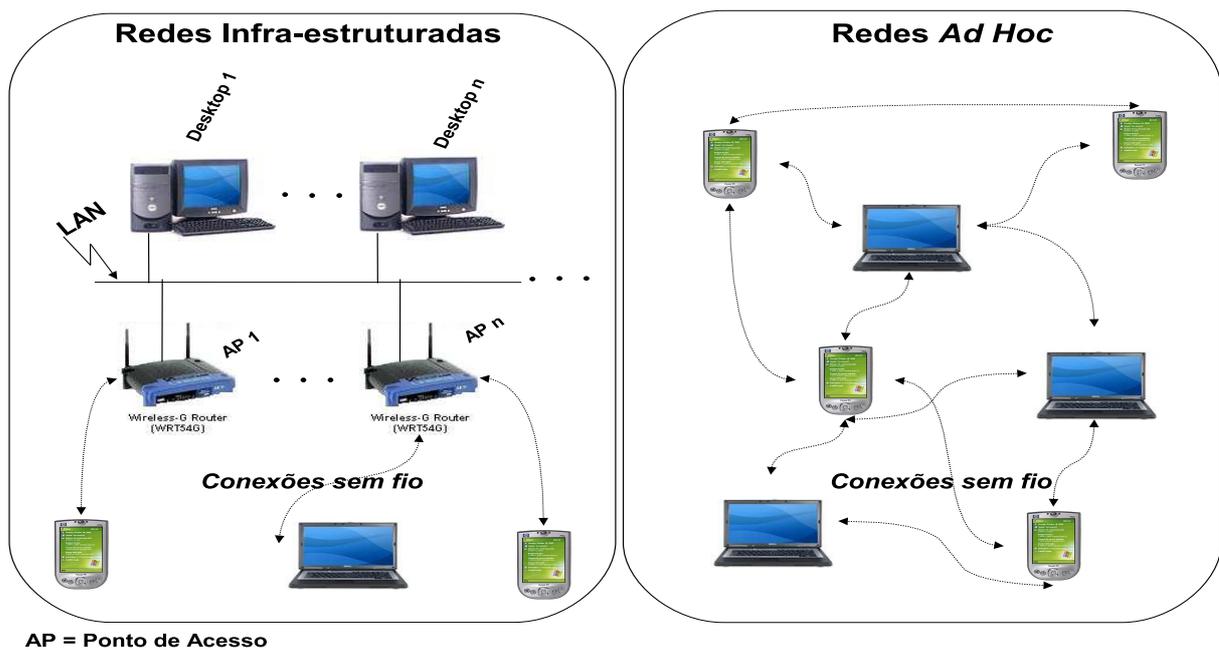


Figura 1.1: Configurações de redes sem fio.

Em situações de desastre envolvendo desabamentos, pode-se, por exemplo, utilizar robôs controlados a distância, capazes de transmitir imagens capturadas sob os escombros para um computador com maior capacidade de processamento. O robô utiliza para isso um ponto de acesso, e forma assim uma rede sem fio infra-estruturada.

Por fim, nos ambientes hospitalares - caracterizados por uma constante movimentação de profissionais de saúde - é preciso que as informações sejam disponibilizadas rápida e eficientemente. Para isso, uma aplicação que utilize uma rede sem fio infra-estruturada, em que cada médico ou enfermeira faça uso de um dispositivo móvel, pode ser a solução para: a interação entre as equipes espalhadas pelo hospital e entre equipes de turnos diferentes; a disponibilização de resultados de exames; e, a transmissão de imagens de dispositivos que monitoram os pacientes para um médico ou enfermeira, onde quer que eles estejam.

1.4 Alguns Problemas nas Redes Sem Fio

Determinados problemas das redes sem fio simplesmente não existem em LANs. Por isso, as aplicações naqueles ambientes devem ser capazes de se adaptar aos diversos contratemplos que podem ocorrer durante a transmissão.

A capacidade de transmissão pode sofrer uma queda, ou até mesmo ser interrompida

por um período, devido a possíveis interferências ou simplesmente à atuação de um mecanismo próprio das redes sem fio. Esse mecanismo, ao detectar uma queda na qualidade da conexão, com qualquer dispositivo móvel, reduz a capacidade de vazão naquele nó da rede.

Essas variações na taxa de transmissão e as interferências ainda podem produzir uma alta variação no atraso dos pacotes, o chamado *jitter*, e uma grande perda de pacotes. Esses problemas, durante transmissões de dados multimídias, causam distúrbios graves na reprodução dos dados, como pulos ou congelamentos na mídia. No entanto, esses efeitos podem ser amortizados com o uso de *buffers* de memória.

Outra característica dessas redes são as desconexões abruptas, que podem chegar à ordem de segundos e, por isso, prejudicar o recebimento dos pacotes. Para evitar essas desconexões, podem ser realizadas migrações entre pontos de acessos os chamados *handoffs*. Essas migrações, representadas na Figura 1.2, devem ocorrer sem a interferência do usuário e antes que aconteça uma queda acentuada na qualidade da conexão.

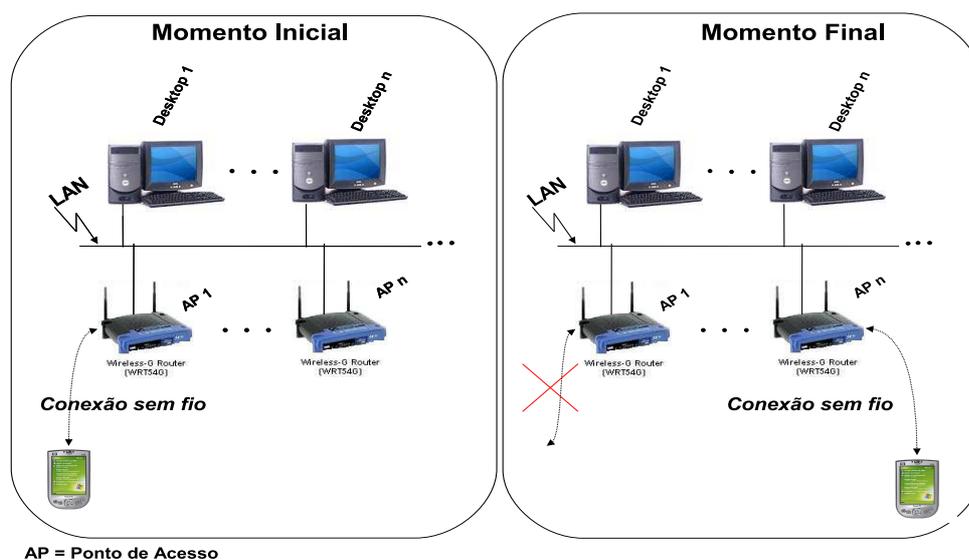


Figura 1.2: Migração entre pontos de acesso.

1.5 Objetivo

Na tentativa de superar alguns problemas intrínsecos às redes sem fio, e motivado pelas características dos ambientes hospitalares, este projeto apresenta um aplicativo para a transmissão de *streams* multimídia em redes infra-estruturadas.

Além disso, a aplicação busca adaptar-se às constantes mudanças na rede. Essas

adaptações são realizadas com o uso de *buffers*, para reduzir os efeitos no atraso da entrega dos pacotes, e com uma política de migração. Tal política é responsável por determinar o melhor momento e o melhor ponto de acesso para realizar uma eventual migração de forma transparente ao usuário, minimizando os efeitos de uma desconexão repentina.

Diante da escassez de implementações e com as dificuldades inerentes das redes sem fio, como as inúmeras interferências, este trabalho tem como principal objetivo o desenvolvimento de um aplicativo em um ambiente real, possibilitando testar as três políticas de migração propostas. A opção por não utilizar um ambiente de simulação para validar os resultados implica que alguns parâmetros do aplicativo, que visam fornecer qualidade de serviço, sejam conhecidos apenas após diversos testes.

1.6 Roteiro

Neste primeiro capítulo, foram apresentados os conceitos de rede sem fio, as configurações *ad hoc* e infra-estruturadas, alguns problemas intrínsecos desses ambientes e alguns exemplos de aplicações, além do objetivo deste trabalho.

No segundo capítulo, descrevem-se o conceito de qualidade de serviço e alguns requisitos de qualidade necessários em redes sem fio infra-estruturadas.

No terceiro capítulo, são apresentados alguns trabalhos relacionados que ajudam a perceber a diversidade dos problemas existentes nos ambientes sem fio.

No quarto capítulo, são detalhados os mecanismos de adaptação deste experimento, como as políticas de migração propostas e o gerenciamento de *buffers*.

No quinto capítulo, o sistema desenvolvido será descrito, além de algumas peculiaridades do código e das ferramentas e bibliotecas utilizadas.

No sexto capítulo, são apresentados os cenários e as métricas utilizadas nos testes, juntamente com os resultados alcançados.

Por fim, no sétimo capítulo, encontram-se as considerações finais de todo o projeto, além de sugestões para trabalhos futuros que poderão prover mais qualidade ao sistema.

No Apêndice A, está registrada toda a descrição técnica do sistema, seus diagramas de classe, e de eventos e ainda as estruturas de todas as mensagens enviadas e recebidas pela aplicação.

Capítulo 2

Qualidade de Serviço

2.1 Introdução

Atualmente, o avanço da tecnologia de aparelhos portáteis como *Laptops*, *Palm-Tops* e celulares fez com que o uso desses dispositivos móveis crescesse entre a população, aumentando também o número de aplicações desenvolvidas especialmente para computação sem fio. Os mais diversos segmentos profissionais adotaram a tecnologia das redes sem fio para facilitar ou agilizar seus negócios, utilizando principalmente aplicações multimídia. Com isso, muitas pesquisas na área de transferência de dados em redes sem fio, principalmente áudio e vídeo, estão buscando novas soluções para prover qualidade a esse tipo de aplicação.

Qualidade de Serviço (QoS) é o assunto que será tratado neste capítulo, com o objetivo de apresentar sua definição em linhas gerais, indicar os requisitos mais importantes e situar QoS em sistemas multimídia que se comunicam através de redes sem fio. Muitos trabalhos discutem esse assunto, desenvolvendo novas soluções ou apenas apresentando pesquisas sobre QoS (e.g., [Chalmers e Sloman 1999], [Aurrecoechea et al. 1998]). A próxima seção irá detalhar o que é Qualidade de Serviço e quais são os requisitos que devem ser especificados pelas aplicações. Em seguida, serão apresentados trabalhos realizados nessa área.

2.2 O que é Qualidade de Serviço

Aplicações multimídia, assim como quaisquer outras que trabalhem com fluxos contínuos, necessitam que haja um cuidadoso gerenciamento dos recursos do sistema para que se possa prover Qualidade de Serviço ao usuário. Prover qualidade de serviço significa

conhecer quais são os pontos importantes que a aplicação considera que devam ser observados ao longo da execução a fim de se manter um nível de serviço adequado ao usuário final. Para que a transferência de fluxo contínuo, especialmente multimídia, seja confiável, é essencial que os requisitos de QoS sejam configuráveis e preserváveis, incluindo, além dos dispositivos finais, também os componentes da rede.

Para desenvolver uma camada de suporte a QoS é preciso seguir alguns princípios, para que os serviços oferecidos sejam adequados à aplicação. Primeiramente, é importante que haja transparência nos mecanismos de gerenciamento de QoS para a aplicação, o que evita a execução de mecanismos complexos por parte da aplicação. O trabalho de especificação e monitoramento dos requisitos de QoS deve ser de responsabilidade da camada de suporte. Adicionalmente, esta camada deve estar presente não somente nas extremidades da transmissão, mas também ao longo do caminho em que os dados trafegam, o que permite prover QoS fim-a-fim.

A arquitetura de QoS deve ser desenvolvida de forma que haja uma divisão entre suas principais atividades, tais como transferência da mídia, dados de controle e gerenciamento dos requisitos. Essa divisão propicia ao sistema melhor desempenho, bem como a definição de uma variedade de regras que servirão de guia para as diferentes funcionalidades e para a comunicação entre essas atividades.

Após seguir esses princípios, que devem governar a construção de um sistema, é necessário determinar a especificação de QoS, para definir os requisitos e as políticas de gerenciamento de qualidade de serviço que a aplicação requer. Normalmente, essa especificação apresenta diferenças em cada camada do sistema e é usada para configurar e manter os mecanismos de QoS, tanto nos dispositivos finais como na rede. Um ponto importante a ser ressaltado é que, em uma especificação de QoS, apenas se descrevem quais são os requisitos solicitados pela aplicação, e não como serão os mecanismos para alcançar essas solicitações.

Inicialmente, é preciso especificar quais são os requisitos necessários para o desempenho de um fluxo. Para a comunicação multimídia, é muito importante que o sistema possa garantir altas taxas de vazão, baixo *jitter* e baixa perda de pacotes. Essas métricas de performance devem ser definidas pela aplicação e variar de acordo com o seu tipo. Dessa forma, a especificação de desempenho possibilita ao usuário expressar quais são os requisitos importantes para manter a qualidade e a especificação do nível de serviço, e permite que esses requisitos sejam definidos de forma quantitativa. Assim, a aplicação é capaz de distinguir graus de comprometimento de cada recurso solicitado e deve pagar

um custo agregado de acordo com o nível de serviço atribuído a cada recurso. Se não fosse assim, qualquer usuário iria especificar a melhor qualidade para todos os serviços.

Como será visto mais detalhadamente à frente, para prover QoS em redes sem fio, é necessário incluir mecanismos de adaptação que possam ajudar a aplicação a se adequar às diferentes condições da rede, geradas pelas constantes mudanças do ambiente. Por isso, é preciso também especificar políticas de gerenciamento que irão capturar o grau de adaptação de QoS tolerado por um fluxo e as medidas a serem tomadas no caso de violação do QoS especificado.

Definir Qualidade de Serviço significa definir características intrínsecas de um sistema, que afetem visivelmente a qualidade dos resultados. Tomando-se como exemplo os dados multimídia, essas características incluem a qualidade da imagem e a velocidade de resposta do sistema. Pode-se dividi-las em dois grupos: aquelas que se baseiam na tecnologia existente no sistema e aquelas definidas pelo usuário.

Exemplos de características fundamentadas na tecnologia disponível no sistema são atraso, tempo de resposta de uma requisição, *jitter*, capacidade de largura de banda e parâmetros de confiabilidade, como perda ou corrupção de dados. Já as características de QoS definidas pelo usuário podem ser priorização para determinados tipos de dados, qualidade perceptível da mídia - como detalhes da imagem, taxa de reprodução e sincronização de áudio e vídeo - custo do serviço e segurança para os dados, como mecanismos de autenticação.

Para que essas especificações sejam efetivamente utilizadas num sistema, é necessário incluir mecanismos de gerenciamento de QoS. Esse gerenciamento pode ser definido como uma necessidade de supervisão e controle das especificações para garantir que as propriedades desejadas de qualidade de serviço sejam obtidas e mantidas, tanto para fluxos de dados contínuos quanto para fluxos discretos.

Os autores de [Chalmers e Sloman 1999] dividem as funções de gerenciamento em dois grupos: primeiramente, aquelas funções que são estáticas, ligadas às propriedades ou requisitos que se mantêm constantes durante alguma atividade; em segundo lugar, está o grupo que define as funções dinâmicas, responsáveis por responder aos aspectos que mudam constantemente no ambiente.

As funções estáticas são especificação, negociação e controle de admissão. A especificação é a definição dos requisitos de QoS e seus valores. Como já descrito anteriormente, esse passo é de primordial importância, pois é a partir da definição dos requisitos que serão

desenvolvidos os mecanismos de controle e gerenciamento de QoS. Como esses requisitos têm níveis e custos, é necessária uma negociação entre a aplicação e os recursos da rede. Após essa negociação, é possível saber se o fluxo será admitido com as características desejadas, de acordo com a disponibilidade do ambiente. Os autores ainda sugerem que mecanismos de reserva de recursos podem ajudar a garantir qualidade para a aplicação.

Já as funções dinâmicas são o monitoramento, que mede periodicamente os requisitos de QoS escolhidos; a manutenção, que modifica os parâmetros do sistema para manter o QoS fornecido; a renegociação do contrato, que é requerida quando a função de manutenção não consegue alcançar os parâmetros especificados pelo contrato; e a adaptação, ou seja, a necessidade que a aplicação tem de se adaptar às novas condições do sistema e, conseqüentemente, aos novos valores para os parâmetros de QoS.

As funções de especificação, negociação, controle de admissão e reserva de recursos são os mecanismos de fornecimento de QoS. O monitoramento, a manutenção e a adaptação são aqueles mecanismos que gerenciam os requisitos de QoS especificados pela aplicação. Os mecanismos de controle definidos por [Aurrecoechea et al. 1998] são o policiamento de QoS, que verifica e assegura o cumprimento do contrato de QoS negociado pelo sistema e pela rede; e a sincronização dos fluxos, que envolve o agrupamento da informação temporal de cada *stream* com seus respectivos dados.

Esses são apenas alguns pontos importantes sobre a determinação de Qualidade de Serviço num sistema. Quando estamos lidando com um ambiente sem fio, algumas outras questões devem ser levadas em consideração. A próxima seção irá apresentar trabalhos que têm como objetivo fornecer QoS para aplicações em redes sem fio e destacar quais os pontos mais importantes levados em consideração por cada artigo.

2.3 Requisitos de QoS em Redes Infra-estruturadas

Nas redes sem fio infra-estruturadas, os principais requisitos a se comentar são:

- **Largura de Banda** - É a capacidade total de transferência da rede (em Mbps) em um dado momento. Nesse tipo de rede, é necessário implementar mecanismos que garantam uma estimativa razoável da largura de banda disponível, já que a interface de rede não notifica a aplicação sobre a capacidade atual do enlace. A estimativa desse valor pode prevenir uma perda excessiva de pacotes, por exemplo;
- **Jitter** - É a variação do atraso fim-a-fim na entrega dos pacotes. Na transmissão

de fluxo multimídia esse problema deve ser combatido, porque pode prejudicar a qualidade de reprodução do áudio e/ou vídeo. A estimativa da largura de banda, juntamente com a utilização de *buffers*, pelo receptor e pelo servidor, ajudam a minimizar ou eliminar esse problema;

- **Perda de pacotes** - Pode ser gerada por congestionamento na rede, por colisões, pelo número excessivo de retransmissões na camada MAC ou por interferências no meio. Diferentemente do que acontece com o *jitter*, as conseqüências da perda de pacotes não são *glitches*, mas congelamento de imagem e/ou som;
- **Qualidade do sinal** - A potência do sinal é medida em decibéis, enquanto sua qualidade, em intervalos de valores, que variam de acordo com o fabricante dos NICs (*Network Internet Cards*). A qualidade do sinal é uma métrica importante para a realização de adaptações nas aplicações, pois o sinal sofre um desvanecimento de acordo com a distância em relação ao ponto de acesso.

Diversos trabalhos vêm estudando adaptações, com o objetivo de superar problemas inerentes às redes sem fio. A seguir, serão apresentados algumas dessas pesquisas. A equipe Microsoft, através do projeto *qWave*, apresenta uma definição para QoS e alguns parâmetros importantes de serviço da rede para a transmissão de *streamings* de vídeo e áudio. Indra Laksono propõe o emprego concomitante da prioridade de fluxo e de técnicas de uso de *buffers*. Amit Mahajan et al. sugerem que o usuário estabeleça uma prioridade entre os fluxos de áudio e vídeo. Arlindo da Conceição e Fábio Kon inferem o valor da largura de banda, assim como Pengpeng Ni, porém por meio de métodos distintos. Por fim, Arlindo da Conceição propõe adaptações com base na qualidade do sinal de cada ponto de acesso.

A experiência relatada em [Microsoft 2004], chamada de *qWave*, traz a seguinte definição de QoS: "QoS se refere a mecanismos usados para fornecer um nível desejado de serviço da rede para uma aplicação sobre uma rede IP". A equipe ainda salienta que os parâmetros mais importantes de serviço da rede devem ser largura de banda, *jitter* e perda de pacotes. Esse trabalho apresenta como opções para se alcançar o nível de QoS desejado a reserva de banda na rede ou o fornecimento de um serviço preferencial para determinados tráfegos selecionados, o que implica a concessão de um serviço de melhor esforço para outros tipos de tráfego.

Com relação ao tráfego de *streaming* de áudio e vídeo em redes sem fio, a equipe da Microsoft apresenta como maiores problemas aqueles que sempre são citados em traba-

lhos com redes *wireless*: limitação de largura de banda, grande variação do atraso, alcance limitado e instabilidade da banda devido a obstruções, além de interferências e problemas do meio físico. Um ponto interessante reforçado nesse documento é que aplicações desenvolvidas para redes sem fio, principalmente as que trabalham com áudio e vídeo, devem ser adaptáveis. Se a rede sofrer alguma degradação, a aplicação deve invocar métodos de adaptação sem que o impacto para o usuário seja significativo; exemplo de um desses métodos é a adaptação para uma qualidade inferior de vídeo.

Em relação à largura de banda, [Laksono 2004] destaca a insuficiência das técnicas convencionais de uso de *buffers* e de prioridade de fluxo quando aplicadas isoladamente nas redes sem fio. Nesse ambiente, há perdas súbitas e severas na largura de banda, o que pode acarretar perda de pacotes. Por sua vez, a técnica tradicional de bufferização sofre também com essa instabilidade, o que leva à necessidade de *buffers* maiores para fornecer QoS no pior cenário. A solução é trabalhar com as duas técnicas simultaneamente: se há prioridade de fluxo, a transmissão não será prejudicada e não será necessário um *buffer* tão grande, mesmo com a queda na taxa de transferência. Ou seja, se as duas técnicas aplicadas separadamente não ajudam muito na transmissão multimídia via *wireless*, quando empregadas em conjunto, podem resultar em melhor qualidade.

[Mahajan et al. 2002] relacionam largura de banda e perda de pacotes, embora não tratem diretamente do primeiro item. Os autores sugerem que a aplicação forneça uma interface amigável na qual o usuário escolha o fluxo prioritário na transmissão - o áudio ou o vídeo. A partir dessa escolha, ao detectar a queda da largura de banda, a aplicação interrompe o fluxo da mídia preterida. É importante ressaltar que a aplicação não busca inferir o valor da largura de banda; na verdade, ao detectar o recebimento de mais pacotes da informação preterida, em detrimento da preferida, ela realiza uma adaptação que remodela o fluxo de acordo com as preferências do usuário.

Já [da Conceição e Kon 2006b] preocupam-se em inferir o valor da largura de banda. Uma vez que o *hardware* não fornece a taxa de transmissão, é preciso inferir essa taxa para realizar adaptações na aplicação. Uma das formas de fazer essa inferência é a técnica de pares de pacotes. O artigo utiliza métodos heurísticos para encontrar patamares na taxa de transmissão, isto é, tenta determinar um valor de corte entre a capacidade mínima e a capacidade máxima da banda. Para isso, aplica-se a técnica de desvio padrão sobre o tempo de recebimento entre os pacotes, e o resultado é utilizado na estimativa da taxa de transmissão. Dessa forma, o método analisa o tempo de recebimento entre os pacotes e estima a taxa de transmissão. O artigo busca também inferir a capacidade máxima da

rede através de um método simples que utiliza o tamanho do pacote enviado e o tamanho dos pacotes recebidos.

Em [Pengpeng 2003], a estimativa da largura de banda é realizada também com o uso do mecanismo de pares de pacotes. A largura de banda é calculada a partir da razão do tamanho dos pacotes pela dispersão, que é a diferença entre o tempo de chegada do segundo em relação ao primeiro pacote. Nesse trabalho, a técnica da média exponencial é utilizada para examinar e medir uma seqüência de valores de dispersão e largura de banda ao longo do tempo.

Nas redes sem fio infra-estruturadas a qualidade do sinal é um requisito fundamental para um possível processo de adaptação. Devido ao mecanismo *Automatic Repeat reQuest* (ARQ) das redes sem fio IEEE 802.11, utilizado para a retransmissão de pacotes perdidos, a degradação na qualidade do sinal muitas vezes só será percebida quando a qualidade da conexão já estiver um tanto deteriorada.

Em [da Conceição e Kon 2006a], a qualidade do sinal do ponto de acesso utilizado é monitorada pelo cliente, que define diferentes faixas de valores. Cada faixa pode corresponder a um nível de operação e, no momento que ocorre uma mudança nesse nível, o cliente comunica ao servidor. Em aplicações que utilizam fluxos multimídia, uma adaptação possível pode ser a redução da taxa de transferência, por parte do servidor, no momento em que ele identifica uma queda no nível de operação do cliente. Podem ser realizadas também outras formas de adaptação, como a redução na qualidade dos dados multimídia e a migração de ponto de acesso.

Uma das formas de realizar a redução na qualidade dos dados multimídia está descrita em [Mahajan et al. 2002]. Nesse caso, o servidor armazena diversas versões da informação multimídia com qualidades distintas. Ao detectar a diminuição da capacidade do enlace, a aplicação reduz a qualidade dos dados e passa a transmitir uma versão com qualidade inferior, diminuindo assim a perda de pacotes.

Uma outra opção de adaptação são as migrações, os chamados *handoffs*. A mudança de ponto de acesso pode proporcionar à aplicação uma melhoria considerável na qualidade da conexão. Sendo assim, as aplicações nas redes sem fio devem fornecer mecanismos para monitorar e detectar o melhor ponto de acesso e o melhor momento para realizar a migração, de forma automática e transparente aos usuários.

As migrações poderão ocasionar curtos períodos de desconexão para as aplicações, na ordem de segundos. Por isso, tais procedimentos devem ocorrer somente em casos extre-

mos, e seus efeitos devem ser reduzidos através do uso eficiente de *buffers*. Alguns trabalhos como [Ramani e Savage 2005], [Tan et al. 1999], [Dutta et al. 2003] e [Pack e Choi 2002] buscam reduzir este período de desconexão.

Mesmo com períodos de desconexão, a migração pode ser, em determinados momentos, a única adaptação capaz de recuperar a qualidade perdida da conexão e a qualidade de apresentação da mídia. Sabendo disso, na tentativa de oferecer uma melhor qualidade de serviço às aplicações de transferência multimídia em redes infra-estruturadas, o presente trabalho apresenta políticas desenvolvidas com o objetivo de determinar o melhor momento e o melhor ponto de acesso para a operação de *handoff*. Mais detalhes sobre as políticas desenvolvidas serão apresentados no Capítulo 4.

2.4 Conclusão

Com os avanços da tecnologia de redes sem fio e de aparelhos portáteis, surgem novos desafios para garantir a qualidade de serviço em ambientes que podem proporcionar mobilidade. Os problemas característicos desses ambientes podem ser superados, ou reduzidos, com a monitoração periódica de alguns requisitos, como largura de banda, *jitter*, perda de pacotes e qualidade do sinal.

Alguns problemas intrínsecos às redes sem fio necessitam ser considerados, como interferências e perdas súbitas na qualidade da conexão. Diante disso, as aplicações devem ser adaptáveis às constantes mudanças no ambiente, e tais adaptações devem ocorrer de forma automática e transparente ao usuário. A adaptação deve considerar o objetivo, o tipo do fluxo de dados e o nível de qualidade requisitado pela aplicação.

Para fornecer suporte às adaptações, ou até mesmo realizá-las, com base no nível de qualidade de cada aplicação, um *middleware* pode ser desenvolvido como uma camada entre a aplicação e a rede sem fio, oferecendo diferentes serviços como, por exemplo, monitoração das potências dos sinais dos pontos de acesso.

Neste capítulo foram apresentados alguns trabalhos que buscam oferecer formas de adaptação às aplicações em ambientes sem fio. Utilizando os principais requisitos das redes sem fio, algumas propostas de adaptação, com as suas limitações, produzem melhoras significativas e garantem um maior nível de qualidade de serviço.

