

SAMIR BONHO

**SISTEMA MICROCONTROLADO PARA
TRANSMISSÃO DE SINAIS DE ECG PELA
INTERNET**

FLORIANÓPOLIS

2006

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**SISTEMA MICROCONTROLADO PARA
TRANSMISSÃO DE SINAIS DE ECG PELA
INTERNET**

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica.

SAMIR BONHO

Florianópolis, junho de 2006.

SISTEMA MICROCONTROLADO PARA TRANSMISSÃO DE SINAIS DE ECG PELA INTERNET

Samir Bonho

‘Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em Engenharia Biomédica, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’

Prof. Raimes Moraes, Ph.D
Orientador

Prof. Nelson Sadowski, Dr

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Raimes Moraes, Ph.D
Presidente

Prof. Fernando Mendes de Azevedo, D.Sc.

Prof. Renato Garcia Ojeda, Dr.

Prof (a). Mirela Sechi Moretti Annoni Notare, Dr (a).

Aos meus heróis, Sérgio e Clébes,
minhas eternas fontes de amor, coragem e paz.

AGRADECIMENTOS

Durante estes dois anos de mestrado, inúmeras pessoas foram importantes para que este trabalho fosse concluído.

Começo agradecendo ao amigo Paulo Pizarro, pelos esclarecimentos prestados no começo da dissertação.

Aos amigos de laboratório, Paim, Baggio, Daniel e Ricardo, por todas risadas, conselhos e idéias. A melhor e mais divertida equipe de trabalho que tenho conhecimento.

Aos meus colegas de apartamento, Wlad e Rafa, pela amizade e momentos de descontração.

A meus irmãos e irmã de coração, Mineiro, Mário, Xisto, James, Batata, Pedro, Telles e Ju, por estarem presentes nos melhores anos de minha vida.

Aos grandes amigos Cristiano e Locatelli, pelo companheirismo durante os momentos em que nos lançávamos no mar de idéias, deslizando sobre as ondas do conhecimento.

A Tamires, por me mostrar uma nova visão da vida. Seu olhar ilumina meu caminho e me torna uma pessoa melhor...

Às minhas vizinhas, Martha, Vic e Simone, pelas jantãs e sessões de cinema que proporcionavam ótimos momentos.

Ao amigo e professor Raimes, pela orientação, apoio, paciência e bom-humor.

À minha família, que, apesar da distância, sempre me incentivou.

Aos colegas de mestrado, Pedrão, Eduardo, Juliano, Dai, Wilson, Igor, Fabrício, Vivi, Raul, Fábio, Roselle e Márcia, pela amizade.

À CAPES, pelo apoio financeiro.

“Se as coisas são inatingíveis... ora!
Não é motivo para não querê-las...
Que tristes os caminhos se não fora
a mágica presença das estrelas!”

(Mário Quintana)

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

SISTEMA MICROCONTROLADO PARA MONITORAMENTO DE SINAIS DE ECG PELA INTERNET

SAMIR BONHO

Junho/2006

Orientador: Raimés Moraes, Ph.D.

Área de Concentração: Engenharia Biomédica

Palavras-chave: Eletrocardiograma, Telemedicina, Internet

Número de Páginas: 97

RESUMO: Este trabalho aborda o projeto e desenvolvimento de um sistema de telemedicina para eletrocardiograma (ECG), composto por um módulo de aquisição portátil, um módulo micro-controlado e um *software* executado em um computador remoto. O sistema desenvolvido amostra e envia os sinais de ECG do paciente pela Internet. No computador remoto, o sinal de ECG é apresentado em tempo-real na tela para análise de um especialista. O módulo de aquisição capta duas derivações de ECG por meio de eletrodos postados no tórax do paciente. Estas amostras são transmitidas ao módulo micro-controlado via RF (*Bluetooth*). Este módulo (baseado no microcontrolador eZ80190 - *Zilog*) recebe os sinais de ECG (via *Bluetooth*), encapsula-os dentro de pacotes RTP (*Real Time Protocol*) e os transmite pela Internet a um computador pessoal (PC). O *software*, escrito em linguagem C++ (sistema operacional *Windows*), executado no PC remoto recebe e mostra o sinal de ECG na tela. Para uma avaliação inicial do sistema, quatro diferentes topologias de aplicação foram simuladas. Parâmetros relacionados à qualidade de transmissão são apresentados: *jitter* e perdas.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

MICROPROCESSOR-BASED SYSTEM TO ECG MONITORING THROUGH INTERNET

SAMIR BONHO

June / 2006

Advisor: Raimes Moraes, Ph.D.

Area of Concentration: Biomedical Engineering.

Keywords: Electrocardiogram, Telemedicine, Internet

Number of Pages: 97

ABSTRACT: This work presents a system for remote monitoring of patients that comprises a portable device, a microprocessor-based Internet server and software running on a remote computer. The developed system samples and sends the patient's ECG waveforms over the Internet. At the remote computer, the ECG is presented in real time on the screen to be analyzed by a health professional. The portable device connected to the patient samples two ECG leads. The samples are sent to the microprocessor-based Internet server via RF (Bluetooth). Bluetooth was chosen because it operates on ISM band, presenting low power and low cost. The Internet server (based on the eZ80190 - Zilog) receives (via Bluetooth), encapsulates and transmits the ECG signals as well as controls the data flow. The ECG samples are packed into RTP (Real Time Protocol). The remote computer receives and unpacks the RTP packets to display the ECG on the screen. The software for Windows OS written in C++ calculates other four ECG leads. To assess the system performance, four different communication topologies were simulated. Parameters related to the quality of the transmission are presented: Jitter and loss rate.

SUMÁRIO

ÍNDICE DE FIGURAS	<i>xi</i>
ÍNDICE DE TABELAS	<i>xii</i>
1 INTRODUÇÃO	13
1.1 TELEMEDICINA	13
1.2 OBJETIVOS	14
1.3 JUSTIFICATIVA	15
1.4 ORGANIZAÇÃO DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 BOMBA CARDÍACA	17
2.1.1 Ciclo Cardíaco	18
2.2 ELETROCARDIOGRAMA	18
2.2.1 Características Elétricas do ECG	20
2.2.2 Sistema de Derivações	21
2.3 TRANSMISSÃO DE SINAIS DE ECG	23
2.4 TCP/IP	28
2.4.1 Modelo <i>Open Systems Interconnection</i> (OSI)	29
2.4.2 Modelo TCP/IP	32
2.4.3 IP	35
2.4.4 TCP	36
2.4.5 UDP	37
2.5 ESCOLHA DO PROTOCOLO DE TRANSPORTE	38
2.6 RTP	39
2.6.1 Cabeçalho RTP	40
2.7 RTCP	44
3 METODOLOGIA	46
3.1 MÓDULO DE AQUISIÇÃO DE SINAL DE ECG	46
3.2 MÓDULO DE COMUNICAÇÃO	48
3.1.1 Funcionamento	50
3.2 FIRMWARE	51
3.2.1 Recepção do Eletrocardiograma	54
3.2.2 Encapsulamento e transmissão	55
3.3 SOFTWARE PARA MICROCOMPUTADOR	59
3.3.1 Conexão com a Internet	61
3.3.2 Ordenação dos pacotes	61
3.3.3 Determinação da Frequência Cardíaca	62

3.3.4	Cálculo das Derivações	64
3.3.5	Visualização do sinal de ECG	66
3.3.6	Diagrama de Classes	68
4	RESULTADOS	70
4.1	TESTES	70
	Software de Avaliação	71
4.2	CENÁRIO 1	73
4.2.1	Resultados: Cenário 1	73
4.3	CENÁRIO 2	76
4.3.1	Resultados: Cenário 2	77
4.4	CENÁRIO 3	79
4.4.1	Resultados: Cenário 3	80
4.5	CENÁRIO 4	83
4.5.1	Resultados: Cenário 4	83
4.6	RESULTADO GERAL	86
5	DISCUSSÃO E CONCLUSÃO	88
5.1	DISCUSSÃO	88
5.2	CONCLUSÃO	90
5.3	TRABALHOS FUTUROS	91
	REFERÊNCIAS	92

ÍNDICE DE FIGURAS

Figura 2.1 - Representação do coração humano (adaptado de health.yahoo.com)	17
Figura 2.2 - Sinal de ECG obtido pela derivação II de Einthoven [36]	19
Figura 2.3 - Faixa de frequência da eletrocardiografia [12]	20
Figura 2.4 - Derivações bipolares de Einthoven.	21
Figura 2.5 - Derivações pré-cordiais.	22
Figura 2.6 - Derivações aumentadas.	23
Figura 2.7 - Camadas do Modelo de referência OSI [30]	29
Figura 2.8 - Relação das camadas TCP/IP com as do modelo OSI [16]	33
Figura 2.9 - Cabeçalho UDP	37
Figura 2.10 - Pseudo-Cabeçalho do protocolo UDP	37
Figura 2.11 - Cabeçalho RTP	40
Figura 3.1 - Sistema de monitoramento proposto	46
Figura 3.2 - Módulo de aquisição do sinal de ECG	47
Figura 3.3 - Diagrama em blocos das etapas de processamento realizado pelo módulo de aquisição.	47
Figura 3.4 - Placa de desenvolvimento eZ80190	49
Figura 3.5 - KC11 da KC Wirefree	49
Figura 3.6 - Diagrama de Blocos do Módulo de Comunicação	50
Figura 3.7 - Módulo de Comunicação.	51
Figura 3.8 - Fluxograma do firmware desenvolvido	53
Figura 3.9 - Cabeçalhos dos protocolos de um pacote de dados RTP	55
Figura 3.10 - Cabeçalho RTP para a transmissão do eletrocardiograma	58
Figura 3.11 - Fluxograma do Software	60
Figura 3.12 - Pacotes fora de ordem causados por atraso na rede	62
Figura 3.13 - Aquisição das derivações aumentadas aVL (a), aVR (b), e aVF (c).	65
Figura 3.14 - Tela do software executado em sistema operacional Windows.	67
Figura 3.15 - Diagrama de classes em UML do software Core 1.0	68
Figura 4.1 - Tela do software de avaliação do sistema.	72
Figura 4.2 - Cenário 1: Rede Local	73
Figura 4.3 - Tela do software de avaliação do sistema: Cenário 1 - Teste 1	74
Figura 4.4 - Tela do software de avaliação do sistema: Cenário 1 com tráfego - Teste 8	75
Figura 4.5 - Interconexão das redes do IEB e GRUCAD através da rede UFSC.	76
Figura 4.6 - Tela do software de avaliação do sistema: Cenário 2 sem tráfego - Teste 2	78
Figura 4.7 - Tela do software de avaliação do sistema: Cenário 2 com tráfego - Teste 6	79
Figura 4.8- Comunicação pela Internet.	80
Figura 4.9 - Tela do software de avaliação do sistema: Cenário 3 sem tráfego - Teste 5	81
Figura 4.10 - Tela do software de avaliação do sistema: Cenário 3 com tráfego - Teste 7	82
Figura 4.11 - Comunicação pela Internet entre São Paulo e Florianópolis	83
Figura 4.12 - Tela do software de avaliação do sistema: Cenário 4 sem tráfego - Teste 2	84
Figura 4.13 - Tela do software de avaliação do sistema: Cenário 4 com tráfego - Teste 9	85

ÍNDICE DE TABELAS

<i>Tabela 2.1 - Vazão típica de aplicações de rede. [15]</i>	25
<i>Tabela 3.1 - Comparativo entre datagramas de diferentes tamanhos</i>	56
<i>Tabela 4.1 – Cenário 1: Sem tráfego adicional</i>	74
<i>Tabela 4.2 – Cenário 1: Tráfego adicional gerado</i>	75
<i>Tabela 4.3 – Cenário 2: Sem tráfego adicional</i>	77
<i>Tabela 4.4 – Cenário 2: Tráfego adicional gerado</i>	78
<i>Tabela 4.5 – Cenário 3: Sem fluxo adicional</i>	81
<i>Tabela 4.6 – Cenário 3: Fluxo adicional gerado</i>	82
<i>Tabela 4.7– Cenário 4: Sem fluxo adicional</i>	84
<i>Tabela 4.8 – Cenário 4: Fluxo adicional gerado</i>	85
<i>Tabela 4.9 - Largura de banda dos cenários</i>	86
<i>Tabela 4.10 – Média do jitter e taxa de perdas máxima</i>	87

1 INTRODUÇÃO

1.1 TELEMEDICINA

A telemedicina¹ tem evoluído nos últimos anos e despertado o interesse de muitas organizações ligadas à área da saúde [1].

O aumento da demanda por serviços de assistência domiciliar (*homecare*) justifica-se frente às vantagens que a assistência médica doméstica traz para o sistema de saúde. Pode-se citar, dentre as principais vantagens, a melhora da qualidade de vida e a humanização do atendimento aos pacientes, a redução do risco de infecção hospitalar e a redução substancial nos custos relacionados ao atendimento. Um outro importante benefício trazido pelo atendimento *homecare* é a possibilidade de ampliação do número de vagas nos hospitais, já que a internação para investigação diagnóstica ou tratamento de doenças crônicas torna-se assim, desnecessária. Estas vagas podem ser reservadas para casos de alta complexidade, otimizando o sistema de saúde [2].

Um dos motivos do crescimento da telemedicina é o avanço nas tecnologias de transmissão de dados. A confiabilidade nas comunicações impulsionou o desenvolvimento de ferramentas que auxiliam ou complementam os serviços de *homecare*.

Os protocolos de comunicação presentes na Internet fornecem a base necessária para que a transmissão em tempo real seja satisfatória. Aspectos que comprometiam este tipo de comunicação, como perdas e atrasos nos dados, podem ser identificados e corrigidos à medida do possível para que a aplicação final seja satisfatória.

¹ Segundo a *American Telemedicine Association* (ATA), a Telemedicina é "o uso de informação médica veiculada de um local para outro, por meio de comunicação eletrônica, visando à melhora da saúde do paciente."

Um sistema de telemetria médica deve fornecer ao médico as facilidades necessárias para que este possa analisar o sinal sem receio sobre a qualidade dos dados recebidos.

O monitoramento em tempo real de eletrocardiograma (ECG) utilizando a Internet como meio de comunicação é bem difundido no meio científico [3] [4] [5] [6]. Existem diversas arquiteturas para este tipo de monitoração, baseadas na aquisição do ECG, processamento e transmissão cliente-servidor [7]. A maioria dos sistemas *homecare* para monitoração de sinais de ECG apresentados até hoje utiliza um PC dedicado para a transmissão dos dados. Isto resulta numa solução onerosa para o monitoramento remoto, o que dificulta a difusão deste tipo de aplicação/tecnologia em países em desenvolvimento.

1.2 OBJETIVOS

O objetivo deste trabalho é implementar sistema para transmissão de sinais de eletrocardiograma pela Internet em tempo real.

Os objetivos específicos do trabalho são:

- Implementar *firmware* para gerenciar a execução das seguintes atividades:
 - desenvolver um algoritmo para a recepção do sinal enviado pelo módulo de aquisição;
 - desenvolver um algoritmo para o encapsulamento e transmissão do ECG, através de uma rede IP, de acordo com a estrutura especificada no protocolo RTP;
 - adaptar a estrutura definida no protocolo RTP (*Real Time Protocol*) para a sua utilização na transmissão do sinal de eletrocardiograma;

- desenvolver um algoritmo para o controle no envio dos pacotes RTP de acordo com a estrutura especificada no protocolo RTCP.
- Desenvolver *software* para microcomputador PC para executar as seguintes tarefas:
 - desenvolver um algoritmo para a recepção e visualização dos sinais transmitidos pelo módulo de comunicação;
 - desenvolver um algoritmo para minimização do efeito *jitter* dos pacotes RTP recebidos; e
 - desenvolver um algoritmo para avaliar a qualidade de transmissão dos dados através da rede, utilizando para tal, dados fornecidos pelos pacotes RTP.

1.3 JUSTIFICATIVA

O desenvolvimento deste sistema complementa o sistema de monitoração remota de sinal de eletrocardiograma em tempo real desenvolvido no laboratório [8] [9].

No sistema desenvolvido anteriormente, o eletrocardiograma do paciente é captado por módulo micro-controlado e enviado por rádio frequência a um computador que por sua vez o transmite a um computador remoto.

Para possibilitar a disseminação da monitoração remota, o presente trabalho apresenta módulo micro-controlado que em conjunto com o módulo de aquisição pode ser repassado ao paciente, reduzindo os custos totais do sistema.

1.4 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está dividido em cinco capítulos. A fundamentação teórica encontra-se no Capítulo 2. Este Capítulo 2 descreve brevemente o

funcionamento mecânico e elétrico do coração, descrevendo também as características básicas do ECG. Há ainda uma discussão sobre os recursos de rede solicitados na transmissão de sinais de ECG. O Capítulo 2 segue com uma descrição da família de protocolos TCP/IP, com foco voltado para a camada de transporte e seus principais protocolos de comunicação.

No Capítulo 3, o sistema de telemetria proposto é descrito, sendo subdividido em três partes: *hardware*, que compreende os módulos de aquisição e comunicação; o *firmware* e *software* para o computador remoto.

O Capítulo 4 descreve os testes realizados e os resultados obtidos na avaliação do sistema.

No Capítulo 5, realiza-se a discussão sobre o sistema de telemetria desenvolvido, sendo apresentadas a conclusão e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo 2 apresenta conceitos sobre a fisiologia do coração e seu comportamento elétrico. Aborda também parâmetros sobre a qualidade de transmissão de um sistema de monitoração em tempo-real. Realiza ainda uma descrição geral dos protocolos empregados na transmissão de dados pela Internet.

2.1 BOMBA CARDÍACA

O coração é constituído por duas bombas distintas: a direita (que bombeia sangue para os pulmões) e a esquerda (que bombeia sangue para os órgãos periféricos). Cada uma destas bombas possui duas câmaras: átrio e ventrículo. O átrio funciona como uma bomba de escorva, auxiliando o preenchimento do ventrículo com sangue. O ventrículo, por sua vez, fornece a força principal para propeler o sangue para a circulação pulmonar, pelo ventrículo direito, ou para a circulação periférica, pelo ventrículo esquerdo. A Figura 2.1 ilustra a anatomia do coração humano.

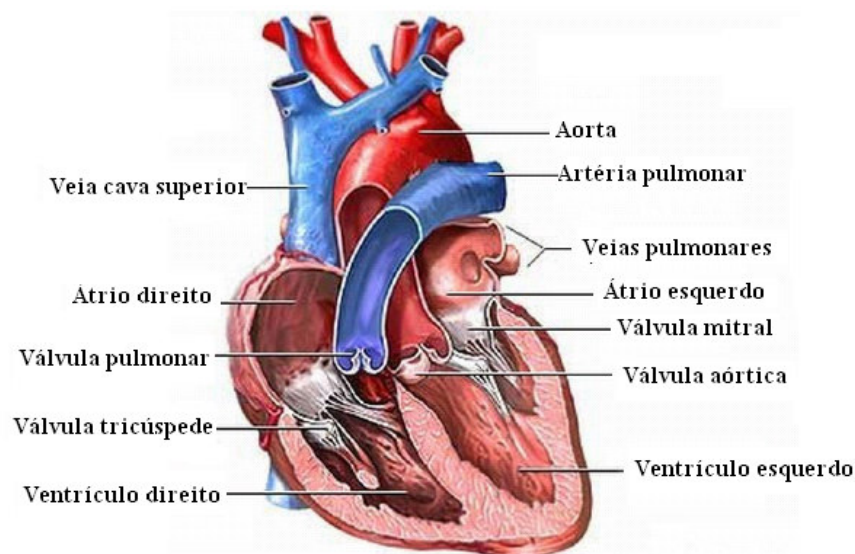


Figura 2.1 - Representação do coração humano (adaptado de health.yahoo.com)

O funcionamento do coração como bomba cardíaca dá origem ao ciclo cardíaco, onde existe o sincronismo entre o relaxamento e a contração do músculo cardíaco.

2.1.1 Ciclo Cardíaco

O ciclo cardíaco consiste em um período de relaxamento, chamado de diástole, durante o qual o coração se enche com sangue, seguido de um período de contração chamado de sístole. Cada ciclo inicia-se pela geração espontânea de um potencial de ação no nodo sinusal.

O potencial de ação (PA) propaga-se rapidamente pelos átrios, mas é retardado por mais de 0,1s na região do nodo atrioventricular (nodo A-V) quando de sua passagem dos átrios para os ventrículos. Isso permite que os átrios contraíam-se antes dos ventrículos, ejetando sangue para os ventrículos antes que estes comecem a se contrair. Ao atingir o feixe de His, o PA espalha-se rapidamente por meio das fibras de Purkinje para o endocárdio. O PA passa, então, a se propagar lentamente pelo músculo ventricular em direção ao epicárdio.

A ritmicidade e condução do PA são moduladas pelos nervos simpáticos e parassimpáticos presentes no coração [10].

2.2 ELETROCARDIOGRAMA

Durante a estimulação do coração, a distribuição de cargas elétricas pode ser modelado por um dipolo. A colocação de um par de eletrodos sobre o corpo permite o registro da magnitude da projeção do dipolo elétrico na direção dada pelos eletrodos. Este registro é denominado de eletrocardiograma [8].

