UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DISSERTAÇÃO DE MESTRADO

SISTEMA DE TRANSMISSÃO DE VÍDEO PARA VIGILÂNCIA UTILIZANDO BLUETOOTH

Alisson Vasconcelos de Brito

Elmar Uwe Kurt Melcher
(Orientador)

Campina Grande – PB
Fevereiro - 2003

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

SISTEMA DE TRANSMISSÃO DE VÍDEO PARA VIGILÂNCIA UTILIZANDO BLUETOOTH

Alisson Vasconcelos de Brito

Elmar Uwe Kurt Melcher
(Orientador)

Dissertação submetida à Coordenação do Curso de Pós-graduação em Informática da Universidade Federal de Campina Grande, como parte dos requisitos necessários para obtenção do grau de mestre em Informática.

Área de concentração: Ciência da Computação.

Linha de pesquisa: Redes de Computadores e Sistemas Distribuídos.

Campina Grande – PB
Fevereiro – 2003

BRITO, Alisson Vasconcelos de

B862S

Sistema de Transmissão de Vídeo para Vigilância Utilizando Bluetooth

Dissertação (Mestrado), Universidade Federal de Campina Grande, Coordenação de Pós-Graduação em Informática, Campina Grande – Pb, fevereiro de 2003.

86p. Il.

Orientador: Dr. Elmar Uwe Kurt Melcher

Palavras-chaves:

1. Redes de computadores sem fio
2. Simulação
3. Bluetooth
4. Transmissão de Vídeo

CDU – 621.391

"SISTEMA DE TRANSMISSÃO DE VÍDEO PARA VIGILÂNCIA UTILIZANDO BLUETOOTH"

ALISSON VASCONCELOS DE BRITO

DISSERTAÇÃO APROVADA EM 25.02.2003

PROF. ELMAR UWE KURT MELCHER, Dr.
Orientador

PROFª MARIA IZABEL CAVALCANTI CABRAL, D.Sc
Examinadora

PROF. SÉRGIO VANDERLEI CAVALCANTE, Ph.D
Examinador

CAMPINA GRANDE – PB

"Se não tens fé em Deus, em ti mesmo,
no ideal que abraças, ou no trabalho que realizas,
observa se o carro é capaz de avançar com precisão,
quando o motor não encontra o apoio do pedal de
embreagem."

(Emmanuel por Chico Chavier)

Agradecimentos

A Deus, meu eterno guia e inspirador.

À minha mãe, Graça, sempre tranqüila, confiante e amorosa.

Ao meu pai, Sérgio, pelos seus conselhos, força e segurança transmitidos.

À minha namorada, Ana Laura, meu amor, minha confidente, minha inspiração.

Aos meu irmãos, Sérgio e Gracielle, meus maiores e eternos amigos.

Aos meus amigos, Alexandre, Petrônio e Eloi, nos quais eu sempre pude confiar e sempre estiveram dispostos a me apoiar.

Ao meu orientador, Elmar, que me ensinou o mais importante, ser um mestre.

Resumo

O interesse atual em proteger os patrimônios contra a ação de infratores e curiosos tem aumentado a procura por melhores sistemas de vigilância. Esse trabalho descreve um sistema de transmissão de vídeo sem fio para vigilância mais prático, dinâmico, ágil e seguro do que os tradicionais, fazendo uso da tecnologia Bluetooth e da compressão de vídeo pelo padrão MPEG-4.

Resultados adquiridos através de simulação utilizando o ambiente Ptolemy são apresentados, testando como o sistema se comporta quando um servidor transmite vídeos variando o número de clientes, o tamanho e o tipo dos pacotes, a taxa de quadros e o tamanho do *buffer* dos clientes. Esses resultados ressaltam as vantagens e a viabilidade desse nosso sistema.

Pesquisas sobre as três tecnologias envolvidas nesse trabalho também são apresentadas, as redes Bluetooth, a codificação MPEG-4 e a simulação e modelagem com o Ptolemy.

Abstract

The current interest in protection of private property against trangressors and onlookers has increased the demand for better surveillance systems. This work describes a wireless video transmission system for surveillance purposes that is more practical, agile, dynamic, and secure than traditional ones by using Bluetooth technology, with a wireless self configurable network, together with MPEG-4 video encoding, with extra protection and high compression rates.

Results acquired from simulation using the Ptolemy environment are presented showing how the system behaves when a server transmits videos varying packet size and type, frame rate and the client buffer size. These results show our system's advantages and viability.

Research about the three technologies involved in this work is also presented: Bluetooth networks, MPEG-4 encoding and Ptolemy modeling framework.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

1. Introdução

Nesta introdução apresentamos as motivações que nos levaram a desenvolver esse trabalho, introduzindo os temas mais importantes para a sua execução. São apresentados também os objetivos gerais e específicos, um resumo dos resultados encontrados, e explicações de como a Dissertação está organizada.

1.1. Motivações

A busca cada vez maior pela proteção de patrimônios tem proporcionado grandes investimentos em pesquisa e desenvolvimento. Especificar um sistema de segurança, através de câmeras de vídeo, inovador que atenda aos requisitos dos clientes é uma das motivações de nosso trabalho.

Para tanto, apoiamo-nos em duas tecnologias atuais, Bluetooth e MPEG-4 que, juntas, são capazes de criar um sistema de vigilância mais dinâmico e flexível do que os atuais. O estudo dessas tecnologias também nos motiva, já que serve de base para outros estudos. Os testes feitos unindo essas duas tecnologias trouxeram resultados importantes para a área de redes de computadores sem fio e da codificação de vídeo, além de deixar contribuições para a área da simulação digital.

1.1.1. TECNOLOGIA BLUETOOTH

São várias as tecnologias de redes sem fio existentes. Bluetooth é uma tecnologia que visa difundir a comunicação sem fio entre os dispositivos eletrônicos, sejam eles grandes ou pequenos. A idéia é criar dispositivos com transceptores de baixo custo, baixo consumo de energia e curto alcance. O que faz essa tecnologia especial é o fato das pessoas não precisarem conectar, instalar, habilitar ou configurar nenhum equipamento (Bray et. al., 2001, p. 1).

Dispositivos Bluetooth formam pequenas redes apenas aproximando-se uns dos outros. Eles podem ser configurados para constantemente procurarem outros dispositivos que estejam aos seus redores. Quando detectado, o dispositivo apresenta aos outros quais serviços ele dispõe. Eles podem, ou não, escolher um dos serviços que deseja utilizar. Por exemplo, uma pessoa, ao entrar numa sala com seu telefone celular, recebe uma mensagem avisando que uma impressora está presente no local e que o único serviço do qual ela dispõe é o de impressão. A pessoa então pode mandar imprimir sua agenda pela impressora detectada.

Os dispositivos podem ser configurados para permitir a execução de algumas tarefas sem nenhuma ação do usuário. Essa capacidade de auto-configuração é chamada computação oculta (Miller, 2001, p. 44). Uma vez configurados, dispositivos enviam sinais e executam tarefas assim que identificam outros dispositivos. Eles podem, por exemplo, enviar mensagens para um grupo de dispositivos avisando assim que um novo dispositivo entrar na área da rede, ou então, uma câmera de vídeo em um sistema de vigilância pode enviar uma alerta para o celular do vigilante, que pode discar automaticamente para um número de um serviço de segurança, assim que algum movimento estranho for detectado.

Para um equipamento ser considerado Bluetooth ele deve possuir um pequeno *microchip* com um transceptor que implementa uma pilha de protocolos. Por ter seu tamanho e seu consumo de energia reduzidos, o alcance dos dispositivos é limitado a 100 metros e sua taxa de transmissão é no máximo de 723,32Kbps.

1.1.2. TRANSMISSÃO DE VÍDEOS COM MPEG-4

A transmissão de vídeo através de redes sem fio possui peculiaridades que outras transmissões não possuem. A transmissão de dados por ondas de rádio sofre muitas interferências que as transmissões por cabos não sofrem, levando a necessidade de uma maior proteção dos dados. Diferentemente de transmissões de dados comuns, vídeos possuem características que adicionam ao sistema mais requisitos a serem atendidos. Os vídeos são sempre relacionados com tempo real, atrasos podem ser tão prejudiciais quanto perdas de dados.

Especialmente em transmissões de vídeo através de redes sem fios, os vídeos devem ser tratados (codificados) antes de serem transmitidos. Há vários codificadores existentes atualmente. O MPEG-4 é um deles e visa popularizar a exibição de vídeos para várias

aplicações, tanto em infraestruturas com altas taxas de bits, como a televisão digital, quanto em infraestruturas com baixa largura de banda e altas taxas de erros, como as redes sem fio.

O MPEG-4 provê, entre outras, as seguintes funcionalidades:

- Compressão, podendo diminuir o tamanho do vídeo em mais de 100 vezes, sem perdas significativas de qualidade;
- Proteção contra erros, permitindo que quadros inteiros sejam perdidos, sem que o vídeo como um todo seja gravemente afetado;
- Infraestrutura de sincronização: mesmo o sistema causando atraso dos dados transmitidos, os decodificadores podem recompor parte da sincronização do vídeo.

1.1.3. SIMULAÇÃO DIGITAL COM PTOLEMY

Para a concretização física desse trabalho seriam necessários vários investimentos não disponíveis para nós. Optamos pela simulação digital que é capaz de recriar situações reais que custariam muito dinheiro para serem implementadas com dispositivos Bluetooth reais.

Além disso, as simulações executadas aqui servem como base para que trabalhos futuros possam ser concretizados, observando os métodos utilizados e os resultados obtidos.

Para as simulações dos dispositivos e da infraestrutura Bluetooth, optamos pelos programas do *framework* Ptolemy. Esse *framework* foi desenvolvido para modelagem, simulação e projeto de sistemas concorrentes heterogêneos (Lee, 2001).

O *framework* em sua segunda versão (Ptolemy II) é escrito em Java e possui uma ferramenta visual (Vergil), além de um rico conjunto de componentes e mecanismos de interação. Ele também permite aos usuários uma fácil criação de seus próprios componentes e ambientes de simulação. Os componentes do Ptolemy II podem ser utilizados em projetos avançados e podem interagir com outros componentes desenvolvidos pelos usuários.

Todos os programas que fazem parte do *framework* Ptolemy são bem documentados, abertos e gratuitos. Programas mais avançadas para aplicações específicas, no entanto, podem ter um custo elevado.

1.2. Objetivos da Dissertação

1.2.1. OBJETIVO GERAL

Esse trabalho de Dissertação de Mestrado tem como objetivo principal:

- Especificar e demonstrar a viabilidade de um sistema de transmissão de vídeo para ser utilizado por sistemas de vigilância de bens e propriedades que utilize a tecnologia Bluetooth.

1.2.2. OBJETIVOS ESPECÍFICOS

Os objetivos específicos desse trabalho são:

- Testar e mostrar em quais condições vídeos podem ser transmitidos com boa qualidade utilizando a tecnologia Bluetooth aliada com o padrão MPEG-4;
- Comprovar a viabilidade da transmissão de vídeo através de dispositivos Bluetooth e servir de base para uma futura especificação formal da mesma;
- Difundir e apoiar a utilização da codificação MPEG-4 para transmissões de vídeo em redes sem fio;
- Apresentar o projeto Ptolemy e seus programas como uma ferramenta para a simulação digital de redes de computadores, inclusive redes sem fio.

1.3. Escopo e Relevância

Esse trabalho restringe-se às transmissões de vídeo adequadas aos sistemas de vigilância, não ignorando que essas transmissões também podem adequar-se às outras aplicações.

Adotamos como características próprias dos vídeos para vigilância:

- São uniderecionais (um determinado dispositivo não transmite e recebe vídeos simultaneamente);
- Não necessitam de som (apenas as imagens são transmitidas);
- Possuem requisitos reduzidos de qualidade (a percepção do ato e do infrator é suficiente).

A união dessas três tecnologias é a maior relevância desse trabalho. Simular redes Bluetooth com as ferramentas do Ptolemy II, utilizando codificação de vídeo MPEG-4, possui um caráter inovador que serve de exemplo para outros trabalhos. Apresentam-se contribuições novas na área de redes de computadores sem fio ao efetuar a transmissão de vídeos através de redes Bluetooth. Apesar de não ter sido projetada especificamente para esta aplicação, essa tecnologia, aliada com a codificação MPEG-4, possui capacidade de transmitir vídeos com qualidade. Além disso, são apresentadas contribuições para as áreas da codificação de vídeo e da simulação digital.

Esse trabalho não só contribui para a criação de sistemas de vigilância, como também dá apoio à criação de:

- Sistemas de transmissão de vídeo de qualquer natureza através de redes Bluetooth e de outras redes sem fio;
- Transmissões de vídeo sobre meios sofrendo altas taxas de erros de bits que desejam utilizar o padrão MPEG-4 como padrão de codificação de vídeo;
- Projetos que desejam utilizar a ferramenta Ptolemy para modelar e simular redes Bluetooth, bem como redes de computadores em geral.

1.4. Organização da Dissertação

O Capítulo 2, intitulado "Transmissão de Vídeo em Redes sem Fio", apresenta os aspectos mais relevantes numa transmissão de vídeo através de redes sem fio. Suas propriedades, os componentes necessários, suas peculiaridades, os padrões existentes e o padrão MPEG-4 são mostrados nesse capítulo.

Já no Capítulo 3 ("Redes sem Fio e a Tecnologia Bluetooth") são explicados os padrões para redes sem fio mais comuns e as características mais importantes da tecnologia Bluetooth, justificando o porquê da opção por essa tecnologia.

A arquitetura do nosso sistema é apresentada no Capítulo 4, "Arquitetura do Sistema de Transmissão de Vídeo para Vigilância". Esse capítulo caracteriza o que é um sistema de transmissão de vídeo para vigilância e quais seus requisitos. Definimos quais os elementos que fazem parte desse sistema e como a tecnologia Bluetooth e o padrão MPEG-4 podem auxiliar a atendermos seus requisitos.

Detalhes de como foram feitos a implementação e as simulações do sistema estão descritos no Capítulo 5 (“Implementação e Simulação”). Nesse capítulo monstramos como os elementos do sistema foram implementados através da simulação, apresentamos o ambiente Ptolemy, mostramos como ele foi utilizado e como ele foi importante para a concretização desse trabalho.

No Capítulo 6, “Resultados”, é explicado como foram organizadas as simulações. Também é feita uma análise completa de todos os resultados encontrados. A conclusão desse trabalho, ressaltando suas contribuições e sugestões para trabalhos futuros estão no Capítulo 7, “Conclusão”.

Capítulo 2

2. Transmissão de Vídeo em Redes sem Fio

A transmissão através de canais sem fio possui características diferentes das transmissões através de cabos. Quando dados são transmitidos através de ondas de radiofreqüência, a perda de bits, a baixa taxa de bits e atrasos devidos às retransmissões são fatores comuns que o sistema deve contornar para proporcionar uma transmissão de boa qualidade.

Transmitir vídeos é bem diferente de transmitir dados comuns. Essa peculiaridade dos vídeos é outro fator que faz com que tenhamos ainda mais cuidados ao falarmos em transmissão em redes sem fio. A união desses dois elementos tão peculiares, a transmissão de vídeo e as redes sem fio, é o tema desse capítulo.

Apresentaremos quais as principais características de uma transmissão de vídeo, no que ela se diferencia das outras transmissões, quais as dificuldades em se transmitir através de canais sem fio, quais técnicas são mais usadas e quais padrões estão presentes no mercado atualmente.

Esse capítulo está organizado da seguinte forma: apresentaremos na Seção 2.2 uma classificação de sinais de vídeo, uma introdução ao processo de codificação e qual sua importância. Já a Seção 2.3, fala dos elementos necessários para uma transmissão de vídeo através de redes sem fio. A Seção 2.4 traz algumas propriedades que são de maior relevância para uma transmissão de vídeo através de meios sem fio. Os padrões de codificação de vídeo, bem como seus objetivos e principais funcionalidades, são apresentados na Seção 2.5. O padrão MPEG-4 é mais bem detalhado na Seção 2.6 e a conclusão do capítulo é apresentada da Seção 2.7.

2.1. Codificação de vídeo

São várias as aplicações hoje que utilizam transmissão de vídeo. Uma das principais razões para a difusão de todas essas aplicações está no avanço das técnicas de codificação de vídeo que facilitam e melhoram sua utilização. Um vídeo antes de codificado é chamado de vídeo puro. Quando codificado, além de precisar de muito menos bits para ser representado digitalmente, geralmente possui proteção contra erros e incorpora mecanismos de sincronização.

Um vídeo é formado por uma seqüência de quadros (*frames*). Cada quadro é uma imagem instantânea da cena registrada. Um quadro é formado por uma matriz de *pixels* (*picture elements*), que são as unidades mais elementares de uma imagem. A dimensão dos quadros de um vídeo é dada pelo número de *pixels* que os quadros possuem na horizontal e na vertical. Existem vários padrões de tamanho de quadros, que refletem também o tamanho total de um vídeo.

Na Tabela 2-1, produzida com dados retirados de Barnett (2000), podemos observar as dimensões dos padrões para formatos de vídeos digitais recomendados pela ITU[1] e que são usados pelos padrões H.261, H.263 da ITU e pelos padrões MPEG[2] da ISO[3].

Padrão	Linhas horizontais *(pixels)*	Linhas verticais *(pixels)*	Dez minutos de vídeo a 30 quadros por segundo (Mbytes)
SIF	352	288	1522,5
CIF	352	288	1800,0
QCIF	176	144	450,0

Tabela 2-1: Dimensões de alguns padrões de tamanho de vídeo mais utilizados.

O formato SIF (Standard Input Format) utiliza valores diferentes para resolução vertical nos padrões PAL e NTSC (vídeos analógicos). Já os formatos CIF (*Common Interchange Format*) e QCIF (*Quarter Common Interchange Format*) suportam os formatos digitais PAL e NTSC com os mesmos parâmetros (Barnett, 2000). O formato QCIF é a versão reduzida do CIF, ocupando um quarto dos bits ocupados pelo CIF.

[1] *International Telecommunications Union*
[2] *Motion Picture Expert Group*
[3] *International Standard Organization*

2.2. Transmissão de vídeo em redes sem fio

Uma transmissão de vídeo através de redes sem fio é formada por cinco elementos: fonte de vídeo, codificador, canal sem fio, decodificador e sorvedouro. A Figura 2-1, mostra um esquema geral de uma transmissão de vídeo através de redes sem fio. O que diferencia uma transmissão de vídeo em redes sem fio das outras transmissões são as características do canal sem fio. Dados transmitidos através de redes sem fio sofrem mais interferências; canais sem fio são intrinsecamente errôneos (inserem muitos erros aos dados transmitidos) e geralmente não são capazes de transmitir a altas taxas de bits.

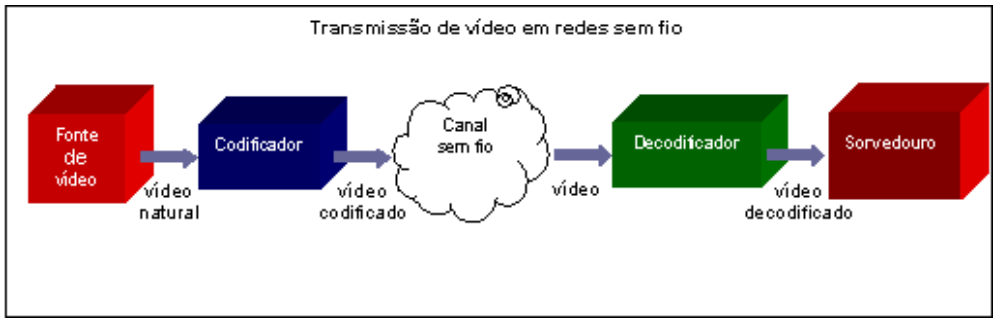

Figura 2-1: Esquema geral de uma transmissão de vídeo em redes sem fio.

A fonte de vídeo tanto pode ser natural, como artificial. Uma fonte natural é aquela que apenas produz vídeos de cenas que realmente aconteceram. As fontes artificiais trabalham com vídeos sintéticos, sejam gerados por computador, desenhos, montagem de figuras etc. Num sistema digital, os vídeos naturais gerados pela fonte servem de entrada para o codificador que irá digitalizar, comprimir e inserir bits de controle ao fluxo de bits recebido.

O Codificador é aquele que vai adicionar um tratamento ao vídeo antes de sua transmissão. Esse tratamento (codificação) é principalmente representado pela digitalização (amostragem e quantização) e pela compressão do vídeo. A evolução das técnicas de compressão tem popularizado a utilização dos vídeos por aplicações que antes não atendiam aos pré-requisitos para uma transmissão de um vídeo puro. Podemos observar na Tabela 2-1, que é necessário muito espaço para armazenar 10 minutos de vídeo quando apresentado a uma taxa de 30fps (*frames* por segundo), que resulta num vídeo de alta qualidade. Desta forma, um CD-ROM é capaz de armazenar apenas cerca de 15 minutos de vídeo no formato QCIF, isto torna impraticável o armazenamento e a transmissão de vídeos digitais sem compressão através de redes sem fio. Mesmo que um sistema possua banda para transmissão e memória para armazenamento, não é desejável que esse sistema dedique-se apenas à transmissão de um

só vídeo. O avanço nas técnicas de compressão tem popularizado a utilização de vídeo por várias aplicações.

A compressão tenta eliminar ao máximo as redundâncias contidas nos vídeos. A compressão mais comumente utilizada é a compressão com perdas, devido à necessidade de diminuir ainda mais o volume de informações contidas num vídeo, mesmo que isso resulte em uma qualidade reduzida do vídeo transmitido.

Na Tabela 2-2, baseada em Barnett (2000), podemos observar dois padrões bastante utilizados de tamanho de vídeos digitais, bem como a banda necessária para uma rede manter a taxa de quadros especificada, e quantos bits são necessários para armazenar dez minutos de vídeo colorido em cada padrão.

Taxa de quadros (fps)	CIF (Mbps)	QCIF (Mbps)	SIF (Mbps)
30	24,0	6,0	20,0
15	12,0	3,0	-
10	8,0	2,0	-
7,5	6,0	1,5	-

Tabela 2-2: Taxa de bits necessária para transmitir vídeos com diferentes taxas de quadros sem compressão.

A digitalização se destaca por possibilitar que os vídeos sejam comprimidos, armazenados, transmitidos, processados, exibidos e distribuídos com maior facilidade, agilidade e desempenho.

Taxa de quadros (fps)	CIF (Mbps)	QCIF (Mbps)	SIF (Mbps)
30	1,0	0,25	0,85
15	0,5	0,12	-
10	0,35	0,08	-
7,5	0,25	0,06	-

Tabela 2-3: Taxa de bits aproximada necessária para transmitir vídeos em diferentes taxas de quadros no padrão MPEG-2.

Como podemos observar na Tabela 2-3, o tamanho dos vídeos pode ser reduzido cerca de 24 vezes quando codificados pelo padrão MPEG-2, com isso a capacidade do canal necessária para transmitir esses vídeos é também reduzida em 24 vezes, outros padrões, dependendo das condições e configurações do vídeo original, podem reduzir ainda mais o tamanho de vídeos puros, chegando a torná-los cerca de 120 vezes menores (MPEG-4 e H.263, por exemplo) do que os vídeos originais.

Hoje em dia são muitas as aplicações que fazem uso dos vídeos digitais, por exemplo:

- Videoconferência;
- Videofone;
- TV digital;
- Vídeo na Internet;
- Diagnose e tratamento médico à distância;
- HDTV, entre outras aplicações.

O decodificador recebe vídeos codificados e reconstitui-os a ponto de poderem ser exibidos. É também papel do decodificador corrigir, na medida do possível, os bits que chegam corrompidos. Os padrões de codificação de vídeo utilizam meios diferentes de proteção contra erros.

Os vídeos decodificados são entregues ao sorvedouro que irá exibi-los ou armazená-los. Como geralmente a compressão implementada pelo codificador insere perdas aos dados, o vídeo reconstituído pelo decodificador nunca é idêntico ao vídeo codificado, mesmo assim, os vídeos podem chegar ao sorvedouro com ótima qualidade, isso porque a compressão busca eliminar redundâncias que são praticamente imperceptíveis aos olhos humanos.

2.3. Propriedades de uma transmissão de vídeo

Há duas formas básicas de transmitir dados em ambientes móveis. A primeira é a entrega livre de erros, que é garantida por protocolos que utilizam o esquema de *Automatic Repeat reQuest* (ARQ). Nesse esquema, os pacotes perdidos ou corrompidos são retransmitidos, garantindo que nenhum dado seja desperdiçado. Em compensação, isso faz com que o atraso e a taxa de transmissão sejam variáveis de acordo com a condição do canal. Aramvith et. al. (1998) faz estudos sobre os efeitos da retransmissão de pacotes numa transmissão de vídeo e propõe um esquema prático de controle da taxa de transmissão baseado numa predição estatística da situação futura do canal, tentando realocar para cada quadro um numero ideal de bits, diminuindo, então, o número de quadros perdidos na transmissão.

Na segunda forma de transmissão, os atrasos e a taxa de bits são constantes, no entanto os erros nos dados são freqüentes.

Segundo Hagenauer e Stockhammer (1999), há três tipos de serviços que podem ser providos em ambientes móveis, que se diferenciam pelos seus requisitos de desempenho. O primeiro serviço é formado por aquelas aplicações que têm como seu maior requisito de desempenho o atraso. Serviços bidirecionais, como bate-papo, vídeo-conferência e vídeo-telefonia, têm como principal objetivo a entrega de mensagens de forma sincronizada. Segundo Cox e Kroon (1998), recomenda-se que os atrasos fim-a-fim não ultrapassem os 400ms, e que o ideal seriam atrasos menores que 200ms.

Já serviços de entrega de dados binários são muito mais sensíveis a taxa de erros. Nesse tipo de serviço geralmente são utilizados mecanismos de ARQ para proteção e FEC (*Forward Error Correction*) para correção dos dados.

O terceiro tipo de serviço, no qual nosso sistema está inserido, é caracterizado por sua maior sensibilidade à taxa de transmissão. A principal aplicação utilizada aqui é a transmissão unidirecional de áudio e vídeo. Para essas aplicações, o atraso não é algo tão crucial quanto a taxa de bits que o sistema é capaz de transmitir. O sistema deve transmitir a uma taxa máxima de bits para garantir uma qualidade constante da aplicação, para isso, o codificador deve prever as condições do canal a fim de controlar a taxa de bits, e o decodificador deve cancelar ao máximo o efeito dos erros sobre os dados, diminuindo a perda dos dados e evitando a necessidade de retransmissão.

Na transmissão de vídeo através de dispositivos móveis, há sempre decisões que devem ser tomadas para prover robustez contra erros. Aumentar a compressão dos vídeos resulta numa diminuição do tamanho dos mesmos que, por sua vez, resulta numa menor necessidade por taxa de bits, porém a compressão não deve ser utilizada demasiadamente. Segundo Hagenauer, et. al. (1996), quanto maior a compressão impressa ao vídeo, maior sua sensibilidade aos erros do canal, por isso pode ser mais vantajoso deixar parte da redundância na fonte e utilizá-la no decodificador para melhorar a qualidade da decodificação. Compressão e qualidade estão em lados opostos da balança e escolher que peso colocar em cada lado não é uma escolha trivial.

Há varias técnicas para prover vídeos através de canais sem fio. Batra e Chang (1998) propõem uma técnica baseada na segmentação do vídeo. Diferentemente da segmentação

convencional, onde o vídeo é separado em segmentos de bits por tipo de quadros, cabeçalho e dados, a técnica de Batra e Chang (1998) visa separar o vídeo em segmentos baseados no conteúdo do vídeo (objetos de mídia, estrutura da cena, atributos dos objetos).

Já Motta, et. al. (2000) apresenta um algoritmo para minimizar o número de quadros perdidos, mantendo a decodificação sincronizada, sendo que, para isso, mais complexidade é inserida na codificação, o que aumenta a complexidade do sistema.