Por fim, a migração entre pontos de acesso nas redes infra-estruturadas pode ser, muitas vezes, não só importante como a única adaptação possível. Nesse caso, as aplicações

devem ser capazes de inferir o melhor momento e o melhor ponto de acesso para realizar a migração de forma transparente ao usuário. Utilizando o requisito da qualidade do sinal, este trabalho apresenta a implementação de um protocolo de migração sobre redes infra-estruturadas.

Com o objetivo de apresentar o ambiente para o qual a aplicação e a técnica de adaptação foram desenvolvidas, o próximo capítulo apresenta maiores detalhes sobre os trabalhos relacionados nas áreas de redes sem fio, transferência multimídia e ambiente hospitalar.

Capítulo 3

Trabalhos Relacionados

3.1 Introdução

Na tentativa de oferecer melhor qualidade de serviços em ambientes sem fio, alguns trabalhos buscam amenizar problemas intrínsecos a esse tipo de rede. Tais problemas podem ser a alta variabilidade na qualidade da conexão (devido a interferências e compartilhamento do meio), períodos de desconexão que causam perdas significativas de pacotes e altas variações no atraso das mensagens ou, ainda, a grande variação na taxa de transferência ocasionada por mecanismos próprios das redes sem fio.

A possibilidade da transferência de dados multimídia sobre redes sem fio, conquistada com o avanço da tecnologia, permite o desenvolvimento de diversos tipos de aplicações, como a videoconferência. Contudo, esse tipo de aplicação necessita de qualidade fim-a-fim, o que implica em um pequeno atraso na entrega das mensagens, uma baixa perda de pacotes e uma alta largura de banda, o que muitas vezes não é possível nos ambientes sem fio. Sendo assim, alguns trabalhos apresentam mecanismos de adaptação que levam em conta as características das *streams* de áudio e vídeo.

Com a transferência multimídia sobre redes *wireless*, sistemas estão sendo desenvolvidos para fins militares, de resgate, turismo, segurança, ambientes hospitalares, entre outros. Nesse último caso, a intensa movimentação nesses ambientes e a grande necessidade de acesso às informações, a todo momento e em todo lugar, são características suficientes para a implantação de um sistema que permita uma comunicação rápida e eficiente, que garanta a mobilidade e diminua os erros.

Neste capítulo, serão detalhados os principais problemas intrínsecos às redes sem fio e às transmissões multimídia, por meio da descrição de alguns trabalhos nessas áreas. Por

fim, serão ressaltadas algumas características e necessidades dos ambientes hospitalares, uma vez que o presente trabalho busca desenvolver um mecanismo que forneça qualidade nas transmissões multimídia sobre redes sem fio, motivado por esses ambientes.

3.2 Redes Sem Fio

O aumento da capacidade na transferência efetiva de dados através de redes sem fio permite o uso de aplicações multimídia neste ambiente, o que fornece aos usuários maior conforto e mobilidade. Porém, tais aplicações necessitam de uma maior qualidade na conexão e não aceitam um nível muito alto de perda de pacotes e um *jitter* muito grande na recepção de informações. Em busca de superar muitos problemas existentes nas redes sem fio, vários trabalhos oferecem sugestões de adaptações, nas camadas de aplicação e nas camadas inferiores, a fim de fornecer uma maior qualidade de serviço.

Pela necessidade de alcançar melhor qualidade na transferência de dados multimídia, muitos pesquisadores e desenvolvedores estão buscando superar os novos desafios trazidos pelas redes IEEE 802.11. Tais desafios, como a necessidade de migrações entre pontos de acesso (*handoffs*) e a alta variabilidade da qualidade da conexão, fazem com que as aplicações desenvolvidas sobre tais redes necessitem adaptar-se a essas variações de forma automática e transparente ao usuário.

Antes mesmo de discutir problemas intrínsecos às redes sem fio, como os chamados *handoffs*, deve-se destacar, como em [da Conceição e Kon 2006a], que os protocolos TCP e UDP não foram projetados levando em consideração as peculiaridades deste tipo de rede. Devido à incapacidade desses protocolos de lidarem com algumas características próprias das redes sem fio, uma das alternativas é a implementação de técnicas adaptativas na camada de aplicação.

Uma realidade das redes sem fio é o fato de outros dispositivos eletrônicos, que vão desde fornos microondas até interfaces *Bluetooth*, usarem a mesma faixa de frequências de rádio que a rede IEEE 802.11, o que ocasiona interferências. Outros fatores, como a sobreposição de objetos, a umidade do ar e até mesmo a orientação e as alturas das antenas, podem interferir na qualidade da conexão nesse tipo de rede, causando o desvanecimento do sinal. Ainda, a mobilidade e a distância ao ponto de acesso são responsáveis pela deterioração do sinal, o que pode causar uma maior perda de pacotes e uma diminuição da taxa efetiva de transmissão. Assim, a qualidade do sinal pode variar substancialmente e em curtos intervalos de tempo, o que dificulta o desenvolvimento de algoritmos adaptativos.

Um outro ponto importante, descrito por alguns autores, está na existência de mecanismos como *Automatic Repeat reQuest* (ARQ), *Power Saving Mode* (PSM) e *Request to Send / Clear to Send* (RTS / CTS) que podem dificultar o desenvolvimento de aplicações adaptativas sobre as redes IEEE 802.11.

O mecanismo ARQ realiza a confirmação da entrega de cada pacote através do envio, por parte do cliente, de um outro pacote indicando o recebimento. Caso o pacote de confirmação não seja recebido em um determinado intervalo de tempo, o mecanismo reenvia o mesmo quadro. Isso se repete um número limitado de vezes ou até que a entrega seja confirmada. Assim, pacotes podem ser entregues fora da ordem e, ainda, pode ocorrer uma percepção errônea na taxa de perda de pacotes. Por isso, na prática, a qualidade da conexão já está bastante comprometida quando as perdas de pacotes começam a ser percebidas nas transmissões sobre redes sem fio.

O mecanismo ARC é responsável pela redução da velocidade de transmissão. No caso das redes infra-estruturadas, a taxa de transmissão será reduzida em um ponto de acesso todas as vezes que for identificada a degradação do sinal em função da mobilidade, mesmo que seja apenas de um cliente. Ou seja, conforme uma unidade móvel se distancia do ponto de acesso, esse ponto reduz a taxa de envio, o que produz má distribuição da capacidade de transmissão a todas as unidades móveis conectadas ao ponto.

O mecanismo PSM pode inserir alta variação na latência e produzir períodos de inatividade, que podem ser interpretados pelo protocolo TCP como congestionamento.

Outro desafio em redes sem fio, descrito por [da Conceição e Kon 2006a], é a diversidade de comportamentos entre dispositivos Wi-Fi, ocasionada pela não especificação de alguns pontos pelo padrão IEEE 802.11. A falta de padronização entre diferentes fabricantes, em pontos que exercem forte influência no comportamento das transmissões sobre redes sem fio, dificulta a criação de algoritmos genéricos adaptáveis. Para um melhor entendimento, podemos citar a falta de um padrão em dois importantes pontos: no mecanismo de ARC e nos níveis de qualidade do sinal. Diferentes fabricantes implementam diversos algoritmos para o mecanismo de ARC, o que ocasiona diferentes taxas de perda de pacotes, latência, *jitter* entre outros. A falta de um padrão na definição dos níveis de qualidade do sinal permite que fabricantes como Atheros, Symbol e Cisco produzam NICs que traduzam a qualidade do sinal em uma faixa entre 0 a 60, 0 a 31 e 0 a 100 respectivamente. Como se somente isso não fosse o suficiente, ainda o valor médio da qualidade do sinal nesses NICs, que são, 30, 16 e 50, correspondem a diferentes valores da potência do sinal (-65dBm, -70dBm, -58dBm).

Daí conclui-se que as aplicações adaptativas desenvolvidas para redes sem fio, que utilizam equipamentos de diversos fabricantes, necessitam de componentes especiais capazes de superar a falta de padronização.

Na tentativa de adaptar a transmissão de fluxos contínuos às condições da conexão, com base na qualidade do sinal, um algoritmo adaptativo no nível da aplicação é apresentado em [da Conceição e Kon 2006a]. Observando-se a potência do sinal e a taxa de recepção na unidade móvel, pode-se perceber que afastar a unidade do ponto de acesso interfere diretamente na taxa de recepção. Dessa forma, a aplicação determina, dinamicamente, três limiares de qualidade do sinal, usados para definir quatro faixas para adaptação. Com isso, o cliente identifica, em intervalos fixos de tempo, seu nível de adaptação e envia uma mensagem ao servidor. Ao receber a mensagem, o servidor pode implementar políticas de adaptação diferentes para cada faixa. Contudo, o envio periódico de mensagens que indicam o nível de adaptação do cliente faz aumentar o congestionamento na rede.

A transferência de dados multimídia sobre as redes IEEE 802.11, como apresentado anteriormente, sofre com algumas características desse tipo de rede, como latência e variabilidade da qualidade do enlace. Ainda, devido ao grande volume de informações transmitidas neste tipo de aplicação, a largura de banda deve ser suficiente para não gerar grandes retardos e *jitter* na entrega dos pacotes.

Técnicas tradicionais de prioridade de banda e de armazenamento em *buffers*, que fornecem qualidade de serviço, produzem o resultado esperado apenas quando são utilizadas em conjunto sobre as redes sem fio, como descrito no capítulo 2. As reduções súbitas e severas na taxa real de transferência podem acarretar uma visualização sem qualidade ou até mesmo pequenos congelamentos da imagem. Por isso, o trabalho apresentado em [Laksono 2004] combina as duas técnicas, ou seja, adaptação da taxa de transferência de acordo com a largura de banda disponível, com o armazenamento em *buffer*.

Com o objetivo de oferecer qualidade de serviço, [Mahajan et al. 2002] apresenta uma arquitetura que implementa novos conceitos para adaptação na camada de aplicação para as transferências multimídia sobre redes sem fio. Esses novos conceitos se baseiam nas preferências dos usuários fornecidas ao sistema através de interfaces amigáveis, como mencionado no Capítulo 2.

Na arquitetura proposta nesse último trabalho, a adaptação é baseada na preferência do usuário, de acordo com um percentual fornecido e a largura de banda disponível em um determinado momento. Dois módulos são definidos: o cliente, que possui mobilidade,

e um servidor de dados multimídia. Depois que o cliente estabelece uma conexão com o servidor, aquele envia, periodicamente, uma mensagem ao servidor sobre sua largura de banda disponível. O servidor armazena múltiplas cópias da *streaming* de dados codificadas em diferentes níveis de qualidade. Com base nas mensagens do cliente e no conhecimento da preferência do usuário, o servidor seleciona dinamicamente a cópia da *stream* de áudio e/ou vídeo que será transmitida. A mensagem transmitida pelo cliente contém o nível de QoS que foi determinado após uma análise da quantidade de dados recebidos. Outra vez pode-se destacar o aumento do número de mensagens na rede proporcionando um maior congestionamento.

A justificativa de introduzir uma adaptação que leve em conta a preferência do usuário, segundo [Mahajan et al. 2002], está na observação da dependência do tipo de dado a ser transmitido em cada momento. Ou seja, em uma apresentação de *slides*, o mais importante será o fluxo de vídeo, e na transferência de um anúncio jornalístico, por exemplo, muitas vezes o mais importante será o fluxo de áudio. Dessa forma, o conteúdo a ser transmitido está diretamente relacionado com o motivo de priorizar um ou outro fluxo. Assim, valores percentuais são associados aos tipos de dados que o usuário espera receber.

Na prática, a simples eliminação de um fluxo em uma transferência de áudio e vídeo não é suficiente para prover uma adaptação que forneça uma qualidade de serviço razoável. O ambiente sem fio sofre demasiadamente com alterações abruptas em sua largura de banda e em seus recursos, como mencionado anteriormente. Por isso, muitas vezes o cliente pode ficar muito tempo desconectado e precisar migrar entre pontos de acesso, e isso pode gerar perdas de pacotes e diminuir a qualidade na reprodução dos dados, o que ocasiona congelamentos e pulos no áudio e/ou vídeo.

Nas redes IEEE 802.11, o mecanismo de ARC não produz nenhuma notificação para a camada de aplicação. Para suprir a falta dessa notificação, em [da Conceição e Kon 2006b] é proposto o uso da técnica de pares de pacotes para tentar inferir a mudança na taxa de transmissão e a vazão máxima das redes IEEE 802.11. Em [Pengpeng 2003], esta técnica também é utilizada com o objetivo de estimar a largura de banda disponível, como mencionado no Capítulo 2.

A tentativa de estimar a largura de banda e a vazão máxima disponível nas redes sem fio utilizando a técnica de pares de pacotes requer certos cuidados e apresenta algumas limitações. Primeiramente, deve-se se preocupar em enviar um pacote imediatamente após o outro, sem realizar qualquer tipo de operação neste intervalo de tempo, o que pode implicar uma medição errada. Outra precaução está na resolução do relógio do sistema, que

deve ser da ordem de nanosegundos; caso contrário, a precisão da medição entre os tempos de chegada dos dois pacotes fica reduzida, como citado em [da Conceição e Kon 2006b]. Por fim, é preciso preocupar-se com o tamanho dos pacotes uma vez que eles não podem ser muito grandes (pode ser, por exemplo, de 1024 bytes), o que geraria fragmentações, e também não devem ser muito pequenos. Caso o pacote tenha um tamanho reduzido, a perda de pacotes, quadros e mensagens também será reduzida, o que diminui a atuação do mecanismo de ARC. Não atuando o mecanismo, não haverá alteração na taxa de transmissão, o que dificulta a inferência dos valores através do algoritmo proposto.

Outro problema intrínseco às redes sem fio são os chamados *handoffs*. Eles acarretam períodos de desconexão que podem durar alguns segundos. Buscando diminuir este tempo de desconexão e realizar a migração em um menor intervalo de tempo, trabalhos como [Ramani e Savage 2005], [Tan et al. 1999], [Dutta et al. 2003] e [Pack e Choi 2002] apresentam diferentes soluções.

A utilização das redes sem fio será cada vez mais comum, devido ao conforto e facilidade de mobilidade que proporcionam. Contudo, essa realidade traz consigo problemas novos e que necessitam ser resolvidos a fim de oferecer qualidade de serviço fim-a-fim às aplicações deste ambiente. Muitos dos problemas intrínsecos às redes *wireless* podem ser resolvidos com a migração para um novo ponto de acesso, onde se pode encontrar uma conexão que ofereça melhores condições para a transferência dos dados. Devido a isso, o presente trabalho apresenta um novo mecanismo de adaptação que determina o melhor momento e o melhor ponto de acesso, e realiza a mudança de forma automática e transparente. Este novo mecanismo de adaptação será apresentado com mais detalhes no capítulo 4.

3.3 Transmissão Multimídia em Redes Sem Fio

Nos padrões de vídeo MPEG e H.26x, por oferecer escalabilidade temporal e espacial, existe uma possibilidade de adaptação, afetando diretamente a qualidade do vídeo. Ou seja, pela existência de múltiplos quadros, como os I, P e B, diferentes qualidades na visualização do vídeo podem ser fornecidas. Tal divisão pode permitir, facilmente, a degradação do vídeo quando a largura de banda e outros recursos se tornarem escassos.

Esta seção fará uma breve apresentação sobre o que são os dados multimídia e mostrará um trabalho que desenvolve soluções para fornecimento de QoS em transmissões multimídia e cria adaptações direto no formato ou na codificação dos dados de vídeo.

3.3.1 Formato Multimídia

Chamamos de dados multimídia as informações de áudio e vídeo que são capturadas e codificadas em diferentes tipos de compactação. Nesta seção, serão apresentados, brevemente, os formatos dessas mídias e serão citados os padrões de codificação mais comumente utilizados.

Independentemente do formato de compressão empregado, qualquer vídeo é formado por uma seqüência de imagens, que também podem ser chamadas de quadros. A estrutura de um vídeo é composta de duas partes: a resolução espacial – que compreende a resolução geométrica (formato dos objetos que compõem a cena) mais a resolução de cor – e a resolução temporal, que define a quantidade de quadros por unidade de tempo que deverão ser reproduzidos.

Os quadros que em seqüência formam um vídeo podem ser classificados em dois tipos: aqueles que são codificados de forma independente – *Intracoded frames*, ou quadros I – e os quadros estimados. Esses últimos são ainda divididos em quadros P, ou *Predictive frames* e quadros B, *Bidirectional frames*.

Esses três tipos de quadros são intercalados e em seqüência formam um GOP ou *Group of Pictures*. Um GOP é definido como o conjunto de quadros entre dois quadros I sucessivos. Ele possui tamanho variável (entre 12 e 15 quadros), e a seqüência dos quadros normalmente é formada da seguinte maneira: I B B P B B P B B P B B. Outros detalhes podem ser encontrados em [Watkinson 2003].

Existem vários tipos de compressão de dados de vídeo. Os mais conhecidos são os das famílias MPEG e H.26x. Entre as versões dos padrões, existem pequenas diferenças.

O padrão MPEG – *Moving Picture Experts Group* – foi padronizado pela ISO e já possui algumas versões como MPEG-1, MPEG-2 e MPEG-4. A primeira é usada para armazenamento de áudio e vídeo com qualidade de VHS em CD-ROMs. Já a versão MPEG-2 obteve alguns melhoramentos e apresenta 4 níveis de resolução do vídeo: baixa (que é compatível com MPEG-1), principal (que apresenta resolução do vídeo com qualidade de estúdio), alta 1440 (presente em televisores de alta definição – HDTV) e alta (presente em HDTV de tela larga). O padrão MPEG-4 surgiu com o objetivo de abranger aplicações multimídia interativas, com taxas baixas.

O padrão H.261 é comumente usado para serviços de videotelefonia e videoconferência em redes digitais de serviços integrados (ISDN). Já o padrão H.263 é mais utilizado para aplicações com transmissão de vídeo em redes sem fio e redes telefônicas tradicionais

(PSTNs). Ele utiliza os mesmos conceitos do H.261, porém oferece algumas facilidades que fazem melhorar a qualidade. Esse último padrão define os mesmos tipos de quadros (I, P e B) e é compatível com o MPEG-2.

3.3.2 Adaptação Multimídia

O trabalho [Isovié e Fohler 2004] apresenta um mecanismo para descartar alguns quadros, I, P e/ou B do padrão MPEG e assim simplificar o processo de decodificação em ambientes com limitações de recursos, como baixa capacidade de processamento. Sendo assim, numa eventual falta de recursos disponíveis, nem todos os quadros serão decodificados. Para isso, o trabalho propõe que sejam utilizados critérios para selecionar os quadros que devem ser decodificados e os que podem ser descartados. Após a priorização dos quadros segundo esses critérios, foi proposto um algoritmo que garante ser exibido, com os recursos no momento disponíveis, o melhor *stream* MPEG possível. Como entrada, esse algoritmo recebe o *stream* MPEG, a quantidade de recursos disponíveis no sistema e os valores de importância dos quadros.

Quando acontece de haver falta de recursos para que um GOP seja decodificado e reproduzido, um ou mais quadros deverão ser descartados a fim de que não seja ultrapassado o prazo final para que a imagem seja reproduzida. Esse descarte pode ser feito de duas formas: o salto reativo, que descarta o quadro corrente, ou o salto preventivo, que percebe a falta de recursos antes que ela aconteça e já descarta alguns quadros.

Em vez de usar o salto de *frames* de forma reativa, os autores utilizam o pulo preventivo de quadros, que decide saltar uma quantidade determinada de quadros antes que o *deadline* seja perdido. O descarte preventivo utiliza alguns critérios para decidir se o quadro deve ser saltado ou não. Esses critérios devem ser definidos de forma eficiente, pois, se um quadro for descartado erroneamente ou em um momento não apropriado, isso pode causar um distúrbio perceptível na qualidade do vídeo que está sendo reproduzido.

Dessa forma, o trabalho [Isovié e Fohler 2004] define diversos critérios e apresenta um algoritmo utilizado para combinar todos os critérios e estabelecer prioridades para os quadros. Estabelecidas as prioridades de cada quadro é possível fazer, facilmente, a adaptação do vídeo. Os autores utilizam-se da análise da disponibilidade da CPU como parâmetro para identificar a falta de recurso do sistema e iniciar a adaptação. Contudo, é possível usar também largura de banda e qualidade do sinal.

3.4 Ambiente Hospitalar

Em um ambiente hospitalar é uma constante a movimentação de médicos, enfermeiras, responsáveis por laboratórios, entre outras pessoas. Esta movimentação tem diversas razões, como visitas a pacientes, localizações dos diversos recursos e documentos, como aparelhos para diagnósticos e registro médico do paciente, ou ainda, para uma simples consulta a outros médicos a fim de se concluir um diagnóstico. Esses profissionais necessitam das informações a todo tempo e em qualquer lugar. Contudo, vale a pena ressaltar que a localização do agente de saúde muitas vezes pode indicar o tipo de informação que será requisitada. Por exemplo, um médico próximo a um leito provavelmente precisa acessar o prontuário do paciente.

As diversas equipes nos hospitais são distribuídas em tempo e espaço, como citado nos trabalhos [Favela et al. 2004] e [Favela et al. 2003], e as informações de cada equipe e turno devem ser disponibilizadas para as demais. Essa comunicação pode ser realizada através de ferramentas simples, como um quadro branco ou um registro sobre as condições do paciente descritas em um livro, por exemplo. Sendo assim, uma importante tendência na comunicação interna nos hospitais pode ser o uso de sistemas que armazenem eletronicamente os registros do paciente, o que facilita o acesso às informações clínicas dele.

Diante desse ambiente dinâmico, as aplicações precisam ser desenvolvidas com alta capacidade de adaptação e gerar respostas de acordo com o contexto do usuário.

3.5 Conclusão

Com os trabalhos apresentados neste capítulo pode-se perceber a importância de adaptações que busquem garantir qualidade de serviço nas transmissões multimídia sobre redes sem fio.

Um dos trabalhos apresenta a adaptação com base na percepção do usuário, o que pode gerar um resultado positivo, uma vez que fluxos não muito importantes podem deixar de ser transmitidos quando algum recurso se tornar escasso. Em outros trabalhos, as adaptações nas *streams* multimídia pode evitar o envio de informações redundantes e até mesmo o envio de quadros desnecessários, o que é fundamental em ambientes sem fio com pouco poder de processamento. Por fim, outros autores buscam inferir a capacidade de transferência da rede para diminuir a taxa de perda e o atraso dos pacotes. Mas em

um ambiente tão instável em que podem ocorrer, até mesmo, períodos de desconexão, somente essas adaptações certamente não são suficientes.

Diante das instabilidades inerentes às redes sem fio infra-estruturadas, as aplicações devem buscar outros pontos de acesso na tentativa de encontrar melhores condições para a transferência dos dados. Essa migração deve ser rápida e ocorrer de forma transparente ao usuário.

Como a maior parte dos trabalhos, nessa área de pesquisa, utiliza de simulação para comprovar a eficiência de seus algoritmos, a utilização da técnica de migração não é valorizada. A migração é uma técnica que necessita ser implementada e testada, pois a grande diversidade de problemas envolvidos torna inviável sua simulação. A implementação e os testes devem ser realizados sobre um ambiente real que contenha diversos pontos de acesso e um, ou mais, dispositivos móveis realizando as migrações entre os roteadores.

Neste projeto optou-se por uma adaptação realizada pela aplicação cliente, porém transparente ao usuário. Assim, nenhuma mensagem sobre o nível de adaptação necessita ser enviada do cliente ao servidor, diminuindo o fluxo de dados na rede. No atual experimento, a percepção do usuário também está sendo levada em conta através de um indicador visual da qualidade do sinal do ponto de acesso ao qual o cliente está conectado. Dessa forma, o cliente pode decidir, por exemplo, continuar se afastando do ponto de acesso o que poderá provocar uma desconexão ou uma migração controlada.

Para suprir a falta de experimentos com implementações, e com foco nas características dos ambientes hospitalares, o próximo capítulo apresenta alguns mecanismos de adaptação implementados nesta dissertação. Através de algumas políticas de migração, a aplicação é capaz de determinar o melhor momento e o melhor ponto de acesso para um possível *handoff*, realizado de forma automática e transparente ao usuário. Além disso, o gerenciamento de *buffers*, realizado pela aplicação, possibilita melhora na qualidade de serviço na transmissão, recebimento e visualização de *streams* multimídia. Por fim, a indicação visual da potência do sinal do ponto de acesso conectado permite ao usuário prevenir-se contra uma desconexão iminente.

Capítulo 4

Mecanismos de Adaptação Propostos

4.1 Introdução

O desenvolvimento de aplicações sobre redes infra-estruturadas precisa se preocupar com os diversos problemas intrínsecos a esse tipo de ambiente, tais como *handoffs*, altas variações no atraso e grande perda de pacotes. Tais problemas afetam diretamente as aplicações, principalmente aquelas que realizam transferência multimídia.

Técnicas de uso de *buffers* devem ser utilizadas por aplicações multimídia para diminuir os efeitos das altas variações no atraso dos pacotes. Esses efeitos podem ser congelamentos da imagem, no caso de transmissão de vídeo, e pequenas pausas nas reproduções de áudio.

Quedas abruptas na conexão são comuns nas redes sem fio e causam períodos de desconexão, que podem chegar a ordem de segundos. Para minimizar, ou até mesmo impedir tais quedas, adaptações devem ser implementadas pelas aplicações. Uma dessas adaptações pode ser a migração entre pontos de acesso de forma automática e transparente ao usuário.

Assim, os mecanismos de adaptação implementados nesse sistema são: o gerenciamento de *buffers*, tanto no servidor quanto no cliente; a definição de políticas que buscam identificar o melhor momento para realizar uma migração entre pontos de acesso; e um indicador visual da qualidade do sinal. Esses mecanismos foram desenvolvidos com o objetivo de atender às peculiaridades das redes sem fio, conforme já detalhado em capítulos anteriores, e ainda oferecer melhor qualidade de serviço em transferências multimídia sobre redes infra-estruturadas.

4.2 Caracterização do Ambiente do Projeto

A motivação deste projeto se encontra nas aplicações para ambientes hospitalares, caracterizados pela constante movimentação de pessoas e pela grande necessidade de comunicação ágil e eficiente. Para garantir a comunicação em todo o interior de um hospital pode-se utilizar uma rede sem fio infra-estruturada com a quantidade de pontos de acesso necessária. Para simular esse ambiente foi instalada uma rede infra-estruturada com três roteadores da Linksys, modelo WRT54G, padrão 802.11b/g, configurados como pontos de acesso, fornecendo cobertura do sinal para todo o andar do Instituto de Computação da Universidade Federal Fluminense.

A constante movimentação de pessoas no local dos testes permitiu uma aproximação da realidade diária de um hospital. Os dados transmitidos através da rede sem fio foram enviados por um único servidor, um *desktop* conectado a LAN do Instituto. Assim, o cliente aqui representado por um dispositivo de pequeno porte, *Pocket PC*, se conecta com o servidor através de um dos pontos de acesso. Após estabelecida a conexão, o cliente e o servidor podem transmitir, entre si, mensagens de texto e o servidor pode enviar ao cliente *streams* de vídeo *on-line*.

Por fim, é importante ressaltar que este projeto buscou utilizar apenas dispositivos *off-the-shelf* para que não fosse necessária a alteração das configurações de *hardware* e *software* daqueles que desejassem participar da rede. Além disso, a baixa capacidade de processamento dos dispositivos móveis disponíveis limitou a transferência do vídeo em poucos quadros por segundo.