O eletrocardiograma considerado normal é formado pela onda P, pelo complexo QRS e pelas ondas T e U. A onda P é causada pelos potenciais elétricos gerados quando os átrios se despolarizam, antes da contração atrial. O complexo

QRS é causado pelos potenciais gerados à medida que os ventrículos se despolarizam. Assim, o complexo QRS inicia-se pouco antes da sístole ventricular. A onda T é causada pelos potenciais gerados quando os ventrículos voltam a se polarizar, ou seja, no momento em que a musculatura ventricular começa a relaxar. Esse processo ocorre, normalmente, de 0,25 a 0,35s após a despolarização. A onda T é conhecida como onda de repolarização da musculatura ventricular.

A onda U é uma pequena deflexão de baixa frequência que se inscreve após a onda T. A amplitude desta onda normal é geralmente proporcional à da onda T e corresponde de 5% a 25% do total desta. A gênese da onda U é ainda objeto de discussão. Atualmente, a última hipótese apresentada baseia-se na relação temporal entre traçados eletrocardiográficos e registros de potenciais de ação de células miocárdicas, com foco nas células M. [11]

Dessa forma, o eletrocardiograma é composto por ondas de despolarização e de repolarização. A Figura 2.2 ilustra um sinal normal de ECG.

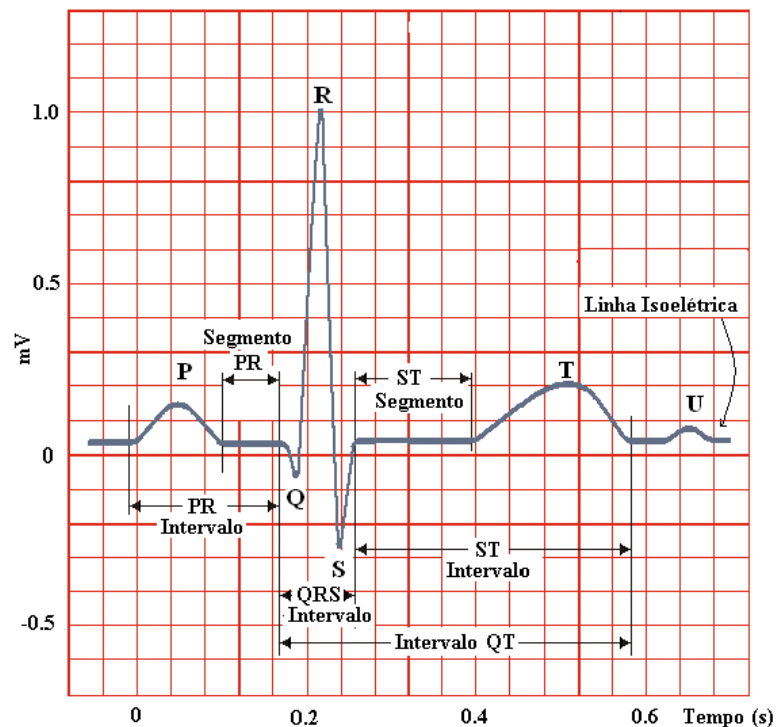


Figura 2.2 - Sinal de ECG obtido pela derivação II de Einthoven [36]

As características do ECG, ou seja, o formato do traçado do eletrocardiograma e os valores normais das tensões registradas, dependem da maneira com a qual os eletrodos são aplicados à superfície do corpo, e quão perto se encontram do coração.

2.2.1 Características Elétricas do ECG

O registro do ECG depende da posição dos eletrodos na superfície do corpo (Seção 2.2.2). Em registros não invasivos, as tensões do complexo QRS são da ordem de 1 mV entre a linha de base e o pico da onda R. A onda P possui tensões entre 0,1 e 0,3 mV; a onda T, tensões entre 0,2 e 0,3 mV [12].

A banda de frequência utilizada na aquisição do sinal depende do tipo de aplicação desejada. A eletrocardiografia clínica demanda uma faixa de frequência entre 0,05 e 100 Hz, já a monitoração utiliza banda entre 0,5 e 50 Hz. Para determinação da frequência cardíaca, a faixa de frequência utilizada é centrada em 17 Hz. [12]

A banda de frequência das três aplicações citadas é observada na Figura 2.3.

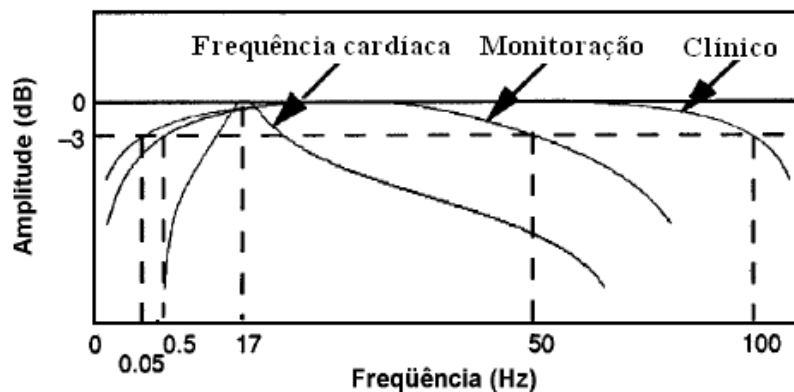


Figura 2.3 - Faixa de frequência da eletrocardiografia [12]

A restrição da banda da eletrocardiografia de monitoração contribui para a atenuação de ruídos de baixa frequência causados pela movimentação dos

eletrodos, e de maior frequência devidos às contrações musculares e interferências da rede de energia.

2.2.2 Sistema de Derivações

O eletrocardiograma de repouso é formado por 12 derivações: 6 do plano frontal do corpo humano (bipolares e unipolares aumentadas) e 6 do plano transversal (pré-cordiais).

Einthoven propôs o primeiro sistema de derivações. Este sistema baseia-se na utilização de três eixos de projeção no plano frontal do corpo humano. Os eixos formam um triângulo equilátero conhecido como o triângulo de Einthoven, sendo os registros sobre estes denominados de derivações bipolares. Conforme a Figura 2.4, a derivação I é registrada por eletrodos colocados no braço direito e braço esquerdo; a derivação II, por eletrodos sobre braço direito e perna esquerda e a derivação III, por eletrodos no braço esquerdo e na perna esquerda [10].

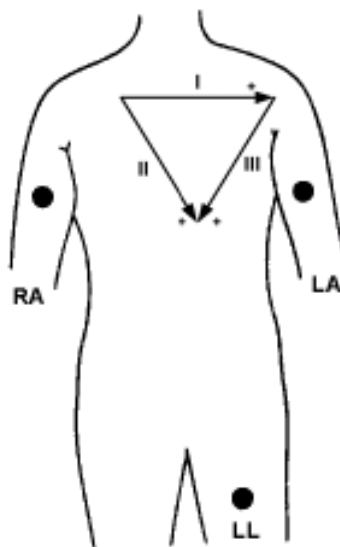


Figura 2.4 - Derivações bipolares de Einthoven.

As posições dos eletrodos são definidas pelos pontos RA (abreviatura de Righth Arm, em português, braço direito), LA (abreviatura de Left Arm, em português, braço esquerdo) e LL (abreviatura de Left Leg, em português, perna esquerda).

As derivações pré-cordiais registram a projeção do dipolo cardíaco sobre o plano transversal do corpo humano. As derivações pré-cordiais são seis: **V1**, **V2**, **V3**, **V4**, **V5**, **V6**. A Figura 2.5 ilustra estas derivações. [12]

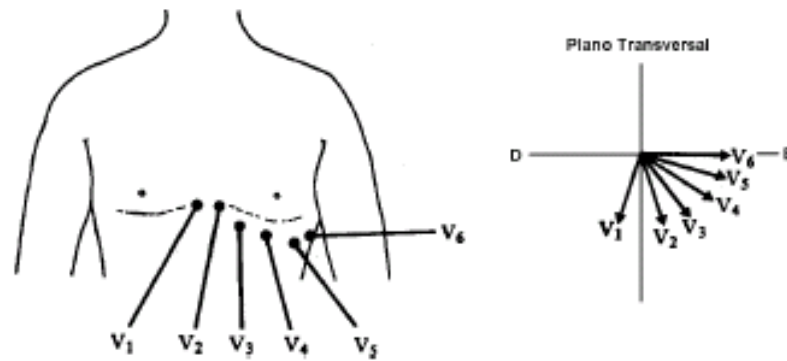


Figura 2.5 - Derivações pré-cordiais.

Essas derivações captam projeções transversais do vetor cardíaco.

As derivações unipolares aumentadas (aVL, aVR e aVF), cujas direções sobre os quais o registro do ECG são mostradas na Figura 2.6, proporcionam formas de onda com amplitudes 50% maiores que as obtidas nas derivações unipolares comuns (VR, VL e VF), as quais são obtidas com referência à média das tensões presentes em todos os eletrodos. [12]

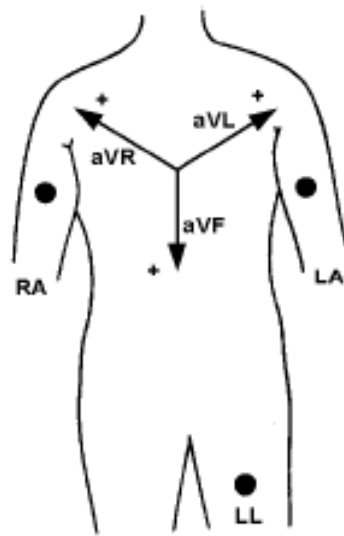


Figura 2.6 - Derivações aumentadas.

O potencial de referência de cada derivação aumentada corresponde à média das tensões presentes somente nos eletrodos opostos ao sentido do vetor correspondente.

2.3 TRANSMISSÃO DE SINAIS DE ECG

Após serem devidamente captados, os sinais de eletrocardiograma, já em formato digital, devem ser encapsulados e transmitidos. Na monitoração em tempo-real, o sinal deve ser enviado o mais rapidamente possível, em intervalos de tempo regulares, para que o sistema de recepção possa apresentar e processar o sinal de ECG. Na aplicação final, o médico deve emitir um diagnóstico sobre o paciente sem que haja dúvidas sobre a qualidade do sinal interpretado.

A transmissão de dados através de uma rede IP apresenta uma série de problemas que podem vir a prejudicar o monitoramento remoto de sinais biológicos.

Atualmente, a transmissão dos dados na Internet enquadra-se dentro de uma prática de melhor esforço (*Best Effort*), sendo que não há garantias de que os pacotes enviados na rede chegarão ao seu destino. Quando se trata de uma

aplicação crítica, como por exemplo, a telemedicina, requer-se que recursos sejam introduzidos para que estas aplicações sejam satisfatórias.

Um destes recursos é a chamada Qualidade de Serviço (*Quality of Service - QoS*)². Num primeiro momento, o termo "Qualidade de Serviço" pode ser entendido como um conjunto de parâmetros (atrasos, vazão, perdas, etc) que devem se encontrar dentro de limites especificados (valor mínimo e valor máximo).

Existem, essencialmente, duas maneiras de oferecer garantias QoS. A primeira procura disponibilizar recursos suficientes para a máxima taxa de transmissão esperada, com uma margem de segurança substancial. O segundo método é negociar reservas de recursos e aceitá-las com a condição de que os roteadores da rede possam garantir os mesmos [13].

Os parâmetros de qualidade de serviço são definidos pela aplicação através de uma "Solicitação de Serviço". A solicitação de QoS da aplicação é denominada tipicamente de SLA (*Service Level Agreement*).

O monitoramento de ECG em tempo real abrange, basicamente, os seguintes parâmetros de QoS [14] :

- i. vazão (Banda);
- ii. atraso (Latência);
- iii. *jitter* (Variação do Atraso);
- iv. taxa de perdas, taxa de erros; e
- v. disponibilidade.

² Na recomendação E.800 do *Internacional Telecommunication Union (ITU)* [E800 94], a QoS é definida como o efeito conjunto do desempenho do serviço que determina o grau de satisfação de um usuário desse serviço.

i. Vazão.

A vazão ou banda é o parâmetro mais básico de QoS. A rede deve ser capaz de atender a vazão gerada pela aplicação.

No caso da transmissão de eletrocardiograma, a banda gerada é dependente de dois fatores: a taxa de amostragem do sinal e a resolução do conversor analógico/digital (A/D). Uma frequência de amostragem de 250 Hz e resolução de 8 bits do conversor A/D são suficientes para que o sinal de ECG possua qualidade para monitoração [12]. Assim, são transmitidas 250 amostras de 8 bits de cada derivação do ECG.

A aplicação necessita uma vazão de 2000 bps (*bits* por segundo) ou 2 kbps (kilo *bits* por segundo) por canal. No sistema proposto, 2 derivações de ECG são monitoradas, o que solicitará uma vazão mínima de 4 kbps.

Esta banda requerida leva em conta apenas os dados, excluindo o tamanho dos cabeçalhos dos protocolos de transmissão.

Na Tabela 2.1, apresentam-se as vazões típicas de alguns aplicativos que utilizam redes para a comunicação.

Tabela 2.1 - Vazão típica de aplicações de rede. [15]

Aplicação	Vazão
Sinais Vitais (Bioelétricos)	4 kbps a 160 kbps
Voz	10 kbps a 120 kbps
Aplicações <i>Web</i> (WWW)	10 kbps a 500 kbps
Transferência de Arquivos (Grandes)	10 kbps a 1 Mbps
Vídeo (<i>Streaming</i>)	100 kbps a 1 Mbps
Aplicação Conferência	500 kbps a 1 Mbps
Vídeo MPEG	1 Mbps a 10 Mbps
Aplicação Imagens Médicas	10 Mbps a 100 Mbps
Aplicação Realidade Virtual	80 Mbps a 150 Mbps

As menores taxas de vazão, como mostrado na tabela acima, são geradas por aplicativos que realizam a transmissão de sinais bioelétricos, escopo deste trabalho.

ii. Latência

A latência da rede pode ser definida como a somatória dos atrasos impostos pela rede e equipamentos utilizados na comunicação.

Basicamente, a latência de uma rede é imposta por dois tipos de atrasos: fixo e variável. O atraso fixo depende da velocidade de propagação do sinal elétrico no meio físico e dos *handoffs* entre os roteadores da rede. O atraso variável é influenciado pelo tráfego e congestionamento da rede, causado pelo enfileiramento de pacotes nos roteadores.

Na monitoração de ECG, o atraso que um pacote leva na transmissão do sinal do paciente até o computador do médico deve ser o menor possível, para que as características do ECG relevantes para o diagnóstico possam ser observadas com o menor atraso possível.

iii. Jitter

Quando uma aplicação envia pacotes pela rede, esta não garante uma entrega seqüencial dos pacotes ao destino. Estes pacotes chegam de forma desordenada, ou seja, existem atrasos na chegada de cada pacote.

O *jitter* pode ser entendido como a variação no tempo e na seqüência de entrega das informações devido à variação na latência (atrasos) da rede. Em um momento de tráfego intenso na Internet, essas variações podem ser grandes e comprometer o funcionamento da aplicação.

Idealmente, o eletrocardiograma deve ser apresentado de forma contínua em um centro de monitoração, sem que haja interrupção do sinal. Para que isto ocorra, deve ser aplicada uma técnica de buferização para minimizar o *jitter*

causado pela rede. Essa técnica consiste em introduzir um pequeno atraso na apresentação dos dados, colocando os pacotes num *buffer* antes de processá-los. Os pacotes recebidos fora de ordem são seqüenciados dentro deste “*buffer* de atraso”. Após esta ordenação, as informações contidas nos pacotes são apresentadas na tela do computador. Este processo tem o intuito de que, ao final da visualização dos dados, exista uma grande probabilidade de que os próximos pacotes, devidamente ordenados, já estejam disponíveis.

iv. Perdas

Na transmissão de dados através da Internet, pacotes podem ser corrompidos ou se perderem. Geralmente essas perdas ocorrem em função do descarte de pacotes nos roteadores e a erros ocorridos na camada de enlace de dados (Seção 2.4.1). Cabe a aplicação determinar um limite na taxa de erros que garanta uma operação adequada.

No monitoramento de sinais vitais, a quantidade de informação contida em cada pacote determina a vulnerabilidade da aplicação em relação a possíveis perdas. Um pacote com um pequeno segmento do sinal de eletrocardiograma, se perdido, não prejudica o monitoramento do paciente. Caso houvesse uma grande quantidade de informação no pacote, o trecho do sinal de ECG perdido seria significativa a ponto de comprometer a visualização do mesmo.

v. Disponibilidade

A disponibilidade é uma medida da garantia de execução da aplicação ao longo do tempo.

A monitoração remota de ECG é um tipo de aplicação que necessita uma disponibilidade maior que 99,9% [14]. Isto significa que a conexão contratada para a transmissão do ECG não pode ficar mais de 0,1% (quase nove horas num ano) fora do ar, ou sem serviço, sob pena de multa ou outro tipo de ressarcimento.

2.4 TCP/IP

Nos anos 60, o Departamento de Defesa dos Estados Unidos (DoD) interessou-se por um protocolo que estava sendo desenvolvido e utilizado pelas universidades para interligar os seus sistemas computacionais. O DoD tinha interesse em estabelecer a comunicação entre os seus núcleos de processamento espalhados pelo mundo. O maior problema residia na incompatibilidade entre os computadores de diferentes fabricantes que executavam diferentes sistemas operacionais, com distintos protocolos e topologias. O compartilhamento dos dados passou a constituir um problema de difícil solução.

A *Advanced Research Projects Agency* (ARPA) assumiu a tarefa de buscar um protocolo para que equipamentos com diferentes características computacionais comunicassem-se entre si. Realizou-se então, um consórcio entre universidades e fabricantes para o desenvolvimento de padrões de comunicação. Este consórcio especificou e construiu uma rede para testes composta por quatro máquinas, denominada ARPANET. Esta rede originou a Internet.

No final dos anos 70, esta rede inicial evoluiu. Seu protocolo de comunicação foi aperfeiçoado, transformando-se na base para o TCP/IP (*Transmission Control Protocol / Internet Protocol*). A aceitação mundial do conjunto de protocolos TCP/IP deveu-se, principalmente, à versão do sistema operacional UNIX de Berkeley³ que incorporou estes protocolos e os disponibilizou ao domínio público. Assim, equipes técnicas de quaisquer organizações poderiam modificá-los e assim, garantir sua evolução.

Dentre as várias organizações e comitês que participam deste desenvolvimento e divulgação, pode-se destacar o *Internet Engineering Task Force* (IETF). Sua principal função é a manutenção e apoio aos padrões da Internet e TCP/IP, principalmente através da série de documentos *Request for Comments* (RFC). O RFC, além de especificar formalmente os protocolos, inclui informações importantes sobre seu funcionamento e uso. Estes documentos descrevem as

³ Versão do sistema operacional UNIX desenvolvido pela Universidade de Berkeley, Califórnia, EUA.

diversas tecnologias envolvidas e servem de base para as novas tecnologias que deverão manter a compatibilidade com as anteriores, dentro do possível.

Em resumo, o maior trunfo do TCP/IP é o fato destes protocolos apresentarem a interoperabilidade de comunicação entre todos os tipos de *hardware* e todos os tipos de sistemas operacionais.

Para uma melhor compreensão da estrutura do TCP/IP, faz-se necessário tomar conhecimento do modelo *Open Systems Interconnection* (OSI) que se constitui em referência para a interconexão entre computadores. Assim, modelo OSI é apresentado e, na seqüência, os protocolos TCP/IP são discutidos.

2.4.1 Modelo *Open Systems Interconnection* (OSI)

Para facilitar o processo de padronização e obter interconectividade entre máquinas de diferentes fabricantes, a *International Standards Organization* (ISO) criou um modelo de referência para permitir a comunicação entre diferentes máquinas, denominado *Open Systems Interconnection* (OSI).

O modelo conceitual OSI possui sete camadas, mostradas na Figura 2.7, que possuem distintas funções no processo de comunicação entre dois sistemas. Cada camada é independente em relação às demais e executa somente suas funções, sem se preocupar com as demais tarefas em execução nas outras camadas.