2.4. Padrões de codificação de vídeo

São muitos os padrões de codificação de vídeo existentes hoje. Muitos deles assemelham-se e são concorrentes, enquanto outros possuem objetivos específicos. Os dois grupos de estudos fundadores dos padrões mais utilizados no mundo são o ITU-T e o MPEG, criado pela ISO. São eles:

- MJPEG (*Motion Joint Picture Expert Group*): esse grupo é o mesmo que trabalha no JPEG (padrão para codificação de imagens estáticas). Esse padrão trata os vídeos como um conjunto de imagens estáticas JPEGs seqüenciadas. Esses vídeos são mais utilizados quando se faz necessário o acesso aleatório a cada imagem, por exemplo, em aplicações de edição de vídeos. Para sua transmissão, faz-se necessário um canal de, no mínimo, 8 a 10Mbps.

- ISO MPEG-1: criado originalmente para produzir vídeos de alta qualidade para serem armazenados em CD-ROM. Logo depois, sua utilização foi bastante estendida. Esse padrão não é capaz de atingir altas taxas de compressão e sua qualidade foi superada pelo MPEG-2, que substituiu o MPEG-1 alguns anos depois. Uma rede necessita de aproximadamente 2Mbps para transmitir vídeos ao vivo nesse padrão.

- ITU-T H.261: esse padrão foi baseado no MPEG-1, possuindo os mesmos mecanismos de codificação e decodificação, e teve o mesmo destino do padrão da ISO.

- ISO MPEG-2: criado para produzir vídeos de alta qualidade para larga difusão, como o HDTV e o cinema digital. Sua qualidade é bem superior a do MPEG-1 e atinge maiores níveis de compressão. Esse padrão produz vários níveis de

vídeo que vão dos que necessitam de 3Mbps para serem transmitidos, até os que precisam de aproximadamente 100Mbps.

- MPEG-4: projeto criado inicialmente para prover vídeos a todos tipos de aplicações. As três áreas de atuação mais fortes e consagradas do MPEG-4 são a televisão digital, aplicações gráficas interativas (elementos sintéticos) e multimídia interativa para Internet, que tem sido o maior sucesso desse padrão, popularizando ainda a distribuição de vídeos pelo mundo através da grande rede. Uma rede com um canal de apenas 64Kbps é capaz de transmitir um vídeo de qualidade no formato QCIF utilizando esse padrão (ISSO/IEC, 2002). Pereira (2002) explica o propósito geral do padrão MPEG-4 e quais seus objetivos específicos.

- ITU-T H.263: esse padrão possui os mesmos mecanismos de codificação do MPEG-4, sendo que este último possui mecanismos sofisticados para codificação dos vídeos, que interpretam os objetos que compõem as imagens, possibilitando a edição e a manipulação de objetos audiovisuais, naturais, ou não. O H.263 possui apenas o objetivo de alcançar altos níveis de compressão e controle de erros. Versões mais novas desse padrão estão em estudo, como é o caso do H.263+ , do H.263++ e do H.26L. A descrição da primeira versão do padrão ITU-T H.263 consta em ITU-T (1996). A versão 2 do padrão, conhecida como H.263+, foi oficialmente aprovada em janeiro de 1998 (ITU-T, 1997). Este padrão, uma extensão do H.263, provê 12 novos modos de negociação e novas funcionalidades. Esses modos e funcionalidades aumentam a performance da compressão, permitem o uso de fluxo de bits escaláveis, melhoram a performance através de redes de comutação de pacotes, e suportam outras novas funcionalidades, como tamanhos personalizados de imagens, entre outras (Côté et. al., 1998).

2.5. Motion Progress Expert Group – 4 (MPEG-4)

O padrão MPEG-4 propõe-se a contemplar os objetivos de um amplo escopo de aplicações, ao mesmo tempo em que busca satisfazer tanto as necessidades dos autores, dos provedores de serviço, quanto dos usuários finais.

Para alcançar esses objetivos, o MPEG-4 provê meios padronizados para representar objetos de mídia (*media objets*) visuais ou audiovisuais, naturais ou sintéticos, assim como prevê o relacionamento entre vários objetos de mídia formando objetos compostos.

Os objetos de mídia também podem ser multiplexados com outros dados que acrescentam ao sistema, sincronização, segurança e proteção contra erros.

Objetos diferentes podem ser unidos formando cenas complexas que, por exemplo, podem ser editadas, visualizadas de diferentes posições, objetos podem ser inseridos ou removidos de uma cena, assim como, posições de objetos podem ser modificadas.

Nas transmissões, os objetos de mídia são transmitidos juntamente com dados de configuração que podem ser utilizados, por exemplo, para determinar os recursos do decodificador, assim como, trazer informações que auxiliem a sincronização da transmissão. Essa característica é especialmente interessante para redes sem fio, visto que nesse tipo de rede os recursos são escassos (pouca largura de banda, número de usuários limitados, taxas de transmissões baixas e inconstantes). Os dados de configuração trazem informações que podem ser utilizadas para atingir a Qualidade de Serviço (QoS) almejada para a transmissão (ex.: taxa máxima de bits, taxa de erro de bit, prioridade do vídeo etc.). Tanto um servidor, quanto um cliente de vídeo podem utilizar esses dados de modo a otimizar as transmissões (ISO/IEC, 2002).

Buscando abranger o maior número de aplicações possíveis, o MPEG-4 é separado em perfis (*profiles*), de forma tal que o sistema apenas incorpore os perfis necessários para as determinadas aplicações que o mesmo deseja executar.

Alguns desses perfis são bem aplicados à transmissão de vídeo através de redes sem fio, como é o caso dos perfis (ISO/IEC, 2002):

- *Simple Visual Profile*: provê codificação eficiente de objetos de vídeos retangulares de forma eficiente e resistente a erros. Ideal para aqueles sistemas mínimos de recursos, como celulares;

- *Simple Scalable Visual Profile*: adiciona ao *Simple Visual Profile* a possibilidade de codificar objetos, temporal e espacialmente escaláveis. Isso é útil para aplicações que desejam prover vídeos com níveis de qualidade variando de acordo com a taxa de bits disponível, ou com as limitações de recursos do decodificador, como é o caso da Internet;

- *Core Visual Profile*: adiciona ao *Simple Visual Profile* suporte à codificação de objetos de forma arbitrária. É recomendado para aplicações que desejam interatividade de conteúdo relativamente simples.

A Tabela 2-4, apresentada em Barret (2000), mostra o desempenho de três perfis em seus diferentes níveis de complexidade. O perfil *Main Visual Profile*, mesmo não sendo ideal para canais de baixa transmissão possui, em seu nível mais simples (*Level 2*), uma possibilidade de transmitir vídeos CIF em bandas de 2Mbps, podendo codificar até 16 objetos de mídia. As resoluções ITU-R 601 e 1920x1088 são mais utilizadas para cinema e televisão.

Pelo que podemos ver pela Tabela 2-4, o mínimo para transmitir vídeos MPEG-4 de resolução QCIF é um canal de 64Kbps, onde é possível codificar até 4 objetos de mídia, uma banda que muitas redes sem fio possuem.

Perfis e níveis		Tamanho típico de uma cena	Taxa de bits (bits/s)	Número máximo de objetos
Simple VIsual Profile	L1	QCIF	64k	4
	L2	CIF	128k	4
	L3	CIF	384k	4
Core Visual Profile	L1	QCIF	384k	4
	L2	CIF	2M	16
Main VIsual Profile	L2	CIF	2M	16
	L3	ITU-R 601	15M	32
	L4	1920x1088	38,4M	32

Tabela 2-4: Características dos perfis MPEG-4 para cada nível.

2.6. Sumário

Vimos nesse capítulo que a transmissão de vídeo ao vivo através de redes sem fio é um tipo de transmissão peculiar que necessita de atenção especial. O dilema entre inserir altas taxas de compressão, aumentando em contrapartida a sensibilidade do vídeo aos erros, e torná-los mais resistentes, mas menos comprimidos, é algo que deve ser bem administrado. A escolha entre as técnicas de proteção contra erros é outra dificuldade.

São várias as técnicas e ainda há muito a se fazer para alcançar uma transmissão de qualidade sem inserir muita complexidade ao sistema. Algumas dessas técnicas foram apresentadas aqui e outras foram apenas referenciadas.

A codificação é a etapa mais importante nessa transmissão. São vários os padrões existentes, com seus próprios propósitos e características. Dos padrões de codificação de

vídeo existente, o MPEG-4 é o mais adequado para nossos objetivos nesse trabalho, visto que o MPEG-4, além de ser um dos mais eficientes para aplicações em ambientes sem fio, possui muitos usuários pelo mundo, e seu código de teste é aberto e gratuito.

Capítulo 3

3. Redes sem Fio e a Tecnologia Bluetooth

Hoje em dia há tecnologias de redes sem fio para muitos tipos de aplicação. Este capítulo apresenta uma classificação para redes sem fio, enfatizando as tecnologias mais utilizadas em cada classe de rede. Para cada aplicação e objetivo almejado existe uma tecnologia mais adequada. Saber que tecnologia utilizar é um dos principais passos a ser tomado no desenvolvimento de qualquer sistema.

As redes Bluetooth formam um tipo de rede sem fio diferente de outros padrões existentes. Nessa tecnologia as redes são formadas e as informações são trocadas sem que necessariamente os usuários tomem qualquer atitude. A tecnologia Bluetooth possibilita a criação de um novo tipo de rede sem fio, as *Wireless Personal Area Networks* (WPAN), ou seja, são redes que conectam dispositivos que estão ao redor de uma pessoa, como telefones celulares, relógios, rádios, etc.

Na Seção 3.2 apresentamos as tecnologias para redes sem fio mais utilizadas atualmente, situando cada tecnologia em sua classe apropriada. Também é apresentada uma tabela comparando a taxa de transmissão e o alcance das tecnologias.

A tecnologia Bluetooth, suas principais características e as peculiaridades de uma transmissão de vídeo através desse tipo de rede são mostradas na Seção 3.3. A pilha de protocolos comparada com a pilha do modelo de referência OSI, bem como aspectos de segurança dos dados nessa tecnologia são apresentados nessa seção. A conclusão do capítulo está na Seção 3.4.

3.1. Tecnologias de Redes sem Fio

Há várias formas de classificar redes sem fio. A que achamos mais interessante é a que separa as redes sem fio por suas arquiteturas (Green, 2000, p. 386). Nessa classificação existem três tipos de redes sem fio. O primeiro tipo é formado pelas redes móveis ou

portáteis, aquelas que utilizam o conceito de mobilidade, como é o caso das redes de telefonia celular. O segundo tipo é o chamado prédio-a-prédio (*building-to-building)*, onde conexões sem fio são estabelecidas entre prédios através de enlaces de rádio, infravermelho ou satélite, não havendo aqui a mobilidade. Como um terceiro tipo, podemos apresentar as "redes de mesa" (*desktop*), onde computadores de mesa, *notebooks* e periféricos se comunicam sem a utilização de fios. Devemos também deixar claro que, numa rede sem fio, esses três tipos de arquitetura também podem ser utilizados em conjunto.

A telefonia celular é sem dúvida o tipo de comunicação sem fio mais utilizado atualmente. As tecnologias de telefonia celular têm evoluído bastante nos últimos anos. Apesar de possuir um propósito inicial de apenas transmitir voz, essa tecnologia tem avançado também em outros mercados, unindo-se, cada vez mais, ao grande mercado da transmissão de dados e da Internet. A primeira e a segunda geração (ver Tabela 3-5) não apresentam muita capacidade de transmissão, mas a terceira geração, ou 3G, como é conhecida, tem mostrado ser capaz de transmitir e receber som e imagem de boa qualidade.

O mercado das redes de mesa hoje é dominado pelos padrões da IEEE 802.11a (http://grouper.ieee.org/groups/802/11/index.html) e IEEE 802.11b. Esses padrões foram criados com o intuito de possibilitar a comunicação entre uma LAN (*Local Area Network*) convencional e dispositivos sem fios, como *notebooks* e impressoras, e a criação de uma LAN totalmente sem fio, chamada de "*Wireless* LAN" (WLAN). O padrão 802.11a é capaz de transmitir a 6, 12 ou 24Mbps, já o 802.11b transmite 11, 22 ou até 54Mbps (www.broadcom.com), dependendo do modo de transmissão escolhido.

Outra tecnologia para redes de mesa destinada ao lar é o HomeRF, uma tecnologia que foi criada para evitar problemas com toda a fiação dos vários equipamentos que os lares tendem a possuir, tais como computadores, televisores, DVDs e eletrodomésticos em geral. A HomeRF é uma rede fácil de configurar, utiliza a tecnologia do IEEE 802.11, sendo que com um custo menor, e também pode ser caracterizada como uma WLAN.

A transmissão através de luz infravermelha também é bastante utilizada para formar "redes de mesa" com taxas acima de 1Mbps. Essa transmissão é direcional e não penetra obstáculos, isso obriga os receptores a sempre estarem apontando, sem nenhum obstáculo, para o transmissor. Geralmente as transmissões dessa natureza são feitas utilizando os protocolos da associação IrDA (*Infrared Data Assossiation*). A maior utilização desse padrão (http://www.irda.org) é a comunicação entre dispositivos próximos (2 metros, no máximo) e

de baixa taxa de transmissão. Esse protocolo também é utilizado para formar redes locais, embora não seja muito comum.

Comunicação sem fio entre prédios é bastante comum pela dificuldade e pelo custo de fazer uma transmissão entre prédios distantes através de fios. O mais comum nesses casos é a transmissão através de antenas direcionais de rádio. Esse tipo de comunicação transmite uma grande quantidade de dados. Atualmente a utilização dessa tecnologia tem aumentado, devido principalmente à insatisfação de muitos clientes com a baixa qualidade do acesso discado à Internet. Um padrão em desenvolvimento para esse tipo de aplicação é o IEEE 802.16 WirelessMAN. Seu principal foco é criar redes metropolitanas sem fio utilizando eficientemente a banda entre 10 e 66GHz (Eklund, et. al., 2002). Sem a necessidade de utilizar cabos, esse padrão foi desenvolvido buscando substituir outros padrões de transmissão de altas taxas de bits, como o DSL e o ATM, por exemplo, pois estes possuem a dificuldade da instalação de cabos.

O último tipo de rede é caracterizado como uma "*Wireless Personal Area Network*" (WPAN). Este surgiu com a criação das redes *Bluetooth* (http://www.bluetooth.org) criadas, a priori, para conectar qualquer dispositivo, por menor que seja, a outro, com baixo custo e consumindo pouca energia. Nessa tecnologia as redes se formam apenas com a aproximação dos dispositivos, sem que os usuários tomem maiores atitudes. Elas também são chamadas de redes *ad hoc*. Nos dias de hoje, a WPAN está em processo de padronização e em breve se tornará o padrão IEEE 802.15 (http://ieee802.org/15). A tecnologia Bluetooth tem sido a base para a formação desse padrão.

Tecnologia	Capacidade de transmissão	Alcance máximo	Outras características
IrDA	9600bps até 4Mbps	20cm ou 2m	• Direcional • Problemas com obstáculos
IEEE 802.11b	1 ou 2Mbps	30m	• Conecta dispositivos a LANs • Computadores *notebooks* e periféricos
IEEE 802.11a	5Mbps	30m	
HomeRF	1 ou 2Mbps	50m	• Criados para substituir os fios de uma casa.
Celular: 1ª. e 2ª. gerações	8 ou 13Kbps	Vários quilômetros	• Apenas voz e textos • 2a. geração inclui o WAP
Celular: 3ª. geração	2Mbps	Vários quilômetros	• Transmissão de vídeo e imagem
Bluetooth	0,7Mbps	10 ou 100m	• Baixo custo • Pequenas dimensões • Baixo consumo de energia

Tabela 3-5: Comparativo entre as tecnologias de redes sem fio.

3.2. A Tecnologia Bluetooth

"Bluetooth é uma tecnologia de transmissão de dados e voz via rádio de baixo custo, baixo consumo e curto alcance, onde as pessoas não precisam mais conectar, plugar, instalar, habilitar ou configurar nenhum equipamento" (Bray, et. al., 2001, p.1). Opera na banda ISM (*Industrial, Scientific and Medical*) em 2,45GHz, e seu objetivo é substituir os fios que estão sendo utilizados para conectar os mais diversos equipamentos.

Uma decisão que deve ser tomada quanto à especificação de uma rede sem fio é sobre a sua área de cobertura. Um dispositivo Bluetooth apresenta um transceptor de curto alcance, mas que, devido a isso, necessita de menos potência, consome menos energia e é bem menor, podendo ser instalado em canetas, PDAs, celulares etc.

Essa decisão por tamanhos e custos reduzidos sacrificou a capacidade de transmissão desses transceptores, os quais chegam no máximo a 723,32Kbps[4] numa transmissão unidirecional. Essa baixa taxa de bits é suficiente para muitas aplicações, porém dificulta aplicações mais "pesadas" de serem executadas, como a transmissão de vídeo, por exemplo.

[4] Para nosso trabalho estamos sempre nos baseando no padrão Bluetooth versão 1.1B.

Bluetooth é uma rede *ad hoc*, aquela onde sua arquitetura se forma espontaneamente apenas com a aproximação dos dispositivos. Uma rede Bluetooth possui um esquema de configuração transparente, caracterizada pela formação de grupos de transmissão (*piconets*) apenas com a aproximação de dispositivos que desejam conectar-se uns aos outros. Da mesma forma, quando dispositivos se distanciam, a rede é desfeita, isso tudo sem a necessidade dos usuários desses dispositivos tomarem conhecimento.

A tecnologia também usa a computação oculta (Miller, 2001, p. 44), que torna as redes Bluetooth ainda mais interessantes. Uma rede Bluetooth possibilita que dispositivos executem atividades com autorização, entretanto sem nenhuma ação do usuário. Um usuário de um celular pode receber as mais novas promoções dos seus produtos favoritos assim que ele passar em frente à loja onde ele fez compras uma certa vez, ou então, uma câmera de vídeo em um sistema de vigilância pode enviar um alerta para o celular do vigilante se uma das câmeras for desativada.

3.2.1. PILHA DE PROTOCOLOS

O padrão Bluetooth é organizado em camadas, baseando-se no modelo OSI. Na Figura 3-2 podemos observar como a pilha de protocolos está organizada.

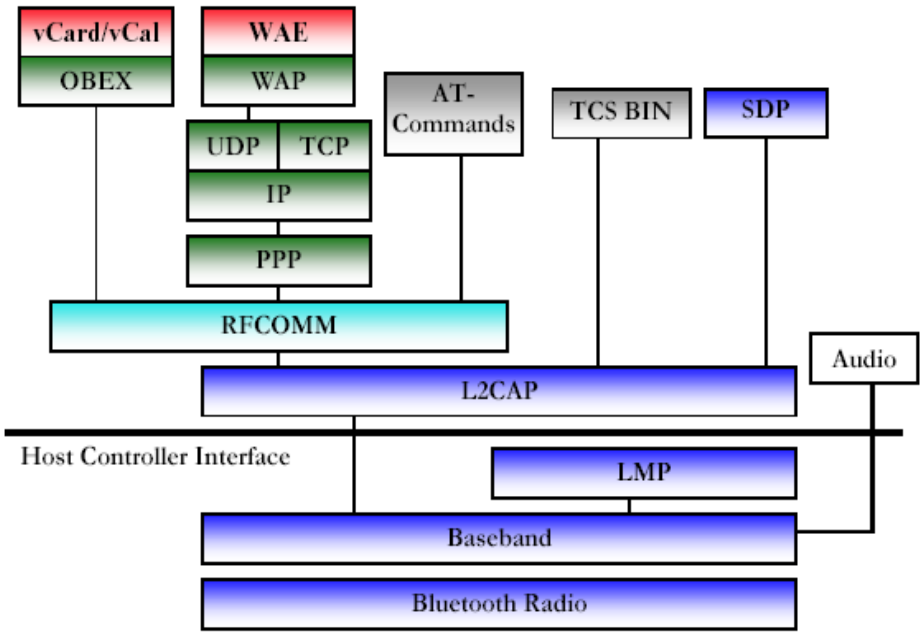

Figura 3-2: Pilhas de Protocolos Bluetooth.

A linha horizontal denominada *Host Controller Interface* separa os protocolos destinados às camadas físicas (inferiores à linha horizontal) dos protocolos destinados à aplicação (superiores à linha).

Os protocolos SDP (*Service Discovery Protocol*), L2CAP (*Logical Link Control and Adaptation*), LMP (*Link Management Protocol*), *Baseband* e *Bluetooth Radio* formam o grupo chamado *Core Protocols*, o coração da arquitetura, sem os quais um dispositivo não pode ser caracterizado como Bluetooth. Os protocolos OBEX, WAP, UDP, TCP, IP e PPP são os protocolos adotados pelo Bluetooth, buscando uma maior compatibilidade com outros padrões existentes atualmente. O protocolo RFCOMM (*Radio Frequency Communication*) faz uma emulação de uma comunicação serial RS-232. As aplicações são posicionadas (vCard/vCal e WAE) no topo da pilha de protocolos (Mettälä, 1999).

Os protocolos *Bluetooth Radio*, *Baseband* e LMP são essenciais na manutenção dos enlaces. O *Bluetooth Radio*, juntamente com o *Baseband*, forma o equivalente à camada física do modelo OSI (Bray, et. al., 2000, p.7). O protocolo LMP é responsável pela configuração e controle dos enlaces (Bluetooth SIG, 2001).

O protocolo RFCOMM é utilizado para simular uma comunicação serial entre os dispositivos, já o protocolo L2CAP dá suporte à multiplexação de conexão, possibilitando um mesmo dispositivo manter mais de uma conexão com dispositivos diferentes simultaneamente. O protocolo SDP é implementado para que os dispositivos possam localizar quais os serviços estão disponíveis na rede a cada instante. Sempre que um dispositivo deseja utilizar algum serviço de outro dispositivo, ele aciona o SDP, o qual pesquisa quais os serviços que os outros dispositivos da *piconet* estão disponibilizando, procurando apenas aqueles que são compatíveis com o dispositivo pesquisador.

A Figura 3-3 relaciona os protocolos Bluetooth com o modelo de referência OSI. Essa relação está definida em Bray, et. al. (2000, p.7).

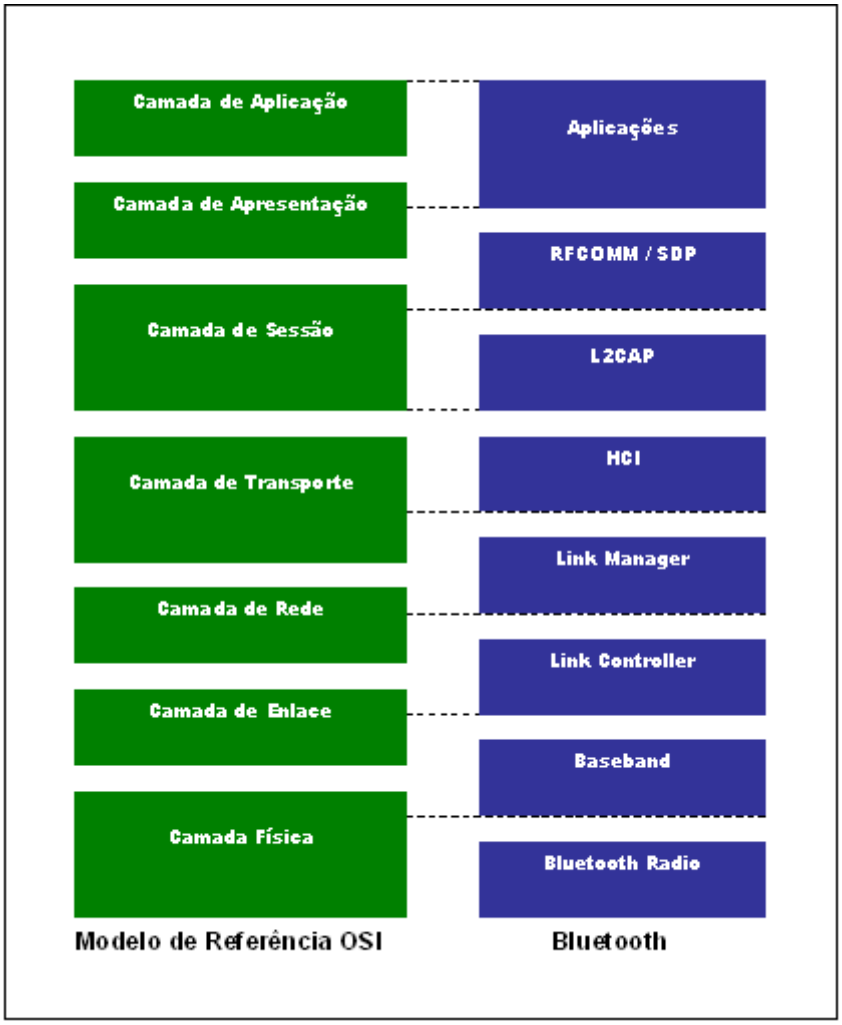

Figura 3-3: Modelo de Referência OSI e Bluetooth.

3.2.2. SEGURANÇA EM REDES BLUETOOTH

Comunicações em redes Bluetooth geralmente são formadas por várias *piconets* (dispositivos de uma área que seguem uma mesma regra de sincronização) situadas num mesmo ambiente. Dentro de uma *piconet*, o tempo é dividido em janelas de tempo de 625µs. Para evitar que *piconets* interfiram umas às outras, as redes Bluetooth usam uma técnica chamada de salto de freqüência (*Frequency-Hopping-Spread-Spectrum*), onde dispositivos numa mesma *piconet* mudam a freqüência de transmissão e recepção de seus transceptores de forma sincronizada (Mettälä, 1999). Todos dispositivos de uma mesma *piconet* estão sincronizados segundo uma seqüência de freqüências definida pseudo-aleatoriamente por um dispositivo, chamado de *master*.

Cada dispositivo tem o seu alcance máximo (10m ou 100m), como podemos ver na Figura 3-4, em que um dos dispositivos não faz parte nem da *Piconet* A, nem da *Piconet* B, isso porque ele não está na área de alcance nem do *Master* A, nem do *Master* B. Uma *piconet*

(ver Figura 3-4) é formada por um dispositivo mestre (*master*) e por dispositivos escravos (*slaves*). Mais de um *master* podem formar uma rede maior (*scatternet*), formada por um conjunto de *piconets*, onde cada *piconet* segue a sincronização determinada pelo seu *master*, e o *master* uma *piconet* deve ser *slave* da outra *piconet* (Figura 3-4).