4.3 Gerenciamento de *Buffers*

Os *buffers* são um dos mecanismos de adaptação que visam a garantir qualidade de serviço na transmissão de dados em redes sem fio. Esses mecanismos atuam tanto no servidor quanto no cliente e, em ambos os casos, apresentam funções diversas.

No servidor, o *buffer* é utilizado primeiramente para controlar a taxa de envio de quadros de vídeo. No presente trabalho, a taxa de captura fornecida pela câmera disponível em nosso laboratório varia entre 1 até 15 *frames* por segundo (fps). Cada *frame*, ao ser capturado, recebe um número seqüencial e um carimbo de tempo, em milissegundos. Como não foi desenvolvido um *driver* de captura para a câmera, há a necessidade de se utilizar a taxa estabelecida pela configuração disponível, que é de no máximo 15 fps. Para simular

taxas menores o aplicativo servidor armazena no *buffer* os quadros capturados e, por ser o cliente um *pocket PC*, apenas o primeiro quadro de cada segundo é transmitido. Isso ocorre pois a taxa de visualização nesses dispositivos limita-se a 1 fps, devido à sua baixa capacidade de processamento e à inexistência de processadores dedicados somente à parte gráfica.

Outra função que justifica a existência do *buffer* no servidor é a possibilidade de enviar informações para mais de um cliente simultaneamente. Vale observar que essa conexão simultânea não está disponível na versão atual da implementação. Pode-se visualizar, então, o seguinte cenário: dois dispositivos, com capacidades de processamento diferentes, conectados ao mesmo servidor, demandam taxas de transferência distintas. Ou seja, um *notebook* e um *pocket PC* podem receber, respectivamente, 12 (ou mais) fps e 1 fps. Dessa forma, toda a mídia que será transmitida é armazenada no *buffer*, e uma tarefa é disparada de acordo com a taxa de transmissão possível para o dispositivo conectado. Essa tarefa é responsável por recuperar a informação do *buffer* e enviá-la.

Já no cliente, o uso de um *buffer* destina-se a três funções. A primeira é a redução do impacto da variação do atraso, o *jitter*, sobre a visualização da mídia. A segunda é a possibilidade de ordenar pacotes recebidos fora da seqüência. Essa desordem pode ocorrer devido ao mecanismo de retransmissão existente nos pontos de acesso. Contudo, na transferência de poucos *frames* por segundo, como é o caso do atual experimento, essa desordem na entrega dos pacotes será, naturalmente, minimizada. Por último, a terceira função, o controle da taxa de visualização, é realizada por uma tarefa, disparada de tempos em tempos, que retira um quadro do *buffer*, decodifica-o e o visualiza.

Assim, o gerenciamento de *buffers* é uma técnica que permite superar problemas: no caso do cliente, diminui-se o efeito do *jitter* e, no servidor, permite-se controlar a taxa de envio. Em ambos os casos, o resultado final pretende ser a melhora na qualidade do serviço.

4.4 Políticas de Migração

A migração entre pontos de acesso, realizada de forma transparente, representa a principal forma de adaptação proposta pela aplicação desenvolvida. Considerando-se os limites de processamento dos dispositivos portáteis e sabendo ser a migração um processo que apresenta duração da ordem de segundos, essa adaptação deve ser realizada o menor número de vezes possível.

Na tentativa de minimizar as migrações e ainda reduzir a taxa de perda durante a transmissão, foram propostas três diferentes políticas. Para dar suporte a essas políticas, a aplicação faz uma busca dos sinais de todos os pontos de acesso periodicamente (parâmetro da aplicação). Após um certo número de buscas - valor que pode ser facilmente alterado - a política é aplicada. Ao final de cada ciclo os valores dos sinais que foram armazenados são descartados. Como a existência de outros dispositivos sem fio no ambiente não está descartada, o sistema identifica apenas os sinais dos dispositivos configurados como infra-estruturados (e não *ad hoc*).

A escolha da política a ser utilizada é realizada pelo usuário através de uma interface simples, e essa política pode ser alterada a qualquer momento durante a execução da aplicação. Todas as políticas têm o objetivo de determinar o melhor ponto de acesso, mas se diferenciam na forma de fazê-lo.

A aplicação ainda oferece ao usuário, além dessas três políticas, a possibilidade de não utilizar nenhuma política de migração. Com isso, a mudança de ponto de acesso só deverá ocorrer após o reconhecimento de uma queda abrupta da conexão. Nessa circunstância, a aplicação desenvolvida é capaz de identificar a queda e realizar um processo de reconexão. Para essa reconexão de urgência, o sistema ignora a política de migração escolhida e identifica o melhor ponto de acesso, ao qual o cliente deverá se conectar, obrigatoriamente através da política chamada de Média Simples. Nas próximas subseções são apresentadas cada uma das políticas propostas.

4.4.1 Média Simples

Essa política define o melhor ponto de acesso através da média simples das três últimas buscas realizadas. Para esse fim, o sistema registra o valor zero para a qualidade do sinal de um ponto de acesso não alcançável em um determinado momento. O ponto de acesso com a melhor média é sempre escolhido como substituto, caso não seja o mesmo ao qual o cliente se encontra conectado.

4.4.2 Média Simples com Limite Inicial

A principal característica dessa política é o fato de só realizar a migração quando o valor da qualidade do sinal recebido do ponto de acesso ao qual o cliente está conectado for inferior a um valor pré-determinado. Caso a qualidade do sinal seja inferior a esse valor, o novo roteador será determinado através da média simples, de forma semelhante ao que

ocorre na política anterior. Assim, se a média das últimas três medições de algum dos outros pontos de acesso for melhor do que a do atual roteador, a migração será realizada.

4.4.3 Diferentes Níveis

Nessa política são estabelecidas diferentes faixas de valores, que determinam níveis distintos para as qualidades dos sinais capturados, como pode ser visto na Figura 4.1. Identificando-se esses patamares, a migração só ocorrerá para pontos de acesso com a qualidade do sinal pertencente a faixas melhores.

Outro ponto importante nessa política é a existência de um valor Δ , próximo ao limite inferior de cada faixa de valores, utilizado como pré-requisito para iniciar a busca por um roteador em melhores condições de operação. Essa busca consiste na verificação da possível existência de um ponto de acesso cuja média das últimas três potências do sinal pertença a uma faixa melhor do que a atual; obviamente, se o valor atual já estiver no melhor nível, nada será feito. Assim, a busca por um melhor ponto de acesso é iniciada somente quando a potência do sinal estiver entre o limite inferior da faixa de valores e esse limite acrescido de Δ . Por fim, a migração será aplicada todas as vezes que a busca for bem sucedida.

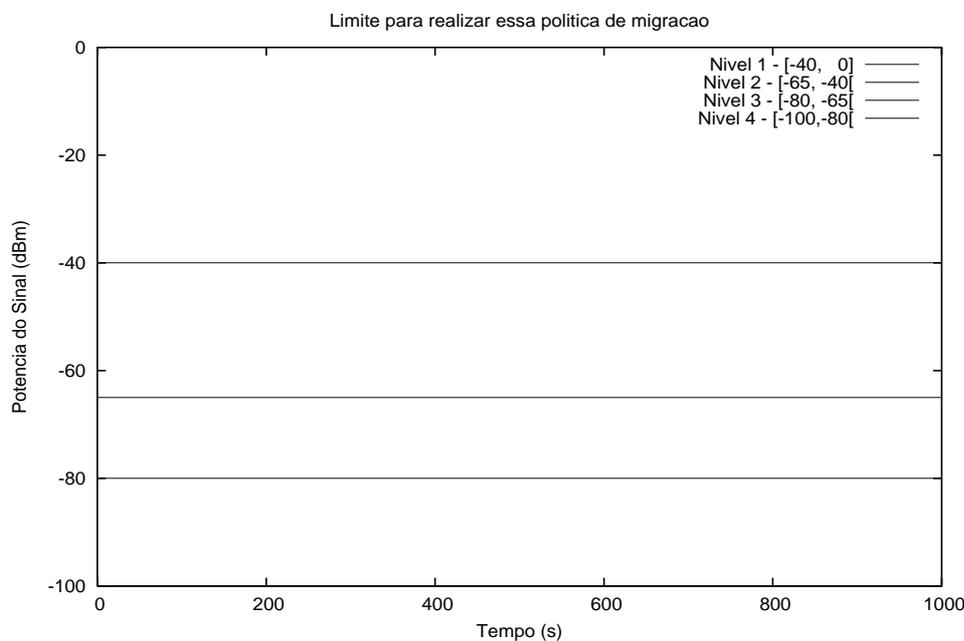


Figura 4.1: Limites de cada faixa da política de migração.

De acordo com a Figura 4.1 são definidos quatro níveis distintos. No nível 1, em que o intervalo é definido entre -40, inclusive, e 0 dBm, nada será feito, pois o cliente se encontra

na melhor faixa de valores possível. No intervalo entre -65, inclusive, e -40 dBm, a faixa 2, o cliente inicia a aplicação da política quando a potência do sinal estiver entre -65 e $-65 + \Delta$ e poderá realizar uma migração caso exista um ponto de acesso cujo a média das potência dos sinais esteja na faixa 1. Já no terceiro nível, definido entre -80, inclusive, e -65 dBm, o sistema poderá migrar tanto para um ponto de acesso na faixa 1 quanto para um que esteja na faixa 2, com prioridade para o que estiver no primeiro nível. Por fim, quando o roteador conectado entrar na quarta faixa de valores, a migração deverá ocorrer de forma urgente para o ponto de acesso que estiver na melhor faixa possível. Sendo assim, nessa faixa não existe o valor Δ e por isso, no momento que o cliente identifica que a qualidade do sinal capturado do ponto de acesso pertence a faixa 4 é iniciada, imediatamente, a busca por um roteador em melhores condições. Se ele não existir, o sistema não realizará nenhuma migração e uma desconexão é iminente, caso o usuário continue se afastando do ponto de acesso ao qual está conectado.

4.5 Indicação da Qualidade do Sinal

O último mecanismo de adaptação é um indicador da qualidade do sinal do ponto de acesso ao qual o sistema está conectado. O objetivo é alertar o usuário sobre uma baixa qualidade do sinal e, conseqüentemente, sobre a possibilidade de desconexão em um futuro próximo, caso ele continue se afastando do ponto de acesso.

Esse mecanismo consiste em indicar ao usuário três faixa da qualidade do sinal, como pode ser visto na Figura 4.2. Periodicamente, o aplicativo analisa a qualidade do sinal e atualiza a cor do indicador (o nome do ponto de acesso ao qual o cliente está conectado, no campo *Signal Strength*), como descrito a seguir:

- A cor vermelha indica que a potência do sinal está na faixa de -100 a -70 dBm, ou seja, muito baixa, e que uma possível desconexão é iminente. Nesse caso o usuário deve parar de se mover ou retornar em direção ao ponto de acesso, caso não deseje uma desconexão abrupta ou uma migração;
- A cor amarela é utilizada para alertar ao usuário sobre um possível afastamento do roteador. Ela sinaliza que a potência do sinal está na faixa dos -70 aos -40 dBm;
- A cor verde sinaliza uma melhor potência do sinal (superior a -40 dBm) determinada pela proximidade do usuário em relação ao ponto de acesso.



Figura 4.2: Indicação por cores da qualidade do sinal do ponto de acesso.

4.6 Conclusão

A instabilidade das redes sem fio requer a existência de procedimentos que realizem adaptações nas aplicações executadas nesses ambientes. Diante disso, na tentativa de fornecer uma maior qualidade de serviço na transferência multimídia sobre redes infra-estruturadas, foram propostos, neste projeto, mecanismos de adaptação como o gerenciamento de *buffers*, diferentes políticas de migração e a indicação visual da potência do sinal.

As políticas de migração, a princípio, podem ser executadas tanto no servidor quanto no cliente. Contudo, a execução das políticas no servidor implica na transmissão de mensagens adicionais por parte do cliente, indicando ao servidor as potências dos diversos sinais capturados e aumentando, assim, o fluxo de dados na rede. Mas, a execução no cliente pode prejudicar a recepção e visualização do vídeo, devido ao baixo poder de processamento dos dispositivos de pequeno porte.

Por fim, com a escassez de implementações sobre redes sem fio, este trabalho busca desenvolver um aplicativo para transmitir *streams* de vídeo e implementar esses três mecanismos de adaptação. Assim, no próximo capítulo, são descritos o sistema proposto, alguns detalhes do código e algumas bibliotecas utilizadas.

Capítulo 5

Implementação do Sistema Proposto

5.1 Introdução

O ambiente hospitalar, como visto anteriormente, é caracterizado pela constante movimentação dos profissionais e pela necessidade da troca rápida e eficiente de informações entre equipes. Diante dessas características, a implementação de uma aplicação móvel sobre redes infra-estruturadas pode ser uma solução eficiente e de baixo custo.

Assim, esse capítulo têm o objetivo de descrever o aplicativo desenvolvido para a transferência de *streams* de vídeo, com qualidade de serviço, sobre redes infra-estruturadas e testar o desempenho das políticas de migração propostas. A qualidade fornecida pelo aplicativo está garantida pelo uso de *buffers*, pelo uso de uma política de migração e através de indicação visual sobre a qualidade do sinal, como descritos no Capítulo 4.

Serão apresentados com mais detalhes neste capítulo o sistema desenvolvido com foco nos ambientes hospitalares, os programas que formam o ambiente de desenvolvimento, as bibliotecas e as configurações dos *hardwares* utilizadas.

5.2 Proposta do Sistema

A diversidade dos problemas intrínsecos às redes sem fio e a dificuldade de realização de testes das aplicações móveis em ambientes reais são alguns fatores que justificam o freqüente uso de simulações, como forma de comprovar a eficácia das técnicas de adaptação já desenvolvidas nessa área de pesquisa.

Na tentativa de suprir essa escassez de implementações, esse projeto busca desenvolver um aplicativo para suporte a transmissão de *streams* multimídia em redes infra-

estruturadas. Além de realizar a captura, transmissão e recepção dos dados multimídia, esse aplicativo realiza diversas tarefas concorrentes, tais como: gerenciamento das conexões, compactação e descompactação das *streams* de vídeo, monitoração da qualidade dos sinais emitidos pelos pontos de acessos e gerenciamento de *buffers* de armazenamento, como esquematizado na Figura 5.1.

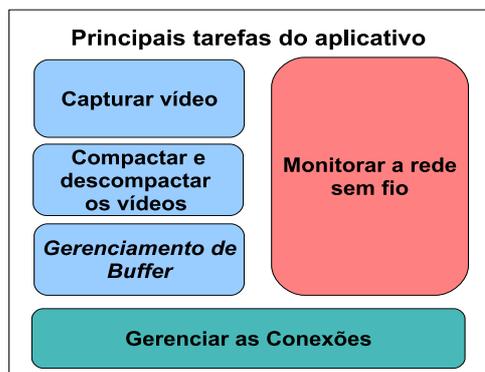


Figura 5.1: Resumo das tarefas do aplicativo.

O sistema é composto por duas aplicações, uma cliente e a outra servidor. O módulo servidor foi desenvolvido para ser executado apenas em computadores pessoais, do tipo *desktop* e *notebook* e também pode desempenhar as funções de cliente quando necessário. Já a parte cliente é representada por uma outra aplicação, desenvolvida apenas para dispositivos portáteis que utilizam o sistema operacional WindowsCE, como os *Pocket PCs*. Assim, o cliente se conecta ao servidor e pode, por exemplo, requisitar o início de um fluxo multimídia. Depois disso, o servidor captura os dados vindos de uma câmera, realiza a compressão e os armazena em *buffer*. Os quadros compactados são armazenados para facilitar o envio em diversas taxas de transferência. Ou seja, uma tarefa, disparada de tempos em tempos, recupera do *buffer* um quadro e o transmite. Assim, no caso dos dispositivos portáteis *Pocket PCs*, essa tarefa deve ser acionada a cada um segundo, uma vez que os dispositivos atuais são limitados na capacidade de processamento e por isso devem receber apenas 1 *frame* por segundo.

Para efeito do teste das técnicas propostas no Capítulo 4, foi desenvolvido um aplicativo que implementa características de um programa de videoconferência. Sendo assim, após um cliente se conectar ao servidor a videoconferência é estabelecida. A transmissão de *streams* de vídeo ocorre apenas no sentido do servidor para o cliente. Essa limitação se dá devido a utilização de *Pocket PCs* sem câmera acopladas e principalmente, por essa função não ser fundamental nos experimentos em foco.

As mensagens de controle da videoconferência são transmitidas utilizando uma cone-

xão confiável. Através desta, tanto o cliente quanto o servidor podem requisitar outros tipos de conexões, que serão responsáveis exclusivamente pela transferência de texto, de vídeo ou ainda de áudio. É pertinente destacar que, embora exista uma indicação de uma conexão de áudio, a aplicação não está preparada para capturar o áudio e transmiti-lo. Assim, uma videoconferência é composta por uma conexão de controle, e esta poderá gerenciar até três conexões para os fluxos de texto, vídeo e, para o futuro, áudio, como mostra a Figura 5.2.

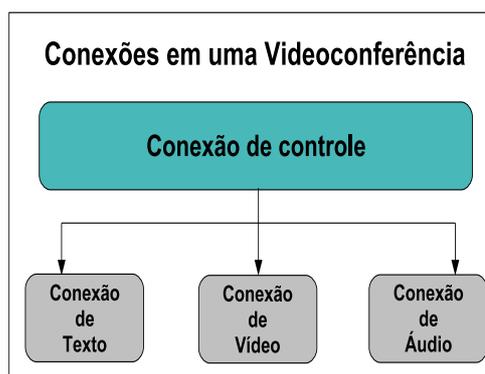


Figura 5.2: Conexões possíveis em uma videoconferência.

As classes que gerenciam as conexões são capazes de detectar qualquer queda na conexão e iniciar um processo de reconexão, sem a intervenção do usuário. Caso esta reconexão não ocorra dentro de um determinado tempo, toda a videoconferência será encerrada automaticamente.

Após estabelecer uma conexão de vídeo, o servidor inicia a captura e a compressão das *streams* de vídeo, no padrão H263 ou JPEG. A aplicação cliente recebe essas *streams*, armazena-as em um *buffer* local, decodifica-as e as visualiza, como mostrado na Figura 5.3.

Como dito anteriormente, as aplicações que realizam transferência multimídia sobre redes sem fio necessitam adaptar-se às constantes mudanças ocorridas na rede durante as transmissões. Para isso, o cliente dispõe de um componente responsável por monitorar a qualidade dos sinais dos pontos de acesso e receber os eventos gerados pelo *hardware* local (NIC). Esse componente é capaz de determinar o melhor ponto de acesso e o melhor momento para realizar a migração. Para isso, o componente utiliza os sinais capturados dos diversos pontos de acesso. Além de implementar as políticas de migração, que foram apresentadas na seção 4.4, esse componente recebe os eventos ocorridos diretamente do NIC. Com isso, é capaz de identificar imediatamente uma desconexão e iniciar um processo de reconexão. Quando a reconexão não é possível, uma migração de emergência é realizada, de forma transparente ao usuário. A Figura 5.4 apresenta esse componente.

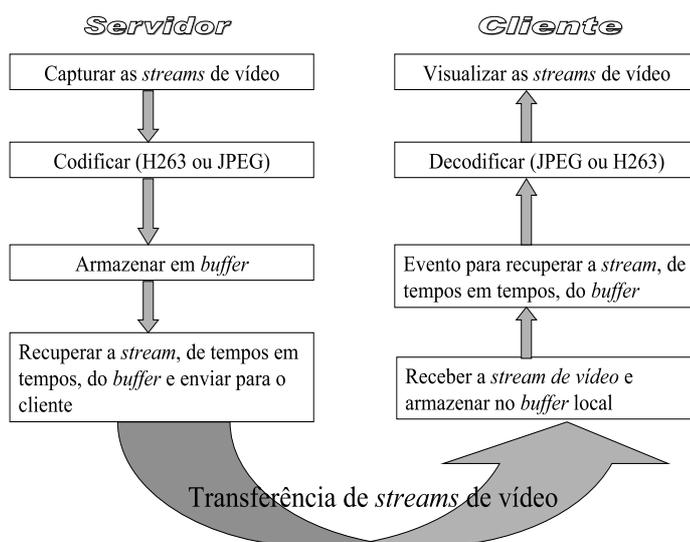


Figura 5.3: Transferência dos quadros de vídeo.

Esse componente contém uma biblioteca que interage com o *hardware* diretamente: ela recebe os eventos gerados pelo NIC e comunica imediatamente ao gerenciador. O gerenciador é responsável por providenciar uma reconexão de urgência em momentos de desconexões abruptas, capturar os sinais dos pontos de acesso, utilizando para isso a classe específica, e ainda utilizar a política de migração, escolhida pelo usuário, para determinar o melhor ponto de acesso para uma eventual migração.

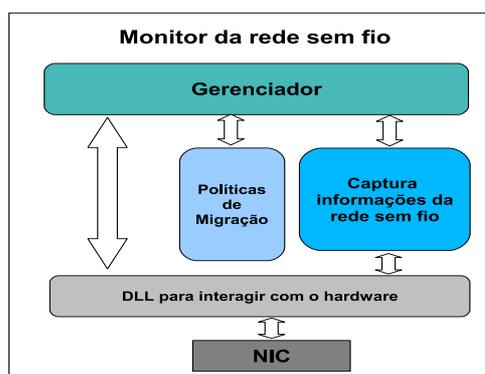


Figura 5.4: Componente para monitorar a rede sem fio.

5.3 Interface e Descrição do Sistema

Para efeito do teste dos mecanismos de adaptabilidade a aplicação pode ser iniciada por dois computadores pessoais, do tipo *desktop* ou *notebook*. O mesmo programa pode funcionar como servidor em um terminal e como cliente em outro. Já no caso do aplicativo

ser executado em um computador pessoal em conjunto com um dispositivo portátil, o primeiro executará o módulo servidor, e o segundo, o módulo cliente. Para estabelecer uma conexão e então iniciar uma videoconferência, o servidor entra em estado de espera por uma conexão e o cliente se conecta a ele utilizando seu endereço IP.

Na execução que utiliza apenas computadores pessoais, as interfaces gráficas da mesma aplicação podem exercer as funções de servidor e de cliente. A Figura 5.5(a) mostra o menu *File*, com a opção "aguardar por uma conexão" selecionada. Dessa forma, o servidor cria uma tarefa independente que fica aguardando por uma conexão. Já a Figura 5.5(b) mostra a opção "conectar" selecionada no menu *File*, indicando que a aplicação está sendo executada no módulo cliente. Esse menu visualiza um formulário no qual o usuário pode digitar o endereço IP da máquina onde está sendo executado o servidor. Essa opção também está disponível na versão para dispositivos portáteis, que será apresentada adiante.

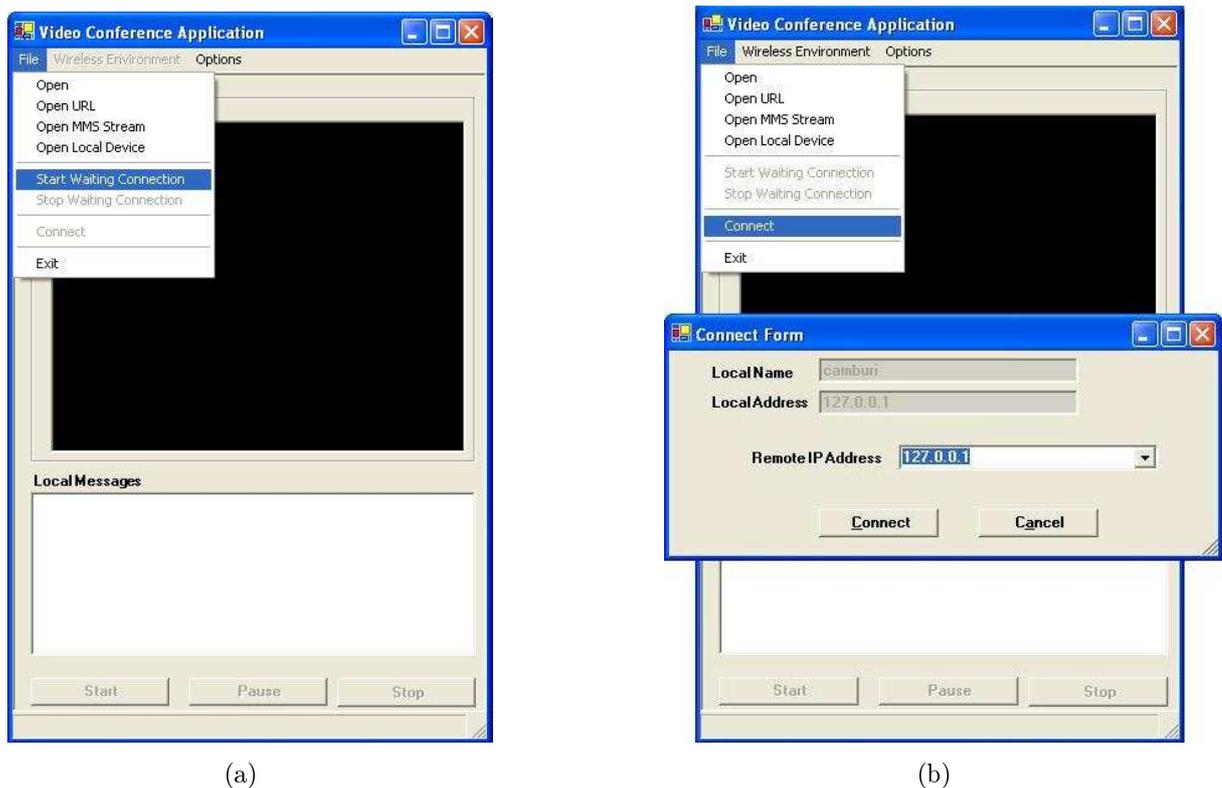


Figura 5.5: Iniciando o aplicativo servidor: (a) Iniciar a espera por uma conexão; (b) Conectando com o computador remoto.

No momento em que um cliente se conecta ao servidor, na implementação para teste, esse servidor interage com o usuário, através de um formulário (Figura 5.6(a)), para decidir se aceita ou rejeita a conexão com o cliente, porém essa decisão poderia ocorrer de forma automática. Após a decisão, uma mensagem é retornada ao cliente, indicando a decisão

do servidor. Caso a mensagem recebida pelo cliente seja positiva, uma janela para a apresentação da videoconferência será exibida em cada dispositivo. A Figura 5.6(c) mostra as mensagens e a janela que exibe uma videoconferência. Os dispositivos móveis executam uma versão com menos recursos gráficos, devido às suas limitações de processamento.

A videoconferência tem um identificador formado por um valor numérico seguido de um palavra, que está registrado na interface gráfica juntamente com o nome e o endereço IP do computador remoto. A palavra no identificador da videoconferência indica qual a característica da aplicação que está sendo executada: servidor, a que espera por uma conexão; cliente, a que iniciou a conferência. Por fim, a Figura 5.6(b) mostra a caixa de diálogo utilizada para confirmar, ou não, o início de conexões para a transferência de vídeo, usada também durante o processo de conexão de texto e áudio.