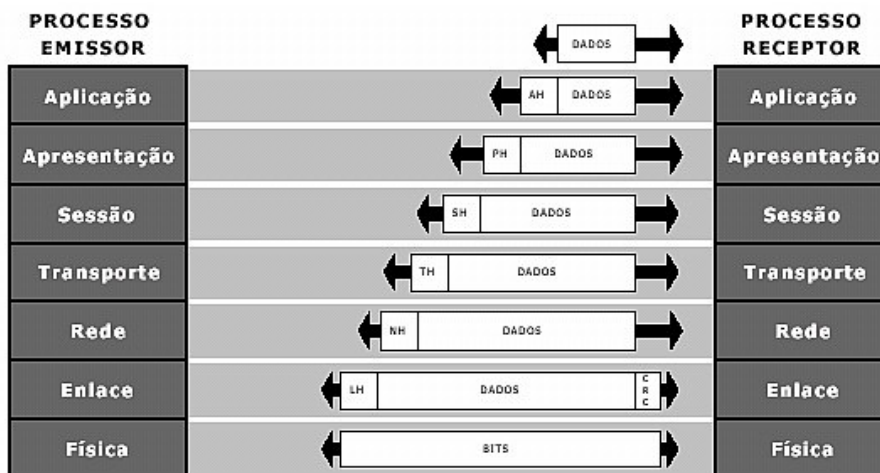


Figura 2.7 - Camadas do Modelo de referência OSI [30]

As camadas definidas no modelo OSI são descritas a seguir.

a) Camada Física

O nível físico tem a função de transmitir uma seqüência de *bits* através de um canal de comunicação. Os diferentes padrões estabelecidos neste nível objetivam que o nível lógico de um *bit* enviado por uma estação seja corretamente transmitido até o receptor. Este nível aborda basicamente características mecânicas e elétricas, modo de funcionamento e suporte de comunicação da interface física. Algumas questões definidas nesta camada são exemplificadas abaixo:

- modos de transmissão e conexão adotados;
- nível de tensão para a representação dos *bits* 0 e 1;
- duração do sinal de tensão; e
- tipos de conectores e pinagem.

b) Camada de Enlace de Dados

O principal objetivo do nível de enlace é receber/transmitir uma seqüência de *bits* do/para o nível físico e transformá-los em uma mensagem que esteja livre de erros de transmissão, a fim de que essa informação seja utilizada pela camada de Rede.

Perdas de dados, retransmissão e controle de fluxo devem ser gerenciados pela camada de enlace.

c) Camada de Rede

A Camada de Rede tem a função de executar o roteamento dos pacotes entre fonte e destino quando estes fazem parte de redes distintas. Os caminhos a

serem utilizados devem ser avaliados quanto ao tráfego de pacotes na rede para que congestionamentos sejam evitados.

d) Camada de Transporte

As funções básicas do nível de transporte compreendem o aceite de dados da camada de sessão, a partição destes em unidades menores (se necessário), o seu envio para a camada de rede e, dependendo da classe de serviço solicitada, assegurar que todos os dados chegarão corretamente ao outro extremo. Os processos de estabelecer, manter e encerrar as conexões entre dois computadores são realizados por esta camada.

Na camada de transporte, mecanismos de controle de fluxo e de erro encontram-se implementados para realizar, de forma confiável, a transmissão de dados.

Os protocolos UDP e TCP (Seções 2.4.4 e 2.4.5) da família TCP/IP são exemplos de protocolos da camada de transporte.

e) Camada de Sessão

Este nível permite que duas aplicações em computadores diferentes estabeleçam uma sessão de comunicação. É na sessão que se define, por exemplo, como será feita a transmissão dos dados.

O gerenciamento da comunicação é responsabilidade desta camada. Para tal, este nível executa, dentro de cada sessão, mecanismos para o intercâmbio de dados, controle de conversação e sincronização entre os computadores.

f) Camada de Apresentação

Esta camada tem como função assegurar que a informação transmitida possa ser devidamente interpretada e utilizada pelo receptor. O nível de

apresentação tem funções associadas à sintaxe e à semântica dos dados transmitidos. Por exemplo, um aplicativo pode apresentar uma mensagem em ASCII⁴ mesmo que a estação transmissora utilize o EBCDIC⁵ na codificação da mensagem.

O nível de apresentação é também responsável por aspectos como criptografia e compressão de dados.

g) Camada de Aplicação

A camada de aplicação faz a interface entre o protocolo de comunicação e o aplicativo que solicita ou recebe a informação através da rede.

Um aplicativo de *e-mail*, por exemplo, ao requerer a recepção dos *e-mails*, entra em contato com esta camada e efetua a solicitação. O nível de aplicação compreende o maior número de protocolos, pois tem de disponibilizar, a vários usuários, os diferentes recursos de que necessitam.

O *File Transfer Protocol* (FTP) e o *Simple Mail Transfer Protocol* (SMTP) são exemplos de protocolos que fazem parte desta camada.

2.4.2 Modelo TCP/IP

A Internet não pode ser vista como uma única rede, pois seus muitos integrantes podem ter topologias, larguras de banda, retardos, tamanhos de pacote e outros parâmetros completamente distintos. Contudo, dados enviados por um computador devem chegar ao destinatário, mesmo que estes estejam em redes com diferentes características.

Muitas vezes, dois computadores podem fazer parte de redes com as mesmas especificações, porém executam diferentes sistemas operacionais (SO).

⁴ O ASCII (*American Standard Code for Information Interchange*) é um conjunto de códigos de oito *bits* para representação de letras, números e pontuação em um computador.

⁵ O EBCDIC (*Extended Binary Coded Decimal Interchange Code*) é um código de oito *bits*, desenvolvido pela IBM, usado na codificação dos caracteres alfa-numéricos de computadores.

Mesmo assim, o dado deve ser reconhecido no computador de destino, independentemente do fato deste possuir, ou não, o mesmo SO da máquina remetente.

O protocolo TCP/IP apresenta-se como solução para a comunicação dentro da Internet. O TCP/IP fornece um serviço confiável, simples e eficiente para a transmissão de dados dentro das diferentes arquiteturas de rede e parâmetros que compõem a Internet. Uma importante característica destes protocolos é que estes independem de equipamentos e sistemas operacionais, podendo ser utilizados em redes com computadores não padronizados, fato que viabiliza a Internet.

É importante salientar que o termo “protocolo TCP/IP” é utilizado como designação comum para uma família de protocolos de comunicação de dados, sendo que o *Transmission Control Protocol* (TCP) e o *Internet Protocol* (IP) são apenas dois deles.

A Figura 2.8 relaciona as 4 camadas do modelo TCP/IP com as camadas já discutidas do modelo OSI.

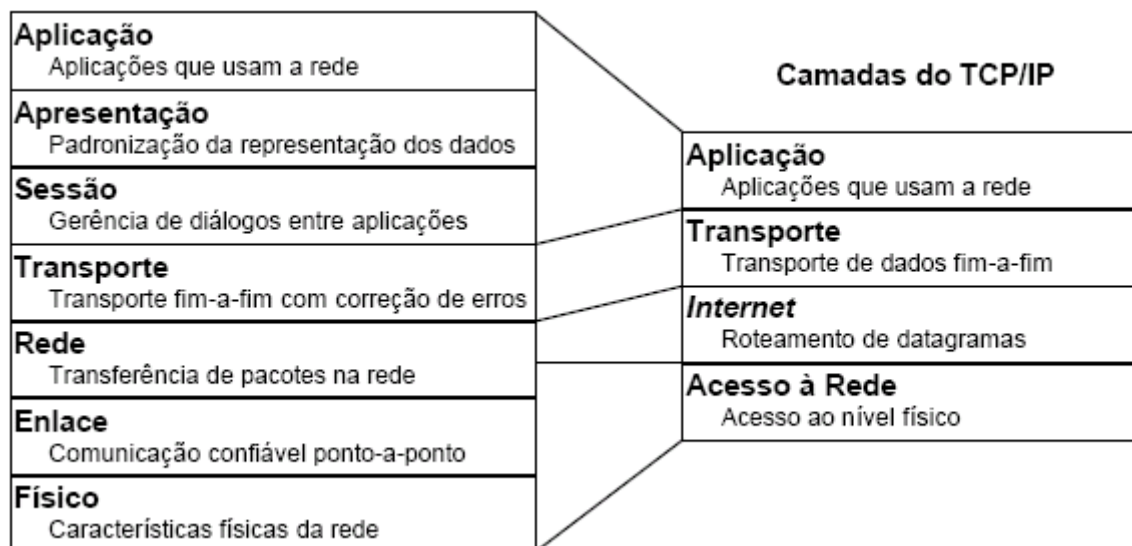


Figura 2.8 - Relação das camadas TCP/IP com as do modelo OSI [16]

As quatro camadas da arquitetura do TCP/IP são: acesso à rede, Internet ou inter rede, Transporte e Aplicação [16]. Cada uma executa um conjunto bem

definido de funções de comunicação. No modelo em camadas TCP/IP, não existe uma estrutura formal para cada camada, conforme ocorre no modelo OSI. A arquitetura TCP/IP estabelece um protocolo próprio para cada camada, assim como a interface de comunicação entre duas camadas adjacentes.

a) Camada de acesso à rede

Esta camada não possui um padrão para o acesso ao nível físico da sub-rede, possibilitando a conexão de qualquer tipo de rede, desde que haja uma interface que compatibilize a tecnologia da rede com o protocolo IP. Desta forma, um número muito grande de tecnologias pode ser utilizado na sub-rede de acesso, como *Ethernet*, *Token Ring*, *FDDI*, *X.25*, *Frame Relay*, *ATM* e outros.

Para que todas estas tecnologias sejam compatibilizadas com o TCP/IP, existe a necessidade de uma conversão de endereçamentos do formato utilizado pela sub-rede e o formato IP. Esta conversão é realizada pelos denominados *gateways* que tornam a interconexão das redes transparente para o usuário.

b) Camada Internet ou de inter rede

A tarefa desta camada é fazer com que pacotes enviados em um ponto da rede cheguem ao seu destino, independentemente do caminho e das diferentes redes em que trafeguem.

Esta camada especifica vários protocolos, porém o mais difundido é o IP (*Internet Protocol*) que será explicado com maiores detalhes na Seção 2.4.3 .

c) Camada de Transporte

A camada de transporte tem por finalidade prover uma comunicação confiável entre dois processos, estando esta a ocorrer dentro da mesma rede ou

não. Deve também, garantir que os dados sejam entregues livres de erros, em seqüência e sem perdas ou duplicação.

Os principais protocolos desta camada são: *Transmission Control Protocol* (TCP) e *User Datagram Protocol* (UDP). TCP é um protocolo orientado a conexão com detecção e correção de erros. O UDP é um protocolo não orientado a conexão e não confiável, ou seja, não oferece nenhuma garantia em relação à entrega dos dados ao destino.

Os protocolos TCP e UDP são discutidos em maiores detalhes nas Seções 2.4.4 e 2.4.5.

d) Camada de Aplicação

A camada de aplicação contém todos os processos que utilizam serviços das camadas inferiores (Acesso à rede, Internet e Transporte) para transmitir dados através da rede.

As aplicações na arquitetura TCP/IP são implementadas de uma maneira isolada, ou seja, não existe um padrão que defina como deve ser estruturada uma aplicação. Cada aplicação possui seu próprio padrão.

São exemplos de protocolos da camada de aplicação: *Domain Name Service* (DNS), *Telnet*, *File Transfer Protocol* (FTP), etc.

2.4.3 IP

Ao contrário dos protocolos da camada de rede mais antigos, o IP foi desenvolvido desde o início pensando na interconectividade. O protocolo IP oferece o transporte de dados da fonte para o destino, possibilitando que estes sejam entregues mesmo que os computadores façam parte de redes diferentes. Roteadores estabelecem a conexão das várias redes e garantem que os pacotes IP recebidos sejam repassados para a rede apropriada.

O endereço IP é composto de quatro octetos divididos em duas partes: identificador de rede e identificador de *host*. De acordo com o tipo de classe definido pelos primeiros *bytes* do primeiro octeto, o IP pode estar identificando apenas a subrede, definida pelo número de máscara.

Este protocolo usa o identificador de rede para definir a melhor rota através de uma tabela de roteamento mantida e atualizada por roteadores. O IP recebe os dados da camada superior (camada de transporte) na forma de segmentos. Ocorre então, o particionamento destes e os segmentos de dados, assim constituídos, passam a se chamar datagramas. Estes datagramas são então codificados para envio à camada inferior (física) para transmissão no meio físico [17].

2.4.4 TCP

O TCP (*Transmission Control Protocol*) foi projetado especificamente para oferecer um fluxo de *bytes* fim a fim confiável em uma rede não-confiável.

A camada IP não oferece qualquer garantia de que os datagramas sejam entregues de forma apropriada. Datagramas duplicados, perda de pacotes e congestionamento são alguns problemas que a Internet apresenta. Para que isto não prejudique a transmissão dos dados, o TCP apresenta alguns mecanismos que garantem o sucesso da comunicação. Para estabelecer uma conexão, o módulo do TCP utiliza um mecanismo de negociação denominado de *three-way handshake* [18]. Os problemas de datagramas duplicados e fora de ordem são solucionados utilizando-se números de seqüência. A perda de pacotes é tratada por um mecanismo de confirmação de entrega (*acknowledgement*). Um mecanismo de controle de fluxo é também implementado pelo TCP, evitando congestionamento e transbordamento em receptores mais lentos.

Na transmissão UDP, não há confirmações ou entrega garantida dos datagramas. O processamento de erros e a retransmissão devem ser tratados por protocolos de camada superior, a camada de aplicação.

2.5 ESCOLHA DO PROTOCOLO DE TRANSPORTE

O TCP apresenta uma série de mecanismos que são úteis a aplicativos que prezam por confiabilidade na transmissão, porém estes mecanismos apresentam algumas desvantagens.

O protocolo TCP fornece um serviço confiável e orientado à conexão. É uma característica deste protocolo garantir que os dados cheguem ao receptor na mesma ordem em que foram transmitidos, sem perdas. O TCP oferece também mecanismos para o controle de fluxo e congestionamento da rede, decidindo a taxa de transmissão dos dados conforme sobrecarga na rede.

O TCP garante a confiabilidade na comunicação através da retransmissão dos pacotes perdidos ou corrompidos. Caso um pacote se perca, o receptor envia o pedido de retransmissão à fonte, informando qual pacote foi perdido ou corrompido. Enquanto o pacote perdido não for retransmitido, a aplicação não tem acesso a outros pacotes, mesmo que estes já tenham sido recebidos pelo computador. Todo este mecanismo adiciona um atraso extra na comunicação.

No monitoramento de ECG, no qual o sinal tem que ser atualizado em intervalos de tempo regulares, uma grande demora na entrega dos pacotes compromete a funcionalidade do sistema.

Descartado o uso do TCP no sistema de monitoramento, uma opção para a transmissão do ECG é a utilização do protocolo UDP. O UDP utiliza a técnica de melhor-esforço, não adicionando atraso na comunicação uma vez que não existe a retransmissão de pacotes. Outra vantagem é que o UDP não adiciona grande *overhead* aos pacotes, diminuindo a largura de banda utilizada pelo aplicativo.

Este protocolo seria a solução para o transporte do ECG pela rede, não fosse o fato de o UDP não possuir nenhum mecanismo para o sincronismo dos dados. Além disto, não há nenhum campo ou ferramenta que possibilite o controle de fluxo e congestionamento da rede.

É importante que uma aplicação de monitoramento em tempo-real apresente algum mecanismo de sequenciamento, ou seja, os dados devem ser visualizados na ordem correta e no tempo certo.

A solução encontrada foi a utilização do protocolo RTP, que utiliza o UDP para o transporte dos dados, incluindo campos que solucionam os problemas de sincronização, fluxo e congestionamento.

2.6 RTP

O RTP (*Real Time Protocol*) é um protocolo especificado pelo IETF na RFC 1889 [20]. Conceitualmente, o RTP provê transporte de rede fim-a-fim necessário a aplicações de tempo real, como por exemplo, um aplicativo de videoconferência. Ele tem por objetivo fornecer um mecanismo para levar dados sensíveis ao atraso de um ponto a outro da rede, em tempo real. Em relação a esse aspecto de transporte, o RTP atua como um mediador entre os dados a serem transmitidos e os meios que efetivam a transmissão. Resumidamente, este protocolo funciona como uma interface entre as aplicações de tempo real e os protocolos da camada de transporte.

O RTP foi designado para ser independente das camadas de rede e de transporte. Sendo independente, o RTP pode ser implementado sob qualquer protocolo, embora seu uso mais comum seja sobre o UDP. Com a utilização de UDP, além das vantagens obtidas com a simplicidade do protocolo, dois importantes serviços são disponibilizados: a multiplexação (através do conceito de portas) e a correção de erros (através do *checksum*).

Como o RTP necessita dos recursos oferecidos pelos protocolos das camadas inferiores, ele não utiliza qualquer mecanismo para assegurar a entrega de dados em ordem ou com atraso constante. Estes serviços devem ser fornecidos

pelos protocolos nos quais o RTP está encapsulado. Em adição, o RTP não fornece qualquer tipo de reserva de recursos, não garantindo qualidade de serviço em aplicações de tempo real. Novamente, políticas de Qualidade de Serviço (QoS) devem ser esperadas por protocolos de níveis abaixo do RTP.

2.6.1 Cabeçalho RTP

Um pacote RTP é um pacote de dados consistindo de um cabeçalho RTP fixo, uma lista de fontes de contribuição, ou seja, fontes que estejam enviando dados, e um *payload* (conteúdo do pacote). O pacote da camada inferior que esteja encapsulando o RTP pode conter um ou mais pacotes RTP, dependendo do método de encapsulamento empregado.

Algumas considerações em relação à formatação das informações devem ser mencionadas. O tempo absoluto é representado usando o formato de *timestamp* do NTP (*Network Time Protocol*), ou seja, escrevendo-se o número de segundos passados desde 01/01/1900 às 00:00 com 32 *bits* para a parte inteira e 32 *bits* para a parte decimal. Existe também uma maneira compacta com apenas 16 *bits* para a parte inteira e 16 *bits* para a fração. Os campos com valores numéricos inteiros são transmitidos no esquema *big-endian*, ou seja, o octeto mais significativo é transmitido primeiro.

Os campos constituintes do cabeçalho RTP são apresentados na Figura 2.11. Os doze primeiros octetos estão presentes em cada pacote RTP.

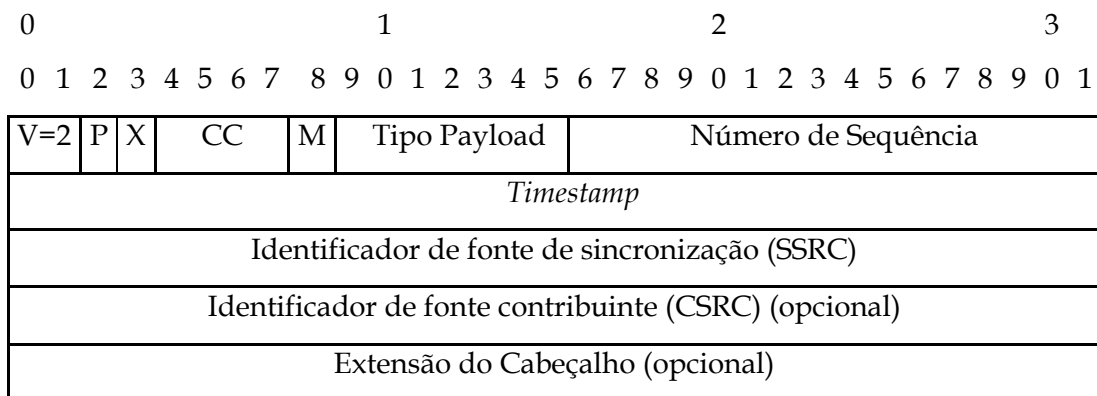


Figura 2.11 - Cabeçalho RTP

Uma breve descrição dos campos presentes no cabeçalho RTP é apresentada a seguir.

i. Versão (V) - 2 bits:

Identifica a versão do RTP. A versão atual é a "2". O valor "1" é usado para a primeira versão e a versão "0" é usada para o protocolo inicialmente implementado na ferramenta de áudio "vat".⁶

ii. Padding (P) - 1 bit:

Se este *bit* estiver em nível lógico alto, há um ou mais octetos de preenchimento no final do pacote. O último octeto de preenchimento informa quantos octetos de preenchimento devem ser ignorados.

O preenchimento pode ser necessário ao se empregar algum esquema de criptografia que necessite de um tamanho mínimo de pacote.

iii. Extensão (X) - 1 bit:

Quando este campo se encontrar ativo, o cabeçalho fixo é seguido por um cabeçalho de extensão.

iv. Contador CSRC (CC) - 4 bits:

Contém o número de endereços CSRC (Identificador de Fonte Contribuinte) que se seguem imediatamente após o cabeçalho.

v. Marker (M) - 1 bit:

⁶ *Vat* é uma aplicação multimídia em tempo real para conferências de áudio através da Internet.

A interpretação desse *bit* depende do tipo de perfil RTP adotado, tendo como objetivo sinalizar eventos significativos no fluxo RTP

vi. Tipo Payload (PT) – 7 bits:

Este campo realiza a definição do tipo de dados transportado pelo pacote RTP, objetivando distinguir um formato particular de dados dos demais.

O formato do campo de *payload* é completamente livre e cada aplicação deve definir um perfil. Informações adicionais necessárias a um formato particular de *payload* (tal como codificações de vídeo) podem ser inseridas neste campo. Dessa forma, um cabeçalho com essas informações precede os dados, possuindo significado exclusivo para a aplicação.

vii. Número de Seqüência – 16 bits:

O usuário utiliza este número para detectar perda de pacotes e para restaurar a seqüência correta dos pacotes. É incrementado de um a cada pacote RTP enviado. O valor inicial é aleatório.

viii. Timestamp – 32 bits:

Este campo reflete o instante de amostragem do primeiro byte dos dados transportados no pacote RTP. Este instante deve ser derivado de um *timer* que seja linearmente incrementado, permitindo sincronização na reprodução da mídia e cálculos de *jitter*. A freqüência do *timer* depende do tipo de formato dos dados sendo transportados e pode ser especificado para cada perfil utilizado. Para aplicações de áudio, o *timestamp* é tipicamente incrementado com a quantidade de amostras no pacote, visto que a freqüência de amostragem é fixa. Baseado neste valor, a aplicação receptora pode então reproduzir os dados de áudio no tempo exato.

ix. SSRC (Identificador de fonte de sincronização) – 32 bits:

Este campo identifica a aplicação que enviou o pacote. No caso de uma aplicação enviar diferentes mídias ao mesmo tempo, como dados de áudio e vídeo em pacotes separados, a aplicação pode agrupar os pacotes recebidos de acordo com o valor do SSRC. Todos os pacotes RTP com um SSRC comum possuem uma mesma referência de tempo e de ordenação. Se a aplicação mudar seu endereço de transporte durante a transmissão, ela também deve mudar seu SSRC.