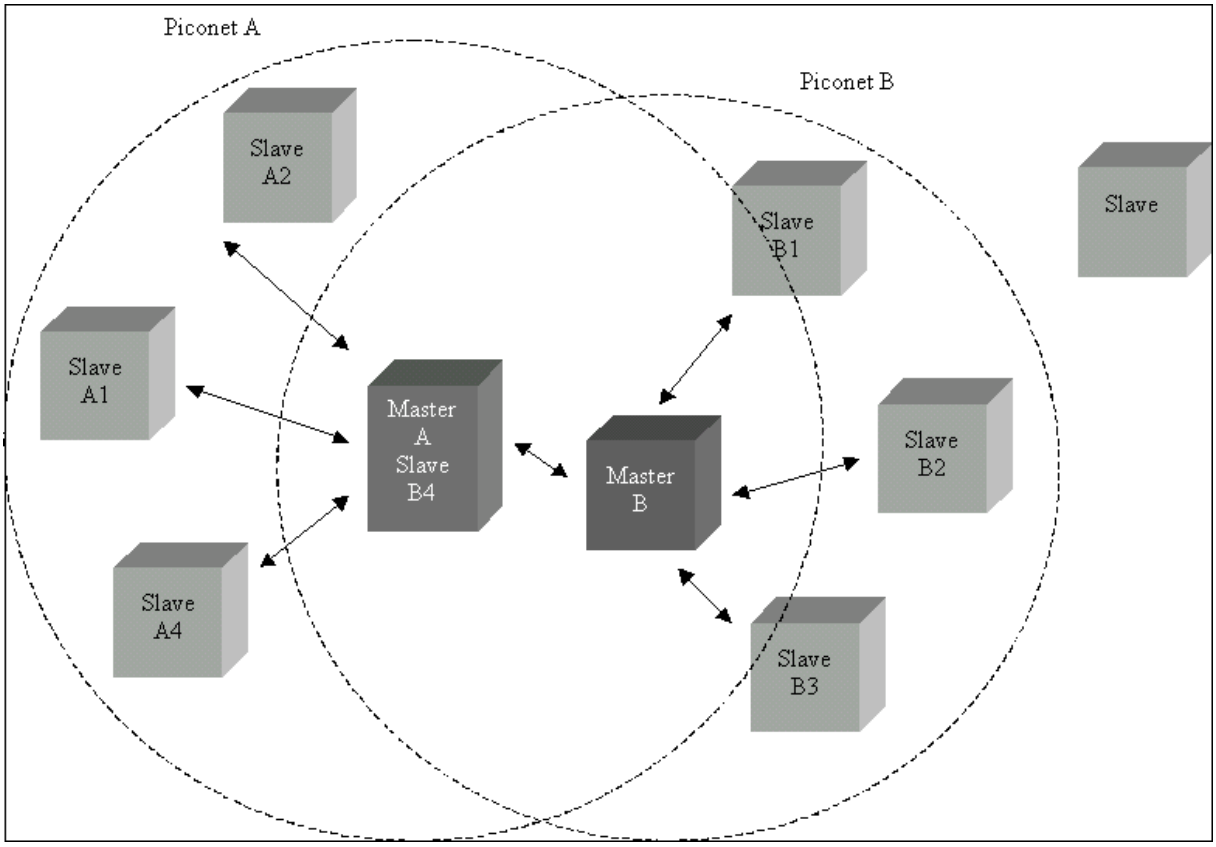

Figura 3-4: Duas *piconets* em regiões próximas formando uma *scatternet.*

Com isso, dispositivos Bluetooth, mesmo estando numa mesma sala, apesar de serem de *piconets* diferentes, não sofrem interferências, nem conseguem ouvir a comunicação da outra *piconet*, uma vez que os saltos de freqüência são definidos pseudo-aleatoriamente em cada *piconet*, dando mais segurança ao sistema. Desta forma, fica difícil um transgressor inserir ondas de RF na mesma freqüência dos dispositivos para atrapalhar o funcionamento do sistema. Para isso, o transgressor teria que encontrar em que seqüência pseudo-aleatória o *master* e seus *slaves* estão saltando suas freqüências e para quais valores.

Mesmo se os saltos de freqüência não adiantarem, o Bluetooth ainda conta com a possibilidade de utilizar um forte esquema de criptografia a fim de dificultar que elementos não autorizados espionem o que está sendo transmitido ou transmitam dados fazendo-se de usuário autorizado (*middleman attack*).

Em Jakobsson e Wetzel (2001) são apresentadas falhas de segurança na especificação Bluetooth em sua versão 1.0B. Este artigo menciona que a especificação, da forma que era feita, possibilitava a falsificação de identidade de dispositivos, ataques por localização geográfica de dispositivos e a espionagem de dados que transitam na rede. Esse artigo apresenta que a espionagem dos dados e a falsificação de identidade são possíveis através da procura exaustiva dos códigos trocados entre os dispositivos que identificam tanto a *piconet* quanto os dispositivos que fazem parte da mesma. Na arquitetura Bluetooth cada dispositivo possui um endereço de identificação, sendo único no mundo. No momento do estabelecimento da conexão esses endereços são trocados e é montado um endereço de rede que será utilizado por todos os dispositivos da *piconet*.

A versão 1.1B da tecnologia, na qual nos baseamos para execução desse trabalho (Bluetooth SIG, 2001), apresenta modificações, de maneira que amenizam esses problemas de segurança e ainda nenhum trabalho foi publicado mencionando alguma falha de segurança nessa nova versão. Em Bluetooth Security Expert Group (2002) há dicas de como proteger ainda mais as *piconets* contra invasões e quebras de privacidade, além de apresentar procedimentos de segurança que devem ser tomados por várias aplicações. Esse artigo não recomenda o uso dos endereços de identificação diretamente, todavia o uso de endereços de identificação combinados com outros códigos. O problema é que, quando o endereço de identificação é compartilhado com os dispositivos autorizados, esses dispositivos permanecem com o endereço, mesmo depois de serem desautorizados, podendo, então, utilizar esse endereço em ataques às conexões desse dispositivo em outras oportunidades. Utilizando como chave sempre o endereço do dispositivo combinado com outros códigos, esse tipo de fraude fica mais difícil de ocorrer.

Em Xydis e Simon (2002) é apresentado um comparativo entre a segurança nas redes Bluetooth e IEEE 802.11, apontando falhas de segurança presentes no segundo padrão não existentes no primeiro.

3.2.3. CONEXÕES SÍNCRONAS E ASSÍNCRONAS

As redes Bluetooth utilizam dois tipos de conexões bem definidas, uma síncrona e outra assíncrona. Nas conexões síncronas, *Synchronous Connection-Oriented* (SCO), os dispositivos trocam pacotes de forma periódica. É assegurado que cada dispositivo receberá seus pacotes a cada período.

Já nas conexões assíncronas, *Asynchronous Connection-Less* (ACL), os dispositivos enviam pacotes aos outros em qualquer ordem. Um servidor de vídeo, por exemplo, pode deixar de enviar um pacote de vídeo a um cliente para beneficiar outro que está com sua qualidade de recepção prejudicada pelo atraso de pacotes.

Em conexões SCO um dispositivo *master* pode ter no máximo três clientes simultâneos, contra sete das conexões assíncronas que são conexões obrigatoriamente bidirecionais, e transmitem a 64Kbps (Bray, et. al., 2001, p. 93).

Para transmissões de vídeo, utilizar conexões SCO não é apropriado, pois vídeos comprimidos são transmitidos em VBR-RT (*Variable Bit Rate – Real Time*), já que os quadros dos vídeos possuem tamanhos bastante variados. Em conexões SCO a taxa de bits é constante, mais adequada para transmissões CBR (*Constant Bit Rate*). Além disso, as taxas de transmissão das conexões SCO são reduzidas, e o número de conexões entre dispositivos limita-se a três. A solução é utilizar conexões ACL, onde os dispositivos *master* podem procurar ao máximo evitar a perda de sincronismo. Com esse tipo de conexão, um maior número de conexões é permitido e maiores taxas de bits possibilitam uma transmissão de vídeo com qualidade.

Visando aumentar as taxas de transmissão do sistema, podem-se utilizar conexões ACL de forma unidirecional, visando sempre o sincronismo dos dados, trabalhando para que os dispositivos recebam os dados em taxas de bits adequadas. A Figura 3-5 mostra um exemplo de um servidor de vídeo com dois clientes utilizando conexões ACL.

Nesse exemplo um dispositivo *master* (servidor) transmite um vídeo para dois *slaves* (clientes) simultaneamente. Em cada janela de tempo apenas um dispositivo é autorizado a transmitir. Quando o *master* envia um pacote, os *slaves* ocupam uma pequena faixa de tempo, identificando se são os destinatários da mensagem, se for o caso, o *slave* receptor envia uma pequena mensagem de reconhecimento para o *master*. O *Master* começa transmitindo apenas para o *Slave 2*, por ainda não ter dados para enviar para o *Slave 1*, só a partir da quinta janela de tempo, o *Master* começa a transmitir para o *Slave 1*, e a dividir o seu tempo entre esses dois dispositivos com mais igualdade. O *Master* busca sempre manter o máximo de sincronismo, como é mostrado no exemplo, quando ele identifica que um de seus *slaves* não recebeu dados suficientes, ele pode aumentar o tamanho do pacote para enviar mais dados ao *slave* defasado, como é o caso do *Slave 1*, que recebe um pacote maior no fim do exemplo apresentado na Figura 3-5.

Em cada janela de tempo uma freqüência é utilizada. Na janela 0 a freqüência é f(0), na janela 1 a freqüência é f(1) etc. Como apenas os dispositivos da *piconet* sabem que valores são esses, apenas eles são capazes de receber os pacotes transmitidos. Cada pacote contém um código que identifica a *piconet* e para quem é o pacote que está sendo transmitido, além de outras informações importantes.

O mais importante numa transmissão de vídeo não são quantos dados um cliente de vídeo recebe, nem com que velocidade eles são recebidos, o mais importante é que os clientes recebam os vídeos nas taxas de bits adequadas para cada momento. Se em determinado instante, o vídeo demandar baixa taxa de bits, então é inútil utilizar muita banda para esse vídeo, porém se o vídeo demandar de uma alta taxa de bits, o sistema deve se preocupar em enviar os dados para esse cliente da melhor forma possível.

Em Bluetooth o que se tem a fazer é controlar para que nenhum *slave* passe muito tempo sem receber pacotes, e os pacotes devem ser grandes o suficiente para conter todos os bits que o *slave* esteja necessitando naquele momento.

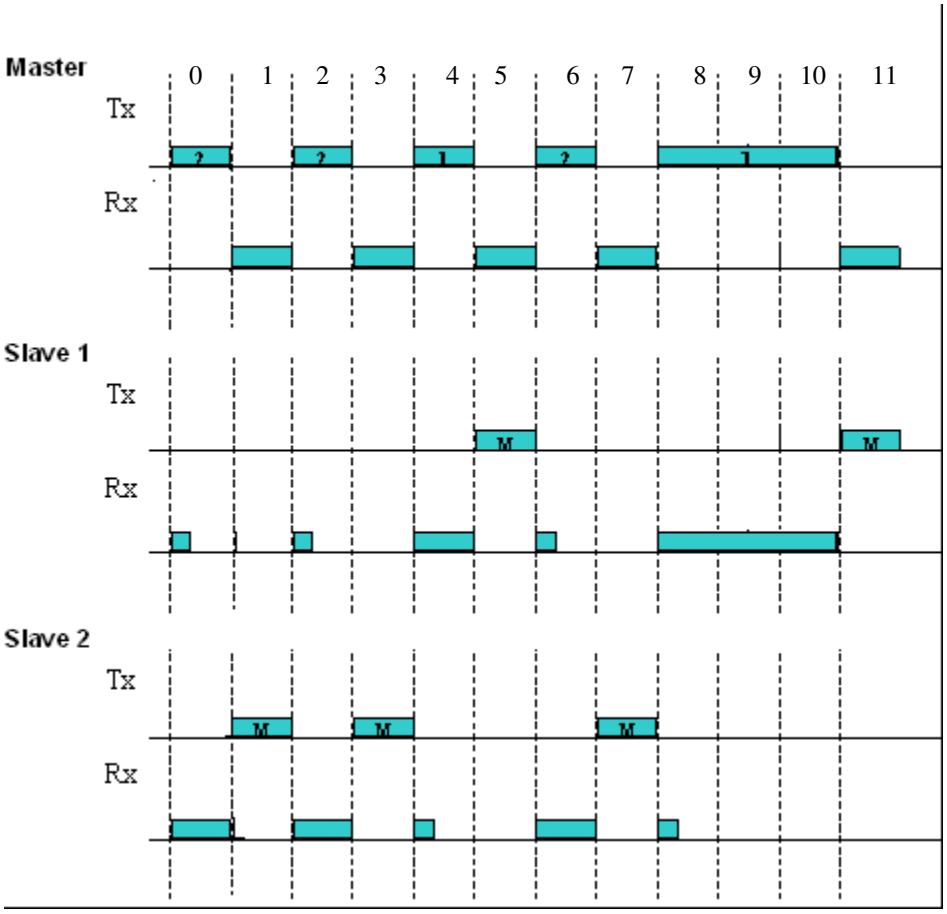

Figura 3-5: Divisão do tempo de um dispositivo *Master* transmitindo vídeo para dois *Slaves* numa conexão ACL.

3.2.4. TIPOS DE PACOTES

A especificação Bluetooth prevê a utilização de vários tipos de pacotes, dentre os quais estão pacotes de controle e pacotes de dados. Para conexões ACL é possível utilizar sete tipos diferentes de pacotes de dados que se diferenciam pelos seus tamanhos e por suas capacidades de proteção contra erros.

A Figura 3-6, mostra a estrutura básica dos pacotes Bluetooth. O *Access Code*, além de identificar o início de um pacote, é utilizado para endereçar o pacote a um dispositivo específico. Os *slaves* detectam a presença de um pacote casando o *Access Code* do pacote com o do que seu *master* o enviou na formação da *piconet*. Ele pode ter 72 ou 68 bits. O *Header* contém todas informações de controle associadas ao pacote e ao enlace, e o *Payload* contém os dados úteis transmitidos para o dispositivo e pode variar de 0 (pacotes de controle) a 2745 bits.

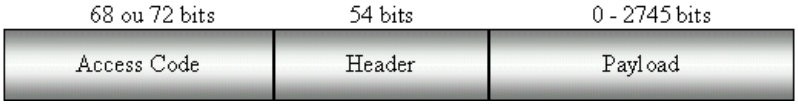

Figura 3-6: Estrutura dos pacotes Bluetooth

Como podemos observar na Tabela 3-6, os pacotes podem ocupar 1, 3 ou 5 janelas de tempo. Os pacotes que DM1 e DH1 ocupam uma janela de tempo (0,625ms), os pacotes DM3 e DH3 ocupam três janelas de tempo (1,875ms) e os pacotes DM5 e DH5 ocupam cinco janelas de tempo (3,125ms). O tamanho dos pacotes é apresentado sempre em números ímpares para que os dispositivos que iniciem transmitindo em janelas de tempo ímpares continuem transmitindo em janelas ímpares e recebendo em janelas pares, assim como os dispositivos que transmitem em janelas pares permaneçam sempre desta forma (observar Figura 3-5).

Na Figura 3-5 podemos também observar que o tamanho das áreas de dados é um ponto determinante no desempenho da transmissão. Mesmo possuindo tamanhos diferentes, pacotes de um mesmo tipo ocupam sempre o mesmo número de janelas de tempo, o que faz com que eles sempre levem o mesmo tempo para serem transmitidos.

Os pacotes AUX1 são utilizados como controle, principalmente em momentos em que o *master* está com problemas para configurar as condições da transmissão.

Tipo de pacote	Taxa do código FEC na área de dados	Tempo de transmissão (ms)	Tamanho da área de dados (Bytes)	Taxa de transmissão max. (Kbps)	Janelas de tempo ocupadas
DM1	2/3	0,625	0-17	108,8	1
DH1	1	0,625	0-27	172,8	1
DM3	2/3	1,875	0-121	387,2	3
DH3	1	1,875	0-183	585,6	3
DM5	2/3	3,125	0-224	477,8	5
DH5	1	3,125	0-339	723,2	5
AUX1	1	0,625	0-29	185,6	1

Tabela 3-6: Tipos de pacote para conexões ACL. Fonte: (Bluetooth SIG, 2001).

Um esquema de ARQ (*Acknowledge ReQuest*) é utilizado para evitar a perda de pacotes. Esse tipo de esquema retransmite pacotes sempre que eles são perdidos, diminuindo significativamente a perda de informações durante as transmissões. Em contrapartida, a taxa de transmissão cai. Para diminuir a taxa de perdas de bits, sem diminuir muito a taxa de transmissão, a especificação Bluetooth possibilita a utilização de pacotes que possuem uma maior proteção interna contra perda de dados, diminuindo o número de retransmissões.

Todos os pacotes já possuem proteção em seus cabeçalhos pelo código FEC (*Forward Error Correction*) de 1/3 (a cada 3 bits, 2 são de redundância e apenas 1 é dado útil).Os pacotes DH são destinados a transmissões de grande quantidade de dados, mas com pouca proteção contra erros, já os pacotes DM possuem tamanhos reduzidos por também possuírem proteção FEC em suas áreas de dados com uma taxa de 2/3 (a cada 3 bits, 1 é de redundância e 2 são dados úteis). Nesses pacotes os dados apresentam um terço de redundância, em compensação, a cada grupo de quinze bits, o algoritmo é capaz de corrigir qualquer erro simples e detectar qualquer erro duplo, o que faz deles, pacotes bem mais tolerantes a erros. Para cada aplicação sendo executada em ambientes diferentes, deve haver uma escolha entre desempenho e robustez, pacotes mais robustos (DM) transmitem a menores taxas de bits. Segundo Zurbes (2000), o pacote DH5 é o tipo de pacote mais indicado quando se deseja alcançar altas taxas de transmissão em condições normais de interferência. Para ambientes onde as interferências são maiores, pacotes com maior taxa de proteção são mais indicados, como o DM5.

3.3. Sumário

De todos os padrões e tecnologias de redes sem fio existentes apresentados nesse capítulo, as redes Bluetooth são as que nos atraíram mais para a execução desse trabalho. Para

o tipo de sistema que desejamos aplicar, mesmo não apresentando uma especificação para transmissão de vídeo, a tecnologia Bluetooth oferece vantagens em termos de flexibilidade, praticidade e baixo custo.

Capítulo 4

4. Arquitetura do Sistema de Transmissão de Vídeo para Vigilância

Um sistema de transmissão de vídeo, para ser utilizado por sistemas de vigilância deve atender requisitos adicionais a outros sistemas de transmissão de vídeo. Quando se deseja transmitir vídeo para entretenimento, por exemplo, o mais importante é a qualidade e a continuidade do serviço, já em sistemas de vigilância, a segurança e a proteção do sistema são mais importantes.

Para a implementação de um sistema de transmissão de vídeo para vigilância utilizando Bluetooth, vários pontos devem ser levados em consideração. Como inserir a tecnologia Bluetooth ao sistema? Qual o papel dos codificadores e decodificadores MPEG-4? Quais são os elementos pertencentes ao sistema? Quais protocolos Bluetooth devem ser implementados? Essas e outras questões precisam ser respondidas antes da implementação do sistema. Esse capítulo apresenta respostas para essas e outras perguntas chaves para a concretização desse trabalho.

Na Seção 4.2 são apresentados os requisitos funcionais mais importantes para a execução de um sistema de transmissão de vídeo para vigilância, na Seção 4.3 são apresentados os elementos pertencentes ao sistema, na Seção 4.4. são relacionadas as vantagens de adicionar a esse sistema as características da tecnologia Bluetooth. Já na Seção 4.6 são apresentados os aspectos mais importantes, os quais devem ser levados em conta no momento da implementação do sistema. Os protocolos Bluetooth necessários para a implementação do sistema são relacionados na Seção 4.7, e a Seção 4.8 apresenta a arquitetura do sistema, mostrando como todos os elementos apresentados relacionam-se formando um só sistema. O capítulo é concluído na Seção 4.8.

4.1. Sistema de Transmissão de Vídeo para Vigilância

Um sistema de transmissão de vídeo para vigilância possui aspectos singulares em relação aos sistemas de vídeo para outras aplicações. Um sistema de vigilância possui requisitos especiais que não abrangem apenas a qualidade dos vídeos transmitidos, mas também requisitos de segurança e proteção. Podemos citar como requisitos de um sistema de transmissão de vídeo para vigilância:

Qualidade: vídeos devem chegar aos receptores com o máximo de fidelidade. A qualidade é o requisito mais importante de qualquer sistema de transmissão de vídeo. Cada aplicação tem seus requisitos próprios de qualidade. Um vídeo pode ser de excelente qualidade para um sistema de videoconferência, mas pode ser de péssima qualidade para uma TV digital. A análise da qualidade de um vídeo é uma tarefa extremamente subjetiva. *Para um sistema de vigilância a qualidade dos vídeos deve ser suficiente para possibilitar a detecção do ato e do infrator no momento da infração.*

Sincronismo: Deve haver uma defasagem mínima entre os vídeos capturados e os vídeos recebidos. Tratando-se de vídeo ao vivo, o tempo é um requisito importante. Para um sistema de vigilância é essencial que as imagens registradas pelas câmeras sejam referentes a um fato imediato. Atrasos entre a captura do vídeo e sua exibição podem inviabilizar o sistema. Quando um vigilante vê através das câmeras alguém numa sala, por exemplo, ele quer ter a certeza de ser um fato atual, e não uma imagem antiga do acontecimento.

Confiabilidade: Os vídeos devem ser transmitidos e recebidos apenas por dispositivos e pessoas autorizadas. A confiabilidade do sistema é outro requisito crucial. Se um infrator for capaz de gerar imagens espúrias, todo o sistema estará comprometido. Da mesma forma, apenas as pessoas autorizadas devem ser capazes de receber os vídeos. Para os bandidos, poder assistir as imagens registradas pelas câmeras seria uma grande ajuda. O sistema deve possuir um mecanismo eficaz de proteção contra esses tipos de ações.

Robustez: O sistema não pode ser desativado, total ou parcialmente, por usuários não autorizados. Ele deve poder se proteger e remediar contra ações capazes de desativá-lo. Por exemplo, se uma das câmeras parar de funcionar, o sistema deve ser capaz de detectar o problema e emitir alguma espécie de alerta para os responsáveis por ele. Ele também deve possuir mecanismos alternativos de funcionamento. Se houver um corte na energia elétrica, por exemplo, o sistema deve possuir fontes extras de energia para continuar funcionando mesmo assim.

4.2. Elementos do Sistema

Um sistema de transmissão de vídeo para vigilância é constituído de quatro elementos: servidores de vídeo, clientes de vídeo, sensores e elementos de controle. Vamos analisar cada um destes elementos.

4.2.1. SERVIDORES DE VÍDEO

Servidores de vídeo são aqueles que transmitem os vídeos para os outros dispositivos do sistema. Ele tanto pode ser um dispositivo de distribuição de vídeos, recebendo vídeos de várias câmeras diferentes e transmitindo para os demais dispositivos, como pode também ser uma simples câmera, capturando as imagens e as transmitindo para os clientes.

As transmissões de vídeo são sempre no sentido dos servidores para os clientes de vídeo. Se uma câmera está transmitindo vídeos para um dispositivo que irá distribuir para vários monitores, então para a primeira parte da transmissão, a câmera é um servidor de vídeo e o dispositivo é um cliente de vídeo, já para segunda parte da transmissão, o dispositivo é um servidor e os monitores são clientes de vídeo.

4.2.2. CLIENTES DE VÍDEO

Pode ser qualquer elemento capaz de receber vídeos, podendo armazená-los, repassá-los para outros dispositivos, apenas exibi-los, ou executar mais de uma dessas tarefas. Podem ser clientes de vídeo, monitores, distribuidores de vídeo, repositórios de vídeos, etc.

4.2.3. SENSORES

Sensores são dispositivos encarregados de detectar modificações no estado atual do ambiente. São específicos em analisar apenas um aspecto do ambiente (verifica se uma porta foi aberta, se uma janela foi quebrada, se houve presença de alguma pessoa numa sala, etc.). Sua principal função é enviar alertas para os elementos de controle quando alguma modificação no estado atual do ambiente for detectada.

4.2.4. ELEMENTOS DE CONTROLE

São os elementos capazes de modificar as configurações do sistema, acionando alarmes sonoros, travando portas, discando para algum número, etc. Eles recebem alarmes dos sensores e executam suas tarefas automática ou manualmente. Há tarefas as quais os

vigilantes não desejam que sejam executas sem a sua intervenção, como a discagem para a polícia, por exemplo.

Um alarme vindo de um sensor pode ativar vários elementos de controle, por exemplo, um alarme sonoro pode disparar quando uma janela for quebrada, ao mesmo tempo em que portas de segurança são travadas e um modem disca para a central de polícia.

Os sensores e os elementos de controle não fazem parte do escopo do nosso trabalho, já que estamos interessados apenas na transmissão de vídeo, nesse caso, nosso trabalho tratará apenas dos servidores e dos clientes de vídeo.

4.3. Transmissão de Vídeo Utilizando Bluetooth

As diferenças entre um sistema de transmissão de vídeo para vigilância convencional e um sistema com Bluetooth, o qual estamos propondo, estão nas novas funcionalidades que serão acrescentadas com a utilização de dispositivos Bluetooth. São elas:

Praticidade: Clientes e servidores de vídeo podem ser deslocados pelo ambiente com maior facilidade.

Dinamismo: Novos dispositivos podem ser adicionados à configuração do sistema ao entrarem no ambiente de vigilância, da mesma forma, podem ser retirados do sistema apenas pelo seu desligamento, ou pela sua separação física.

Agilidade: Dispositivos podem se comunicar sem a intervenção, mas com a autorização dos usuários.

Segurança: Sinais são mais difíceis de serem falsificados ou corrompidos, do que numa transmissão por radiofreqüência convencional;

4.4. Implementação do Sistema

Para a viabilização desse sistema alguns dispositivos devem ser equipados com *microchips* Bluetooth. Há no mercado *microchips* Bluetooth minúsculos. O menor disponível para compra que encontramos é o BCM 1013 da Broadcom (http://www.broadcom.com) lançado em setembro de 2001. Ele possui apenas 8mm x 8mm e incorpora todos os protocolos necessários para a implementação de aplicações Bluetooth. Com chips dessas dimensões praticamente qualquer dispositivo eletrônico pode ser equipado com as funcionalidades Bluetooth.

Em um sistema completo de transmissão de vídeo para vigilância o ideal é todos os elementos serem equipados com *chips* Bluetooth, servidores, clientes de vídeo, sensores e elementos de controle, entretanto como estamos preocupados apenas com a transmissão de vídeo, vamos adotar que apenas os servidores e os clientes de vídeo estão equipados com esses *chips*.

A configuração do sistema forma-se a medida em que dispositivos tentam comunicar-se com os outros. Por exemplo, a Figura 4-7 mostra uma configuração com dois monitores e cinco câmeras de vigilância. Nesse exemplo podemos notar que há uma interação entre a Piconet A e a Piconet B. Podemos adotar que o Master B foi o primeiro dispositivo a entrar no sistema, por isso optou por ser *master* logo de início. As câmeras de vídeo dessa *piconet* (Slave B) ao se aproximaram desse monitor, receberam uma solicitação para enviarem suas imagens, para isso, como o monitor é *master*, as câmeras devem ser *slaves*. Depois disso, um novo monitor foi inserido ao sistema (Master A / Slave B). Esse monitor, para receber as imagens de outras câmeras, denominou-se *master* e formou a Piconet A. As câmeras Slave A se aproximaram do monitor e logo foram tornando-se parte da Piconet A. Como o monitor da Piconet A estava dentro da região de alcance do monitor da Piconet B, eles foram configurados para trocar vídeos entre si, dessa forma o monitor da Piconet A entrou para a Piconet B como um *slave,* tendo o Master B como seu *master*. Nesse caso, tanto o monitor da Piconet A, quanto o monitor da Piconet B, podem enviar algumas das imagens recebidas para o monitor da outra *piconet*, como se fossem câmeras de vídeo. Como temos duas *piconets* juntas formando uma grande rede, dizemos que temos uma *scatternet.* Esse é um exemplo de como essa *scatternet* poderia ter se formado. Há várias maneiras de se formar uma mesma configuração de *scatternet*.

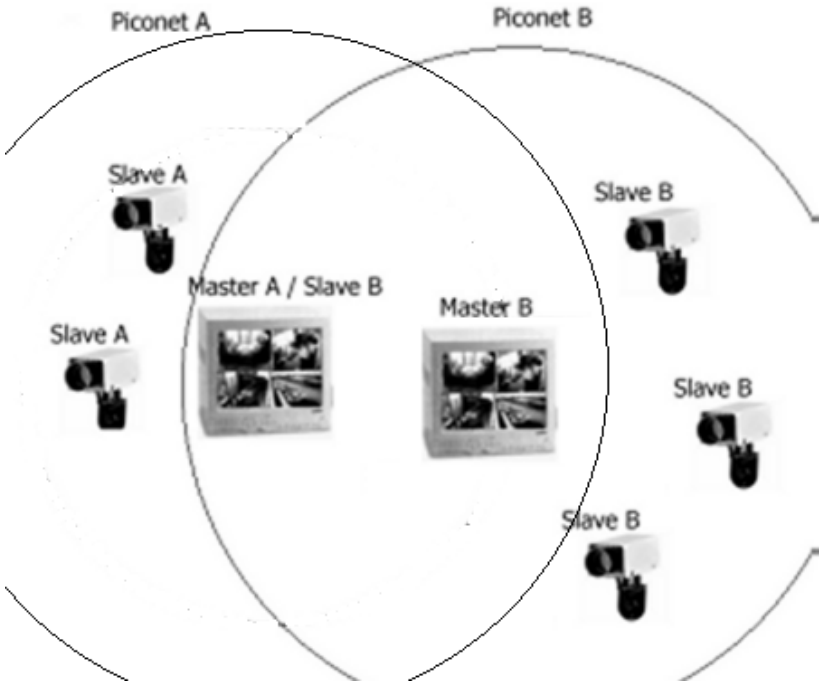

Figura 4-7: Exemplo de uma configuração para a *scatternet* do sistema com duas *piconets*.