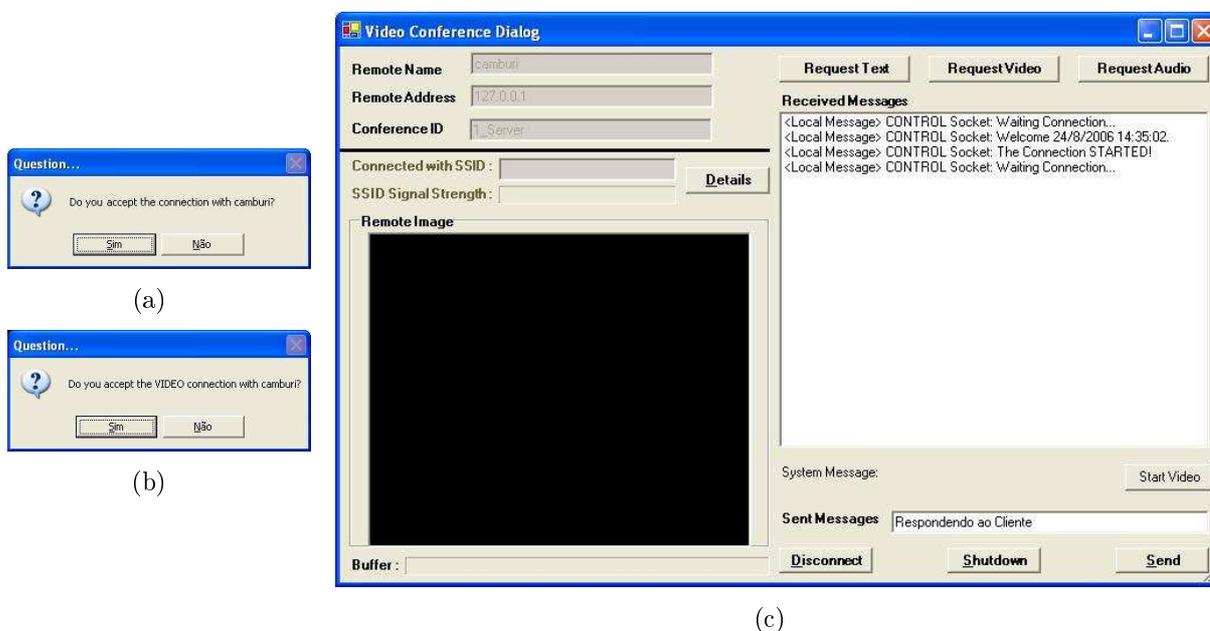


Figura 5.6: Iniciando uma conexão com o servidor: (a) Aceitar ou rejeitar uma conexão; (b) Aceitar ou rejeitar uma conexão de vídeo; (c) Formulário que representa a videoconferência.

A interface gráfica do sistema permite ao usuário capturar as *streams* de vídeo de dispositivos locais, como as *webcams*, por exemplo. Após a conexão de vídeo ser estabelecida, o servidor realiza o processo de captura, compressão e armazenamento do vídeo e, em seguida, transfere a *stream* ao cliente. A Figura 5.7(a) apresenta a captura do vídeo no servidor e a Figura 5.7(b) mostra o vídeo no cliente.

A Figura 5.8 mostra algumas imagens do sistema cliente desenvolvido para pequenos dispositivos, executado no simulador de Pocket PC 2003. Na primeira, visualiza-se o

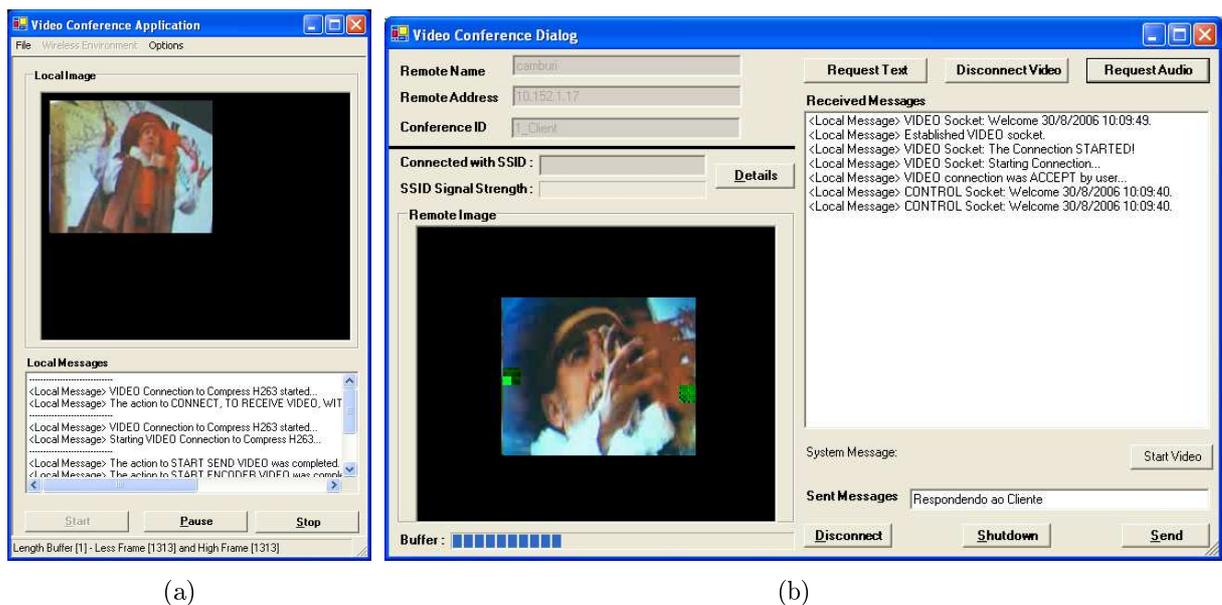


Figura 5.7: Transferência de Vídeo: (a) Vídeo capturado localmente; (b) Vídeo recebido pela máquina remota.

menu principal da aplicação, no qual existe a opção "connect"; na segunda, o usuário pode digitar o endereço IP da máquina em que está a aplicação servidor; por fim, na terceira, é mostrado o recebimento de um vídeo pela aplicação.



Figura 5.8: Imagens do aplicativo cliente, no *Pocket PC*: (a) *Menu File*; (b) Formulário de conexão com o aplicativo servidor; (c) Vídeo no dispositivo.

Nas seções 5.3.1, 5.3.2 e 5.3.3 serão descritas, de forma breve, a comunicação entre os módulos cliente e servidor, a estrutura do código e a implementação.

5.3.1 Comunicação

O componente responsável por toda a comunicação do sistema utiliza diversos tipos de mensagens¹: as de controle, que estabelecem e encerram todos os tipos de conexões e as mensagens que contêm informações sobre a rede sem fio, que eventualmente podem ser trocadas entre o módulo cliente e o servidor, entre outras. Essas mensagens, com informações sobre as condições da rede sem fio, são utilizadas apenas se o usuário definir o servidor como responsável pela execução das políticas de migração. A vantagem disso é que o servidor pode conhecer o grau de utilização de cada ponto de acesso e utilizar essa informação para determinar o melhor roteador de destino, mas, isso pode gerar uma desvantagem que é a sobrecarga de mensagens na rede.

O sistema estabelece que as mensagens de controle devem ter garantia de entrega e, portanto, utiliza nessa transmissão o protocolo de comunicação TCP. Já as mensagens de texto e vídeo são transmitidas por meio do protocolo UDP. Contudo, o sistema permite que o usuário configure o protocolo TCP também para a transmissão dessas últimas mensagens, embora isso não seja recomendado, principalmente nas redes sem fio, uma vez que esse protocolo produz um grande número de mensagens adicionais e diversos problemas em função da forma como interpreta as perdas de mensagens.

5.3.2 Codificação

A aplicação desenvolvida para computadores pessoais tem uma estrutura organizada em seis diretórios; já no sistema criado para os dispositivos de pequeno porte os diretórios são reduzidos a apenas três. O diretório raiz dos dois sistemas armazena as classes iniciais e algumas interfaces gráficas.

Os diretórios comuns aos dois sistemas são:

- *VideoConference* - armazena as classes que compõem uma videoconferência e sua interface gráfica. Conta ainda com outros subdiretórios: um reservado para as classes do *buffer* utilizado na recepção dos vídeos; outro para as classes que gerenciam o evento responsável por retirar um quadro de vídeo do *buffer*, decodificá-lo e visualizá-

¹A estrutura das principais mensagens do aplicativo pode ser encontrada no apêndice A.

lo; por último, um terceiro que armazena as constantes e funções utilizadas por todas as classes de uma videoconferência.

- **Connections** - armazena todas as classes que gerenciam as conexões de controle, texto e vídeo do sistema. Existe ainda um subdiretório com as classes que fornecem conexões que utilizam os protocolos TCP e UDP, e outro com as classes que contêm as constantes e funções gerais utilizadas por esse componente.
- **WirelessEnvironment** - contém as classes que formam o componente responsável pela captura e gerenciamento das informações da rede sem fio, como o nome do roteador ao qual o cliente está conectado e a qualidade do sinal dos pontos de acesso alcançáveis. Esse diretório contém, ainda, outros dois subdiretórios que armazenam as classes dos protocolos de migração e as constantes e funções gerais utilizadas por esse componente.

Já a aplicação desenvolvida para *desktops* e *notebooks* contém outros três diretórios, isso porque essa aplicação deve ser capaz de capturar os vídeos, codificá-los e armazená-los antes de transferi-los ao cliente. Os diretórios são descritos com mais detalhes a seguir:

- **BufferCapturingVideo** - esse diretório contém as classes de um *buffer* no qual são armazenadas as *streams* de vídeo capturadas e codificadas. Existem ainda as classes do evento responsável por retirar do *buffer* um quadro de vídeo e transferi-lo ao cliente e ainda, um coletor de lixo que é acionado de tempos em tempos para retirar do *buffer* os quadros já transmitidos.
- **CaptureImage** - nesse diretório existem classes que recuperam *streams* de vídeo de diversas fontes, tais como câmera, arquivos (*.avi), URL de Microsoft Media Service (MMS) *streams* e URL de imagens JPEG que são capturadas por *web* câmeras e atualizadas constantemente na Internet. Porém, esse diretório está sendo utilizado apenas para captura da *webcam* quando a codificação do vídeo definida pelo usuário for o padrão JPEG. Quando o usuário define o padrão H263, o sistema utiliza as classes do diretório **ControlerH263**.
- **ControlerH263** - contém classes utilizadas para iniciar um outro pequeno sistema e métodos utilizados na comunicação entre as aplicações. Desenvolvido com a linguagem C++, esse outro sistema captura as imagens da *webcam*, codifica-as no padrão H263 e envia cada quadro para a aplicação principal.

Os códigos dos sistemas cliente e servidor garantem independência de alguns componentes. Assim, classes de diretórios como *Connection* e *WirelessEnvironment* podem ser facilmente reutilizadas por qualquer outro projeto. Outros detalhes sobre a estrutura do código desenvolvido podem ser encontrados no apêndice A.

5.3.3 Implementação

Para dar suporte ao sistema, foram desenvolvidas algumas bibliotecas e uma pequena aplicação, que serão detalhadas a seguir. Todas utilizam a linguagem C++.

O módulo servidor faz uso de duas DLLs, uma para decodificar cada quadro de vídeo que está no formato H263, e outra para recuperar informações específicas da rede sem fio. Esta última é uma derivação do projeto *WRAPI* (*Wireless Research API*), que será visto com mais detalhes nas próximas seções. Foram necessárias mudanças na biblioteca WRAPI, uma vez que algumas funções não estavam implementadas, e outras simplesmente não funcionavam no sistema operacional WindowsXP, como já estava previsto no projeto original. Ainda para o módulo servidor, uma pequena aplicação foi desenvolvida para capturar as imagens da câmera e codificá-las em formato H263.

O módulo cliente, construído para dispositivos móveis, faz uso de outras duas DLLs, uma para decodificar as *streams* de vídeo que estão no padrão H263, e a outra para interagir diretamente com o NIC, recebendo imediatamente todos os eventos que o *hardware* pode produzir - por exemplo, o evento gerado quando o dispositivo móvel se desconecta do ponto de acesso. Embora as funções sejam parecidas com aquelas das DLLs desenvolvidas para o módulo servidor, é necessário o desenvolvimento de versões distintas, pois os *hardwares* e os sistemas operacionais são muito diferentes nos computadores pessoais e nos dispositivos portáteis. A DLL para interação do módulo cliente com o NIC foi desenvolvida com base no sistema WiFi-Gator da HP, que também será apresentado com mais detalhes nas próximas seções.

As bibliotecas para interagir diretamente com o *hardware* e obter informações sobre as redes sem fio utilizam o *NDIS User Mode I/O Protocol* (NDISUIO). O NDIS é a forma mais simples para interagir com os diversos adaptadores de rede (Ethernet, WLAN, entre outros) a partir da camada de aplicação. O *driver* NDISUIO fornece diversos suportes, como filtros de pacotes, envio e recebimento de dados e manipulação dos eventos *plug-and-play* produzidos pelo adaptador de rede. Ainda, NDISUIO pode diretamente abrir um *NDIS miniport driver* (i.e., *network card driver*) para enviar requisições, alterar e recuperar

informações. O protocolo NDISUIO fornece uma interface entre a camada de aplicação² e o NDIS, por meio do *DeviceIoControl*.

Para controlar os eventos e interagir com o NIC, a aplicação necessita desligar o WZC (*Wireless Zero Configuration*), um serviço do Windows que é normalmente iniciado em tempo de *boot*. Esse serviço fornece uma autoconfiguração para adaptadores sem fio 802.11, os NICs; para isso, busca pontos de acesso disponíveis e se associa com o que possui melhor qualidade do sinal. Esse serviço automaticamente se conecta ao *driver* NDISUIO impedindo que outras aplicações ou DLLs, como a WRAPI ou WiFi-Gator, utilizem suas funções. Por isso, antes de interagir com o NIC, o serviço WZC deve ser desligado, tanto nos dispositivos móveis quanto nos computadores pessoais. Contudo, a aplicação cliente, para os dispositivos portáteis, desliga esse serviço automaticamente no momento de sua iniciação.

Por fim, as aplicações servidor e cliente foram desenvolvidas por meio da linguagem de programação C# e fazem uso de múltiplas *threads* e programação orientada a eventos. No apêndice A pode ser encontrada toda a documentação técnica da aplicação desenvolvida e também os diagramas de classes e de eventos.

5.4 Ferramentas

Para o desenvolvimento da aplicação foram utilizados os ambientes da Microsoft, Visual Studio .NET e eMbedded Visual C++ 4.0. Os dois ambientes permitem o desenvolvimento de sistemas para dispositivos portáteis, como o Pocket PC 2003. Porém, o primeiro é um ambiente mais completo e comporta linguagens como VB.NET, J#, C++ e C#, e o segundo só é compatível com a linguagem C++, mas, por ser gratuito, pode ser uma alternativa para o desenvolvimento de aplicações portáteis.

No Visual Studio .NET, foram desenvolvidas as aplicações servidor e cliente, por meio da linguagem C#; já todos os códigos desenvolvidos em C++ foram feitos com base no ambiente eMbedded Visual.

²Mesmo sendo executado pelo núcleo do sistema operacional, através de chamadas IOCTL ao *driver* NDISPROT, NDIS pode ser acessado pela camada de aplicação. Mais detalhes se encontram disponíveis na Web em <http://sysnet.ucsd.edu/pawn/wrapi/>.

5.5 Bibliotecas

Todas as bibliotecas e o sistema de captura e codificação H263 utilizados pela aplicação foram implementados nesse projeto. Esses componentes se basearam em projetos *open source* disponíveis na Internet e são listados nesta subseção.

O projeto VideoNet [Talekar 2004], desenvolvido para a captura, codificação H263 e transferência de quadros de vídeo sobre redes LAN, está disponibilizado como *open source* e foi utilizado como uma fonte de consulta para o desenvolvimento do sistema de captura e codificação H263. O sistema implementado, chamado EncoderH263, está apresentado com mais detalhes no apêndice A.

As quatro bibliotecas (DLLs) foram desenvolvidas neste projeto para suprir duas necessidades: a de decodificar as *streams* de vídeo, codificadas em padrão H263, e a de interagir com as placas de rede sem fio. Entretanto, foram desenvolvidas duas versões para cada necessidade, uma para os computadores pessoais e outra para os dispositivos móveis.

Visando os computadores pessoais, foram desenvolvidas as bibliotecas DecoderH263 e WRAPI para cada tipo de necessidade destacada anteriormente. A primeira tem como base o projeto VideoNet e realiza algumas alterações, e a segunda foi criada a partir do projeto WRAPI ([Balachandran e Voelker 2002]). Esse projeto, desenvolvido na Universidade da Califórnia, consiste em um conjunto de ferramentas responsáveis por monitorar as redes sem fio para sistemas que utilizam o WindowsXP. A biblioteca de *software* WRAPI 2.0 (wrapi.dll) permite às aplicações executadas em estações móveis, como *notebooks*, obter informações sobre a rede IEEE 802.11 à qual estão conectadas. A versão original dessa biblioteca foi desenvolvida para redes sem fio IEEE 802.11b, o que gerou a necessidade de alguns ajustes, pois a rede sem fio aqui utilizada tem como base o padrão IEEE 802.11g.

Nos dispositivos portáteis, as bibliotecas DecoderH263ARM e WiFiConnection substituem as duas anteriormente apresentadas. Novamente, a primeira tem como base o projeto VideoNet, mas, nesse caso, muitas alterações foram necessárias, pois o projeto original não foi desenvolvido para os processadores ARM dos *pocket PCs*. A segunda biblioteca foi desenvolvida a partir do projeto WiFi-Gator da HP. Esse projeto tem o objetivo de interagir diretamente com a placa de rede sem fio e capturar os eventos por ela gerados. Algumas funcionalidades foram adicionadas na criação da biblioteca WiFiConnection. Ou seja, embora a interface com a placa de rede forneça várias funcionalidades, algumas não estavam implementadas; ainda, os eventos capturados pela placa passaram a ser trans-

portados para a aplicação cliente principal, codificada em C#.

5.6 Configuração

Os dispositivos móveis utilizados são um Pocket da HP com processador Intel PXA255 de 400MHz, memória de 64MB SDRAM (56MB acessíveis pelo usuário), ROM de 32MB e Sistema Operacional Microsoft Mobile Pocket PC 2003, e um laptop Latitude 120L da Dell, com processador Pentium M 740 (1,73 GHz) e Sistema Operacional Windows XP Home, SP2.

O *desktop*, utilizado para a execução do módulo servidor, é um Pentium 4, 2,80GHz, com 512MB de memória RAM e sistema operacional Windows XP *Home Edition* com SP2. Os pontos de acessos, conectados à rede LAN, são da Linksys, modelo WRT54G, padrão 802.11b/g.

5.7 Conclusão

Todas as dificuldades encontradas na implementação das adaptações propostas em relação à tecnologia sem fio em dispositivos móveis, foram superadas. No entanto, os dispositivos de pequeno porte aqui utilizados são limitados em processamento, e isso impossibilitou a transferência de uma taxa de vídeo maior que a de um quadro por segundo. Dentre os obstáculos superados durante a implementação do projeto, é importante ressaltar os seguintes:

- O serviço WZC, iniciado no tempo de *boot* dos sistemas operacionais Windows, precisa ser desligado antes das aplicações cliente e servidor serem iniciadas. Enquanto a aplicação cliente desliga o serviço automaticamente ao ser iniciada, a aplicação servidor requer que o usuário desligue o serviço diretamente no painel de controle do Windows XP;
- A interação com os NICs através dos *drivers* NDISUIO e do conjunto de identificadores de objetos (OIDs) especificados pelo NDIS é a forma mais simples e eficiente para interagir com os adaptadores de rede.

Por fim, no próximo capítulo, serão apresentados o cenário e as métricas utilizadas para testar o sistema, juntamente com os resultados obtidos. Os testes foram realizados

em um ambiente real e ajudaram a definir as melhores faixas de valores para os níveis de qualidade do sinal nas políticas de migração.

Capítulo 6

Avaliação do Sistema

6.1 Introdução

Os diversos problemas das redes sem fio e a diversidade dos motivos que causam interferências na conexão são alguns fatos que dificultam a realização de testes em redes infra-estruturadas. Essa dificuldade ocorre porque muitas vezes é impossível conhecer todos os fatores que, de uma forma ou outra, influenciam nos resultados de cada teste. Assim, a realização dos testes, nesses ambientes, deve ser exaustiva.

A implementação das políticas de migração levou a muitos testes que objetivavam propor os valores limites de cada faixa e garantir o menor número possível de migrações. Esse capítulo descreve os cenários em que os testes foram realizados, as métricas utilizadas e os resultados alcançados.

6.2 Cenário

O ambiente de realização dos testes é formado por uma rede LAN, três pontos de acesso, chamados de **str**, **compmov** e **compmov2**, um *pocket PC* e um *notebook*.

Utilizou-se a rede do laboratório da pós-graduação do Instituto de Computação da Universidade Federal Fluminense. Os pontos de acesso foram espalhados de forma a garantir a conectividade no interior do laboratório e em toda a extensão do corredor do Instituto.

Nesse cenário, foram realizados testes tanto com os dispositivos portáteis *pocket PCs* quanto com o *notebook*. Contudo, somente a aplicação desenvolvida para o primeiro tipo de dispositivo é capaz, nessa versão do sistema, de realizar a troca automática de ponto

de acesso.

Os testes com o *notebook* visaram apenas a comprovação da possibilidade de transferência de vídeo em taxas mais elevadas e ao aprendizado sobre a interação com os adaptadores de rede sem fio nesses equipamentos. Os resultados com esses dispositivos foram satisfatórios, comprovando a eficiência do sistema em altas taxas de transferência. Contudo, o objetivo dos testes apresentados nesse capítulo é avaliar a variação da taxa de perda e do atraso médio dos pacotes, juntamente com o *jitter*, em cada política de migração proposta.

As políticas de migração foram testadas somente em dispositivos do tipo *pocket PC* e a velocidade de deslocamento do cliente nos testes foi compatível com uma pessoa andando normalmente. Para esses testes, os pontos de acesso foram distribuídos no espaço do Instituto de duas formas distintas. A primeira foi caracterizada por uma sobreposição considerável das potências dos sinais, ao contrário da segunda, que buscou minimizar essa sobreposição, como mostra, de uma maneira aproximada, a Figura 6.1

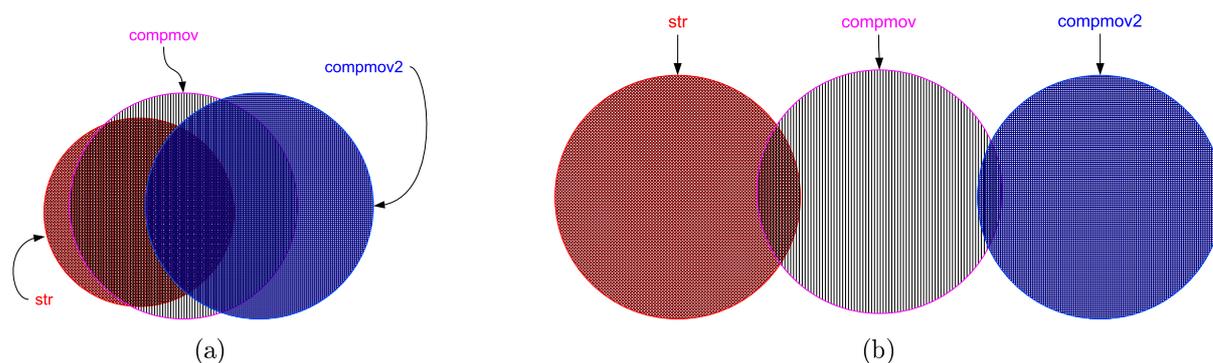


Figura 6.1: Disposição dos sinais dos pontos de acesso: (a) Com muita sobreposição; (b) Com sobreposição reduzida.

Na Figura 6.1(a), o raio de alcance do ponto de acesso **str** é reduzido. Isso ocorre devido sua localização mais isolada, caracterizada pela existência de um número maior de obstáculos, como paredes de concreto, que ajudam na redução da propagação do sinal.

Já na Figura 6.1(b), os pontos de acesso **str** e **compmov2** foram afastados em relação ao **compmov**, o que permitiu uma propagação do sinal mais homogênea.

6.3 Métricas

Com relação à transferência dos quadros de vídeo, a aplicação pode ser configurada para transmitir nas taxas 15, 12, 6 ou 1 quadro por segundo. Porém, para os dispositivos *Pocket PCs* disponibilizados, só é possível o envio de apenas um quadro por segundo. Essa limitação ocorre devido ao baixo poder de processamento apresentado por um processador de apenas 400MHz e pela falta de um processador exclusivo para a parte gráfica. Essa limitação tecnológica, juntamente com a necessidade de executar diversas tarefas em paralelo, tais como: interagir com o adaptador de rede sem fio; receber e armazenar os quadros de vídeo no *buffer*; retirar cada quadro desse *buffer*, decodificá-lo e visualizá-lo; e, ainda, analisar as potências dos sinais de cada ponto de acesso e determinar a necessidade de migração explicam a taxa de transmissão reduzida.

Os pontos de acesso utilizados, da Linksys, modelo WRT54G, padrão 802.11b/g, determinam a leitura de sua potência do sinal a partir de outros dispositivos na faixa de valores entre -10 e -90 dBm, que representam o melhor e o pior valor, respectivamente. O aplicativo desenvolvido registra o valor zero (nos gráficos indicado por -100 dBm), todas as vezes que um dos roteadores não está alcançável.

As potências de todos os pontos de acesso são capturadas e armazenadas a cada três segundos. No final da terceira varredura, o sistema aciona a política de migração escolhida, determina o novo ponto de acesso e só então realiza a migração, se necessário. No fim, os valores armazenados são removidos e inicia-se um novo ciclo. Foram realizados ainda, testes com varreduras a cada cinco segundos e aplicação da política de migração a cada três varreduras, porém, o uso desses valores proporcionou um grande número de desconexões abruptas. Assim, pode-se concluir que o tempo total de nove segundos (três segundos para varrer os sinais e três varreduras para acionar a política de migração) é um tempo suficiente para não ocorrer desconexões antes da realização de uma migração controlada. Essas desconexões são facilmente explicáveis, pois a velocidade de deslocamento ocasiona a perda do sinal antes da política de migração ser acionada. Ou seja, quinze segundos (na melhor hipótese, pois existe a concorrência entre as diversas *threads* do aplicativo), entre uma análise e outra dos sinais, é um tempo demasiadamente grande e deve ser evitado.

Para determinar a taxa de perda e o atraso médio dos pacotes, juntamente com o *jitter*, o aplicativo faz uma análise *off-line* do registro de cada grupo de nove pacotes recebidos. A definição da quantidade de pacotes analisados em cada grupo baseou-se no intervalo de tempo médio entre diferentes aplicações da política de migração, que está em torno

de nove segundos. É importante ressaltar que caso em um grupo de nove pacotes seja recebido apenas um, ou nenhum, o atraso médio e o *jitter* não são calculados e, portanto, nos gráficos correspondentes nada será registrado nesse instante.

Diversos testes foram realizados nos dois cenários propostos e os resultados são detalhados na próxima seção.

6.4 Resultados Obtidos

Os testes realizados em simuladores permitem a experimentação em cenários completamente sem interferências e a execução de diferentes testes sempre nas mesmas condições. Porém, isso não é possível em ambientes reais, pois a interferência, em alguns casos, não pode ser prevista. Na tentativa de manter um padrão, todos os testes foram realizados sempre em um tempo total de aproximadamente quinze minutos e utilizando a mesma rota. Os gráficos (a), (b) e (c) das figuras 6.3 a 6.10 representando, respectivamente, as potências dos sinais nos pontos de acesso **str**, **compmov** e **compmov2** ao longo de cada teste indicam que, aproximadamente, o mesmo padrão foi obtido. Ressalta-se, contudo, que um padrão único é impossível de ser alcançado em um cenário real.