O identificador é escolhido aleatoriamente e a chance de duas máquinas, comunicando-se acidentalmente, terem o mesmo SSRC é extremamente pequena. Em casos raros em que isto acontece, especificações da RFC1889 fornecem um apropriado curso de ações que devem ser tomadas para resolver o problema.

x. CSRC (Identificador de fonte contribuinte) - 1 a 16 itens, 32 bits cada:

Quando um fluxo RTP é o resultado de uma combinação de vários fluxos contribuintes, a lista contendo os SSRCs de cada um dos fluxos contribuintes é adicionada ao cabeçalho RTP do fluxo resultante através do campo CSRC. O fluxo resultante tem o seu próprio SSRC.

O número máximo de fontes contribuintes é dezesseis. Caso haja um número maior de fontes contribuindo com o fluxo de dados, apenas as dezesseis primeiras serão incluídos na lista CSRC.

xi. Extensão do cabeçalho:

Para permitir implementações particulares do RTP, um mecanismo de extensão é fornecido. Esse mecanismo permite que informações adicionais sejam inseridas ao cabeçalho do RTP, tendo como princípio que este cabeçalho possa ser

ignorado por aplicações sem extensão. Apenas uma extensão pode ser adicionada ao cabeçalho padrão.

2.7 RTCP

O RTCP (*Real Time Control Protocol*) foi criado pelo IETF para ser um protocolo de controle que auxiliasse o RTP na sua tarefa de transmissão de dados em tempo real. Seu principal objetivo é disponibilizar *feedback* QoS para que os participantes de uma conferência multimídia possam adaptar-se dinamicamente a problemas na rede. Porém, o RTCP possui outras aplicações como a troca de informações de usuários e, mais importante, a associação de mídias provenientes de diferentes sessões.

Um pacote RTCP é um pacote de controle consistindo de um cabeçalho fixo seguido por elementos estruturados variando de acordo com o tipo de pacote RTCP. Assim como o RTP, o RTCP pode ser expandido para um perfil particular visando acomodar adaptações necessárias à determinada aplicação.

O protocolo de controle do RTP é baseado na transmissão periódica de pacotes de controle para todos os participantes de uma sessão, usando os mesmos mecanismos de comunicação utilizados para os dados. O protocolo da camada inferior deve fornecer multiplexação de dados e de pacotes de controle, usando, por exemplo, número de portas separadas.

O RTCP realiza 4 funções:

- 1) A função principal é fornecer *feedback* sobre a qualidade da transmissão dos dados, estando relacionado ao controle de fluxo e congestionamento normalmente ligado aos protocolos de transporte. O *feedback* é útil para o controle de codificações adaptativas, sendo importante na detecção de erros de transmissão através de “relatórios de recepção”. Utilizando uma rede como a TCP/IP, uma estação que não está participando da conferência pode monitorar a

qualidade da comunicação sem receber os dados RTP, sendo definida como “estação monitora”.

2) O RTCP distribui um identificador que pode ser usado para agrupar diferentes *streams* (seqüência de bytes), como por exemplo, áudio e vídeo pertencentes a um mesmo participante.

3) Enviando pacotes RTCP periodicamente, cada sessão pode determinar o número de participantes. O RTP identifica isto, uma vez que podem existir participantes exclusivamente passivos, ou seja, que não enviam nenhum pacote RTP. Esses participantes passivos enviam somente pacotes RTCP.

4) Uma função opcional é a distribuição das informações sobre os participantes. Essas informações podem ser usadas em uma interface com o usuário, por exemplo, mostrando os dados dos participantes.

Para que as funcionalidades mencionadas sejam suportadas, o RTCP especifica cinco tipos de pacotes. O *sender report* (SR), usado por participantes ativos (transmissores), é um pacote que fornece estatísticas sobre a transmissão e recepção de pacotes RTP. O *receiver report* (RR) é utilizado por um participante passivo (receptor) para se informar sobre a recepção de pacotes RTP durante a sessão. Informações (como nome, e-mail, telefone) podem ser transmitidas por pacotes RTCP do tipo *source description* (SDS). Existem também pacotes RTCP do tipo *application* (APP), que fornecem dados pré-definidos pela aplicação. Finalmente, quando um participante está para deixar uma sessão RTP, ele envia um pacote RTCP do tipo *goodbye* (BYE), com a finalidade de avisar a todos os participantes de sua saída.

3 METODOLOGIA

Este Capítulo 3 apresenta o sistema de transmissão desenvolvido, detalhando o *hardware*, o *firmware* e o *software* que compõe o sistema.

O sistema proposto é constituído pelos seguintes módulos (Figura 3.1).

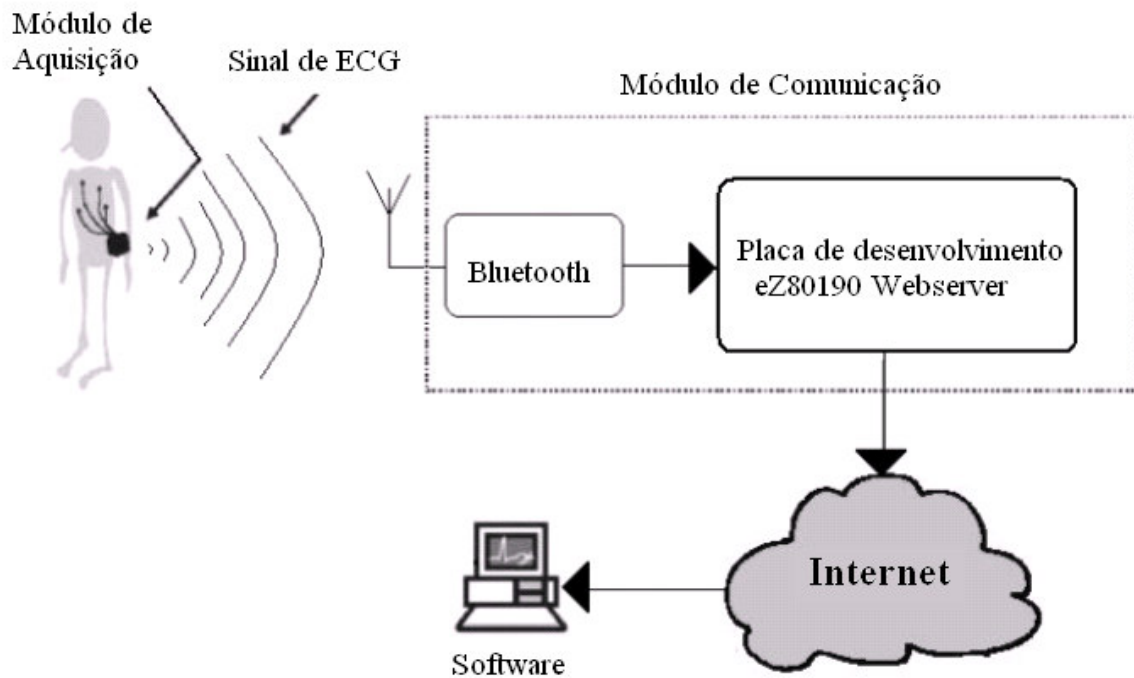


Figura 3.1 - Sistema de monitoramento proposto

O módulo de aquisição adquire o sinal de ECG e o transmite por rádio frequência ao segundo módulo. O módulo de comunicação recebe o ECG e o envia pela Internet a um computador situado em um centro de supervisão, onde é analisado por um especialista.

Os módulos e o *software* do sistema são detalhados nas próximas seções.

3.1 MÓDULO DE AQUISIÇÃO DE SINAL DE ECG

A função de aquisição do sinal de ECG é realizada pelo *hardware* desenvolvido neste laboratório de pesquisa por Kolm [21]. O *hardware* adquire o sinal de duas derivações de ECG (bipolares I e II) por meio de eletrodos colocados

no corpo do paciente. A Figura 3.2 mostra o módulo de captação posicionado em voluntário.



Figura 3.2 - Módulo de aquisição do sinal de ECG

Após amplificação (AD620 - *Analog Devices*) e filtragem através de filtros passa-alta (frequência de corte = 0,5Hz) e passa-baixa (frequência de corte = 50Hz), os sinais de ECG são amostrados (8 bits de resolução e taxa de amostragem de 250Hz) por um microcontrolador ADuC841 (*Analog Devices*). A Figura 3.3 ilustra o processamento realizado pelo módulo em questão.

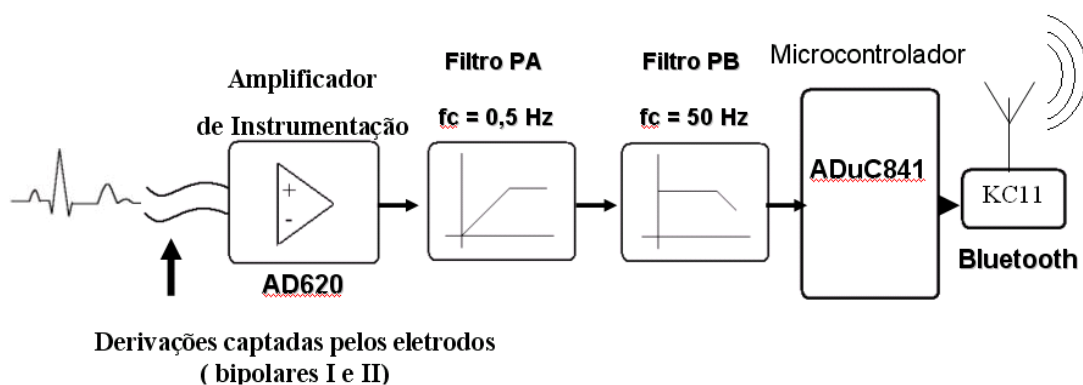


Figura 3.3 - Diagrama em blocos das etapas de processamento realizado pelo módulo de aquisição.

Após, os sinais amostrados são enviados para o módulo de comunicação através do componente KC11 (*KC Wirefree*) que utiliza tecnologia sem fio Bluetooth para a transmissão.

3.2 MÓDULO DE COMUNICAÇÃO

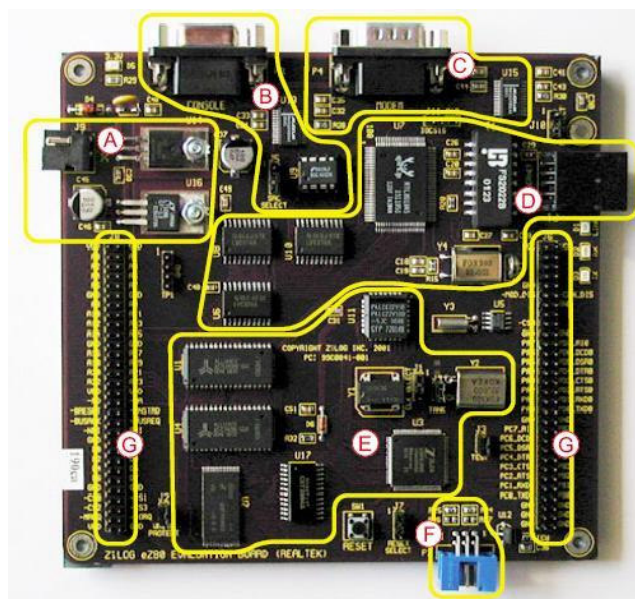
O módulo de comunicação é composto pela placa de desenvolvimento eZ80190 *Webserver* (Zilog) e um dispositivo *Bluetooth* KC11 (*KC Wirefree*).

A placa de desenvolvimento tem como principais características:

- dois canais DMA;
- duas portas DB9 RS-232;
- interface *Ethernet* 10MBPS;
- processador eZ80190 de 40 MHz de *clock*;
- 1 Mbyte de RAM; e
- bibliotecas para aplicações de rede.

A escolha desta placa deve-se ao fato dela possuir as interfaces de rede e serial necessárias para o desenvolvimento do sistema. As facilidades de comunicação TCP/IP oferecidas também contribuíram para o uso desta placa de desenvolvimento.

A Figura 3.4 mostra a placa de desenvolvimento. As áreas circundadas em amarelo ilustram os diferentes blocos que compõe a placa.



- | | |
|-----------------------|---------------------|
| A. Alimentação | E. CPU e Memória |
| B. Interface Serial | F. Interface ZDI |
| C. Interface Serial | G. Slot de expansão |
| D. Interface Ethernet | |

Figura 3.4 - Placa de desenvolvimento eZ80190

O dispositivo *Bluetooth KC11* utilizado no módulo transmissor possui alcance máximo de 200 metros, comunicação UART e uma taxa de transmissão máxima de 723,2 Kbps. O dispositivo ainda oferece uma interface serial DB9. A Figura 3.5 mostra o dispositivo.

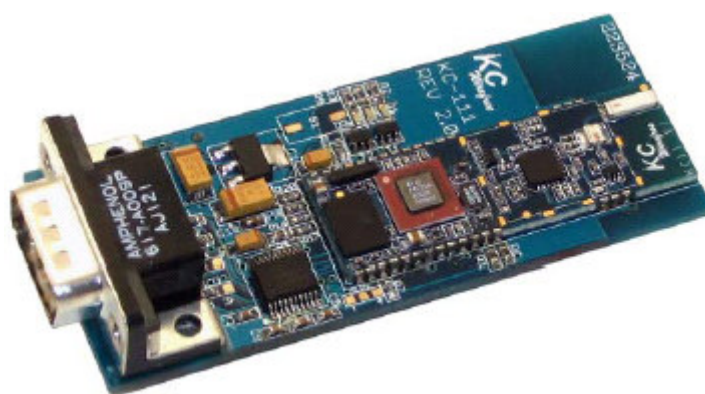


Figura 3.5 - KC11 da KC Wirefree

O protocolo *Bluetooth* tem sido proposto para estabelecer um padrão mundial na transferência de dados entre dispositivos eletrônicos por rádio frequência, substituindo os cabos utilizados na comunicação.

Inicialmente, a especificação da tecnologia *Bluetooth* baseou-se na norma IEEE 802.11 para padronização das *Wireless Local Area Network* (WLAN). Porém, durante o seu desenvolvimento, foi proposta a definição de *Personal Area Network* (PAN) para melhor adequar a utilização desta tecnologia e similares.

O *Bluetooth* opera na banda destinada a aplicações industriais, científicas e médicas (ISM) de 2,4 GHz a 2,4835 GHz. A comunicação *Bluetooth* não gera e não sofre interferência de outros equipamentos. Para tal, o protocolo utiliza técnicas de espalhamento espectral e controle adaptativo de potência. [22]

3.1.1 Funcionamento

Diagrama de blocos do módulo de comunicação é apresentado na Figura 3.6.

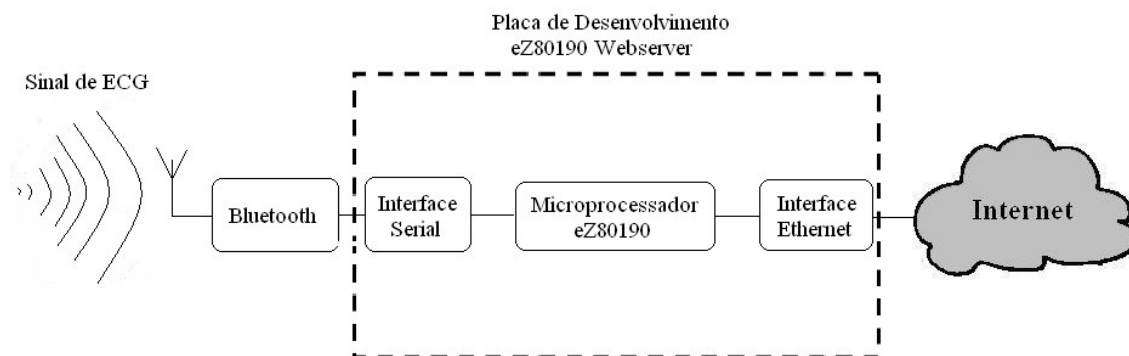


Figura 3.6 - Diagrama de Blocos do Módulo de Comunicação

O sinal de ECG enviado pelo módulo de aquisição é recebido pelo dispositivo *Bluetooth* KC11 a uma taxa fixa de transmissão de 57.600 bps. Esta

largura de banda é suficiente para a transmissão de até doze derivações com resolução de oito bits e frequência de amostragem de 250 Hz.

O KC11, conectado a interface serial do módulo de transmissão, disponibiliza o ECG recebido ao microprocessador eZ80190. Após o seu encapsulamento, o sinal de ECG é enviado para o microcomputador que possibilitará sua visualização. A conexão com o meio físico da rede é feita pela interface *ethernet* RJ45, presente na placa.

A Figura 3.7 apresenta o sistema de transmissão constituído pela placa de desenvolvimento eZ80190 *Webserver* e o módulo *Bluetooth* KC11.

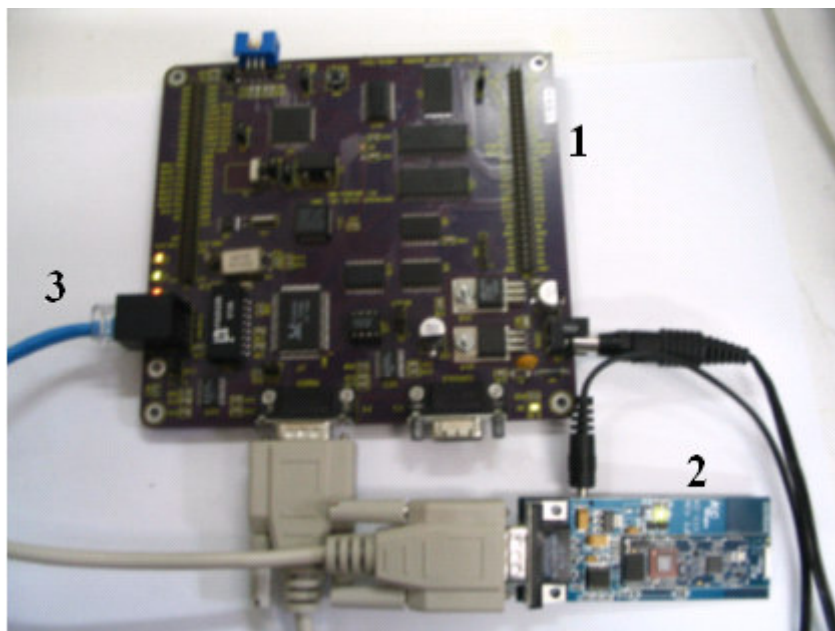


Figura 3.7 - Módulo de Comunicação.

(1) Placa de desenvolvimento; (2) KC11 conectado à interface serial da placa; (3) Cabo de acesso à rede.

O controle da transmissão dos dados é feita por firmware gravado na memória interna *flash* do microprocessador eZ80190.

3.2 FIRMWARE

O *firmware* foi desenvolvido para o controle do módulo de comunicação. A execução do *firmware* é realizada pelo microprocessador eZ80190 (Zilog). O sinal

de eletrocardiograma, contendo duas derivações, enviado pelo módulo de aquisição, passa por dois algoritmos: um responsável pela recepção do sinal e outro que encapsula e transmite o ECG. Estes algoritmos estão codificados em linguagem Ansi C.

O fluxograma da Figura 3.8 ilustra as tarefas executadas pelo *firmware* desenvolvido.