No exemplo da Figura 4-7, as câmeras são sempre servidores de vídeo. Os monitores fazem o papel de clientes de vídeo quando estão recebendo os vídeos das câmeras, porém também fazem o papel de servidores de vídeo quando estão transmitindo imagens para outro monitor.

4.5. Protocolos Necessários

Mesmo a transmissão de vídeo em Bluetooth ainda não tendo sido especificada. Alguns trabalhos (*white papers*) nos mostram que essa transmissão é viável. A empresa Alphamosaic (http://www.alphamosaic.com) lançou um processador de vídeo chamado *VideoCore*®, anunciando que esse processador está incorporado com a tecnologia Bluetooth e o padrão MPEG-4 para transmitir vídeos com qualidade. A Sony (http://www.sony.com) já vende uma câmera de vídeo (*Sony DCR-IP7 Micro-MV Digital Video Camera*) com o chip Bluetooth, mas não menciona nada sobre vídeo ao vivo utilizando a tecnologia. No evento CeBit 2001 em Hanover na Alemanha, três empresas, Silicon Wave, AmbiCom e ViVoDa, uniram suas forças para demonstram a transmissão de vídeo entre uma câmera e um notebook, ambos equipados com CODECS MPEG-4 e chips Bluetooth (trabalho disponível em http://www.v2dc.com/news.html). Assim como outras pesquisas disponíveis sobre o assunto, esses trabalhos, apesar de seus valores, não falam nada sobre medida de qualidade, nem até que condições (número de clientes, taxa de erro do canal, tipo do vídeo etc.) essas

transmissões são possíveis. Falta que suprimos nesse trabalho. Para tais transmissões, alguns protocolos existentes precisam ser utilizados.

Nem toda aplicação usa toda a pilha de protocolos. Há vários modelos de uso especificados pela tecnologia Bluetooth. Cada modelo determina quais e como os protocolos devem ser utilizados. Para uma transmissão de vídeo para sistemas de vigilância, apenas alguns protocolos são necessários. Como, por exemplo, para sistemas de vigilância o áudio geralmente não é utilizado, o protocolo relacionado ao áudio foi excluído desse nosso modelo de uso.

A Figura 4-8 apresenta os protocolos necessários para dar sustentação aos vídeos. Os protocolos *Bluetooth Radio*, *Baseband* e LMP (*Link Manager Protocol*) estão incluídos por serem essenciais na manutenção dos enlaces Bluetooth. O *Bluetooth Radio,* juntamente com o *Baseband,* formam o equivalente à camada física do modelo OSI. O protocolo LMP é responsável pela configuração e controle dos enlaces.

O protocolo RFCOMM (*Radio Frequency Communication*) é necessário para manter uma comunicação serial entre os dispositivos, já o protocolo L2CAP (*Logical Link Control and Adaptation*) é necessário para dar suporte à multiplexação de dados, possibilitando a um mesmo dispositivo enviar ou receber mais de um vídeo ao mesmo tempo. O protocolo SDP (*Service Discovery Protocol*) é utilizado com o intuito dos dispositivos poderem localizar quais os serviços estão disponíveis na rede a cada instante (Bray, et. al., 2000, p.7).

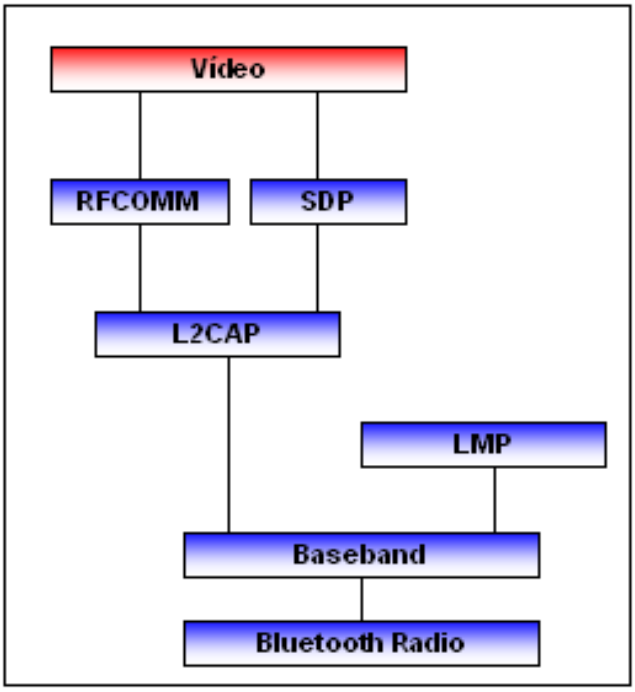

Figura 4-8: Protocolos necessários para uma transmissão de vídeo em Bluetooth.

Nossa aplicação, a transmissão de vídeo, fica no topo da pilha de protocolos, utilizando diretamente os protocolos RFCOMM e SDP. O protocolo SDP é o responsável por avisar aos outros dispositivos (servidores e clientes de vídeo) que tal dispositivo dispõe de alguns serviços (transmitir, receber vídeos, ou outros serviços) e também de enviar para o protocolo de aplicação a lista de todos os serviços que estão disponíveis na *piconet*, e quais dispositivos estão oferecendo cada serviço.

4.6. Codificação de Vídeo Utilizando MPEG-4

Para um sistema de vigilância, os vídeos devem ser exibidos em tempo real, isso significa que o espectador deve assistir o vídeo no momento em que está acontecendo ou, pelo menos, algo bem próximo a isso. A exibição dos quadros deve ocorrer na mesma taxa que a captura para não provocar aceleração ou desaceleração nos movimentos da cena apresentada. Para tal, são necessários alguns ajustes a serem feitos sobre os dados antes de serem transmitidos.

Um vídeo puro, capturado por câmeras e sem tratamento algum, necessita de uma banda de cerca de 100Mbps para ser transmitido (Bovik, 2000). Essa banda não é alcançada pela maioria das infraestruturas de redes existentes, mesmo as redes que alcançam não devem ser dedicadas apenas a vídeos. São por esses motivos que a codificação de vídeo é usada largamente. A codificação acrescenta robustez contra erros, infra-estrutura de sincronização e

compressão do vídeo. O padrão de codificação indicado para nosso sistema é o MPEG-4, desenvolvido tanto para transmitir áudio e vídeo sobre redes com altas taxas de bits, como cinema e TV digital, quanto sobre redes com menores taxas de transmissão, como as redes sem fio e a Internet.

Como visto no Capítulo 3, o MPEG-4 é capaz de inserir um alto grau de compressão sobre os vídeos, possibilitando transmissões de vídeos de uma fonte natural sobre redes de até 64Kbps de banda, uma banda relativamente pequena. Como as redes Bluetooth atingem no máximo 723,32Kbps numa transmissão ponto-a-ponto unidirecional, esse padrão de codificação é adequado para transmitir vídeos através de redes Bluetooth.

Há outros padrões de codificação com taxas de compressão tão boas quanto as do MPEG-4, todavia há outros fatores que justificam a escolha desse padrão, tais como:

- Maior facilidade nossa em encontrar pessoas no mundo trabalhando com esse padrão;
- Código aberto e gratuito (http://mpeg.telecomitalialab.com);
- Excelente documentação;
- Maior número de programas implementados de fácil acesso (codificadores, decodificadores, conversores e visualizadores).

4.7. Arquitetura do Sistema

Os servidores e clientes de vídeo, após a inserção de novos elementos (*chip* Bluetooth, codificadores e decodificadores MPEG-4), adicionam novas funcionalidades ao sistema.

Na Figura 4-9 podemos observar como é a arquitetura dos servidores de vídeo. Um servidor de vídeo Bluetooth completo é formado pelo codificador MPEG-4 e pelo *microchip* Bluetooth. Ele recebe vídeos das câmeras, constantemente capturando imagens, ou de outros dispositivos, os quais estejam repassando vídeos capturados. Os vídeos, ao serem recebidos pelo servidor, são repassados para o codificador, que codifica-os. Depois disso, o codificador envia o fluxo de bits já codificado até o *microchip* Bluetooth, este se encarrega de processá-lo para depois transmiti-lo aos clientes na forma de pacotes Bluetooth.

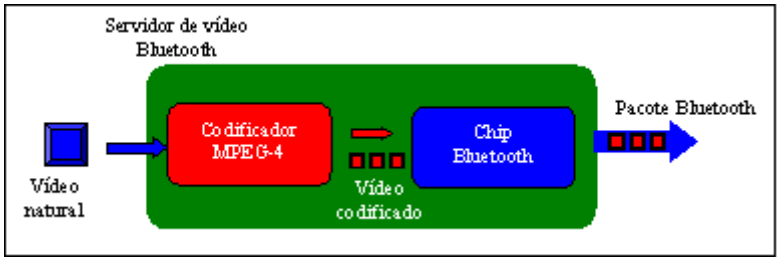

Figura 4-9: Arquitetura do servidor de vídeo Bluetooth.

Já o cliente executa o processo inverso do servidor (ver Figura 4-10), ele recebe os vídeos através do *microchip* Bluetooth, o qual possui o transceptor, desempacota os dados e monta o fluxo de bits MPEG-4 que será repassado para decodificador MPEG-4. Esse decodificador transformará o fluxo de bits MPEG-4 no mais próximo possível ao vídeo natural originalmente transmitido. Como a codificação para MPEG-4 é com perdas, o vídeo decodificado pelo cliente nunca é idêntico ao vídeo transmitido. O vídeo depois de decodificado é exibido em monitores. Se o objetivo do cliente for apenas armazená-los, não é necessária, então, a decodificação dos vídeos, já que o armazenamento deles em formato MPEG-4 é bem mais leve do que o armazenamento deles puros.

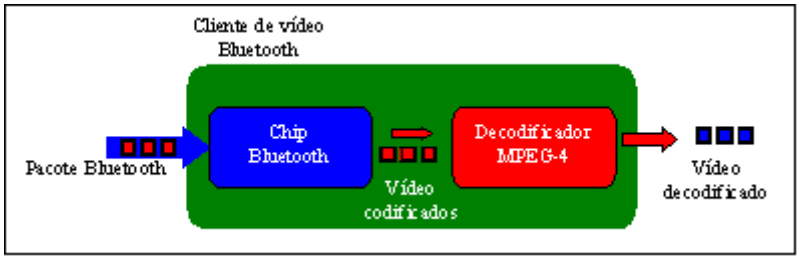

Figura 4-10: Arquitetura do cliente de vídeo Bluetooth.

Na Figura 4-11 podemos observar como os clientes e os servidores de vídeo interagem entre si. Vários servidores podem transmitir vídeos para vários clientes, desde que cada servidor transmita para no máximo sete clientes (restrição da tecnologia Bluetooth). Os servidores transmitem os vídeos MPEG-4 através de pacotes Bluetooth atravessando o canal sem fio e chegando aos clientes que, depois de os decodificarem, podem armazená-los, exibi-los ou repassá-los para outros dispositivos. Da mesma forma, um cliente pode receber vídeos de mais de um servidor ao mesmo tempo, isso graças à implementação do protocolo L2CAP.

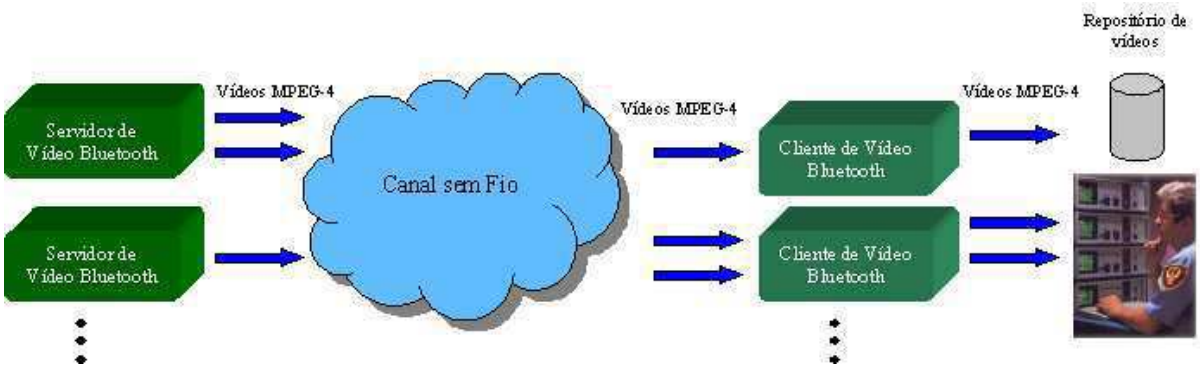

Figura 4-11: Arquitetura do sistema.

O canal sem fio é um meio de transmissão naturalmente errôneo, isso significa que, mesmo havendo banda suficiente, os dados sempre sofrem algum tipo de erro (atenuação, ruído, interferência, etc). Para diminuir os efeitos desses erros inseridos sobre os dados, o sistema deve prover mecanismos próprios de proteção (CRC, ACK, FEC, etc.). A tecnologia Bluetooth protege os dados implementando, se desejado, a retransmissão de pacotes (ACK) e os dados úteis dos pacotes são protegidos com o esquema FEC (*Forward Error Correction*), tornando esses dados mais resistentes aos erros. Além disso, a codificação MPEG-4 também protege os bits dos vídeos codificados. Os decodificadores MPEG-4 são capazes de recuperar grande parte dos vídeos, mesmo quando uma grande porcentagem dos dados é perdida.

4.8. Sumário

Um sistema de transmissão de vídeo, para ser utilizado em vigilância, deve possuir características especiais. Em nossa proposta, adotamos o uso de duas tecnologias (Bluetooth e MPEG-4) adicionando ao sistema mais robustez, maior qualidade dos vídeos, maior proteção dos dados, mais dinamismo, praticidade e agilidade. A forma como essas tecnologias devem ser utilizadas pelo sistema foi apresentada neste capítulo.

Capítulo 5

5. Implementação e Simulação

Para implementar o sistema que projetamos seriam necessários vários equipamentos Bluetooth. Não bastaria apenas uma transmissão entre dois dispositivos, mas sim testes com a carga máxima da tecnologia (oito dispositivos por *piconet*). Seriam necessários, no mínimo 8 kits de desenvolvimento Bluetooth.

A compra desses equipamentos tornou-se algo caro e inviável para nós, então a solução encontrada foi a simulação digital, que é a melhor opção para o teste de sistemas antes de sua realização, economizando tempo e dinheiro.

As simulações devem ser implementas com muito cuidado, elas devem representar todos os aspetos relevantes da situação real que desejamos simular.

Para as simulações optamos pelo ambiente Ptolemy, que possui ferramentas gráficas, é facilmente estendível e adaptável, além de possuir uma numerosa comunidade de usuários pelo mundo. O projeto Ptolemy é apresentado com mais detalhes na Seção 5.2.

Os elementos do sistema que precisaram ser implementados são descritos na Seção 5.3. Já os detalhes de como esses elementos foram implementados estão na Seção 5.4, a Seção 5.5 apresenta relevâncias e decisões que foram tomadas no momento das simulações, as conclusões são apresentadas na Seção 5.6.

5.1. O Ambiente Ptolemy

Lee (2001) define o ambiente da seguinte forma: “O ambiente Ptolemy estuda modelagem, simulação e projeto de sistemas concorrentes heterogêneos”. O autor completa dizendo que modelagem é o ato de representar formalmente um sistema, ou subsistema, e projetar é o ato de definir um sistema, ou um subsistema. Diz também que simular pode ser definido como o ato de executar modelos.

O ambiente visa principalmente os sistemas embutidos, particularmente aqueles que mesclam tecnologias, como dispositivos eletrônicos analógicos e digitais, dispositivos eletromecânicos e *hardwares* dedicados. Outra aplicação importante é em sistemas muito complexos que necessitam de uma simulação prévia antes de sua concretização.

Em sua primeira versão o Ptolemy é implementado todo em linguagem C. Visando uma maior utilização do ambiente pelo mundo, e usufruir os benefícios da orientação a objetos, a segunda versão do projeto (Ptolemy II) foi escrita em linguagem Java e possui os seguintes pontos fortes:

- Provê um rico conjunto de mecanismos de interação que estão embutidos nos domínios do Ptolemy II. Os domínios forçam os desenvolvedores de componentes a pensarem no padrão como os outros componentes do domínio estão interagindo;
- Seus componentes são polimorfos no campo do domínio, isso significa que eles podem interagir com outros componentes mesmo esses sendo de domínios diferentes;
- Possui um *framework* formal e abstrato que descreve modelos de computação possibilitando maior facilidade de extensão desses modelos;
- Possui ferramentas para a simulação dos modelos, permitindo a execução dos mesmos como aplicações locais ou *web*, podendo ser utilizadas para apresentações à distância. Possui uma linguagem própria de marcação (*Mark-up Language*), baseada em XML (MoML), além de possuir uma ferramenta visual (*Vergil*), que facilita a realização e o entendimento de modelos mais complexos;
- Seu código é aberto e gratuito, além de possuir milhares de usuários em todo o mundo.

Dos vários modelos de computação disponíveis, também denominados domínios, o que utilizamos em nossas simulações é o *Discrete-Events* (DE). Nesse domínio os componentes comunicam-se através de eventos dispostos numa linha de tempo simulado. Um evento corresponde a um valor e uma marca de tempo. Os componentes tanto podem disparar eventos, como podem reagir de forma diferente para cada tipo de evento ocorrido. Esse

domínio é ideal para simulação de sistemas de telecomunicações e para especificação de hardware.

5.2. Elementos do Sistema

Para que nossas simulações correspondam em todos os aspetos relevantes à situação real de uma transmissão de vídeo, resolvemos implementar os elementos mais importantes do sistema. São eles, servidor de vídeo, cliente de vídeo, fonte de vídeo e canal sem fio. Todos esses elementos estão presentes em sistemas reais e são indispensáveis. Como visto no capítulo anterior, os elementos que possuem obrigatoriamente *chips* Bluetooth são os clientes e os servidores de vídeo.

5.2.1. FONTE DE VÍDEO

Em nossas simulações, a fonte envia vídeos armazenados em arquivos simulando uma fonte natural transmitindo em tempo real. Os arquivos são codificados antes das simulações e armazenados em formato MPEG-4. Cada fonte tem sua taxa de quadros configurável e sempre transmite os vídeos respeitando essa taxa. Se o receptor da fonte não conseguir processar os quadros na taxa que eles estão sendo enviados, quadros são perdidos, mas a fonte não muda o seu ritmo de transmissão. As fontes de vídeo não são dispositivos Bluetooth.

Implementamos a fonte de vídeo como um elemento separado para aumentar a flexibilidade das simulações. Podemos admitir em uma situação real que uma câmera é um servidor de vídeo que possui uma fonte de vídeo embutida nele, assim como, no caso de querermos simular um servidor enviando mais de um vídeo, podemos dizer que esse servidor possui várias fontes de vídeo.

5.2.2. SERVIDOR DE VÍDEO

O servidor é aquele que recebe vídeos das fontes e repassa-os encapsulados em pacotes Bluetooth para os clientes através de um transceptor Bluetooth (*microchip*). O servidor recebe os vídeos das fontes nas taxas de quadros definidas por elas, por isso ele deve repassá-los aos clientes nessas mesmas taxas, se os clientes não puderem recebê-los, eles serão perdidos e a qualidade dos vídeos será prejudicada. Como vídeos codificados possuem tamanhos de quadros variados, a taxa de transmissão necessária para transmiti-los também é variada. Isso significa que, para alguns quadros de um mesmo vídeo, uma determinada taxa de

bits é suficiente, enquanto que para outros quadros, ela pode não ser. Logo, o sistema para transmitir um vídeo terá seu desempenho bastante variado.

Numa transmissão de vídeos comprimidos há momentos em que o sistema consegue transmitir facilmente alguns quadros do vídeo, enquanto que em outros, certos quadros não podem ser transmitidos por completo.

Um servidor, como é um dispositivo Bluetooth, pode comunicar-se até com sete clientes simultaneamente. Já o número de fontes de vídeo é ilimitado, sendo que não faz sentido um servidor se conectar a mais de sete fontes de vídeos se ele possui no máximo sete clientes.

5.2.3. CLIENTE DE VÍDEO

Os clientes de vídeo possuem transceptores Bluetooth e através deles recebem os vídeos dos servidores em pacotes. Os clientes representam qualquer dispositivo que recebe vídeos de um ou mais servidores. Num sistema real são clientes de vídeo, os dispositivos que recebem vídeos e apenas armazenam em grandes repositórios, os monitores, que recebem os vídeos e apenas os exibem, e os dispositivos de distribuição, que recebem os vídeos de servidores diferentes e os repassam a outros clientes.

O cliente sabe quais são os servidores autorizados a enviar vídeos a ele, então ele recebe cada pacote, verifica no cabeçalho se o pacote deve ser lido ou descartado. Todos os pacotes recebidos de um mesmo servidor formam um vídeo. Esse vídeo é armazenado em arquivo, e deve representar o vídeo transmitido em tempo real. Para amenizar o efeito de congestionamento, os clientes necessitam de uma memória (*buffer*) para armazenar os dados recebidos. Se um pacote de vídeo chega atrasado ao cliente, ele pode manter uma exibição contínua do vídeo usando os quadros armazenados. O cliente está sempre lendo os dados desse *buffer*, se ele está vazio é porque seu servidor, por alguma razão, não pôde lhe enviar dados suficientes, se o *buffer* estiver cheio significa que o cliente está sobrecarregado e os próximos dados recebidos serão desperdiçados. Tratando-se de uma aplicação real, não faz sentido possuir um *buffer* de tamanho muito grande, porque isso ocasionaria um retardo no vídeo exibido em relação à cena observada pela câmera, se o *buffer* for pequeno demais muitos dados são perdidos e todo o sistema pode ser prejudicado.

A escolha do tamanho do *buffer* é uma decisão importante a ser tomada, e varia de situação para situação.

Para exibir os vídeos recebidos, o cliente deve possuir um decodificador MPEG-4, mas como nem todo cliente exibe os vídeos recebidos, esse não é um requisito essencial dos clientes de vídeo.

5.2.4. CANAL SEM FIO

Dependendo do ambiente em que as transmissões de vídeo são feitas, o desempenho do sistema pode variar drasticamente. Como estamos falando de uma transmissão sem fio, um ambiente desfavorável pode causar muitas interferências e causar um grande número de perda de pacotes, afetando todo o sistema.

Como o sistema será implementado através de simulação, o canal sem fio deve ser considerado, sendo implementado da forma que mais se assemelhe aos canais sem fio reais. Existem vários trabalhos sobre a criação de um modelo matemático que melhor represente o efeito dos canais sem fio sobre as transmissões de dados. Para cada tipo de transmissão pode haver vários modelos matemáticos que representam bem a situação real.

Shen, et. al. (1995) faz um estudo sobre a utilização do modelo de Markov para simular canais de transmissão via rádio, mostrando que esses canais, denominados Canais de Markov de Estados Finitos, podem muito bem ser modelados através de cadeias de Markov de estados finitos. Já Nguyen, et. al. (1996) descreve um esforço para caracterizar o comportamento de perdas de pacotes de duas redes sem fio populares nos Estados Unidos. Ele também utiliza uma cadeia de Markov de dois estados, além de utilizar um outro modelo próprio.

Um modelo de canal para transmissões sem fio de pacotes, que tanto pode ser usado em análises teóricas, como em simulações para testes de performance de protocolos de redes, é apresentado por Liebl, et. al. (2000). A correlação entre as sucessivas perdas de pacotes é descrita através de um modelo de Markov. O objetivo desse modelo é poder ser derivado para ser utilizado na simulação de praticamente qualquer protocolo de transmissão de pacotes através de meios sem fio.

Ghose, et. al. (2001) fez simulações de dispositivos Bluetooth modelando enlaces como uma cadeia discreta de Markov de dois estados. Esse também é o modelo utilizado por nós. A Figura 5-12 apresenta esse modelo. O canal sem fio oscila entre os dois estados, BOM e RUIM.

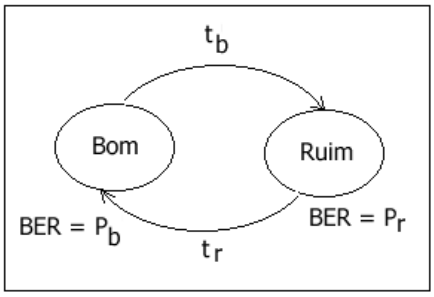

Figura 5-12: Estrutura de transição do modelo de canal de Markov.

No estado BOM a taxa de erro de bit (*Bit Error Rate* - BER) é denominada por P_b, e no estado RUIM a BER é denominada por P_r, onde $P_r >> P_b$. P_b e P_r denotam a probabilidade de ocorrer um erro num bit quando o canal estiver no determinado estado. O tempo médio que o canal passa no estado BOM é dado por t_b, e no estado RUIM é dado por t_r.

Para obter os parâmetros médios dos ambientes de transmissão via rádio, vários canais foram simulados por Ghose (2001), utilizando o mesmo tipo de transmissão dos transceptores Bluetooth. Os parâmetros obtidos foram $P_b = 6{,}879 \times 10^{-5}$, $P_r = 1{,}263 \times 10^{-3}$, $t_b = 437{,}5$ms e $t_r = 55{,}8$ms.

5.3. A Implementação

Para simularmos os elementos do sistema utilizando o Ptolemy II estendemos algumas classes do *framework* Ptolemy, criando nossos próprios componentes, para que eles possam interagir com os outros componentes já implementados no *framework*.

Um componente de simulação é chamado pelo Ptolemy de *Actor* (Ator). Uma simulação é formada por um conjunto de atores que trocam mensagens seguindo as regras de um domínio, gerenciado por um *Director* (Diretor). Os atores se comunicam através da troca de mensagens. Uma mensagem é chamada de *Token*. Há vários tipos de *Tokens* para cada tipo de dado transmitido.

Os dispositivos Bluetooth devem ser simulados na forma de atores. Cada elemento do sistema é um ator, servidor, cliente, canal e fonte de vídeo. O *framework* já possui vários atores implementados separados por tipos. Na Figura 5-13 são mostrados todos os tipos de atores implementados pelo *framework*. Para cada tipo de ator há vários atores já implementados que podem ser utilizados para auxiliar outros projetos. A classe base é *TypedAtomicActor* que estende a classe *Actor*. Os atores *Sink* são aqueles que recebem *Tokens* processam e não repassam para outros atores, já os atores *Source* são aqueles que geram

Tokens sem receber de nenhum outro ator, já os atores *Transformer* são aqueles que recebem *Tokens*, processam e transmitem para outros atores. Desses três tipos de atores vários outros são derivados.