Os testes são iniciados com o cliente conectado ao ponto de acesso **str**. Após permanecer cerca de três minutos, uma migração para o roteador **compmov** é estimulada. Nos próximos três minutos o cliente se mantém o mais próximo possível desse roteador e, ao término desse tempo, ele então se encaminha de volta ao **str** e retorna ao **compmov** em um intervalo total de um minuto. Depois de permanecer mais um minuto próximo ao **compmov**, o cliente se encaminha em direção ao ponto de acesso **compmov2**. Após a migração permanece próximo a esse roteador por um intervalo entre dois e três minutos. Ao fim desse tempo, o cliente retorna em direção ao **compmov**, pára diante deste, durante um minuto, e se encaminha de volta ao **str** onde encerra a conexão. A Figura 6.2 mostra uma aproximação do trajeto que foi realizado durante os testes.

Para um melhor entendimento dos resultados, em cada teste foram registrados em arquivos todas as medições da potência do sinal de cada ponto de acesso. Com esses valores foram criados gráficos que demonstram a variação do sinal ao longo de cada teste.

Os resultados serão analisados em dois grupos distintos, de acordo com a disposição dos roteadores. Cada grupo será dividido em quatro subgrupos, três subgrupos são para cada política de migração analisada e o quarto para o caso de não se utilizar política alguma. Diferentes gráficos indicam a perda e o atraso médio dos pacotes, o *jitter*, as

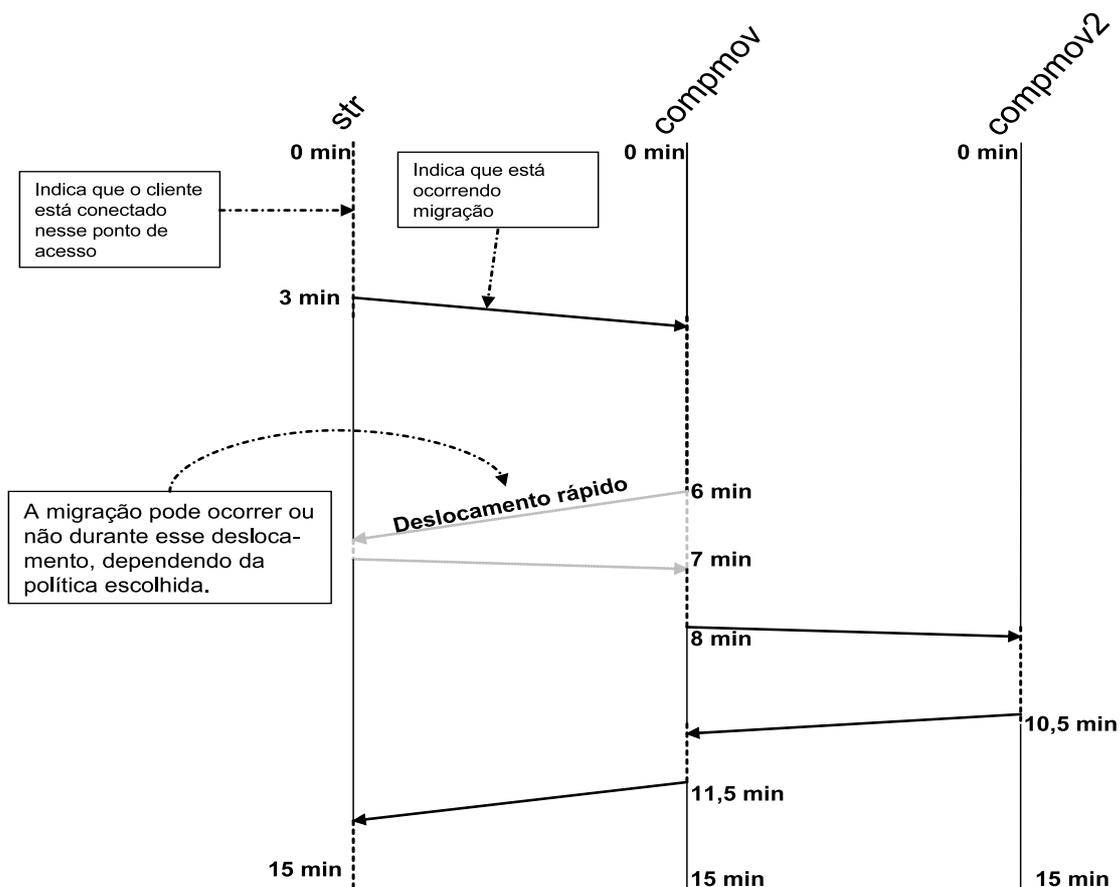


Figura 6.2: Esquema aproximado do trajeto percorrido durante os testes.

potência dos sinais em cada um dos três roteadores e, ainda, em alguns gráficos há a indicação dos momentos que ocorreram migração, ou perda abrupta da conexão.

6.4.1 Disposição dos Pontos de Acesso com Grande Sobreposição dos Sinais

Nesse cenário deve-se destacar o fato da migração ocorrer de um ponto de acesso onde a potência do sinal está baixa, para um outro ponto de acesso que está com a potência do sinal mais alta. Essa observação é fundamental para a análise dos resultados, uma vez que a baixa potência do sinal tem consequência direta na capacidade de transmissão da rede sem fio, através do mecanismo de ARC, como visto no capítulo 3.

6.4.1.1 Sem Política de Migração

Sem uma política de migração estabelecida a mudança de ponto de acesso ocorre apenas quando há a perda da conexão. Nessa ocasião, o tempo de conexão com outro ponto de acesso, ou até o mesmo pode variar entre seis e oito segundos.

A variação da potência dos sinais, de cada ponto de acesso, como visto pelo receptor ao longo do experimento, pode ser visualizada pelas figuras 6.3(a), 6.3(b) e 6.3(c). A Figura 6.3(d), apresenta a variação da perda de pacotes juntamente com os momentos que ocorreram perda de conexão. Nessa ocasião, o intervalo entre o momento que ocorreu a perda da conexão até a captura do evento gerado pelo novo pedido de conexão, variou entre seis e sete segundos. Porém, é importante ressaltar que há a necessidade da espera de outro intervalo de tempo, após a conexão com o novo roteador, para o restabelecimento das conexões de controle e de vídeo da aplicação. Essas reconexões são necessárias pois o novo roteador fornece, via protocolo DHCP, um novo número IP ao cliente.

Observando ainda a Figura 6.3(d), pode-se perceber que a queda de conexão do ponto de acesso *str*, no instante 522s, resultou em um intervalo de vinte e sete pacotes para se estabilizar, chegando a haver 100% de perdas de pacotes.

A Figura 6.3(e) mostra um atraso médio entre os pacotes sempre em torno de um segundo. Esse resultado é coerente, uma vez que a taxa de transmissão do vídeo é de apenas 1 fps. As exceções ocorrem justamente nos momentos de perda da conexão.

Já a Figura 6.3(f) indica a variação do atraso fim-a-fim dos pacotes. Nos momentos de perda da conexão, esse valor, geralmente, sofre uma grande alteração. Isso só não ocorre, como no instante 742s, quando muitos pacotes chegam com um atraso muito grande, em função, por exemplo, da migração, gerando uma variação do atraso pequena.

6.4.1.2 Política de Migração - Média Simples

A maior desvantagem dessa política é o grande número de migrações, que gera muitos momentos de desconexão. Essas desconexões acarretam grandes perdas de pacotes, o que reduz a qualidade do vídeo. Diferentemente das perdas abruptas de conexão, a migração controlada, por qualquer política, necessita de apenas dois segundos, em média, entre a desconexão e a conexão com um novo ponto de acesso. Além desse tempo, de forma semelhante ao caso anterior, e para todas as políticas, existe ainda a espera pelo restabelecimento das conexões de controle e de vídeo.

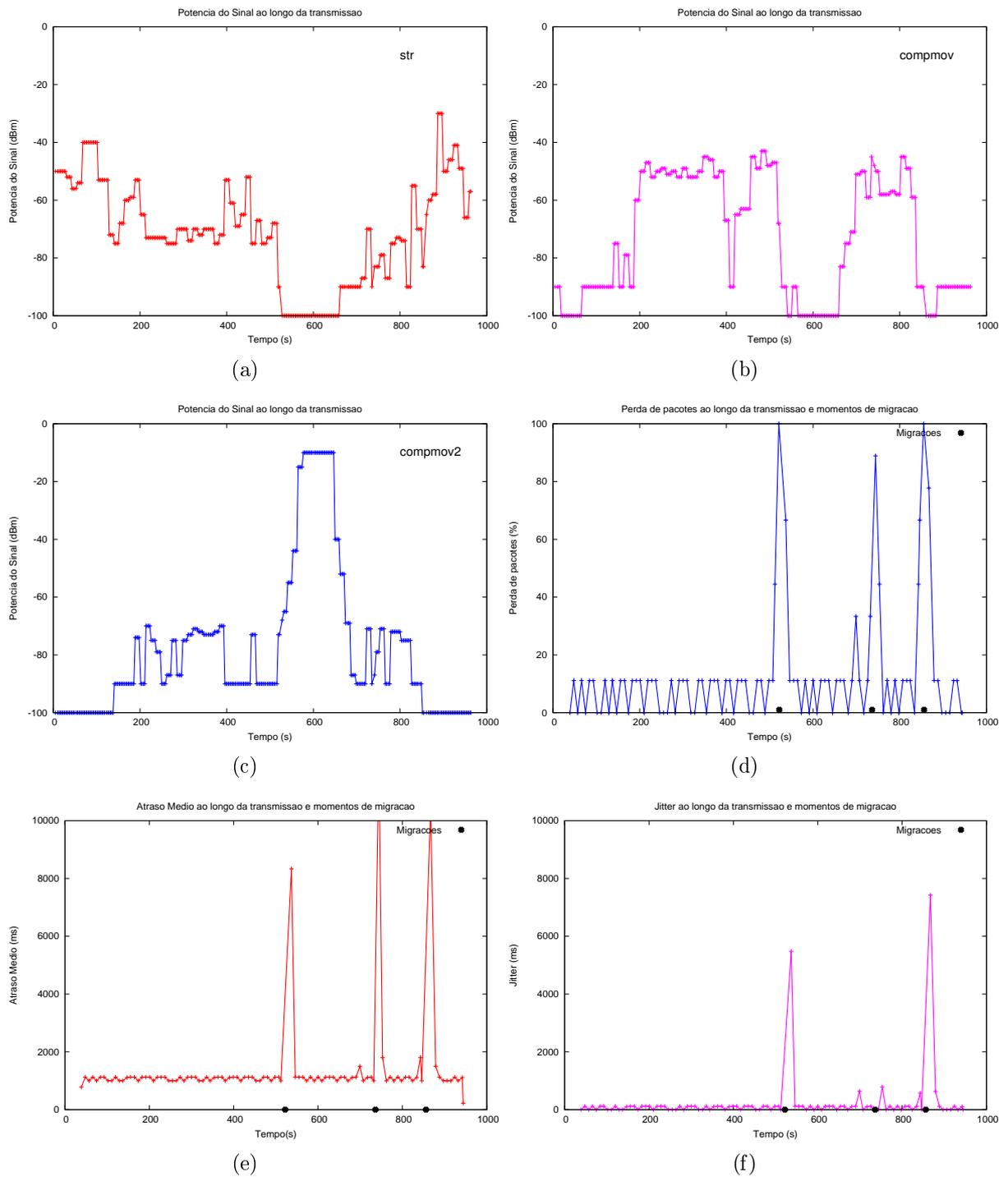


Figura 6.3: Sem política de migração e grande sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

As figuras 6.4(a), 6.4(b) e 6.4(c) apresentam a variação da potência dos sinais ao longo do experimento. A Figura 6.4(d), indica as diversas migrações ocorridas nesse cenário. O fato da política buscar o ponto de acesso com a melhor média, apenas, aumenta o

número de migrações, como pode ser visto no gráfico, entre os instantes 430s e 474s. Nesse intervalo ocorrem quatro migrações, entre os pontos de acesso **str** e **compmov**. Isso se dá devido ao momento do teste em que o cliente se desloca da sua posição próxima ao **compmov** em direção ao **str** e, rapidamente, em menos de um minuto, retorna à posição inicial.

Outro detalhe na Figura 6.4(d), está no instante 557s, onde ocorre uma desconexão abrupta antes da migração do ponto de acesso **compmov** para o **compmov2**. Essa queda justifica o alto valor da perda de pacotes nesse momento. Por fim, no instante 855s, logo após a migração do **compmov** para o **str**, que ocorreu no instante 841s, o sistema faz uma tentativa de migração, fracassada, de retorno ao **compmov**. Ou seja, por um momento a leitura do sinal do **compmov** esteve melhor do que a do **str**, como pode ser visto na Figura 6.4(b), e, por isso, o aplicativo tentou um retorno. Contudo, observando a mesma figura, pode-se perceber que durante a tentativa de conexão houve uma perda inesperada do sinal do **compmov**. Essa perda ocasionou uma espera muito grande pela nova conexão, ultrapassando o tempo máximo (configurado para os testes em 5s) e, assim, obrigou o aplicativo a buscar, imediatamente, um outro ponto de acesso, no caso, o **str**. Esse fracasso na tentativa de conexão com o **compmov**, a espera dos 5s para, somente então, iniciar a busca por outro ponto de acesso e a nova conexão, ocasionou uma um tempo de desconexão na ordem de 11s, explicando a alta perda de pacotes nesse momento.

Na Figura 6.4(e) pode-se encontrar o atraso médio ao longo da transmissão. Verifica-se que os valores mais altos ocorrem justamente nos momentos de migração. Contudo, é importante ressaltar que, no único momento de desconexão abrupta, no instante 557s, o atraso alcança um valor bastante alto. Essa diferença demonstra a importância da ação de migração controlada, evitando assim a ocorrência de uma desconexão em um futuro próximo.

Por fim, a Figura 6.4(f) mostra a variação do atraso e, mais uma vez, no momento da desconexão, devido um grande atraso de diversos pacotes, o valor do *jitter* se reduz.

6.4.1.3 Política de Migração - Média Simples com Limite Inicial

O objetivo é aplicar essa política somente quando a média da potência do sinal for inferior a um determinado valor. Após alguns testes pode-se definir como melhor limite o valor de -85 dBm e, observou-se também que esse baixo valor só é possível ser utilizado devido a sobreposição dos sinais. A explicação para isso está na grande probabilidade de existir um outro ponto de acesso com uma alta potência do sinal todas as vezes que o limite

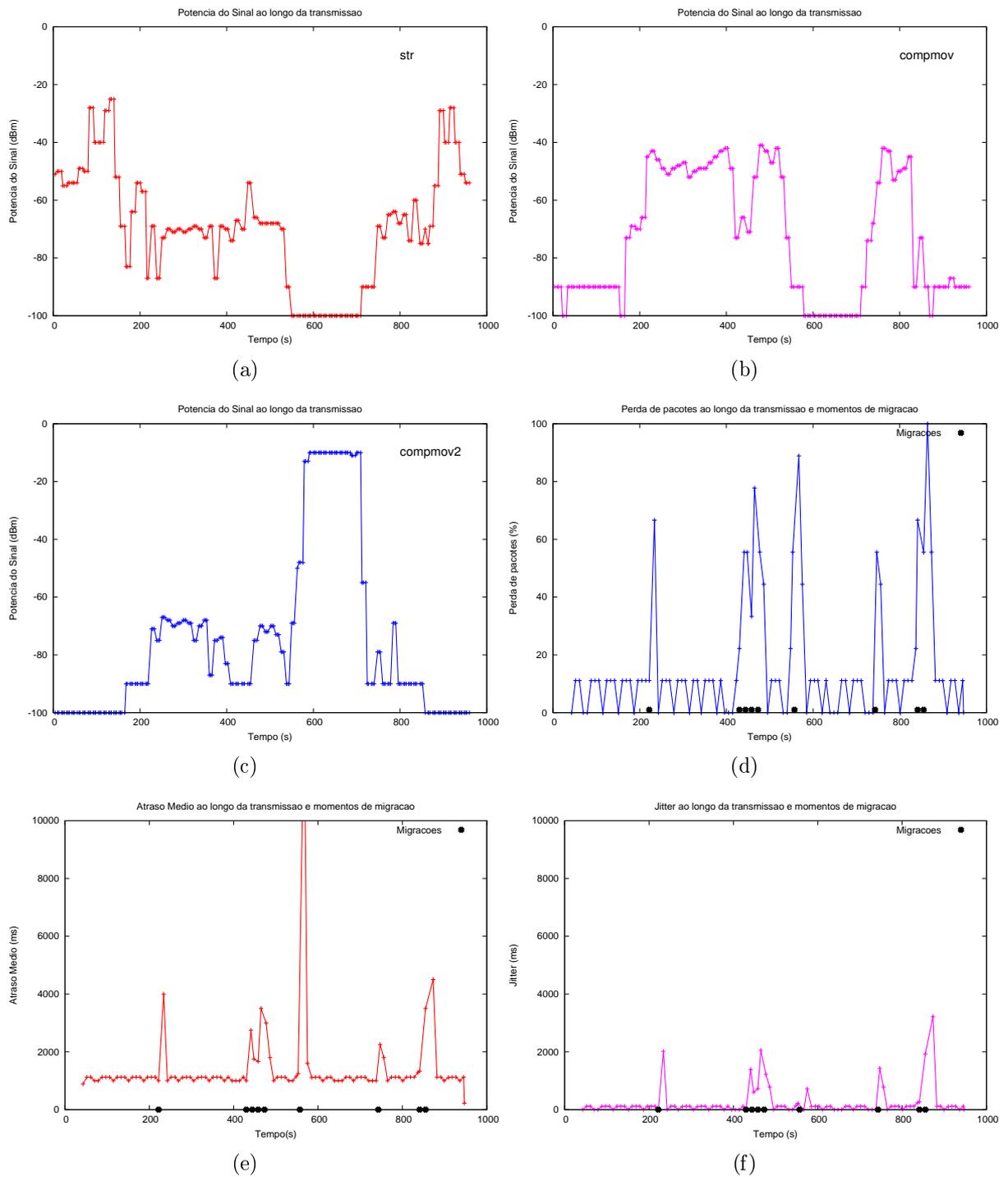


Figura 6.4: Política de migração Média Simples e grande sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

for estabelecido, caso contrário, seria grande a possibilidade de ocorrer uma desconexão. Com isso, nesse cenário, quase sempre, a migração ocorrerá do ponto de acesso com um sinal bastante deteriorado para um outro com uma qualidade melhor do sinal.

Nos gráficos das figuras 6.5(a), 6.5(b) e 6.5(c) pode-se visualizar os diferentes valores das potências dos sinais recebidos, ao longo da transmissão, para cada ponto de acesso. O gráfico da Figura 6.5(d) mostra a perda de pacotes ao longo da transmissão. É importante ressaltar que, como dito anteriormente, são analisados grupos de nove pacotes a cada momento. Assim, comparando os instantes 477s, 552s e 772s desse gráfico (momentos que ocorrem migrações controladas) com os instantes 522s, 736s e 856s da Figura 6.3(d) (momentos de desconexões abruptas), pode-se perceber que, a execução de uma política de migração reduz a perda de pacotes.

Ainda na Figura 6.5(d), pode-se destacar o instante 877s. Nesse momento seria desejável que ocorresse uma migração controlada do ponto de acesso **compmov** para o **str**. Mas, como mostrado na Figura 6.5(b), nesse instante nenhum sinal advindo do roteador **compmov** pode ser capturado, ocasionando uma desconexão inesperada. Essa desconexão justifica o alto valor na perda de pacotes, uma vez houve um total de sete segundos de desconexão, e não apenas os dois segundos necessários para a realização de uma migração controlada.

A Figura 6.5(e) mostra o atraso médio dos pacotes ao longo da transmissão, que permanece, de forma correta, na maioria do tempo, em torno de um segundo. Nesse gráfico pode-se perceber, nos instantes 477s, 552s e 772s em que ocorrem migrações controladas, uma redução considerável do atraso médio dos pacotes quando esses valores são comparados aos dos instantes 522s, 736s e 856s da Figura 6.3(e) em que ocorrem apenas desconexões inesperadas. Essa comparação comprova novamente a eficiência da mudança de ponto de acesso controlada por uma política de migração em relação ao uso de nenhuma política.

Por fim, o gráfico 6.5(f) mostra a variação do atraso dos pacotes ao longo da transmissão. Também nesse gráfico as distorções ocorrem nos momentos de migração, e a maior delas ocorre justamente no instante 877s, onde ocorreu uma perda inesperada da conexão, ao invés da migração controlada.

6.4.1.4 Política de Migração - Diferentes Níveis

O objetivo dessa política foi a tentativa de redução da quantidade de migrações realizadas, uma vez que o ponto de acesso de destino deve possuir a potência do sinal em uma faixa superior a faixa em que o atual roteador se encontra. Aproveitou-se a fase de testes para a definição dos limiares de diferentes níveis de qualidade do sinal.

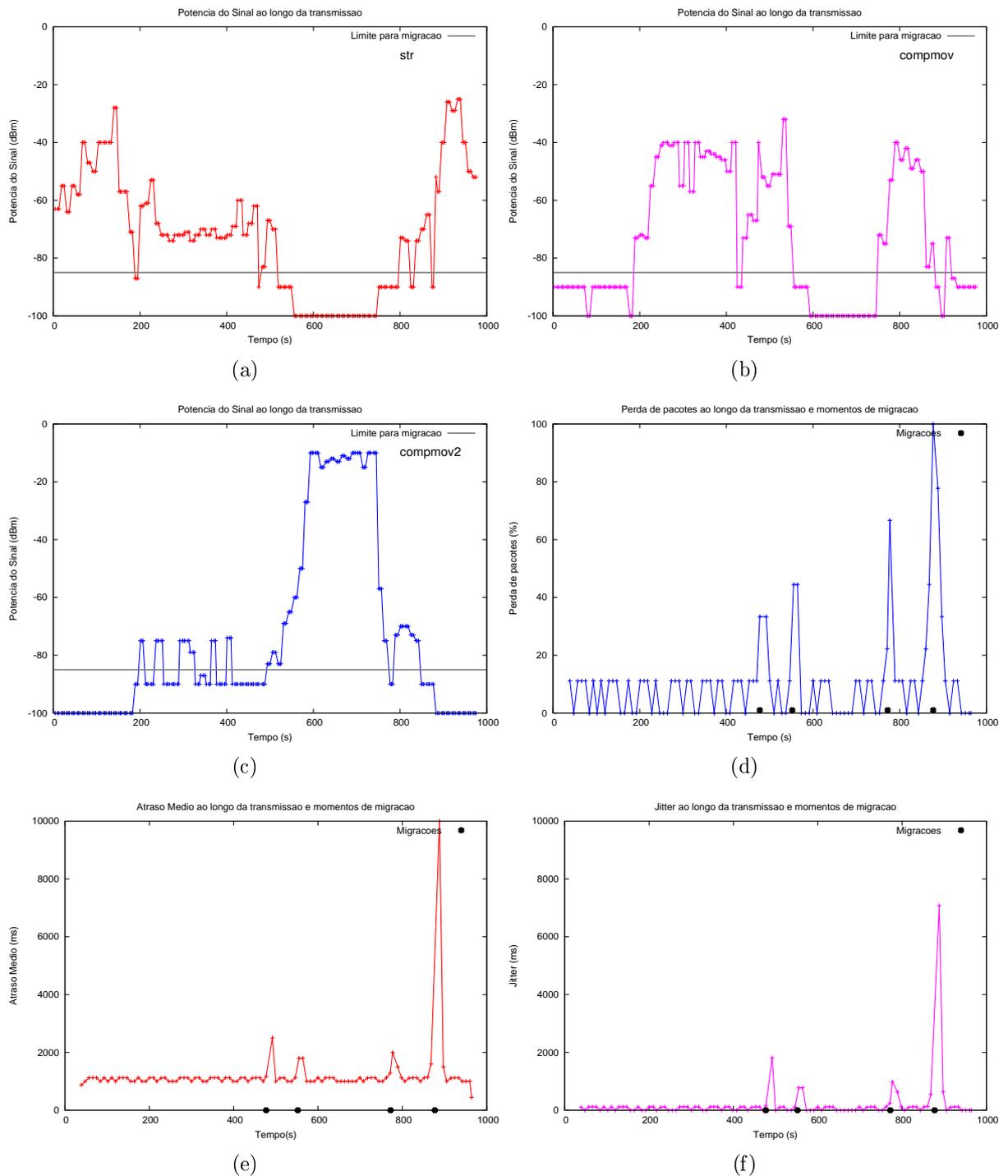


Figura 6.5: Política de migração Média Simples com Limite Inicial e grande sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

Os diferentes níveis utilizados, e todas as potências dos sinais de cada roteador, capturadas ao longo da transmissão, são mostradas através dos gráficos das figuras 6.6(a), 6.6(b) e 6.6(c). Nesse teste ocorreu, por volta do instante 200s, um fato interessante.

Embora o sinal no ponto de acesso *str* tenha alcançado a faixa 4 de valores nenhuma migração foi realizada. Isso ocorreu devido a inexistência de um roteador que estivesse com a qualidade do sinal em um nível melhor no ponto de recepção.

A Figura 6.6(d), mostra a perda de pacotes ao longo da transmissão. Embora nesse teste não tenha ocorrido uma desconexão inesperada, existe uma perda de pacotes relativamente alta nos instantes 562s, 770s e 906s onde ocorrem migrações controladas. A explicação para esses altos valores é a realização de migrações para pontos de acesso cujo a qualidade do sinal não está alta o suficiente para restabelecer as conexões de forma rápida. Essa demora no restabelecimento das conexões acarreta uma perda maior dos pacotes. Ou seja, a conexão com um ponto de acesso cujo sinal seja inferior a -50 dBm ocasiona uma lentidão no restabelecimento das conexões da aplicação, o que justifica uma maior perda dos pacotes. Além dessa lentidão é importante ressaltar a atuação do mecanismo de ARC que, pela existência de uma única conexão cujo sinal está degradado ocasiona a redução na taxa de transmissão dos pontos de acesso.

As figuras 6.6(e) e 6.6(f) representam o atraso médio e a variação do atraso dos pacotes, respectivamente. De forma similar aos testes para esse cenário, as maiores variações ocorrem nos momentos de migração. Na maior parte do tempo pode-se perceber uma variação de um segundo para o atraso médio e, uma variação em torno de zero segundo para o *jitter*, valores dentro do previsto.

Com a execução desses testes e a comparação dos resultados com os da política anterior pode-se concluir que as duas políticas produzem resultados aproximados. Assim, através de uma análise detalhada observa-se que a definição dos diferentes níveis se torna ineficaz nesse cenário, onde a alta sobreposição do sinal pode ocasionar migrações desnecessárias. Ou seja, mesmo que um ponto de acesso esteja em um nível melhor, com a existência da alta sobreposição, essa condição favorável pode deixar de ser verdadeira em um curto espaço de tempo. Também, como existe uma baixa perda de pacotes, devido a baixa taxa de transferência utilizada, mesmo que a atual qualidade do sinal seja inferior a de um outro ponto de acesso, mas, não esteja prejudicando a transmissão, a migração pode ser omitida.