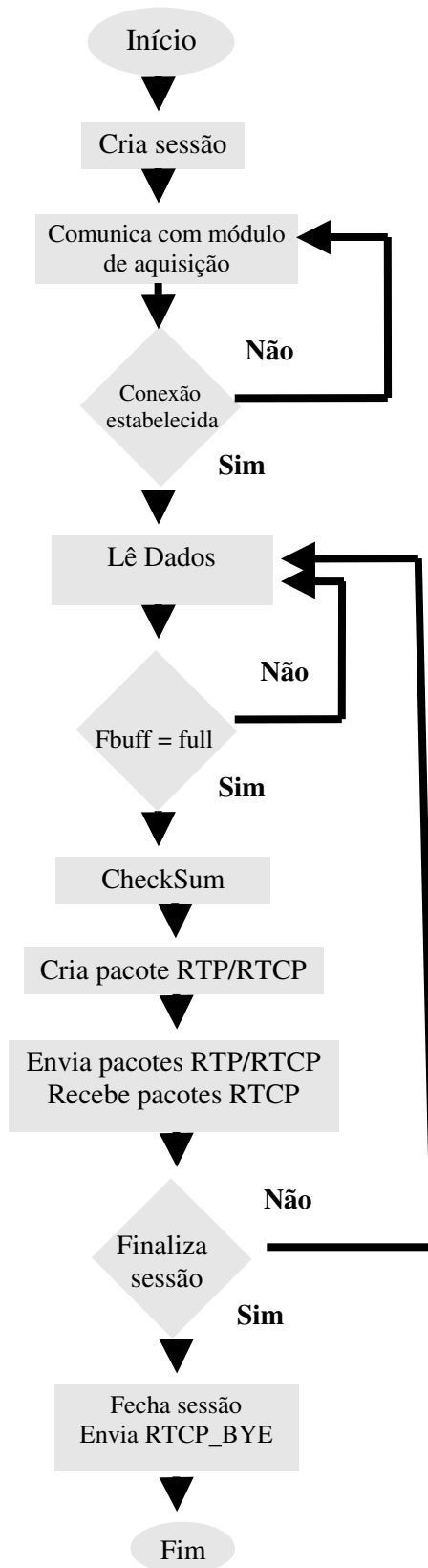


Figura 3.8 - Fluxograma do firmware desenvolvido

Inicialmente, cria-se a sessão de comunicação no protocolo de transporte UDP. As portas utilizadas na transmissão são também estabelecidas. Após este processo, a porta serial da placa de desenvolvimento é configurada para a recepção dos dados enviados pelo dispositivo *Bluetooth*. Uma solicitação de conexão é então, enviada ao módulo de aquisição. Caso a conexão não se estabeleça, o pedido é retransmitido. Estabelecida a conexão, inicia-se a leitura e armazenamento do sinal no *buffer* circular. Este *buffer* é constantemente preenchido com o sinal de ECG. Um *buffer* auxiliar faz a cópia dos dados e envia-os para uma função que os encapsula em pacotes RTP, sendo então, transmitidos ao endereço de destino. Verifica-se em seguida, o recebimento de pacotes RTCP enviados pelo software receptor. Caso não haja nenhum pedido de desconexão, o sistema continua a enviar o sinal de ECG. Caso contrário, um pacote RTCP do tipo BYE é enviado para confirmar o pedido. O *link* de comunicação com o módulo de aquisição é interrompido, as funções em execução finalizadas e as portas de comunicação fechadas.

3.2.1 Recepção do Eletrocardiograma

O módulo de aquisição envia o sinal de ECG periodicamente a fim de manter o aspecto de tempo-real no monitoramento. O algoritmo de recepção destes dados deve ler constantemente a interface de entrada (porta serial) para que dados do ECG não sejam sobrescritos. Este processo não deve ter um tempo de processamento muito longo para não afetar a monitoração em tempo-real.

Para evitar *overflow*, utiliza-se um *buffer* circular no armazenamento dos dados. O tamanho do *buffer* circular foi escolhido levando-se em conta o atraso máximo aceitável para o envio dos dados ao algoritmo de encapsulamento e transmissão.

3.2.2 Encapsulamento e transmissão

No encapsulamento e transmissão do sinal de ECG, tem-se a definição do tamanho dos pacotes de dados a serem enviados ao receptor. Para tal, é considerada a quantidade de informação que este pacote irá transportar e qual *overhead* aceitável para a transmissão do eletrocardiograma.

Em caso da perda de um pacote, a quantidade de informação perdida não deve prejudicar significativamente a visualização do sinal. Por exemplo, um pacote com 50 amostras de uma derivação do ECG (50 *bytes*) corresponde a 200 ms de sinal (para uma frequência de amostragem de 250 Hz). Caso haja a perda deste pacote, 1/5 do sinal não seria visualizado pelo especialista. Para reduzir tal comprometimento do sinal, pode-se reduzir o número de amostras por pacote para 10 amostras, ou seja, 40 ms. Desta maneira, apenas 40 ms do sinal não seriam visualizados.

Cada pacote transporta ainda, informações dos cabeçalhos dos protocolos envolvidos na transmissão. No sistema proposto, cada pacote possui 48 *bytes* adicionais. São 20 *bytes* para o cabeçalho do protocolo RTP, 8 *bytes* para o protocolo da camada de transporte UDP e 20 *bytes* para o cabeçalho IP, presente na camada de rede. A Figura 3.9 mostra um pacote de dados RTP.

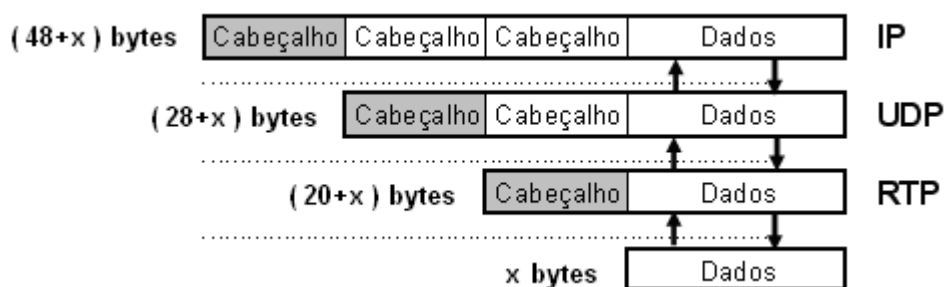


Figura 3.9 - Cabeçalhos dos protocolos de um pacote de dados RTP

Para minimizar o efeito da perda de pacotes, cada pacote deveria conter uma única amostra do sinal (1 *byte*). Porém, o tamanho mínimo do datagrama

seria de 49 bytes (1 *byte* da amostra do sinal + 20 *bytes* do RTP + 8 *bytes* do UDP + 20 *bytes* do IP), ou seja, um *overhead* de 98%. Para a frequência de amostragem utilizada (250 Hz), a largura de banda necessária é de 98 kbps (250 amostras/s * 49 bytes/amostra * 8 bits).

Observa-se que o aumento do número de amostras por pacote provoca a diminuição do *overhead* e, conseqüentemente, a redução da largura de banda. Um pacote com 500 amostras possui 548 *bytes* e um *overhead* de 8,76%. A largura de banda necessária para a transmissão seria de 2,192 kbps (250 amostras/s * 548 bytes/500 amostras * 8 bits). Contudo, a perda de um pacote corresponde à perda 2 s do sinal de ECG.

A transmissão de 50 amostras por pacote, 25 para cada derivação, adequou-se melhor ao objetivo, devido, principalmente, à baixa vazão requerida.

A Tabela 3.1 apresenta um comparativo do número de amostras por pacote, o intervalo de sinal amostrado contido em cada pacote, o *overhead* e largura de banda necessária.

Tabela 3.1 - Comparativo entre datagramas de diferentes tamanhos

t_{pacote} (ms)	Amostras/pacote	<i>Overhead</i> (%)	Vazão (Kbps)
4	1	98	98
20	5	90	21,2
40	10	82	11,6
100	25	65	5,84
200	50	49	3,92
400	100	32	2,96
1000	250	16	2,38
2000	500	8	2,19

Para a transmissão do eletrocardiograma, uma sessão RTP é iniciada. Os pacotes são enviados através de uma porta de comunicação de valor par, segundo especificação da RFC1889. Estes pacotes RTP transportam os dados do ECG

através da rede IP. Em outra porta, de valor ímpar, envia-se os pacotes RTCP responsáveis pela realimentação (*feedback*) da transmissão dos pacotes RTP. A qualidade da conexão é informada por pacotes RTCP.

O endereço para a emissão dos pacotes é do tipo *unicast*, ou seja, cada pacote é enviado diretamente para a interface associada ao endereço. Desta forma, a identificação do paciente monitorado é realizada pelo identificador SSRC do cabeçalho RTP.

O módulo de comunicação é configurado com um endereço de IP fixo. Informações como máscara de sub-rede, *gateway* padrão e servidor DNS também são configuradas no módulo.

O perfil do pacote RTP deve ser adaptado para cada tipo de aplicação. Existem perfis pré-definidos, como por exemplo, os presentes na RFC3551 [23], que estipula valores iniciais para os campos presentes no cabeçalho RTP. Por exemplo, um pacote RTP com o campo *Payload Type* igual a 26 possui dados de vídeo com compactação JPEG.

Como o RTP foi inicialmente projetado para aplicações relacionadas à troca de pacotes de áudio e vídeo, não existe uma RFC que especifica um perfil padrão para aplicações na área de transmissão de sinais biológicos.

Assim, buscou-se criar um perfil adequado para a transmissão de ECG. O campo Versão (V) identifica a versão atual do protocolo RTP: Número 2. O *Padding* (P) recebe o valor zero, pois nenhuma criptografia é utilizada. Como o sistema não necessita de um cabeçalho de extensão, escreve-se zero no campo *Extension* (X). Os campos *CSRC Count* (CC) e *Marker* (M) possuem valor zero.

O campo *Payload Type*, responsável pela identificação do tipo de dado que o pacote RTP carrega, foi escolhido dentro de uma faixa de valores livres (77-95). Para o *Timestamp*, o número da última amostra de ECG contida no pacote é utilizado, uma vez que a taxa de amostragem do eletrocardiograma é fixa. Estes valores simplificam o cálculo da variação do atraso dos pacotes (*jitter*) por parte do computador receptor.

O valor para a fonte de sincronização (campo SSRC) é escolhido de forma aleatória, como recomenda a RFC1889. O campo CSRC não foi utilizado nesta aplicação.

O cabeçalho da aplicação pode ser visualizado na Figura 3.10.

V=2	P=0	X=0	CC=0	M=0	PT=80	Número de Sequência = 1,2,3,4.....
Timestamp = número da amostra / frequência de amostragem						
SSRC = número aleatório						
Dados...						

Figura 3.10 - Cabeçalho RTP para a transmissão do eletrocardiograma

Para o controle da transmissão dos pacotes RTP, optou-se por enviar apenas o pacote RTCP do tipo *Goodbye (BYE)*. Aplica-se quando o módulo de captação parar o envio do ECG ou o software solicitar pedido de desconexão.

A biblioteca do microprocessador utilizado permite a comunicação com redes IP utilizando o protocolo UDP. Para transmitir dados através do protocolo RTP/RTCP, foram criadas estruturas contendo os cabeçalhos para os mesmos. Todas as estruturas foram baseadas na RFC1889 que estabelece o nome, tamanho e tipos de variáveis dos campos. Uma função anexa o cabeçalho RTP/RTCP ao datagrama UDP, formando pacotes RTP/RTCP.

Um cabeçalho de *Checksum* foi adicionado dentro do campo de dados do pacote RTP para que pacotes corrompidos sejam descartados. O *Checksum* é computado pelo transmissor através de algum algoritmo que tem como entrada os *bits* originais a serem transmitidos. Após computar esses *bits*, o transmissor os acrescenta aos *bits* de informação propriamente dita, para então prosseguir com a transmissão do pacote. Quando o pacote é recebido, o receptor, conhecendo o algoritmo utilizado pelo transmissor, pode recomputar os *bits* de redundância e compará-los com os respectivos *bits* recebidos no pacote. Se eles forem diferentes, detectou-se a presença de um erro. Vale salientar que o *Checksum* é uma ferramenta simples para a detecção de erros e não para correção de erros. A adição do *checksum* no *buffer* de dados do pacote garante que o sinal de ECG mostrado na

tela do microcomputador seja idêntico ao sinal correspondente emitido pelo módulo de comunicação. No caso de uma inversão de *bits* no campo de dados do pacote RTP, o *software* descarta o pacote corrompido.

3.3 SOFTWARE PARA MICROCOMPUTADOR

O *software*, denominado Core 1.0, desenvolvido em linguagem C++ para sistema operacional *Windows*, é responsável pela recepção, processamento e visualização dos pacotes enviados pelo módulo transmissor.

Os passos executados pelo aplicativo são mostrados na Figura 3.11.

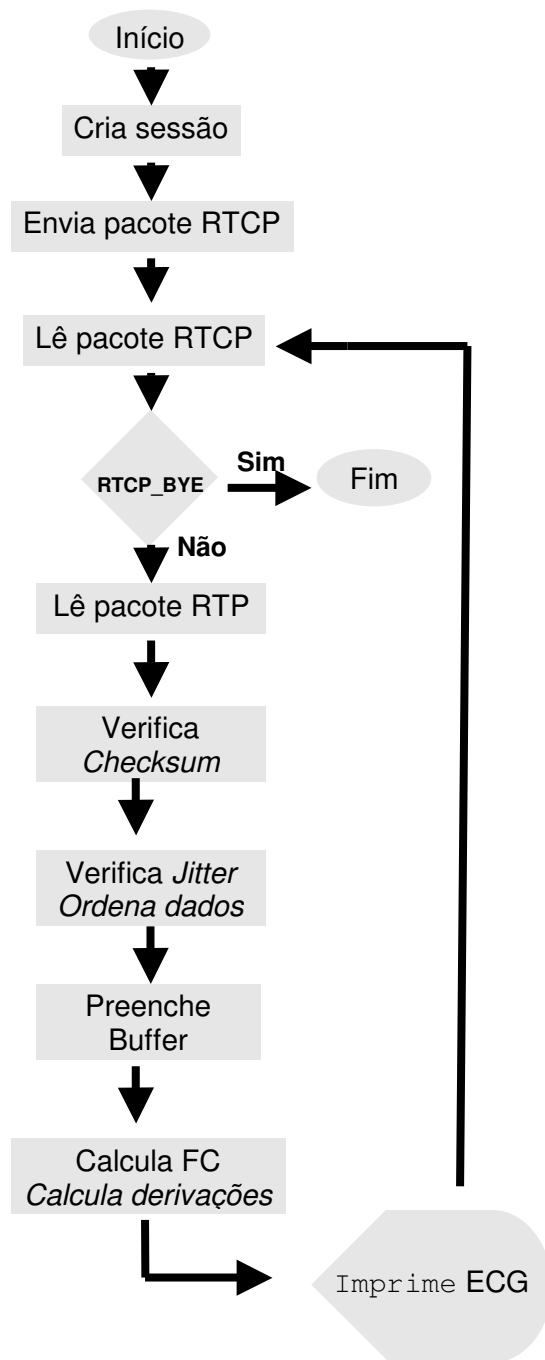


Figura 3.11 - Fluxograma do Software

O aplicativo estabelece conexão com o módulo de comunicação e inicia-se a recepção dos dados. Após o ordenamento dos pacotes, um *buffer* de armazenamento é preenchido com o sinal de ECG para posterior visualização na tela do PC.

3.3.1 Conexão com a Internet

O *software* inicia a conexão com a rede através de *sockets*⁷ datagrama, ou seja, canais para comunicação entre dois computadores. O aplicativo abre dois canais: um para a recepção de pacotes RTP e outro para a troca de pacotes RTCP.

O protocolo de transporte utilizado por estes *sockets* é o UDP, já que os protocolos RTP/RTCP são implementados sobre o mesmo (Seção 2.6). Nesta etapa, os *sockets* são configurados para executar a comunicação através de portas base inicializadas no módulo de transmissão.

Para que a comunicação pudesse ser estabelecida utilizando-se o protocolo RTP/RTCP, uma série de estruturas foram criadas, contendo os cabeçalhos destes protocolos. Para cada campo dos cabeçalhos foi utilizado o número de bits correspondente, indicado pela RFC1889. Estas estruturas foram anexadas aos datagramas UDP, formando pacotes RTP/RTCP.

Após a configuração da conexão, um pacote RTCP do tipo SDES é enviado ao módulo transmissor, identificando o computador receptor. Quando o módulo de comunicação reconhecer o endereço IP e a porta destino, o programa começa a receber o ECG.

3.3.2 Ordenação dos pacotes

Um algoritmo verifica o número de seqüência do pacote e o instante de tempo da amostra de ECG ao qual o pacote pertence. Esta análise propicia que pacotes atrasados sejam ordenados. Por exemplo, dez pacotes com as amostras do ECG são recebidos. Supondo que haja atraso na rede, fazendo com que os pacotes de números 2 e 6 cheguem fora de ordem, o aplicativo deve ordená-los. Realizada

⁷ Em documentos RFC relacionados ao TCP e UDP, um socket em um computador é definido como a combinação de um endereço IP, um protocolo de transporte e o número da porta que o protocolo realiza a comunicação.

esta correção, os pacotes são colocados em ordem cronológica dentro de outro buffer para visualização. A Figura 3.12 ilustra o exemplo mencionado.

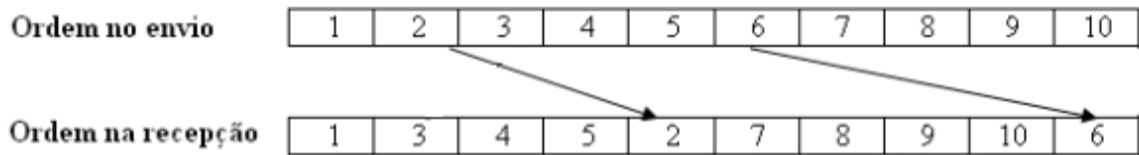


Figura 3.12 - Pacotes fora de ordem causados por atraso na rede

Caso haja perda de um pacote, o trecho de ECG perdido é substituído por zero no eletrocardiograma. Isto é realizado para que o próximo trecho do sinal não se justaponha a amostra anterior, gerando uma forma de onda de difícil interpretação na tela do computador.

3.3.3 Determinação da Frequência Cardíaca

O *software* fornece a frequência cardíaca, atualizando os valores na tela do PC a cada cinco segundos.

A frequência cardíaca é determinada como o inverso da distância de uma onda R até a outra onda R subsequente. Para a localização dos pontos de ocorrência do complexo QRS, da onda R mais especificamente, utilizou-se um algoritmo de detecção [24] baseado na primeira derivada do sinal de ECG. A primeira derivada é calculada para cada ponto do sinal de ECG, segundo a Fórmula (3.1):

$$Y(n) = -2 X(n-2) - X(n-1) + X(n+1) + 2 X(n+2), \quad 2 < n < M; \quad (3.1)$$

Onde $X(n)$ é a amostra do sinal de ECG, $Y(n)$ é a derivada do sinal e M é o número total de amostras do sinal de ECG.

O limiar de detecção é determinado como parte fracionária do valor máximo do vetor da primeira derivada do sinal, segundo expressão (3.2):

$$\text{Limiar} = 0,7 \max [Y(n)] \quad (3.2)$$

Após o cálculo do limiar, é feita a busca dos pontos da derivada do sinal de ECG que excedam este limite. O ponto que exceder o valor limiar é um candidato QRS.

A escolha deste algoritmo de detecção deve-se ao fato deste apresentar baixa complexidade de cálculo. Em aplicativos de tempo-real, deve-se evitar algoritmos que possam adicionar atrasos ao processamento do sistema.

Uma vez localizada a onda R e observados os intervalos entre uma onda a outra (intervalos RR), a frequência cardíaca é estimada com base nos intervalos RR que satisfizerem a condição $0,5 \text{ RR}_{med} \leq \text{RR} \leq 1,5 \text{ RR}_{med}$, onde RR_{med} é a média dos intervalos RR contidos no trecho de sinal de ECG.

Através da equação (3.3) obtém-se a frequência cardíaca em batimentos por minuto (bpm):

$$\text{FC}_i = (1/\text{RR}_i) \cdot 60 ; \quad (3.3)$$

Onde RR_i é dado em segundos.

Calculada a frequência cardíaca para cada segmento válido, é computada a média através da equação (3.4).

$$\text{FC}_{med} = \frac{1}{M} \sum_{i=0}^{M-1} \text{FC}_i \quad (3.4)$$

Onde: FC_{med} = média no intervalo, FC_i = valor da frequência cardíaca para o segmento RR_i e M = número de segmentos RR válidos.

3.3.4 Cálculo das Derivações

As derivações do plano frontal do eletrocardiograma convencional possuem informações redundantes. Estas redundâncias permitem, através do monitoramento de duas derivações (bipolares I e II), o cálculo matemático de outras quatro (bipolar III e aumentadas aVL, aVR e aVF) baseado na lei de Kirchhoff [12]. Apesar das informações redundantes contidas, estas seis derivações ainda são utilizadas por estarem associadas ao conhecimento adquirido durante anos de estudo e observação de eletrocardiogramas.

A segunda lei de Kirchhoff diz que a soma algébrica das tensões instantâneas ao redor de uma malha fechada é igual a zero. Considerando o triângulo de Einthoven (Figura 2.4) uma malha fechada, e obtêm-se:

$$I + III - II = 0 \quad (3.5)$$

Da Eq.(3.5) obtém-se a derivação III em função das derivações I e II:

$$III = II - I \quad (3.6)$$

As derivações aumentadas são tomadas a partir de um dos eletrodos (braço direito, braço esquerdo ou perna esquerda) e a tensão média da derivação oposta a este eletrodo, conforme Figura 3.13. No desenho, a referência é tomada a partir da média de dois eletrodos. A média é feita pelos dois resistores R na entrada inversora do amplificador. O resistor R/2 é usado para casamento da impedância de entrada do amplificador.