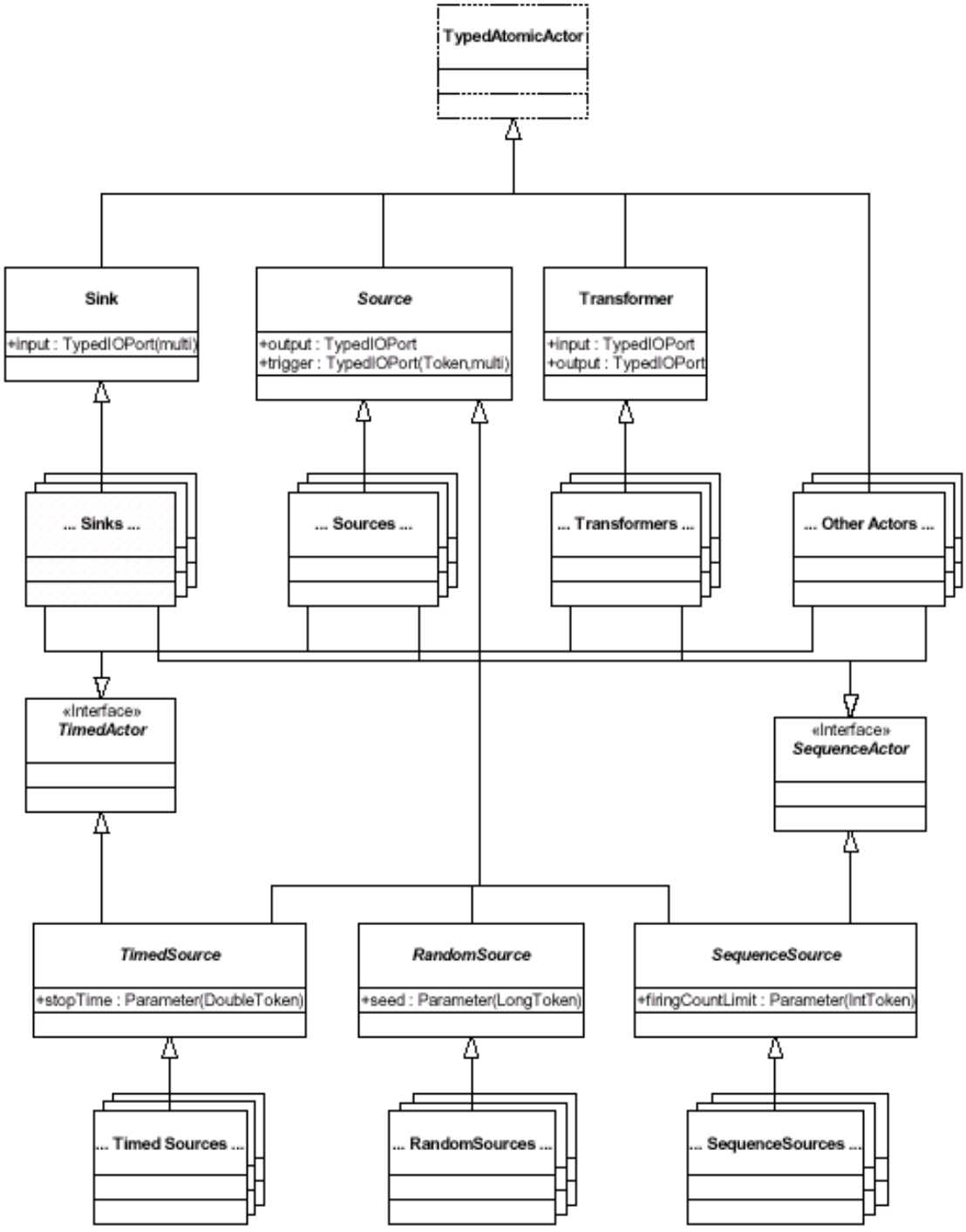

Figura 5-13: Diagrama de classes dos atores Ptolemy (Fonte: DEECS, 2002, p. 4-3).

O projeto *Ptolemy* também inclui uma ferramenta gráfica (*Vergil*) utilizada para executar as simulações com melhor usabilidade. Na Figura 5-14 é apresentada uma tela da ferramenta *Vergil*. À esquerda localizam-se os atores, diretores e utilitários disponíveis para

serem utilizados em qualquer modelo. Todos esses elementos apresentados nessa figura são elementos desenvolvidos pelo projeto *Ptolemy*. A área maior é para a realização dos modelos. Nesse exemplo podemos observar um diretor (*SDF Director*) e três atores fazendo parte desse modelo. As simulações podem ser salvas em arquivos MoML (XML), que podem ser criados ou editados mesmo sem o auxílio da ferramenta.

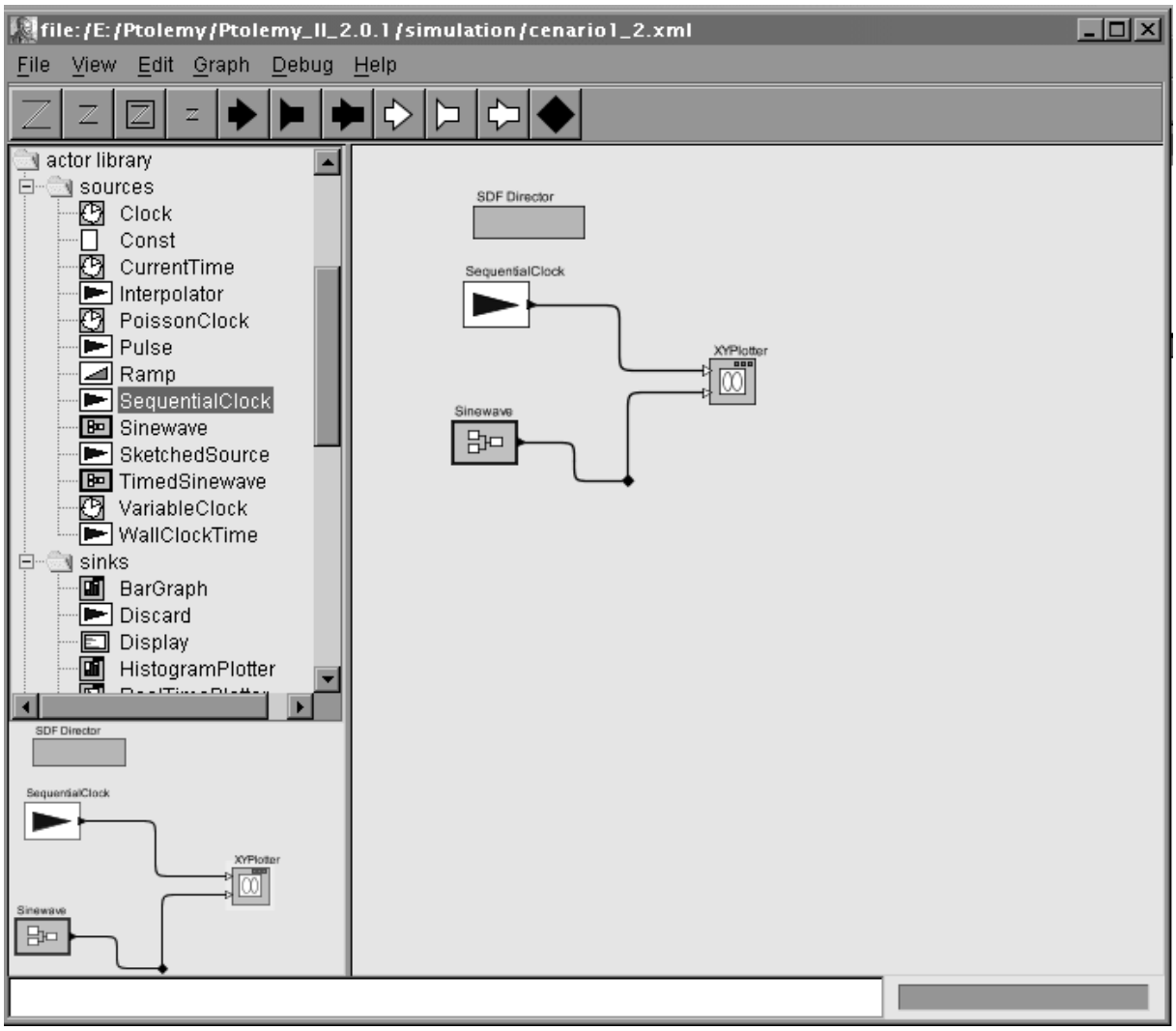

Figura 5-14: Tela da ferramenta Vergil.

Nós implementamos quatro atores que são derivações da classe *TypedAtomicActor* (ver Figura 5-15). Os atores implementados por nós foram: *BluetoothVideoServer*, *BluetoothVideoClient*, *VideoSource* e *Channel*. Com esses quatro atores podemos simular qualquer configuração para nosso sistema de transmissão de vídeo com Bluetooth. Desses quatro atores, os únicos que representam dispositivos Bluetooth são *BluetoothVideoServer* e *BluetoothVideoClient*. Independente dos dispositivos serem *slave*, ou *master,* eles serão ou um cliente, ou um servidor de vídeo. Como os vários algoritmos utilizados na formação das *piconets* (que também determinam se os dispositivos serão *master* ou *slave*) não fazem parte

do escopo de nosso trabalho, determinamos que os servidores de vídeo são sempre *masters* e os clientes são sempre *slaves*. O fato de qual dos elementos será *master* não é determinante no desempenho do sistema como um todo.

O ator *Channel* recebe todos os *Tokens* (pacotes) trocados entre os servidores e os clientes e simula os efeitos dos canais sem fio sobre os *Tokens* transmitidos. Uma vez implementados, os atores podem ser dispostos segundo várias configurações. Cada ator possui atributos que são alterados antes da simulação. Esses atributos modificam características dos atores que alteram o cenário da simulação. Eles são do tipo *Parameter* e sempre quando modificados, são testados pelo método *attributeChanged* definido pelo usuário. Esse método serve como prevenção para que as simulações não sejam prejudicadas por dados inconsistentes. O usuário é quem faz, através desse método, uma avaliação prévia de cada atributo modificado durante a simulação, se algum atributo for entrado com um valor não válido, o método *attributeChanged* lança uma exceção que aparecerá na ferramenta visual como uma mensagem de erro, e o usuário poderá corrigi-lo.

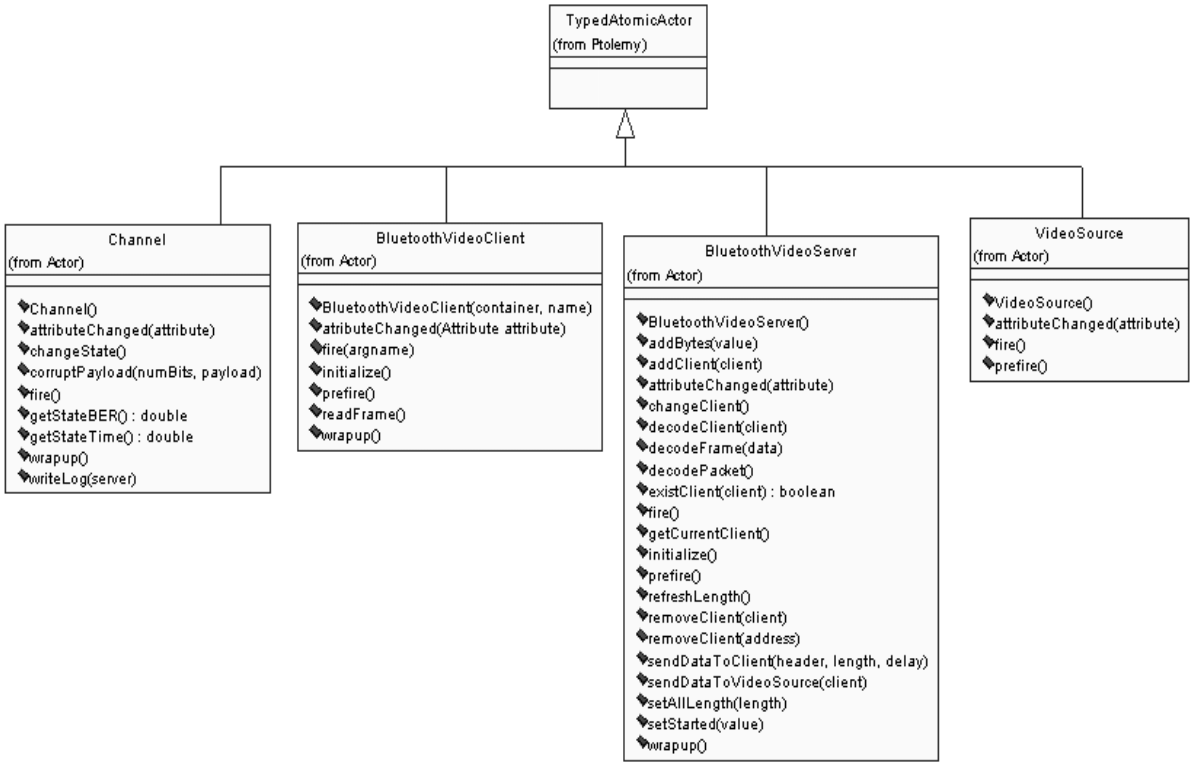

Figura 5-15: Diagrama de classes com nossos quatro atores.[5]

[5] As classes que possuem a identificação (*from Ptolemy*) não foram implementadas por nós.

Todos os atores possuem um método chamado *fire*, que contém as ações que os atores devem executar. Em cada iteração da simulação, cada ator executa seu método *fire* uma única vez. O método é chamado sempre que um *Token* é recebido, ou pode ser chamado pelo próprio ator mesmo sem ele ter recebido nenhum dado. O diretor do modelo é o responsável por gerenciar as ações de todos os atores.

Na Figura 5-16 é apresentada uma tela do *Vergil* quando uma simulação de transmissão de vídeo estava sendo preparada. Podemos observar à esquerda da figura que a lista de atores disponíveis foi acrescida de quatro novos atores agrupados num grupo chamado *bluetooth*.

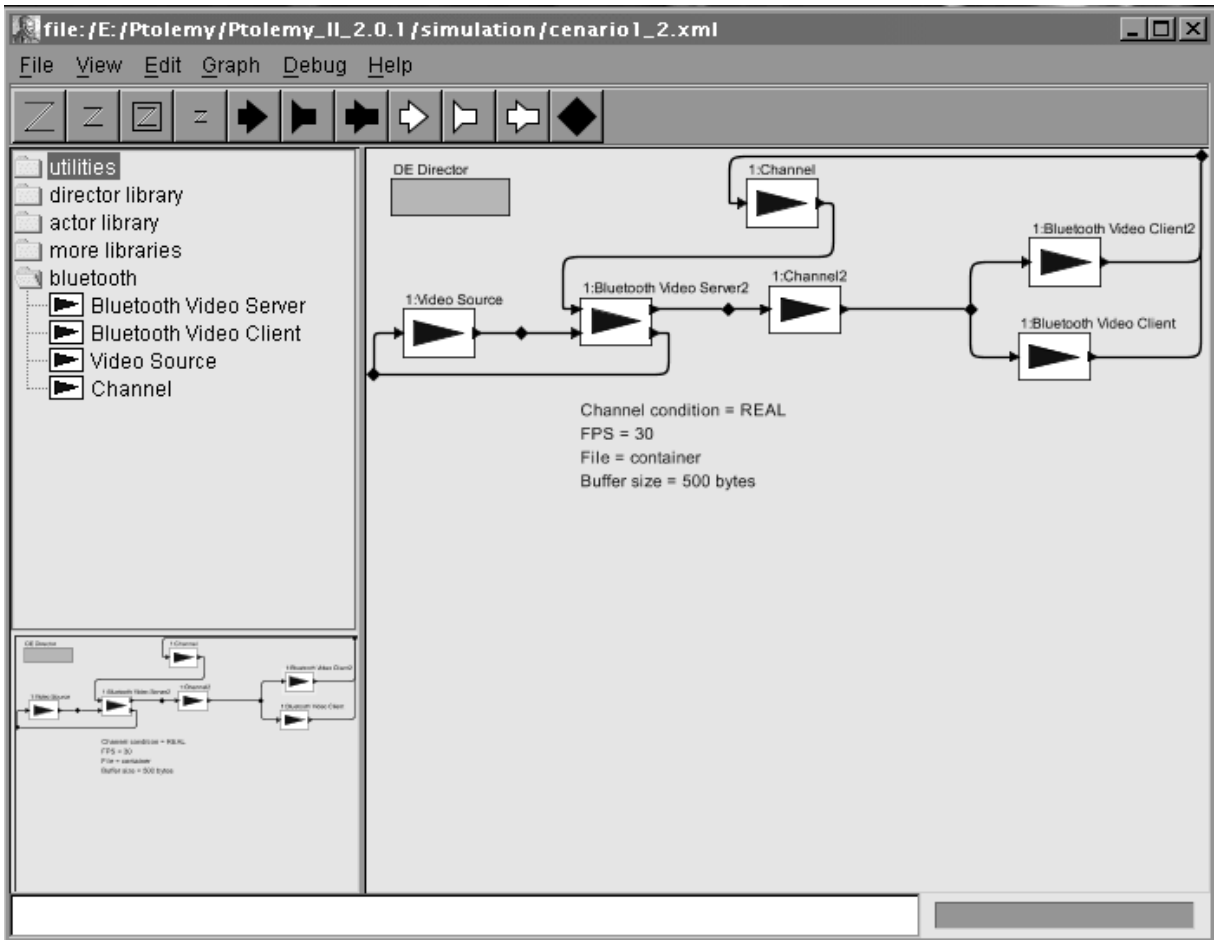

Figura 5-16: Tela do Ptolemy com atores Bluetooth sendo utilizados.

Nessa simulação há uma fonte de vídeo, um servidor, dois canais e dois clientes. Os pacotes que percorrem o sentido do servidor para os clientes passam por um canal enquanto os pacotes que fazem o sentido contrário passam pelo outro canal. A utilização de dois canais é justificada apenas pela clareza da leitura. Os efeitos de utilizar um canal ou dois são os mesmos.

Os dados trocados entre os dispositivos são pacotes Bluetooth, como o *framework* não possui esse tipo de pacote criamos nosso próprio *Token* que é um pacote Bluetooth. Chamamos esse *Token* de *BTPacketToken*, o qual estende a classe *Token* e implementa a interface *BTPacketTokenIF* (ver Figura 5-17). Ele dita o comportamento de todos os pacotes Bluetooth. *BTPacketToken* é um pacote com vários campos. Cada campo é do tipo *Token*. Todos os campos são encapsulados em um *RecordToken* que forma o pacote em si.

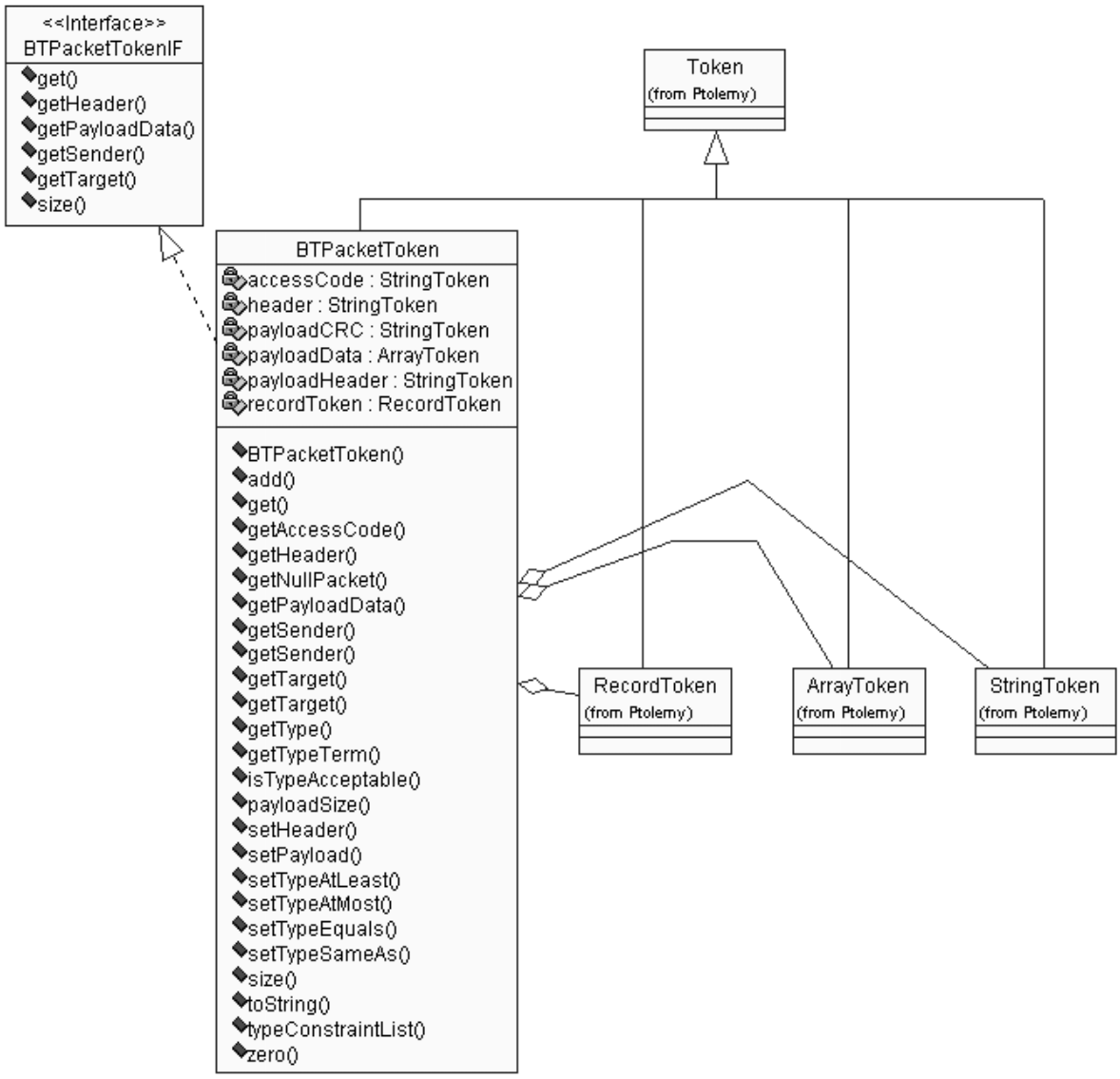

Figura 5-17: Diagrama de classes com o pacote Bluetooth.

5.3.1. BLUETOOTHVIDEOCLIENT

O *BluetoothVideoClient* estende *TypedAtomicActor* (ver Figura 5-18), possui duas portas, uma de entrada e outra de saída, através das quais ele trocará *BTPacketTokens* com o *BluetoothVideoServer*. Essas portas são do tipo *DEIOPort*, que são portas próprias para serem

utilizadas no domínio DE (*Discret Event*). Ele também possui quatro atributos que podem ser configurados para modificar as características do ator antes da simulação.

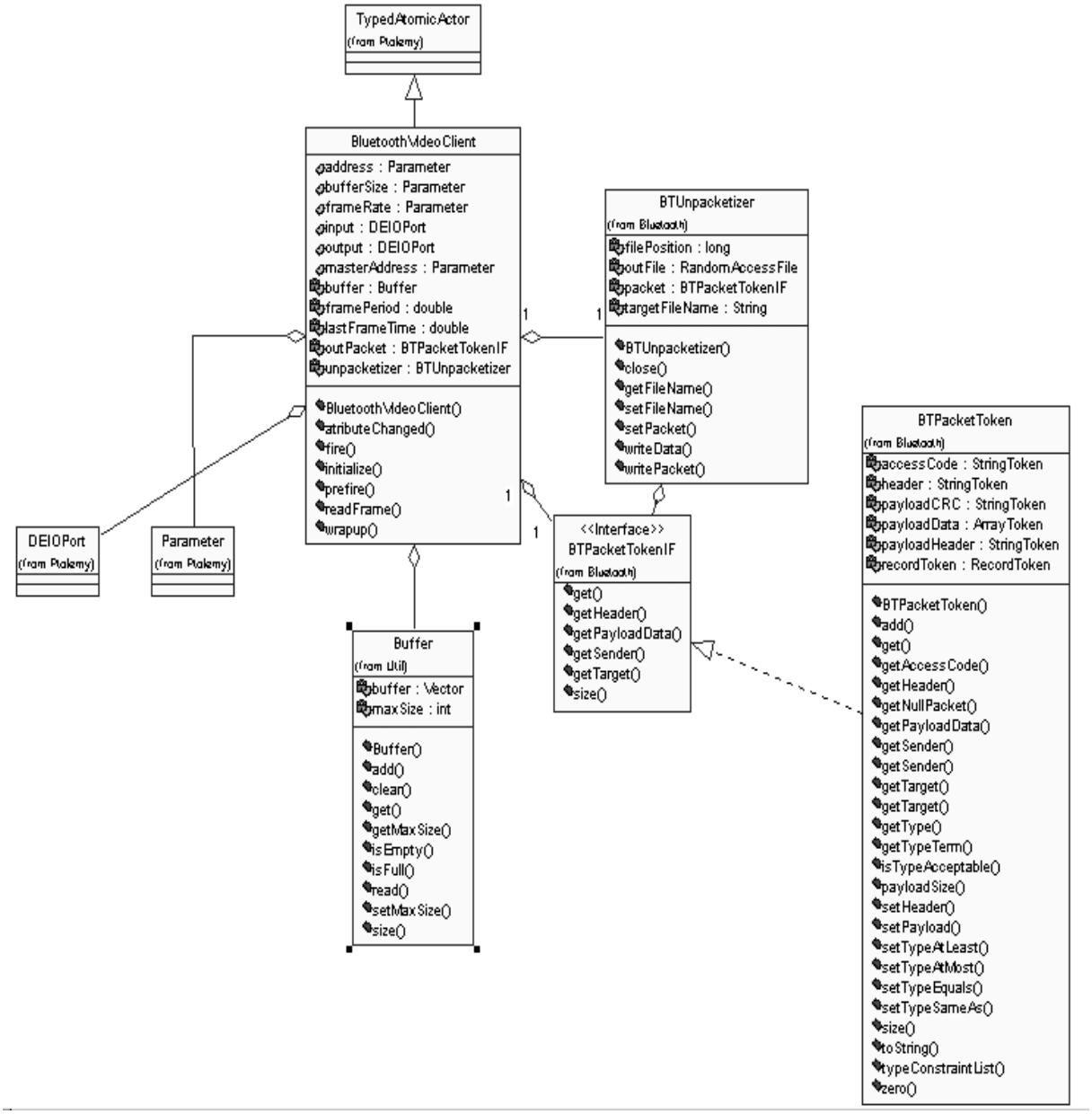

Figura 5-18: Diagrama de classes que detalha os clientes de vídeo Bluetooth.

O *BluetoothVideoClient*, além de outros atributos, possui um objeto da classe *Buffer,* um objeto da classe *BTUnpacketizer* e um objeto da classe *BTPacketTokenIF*. O *buffer* é utilizado para armazenar os dados dos vídeos recebidos. Essa classe armazena bytes numa fila e possui métodos para gerenciar esses dados.

A classe *BTUnpacketizer* tem como função extrair todos os dados úteis dos pacotes que são recebidos pelo cliente e armazená-los num arquivo que será o vídeo reconstituído pelo cliente.

Os clientes possuem parâmetros que podem ser alterados em cada ator antes de sua execução. Esses parâmetros nos clientes são o endereço do cliente, o tamanho de seu *buffer*, a taxa de quadros em que o vídeo vai ser recebido e o endereço de seu *master* (um servidor de vídeo). Na Figura 5-19 é apresentada como é feita a configuração desses parâmetros. Para cada ator, esses parâmetros devem ser configurados. Qualquer erro em um dos parâmetros, como o endereço do *master* errado, por exemplo, prejudica o funcionamento da simulação, porque os clientes vão rejeitar todos pacotes que não sejam se seu *master*.