6.4.2 Disposição dos Pontos de Acesso com Baixa Sobreposição dos Sinais

A característica desse cenário é exatamente a oposta em relação ao cenário anterior. Aqui a migração, quando ocorre, faz o cliente sair de um ponto de acesso cuja potência do

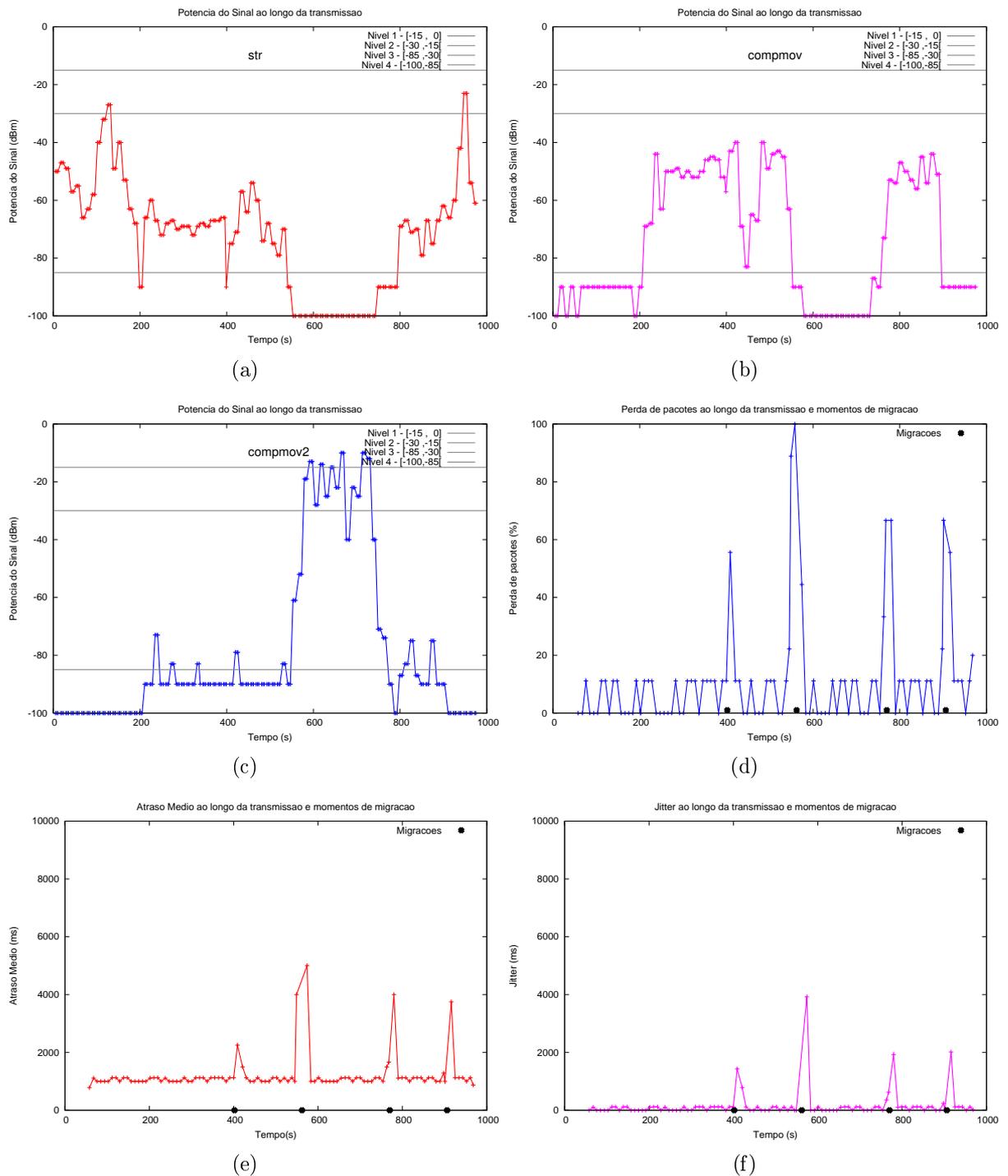


Figura 6.6: Política de migração Diferentes Níveis e grande sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

signal received is low, for another access point with the signal power also low. This observation is fundamental for the analysis of the results, once that the low reception quality has a direct consequence on the transmission capacity, as stated

na subseção 6.4.1, e o restabelecimento das conexões de controle e de vídeo do aplicativo se mostrou mais lento nessas condições. Para demonstrar esse efeito, faremos algumas comparações com o cenário anterior.

6.4.2.1 Sem Política de Migração

Como já mencionado, essa política faz com que nenhum processo de migração seja utilizado, ocasionando desconexões todas as vezes que o sinal do ponto de acesso deixa de ser recebido. Com isso, um processo de busca dos sinais de todos os roteadores alcançáveis é iniciado imediatamente após a captura do evento informando sobre a desconexão. Depois da busca, uma nova conexão se inicia, automaticamente, com o ponto de acesso de melhor sinal.

Durante esse teste ocorreram quatro desconexões, nos instantes 218s, 551s, 790s e 876s. Essas desconexões representam, respectivamente, as seguintes mudanças de ponto de acesso: de **str** para **compmov**, depois para **compmov2**, um retorno a **compmov** e, por fim, de volta a **str**.

Os gráficos nas figuras 6.7(a), 6.7(b) e 6.7(c), representam as qualidades dos sinais de cada ponto de acesso ao longo do experimento. A Figura 6.7(d), representa a perda de pacotes. Existem algumas diferenças importantes, quando comparado ao gráfico da Figura 6.3(d), que devem ser destacadas. No gráfico da Figura 6.7(d) ocorrem quatro desconexões e no do cenário anterior ocorreram apenas três. Isso pode ser explicado pelo diferente nível de sobreposição dos sinais nos dois cenários. Quando existia uma grande sobreposição, a primeira queda abrupta e a nova conexão ocorreram do ponto de acesso **str** para **compmov2** diretamente, evitando a passagem por **compmov** ocorrida aqui. Outra diferença importante está na grande perda de pacotes em cada desconexão, que ocorre devido a nova conexão ser estabelecida com um ponto de acesso que ainda está com a potência do sinal relativamente baixa, aumentando a perda de pacotes.

Ao comparar as figuras 6.7(e) e 6.7(f) com os gráficos nas figuras 6.3(e) e 6.3(f), pode-se observar uma maior variação do atraso médio e do *jitter* no atual cenário em relação ao anterior, em todos os momentos de desconexão. A razão para isso é a mesma apresentada no parágrafo anterior: a mudança de ponto de acesso ocorre de um sinal fraco para outro também fraco.

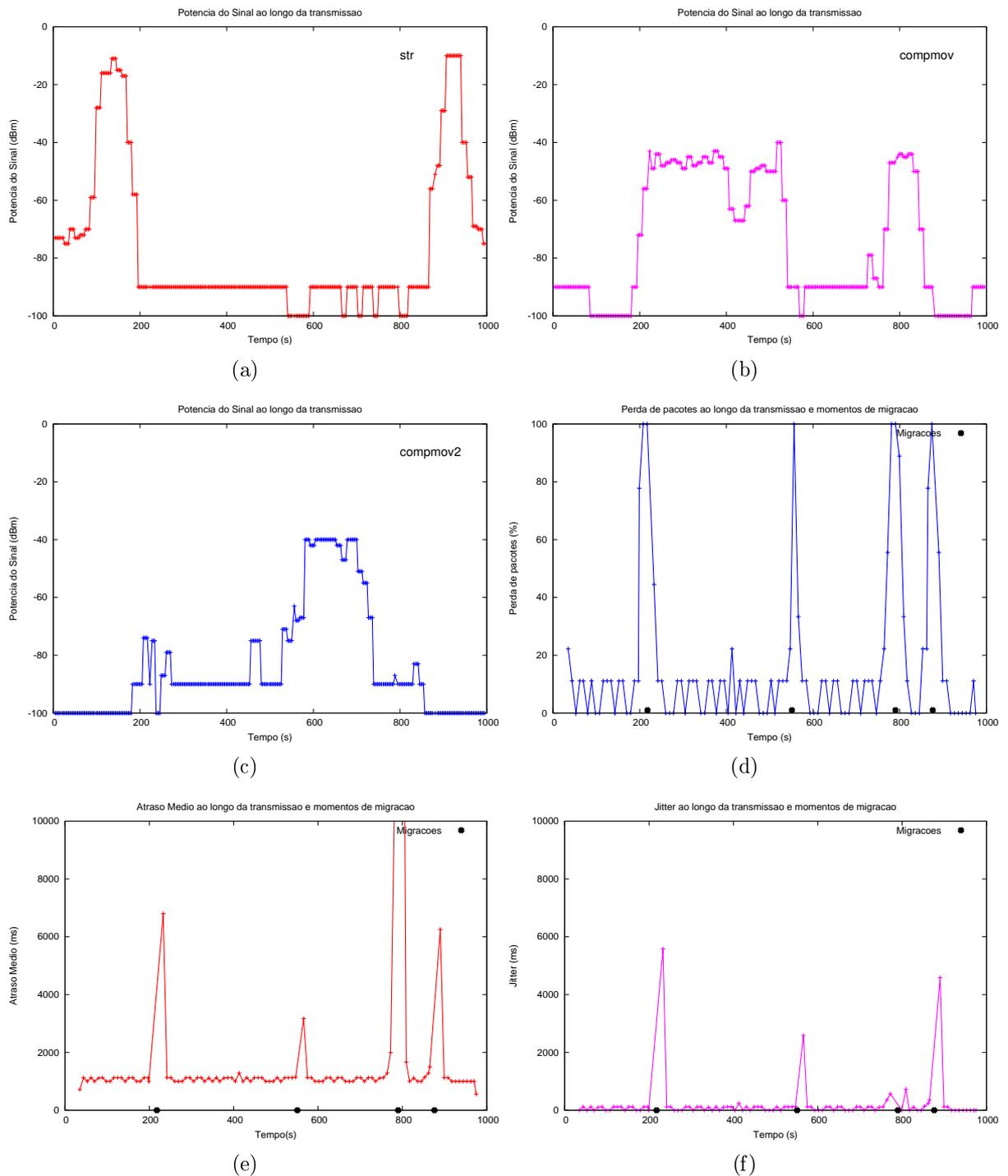


Figura 6.7: Sem política de migração e baixa sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

6.4.2.2 Política de Migração - Média Simples

Essa política, mais uma vez, gerou muitas migrações, o que impossibilita seu uso na prática. Contudo, esse teste permite a percepção de uma redução considerável na perda

de pacotes, quando realiza-se uma migração controlada e impede-se uma queda inesperada da conexão.

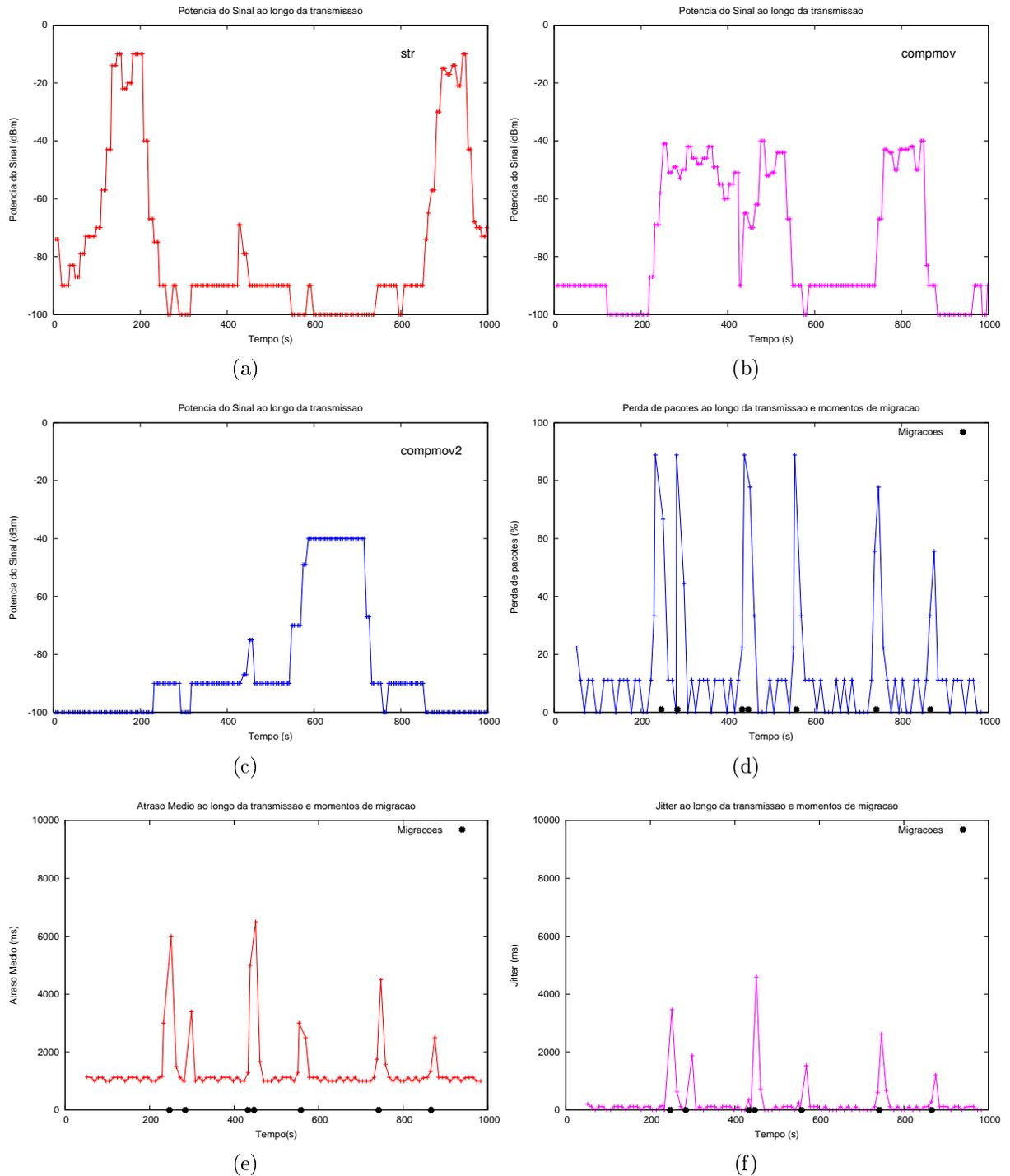


Figura 6.8: Política de migração Média Simples e baixa sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

As figuras 6.8(a), 6.8(b) e 6.8(c) apresentam o valor dos sinais de cada ponto de acesso

durante todo o experimento. No gráfico da Figura 6.8(d), são representados a perda de pacotes e os momentos de cada migração. Ao comparar esse gráfico com o da Figura 6.7(d) pode-se perceber uma considerável redução na perda de pacotes. É importante ressaltar que cada instante no gráfico representa um grupo de nove pacotes. Assim, nos instantes 218s e 790s da Figura 6.7(d), onde mostra perda de 100% dos pacotes, pode-se afirmar que, durante a desconexão houve perda de mais de 18 pacotes. No instante 284s, logo após a migração do ponto de acesso **str** para o **compmov**, ocorreu uma queda inesperada da conexão, causada por uma interrupção não identificada. Essa queda não registrou uma perda muito maior dos pacotes devido ao alto valor da potência do sinal no roteador **compmov**. Esse alto valor fez com que o restabelecimento das conexões de controle e de vídeo do sistema ocorresse de forma mais rápida.

Nas figuras 6.8(e) e 6.8(f), o atraso médio e o *jitter* sofreram uma redução considerável nos momentos de migração, quando comparados aos testes sem política de migração, representados pelos gráficos das figuras 6.7(e) e 6.7(f).

6.4.2.3 Política de Migração - Média Simples com Limite Inicial

Nesse cenário, onde os sinais não sofrem muita sobreposição, o limite inicial para aplicar a política de migração teve de ser alterado. Como resultado de exaustivos testes, constatou-se que aqui o limite de -85 dBm gera um número maior de quedas na conexão. Isso é facilmente explicável, pois sem uma grande sobreposição, esse valor estava sendo alcançado antes mesmo que existisse algum outro ponto de acesso com uma melhor potência do sinal. Por isso, o limite inicial foi alterado para -80 dBm.

As figuras 6.9(a), 6.9(b) e 6.9(c) representam a variação da potência do sinal de cada ponto de acesso ao longo de todo o experimento. O gráfico da Figura 6.9(d) representa a variação na taxa de perda de pacote ao longo do tempo. Quando comparado com o gráfico dessa mesma política, no cenário anterior (Figura 6.5(d)), pode-se perceber um aumento na taxa de perdas durante cada migração, sendo isso explicado, novamente, pelo fato de cada migração conduzir a um ponto de acesso com um sinal melhor do que o atual, mas também debilitado. Dois momentos importantes nesse teste são os instantes 515s e 701s. No primeiro, ocorre uma desconexão inesperada, logo após a migração para o ponto de acesso **compmov2**, de forma similar ao que ocorre no instante 284s do teste anterior. No segundo momento ocorre um fato interessante, pois retornando do roteador **compmov2** em direção ao **compmov** o processo de migração identifica o melhor ponto de acesso como sendo o **str**. Analisando as figuras 6.9(a) e 6.9(b), pode-se perceber que nesse momento os

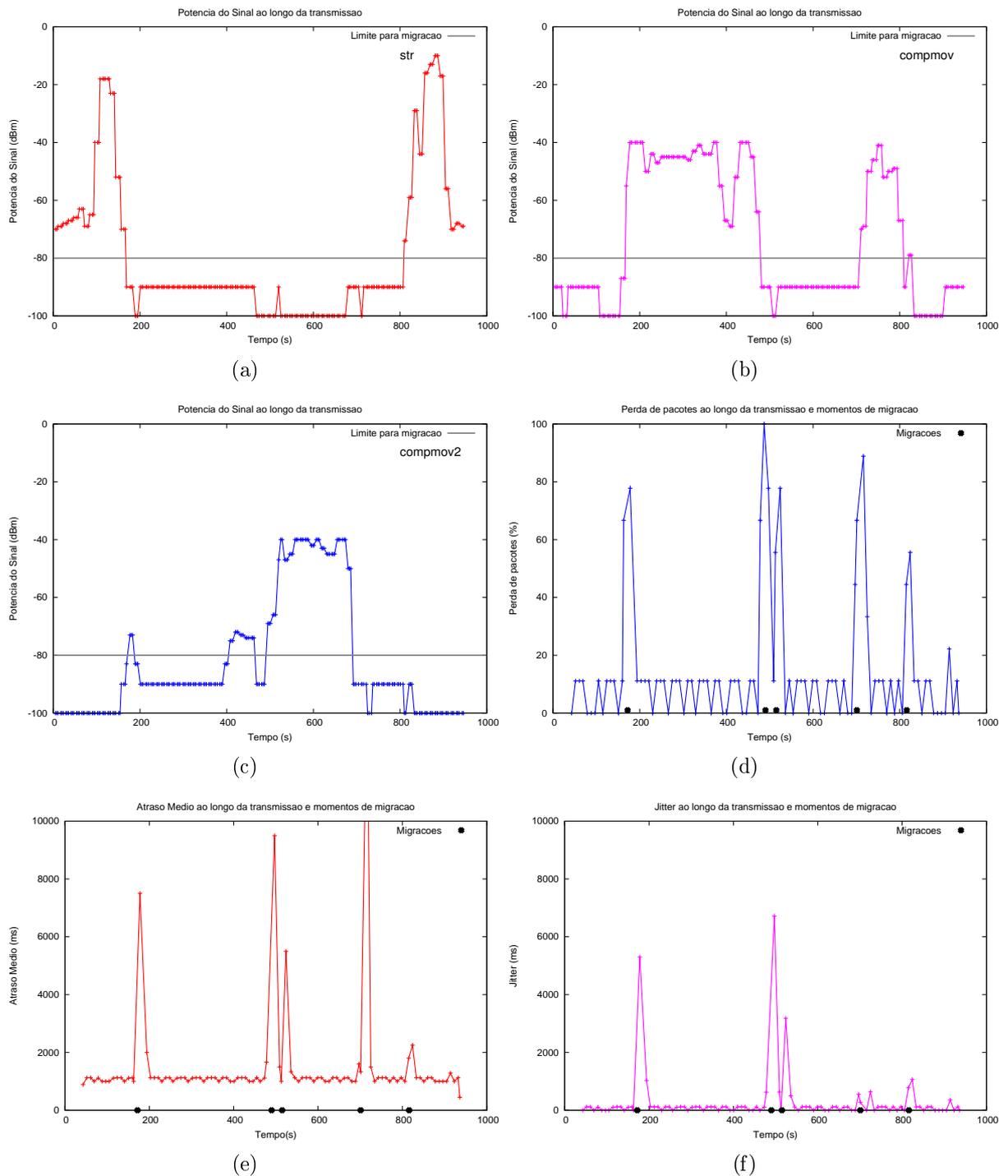


Figura 6.9: Política de migração Média Simples com Limite Inicial e baixa sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

dois roteadores, **compmov** e **str** estavam com a potência do sinal baixa, em -90 dBm, o que levou o algoritmo a decidir pelo **str**. Porém, logo após essa decisão o ponto de acesso **str** deixou de ser alcançável, ativando o recurso de *timeout* da aplicação. Esse recurso realiza

uma nova busca por um novo ponto de acesso todas as vezes que o processo de conexão não finaliza em um tempo máximo de cinco segundos. Assim, o processo de migração que iniciou no instante 701s, com o **str** como destino, terminou apenas no instante 713s, com a conexão no **compmov**.

A Figura 6.9(e), representa o atraso médio durante toda a transmissão. A maior parte do tempo esse atraso se mantém na faixa de um segundo, o que era esperado, uma vez que a taxa de transferência do vídeo é de um quadro por segundo. O maior valor é alcançado justamente no instante 701s, onde ocorre a tentativa de migração, fracassada, com o roteador **str**.

A Figura 6.9(f), representa a variação do atraso ao longo do tempo. A grande perda de pacotes, como a que ocorre no instante 701, pode justificar um *jitter* reduzido uma vez que, a recepção de vários pacotes com um grande atraso produz uma baixa variação.

A comparação desses resultados com aqueles produzidos com essa mesma política, mas na no caso de uma maior sobreposição dos sinais, permite perceber que o cenário anterior produz resultados melhores. Sendo assim, pode-se concluir claramente que a distribuição dos roteadores tem uma influência direta na qualidade do serviço e sugere uma nova definição para os valores de parâmetros importantes, como o limite inicial dessa política.

6.4.2.4 Política de Migração - Diferentes Níveis

Essa política se mostrou, mais uma vez, tão eficiente quanto a política anterior, porque a criação de diferentes níveis para a qualidade dos sinais, determina, sempre, principalmente nesse cenário, o roteador com o melhor sinal como sendo a melhor opção.

As figuras 6.10(a), 6.10(b) e 6.10(c) representam as potências dos sinais de cada ponto de acesso ao longo do experimento. Os altos valores da taxa de perda de pacotes nos momentos de migração, como pode ser visto na Figura 6.10(d), são explicáveis da mesma forma que nos testes anterior. Nesse experimento também ocorreu uma desconexão, por algum motivo não identificado, no instante 587s, e uma tentativa de conexão com o ponto de acesso **compmov** que ultrapassou o tempo limite de cinco segundos, provocando uma conexão de urgência, que custou um período de onze segundos de desconexão (do instante 687s ao 698s).

As figuras 6.10(e) e 6.10(f) mostram, respectivamente, o atraso médio e o *jitter*. Novamente a variação do atraso médio e do *jitter* se mantiveram, na maior parte do tempo,

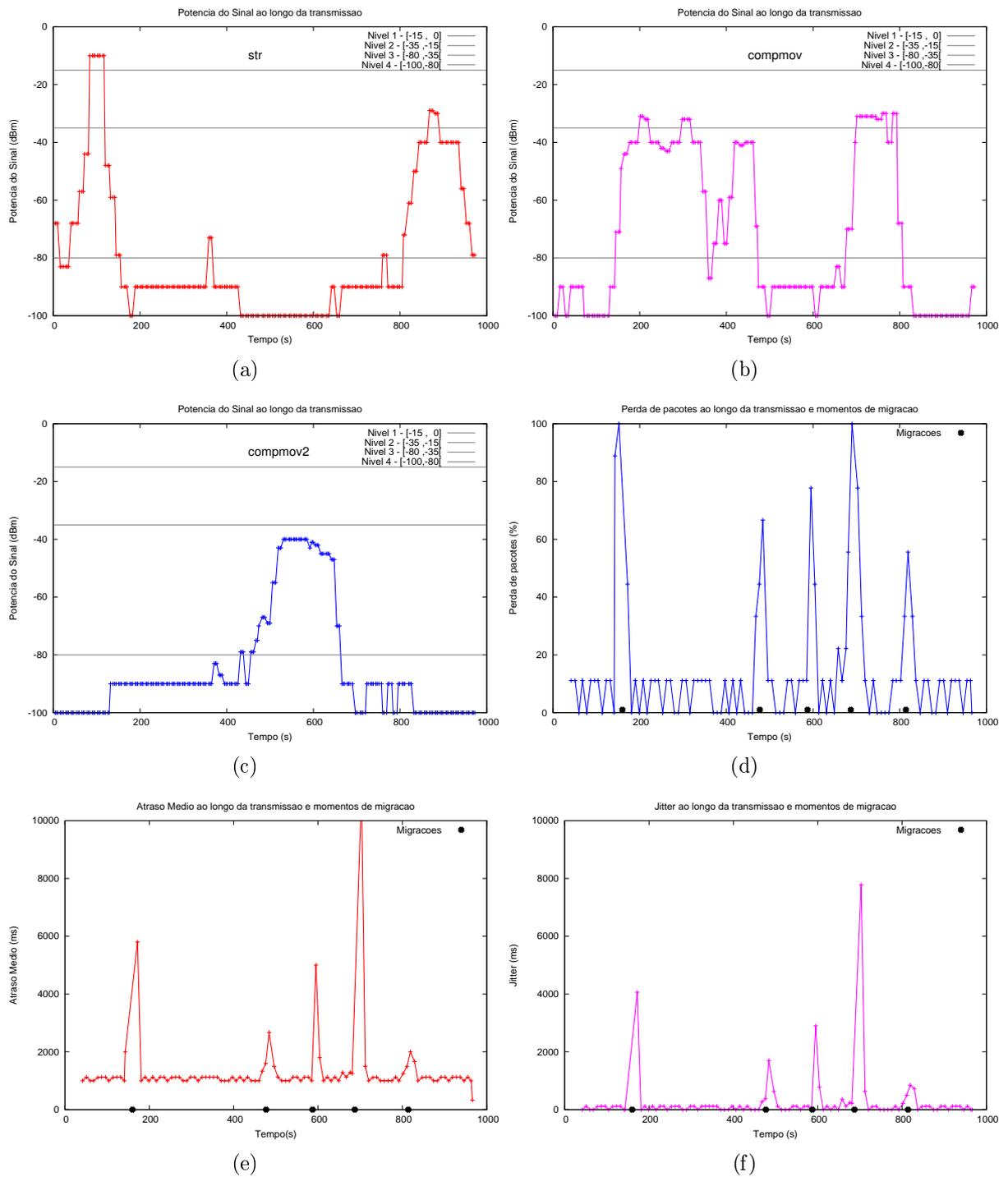


Figura 6.10: Política de migração Diferentes Níveis e baixa sobreposição dos sinais: (a) Ponto de acesso **str**; (b) Ponto de acesso **compmov**; (c) Ponto de acesso **compmov2**; (d) Perda de pacotes; (e) Atraso médio entre os pacotes; (f) Variação do atraso.

nos padrões esperados de um e zero segundo, respectivamente. As maiores variações ocorreram nos momentos de migração controlada, o que já era esperado.

As tabelas 6.1 e 6.2 apresentam uma comparação dos resultados referentes aos cená-

rios com grande sobreposição e com reduzida sobreposição respectivamente. É importante ressaltar que os experimentos em cada cenário não foram realizados nas mesmas circunstâncias, já que o ambiente sem fio está sujeito a diferentes interferências. Porém, as tabelas a seguir podem oferecer uma noção do desempenho das diferentes políticas propostas em cada cenário.