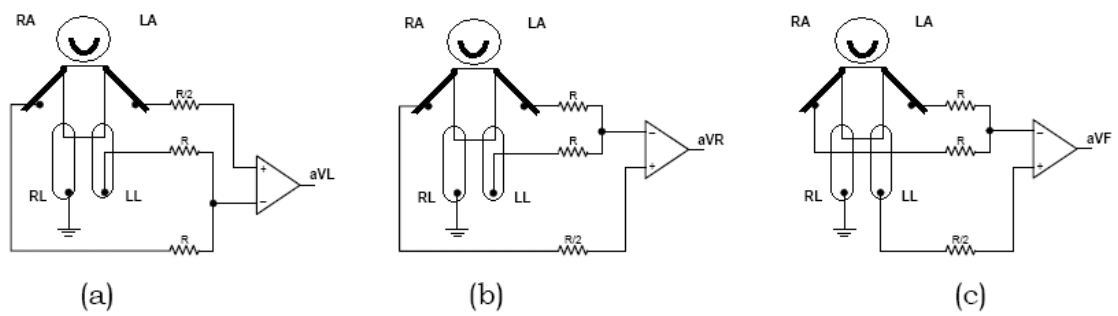


Figura 3.13 - Aquisição das derivações aumentadas aVL (a), aVR (b), e aVF (c).

Na configuração de aquisição de aVL, considera-se a malha de Kirchhoff passando pela derivação I e pela metade da derivação dois, assim:

$$aVL - I + (II/2) = 0 \quad (3.7)$$

Obtem-se, pela Eq. (3.7), a derivação aVL em função das derivações I e II:

$$aVL = I - (II/2) \quad (3.8)$$

Da mesma forma, pode-se obter as derivações aVR e aVF:

$$aVR = - (I + II) / 2 \quad (3.9)$$

$$aVF = II - (I/2) \quad (3.10)$$

O aplicativo, a partir das derivações I e II recebidas, obtêm as derivações III, aVL, aVR e aVF a partir das relações expressas em (3.6), (3.8), (3.9) e (3.10). Por fim, as seis derivações são mostradas na tela do PC em tempo de recebimento dos pacotes de dados.

3.3.5 Visualização do sinal de ECG

O acesso ao vídeo do PC é feito através de funções de manipulação de pixels. Os comandos nativos do compilador utilizado na construção do software demonstraram-se ineficientes em aplicações que exijam a inserção rápida de pixels. A visualização do sinal de ECG em tempo real exige que cada ponto do sinal seja mostrado na tela do PC a cada quatro milisegundos.

Para que isto ocorra, utilizou-se a biblioteca de uso livre (*freeware*) Graphics32. Esta biblioteca é otimizada para a manipulação de pixels no formato de 32 bits e fornece operações rápidas na manipulação direta de pixels e primitivas gráficas superando as funções nativas do compilador. A maior velocidade de operação e desempenho da *Graphics32* se deve ao fato de que ela permite a atualização de somente uma área específica do gráfico, ao contrário das funções nativas que necessitam atualizar o gráfico inteiro a cada novo pixel.

A Figura 3.14 apresenta a tela de visualização do eletrocardiograma.

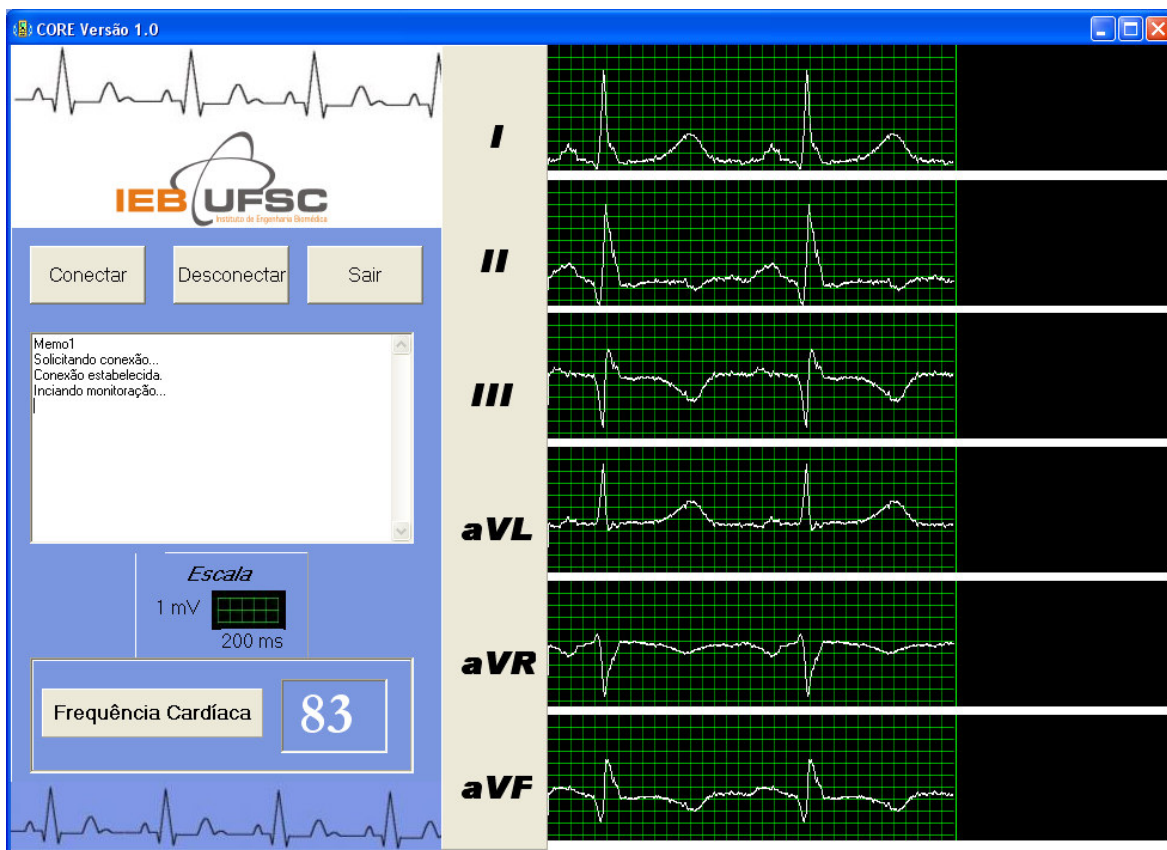


Figura 3.14 - Tela do software executado em sistema operacional Windows.

A tela do aplicativo mostra seis derivações: as bipolares I e II ; as derivações bipolar III e aumentadas aVL, aVR e aVF, calculadas a partir das derivações I e II captadas do paciente.

A interface também ilustra a frequência cardíaca do paciente, em batimentos por minuto. Há ainda um campo com informações sobre a conexão entre o módulo de comunicação e o PC. A escala de tempo e amplitude dos gráficos do ECG é informada abaixo do campo de informações.

3.3.6 Diagrama de Classes

O diagrama de classes em UML (*Unified Modeling Language*) do software desenvolvido é mostrado na Figura 3.15. [30]

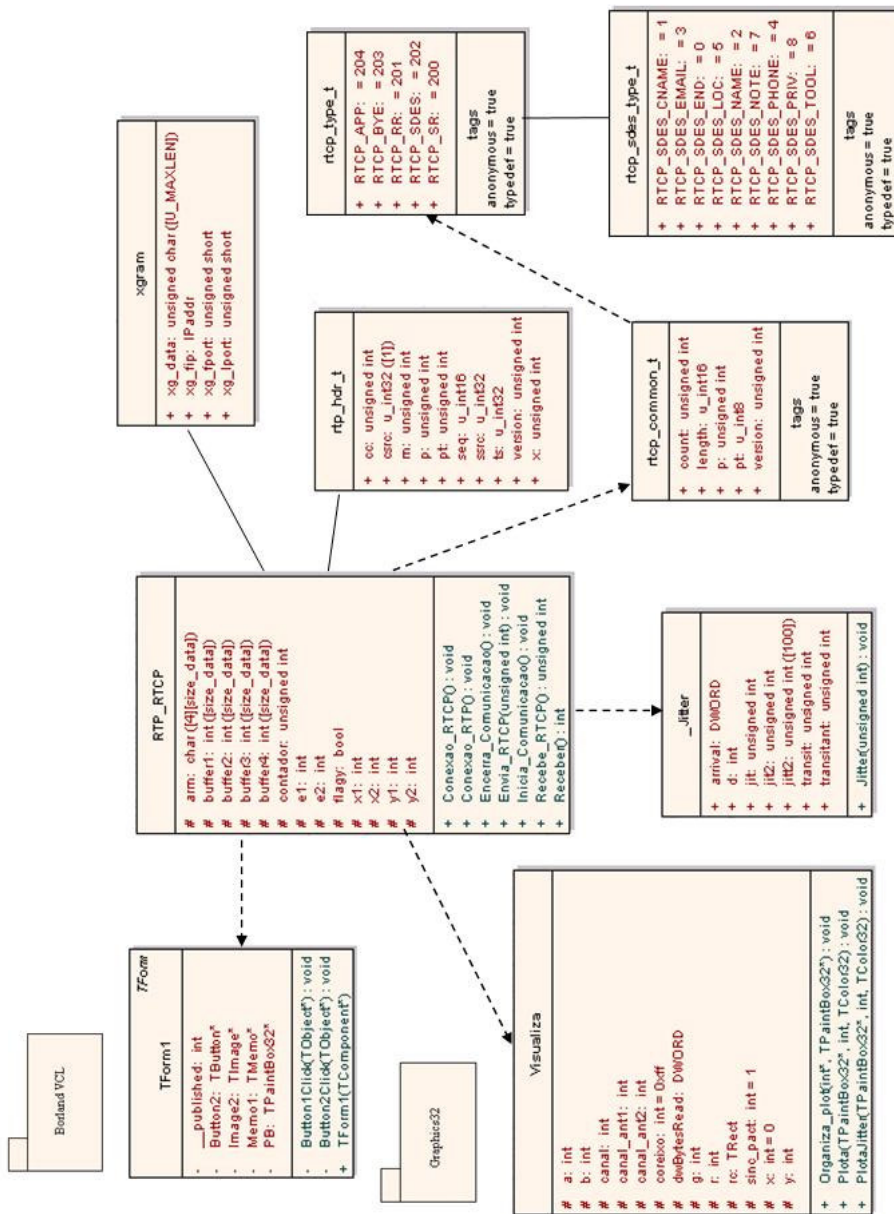


Figura 3.15 - Diagrama de classes em UML do software Core 1.0

O software Core 1.0 é composto por três classes principais: RTP_RTCP, _Jitter e Visualiza. A classe RTP_RTCP contém as funções responsáveis pela

configuração dos *sockets* de comunicação com o módulo. As rotinas de envio dos pacotes RTCP e recepção dos pacotes RTP que contém o ECG também são implementadas nesta classe, bem como o algoritmo de ordenação dos pacotes (Seção 3.3). Dentro da classe `_Jitter` encontram-se as funções responsáveis pelo cálculo do jitter. Na classe `Visualiza` encontram-se a função para cálculo das seis derivações visualizadas, a função que determina a frequência cardíaca, assim como a função responsável pela visualização do ECG na tela do PC.

As demais classes englobam as estruturas com os cabeçalhos dos protocolos RTP/RTCP.

4 RESULTADOS

Para uma avaliação inicial do sistema de monitoramento proposto, foram simulados quatro cenários para a transmissão do eletrocardiograma. Procurou-se explorar, dentro destes cenários, situações em que o sistema possa ser aplicado.

Por ser tratar de um sistema em tempo-real, os parâmetros utilizados para avaliar o desempenho do sistema são a taxa de perdas e a variação do atraso (*jitter*). Ambas informações são obtidas da análise de campos nos pacotes RTP/RTCP.

O acompanhamento dos parâmetros mencionados é realizado por *software* escrito em linguagem C++. Este mostra a taxa de perdas e o *jitter* correspondente a cada transmissão de ECG. A interface visual do mesmo é apresentada nas próximas seções.

4.1 TESTES

Diversos testes para quatro diferentes cenários foram realizados.

O primeiro cenário simula a transmissão do ECG através de uma rede local (*Ethernet*) de um hospital. Para tal, empregou-se a rede local do Instituto de Engenharia Biomédica (IEB) da Universidade Federal de Santa Catarina (USFC).

O segundo cenário utiliza a rede local da UFSC, através de conexão realizada entre dois laboratórios da instituição, o IEB-UFSC e o Grupo de Concepção e Análise de Dispositivos Eletromagnéticos (GRUCAD). Simulou-se neste cenário a transmissão do ECG entre dois hospitais localizados em uma mesma cidade.

O terceiro cenário realiza a simulação da monitoração de um paciente localizado em sua residência, numa conexão entre um hospital da região e sua casa. Utilizaram-se a rede do IEB-UFSC e uma rede proprietária de Florianópolis.

O quarto cenário simula o acompanhamento do ECG de paciente internado em hospital da cidade de Florianópolis por especialistas situados em hospital na

cidade de São Paulo. Para esta simulação, utilizaram-se a rede do IEB e uma rede proprietária localizada em São Paulo. O software, executado em um PC conectado a Internet, recebia os sinais transmitidos pelo módulo de comunicação conectado a Internet pela rede do IEB.

Para a realização dos ensaios, sinais previamente amostrados foram utilizados, dispensando o uso da captação de ECG em voluntário. Duas derivações (bipolar I e II) de ECG digitalizado (amostrado a 250 Hz com 8 *bits* de resolução) foram gravadas na memória interna do microcontrolador ADuC841 que integra o módulo de aquisição. Este envia o ECG armazenado ao módulo de comunicação pelo transmissor RF Bluetooth. Este último módulo estabelece a conexão e envia o ECG pela Internet aos terminais remotos.

Dentro de cada cenário, foram realizadas 10 simulações de 5 minutos. Destas 10 simulações, 5 foram realizadas sem a presença de tráfego, ou seja, durante a monitoração o software de visualização do ECG era o único aplicativo que gerava fluxo de dados entre o PC e a rede. Nas outras 5 simulações gerou-se tráfego adicional na rede através da transferência de arquivos (*download* e *upload*) entre a Internet e o PC.

A avaliação inicial do sistema totalizou 40 simulações, com um tempo de monitoração de 200 minutos.

Software de Avaliação

O *software* de avaliação, ilustrado na Figura 4.1, mostra a taxa de perdas, o número de pacotes esperados, o número de pacotes perdidos e o *jitter* correspondente a cada monitoração. O comportamento do *jitter* durante a transmissão do sinal é apresentado em forma gráfica pelo *software*.

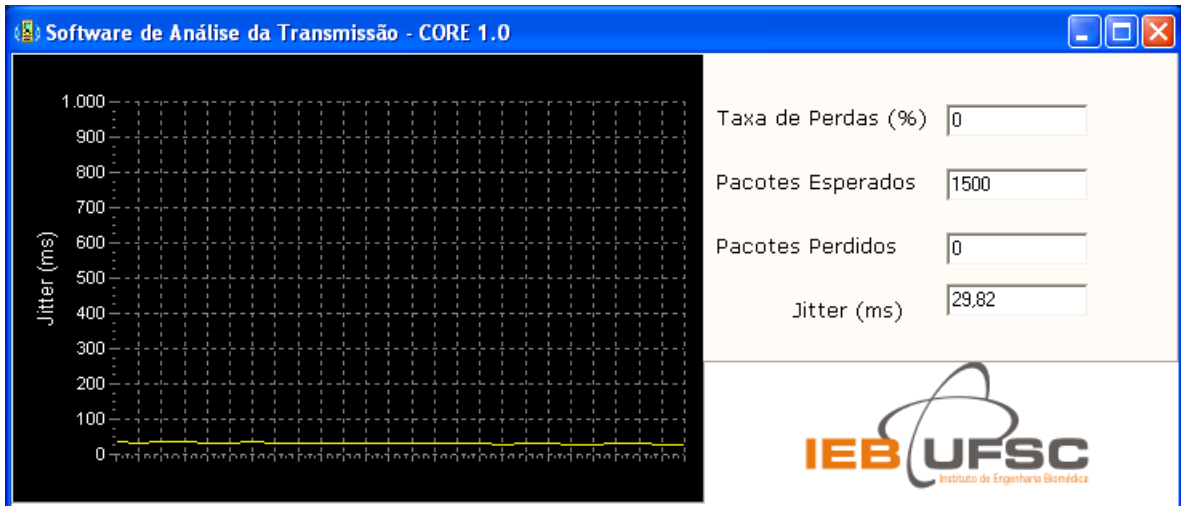


Figura 4.1 - Tela do software de avaliação do sistema.

O *jitter* é definido, segundo a RFC1889, como a diferença D correspondente ao espaço de tempo entre a chegada de um par de pacotes, comparado com os tempos neste par de pacotes no momento em que foram enviados. Logo, seja S_i o campo de *Timestamp* RTP do pacote i , e R_i o tempo de chegada em unidades de *timestamp* RTP do pacote i , então para dois pacotes i e j , D pode ser expresso como:

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (4.1)$$

Seja J definido como *jitter*, tem-se pela Eq. (4.2):

$$J = J + (|D(i-1, i)| - J) / 16 \quad (4.2)$$

O *jitter* é calculado continuamente. Quando cada pacote i for recebido da fonte, calcula-se a diferença D em relação ao pacote prévio ($i-1$) em ordem de chegada (não necessariamente em ordem de seqüência), de acordo com a Eq. (4.2). Esta equação possui um estimador empírico com parâmetro de ganho $1/16$, o que acarreta, segundo a RFC1889, numa boa redução da taxa de ruído (ocasionado por picos de *jitter*), enquanto mantém uma taxa razoável de convergência.

4.2 CENÁRIO 1

Para este teste, o módulo de comunicação foi conectado à rede local do IEB assumindo um endereço IP da *Ethernet*. Após estabelecer comunicação com o módulo, o computador do laboratório conectado à rede local realiza a monitoração dos sinais de ECG enviados. A rede do IEB disponibiliza, para o fluxo de dados local, largura de banda de 10 Mbps. O diagrama da simulação é apresentada na Figura 4.2.

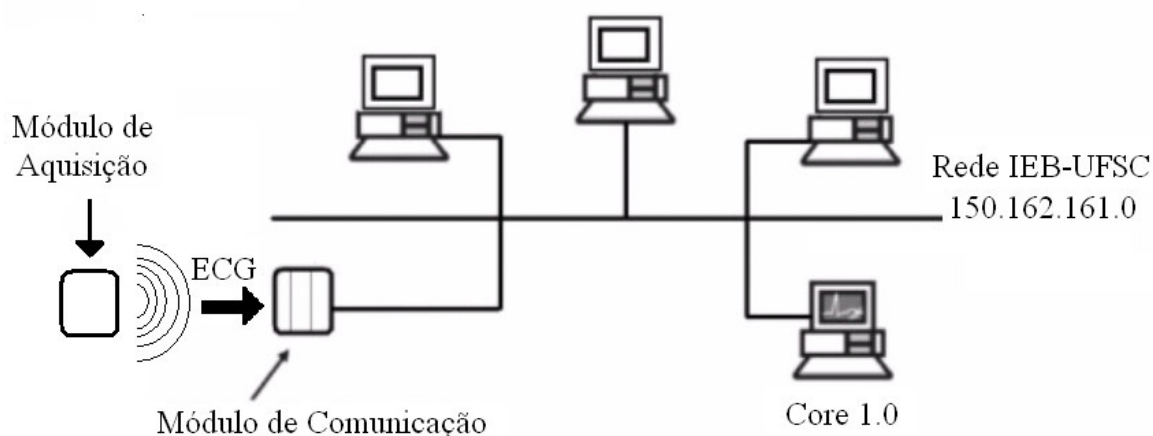


Figura 4.2 - Cenário 1: Rede Local

Através deste cenário procura-se analisar o comportamento do sistema utilizando uma LAN.

4.2.1 Resultados: Cenário 1

Como mostrado anteriormente (Seção 3.2.2), a largura de banda utilizada pelo módulo de comunicação para a transmissão de um sinal de ECG é de 3,92 Kbps. Sendo a largura de banda disponível para a rede local do laboratório de 10 Mbps, a monitoração do sinal ocupa somente 0,04% do total.

Pode-se observar, através da Tabela 4.1, que durante as cinco primeiras monitorações, nenhum pacote foi perdido ou corrompido. Este resultado já era

esperado, tendo em vista a baixa taxa de utilização da rede, incapaz de gerar algum tipo de congestionamento local.

Tabela 4.1 – Cenário 1: Sem tráfego adicional

Cenário 1: Sem fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	1	0	28,75
	2	0	30,85
	3	0	25,30
	4	0	28,90
	5	0	26,56

O gráfico da Figura 4.3 mostra comportamento do *jitter* durante o primeiro teste. Observa-se que não houve grandes variações no *jitter*. Isto é, durante o monitoramento, as variações do tempo que cada pacote levou para chegar ao computador receptor foram pequenas. Porém, a média do atraso existente entre dois pacotes, durante a comunicação, foi de 28,75 ms. Este valor indica que, no momento da monitoração, a rede estava com pouco tráfego de informações.