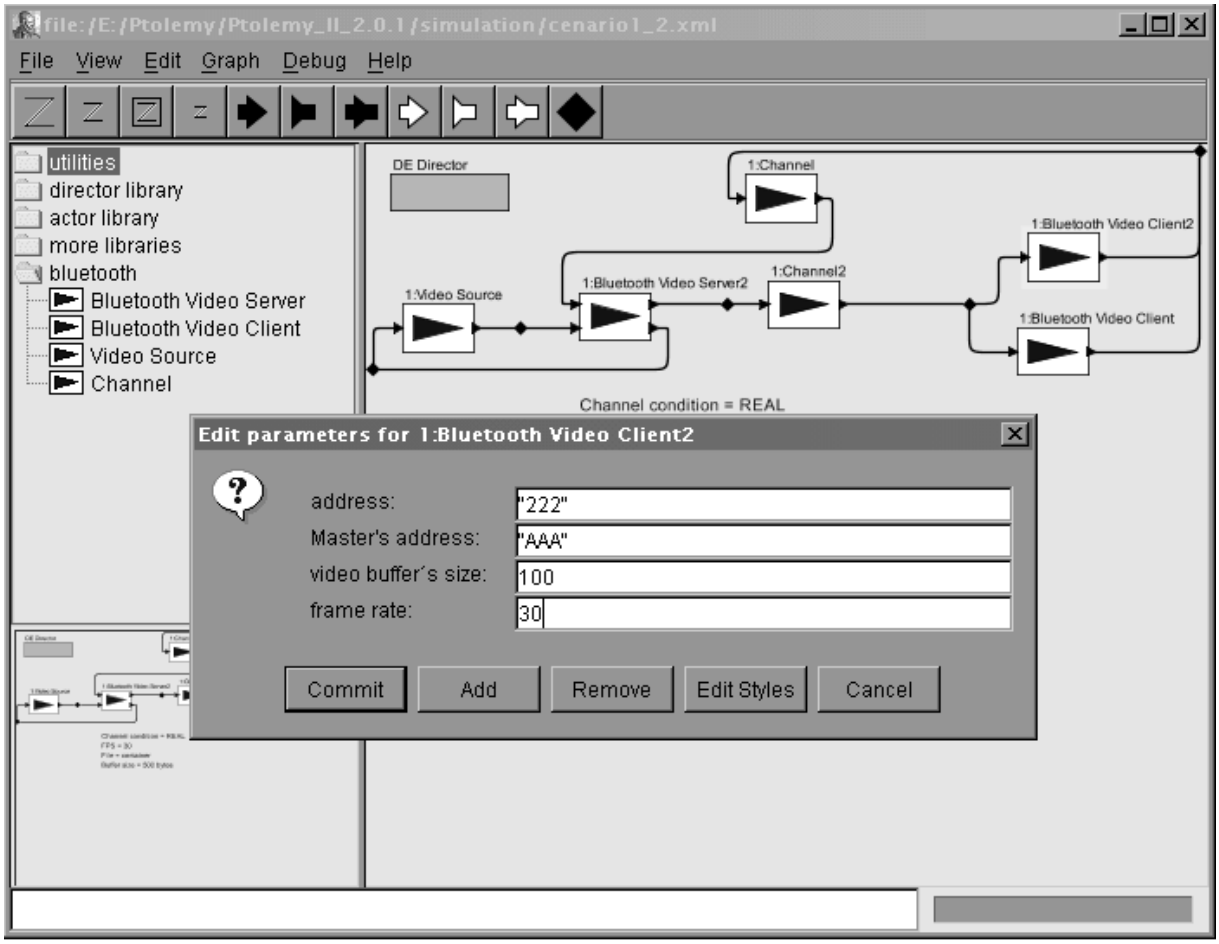

Figura 5-19: Edição de parâmetros de um ator *BluetoothVideoClient*.

5.3.2. BLUETOOTHVIDEOSERVER

Esta classe é a mais complexa do projeto. Ela, ao mesmo tempo em que gerencia várias transmissões para vários clientes, gerencia a recepção dos vídeos pelas fontes de vídeo. Ela possui quatro portas de entrada e saída, duas para comunicação com os clientes e duas

para comunicação com as fontes de vídeo. Os parâmetros configuráveis para o servidor são o seu endereço, os endereços de seus clientes e das fontes de vídeo de onde cada cliente irá receber os vídeos e o tipo de pacote que será utilizado na transmissão. Os pacotes DM1 e DH1 foram mostrados em nosso trabalho anterior (Brito, Melcher, 2002) que são insuficientes para transmissões de vídeos. Fizemos testes intensos das taxas de transmissão alcançadas pela rede utilizando os vários tipos de pacotes e os pacotes que ocupam apenas uma janela chegaram a menos do que a codificação MPEG-4 necessita (64 Kbps).

BluetoothVideoServer também estende a classe *TypedAtomicActor* (ver Figura 5-20) e possui, para cada cliente de vídeo associado, um objeto da classe *Buffer*. Esse *buffer* irá armazenar os dados enviados pelas fontes de vídeo. Muitas vezes as fontes enviam mais dados do que o servidor é capaz de transmitir, então esses dados em excesso são armazenados nesses *buffers*. Nesse caso, o *buffer* é de tamanho "infinito", ou seja ele é capaz de armazenar qualquer quantidade de dados que as fontes puderem produzir. Isso porque estamos adotando que os servidores são capazes de processar os volumes de dados produzidos pelas fontes, caso contrário, o projeto seria inviabilizado.

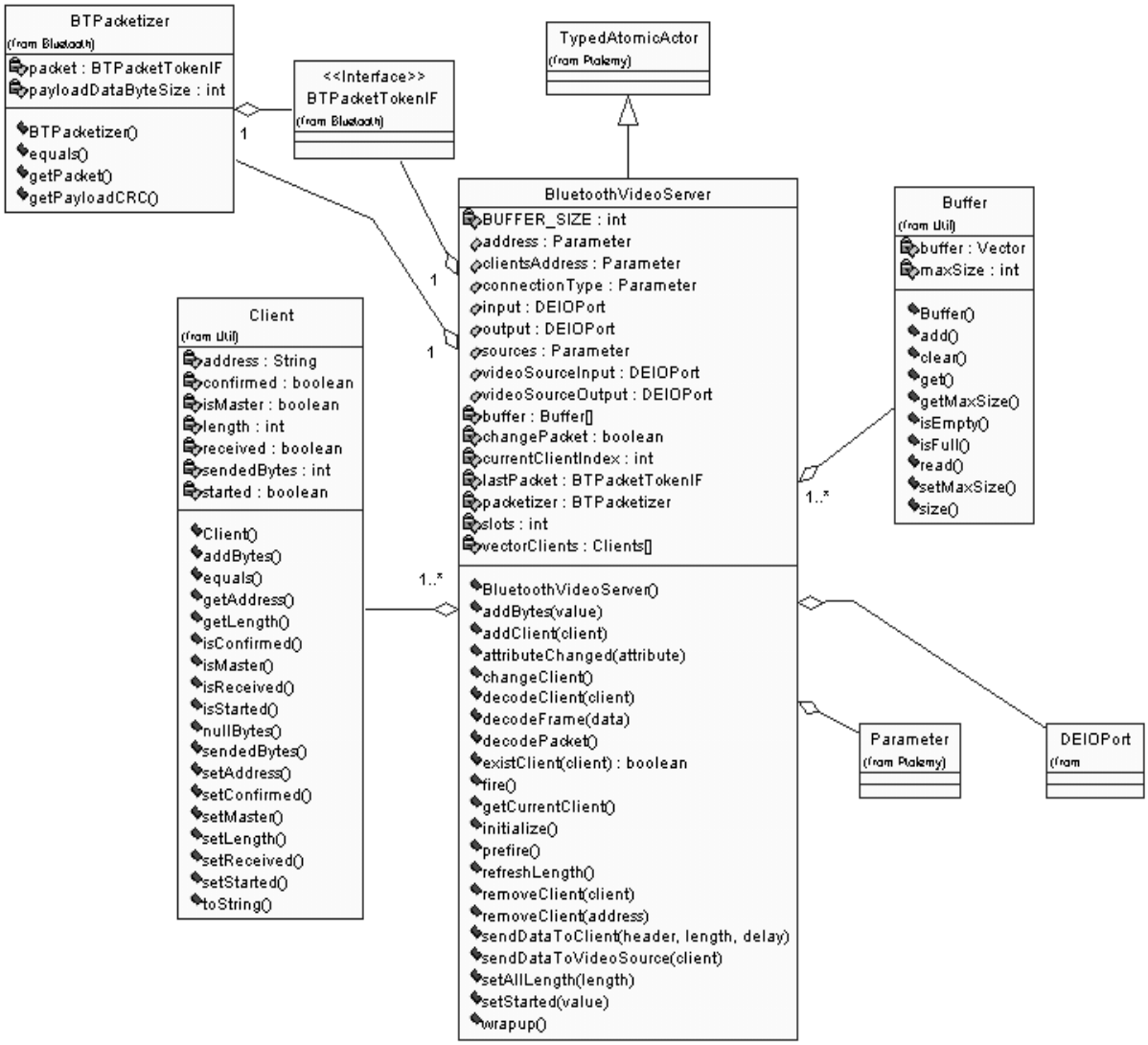

Figura 5-20: Diagrama de classes detalhando *BluetoothVideoServer*.

Os servidores também possuem referências a vários objetos do tipo *Client*, para que os servidores tenham um maior controle sobre suas comunicações com seus clientes. Através desses objetos o servidor sabe quantos dados enviou para cada cliente, quais clientes receberam dados, quem deve ser o próximo cliente a ser atendido, etc. Ele também possui referências a um objeto do tipo *BTPacketTokenIF* e outro do tipo *BTPacketizer*, que recebe os dados do servidor e monta os pacotes antes deles serem transmitidos. O objeto *BTPacketTokenIF* armazena o último pacote transmitido, no caso de necessitar retransmitir esse pacote.

5.3.3. CHANNEL

O objeto *Channel* é aquele que atua sobre todos os pacotes transmitidos inserindo sobre eles o efeito dos canais sem fio. Ele estende a classe *TypedAtomicActor* (ver Figura 5-

21), possui duas portas, uma de entrada e outra de saída, e possui apenas um parâmetro configurável que é a condição do canal. Classificamos cinco tipos de condições do canal, uma que simula a situação normal, onde o canal oscila entre um estado de baixa taxa de erros de bits (estado BOM sempre) e um estado de alta taxa de erros (estado RUIM sempre). Uma condição que simula uma situação péssima do canal com a maior BER, outra condição que simula uma condição ruim, uma condição que simula uma situação boa do canal e outra que simula uma situação ótima, onde a BER é a menor de todas as outras condições. Isso é necessário porque nem sempre o canal fica oscilando entre apenas dois estados. Há muitos casos em que o canal passa por péssimos e ótimos estados. Queremos, com a adição de mais quatro condições de canais, simular como o sistema se comporta em diversas condições de transmissão.

Na condição péssima o canal fica sempre com a mesma BER de P = 2,526 x 10^{-3}, a maior BER das condições que adotamos. Na condição ruim o canal fica sempre no estado RUIM sem oscilar, com BER de P = 1,263 x 10^{-3}. Na situação normal o canal oscila entre o estado BOM e o RUIM. Na condição boa o canal fica sempre no estado BOM com BER de P = 6,879 x 10^{-5}, já na condição ótima, o canal fica sempre no mesmo estado com a menor de todas as BER, P = 3,439 x 10^{-5}.

Para calcularmos quantos bits errados ocorrem num pacote, fizemos os seguintes cálculos. Queremos calcular a probabilidade de num pacote de *n* bits, *k* bits estarem corrompidos. A probabilidade de cada bit estar errado é a BER do canal correspondente. Podemos classificar esse experimento probabilístico como um Ensaio de Bernoulli (Hazzan, 1993) que pode ser representado pela Distribuição Binomial:

Equação 1:

$$\mathbf{P_k = (n!/k!(n-k)!) * p^k * q^{n-k}}$$

Onde,

k: número de bits corrompidos.

n: tamanho do pacote.

p: probabilidade de ocorrer erro em um bit (BER).

q: probabilidade de não ocorrer erro em um bit (1 - p).

Para os pacotes não protegidos (DH) podemos utilizar a Equação 1 diretamente. Já para os pacotes protegidos (DM), os dados úteis são protegidos de forma tal que a cada grupo de 15 bits, erros simples são corrigidos, e erros duplos podem ser detectados. Nesse caso, como não estamos considerando retransmissões, queremos calcular a probabilidade de, num pacote de *n* bits, ocorrerem mais de dois erros. Pela distribuição binomial temos que:

Equação 2:

$$P_0 = (n!/0!(n-0)!) * p^0 * q^{n-0} = q^n$$

Equação 3:

$$P_1 = (n!/1!(n-1)!) * p^1 * q^{n-1} = n!/(n-1)! * p * q^{n-1}$$

Equação 4:

$$P_2 = (n!/2!(n-2)!) * p^2 * q^{n-2} = (n!/2(n-2)!) * p^2 * q^{n-2}$$

Podemos dizer que a probabilidade de três ou mais bits estarem errados é:

Equação 5:

$$P_{3+} = 1 - (P_0 + P_1 + P_2)$$

Equação 6:

$$P_{3+} = 1 - (q^n + (n!/(n-1)! * p * q^{n-1} + (n!/2(n-2)!) * p^2 * q^{n-2}))$$

Como queremos calcular qual a probabilidade de ocorrer 3, ou mais erros em cada grupo de 15 bits, podemos substituir a variável n por 15:

$$P_{3+} = 1 - (q^{15} + (15!/(15-1)! * p * q^{15-1} + (15!/2(15-2)!) * p^2 * q^{15-2}))$$

$$P_{3+} = 1 - (q^{15} + (15!/(14)! * p * q^{14} + (15!/2(13)!) * p^2 * q^{13}))$$

Equação 7:

$$\boxed{P_{3+} = 1 - (q^{15} + (15 * p * q^{14} + (105 * p^2 * q^{13})))}$$

Quando são transmitidos pacotes protegidos DM, a Equação 7 é utilizada para estimarmos para cada 15 bits se há bits corrompidos. Para cada grupo de 15 bits comparamos P_{3+} com um número gerado pseudo-aleatoriamente entre 0 e 1. Se esse número for maior do que P_{3+}, então ocorreu erro no grupo.

Para o cabeçalho de todos os pacotes a distribuição binomial é a mesma, mas como a proteção FEC é maior, denominada FEC de 1/3 (a cada grupo de 3 bits, um pode ser corrigido em caso de erro), calculamos a probabilidade de ocorrer dois ou mais bits errados num grupo de 3 bits.

Equação 8:

$$P_{2+} = 1 - (P_0 + P_1)$$

Equação 9:

$$P_{2+} = 1 - (q^n + n!/(n-1)! * p * q^{n-1})$$

Como queremos calcular a probabilidade para grupos de 3 bits, então a Equação 9 pode ser vista da seguinte forma:

$$P_{2+} = 1 - (q^3 + 3!/(3-1)! * p * q^{3-1})$$

Equação 10:

$$\boxed{P_{2+} = 1 - (q^3 + 3 * p * q^2)}$$

Quando o número de erros de cada região do pacote está determinado, vamos simular a proteção contra erro (FEC) que cada região do pacote possui. Os cabeçalhos de todos os pacotes Bluetooth possuem proteção FEC de 1/3. Já na área de dados, apenas os pacotes DM possuem proteção FEC de 2/3, isso significa que a cada grupo de 15 bits, erros simples são corrigidos, e erros duplos podem ser detectados. Esse ator possui um método que corrompe bits de acordo com o tipo de erro ocorrido.

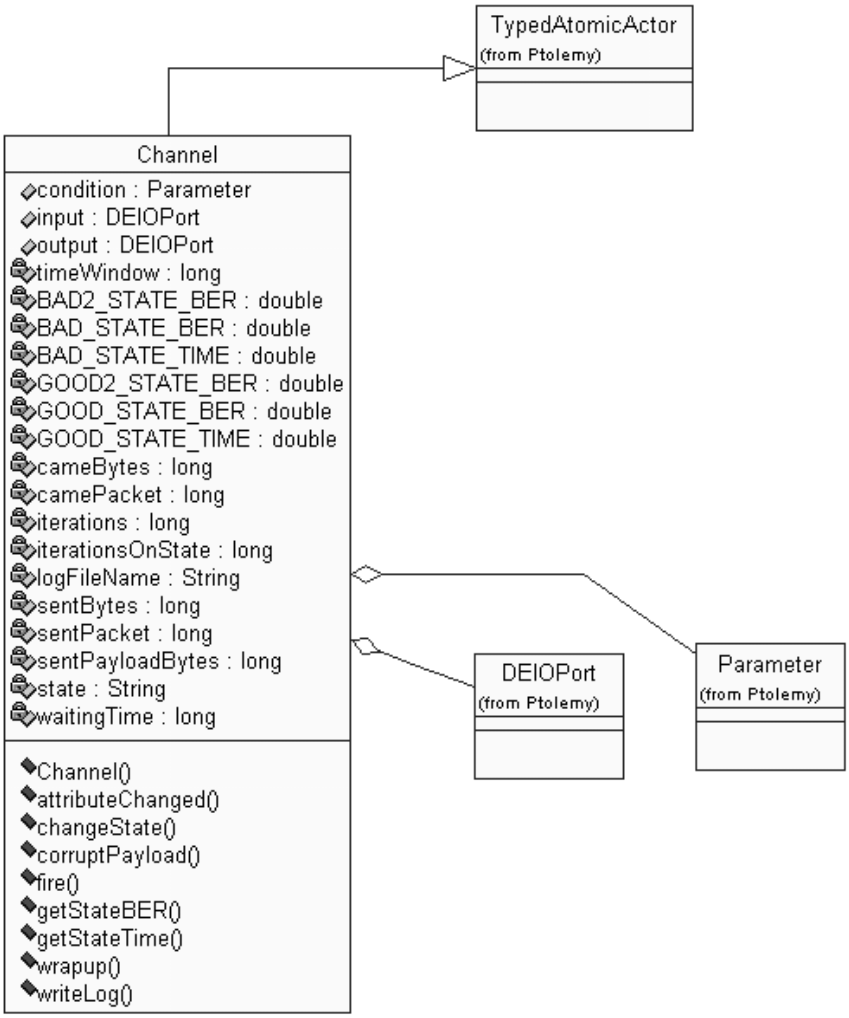

Figura 5-21: Diagrama de classes detalhando a classe *Channel*.

Como todos os pacotes passam pelos canais, decidimos que esse objeto é o ideal para armazenar informações estatísticas (número de pacotes trafegados, porcentagem de pacotes perdidos, tempo de transmissão e número de bits corrompidos) sobre as transmissões. Essas informações são armazenadas através do método *writeLog*.

5.3.4. VIDEOSOURCE

Essa classe também estende a classe *TypedAtomicActor* (ver Figura 5-22), ela possui duas portas para se comunicar com os servidores de vídeo. Uma de saída, por onde os vídeos são enviados, e outra de entrada, por onde ele recebe a realimentação. O servidor avisa a fonte através dessa porta quando ela deve iniciar e encerrar sua transmissão.

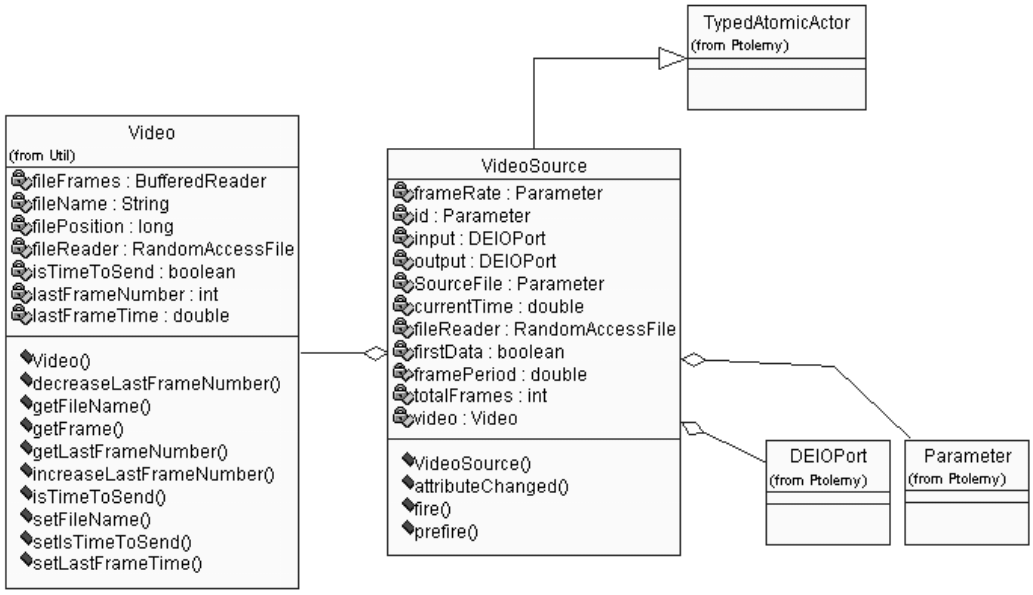

Figura 5-22: Diagrama de classes detalhando o objeto *VideoSource*.

A fonte possui um atributo do tipo *Video* que é o objeto que representa o vídeo que será transmitido. Cada fonte possui apenas um vídeo. A classe *Video* lê dois arquivos, um é o arquivo de configuração do vídeo e outro é o vídeo propriamente dito. No arquivo de configuração estão armazenados o número de quadros que o vídeo possuí, e o tamanho de cada quadro. Isso é necessário porque o sistema funciona baseado na taxa de quadros. Se a transmissão for feita em 30 quadros por segundo, a fonte de vídeo não pode transmitir a uma taxa diferente dessa. A classe *Video* sabe quando é o momento de transmitir o próximo quadro e quando esse momento chega, ela envia para a fonte apenas o quadro da vez. O arquivo de configuração é gerado por um programa implementado por nós que lê o arquivo de vídeo MPEG-4 e identifica nele o número de quadros e o tamanho de cada quadro.

Os parâmetros configuráveis das fontes são o seu código de identificação, o nome do arquivo de vídeo MPEG-4, e a taxa de quadros na qual a transmissão será realizada. Essa taxa deve ser a mesma configurada para os clientes, já que os clientes devem transmitir na mesma taxa em que as fontes estão produzindo os vídeos.

5.4. Simulação

Todos os cenários simulados contêm um servidor e vários clientes. Não simulamos um cenário com vários servidores porque o desempenho do sistema seria o mesmo. Em cada simulação é transmitido um vídeo para um ou mais clientes. Em cada simulação transmitimos

sempre o mesmo vídeo para podermos ter uma base de comparação entre as simulações. Desta forma, enfatizamos o efeito da taxa de bits variável na transmissão de vídeo porque todos as transmissões terão demandas de taxas altas ao mesmo tempo.

O servidor tem que atender a todos os clientes. Ele armazena referências aos clientes numa fila circular. Na janela de tempo em que o servidor tem que enviar um pacote, o cliente da vez é escolhido, e o servidor envia um pacote para ele, seja com dados, ou não.

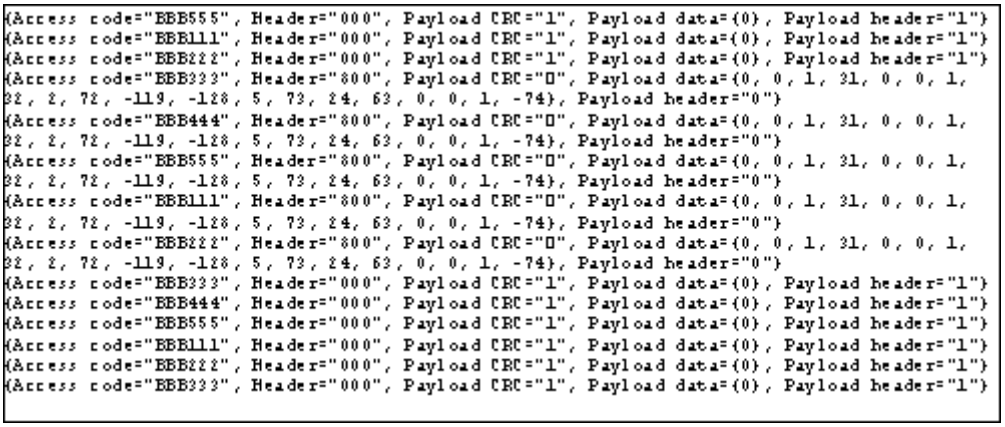

Figura 5-23: Histórico dos pacotes enviados pelo servidor numa simulação de um servidor enviando vídeos para cinco clientes.

Quando tem um quadro para enviar para um cliente o servidor o envia dividido em vários pacotes. O servidor envia um pacote de dados e o cliente responde com um pacote de reconhecimento (apenas com o cabeçalho) que ocupa apenas uma janela de tempo. Quando o servidor não tem dados para enviar para um cliente ele envia um pacote nulo (apenas com o cabeçalho) que ocupa apenas uma janela de tempo. Recebendo esse pacote o cliente sabe que deve aguardar o próximo pacote com dados.

Quando o servidor possui muitos clientes e tem quadros para mandar para mais de um cliente, ele envia um pacote para cada cliente, um em cada janela de tempo, assim, quanto mais clientes o servidor possui, maior é o atraso entre pacotes recebidos pelo cliente.

Na Figura 5-23 podemos observar o histórico de uma transmissão onde um servidor envia vídeos para cinco clientes. Nesse histórico estão apenas os pacotes que foram enviados pelo servidor. Cada pacote possui no campo *Access Code* a informação que identifica seu remetente e destinatário. Quando o servidor não possui dados para enviar para os clientes ele apenas envia um pacote nulo (*header* = "000") com o número 0 como dado útil. Já quando há dados a enviar para os clientes, o servidor envia pacotes não nulos com dados úteis. O servidor está sempre dividindo seu tempo entre os clientes para que nenhum deles sofra

atrasos diferenciados. Nesse exemplo o quadro enviado para os clientes é pequeno e apenas um pacote foi suficiente para transmiti-lo.

5.5. Sumário

Simular dispositivos Bluetooth requer a observação de vários detalhes. Os protocolos devem ser respeitados, o canal deve ser bem modelado, a simulação do tempo real deve ser rigorosa. O Ptolemy apresenta as ferramentas necessárias para executarmos as simulações. Sua flexibilidade e extensibilidade foram determinantes em nossa escolha por este ambiente.

Implementamos quatro componentes, os quais, juntos, são capazes de modelar praticamente qualquer configuração do sistema que estamos desenvolvendo.

Capítulo 6

6. Resultados

Todos os esforços desse trabalho nos levaram a apurar resultados que fortificam as afirmações feitas até aqui sobre a importância e a consistência de uma transmissão de vídeo entre dispositivos Bluetooth.

Esse capítulo apresenta como foram executadas as simulações, quais os objetivos almejados, quais os resultados obtidos e qual a relevância deles.

Para as simulações, adotamos cenários coerentes com as situações reais mais freqüentes, onde dados relevantes são gerados. Esses cenários são apresentados na Seção 6.2, bem como, a maneira como esses cenários foram utilizados.

Na Seção 6.3 são apresentados os resultados obtidos nas baterias de simulações executadas nos cenários, juntamente com uma análise de cada resultado, ressaltando sua relevância frente a esse trabalho. A Seção 6.4 conclui esse capítulo.

6.1. Cenários da Simulação

Todas as simulações foram feitas com um servidor transmitindo vídeos para vários clientes. Testamos como o sistema se comporta frente às seguintes variáveis:

- Número de clientes;
- Condições do canal (BER);
- Tamanho do *buffer* dos clientes;
- Tipo dos pacotes;
- Tamanho dos pacotes;
- Taxa de quadros.

As denominações que adotamos para as condições do canal, na ordem de maior para menor taxa de erros, são: canal péssimo (BAD2), canal ruim (BAD), canal normal (REAL), canal bom (GOOD) e canal ótimo (GOOD2). Testamos os servidores nesses canais transmitindo para dois, até sete clientes. Foram gerados mais de 700 arquivos de vídeos resultantes.

Executamos duas baterias de testes, na primeira simulamos quatro cenários e comparamos com um cenário ideal. O cenário ideal possui *buffer* grande, pacotes grandes sem proteção e uma pequena taxa de quadros que não sobrecarrega o sistema. Aqui o servidor transmite o vídeo de teste *Container* (http://meru.cecs.missouri.edu/free_download/videos), para cada cliente, esses armazenam num arquivo os vídeos que recebem, simulando a condição do vídeo numa situação real. Para cada arquivo recebido, avaliamos sua qualidade através da relação sinal/ruído (*Peak Signal-to-Noise Ratio* - PSNR). Essa relação é uma das mais usadas para medir objetivamente qualidade de vídeos. Para medir essa relação, utilizamos o programa "Videometer" da empresa "Acticom". Este programa testa vídeos no formato YUV. Como nossos vídeos estão no formato MP4, transformamos os vídeos em AVI, para daí convertê-los em YUV e poderem ser testados. Para codificação e decodificação dos vídeos MPEG-4, utilizamos o programa "3ivx D4 4.0", que é o codificador/decoficador (CODEC) MPEG-4 que mais suportou a codificação dos vídeos parcialmente corrompidos. Outros codecs não suportaram a decodificação de alguns vídeos e travavam durante o processo.

Na segunda bateria de simulações, comparamos a transmissão de dois cenários, um com pacotes DH5 (grandes e sem proteção), e outro com pacotes DM5 (grandes e com proteção). Nessa seqüência de simulações, transmitimos os vídeos de teste *Container* e *Foreman* (http://meru.cecs.missouri.edu/free_download/videos) e comparamos a quantidade total de dados perdidos, além do número de pacotes perdidos por erro do canal entre os dois cenários. O objetivo dessa bateria de simulações é testar a eficácia dos pacotes protegidos (DM) em transmissões de vídeo. A seguir são apresentados os cenários simulados:

Cenário 1 (ideal):

- *Buffer* de cada cliente: 1000 bytes.
- Tipo dos pacotes: sem proteção (DH)
- Tamanho dos pacotes: 5 janelas de tempo (DH5)

- Taxa de quadros: 5 quadros por segundo (para isso o vídeo original foi codificado em "câmera lenta").