	Sem Migração	Média Simples	Com Limite Inicial	Diferentes Níveis
Perda (%)	12,85	15,69	10,47	11,20
<i>Jitter</i> (ms)	441	250	201	382

Tabela 6.1: Com grande sobreposição dos sinais

	Sem Migração	Média Simples	Com Limite Inicial	Diferentes Níveis
Perda (%)	16,14	14,42	14,44	14,31
<i>Jitter</i> (ms)	491	497	261	272

Tabela 6.2: Praticamente sem sobreposição dos sinais

6.5 Conclusão

Diante dos resultados, pode-se concluir que a política de migração que utiliza a média e um limite inicial, mesmo simples, se mostrou eficiente nos experimentos realizados. Com uma simples modificação do valor inicial, de acordo com o nível de sobreposição do sinal, é possível realizar apenas as migrações estritamente necessárias e evitar queda de conexão por perda total do sinal.

Como esse experimento se baseia em ambientes hospitalares, onde a disposição dos roteadores é completamente controlável, e limita-se a uma baixa taxa de transferência, a idéia de misturar a taxa de perda de pacotes com a qualidade do sinal, em uma quarta política, utilizando também diferentes níveis para a potência do sinal, foi completamente descartada antes mesmo de ser testada. Contudo, essa idéia poderá produzir bons resultados se o cliente deixar de ser um dispositivo limitado em processamento.

Com esses testes foi possível definir alguns parâmetros em cada política e, ao fim, alcançar um menor número de migrações. Essa redução se mostrou fundamental, uma vez que os dispositivos portáteis, tipo *Pocket PC*, só devem realizar esses procedimentos nos casos mais extremos. Migrações devem ser realizadas antes de ocorrer uma queda na conexão por perda do sinal, uma vez que o tempo para realizar uma nova conexão, nesse último caso, pode chegar a ser três vezes maior que em uma migração controlada.

Com isso, a implantação de adaptações na camada abaixo da de aplicação, ao mesmo tempo que se mostrou eficiente, preserva a individualidade de cada sistema. O uso da técnica de *buffers* proporcionou qualidade à transmissão e buscou oferecer continuidade à exibição do vídeo durante os momentos de desconexão gerados pela migração.

O envio de apenas um quadro por segundo, no caso do uso de dispositivos como os *Pocket PCs*, é consequência apenas de seu baixo poder de processamento. Mas é importante ressaltar que o sistema foi desenvolvido para transportar taxas maiores de vídeo - ou seja, quando se verificar avanço tecnológico nesses dispositivos, o sistema será capaz de fornecer uma qualidade maior aos vídeos.

Finalmente, do ponto de vista da implementação, é importante ressaltar que o uso do NDISUIO, responsável por fornecer uma interface entre a camada de aplicação e o NDIS, por meio do *DeviceIoControl*, reduz, de forma considerável, o tempo necessário para a realização das migrações.

Capítulo 7

Conclusão

7.1 Introdução

As redes sem fio IEEE 802.11 podem oferecer mobilidade, mas os problemas que lhes são peculiares forçam as aplicações a serem capazes de adaptarem-se às constantes mudanças de requisitos importantes da rede como, a redução da taxa de transferência em um ponto de acesso. Assim, garantir o máximo de qualidade fim-a-fim - principalmente na transferência de informações multimídia - tem sido o objetivo de diversos trabalhos. Para isso, alguns requisitos mínimos de QoS devem ser garantidos pela aplicação, como baixo *jitter*, pequenas perdas de pacotes e altas taxas de vazão.

Nas transferências multimídia, altas variações nos requisitos de QoS prejudicam a qualidade final da mídia. Para solucionar alguns desses problemas, como o *jitter*, por exemplo, podem ser empregadas técnicas de gerenciamento de *buffers*. Por isso, na tentativa de fornecer qualidade, e na percepção da existência de poucas implementações, este projeto buscou desenvolver e implementar técnicas de adaptação inseridas entre a camada de aplicação e a rede sem fio.

Dessa forma, neste projeto a migração entre pontos de acesso foi fornecida como uma possível adaptação para os momentos em que os requisitos mínimos de QoS não puderem ser mais alcançados. Embora essa técnica produza momentos de desconexão, que podem ser da ordem de segundos, pode-se perceber um aumento considerável na qualidade do vídeo com a implantação da adaptação proposta, através da redução na perda de pacotes.

Analisadas as diferentes políticas propostas, os resultados foram satisfatórios. Devido à capacidade de processamento limitada dos dispositivos móveis utilizados nos testes, e ao tempo de desconexão em cada migração, entre outros fatores, percebe-se que essa técnica

deve ser utilizada apenas em momentos extremos. Diante disso, o mecanismo de visualização da qualidade do sinal proposto neste projeto, que permite ao usuário identificar as áreas com baixa potência do sinal, possibilita que este decida se quer continuar se afastando do ponto de acesso ou não.

Embora os resultados sejam animadores, existem outras adaptações que, em conjunto com o que já foi implementado, podem produzir resultados ainda melhores. Por exemplo, a redução de quadros de vídeo transmitidos em momentos de congestionamento da rede e a reserva de recursos podem garantir uma maior qualidade da conexão, o que reduzirá a perda de pacotes.

Por fim, alguns problemas das redes sem fio são difíceis de serem resolvidos, como as interferências, provocadas por inúmeras razões. Por isso, nesses ambientes, são comuns a grande perda de pacotes e os altos valores de *jitter*. As adaptações buscam reduzir esses valores e são fundamentais para fornecer qualidade nas transmissões sobre redes sem fio.

7.2 Trabalhos Futuros

Como apresentado no capítulo 3, o mecanismo de ARQ tem como objetivo re-enviar um mesmo quadro várias vezes, caso a entrega não seja confirmada. Por isso, uma possível extensão do trabalho seria fazer com que o aplicativo determine o tamanho máximo para cada pacote transmitido, que pode ser a MTU de uma rede sem fio, a fim de diminuir o número de retransmissões e, assim, diminuir o volume de tráfego na rede.

O acesso ao *driver* da câmera de vídeo com o objetivo de reduzir a taxa de captura é um outro trabalho que necessita ser realizado. Esse controle do dispositivo pode facilitar um possível mecanismo de adaptação do aplicativo, que tenha como objetivo a redução automática da taxa de captura todas as vezes que um congestionamento na rede for detectado.

Outro trabalho futuro é fornecer ao sistema a capacidade de codificar o vídeo capturado por meio de outros padrões, como H263+ ou H264. Como descrito no capítulo 3, os quadros que em seqüência formam um vídeo podem ser classificados em dois tipos: aqueles que são codificados de forma independente e os quadros estimados. O código aqui utilizado para o padrão H263 não implementa essas classificações e, por isso, não foi possível realizar adaptações diretamente na geração e recepção do vídeo. Ao se utilizarem outros padrões, nos momentos de congestionamento da rede sem fio, pode-se optar por enviar um número menor de quadros, o que reduz o tráfego sem diminuir significativamente a

qualidade do vídeo.

Com o objetivo de manter a conectividade sem a necessidade de gerenciar migrações entre pontos de acesso, a utilização de uma rede *Mesh* - redes sem fio comunitárias de acesso banda larga - pode ser interessante. Para isso, é preciso que todos os roteadores e os participantes com mobilidade utilizem algum protocolo de roteamento *Ad Hoc*. No caso deste projeto, onde foi feita a opção pela utilização de dispositivos de pequeno porte, infelizmente, não existiam ainda implementações de algum desses protocolos voltadas para esse tipo de dispositivo. No entanto, no Instituto de Computação da Universidade Federal Fluminense, a aluna de mestrado Luciana E. N. Hilario realizou testes de transmissão de dados multimídia utilizando a rede *Mesh* do Instituto e um *notebook*. Os resultados dos testes mostram que a conectividade pode ser garantida. Porém, como todos os participantes móveis da rede necessitam executar um protocolo de roteamento *Ad Hoc*, mesmo que existisse uma versão de algum desses protocolos para *Pocket PC*, a limitação da capacidade de processamento dificultaria testes com os dispositivos.

Por fim, devido ao mecanismo ARC das redes sem fio, o ponto de acesso reduz a sua taxa de transmissão ao detectar a queda na qualidade da conexão com algum dispositivo móvel a ele conectado. Assim, um único dispositivo pode prejudicar a transmissão de todos os outros. Para diminuir os efeitos dessa redução, pode-se reservar fatias de banda do roteador para diferentes fluxos. Por exemplo, pode-se fazer uma reserva para as mensagens de vídeo e outra para diferentes tipos de pacotes. Para isso, será necessário alterar o *firmware* dos roteadores e inserir filas de prioridades para proporcionar uma reserva de banda. Embora essa alteração elimine a proposta de se utilizar apenas dispositivos *off-the-shelf*, a implantação dessa proposta é possível, pois o ambiente aqui proposto caracteriza-se por ser completamente controlado (proprietário).

Referências

- [Aurrecoechea et al. 1998] Aurrecoechea, C.; Campbell, A. T. e Hauw, L. **A survey of QoS architectures**. *Multimedia Systems*, v. 6, n. 3, p. 138–151, 1998.
- [Balachandran e Voelker 2002] Balachandran, A. e Voelker, G. M. **The wireless research API (WRAPI) – Real-time monitoring and control of a wireless LAN**. University of California San Diego, Dezembro, 2002. Disponível em: <<http://sysnet.ucsd.edu/pawn/wrapi/>>.
- [Chalmers e Sloman 1999] Chalmers, D. e Sloman, M. **A survey of quality of service in mobile computing environments**. *IEEE Communications Surveys and Tutorials*, IEEE CS, v. 2, n. 2, April 1999. <Http://www.comsoc.org/livepubs/surveys/public/2q99issue/2q99-lead.html>. Disponível em: <<http://pubs.doc.ic.ac.uk/Mobile-QoS-survey/>>.
- [Chandramouli 2004] Chandramouli, V. **A detailed study on wireless LAN Technologies**. The University of Texas at Arlington, Janeiro 2004, Disponível em: <http://crystal.uta.edu/~kumar/cse6392/termpapers/Vijay_paper.pdf>.
- [da Conceição e Kon 2006a]da Conceição, A. F. e Kon, F. **Desenvolvimento de aplicações adaptativas para redes IEEE 802.11**. In: *Simpósio Brasileiro de Redes de Computadores*, Curitiba, Paraná, Brasil, v. 1, p. 329–344, Maio, 2006.
- [da Conceição e Kon 2006b]da Conceição, A. F. e Kon, F. **O uso de pares de pacotes para monitoração de taxa de transmissão e da capacidade de vazão em redes IEEE 802.11**. In: *Simpósio Brasileiro de Redes de Computadores*, Curitiba, Paraná, Brasil, v. 1, p. 345–358, Maio, 2006.
- [Dutta et al. 2003] Dutta, A.; Madhanie, S.; Chen, W.; Altintas, O. e Schulzrinne, H. **MobiCom poster: Optimized fast-handoff schemes for application layer mobility management**. *SIGMOBILE Mob. Comput. Commun. Rev.*, v. 7, n. 1, p. 17–19, Janeiro 2003.
- [Favela et al. 2003] Favela, J.; Muñoz, M. A.; Rodríguez, M.; Martínez-García, A. I. e González, V. M. **Context-aware mobile communication in hospitals**. *Computer*, v. 36, n. 9, p. 38–46, 2003.
- [Favela et al. 2004] Favela, J.; Muñoz, M. A.; Rodríguez, M. D. e Martínez, E. A. **Location-aware access to hospital information and services**. *IEEE Transactions on Information Technology in Biomedicine*, v. 8, n. 4, p. 448–455, 2004.
- [Isovié e Fohler 2004] Isovié, D. e Fohler, G. **Quality aware MPEG-2 stream adaptation in resource constrained system**. In: *16th Euromicro Conference on Real-time Systems (ECRTS 04)*, Catania, Sicily, Italy, p. 23–32, Julho, 2004. Disponível em: <<http://www.mrtc.mdh.se/index.php?choice=publications&id=0679>>.

- [Laksono 2004] Laksono, I. **Achieve end-to-end QoS for wireless video streaming**. *Electronic Engineering Times*, Novembro 2004. Disponível em: <www.eetasia.com/ARTICLES/2004NOV/B/2004NOV16_RFT_QA_TA.pdf>.
- [Mahajan et al. 2002] Mahajan, A.; Mundur, P. e Joshi, A. **Adaptive Multimedia System Architecture for Improving QoS in Wireless Networks**. In: *PCM '02: Proceedings of the Third IEEE Pacific Rim Conference on Multimedia*, Hsinchu, Taiwan, p. 712–719, Dezembro, 2002.
- [Microsoft 2004] Microsoft. **Quality windows audio-video experience - qWave**. Abril 2004. Disponível em: <<http://www.microsoft.com/whdc/device/stream/qWave.mspx>>.
- [Pack e Choi 2002] Pack, S. e Choi, Y. **Pre-authenticated fast handoff in a public wireless LAN based on IEEE 802.1x model**. In: *PWC '02: Proceedings of the IFIP TC6/WG6.8 Working Conference on Personal Wireless Communications*, Cingapura, p. 175–182, Outubro, 2002.
- [Pengpeng 2003] Pengpeng, N. **Bandwidth estimation for adaptive multimedia over wireless networks**. Dissertação (Mestrado), Malardalen University Västeras Sweden, Novembro 2003.
- [Ramani e Savage 2005] Ramani, I. e Savage, S. **SyncScan: Practical fast handoff for 802.11 infrastructure networks**. In: *24th Conference on Computer Communications (INFOCOM)*, Miami, Flórida, EUA, v. 1, p. 675–684, Março, 2005.
- [Schiller 2003] Schiller, J. **Mobile communications**. 2003. Addison-Wesley.
- [Talekar 2004] Talekar, N. Y. **VideoNet: Peer to peer video conference application**. Junho 2004. Disponível em: <www.codeproject.com/internet/videonet.asp>.
- [Tan et al. 1999] Tan, C. L.; Lye, K. M. e Pink, S. **A fast handoff scheme for wireless networks**. In: *WOWMOM*, Seattle, Washington, EUA, p. 83–90, Agosto, 1999.
- [Watkinson 2003] Watkinson, J. **The MPEG handbook - MPEG-1, MPEG-2, MPEG-4**. 2003.

Apêndice A - Documentação Técnica do Sistema

A.1 Apresentação

Os dois principais aplicativos desenvolvidos neste experimento foram o código servidor - criado para execução em computadores do tipo *desktop* e *notebook* - e o código cliente desenvolvido para dispositivos de pequeno porte do tipo *Pocket PC*. Cada um dos dois tipos de aplicativos utiliza bibliotecas e um pequeno sistema de captura de vídeo e, assim, realizam tarefas como: codificação e decodificação de quadros de vídeo e interação com a placa de rede sem fio do dispositivo de pequeno porte.

A documentação aqui descrita busca apresentar os dois principais aplicativos através de diagramas de classes simplificados, tabelas de eventos (identificando as classes criadoras e as receptoras) e a estrutura das principais mensagens trocadas entre o cliente e o servidor.

A.2 Aplicativo Servidor

O aplicativo servidor contém, no pacote principal, as classes responsáveis pela iniciação do aplicativo. Este pacote, como indicado na figura A.1, armazena o formulário inicial e outros pequenos formulários. Contém, ainda, a classe que interage com todos esses formulários e o restante do sistema, chamada de *VideoConferenceManager*.

Todas as figuras que representam uma simplificação dos diagramas de classe de parte do aplicativo possuem as seguintes características: os quadros tracejados representam um pacote, ou seja, um diretório do aplicativo; cada caixa sólida descreve uma classe tipo formulário, indicando o nome do formulário correspondente; cada retângulo sombreado representa uma interface ou uma classe simples; as setas tracejadas indicam que uma determinada classe implementa aquela interface; e, as setas normais indicam os objetos que uma determinada classe pode instanciar.

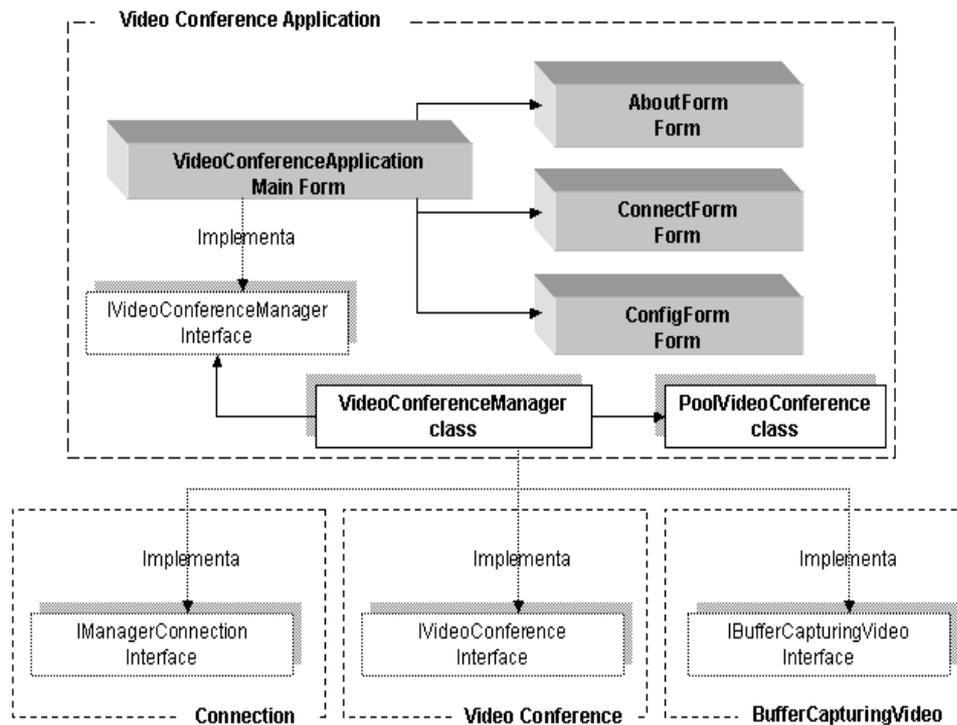


Figura A.1: Classes pertencentes ao pacote principal do aplicativo servidor.

O pacote chamado *VideoConference* contém as classes de exibição do vídeo e interage com os pacotes *Connection* e *WirelessEnvironment* através da implementação das interfaces propostas por estes últimos pacotes. Como pode ser visto na Figura A.2 o pacote *VideoConference* propõe uma interface, que está sendo implementada pela classe *VideoConferenceManager* do pacote principal. Por fim, essa classe utiliza algumas funções da biblioteca *DecoderH263* para a decodificação dos quadros de vídeo codificados com o padrão H263.

Além de implementar os métodos propostos pelas interfaces *IControlConnection* e *IWirelessEnvironment* dos pacotes *Connection* e *WirelessEnvironment*, a classe *VideoConference* recebe alguns eventos gerados em cada pacote. Sendo assim, todas as vezes que um determinado evento ocorre, em um dos dois pacotes, a classe *VideoConference* é acionada imediatamente para tratá-los.

No pacote *Connection*, mostrado pela Figura A.3, são encontradas as classes que gerenciam todas as conexões do aplicativo. Nesse pacote pode-se perceber a existência de outros três pacotes: *Useful*; *Protocols* e *TimerEvents*. No primeiro são armazenadas as constantes e funções utilizadas por todas as classes desse pacote. O segundo tem as classes que implementam os protocolos de comunicação TCP e UDP. As duas classes utilizam-se de um *pool* de portas de conexão, assim, antes de iniciar a espera por uma conexão de

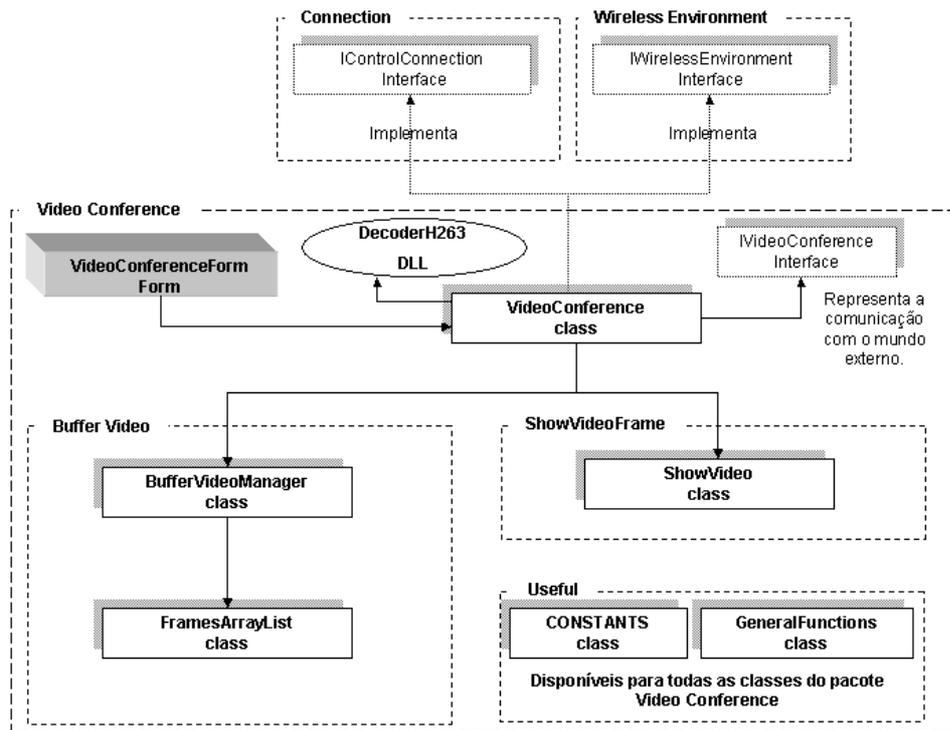


Figura A.2: Classes pertencentes ao pacote *VideoConference* do aplicativo servidor.

controle, texto, áudio ou vídeo, o aplicativo busca uma porta disponível. É importante ressaltar que a conexão chamada de *Manager*, responsável pela recepção inicial de um novo cliente, é realizada sempre por uma porta bem conhecida, no caso a 13000. O terceiro pacote, o *TimerEvents*, armazena classes que representam contadores de tempo para diferentes funções. Assim, quando um tempo de espera chega ao fim é disparado um evento que será capturado, geralmente, pela classe *ControlConnection*.

As classes que fornecem métodos para conexões TCP e UDP se comunicam com as classes que às instanciam também através de eventos. Todos os eventos gerados por essas duas classes são tratados nas entidades que representam os diferentes tipos de conexão do aplicativo, *ManagerConnections*, *ControlConnection*, *TextConnection*, *AudioConnection* e *VideoConnection*.

O pacote *WirelessEnvironment* é responsável por capturar os sinais dos pontos de acesso, utilizando para isso algumas funções de uma versão modificada da biblioteca WRAPI. Além disso, as classes que formam as políticas propostas foram disponibilizadas pois, o aplicativo permite através de uma simples configuração, que o cliente envie os sinais dos pontos de acesso alcançáveis para o servidor e, este, se responsabiliza pela análise e aplicação da política.

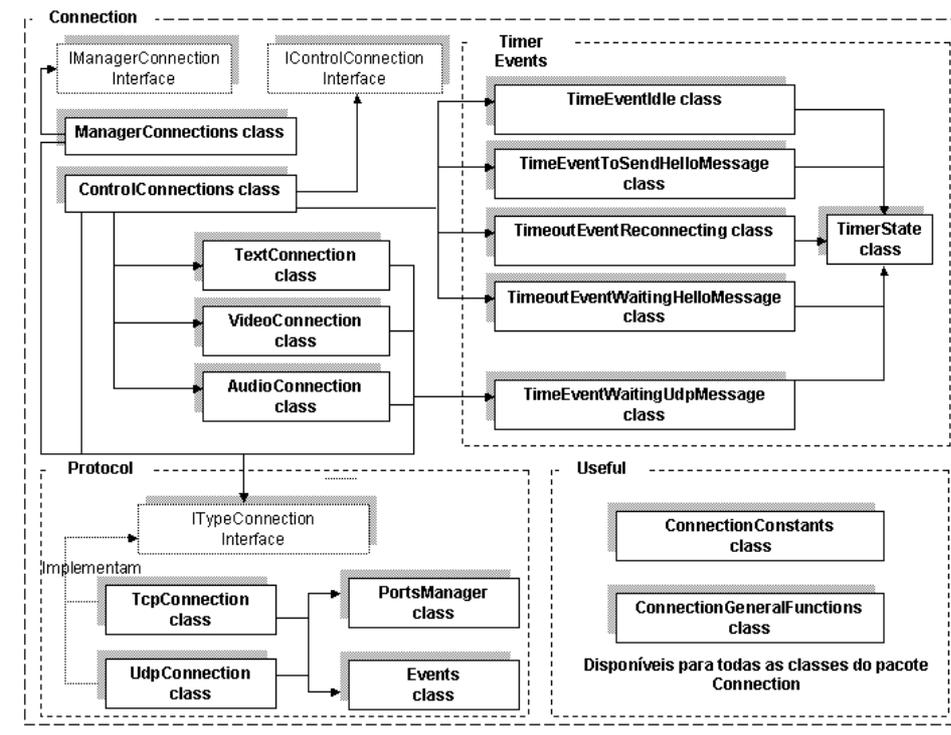


Figura A.3: Classes pertencentes ao pacote *Connection* do aplicativo servidor.

Como pode ser visto na Figura A.4 existe, ainda, dois outros pacotes, um para armazenar as políticas propostas e outro para disponibilizar métodos e constantes específicos do pacote *WirelessEnvironment*. Pode ser destacado ainda, que a classe *IntelligenceProtocolManager* cria um evento, de tempo em tempo, responsável por iniciar uma nova busca nas potência dos sinais dos diversos pontos de acesso.

O pacote *ControlerH263*, como mostrado na Figura A.5 através da classe *Desktop* inicia um pequeno aplicativo, o *EncoderH263*. Responsável por capturar os *frames* de vídeo da câmera e codificá-los. Esse aplicativo se comunica com servidor para transmitir os quadros de vídeo devidamente numerados, com uma carimbo de tempo e codificados com o padrão H263.

O pacote *BufferCapturingVideo*, mostrado na Figura A.6 se responsabiliza por armazenar os quadros de vídeo capturados e codificados. A classe *SenderVideo* dispara um evento, de tempo em tempo. Este evento será capturado pela classe *VideoConference* e, então, um quadro de vídeo será retirado do *buffer* e enviado ao cliente.