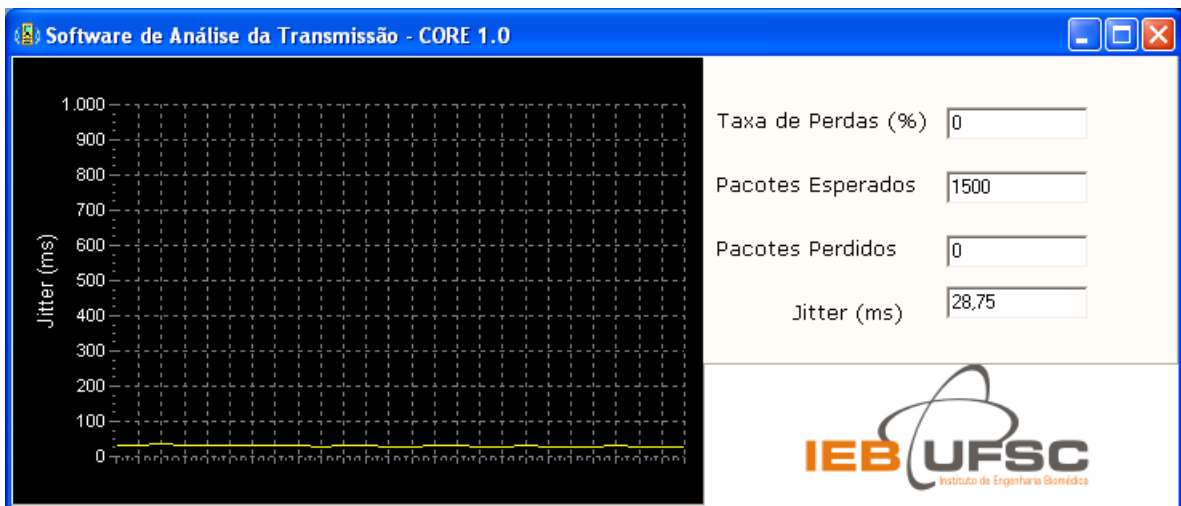


Figura 4.3 – Tela do software de avaliação do sistema: Cenário 1 – Teste 1

Em outras cinco simulações, gerou-se tráfego na rede, através de *downloads* e *uploads* de arquivos, numa taxa de 100 kbps e 200 kbps, respectivamente. A Tabela 4.2 mostra os resultados.

Tabela 4.2 – Cenário 1: Tráfego adicional gerado

Cenário 1: Com fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	6	0	47,63
	7	0	50,62
	8	0	49,94
	9	0	52,63
	10	0	51,65

Verifica-se que, com a intensificação do tráfego de dados na rede, houve um aumento no *jitter* de aproximadamente 20 ms. Ainda assim, não ocorreram perdas.

A Figura 4.4 demonstra o comportamento do sistema durante o teste de número 8.

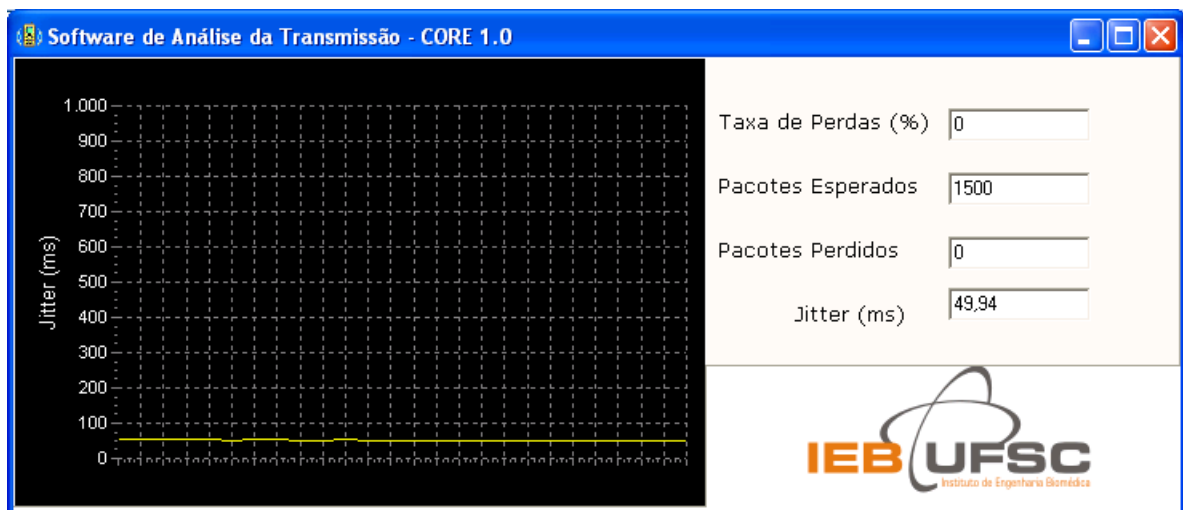


Figura 4.4 - Tela do software de avaliação do sistema: Cenário 1 com tráfego – Teste 8

Pode-se verificar, pelo gráfico da figura acima, a estabilidade do *jitter* na transmissão do ECG durante o teste número 8, o que indica a ausência de erros na

transmissão. A presença de picos no traçado do *jitter* pode indicar a perda de pacotes, já que estes picos estão relacionados com o aumento do atraso dos pacotes. Este aumento é causado pela presença de tráfego na rede.

4.3 CENÁRIO 2

Neste cenário, o módulo de transmissão estava conectado à Internet através da rede local do IEB.

O *software* de monitoramento foi executado por computador conectado à Internet na rede local do GRUCAD. As duas redes fazem parte da rede interna da UFSC que interliga os centros de ensino da universidade. A rede que interliga os dois laboratórios possui largura de banda de 155 Mbps. Um diagrama do cenário é mostrado na Figura 4.5

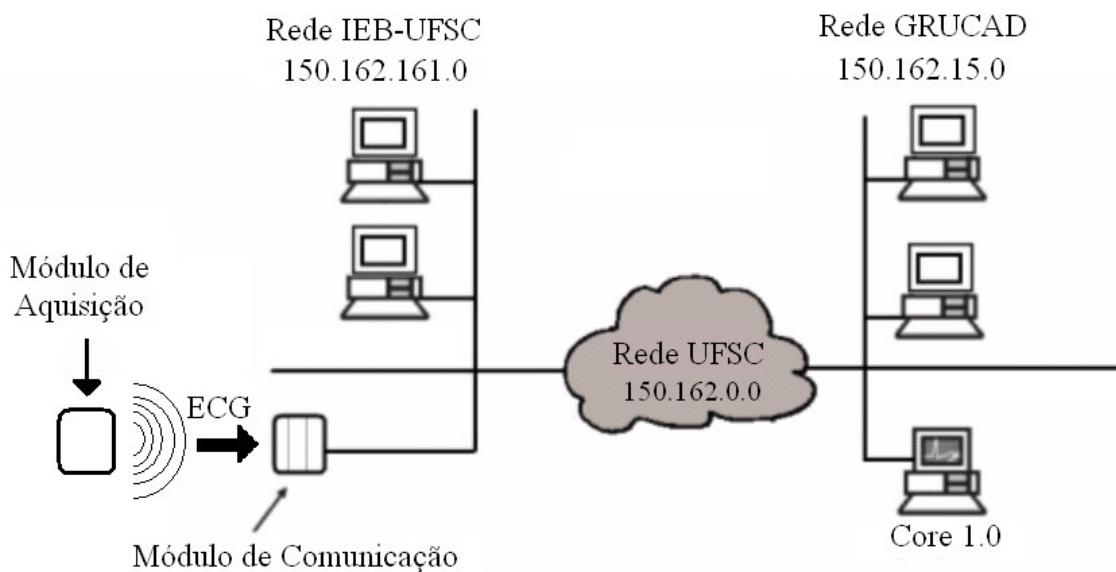


Figura 4.5 - Interconexão das redes do IEB e GRUCAD através da rede UFSC.

As faixas de endereço IP de cada laboratório são informadas na representação do cenário da figura acima.

Objetivou-se simular, através desta disposição, a transmissão do ECG entre hospitais interligados por uma rede metropolitana (*Metropolitan Area Network - MAN*). Seria o cenário equivalente à monitoração cardíaca de paciente em posto de saúde por especialista que poderá assim, dar suporte aos diversos postos de saúde de uma cidade.

4.3.1 Resultados: Cenário 2

A Tabela 4.3 apresenta os resultados obtidos nos cinco testes em que o aplicativo de monitoração era o único *software* que gerava fluxo de dados entre a rede e o PC.

Tabela 4.3 – Cenário 2: Sem tráfego adicional

Cenário 2 Sem fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	1	0	32,51
	2	0	29,82
	3	0	28,63
	4	0	27,63
	5	0	30,74

Verifica-se que os valores de *jitter* dos cinco testes permaneceram dentro de uma faixa máxima de variação de 4 ms. Observa-se esta estabilidade de resposta da rede pela Figura 4.6 , através comportamento do jitter durante o Teste 2.

Neste cenário, como no cenário anterior, a taxa de perdas permaneceu nula, ou seja, dos 1500 pacotes do sinal ECG esperados, nenhum fora perdido ou corrompido. Isto se deve ao fato da rede UFSC apresentar grande largura de banda, contribuindo para que nenhum pacote fosse perdido na monitoração.

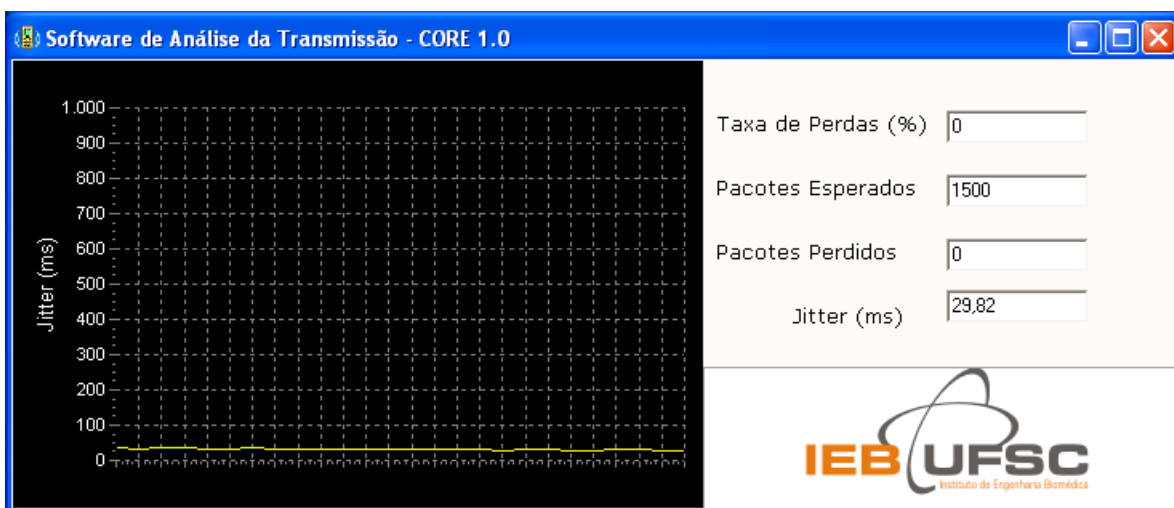


Figura 4.6 - Tela do software de avaliação do sistema: Cenário 2 sem tráfego - Teste 2

Para as simulações com fluxo de dados adicional, gerou-se tráfego na rede, através de *downloads* e *uploads* de arquivos, numa taxa média de 100 kbps e 200 kbps, respectivamente. A Tabela 4.4 mostra os resultados.

Tabela 4.4 – Cenário 2: Tráfego adicional gerado

Cenário 2 Com fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	6	0	61,69
	7	0	58,65
	8	0	62,98
	9	0	65,12
	10	0	59,36

Verifica-se um aumento médio de 30 ms no jitter, quando comparado ao jitter médio dos cinco testes sem o fluxo de dados adicional. Mesmo com o aumento do atraso entre os pacotes, não ocorreram perdas. O ECG mostrado na tela do PC não apresentou descontinuidades, oscilações na imagem (*flickers*) ou travamentos durante os testes.

A Figura 4.7 ilustra os dados de análise da transmissão do ECG durante o Teste 6.

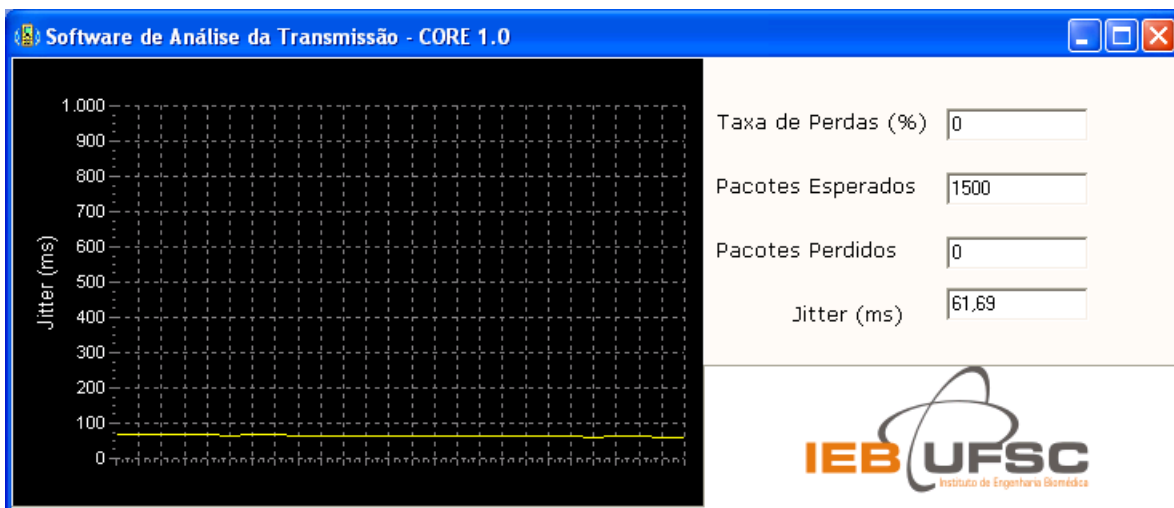


Figura 4.7 - Tela do software de avaliação do sistema: Cenário 2 com tráfego – Teste 6

Na figura acima, observa-se *jitter* de 63,69 ms durante o Teste 6. Percebe-se pelo gráfico que a linha que representa o *jitter* é praticamente constante, ou seja, não apresenta variações, o que indica a ausência de erros na transmissão do ECG.

4.4 CENÁRIO 3

Neste teste, o *software* monitor foi executado em PC situado em residência de Florianópolis. O microcomputador dispunha de uma conexão com a Internet com largura de banda de 256 kbps. Os dados da rede local do condomínio foram enviados por rádio frequência ao provedor de acesso à Internet. O módulo de transmissão estava conectado a Internet no IEB. Uma ilustração do Cenário 3 é apresentada na Figura 4.8.

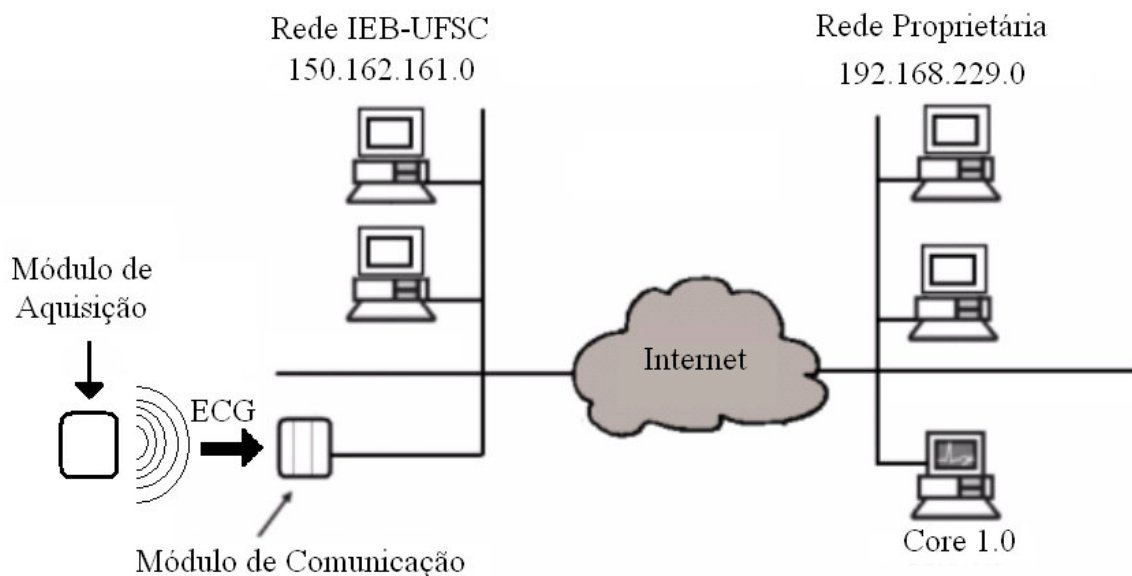


Figura 4.8- Comunicação pela Internet.

Observa-se pela figura acima o *software* Core 1.0 executado em um PC conectado a uma rede proprietária de Florianópolis com conexão à Internet. Neste cenário, objetivou-se emular a monitoração de um paciente em seu lar, sendo seus dados enviados a rede de um hospital da cidade.

4.4.1 Resultados: Cenário 3

A Tabela 4.5 mostra os resultados referentes à simulação deste cenário. Nos cinco testes, todos pacotes transmitidos pelo módulo foram recebidos pelo *software*. O *jitter* não apresentou valores dispare, todos valores permaneceram dentro da faixa de 50 +/- 5 ms. A taxa de perdas foi nula em todos os testes. Este fato é justificado pela baixa vazão gerada pelo módulo de comunicação, onde, para este cenário, a monitoração do sinal de ECG ocupa apenas 1,5% da largura de banda total (256kbps).

Tabela 4.5 – Cenário 3: Sem fluxo adicional

Cenário 3 Sem fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	1	0	48,63
	2	0	49,52
	3	0	51,87
	4	0	52,13
	5	0	45,66

Nos cinco testes houve ausência de variações no *jitter*, exemplificada no gráfico da Figura 4.9, o que indica a ausência de congestionamento nas redes utilizadas pelo sistema.

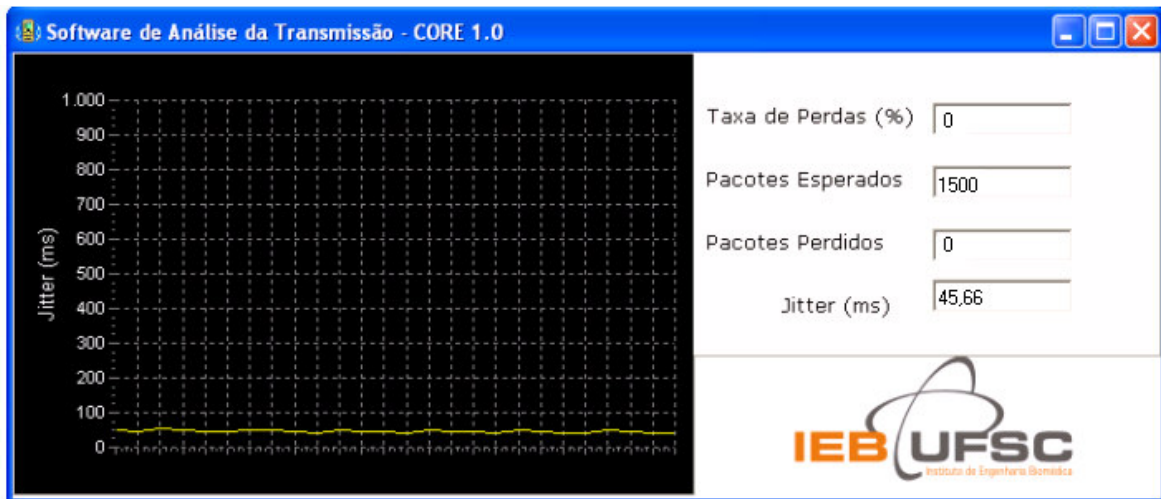


Figura 4.9 - Tela do software de avaliação do sistema: Cenário 3 sem tráfego - Teste 5

Para os testes com presença de fluxo de dados adicional, foi gerado tráfego por meio de transferência de arquivos do PC à uma taxa de 50 kbps para *download* e 50 kbps para *upload*. Observa-se pela Tabela 4.6 que os valores do *jitter* aumentaram em média 50 ms. Isto se deve ao crescimento da vazão gerada pelo computador através da transferência de arquivos. Com a inserção deste fluxo, a conexão entre o PC e a Internet ocupa 40,6% da largura de banda fornecida. Apesar do aumento do *jitter*, o sinal do ECG mostrado na tela do software não

apresentou travamentos ou descontinuidades. Em todos testes não ocorreram perdas de pacotes durante as transmissões do módulo de comunicação para o PC.

Tabela 4.6 – Cenário 3: Fluxo adicional gerado

Cenário 3 Com fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	6	0	98,32
	7	0	103,77
	8	0	102,65
	9	0	100,87
	10	0	95,63

A Figura 4.10 ilustra o aumento e a ausência de variações no *jitter*.