Cenário 2 (altas taxas de quadros):

- *Buffer* de cada cliente: 1000 bytes.
- Tipo dos pacotes: sem proteção (DH)
- Tamanho dos pacotes: 5 janelas de tempo (DH5)
- Taxa de quadros: 10 quadros por segundo.

Cenário 3 (pacotes protegidos):

- *Buffer* de cada cliente: 1000 bytes.
- Tipo dos pacotes: com proteção (DM)
- Tamanho dos pacotes: 5 janelas de tempo (DM5)
- Taxa de quadros: 5 quadros por segundo.

Cenário 4 (*buffer* pequeno):

- *Buffer* de cada cliente: 300 bytes.
- Tipo dos pacotes: sem proteção (DH)
- Tamanho dos pacotes: 5 janelas de tempo (DH5)
- Taxa de quadros: 5 quadros por segundo.

Cenário 5 (pacotes pequenos):

- *Buffer* de cada cliente: 1000 bytes.
- Tipo dos pacotes: sem proteção (DH)
- Tamanho dos pacotes: 3 janelas de tempo (DH3)
- Taxa de quadros: 5 quadros por segundo.

6.2. Resultados

6.2.1. EFEITO DA TAXA DE QUADROS

Nessa seção apresentaremos os resultados encontrados sobre a influência da taxa de quadros sobre a qualidade dos vídeos. Para tanto, efetuamos dois testes, um com uma pequena taxa de apenas 5 quadros por segundo (Cenário 1) e outro com o dobro dessa taxa (Cenário 2).

A Tabela 6-7 apresenta a média da relação sinal/ruído (PSNR) entre o vídeo original e os vídeos transmitidos com 5 e 10 quadros por segundo (*Frame per Second* - fps) com vários clientes. Observamos que, em média, a diferença de qualidade entre os dois cenários é de 7dB. A medida em que o número de clientes mantido pelo servidor aumenta, observamos uma queda na qualidade dos vídeos, independentemente das condições do canal e da taxa de quadros. Isso porque quanto mais clientes o servidor possui, maior é o atraso entre dois pacotes recebidos pelos clientes. Quando o atraso aumenta, o cliente não consegue receber os quadros a uma taxa de bits satisfatória para recepção do vídeo em tempo real.

Clientes	5fps	10fps
2	33,32	21,87
3	25,13	18,16
4	21,84	14,39
5	18,79	11,74
6	15,29	9,70
7	13,29	6,85

Tabela 6-7: Qualidade média (dB) com 5 e 10 quadros por segundo.

No Gráfico 6-1 apresentamos a qualidade dos vídeos em cada tipo de canal. Não apresentamos os dados para o canal de condição péssima (BAD2) porque os vídeos não tiveram qualidade suficiente para serem testados pelos programas utilizados. No Gráfico 6-1 fica visível que no canal normal (REAL) a qualidade é mediana, fica entre a qualidade do canal ruim (BAD) e do canal bom (GOOD). Já o canal ótimo (GOOD2) apresenta sempre maior qualidade do que os outros canais. Quando a taxa de quadros é dobrada (Cenário 2), podemos observar que, mesmo quando o canal está na condição boa (GOOD), a qualidade dos vídeos é menor do que a qualidade na condição ruim (BAD), quando são utilizados apenas 5 quadros por segundo (Cenário 1). Isso demonstra que o sistema é sensível aos efeitos das taxas de quadros. Quanto maior a taxa de quadros, maior o número de bits por segundo que o servidor tem que enviar para seus clientes. Quando a taxa de quadros aumenta demais, o sistema não é capaz de manter a transmissão entre muitos clientes.

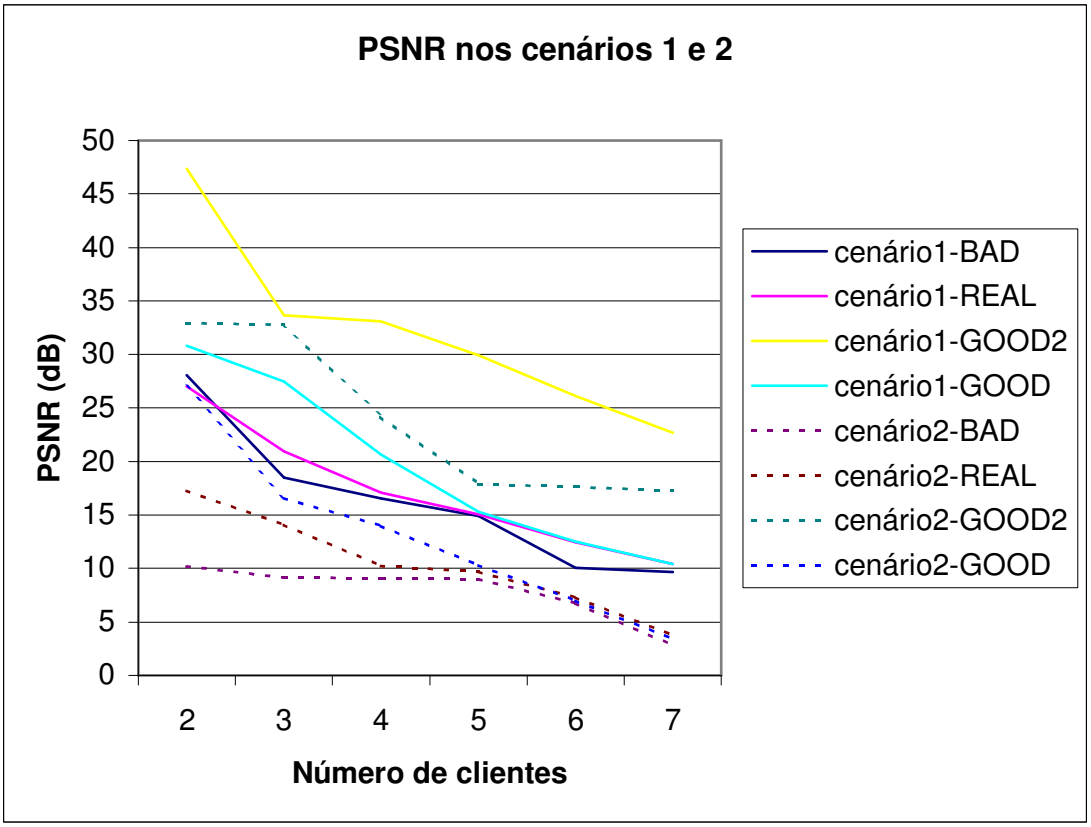

Gráfico 6-1: Comparativo de PSNR entre os cenários 1 e 2, em diversos canais e com vários clientes.

O Gráfico 6-2 apresenta a porcentagem total de dados perdidos no Cenário 1 e no Cenário 2. Essas perdas são causadas pelo excesso de congestionamento do canal e pela perda de pacotes causada por erros de bit nos cabeçalhos dos pacotes. Quando a taxa de quadros aumenta, o servidor tem que transmitir a uma taxa de bits maior, chegando a um limite insuportável para a tecnologia. Notemos que a curva é coerente com a curva da qualidade dos vídeos, quanto maior a perda de dados, menor a qualidade dos vídeos.

Na condição péssima (BAD2) a perda de dados é muito grande, chegando até a 70% de perda quando o servidor possui 7 clientes e transmite a 10 quadros por segundo (Cenário 2). Isso justifica a grande queda de qualidade dos vídeos nessas condições de canal, impossibilitando os programas de avaliarem esse vídeo.

Calculamos que, em média, quando dobramos a taxa de quadros de 5 para 10 quadros por segundo, aumentamos em 11% a perda de dados dos vídeos.

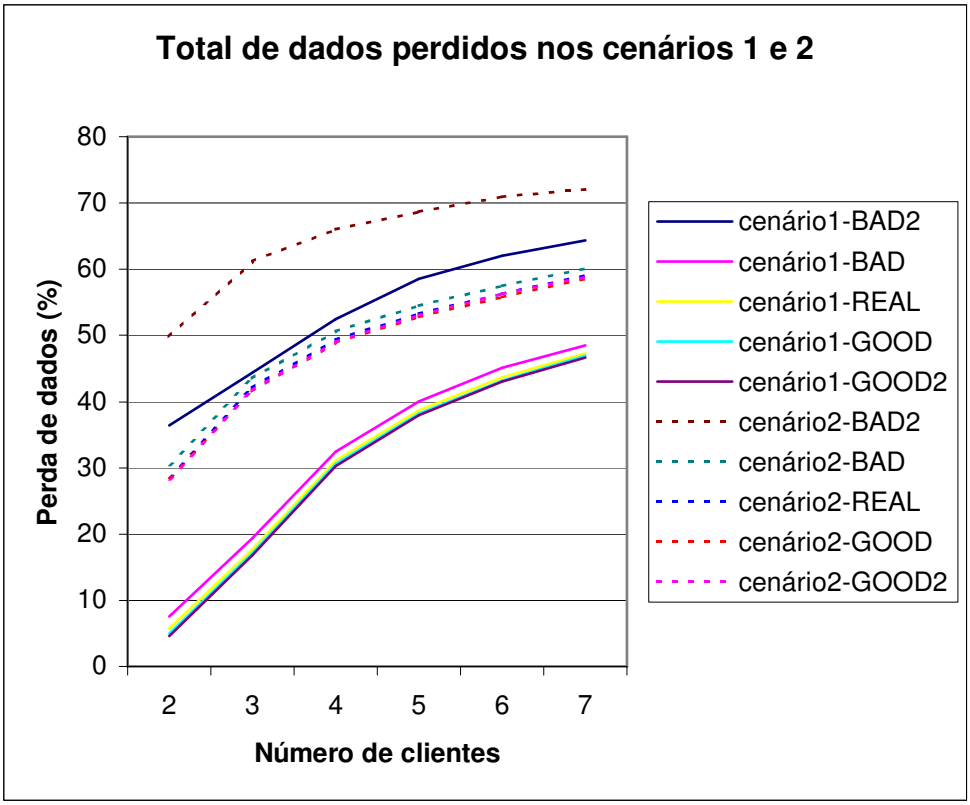

Gráfico 6-2: Porcentagem total de bytes perdidos nos cenários 1 e 2.

6.2.2. IMPORTÂNCIA DO *BUFFER*

O tamanho do *buffer* dos clientes está diretamente relacionado com a qualidade dos vídeos recebidos. Quanto menor for o *buffer*, menor será a taxa de bits que os clientes suportam receber dos servidores. O ideal é que o tamanho do *buffer* nunca seja menor do que o tamanho do maior quadro que será transmitido. A dificuldade em sistemas reais é saber que tamanho é esse.

Para testarmos o efeito do tamanho do *buffer*, executamos duas seqüências de testes, uma com clientes com o *buffer* de 1000bytes (Cenário 1) e outra com clientes com 300bytes de *buffer* (Cenário 4). Em outros testes constatamos que o buffer máximo para os vídeos que testamos é de 1000bytes, e que *buffers* maiores do que isso não resultam em melhoria na recepção dos vídeos.

Dos cinco cenários simulados, esse foi o que obteve os piores resultados em termos de qualidade. A Tabela 6-8 apresenta um comparativo entre a qualidade dos vídeos nesses dois cenários. Constatamos que, em média, os vídeos recebidos pelos clientes com 300bytes de

buffer possuem qualidade 13dB menor do que os vídeos recebidos pelos clientes de 1000bytes de *buffer*.

Clientes	1000 bytes	300 bytes
2	33,32	10,82
3	25,13	8,36
4	21,84	8,67
5	18,79	5,99
6	15,29	5,91
7	13,29	5,68

Tabela 6-8: Qualidade dos vídeos com *buffer* dos clientes em dois tamanhos diferentes.

O Gráfico 6-3 apresenta os valores da qualidade dos vídeos para cada condição do canal. Podemos observar que em nenhuma condição do canal foi melhor transmitir com 300bytes de *buffer* nos clientes (Cenário 4). Nesse cenário, na melhor condição do canal (GOOD2), o sistema obteve piores resultados do que com 1000bytes de *buffer* (Cenário 1) na condição ruim do canal (BAD).

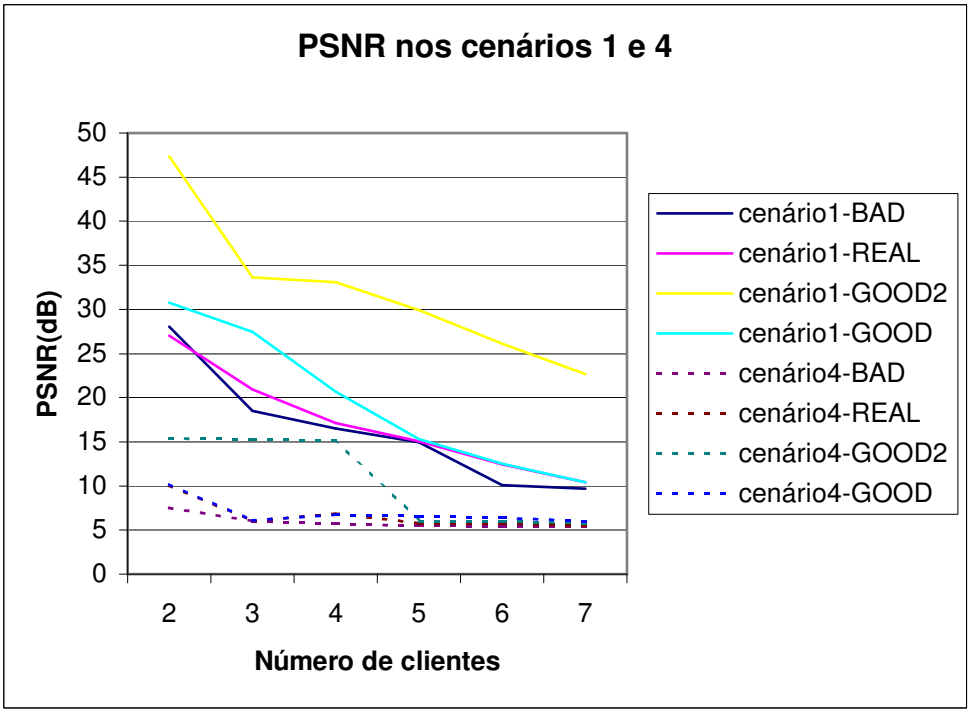

Gráfico 6-3: Qualidade dos vídeos com *buffer* grande e pequeno.

Quando os clientes possuem 300bytes de *buffer*, a medida em que o número de clientes aumenta, a tendência é, em todas as condições de canais, a qualidade dos vídeos se igualar num ponto crítico, próximo a 5dB (todos os vídeos considerados perdidos, com alta taxas de erros, resultaram em qualidade próxima a 5dB).

Já no Gráfico 6-4 podemos observar a taxa de perdas de dados no Cenário 1 e no Cenário 4. A perda de dados no Cenário 4 é sempre maior do que a perda no Cenário 1.

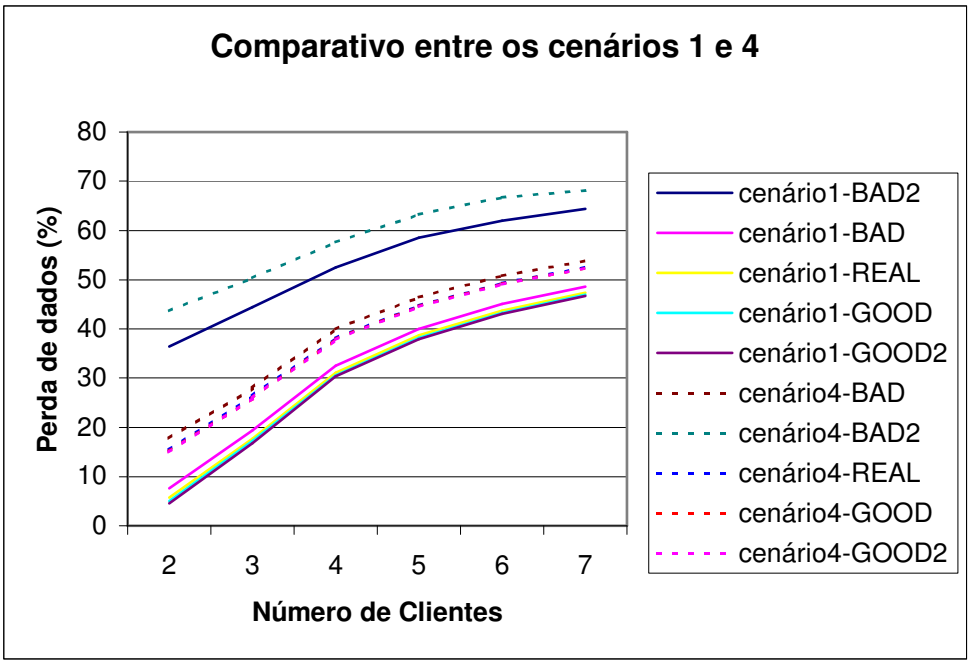

Gráfico 6-4: Perda média de dados nos cenários 1 e 4.

6.2.3. EFEITO DO TAMANHO DO PACOTE

Nesses testes vamos analisar como a diminuição do tamanho dos pacotes afeta a qualidade dos vídeos. No Cenário 1 são transmitidos pacotes DH5, ocupando 5 janelas de tempo e sem nenhum tipo de redundância para proteger os bits da área de dados. É o maior dos pacotes Bluetooth e o que atinge maior taxa de bits (até 723,32Kbps). Já no Cenário 5 são utilizados pacotes DH3, também não possuindo nenhum esquema de proteção na área de dados, porém ocupando apenas 3 janelas de tempo e atingindo menores taxas de bits (no máximo 585,6Kbps).

Na Tabela 6-9 apresentamos a qualidade média dos vídeos transmitidos com esses dois tipos de pacote. A diferença de qualidade dos vídeos entre esses dois cenários é, em média, de 4,7dB, a menor diferença de qualidade entre os cenários simulados e o Cenário 1.

Clientes	Pacote DH5	Pacote DH3
2	33,32	25,02
3	25,13	22,19
4	21,84	17,47
5	18,79	13,46
6	15,29	12,46
7	33,32	25,02

Tabela 6-9: Qualidade média dos vídeos (dB) com pacotes grandes e médios.

Podemos observar pelo Gráfico 6-5 que a diferença de qualidade dos vídeos transmitidos nesses dois cenários não é muito grande. Em canais onde as interferências são menores, como nos canais de boa (GOOD) e ótima (GOOD2) condição, a qualidade dos vídeos chega a se igualar, e até a superar a qualidade dos vídeos transmitidos com pacotes grandes.

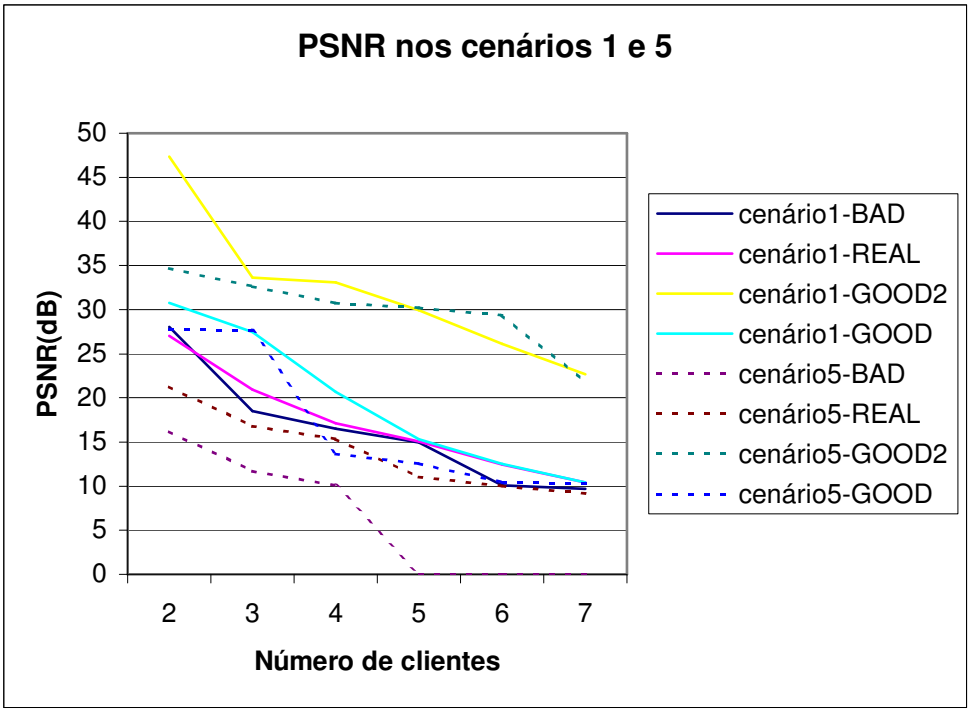

Gráfico 6-5: Qualidade dos vídeos com pacotes grande e médio.

Há dois fatores que influenciam a qualidade dos vídeos quando tratamos de tamanho de pacotes: taxa de transmissão e susceptibilidade a erros. Os pacotes DH5 transmitem em taxas maiores de bits, mas são mais susceptíveis a erros, e a perda de um pacote desse tipo influencia mais negativamente a qualidade dos vídeos. Enquanto que os pacotes DH3 transmitem a taxas de bits menores, mas sofrem menos com as influências do canal. Isso faz com que a qualidade dos vídeos nesses dois cenários não seja tão diferente.

Como podemos observar no Gráfico 6-6, as perdas de dados entre esses dois cenários não são muito diferentes. Quando os vídeos são transmitidos no Cenário 5, são perdidos, em média, 6% mais dados do que no Cenário 1. A menor diferença de perdas de dados quando comparamos os outros cenários com o Cenário 1.

Também podemos observar que a maior diferença entre as perdas de dados no Cenário 5 e Cenário 1 ocorre quando o servidor possui 3 e 4 clientes, já quando o servidor possui menos de 3 clientes, a perda de dados é praticamente a mesma quando utilizamos pacotes DH5 e pacotes DH3, entretanto quando são utilizados mais de 4 clientes, a diferença entre as perdas de dados nos dois cenários diminui novamente. A curva das perdas de dados do Cenário 1 é mais suave do que a curva do Cenário 5.

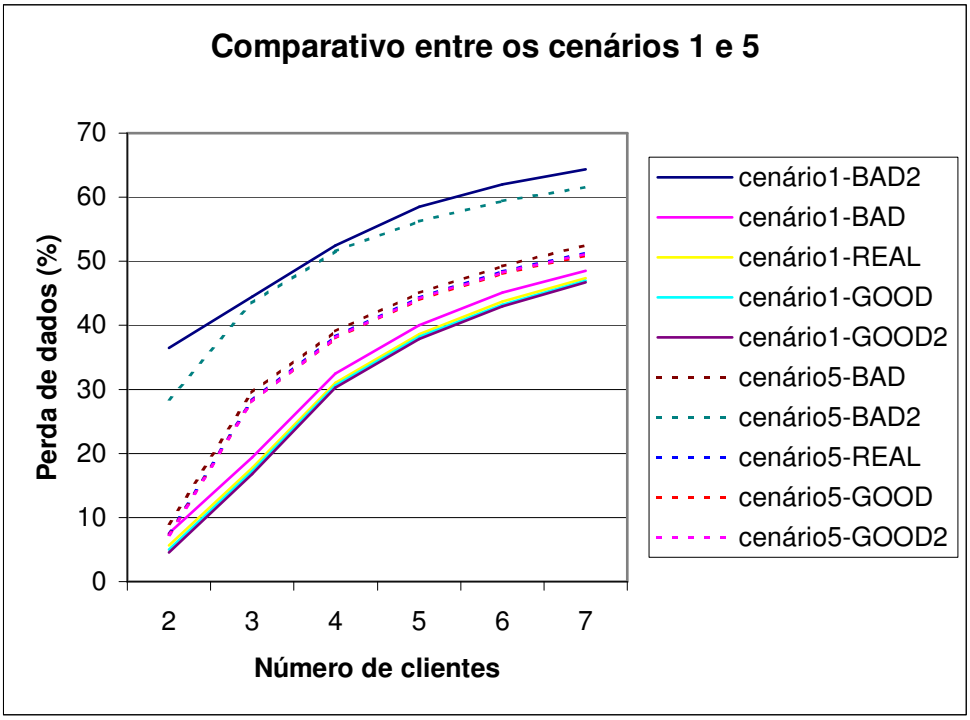

Gráfico 6-6: Perda média de dados nos cenários 1 e 5

6.2.4. EFEITO DOS PACOTES PROTEGIDOS

Essa seção apresenta um comparativo de desempenho entre o Cenário 1 e o Cenário 3. A única diferença entre esses dois cenários é o tipo de pacote que eles utilizam. No Cenário 1 o pacote utilizado é o DH5 (grande e sem proteção), já no Cenário 3 o pacote utilizado é o DM5 (grande e com proteção), que possui o esquema de proteção FEC em sua área de dados, protegendo mais os dados dos efeitos do canal. O que esperávamos era que o Cenário 3 apresentasse resultados mais satisfatórios, principalmente em situações onde o canal exerce mais interferência sobre os dados, mas o encontrado não foi exatamente isso.

No Gráfico 6-7 podemos observar a relação sinal/ruído entre o vídeo original e o transmitido nas simulações.

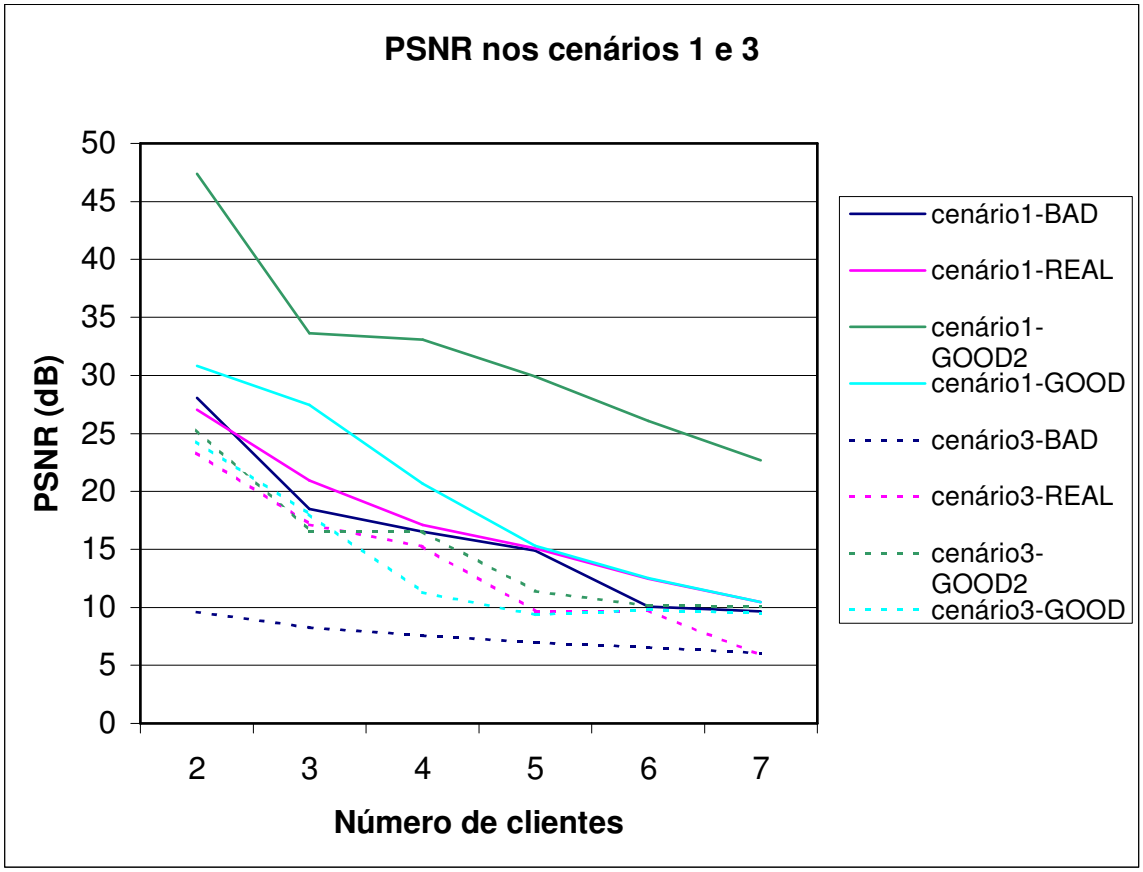

Gráfico 6-7: Comparativo de PSNR entre os cenários 1 e 3, em diversos canais e com vários clientes.