Por fim, o pacote *CaptureImage*, esquematizado na Figura A.7 permite ao aplicativo servidor capturar imagens sem a utilização do aplicativo *EncoderH263*. Dessa forma, o aplicativo pode utilizar o outro *codec* disponíveis no pacote (o JPEG, representado pela

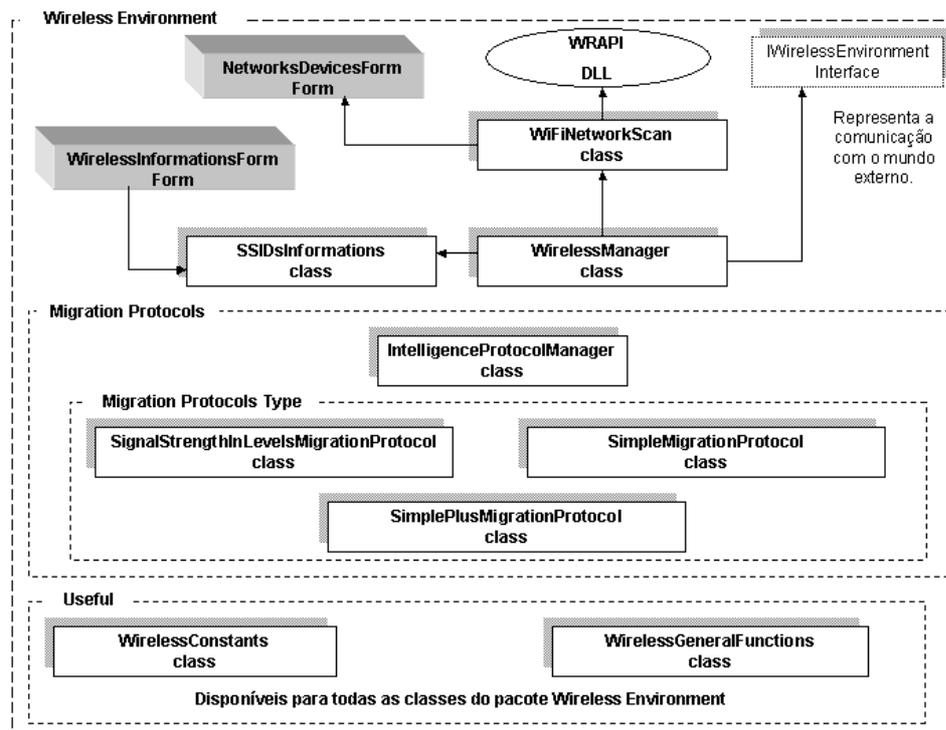


Figura A.4: Classes pertencentes ao pacote *WirelessEnvironment* do aplicativo servidor.

classe *NoneCodec*) e transmitir o vídeo utilizando outro padrão. Esse pacote permite ao aplicativo a captura de *streams* de vídeo de outras fontes, como a Internet, mas, nesses casos o único codec que poderá ser utilizado será o JPEG.

Após a apresentação dos diagramas de classe simplificados de todos os pacotes do sistema servidor, a Figura A.8 mostra as relações entre esses pacotes. Analisando a figura pode-se perceber quais entidades são responsáveis por instanciar objetos de classes localizadas em outros pacotes.

A.2.1 Métodos da Biblioteca DecoderH263

Essa biblioteca do servidor, tem a finalidade de decodificar quadros de vídeo no padrão H263. Assim, a tabela A.1 mostra as duas funções oferecidas por essa biblioteca.

A.2.2 Métodos da Biblioteca WRAPI

A biblioteca WRAPI foi alterada e algumas funcionalidades foram adicionadas. Com a finalidade de interagir com a placa de rede de computadores do tipo *desktops* e *notebooks*,

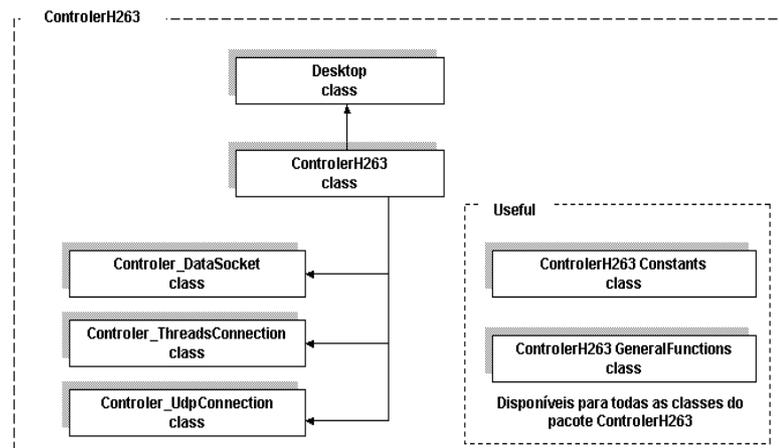


Figura A.5: Classes pertencentes ao pacote *ControllerH263* do aplicativo servidor.

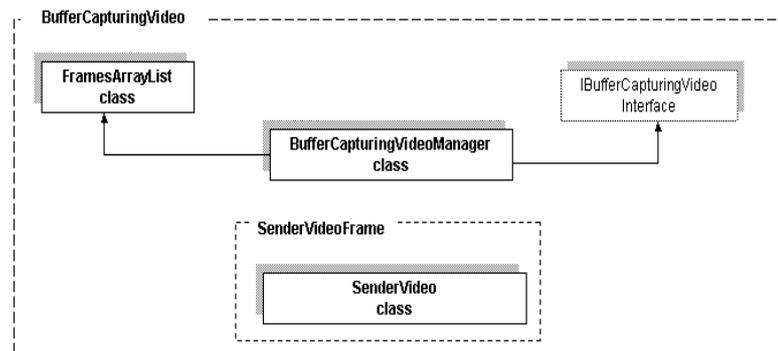


Figura A.6: Classes pertencentes ao pacote *BufferCapturingVideo* do aplicativo servidor.

essa biblioteca fornece funções importantes. Assim, a tabela A.3, no final deste apêndice, mostra as funções disponíveis e comenta as consideradas principais.

A.3 Aplicativo Cliente

O aplicativo cliente também contém, no pacote principal, as classes responsáveis pela iniciação do aplicativo. Este pacote, como apresentado pela Figura A.9, se diferencia do mesmo pacote, no aplicativo servidor, pela interatividade com o pacote *WirelessEnvironment* através do formulário principal do sistema. Outra diferença está na ausência do pacote de captura de vídeo.

Já no pacote *VideoConference* a mudança ocorre por não existir um formulário representando a videoconferência (o formulário inicial engloba esta função) e devido a classe *VideoConference* não implementar as funções do pacote *WirelessEnvironment*. A Figura A.10 mostra as classes existente nesses pacotes e suas relações.

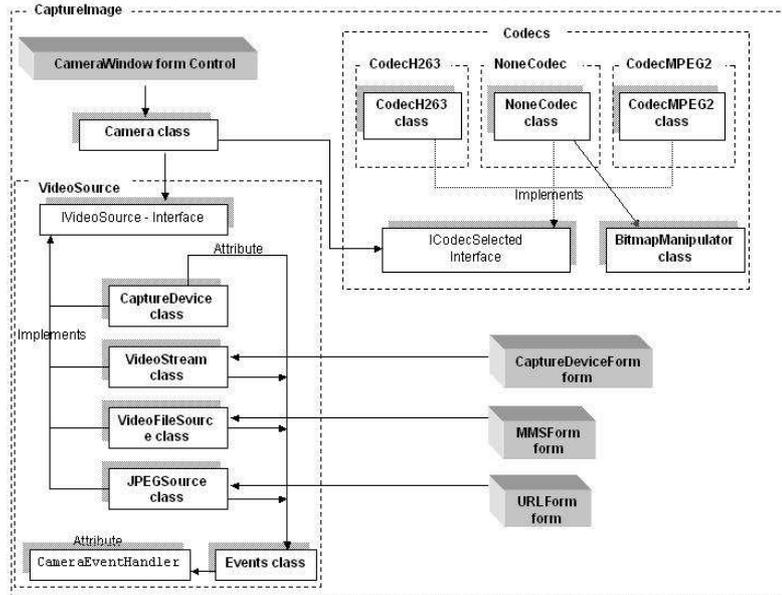


Figura A.7: Classes pertencentes ao pacote *CaptureImage* do aplicativo servidor.

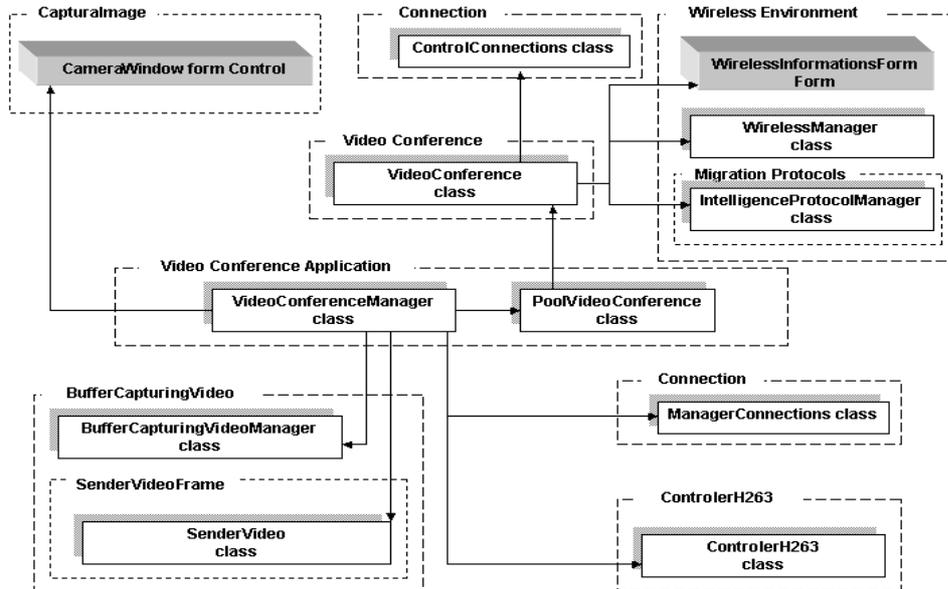


Figura A.8: Relações entre os pacotes do aplicativo servidor.

O pacote *Connection* não sofre qualquer alteração quando comparado ao mesmo pacote do aplicativo servidor e, portanto, a Figura A.3 pode representar o pacote em questão.

Por fim, o pacote *WirelessEnvironment*, como mostrado pela Figura A.11, apresenta dois novos pacotes internos: o *TimerEvents*, que tem uma classe responsável por observar o tempo de uma migração e disparar um evento caso o tempo máximo se esgote; e o *IPaQFuntions*, onde são utilizadas algumas funções da biblioteca *IPaQUtil*, que faz parte do sistema operacional dos dispositivos *Pocket PC*. Essa biblioteca oferece funções como controle da bateria, ligar e desligar a placa de rede, entre outras.

De forma similar ao final da seção anterior, após a apresentação dos diagramas de classe simplificados do sistema cliente, a Figura A.12 mostra as relações entre os quatro

Função	Descrição
InitDecoderH263	Inicia o decodificar.
DecoderFrameH263	Decodifica cada quadro de vídeo. Para isso, recebe o array de bytes de cada quadro, como o primeiro parâmetro; o segundo parâmetro é um array de bytes que retornará o quadro decodificado; e, por último, tem-se um inteiro representando o tamanho do quadro.

Tabela A.1: Funções da biblioteca DecoderH263.

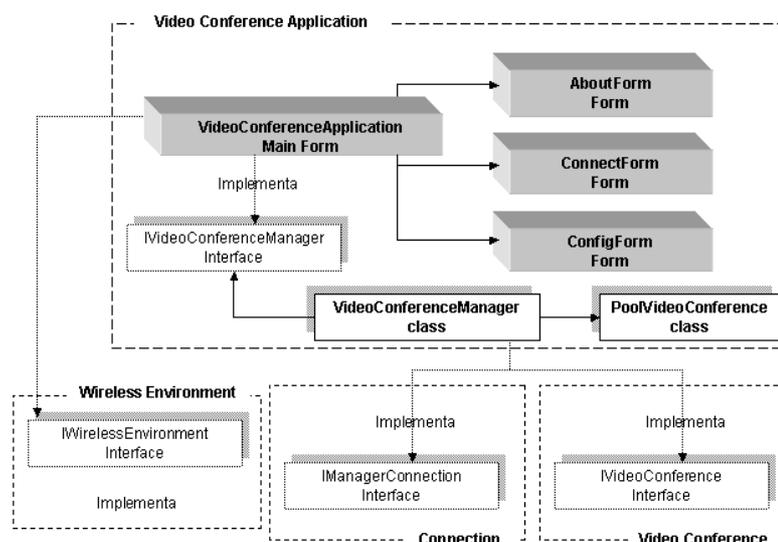


Figura A.9: Classes pertencentes ao pacote principal do aplicativo cliente.

pacotes apresentados.

A.3.1 Métodos da Biblioteca DecoderH263DLL_ARM

Essa biblioteca oferece as mesmas funcionalidades encontradas na tabela A.1 do aplicativo servidor. Devido as características dos processadores ARM nos dispositivos *Pocket PCs* surgiu a necessidade de criar outra biblioteca para o aplicativo cliente.

A.3.2 Métodos da Biblioteca WiFiConnection

A biblioteca *WiFiConnection* foi desenvolvida para os dispositivos de pequeno porte com o objetivo de interagir diretamente com a placa de rede. As funções, juntamente com suas descrições podem ser encontradas na tabela A.4, no final deste apêndice.

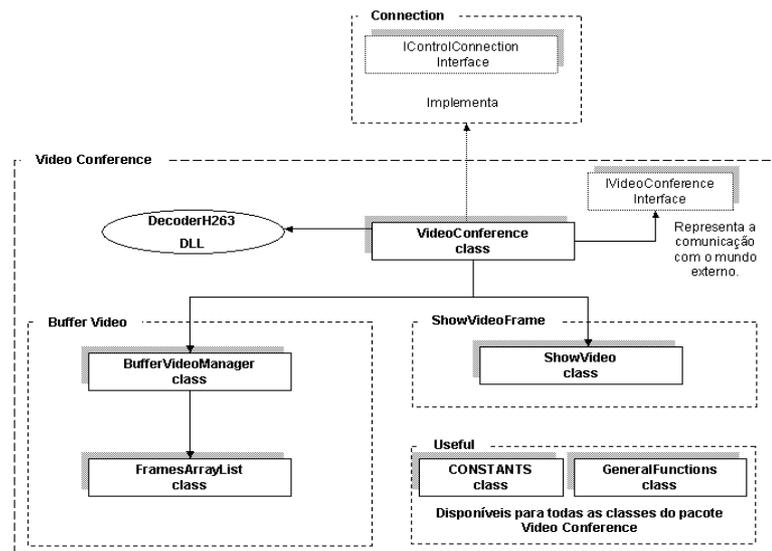


Figura A.10: Classes pertencentes ao pacote *VideoConference* do aplicativo cliente.

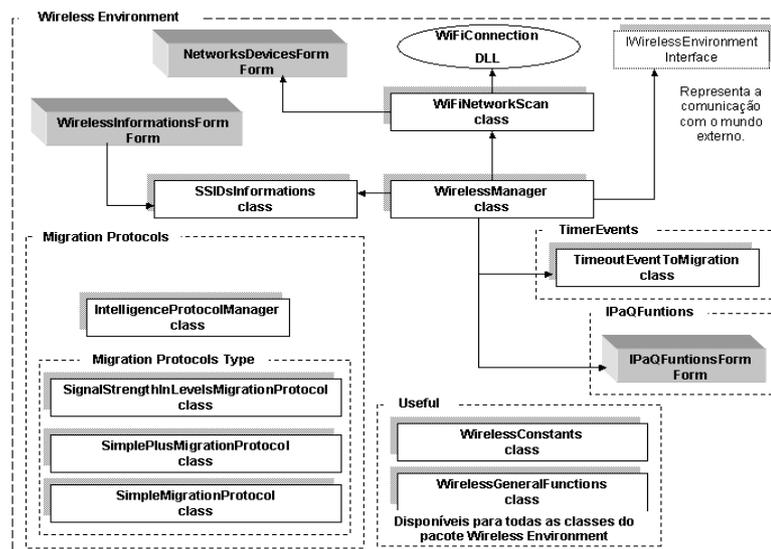


Figura A.11: Classes pertencentes ao pacote *WirelessEnvironment* do aplicativo cliente.

A.4 Estruturas das Mensagens do Aplicativo

A estrutura das principais mensagens do aplicativo pode ser encontrada na tabela A.2 abaixo, onde são descritos os atributos específicos de cada tipo de mensagem.

Observação: Apenas por escolha, inteiros são representado por quatro *bytes*.

A.5 Principais Eventos do Aplicativo

Nesta seção serão detalhados os principais eventos do aplicativo. O objetivo será indicar qual a classe geradora do evento e qual a classe que recebe e trata cada evento. Para isso,

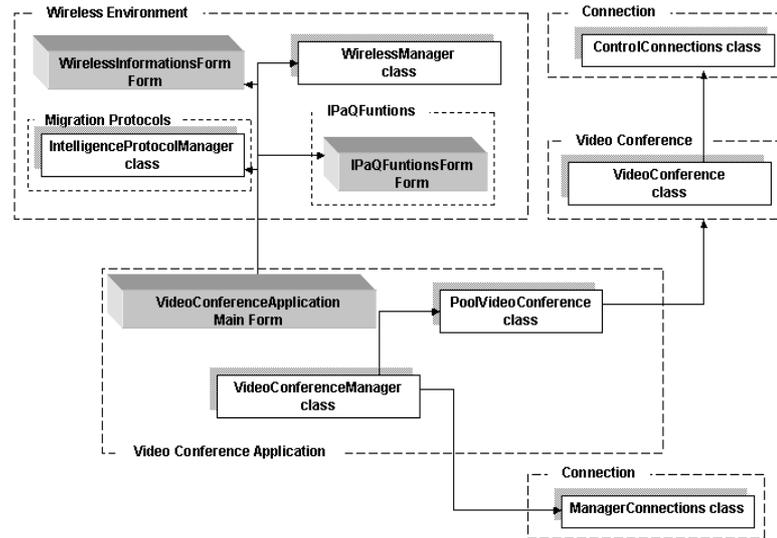


Figura A.12: Relações entre os pacotes do aplicativo cliente.

o evento será descrito em uma tabela esquematizada da seguinte forma: a primeira linha descreve o nome do evento; a primeira coluna na segunda linha indica o nome da classe que gerou o evento; a segunda coluna registra o nome da classe de captura; a terceira coluna marca se o evento existe apenas no aplicativo servidor (para isso a letra *S* será usada), apenas no cliente (*C*) ou nos dois aplicativos (*S/C*); a quarta linha descreve a ação de cada evento.

Tipo da Mensagem	
Parâmetro	Descrição
Mensagem para Conexão "TYPE_CONNECT"	
Tipo da Mensagem: RequestConnect	
Primeiro	Tamanho da Mensagem (inteiro).
Segundo	Tipo de Mensagem (inteiro) (Request Connect). O cliente envia esta mensagem e o servidor responde se aceita ou não essa nova conexão, caso positivo, o servidor envia uma mensagem do tipo resposta (Response Connect).
Terceiro	Tamanho do nome local da máquina (inteiro).
Quarto	Nome Local da Máquina (variável).
Quinto	Tamanho do endereço IP da máquina Local (inteiro).
Sexto	Endereço IP da máquina local (variável).
Sétimo	Codec utilizado na transmissão do Vídeo (H263 ou JPEG) (inteiro).

Oitavo	Código que identifica o tipo de política de migração (inteiro) (OBSOLETO).
Nono	Quem deverá executar a política de migração (Servidor or Cliente) (inteiro) onde 0 é o Cliente e 1 será o Servidor.
Tipo da Mensagem: ResponseConnect	
Primeiro	Tamanho da Mensagem (inteiro).
Segundo	Tipo de Mensagem (inteiro) (Response Connect). O servidor responde se aceita ou não uma nova conexão com o cliente utilizando esse tipo de mensagem.
Terceiro	Tamanho do nome local da máquina (inteiro).
Quarto	Nome Local da Máquina (variável).
Quinto	Tamanho do endereço IP da máquina Local (inteiro).
Sexto	Endereço IP da máquina local (variável).
Sétimo	Tipo de conexão (estático) TCP ou UDP, para transmissão de Texto, Audio ou Video.
Oitavo	Porta para o cliente se conectar com a conexão de controle criada pelo servidor (inteiro)
Mensagens do Tipo Controle "TYPE_CONTROL"	
Sétimo	Tipos de Mensagem de Controle i MSG_CONNECT ii MSG_ACCEPT iii MSG_REJECT iv MSG_DISCONNECT v MSG_CONFIRM_CONNECT São constantes inteiras declaradas na classe CONSTANTS
Tipo da Mensagem	
Parâmetro	Descrição
Mensagens do Tipo Controle "TYPE_CONTROL"	
Mensagens para estabelecer conexões de Texto, Vídeo ou Áudio	
Primeiro	Tamanho da mensagem (inteiro)
Segundo	Tipo da Mensagem (inteiro) - Controle
Terceiro	Subtipos de Mensagem para estabelecer Conexões i MSG_CONNECT_TEXT

	ii	MESG_ACCEPT_TEXT
	iii	MESG_REJECT_TEXT
	iv	MESG_CONNECT_AUDIO
	v	MESG_ACCEPT_AUDIO
	vi	MESG_REJECT_AUDIO
	vii	MESG_CONNECT_VIDEO
	viii	MESG_ACCEPT_VIDEO
	ix	MESG_REJECT_VIDEO
São constantes inteiras declaradas na classe CONSTANTS		
Mensagens do Tipo TEXTO, AUDIO ou VIDEO		
Primeiro	Tamanho da mensagem (inteiro)	
Segundo	Tamanho dos Dados(inteiro)	
Terceiro	Dados (bytes)	

Tabela A.2: Estrutura de algumas mensagens do aplicativo.

E V E N T O S D O A P L I C A T I V O		
Evento: NewShowVideo		
Criado por	Capturado por	Existente no
ShowVideo	VideoConference	(S/C)
Descrição: Retira um quadro de vídeo do <i>buffer</i>, decodifica-o e visualiza-o.		

Evento: EventIdle		
Criado por	Capturado por	Existente no
TimeEventIdle	ControlConnection	(S/C)
Descrição: Caso o servidor não receba uma mensagem de <i>hello</i> do cliente, indicando que ele continua conectado, o gerenciador da conexão de controle encerra toda a videoconferência.		

Evento: EventToSendHelloMessage		
Criado por	Capturado por	Existente no
TimeEventToSendHelloMessage	ControlConnection	(S/C)
Descrição: De tempo em tempo o evento é criado para que a classe que gerencia a conexão de controle envie a mensagem de <i>hello</i> do cliente para o servidor.		

Evento: EventReconnecting		
Criado por	Capturado por	Existente no
TimeoutEventReconnecting	ControlConnection	(S/C)
Descrição: Caso a reconexão não finalize em um prazo pré-determinado, todas as conexões são fechadas e o processo de reconexão inicia novamente.		

Evento: EventWaitingHelloMessage		
Criado por	Capturado por	Existente no
TimeoutEventWaitingHelloMessage	ControlConnection	(S/C)
Descrição: O aplicativo servidor espera por uma mensagem de <i>hello</i> durante <i>t</i> unidades de tempo, após isso, se nenhuma mensagem for recebida, o servidor conclui que o cliente não está mais presente e então encerra a videoconferência.		

Evento: EventWaitingUdpMessage		
Criado por	Capturado por	Existente no
TimeoutEventWaitingUdpMessage	TextConnection ÁudioConnection VídeoConnection	(S/C)
Descrição: Todas as conexões não confiáveis, UDP, enviam uma mensagem de requisitando conexão, caso esta mensagem não seja recebida tanto pelo cliente quanto pelo servidor, o processo de estabelecimento de uma conexão reinia.		

Evento: NewListening		
Criado por	Capturado por	Existente no
TCPConnection UDPConnection	ManagerConnections ControlConnection TextConnection AudioConnection VideoConnection	(S/C)
Descrição: Cria um evento indicando ao aplicativo servidor que a conexão está em estado de espera.		

Evento: NewData		
Criado por	Capturado por	Existente no
TCPConnection UDPConnection	ManagerConnections ControlConnection TextConnection AudioConnection VideoConnection	(S/C)
Descrição: Cria um evento que comunica as classes gerenciadoras das diferentes conexões que uma mensagem acabou de ser recebida.		

Evento: NewException		
Criado por	Capturado por	Existente no
TCPConnection UDPConnection	ManagerConnections ControlConnection TextConnection AudioConnection VideoConnection	(S/C)
Descrição: Cria um evento que comunica as classes gerenciadoras das diferentes conexões que um erro ocorreu nas classes que gerenciam os <i>sockets</i> .		

Evento: NewMessage		
Criado por	Capturado por	Existente no
TCPConnection UDPConnection	ManagerConnections ControlConnection TextConnection AudioConnection VideoConnection	(S/C)
Descrição: Cria um evento que comunica as classes gerenciadoras das diferentes conexões diversos tipos de mensagens pré-estabelecidas (ver a classe ConnectionConstants para conhecer as diferentes mensagens geradas).		

Evento: NewIntelligenceProtocolEvent		
Criado por	Capturado por	Existente no
IntelligenceProtocolManager	VideoConference VideoConferenceApplication	(S) (C)
Descrição: Cria um evento responsável pela execução da busca por novas potência de sinais dos pontos de acesso.		

Evento: NewFrame		
Criado por	Capturado por	Existente no
SendFrame	VideoConferenceManager	(S)
Descrição: Criado para enviar um vídeo do servidor para o cliente.		

Evento: EventToMigration		
Criado por	Capturado por	Existente no
TimeoutEventToMigration	WirelessManager	(C)
Descrição: Todas as vezes que uma migração não finaliza em um tempo pré-determinado esse evento é gerado para que seja iniciado um novo processo de migração.		

Função	Descrição
WRAPIEnumerateDevices	
WRAPIOpenNdisDevice	
WRAPIGetAPList	Essa função retorna a lista dos pontos de acesso alcançáveis, e o tamanho dessa lista. Cada posição da lista contém uma estrutura com dados de um ponto de acesso do tipo, nome e potência do sinal.
WRAPIGetAssociatedAP	Retorna o endereço MAC do ponto de acesso ao qual se está associado.
WRAPISetAssociatedAP	
WRAPIGetSSID	Retorna o nome do ponto de acesso ao qual se está associado.
WRAPISetSSID	
WRAPIGetRTSThreshold	
WRAPISetRTSThreshold	
WRAPIGetFragThreshold	
WRAPISetFragThreshold	
WRAPIGetPacketStats	
WRAPIGetPowerMode	
WRAPISetPowerMode	
WRAPIGetNetworkTypes	
WRAPIGetNetworkTypeInUse	
WRAPISetNetworkTypeInUse	
WRAPIGetSignalStrength	
WRAPIGetNumberOfAntennas	
WRAPIGetDesiredRates	
WRAPIDisassociate	
WRAPIGetTxPowerLevel	
WRAPISetTxPowerLevel	
WRAPIGetNetworkMode	
WRAPISetNetworkMode	
WRAPIGetWEPStatus	
WRAPISetWEPStatus	
WRAPIGetConfiguration	
WRAPISetConfiguration	
WRAPIGetAuthMode	
WRAPISetAuthMode	
WRAPItoConnect	Realiza a conexão com um determinado ponto de acesso. O único parâmetro da função descreve o nome do ponto de acesso ao qual se quer conectar.
WRAPItoDisconnect	Realiza a desconexão com o ponto de acesso.

Tabela A.3: Funções da biblioteca WRAPI.

Função	Descrição
StartWiFiConnection	Esta função inicia alguns parâmetros e, através do primeiro parâmetro permite o encaminhamento dos eventos gerados pelo <i>hardware</i> para a aplicação cliente principal.
FinalizeWiFiConnection	Finaliza alguns parâmetros.
RadioEnable	Habilita a placa de rede, caso esteja desabilitada.
GetAPList	Essa função retorna a lista dos pontos de acesso alcançáveis, e o tamanho dessa lista. Cada posição da lista contém uma estrutura com dados de um ponto de acesso do tipo, nome e potência do sinal.
GetSSID	Retorna o nome do ponto de acesso ao qual se está associado.
GetSignalStrength	Retorna a potência do sinal do ponto de acesso ao qual se está associado.
Connect	Realiza a conexão com um determinado ponto de acesso. O único parâmetro da função descreve o nome do ponto de acesso ao qual se quer conectar.
Disconnect	Realiza a desconexão com o ponto de acesso.
HasZeroConfig	Verifica se existe a interface do WZC do windows.
GetZeroConfigStatus	Retorna o <i>status</i> da interface do WZC do windows.
EnableZeroConfig	Habilita e desabilita a interface do WZC do windows.

Tabela A.4: Funções da biblioteca WiFiConnection.