Podemos observar no Gráfico 6-7 que a qualidade dos vídeos transmitidos no Cenário 3, diferentemente do esperado, é pior do que a qualidade dos vídeos no Cenário 1. Isso quer dizer que a proteção dos pacotes DM5 não está conseguindo manter a qualidade dos vídeos com os erros do canal. Supomos que isso se deve ao fato dos pacotes DM5 transportarem menos dados do que os pacotes DH5, devido à redundância embutida em suas áreas de dados. Como transportam menos dados, são necessários mais pacotes quando usamos pacotes DM5 do que quando usamos pacotes DH5.

A redundância na área de dados diminui a incidência de erros, mas não evita que pacotes sejam perdidos. O erro dos bits dos cabeçalhos dos pacotes causa perda de pacotes. Para proteger o cabeçalho, tanto os pacotes DH5, quanto os DM5 (assim como todos os pacotes de dados), possuem proteção FEC. Supomos então que os maiores causadores da diminuição da qualidade dos vídeos são as perdas de pacotes devido ao congestionamento do canal, que causa o estouro dos *buffers* dos servidores, e não devido ao excesso de erros que o canal insere nos cabeçalhos dos pacotes.

O Gráfico 6-8 apresenta um comparativo entre a quantidade total de dados úteis perdidos durante transmissões no Cenário 1 e no Cenário 3. Podemos observar que realmente a quantidade de dados perdidos quando o servidor utiliza pacotes DM5 (Cenário 3) é maior do que quando ele utiliza pacotes DH5 (Cenário 1), havendo, em média, uma diferença de 11% mais perdas no Cenário 3 do que no Cenário 1.

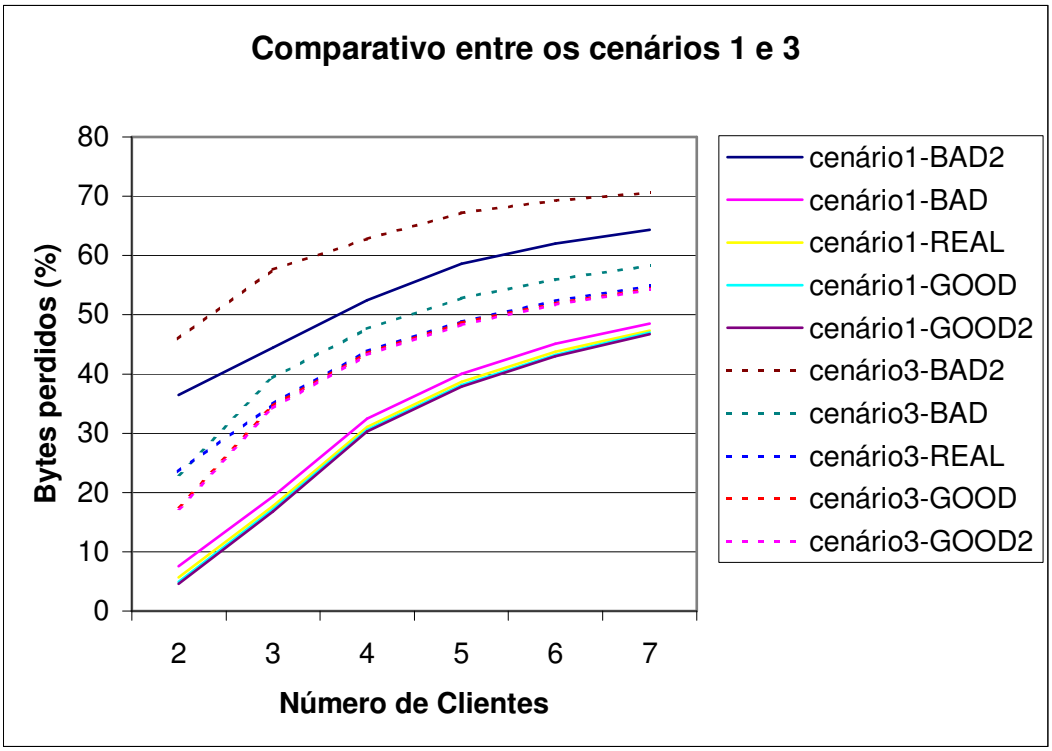

Gráfico 6-8: Porcentagem de bytes perdidos nos cenários 1 e 3.

As razões para essa diferença são visíveis nos gráficos abaixo. O Gráfico 6-9 apresenta a porcentagem de pacotes perdidos devido aos erros inseridos pelo canal sobre seus cabeçalhos. Podemos notar que as perdas de pacotes nos dois cenários são praticamente iguais, isso porque os cabeçalhos dos pacotes DH5 são tão protegidos quanto os cabeçalhos dos pacotes DM5. Isso demonstra que a diferença de qualidade entre os vídeos transmitidos no Cenário 1 e no Cenário 3 não é decorrente das interferências inseridas pelo canal, mas principalmente pela pequena taxa de bits que os pacotes DM5 atingem, que faz com que o sistema não suporte o congestionamento causado pelo aumento no número de clientes.

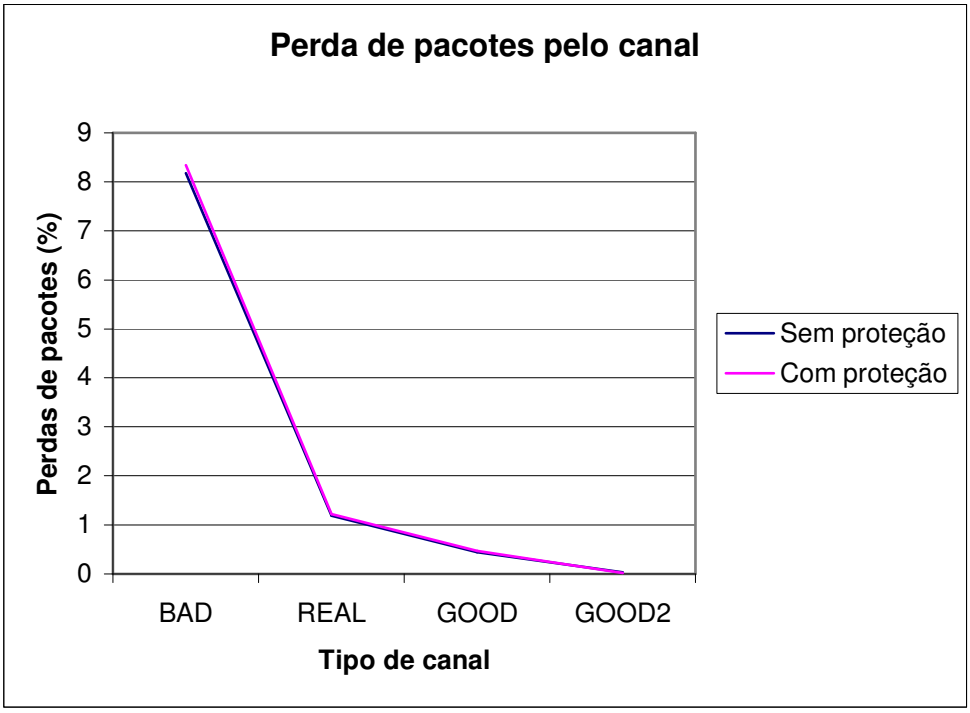

Gráfico 6-9: Porcentagem de pacotes perdidos devido aos erros inseridos pelo canal.

O Gráfico 6-10 mostra que realmente os pacotes DM5 protegem mais os pacotes contra erros do que os pacotes DH5, mas isso não é suficiente para evitar as altas taxas de perdas de dados, que, como vimos, são causadas principalmente pelo congestionamento do canal.

Podemos concluir que, nas condições de nossas simulações:

Para transmissões de vídeo, os pacotes DH5, sem proteção, são mais eficientes do que os pacotes DM5.

Vimos que, as taxas de bits mais elevadas que os pacotes DH5 suportam são mais úteis do que a proteção dos pacotes DM5. Esta proteção deve ser mais útil para transmissões onde perdas não são admissíveis e onde ocorre retransmissão em caso de perda, como arquivos binários, por exemplo. Sugerimos que:

Para a transmissão de vídeo, seria mais útil um tipo de pacote com proteção extra no cabeçalho, ao invés de proteção na área de dados.

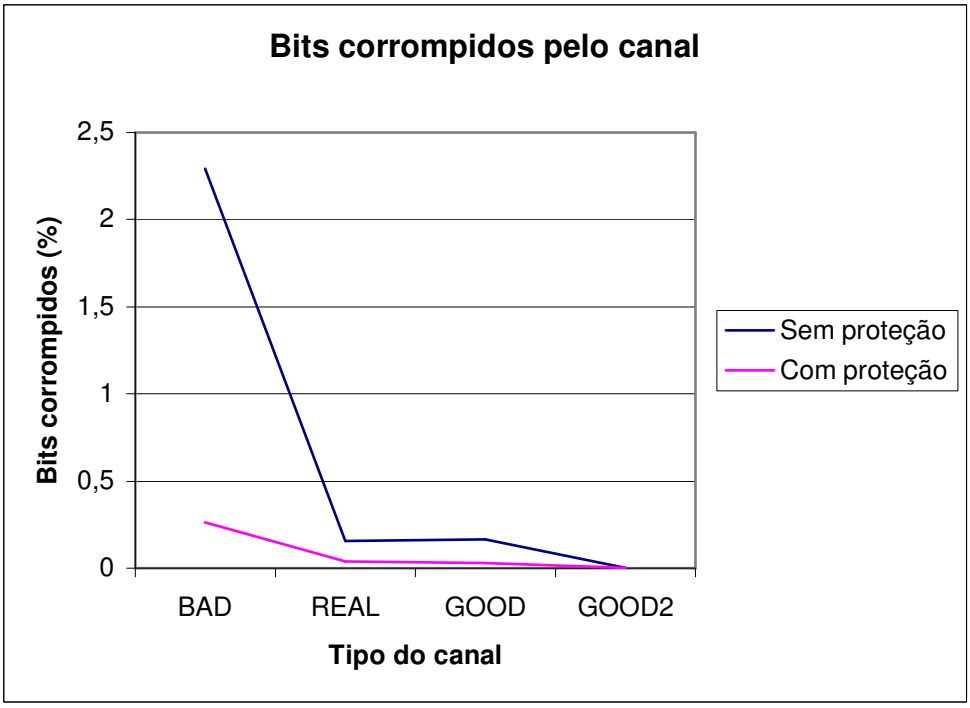

Gráfico 6-10: Porcentagem de bits corrompidos pelas interferência do canal em transmissões com pacotes sem, e com proteção.

6.2.5. COMPARAÇÃO ENTRE OS CENÁRIOS

Em geral, como havíamos mencionado, a qualidade dos vídeos diminui bastante com o aumento do número de clientes. No Gráfico 6-11 observamos a qualidade dos vídeos nos cinco cenários, quando todos eles são transmitidos com o canal em sua condição normal (REAL). O cenário que apresenta vídeos com as melhores qualidades é o Cenário 1, e o que resultou em vídeos de piores qualidades é o Cenário 4 (utiliza *buffer* de 300 bytes).

Observamos que vídeos com qualidades inferiores a 15dB podem ser considerados vídeos perdidos. Para esses vídeos os decodificadores encontraram problemas em exibi-los, e quando exibidos, são imagens de péssima qualidade, onde mal se identifica que vídeo está sendo exibido, por isso, consideramos esses vídeos como perdidos. Da mesma forma que vídeos que perderam em média mais de 40% de seus dados também podem ser considerados perdidos. Com isso, podemos concluir que:

Nas condições de nossos testes, quando um dispositivo utiliza pacotes DH5, buffer de 1000 bytes e taxa de quadros de 5fps, ele só é capaz de manter uma transmissão de vídeo com no máximo 5 clientes.

Outra afirmação importante é que:

A codificação MPEG-4, com os codificadores e decodificadores "3ivx D4 4.0", é capaz de transmitir vídeos, mesmo perdendo até 40% dos dados.

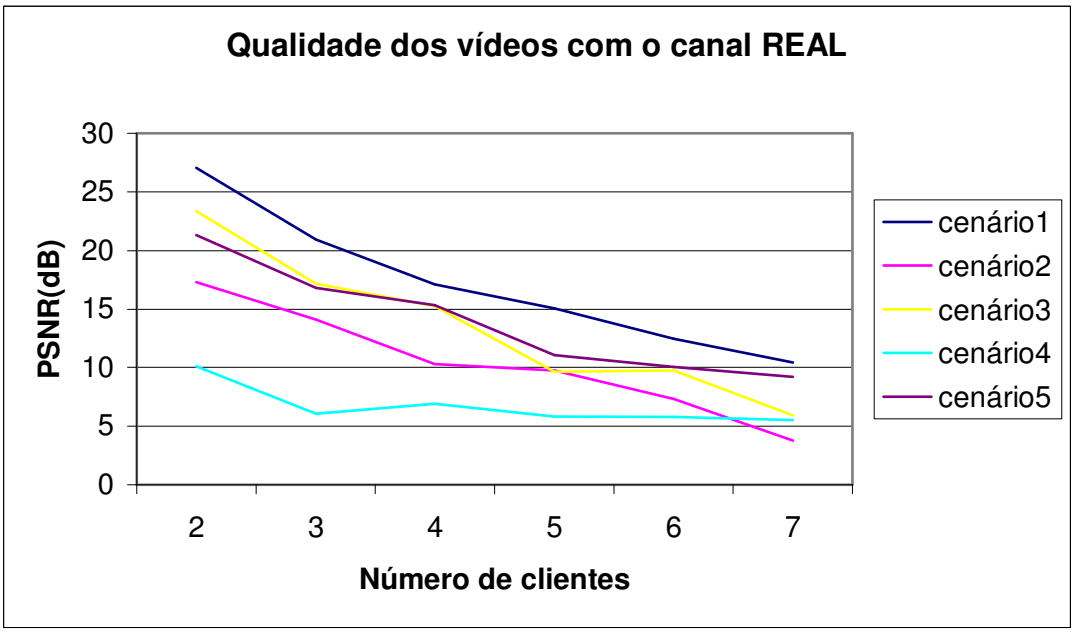

Gráfico 6-11: Comparação entre a qualidade dos vídeos com o canal em seu estado normal.

Já no Gráfico 6-12 é mostrado um comparativo entre a porcentagem de perda de dados nas transmissões de vídeo dos cinco cenários.

Mais uma vez, o cenário com menor perda de dados é o Cenário 1, mas o Cenário 4, com os vídeos de piores qualidades, não é o cenário que apresentou as maiores taxas de perdas de dados.

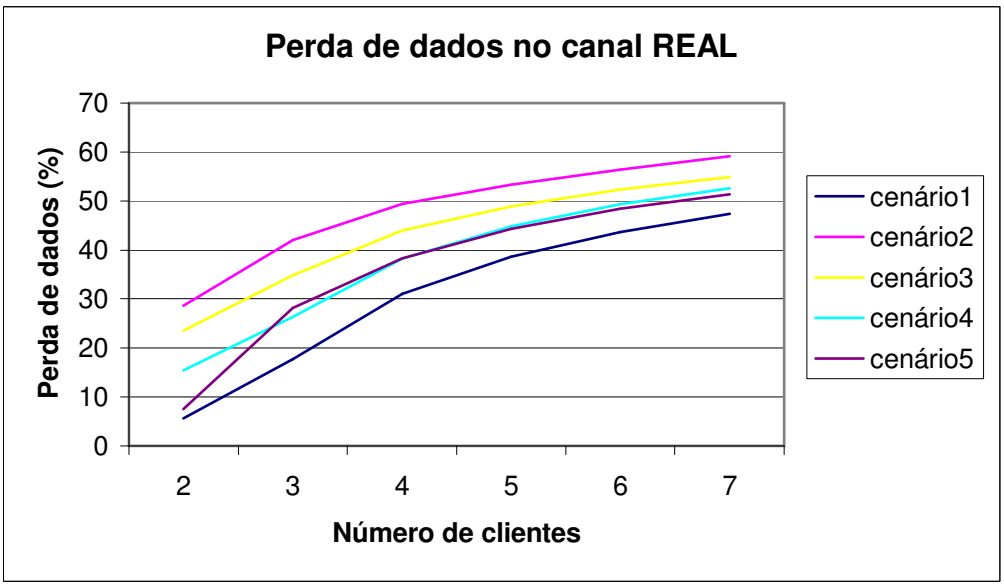

Gráfico 6-12: Perda média de dados no canal normal.

6.3. Sumário

Esse capítulo apresentou todos os resultados encontrados em nossas simulações. As simulações foram executadas divididas em cinco cenários diferentes que representam situações reais. O objetivo dos cenários é testar como o sistema se comporta frente a seis variáveis, tamanho do *buffer*, tipo de pacote, tamanho de pacote, taxa de quadros, tipos de canais e número de clientes. Os resultados foram discutidos e analisados nesse capítulo. Um dos resultados mais interessantes é que a opção por maior proteção FEC na área de dados dos pacotes faz piorar a qualidade dos vídeos recebidos.

Capítulo 7

7. Conclusão

Esse capítulo apresenta as conclusões desse trabalho de Dissertação de Mestrado. Aqui são relatadas as contribuições desse trabalho, uma análise de todos os resultados encontrados e sugestões para trabalhos futuros que possam utilizar essa dissertação como apoio.

7.1. Contribuições

Podemos citar como as principais contribuições desse trabalho:

- Especificamos um sistema de transmissão de vídeo para vigilância que utiliza a tecnologia Bluetooth, mostrando que essa tecnologia adiciona ao sistema mais praticidade, dinamismo, agilidade e segurança.
- Utilizando codificadores e decodificadores MPEG-4, mostramos que, limitada à condição do canal e ao número de clientes, a transmissão de vídeo utilizando a tecnologia Bluetooth é possível e viável.
- Apresentamos informações importantes que podem ser utilizados por outros trabalhos que desejam utilizar a tecnologia Bluetooth;
- Mostramos resultados que demonstram em que condições as redes Bluetooth, aliadas à codificação MPEG-4, são capazes de transmitir vídeo.
- Apresentamos o ambiente Ptolemy e mostramos o quanto ele é poderoso, e que outros trabalhos podem utilizá-lo para modelar e simular tanto redes Bluetooth, quanto qualquer outro tipo de rede.

7.2. Análise dos Resultados

De todos os resultados obtidos e apresentados no capítulo anterior, podemos resumir que, nas condições de nossas simulações:

- Para transmissões de vídeo, transmissões com pacotes DH5 resultam em melhores resultados do que com pacotes DM5;
- Para a transmissões de vídeo, seria mais útil um tipo de pacote com proteção extra no cabeçalho, ao invés de proteção na área de dados;
- Um dispositivo só é capaz de manter uma transmissão de vídeo com no máximo 5 clientes;
- A codificação MPEG-4é capaz de transmitir vídeos para vigilância mesmo perdendo até 40% dos dados;
- Transmitir vídeos a mais de 5 quadros por segundo pode impossibilitar a transmissão;
- Um *buffer* suficiente para os clientes é de 1000 bytes;
- Transmitir vídeos com pacotes DH3 pode ser uma boa solução se o número de clientes for pequeno;
- O maior problema para a transmissão de vídeo sobre Bluetooth são as reduzidas taxas de bits que essa tecnologia oferece.

7.3. Trabalhos Futuros

Após a finalização desse trabalho apresentamos algumas sugestões para trabalhos futuros:

- **Validar as simulações com testes práticos**

  Devido a limitações de tempo e dinheiro, esse trabalho abdicou dos testes práticos, que podem ser feitos em trabalhos futuros.

- **Especificação formal da transmissão de vídeo em Bluetooth**

  Esse trabalho pode ser usado como base para a especificação formal de como deve ser feita a transmissão de vídeo, com pacotes e tipos de conexões próprios.

- **Efetuar testes subjetivos dos vídeos**

Devido a limitação de tempo, não executamos testes subjetivos (onde a avaliação dos vídeos é feita de forma qualitativa por pessoas e não por software) dos vídeos transmitidos, que podem ser feitas em trabalhos futuros.

7.4. Considerações Finais

Unir três tecnologias inovadoras (Ptolemy, Bluetooth e MPEG-4) num mesmo trabalho, apresentando resultados concretos, foi o nosso maior desafio, ao mesmo tempo que foi de gratificação. Foi muito importante para a minha formação desenvolver esse trabalho, que pediu muito da minha capacidade de pesquisa e decisão, e espero que as informações aqui contidas sejam tão importantes para outros pesquisadores, como foram para mim.

Referências Bibliográficas

ARAMVITH, Supavadee, PAO, I-Ming, SUN, Ming-Ting, FELLOW, “A Rate-Control Scheme for Video Transport over Wireless Channels”, IEEE Transactions on Circuits and Systems for Video Technology, v. 11, n. 5, maio de 2001. p. 569-580.

BARNET, Barry, “Basic Concepts and Techniques of Video Coding and the H.261 Standard” , University of Texas at Austin, 2000.

BATRA, P., CHANG, S. F., "Effective algorithms for video transmission over wireless channels", Signal Processing: Image Commun., v. 12, n. 2, p. 147—166, abril de 1998.

BLUETOOTH SIG SECURITY EXPERT GROUP, “Bluetooth Security White Papers”, Abril, 2002. Disponível em http://www.bluetooth.org. Acesso em: 10 dez. 2002.

BLUETOOTH SPECIAL INTEREST GROUP (SIG), “Bluetooth Core Specifications v.1.1.”, February, 2001. Disponível em http://www.bluetooth.org. Acesso em: 10 dez. 2002.

BOVIK, Alan C. “Introduction to Digital Image and Video Processing”, In: Handbook of Image and Video Processing, Edited by Alan C. Bovik, Academic Press, 2000.

BRAY, Jennifer.; Sturman, Charles F. “Bluetooth: connect without cables”. 1. ed. Prentice Hall, 2001.

BRITO, Alisson V., MELCHER, Elmar U. K. “Sistema de Transmissão de Vídeo para Vigilância Utilizando Bluetooth”. Proceedings: VIII Brazilian Symposium on Multimedia and HyperMedia Systems – SBC. Fortaleza, Brazil, 2002.

CÔTÉ, Guy, EROL, Berna, GALLANT, Michael, KOSSENTINI, Faouzi, “H.263+: Video Coding at Low Bit Rates”, IEEE Transactions on Circuits and Systems for Video Technology, v. 8, n. 7, novembro de 1998, p. 849-866.

COX, R., KROON, P., “Low bitrate speech coders for multimedia communication”, IEEE Communications, v. 34, dezembro de 1996, p. 34–41.

DEECS (Department of Electrical Engineering and Computer Sciences), “Ptolemy II, Heterogeneous Concurrent Modeling and Design in Java”, agosto de 2002. University of California at Berkeley. Disponível em http://ptolemy.eecs.berkeley.edu. Acesso em: outubro de 2002.

EKLUND, Carl; MARKS, Roger B.; STANWOOD, Kenneth L.; WANG, Stanley. “A

Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access", IEEE Communications Magazine, junho de 2002, p. 98-107.

GHOSE, Das, A., RAZDAN, A., SARAN, H. e SHOREY, R., "Enhancing performance of asynchronous data traffic over the bluetooth wireless ad-hoc network". Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), vol. 1, março de 2001, p. 591-600.

GREEN, James H. "The Irwin Handbook of Telecommunications", 4. ed. McGraw-Hill, 2000.

HAGENAUER, J., BUCH, G., BURKERT, F., KUKLA, B., "To compress or not to compress?", in Proceedings of CTMC in conjuntion with Globecom'96, Londres, novembro de 1996.

HAGENAUER, Joachim, STOCKHAMMER, Thomas "Channel Coding and Transmission Aspects for Wireless Multimedia", Proceedings of The IEEE Special Issue On Video Transmission for Mobile Multimedia Applications v. 87 , p. 1764-1777, outubro de 1999.

HAZZAN, Samuel, "Fundamentos de Matemática Elementar", v. 5, "Combinatória e Probabilidade". 6. ed. – São Paulo, Atual, p.139, 1993.

HOUAISS, Antônio, VILLAR, Mauro Salles. "Dicionário Houaiss da Língua Portuguesa". Insituto Antônio Houaiss de Lexicografia e Banco de Dados da Língua Português S/C Ltda. Rio de Janeiro. Ed. Objetiva, 2001.

ISO/IEC N4668, "MPEG-4 Overview - (V.21 – Jeju Version)", Editor: Rob Koenen, março de 2002. Disponível em: http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm, acesso em setembro de 2002.

ITU TELECOM. Standardization Sector of ITU, "Video coding for low bitrate communication," ITU-T Recommendation H.263, março de 1996.

ITU TELECOM. Standardization Sector of ITU, "Video coding for low bitrate communication," Draft ITU-T Recommendation H.263 Version 2, setembro de 1997.

JAKOBSSON, M., WETZEL, S, "Security Weaknesses in Bluetooth", RSA Conference 2001. São Francisco, CA, 2001. p. 176-191.

LEE, Edward A. "Overview of the Ptolemy Project", Technical Memorandum UCB/ERL M01/11, University of California, Berkeley, EUA. Março de 2001. Disponível em http://ptolemy.eecs.berkeley.edu. Acesso em: 23 jun. 2002.

LIEBL, Günther, STOCKHAMMER, Thomas, BURKERT, F. "Modelling and Simulation of Wireless Packet Erasure Channels". 10th VIRGINIA TECH/MRPG SYMPOSIUM ON WIRELESS PERSONAL COMMUNICATION, Blacksburg, VA,

USA, maio de 2000. p. 203-214.

LIU, H., ZARKI, M. E., “Adaptive source rate control for real-time wireless video transmission”, Mobile Networks and Applications, v. 3, p. 49-60, 1998.

METTÄLÄ, R. “Bluetooth Protocol Architecture, A Bluetooth SIG White Paper”, Revision 0.95., 1999. Disponível em http://www.bluetooth.org. Acesso em: 08 out. 2002.

MILLER, B. et. al. “Bluetooth Revealed”, 1. ed. Addison Wesley, 2001.

MOTTA, G., STORER, J. A., CARPENTIERI, B., “Improving Scene Cut Quality for Real-Time Video Decoding”, Proceedings of the Data Compression Conference (DCC 2000), Snowbird (UH), Março de 2000.

NGUYEN, G. T., KATZ, R. H., NOBLE, B. D., e SATYANARAYANAN, M. “A Trace-based Approach for Modeling Wireless Channel Behavior”. Proc. Winter Simulation Conference. Dezembro de 1996.

PEREIRA, Fernando, “Tutorial Issue on The MPEG-4 Standard”, Image Communication Journal, Elsevier, março de 2002. Disponível em: http://leonardo.telecomitalialab.com/icjfiles/mpeg-4_si/index.htm, acesso em setembro de 2002.

SANTOS, Ednalva Maria dos, SOUZA, Marlene Aparecida de, MACHADO, Rosa Helena Blanco, ALMEIDA, Rosiléia Oliveira. “O Texto Científico: Diretrizes para Elaboração e Apresentação”. Salvador – BA. UNYAHNA / QUARTETO, 2001. 80 p.

SHEN, Hong, WANG e MOAYERI, Nader. “Finite-State Markov Channel - A Useful Model for Radio Communications Channels”. IEEE Transactions on Vehicular Technology, v. 44, n.1. Fevereiro de 1995. p. 163-171.

XYDIS, Thomas G., Blake-Wilson, Simon. “Security Comparison: BluetoothTM Communications vs. 802.11”, fevereiro de 2002. Disponível em http://www.wilcoxonwireless.com/whitepapers/14Bluetooth_Wifi_Security.pdf. Acesso em: 10 dez. 2002.

ZURBES, Stefan. “Considerations on link and system throughput of Bluetooth networks”, Proceedings of the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2000, v. 2, p. 1315-1319.