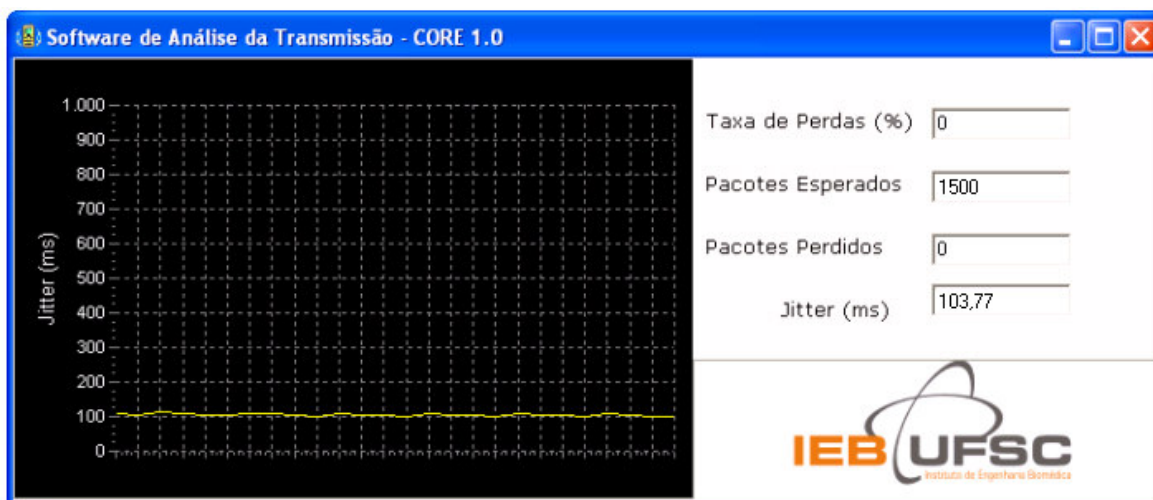


Figura 4.10 - Tela do software de avaliação do sistema: Cenário 3 com tráfego - Teste 7

O gráfico da Figura 4.10 mostra a linha que representa o *jitter* com pequenas variações em torno de 100 ms. Estas pequenas oscilações correspondem à variação do atraso entre os pacotes de ECG. Verifica-se também a ausência de perdas durante este teste, onde os 1500 pacotes enviados pelo módulo de comunicação foram recebidos pelo computador.

4.5 CENÁRIO 4

Visa simular a comunicação entre dois hospitais de diferentes estados. O módulo de transmissão localizado no IEB (Florianópolis) comunicava-se com o software monitor em execução na cidade de São Paulo. A Figura 4.11 mostra a estrutura de comunicação do cenário em questão.

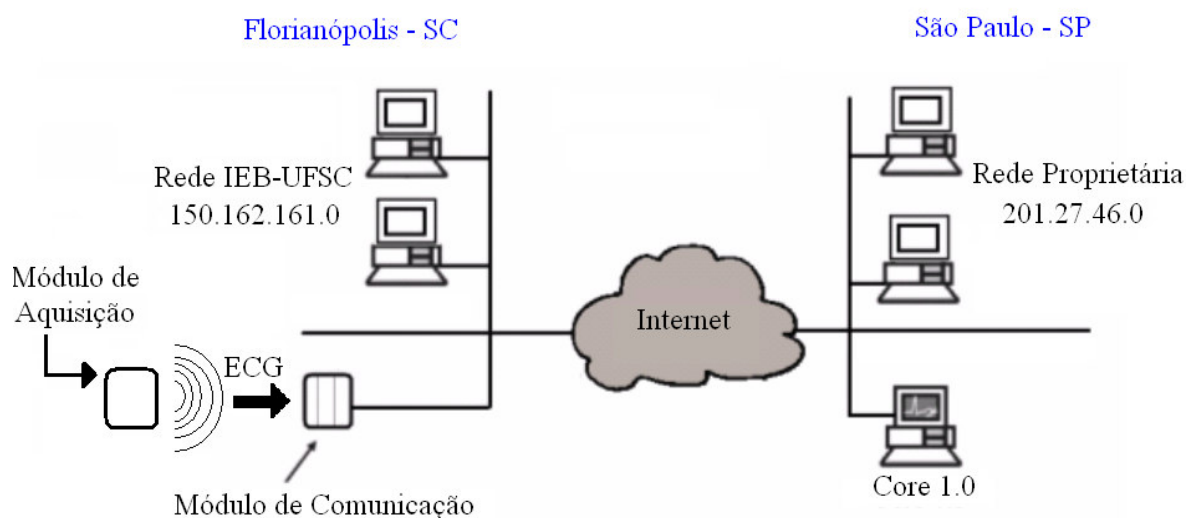


Figura 4.11 - Comunicação pela Internet entre São Paulo e Florianópolis

O *software*, executado em um PC conectado à Internet por meio de rede proprietária, dispunha de largura de banda de 500 Kbps para a recepção do ECG. A distância física entre os extremos da comunicação contribuiu para que a performance do canal de comunicação e o comportamento do sistema sejam avaliados.

4.5.1 Resultados: Cenário 4

Verifica-se pela Tabela 4.7 a média de 57,18 ms no *jitter* dos cinco testes e taxa de perdas nula.

Tabela 4.7– Cenário 4: Sem fluxo adicional

Cenário 4 Sem fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	1	0	55,64
	2	0	59,32
	3	0	55,36
	4	0	54,75
	5	0	60,85

Optou-se pela transmissão do eletrocardiograma entre duas capitais pelo fato destas estarem situadas em redes que possuem um grande fluxo de dados. Como consequência disto, observou-se pequenas variações do *jitter* (gráfico da Figura 4.12) durante as monitorações, porém estas não produziram travamento na visualização do sinal de ECG.

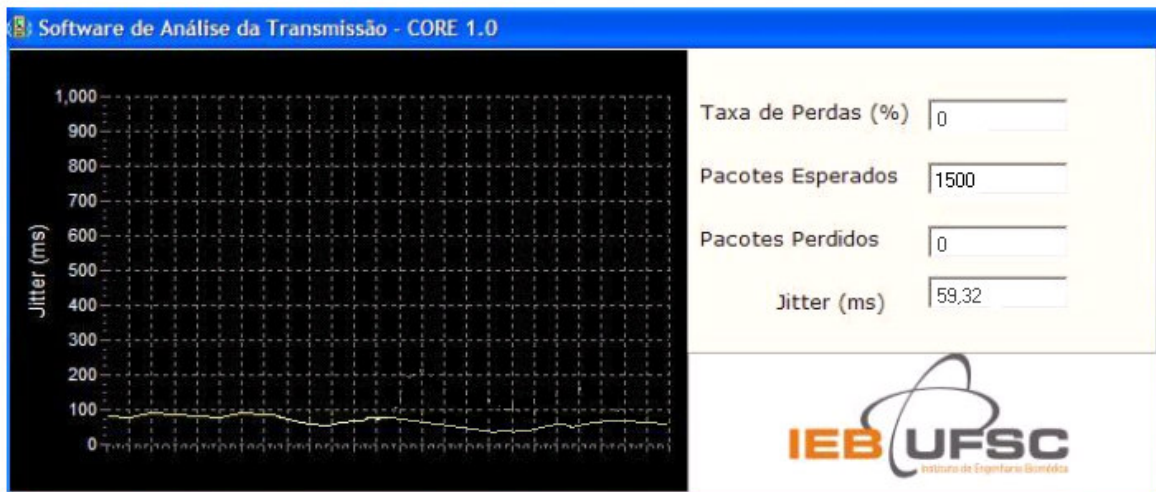


Figura 4.12 - Tela do software de avaliação do sistema: Cenário 4 sem tráfego - Teste 2

Nos testes seguintes, gerou-se tráfego de dados através de *download* de arquivos a um taxa de 100 kbps. Observa-se, pela Tabela 4.8, o aumento do *jitter* em relação aos testes sem o fluxo de dados adicional.

O Teste 9 apresentou taxa de perdas de 1,53 %. Esta perda corresponde a 2,3 segundos do sinal de ECG, de um total de 300 segundos de monitoração. Este teste foi o que apresentou maior *jitter* (130,54ms), possivelmente causado pela perda dos 23 pacotes. Os demais testes não tiveram nenhuma perda de pacotes.

Tabela 4.8 – Cenário 4: Fluxo adicional gerado

Cenário 4 Com fluxo adicional	Testes	Taxa de Perdas (%)	Jitter (ms)
	6	0	105,96
	7	0	103,65
	8	0	109,87
	9	1,53	130,54
	10	0	98,74

A Figura 4.13 ilustra os dados de análise da transmissão do ECG durante o Teste 9.

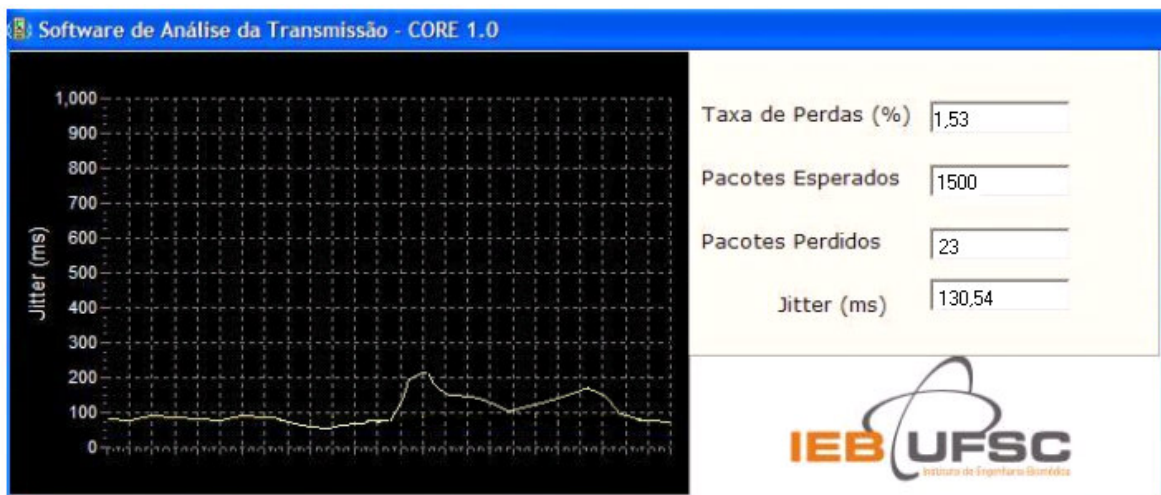


Figura 4.13 - Tela do software de avaliação do sistema: Cenário 4 com tráfego - Teste 9

No gráfico da figura acima, correspondente ao Teste 9, constata-se uma maior variação do jitter. Os dois picos observados na curva do *jitter* indicam o momento na monitoração em que houve a perda dos pacotes. Porém, não ocorreram travamentos na visualização do sinal. O sinal de ECG perdido foi representado com um nível DC nulo (Seção 3.3).

4.6 RESULTADO GERAL

A Tabela 4.9 mostra a largura de banda disponível entre os extremos da comunicação, de cada cenário simulado.

Tabela 4.9 - Largura de banda dos cenários

Cenário	Rede	Largura de banda
1	Ethernet (local)	10 Mbps
2	LAN UFSC	155 Mbps
3	Rede IEB - Rede Proprietária	256 Kbps
4	Rede IEB - Rede Proprietária	500 Kbps

Nos cenários 1 e 2, onde os dados transitaram através de redes locais com grande largura de banda, as simulações apresentaram baixos valores de *jitter* (abaixo de 30 ms) e nenhuma perda.

No terceiro cenário, a largura de banda disponibilizada para o acesso do PC à Internet foi suficiente para que não houvesse perdas de pacotes. O *jitter* aumentou quando comparado aos cenários anteriores, principalmente nos testes em que tráfego de dados adicional foi gerado na rede.

O cenário 4 foi o que apresentou os maiores valores de perdas e *jitter*. Pelo fato dos pacotes de informação seguirem os caminhos estipulados pelos roteadores presentes na Internet, não se tem o controle do ambiente dos testes. Sendo assim, não se podem afirmar quais fatores contribuíram para o quadro deste quarto cenário. Possivelmente, o grande fluxo de dados e as diferentes redes entre os extremos da comunicação (Florianópolis - São Paulo) foram responsáveis pelo aumento do *jitter* e ocorrência de perdas de pacotes.

A média do *jitter* e a taxa de perdas máxima dos 10 testes realizados em cada cenário são apresentados na Tabela 4.10.

Tabela 4.10 - Média do *jitter* e taxa de perdas máxima

Simulações realizadas sem fluxo de dados adicional	Taxa de perdas máxima (%)	<i>Jitter</i> médio (ms)
Cenário 1	0	28,07
Cenário 2	0	29,87
Cenário 3	0	49,56
Cenário 4	0	57,75
Simulações realizadas com fluxo de dados adicional	Taxa de perdas máxima (%)	<i>Jitter</i> médio (ms)
Cenário 1	0	50,49
Cenário 2	0	61,56
Cenário 3	0	100,25
Cenário 4	1,53	109,75

Em todos cenários, a visualização do sinal de ECG na tela do computador não foi comprometida, já que não apresentou travamentos nem sobreposição da imagem.

5 DISCUSSÃO E CONCLUSÃO

O Capítulo 5 traz as considerações finais sobre o sistema de monitoramento de ECG desenvolvido. Inicia-se com a discussão, e em seguida, a conclusão sobre o sistema são apresentadas. Por último, algumas propostas para trabalhos futuros são descritas.

5.1 DISCUSSÃO

As soluções adotadas no projeto possuem vantagens e desvantagens, que são discutidas no texto a seguir.

Uma característica que o sistema apresenta é a versatilidade com relação à comunicação com a rede. O módulo transmissor pode trabalhar com outros sinais bioelétricos. A única mudança, caso a frequência de amostragem utilizada seja modificada, é no tamanho do *buffer* de dados que será transmitido. Todo código para o empacotamento e sinalização não necessita ser modificado.

A adoção do protocolo IP deve-se ao fato deste ser amplamente difundido e de baixo custo de implantação, em relação a outras redes, tais como *frame-relay* e ATM. A Internet consolidou o uso das redes IP, o que facilitou a incorporação deste protocolo na conectividade de sistemas embarcados.

Um aspecto negativo da utilização do protocolo IP é que este não oferece recursos para garantia de Qualidade de Serviço (QoS) para aplicativos de tempo-real. Uma solução é a alocação de recursos na rede IP através do uso do protocolo RSVP (*Resource Reservation Protocol*) ou o uso de métodos como o *DiffServ* (Serviços Diferenciados). No entanto, estas soluções não são largamente utilizadas em grandes redes, como a Internet, sendo sua utilização limitada a redes corporativas ou regionais.

Redes que utilizam os protocolos *frame-relay* ou ATM oferecem um suporte melhor para este tipo de aplicação. Contudo, a falta de flexibilidade em conectividade e o alto custo de implantação destas redes descartam o uso destes protocolos para fins de transmissão de ECG em tempo-real.

Os recursos básicos para o monitoramento em tempo-real foram implementados com o protocolo RTP/RTCP. Este protocolo possibilita o controle de fluxo e a identificação de congestionamentos.

A largura de banda exigida pelo sistema é bastante pequena em relação à oferecida pelas redes IP atuais, motivo pelo qual não se utiliza nenhum algoritmo de compactação no *firmware*.

Nenhum esquema de segurança para a transmissão dos dados do paciente foi implementado. A utilização de algoritmos de criptografia no nível RTP solucionaria este problema. A RFC1889 determina uma criptografia padrão, a DES/CBC (*Data Encryption Standard/Cipher Block Chaining*). Informações para a implementação do algoritmo estão descritos na RFC1423 [24].

Os protocolos utilizados não possuem mecanismos de retransmissão de pacotes perdidos. Para minimizar a perda de dados, poder-se-ia transmitir dados com redundância de conteúdo. Desta maneira, o pacote atual possuiria amostras do segmento do sinal atual e algumas amostras do pacote anterior para serem apresentadas caso o pacote anterior tenha se extraviado.

5.2 CONCLUSÃO

O sistema proposto para a transmissão de ECG em tempo-real foi desenvolvido com sucesso. O módulo de transmissão estabelece comunicação com o aplicativo de telemetria e envia o ECG para visualização em tempo-real na tela do PC.

O *jitter* causado pelo tráfego de dados na Internet não chegou a degradar a qualidade da transmissão, visto que, em todas as simulações realizadas, o sinal mostrou-se inteligível. Ocorreram eventuais perdas de pacotes, porém sempre numa taxa inferior a 2%.

Apesar do sistema de transmissão proposto neste trabalho realizar a aquisição e transmissão do sinal de ECG em tempo real sem que haja perdas e atrasos significativos, como mostrado nos resultados, é importante comentar que o sistema não está validado. Não foram realizados estudos e simulações em que o especialista estivesse presente. No trabalho desenvolvido, apenas um cenário simulado apresentou perdas (2,3 segundos de ECG de um total de 300 segundos de monitoração). Embora esta perda seja pequena, somente o especialista pode determinar se a não visualização de uma pequena parte do sinal de ECG afetará uma pré-análise da eventual disfunção cardíaca do paciente.

Sistemas de telemedicina contribuem fortemente para a melhoria da qualidade de vida e humanizam o atendimento dos pacientes. Por exemplo, pacientes que habitam pequenas cidades, onde o número de cardiologistas é limitado, podem ser atendidos e diagnosticados por especialistas de outras localidades [26].

Com a crescente inclusão da Internet nas residências, sistemas de *homecare* podem ser implantados com menores custos, aproveitando a infra-estrutura existente. O sistema criado possibilita uma redução de custos ainda maior, visto que este dispensa o uso de um microcomputador na transmissão dos dados pela Internet. Com isto, camadas de menor poder aquisitivo da população podem contemplar o uso desta tecnologia.

O sistema de transmissão desenvolvido pode ser também utilizado no monitoramento de pacientes cardíacos em outros ambientes, como hospitais, enfermarias e postos de saúde.

5.3 TRABALHOS FUTUROS

Uma proposta para trabalho futuro seria a adição de rotinas ao *software* de recepção sendo executado no PC para realizar o diagnóstico automático de algumas arritmias. Opcionalmente, algoritmo presente no *firmware* poderia enviar o ECG pela Internet apenas quando uma variação da frequência cardíaca for detectada.

Para ampliar as possibilidades de aplicação do sistema, pode-se adicionar memória externa ao módulo transmissor para armazenar o sinal de ECG amostrado. Este seria enviado a um computador remoto quando o mesmo fosse conectado à rede, funcionando deste modo como um *holter*. Neste caso, o protocolo de transporte utilizado seria o TCP.

No caso da utilização de uma maior frequência de amostragem na aquisição do ECG ou a transmissão de mais derivações, técnicas de compactação [27][28][29] podem ser utilizadas para reduzir a largura de banda ocupada pelo sistema.

Outro aspecto a ser avaliado é a utilização de redundância na transmissão dos pacotes, visando reduzir a taxa de perdas do sistema.

Por fim, sugere-se implementar chave de criptografia no protocolo para assegurar a confidencialidade dos dados na comunicação através da Internet.

REFERÊNCIAS

- [1] FINKELSTIN, S.M.; SPEEDIE, S.; HOFF, M.; DEMIRIS, G. **Tele-homecare: telemedicine in home health care.** In: 21st Annual Conf. and the 1999 Annual Fall Meeting of the Biomedical Engineering. Proceedings of the First Joint. Volume 2, 13-16 Oct. 1999 Page(s):681 vol.2.
- [2] H. Lee, S. Park, and E. Woo. **Remote patient monitoring through World-Wide Web.** In *Proc. XIX Annu. IEEE EMBS Conf.*, Chicago, IL, 1997, pp. 928-931.
- [3] KHOOR, S.; NIERBEL, K.; FUJEDI, K.; KAIL, E. **Telemedicine ECG-telemetry with Bluetooth technology.** In: *Computers in Cardiology 2001*, 23-26 Sept. 2001 Page(s): 585 - 588.
- [4] SAKAMOTO, T.; WEI, Daming. **Implementation of mobile ECG telemonitor using real-time transport protocol (RTP).** In: *IEEE EMBS Asian-Pacific Conference on Biomedical Engineering*. Oct. 2003 Page(s): 32 - 33.
- [5] LAMBERTINI, F.; DEMARTINI, C. **Low-cost home monitoring using a Java-based embedded computer.** In: *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine*, 2003. 24-26 April 2003 Page(s): 342 -345.
- [6] LEE, Ho Sung; PARK, Seung Hun; WOO, Eung Je. **Remote Patient Monitoring Service through World-Wide Web.** In: *Proceedings - 19th International Conference - IEEE/EMBS* Oct. 30 - Nov. 2, 1997 Chicago, IL. USA

- [7] HERNANDEZ, A.I., et al. **Real-time ECG transmission via Internet for nonclinical applications.** In: IEEE Transactions on Information Technology in Biomedicine. Volume 5, Issue 3, Sept. 2001 Page(s):253 - 257.
- [8] PIZARRO, Paulo Jorge Câmara;. **MonitorIP: monitoramento de sinais vitais através de um Rede IP.** Florianópolis, 2003.Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica - Universidade Federal de Santa Catarina.
- [9] MARTINCOSKI, Daniel Henrique. **Sistema para telemetria de eletrocardiograma utilizando tecnologia Bluetooth.** Florianópolis, 2003. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica - Universidade Federal de Santa Catarina.
- [10] GUYTON, Arthur C. Hall, John E. **Tratado de Fisiologia Médica.** 10 ed. Rio de Janeiro: Guanabara, 2002.
- [11] RIBEIRO, R. L.; REIS, P. F.; BOMFIM, A. DE S.; BARBOSA, E. C.; BOGHOSSIAN, S. H. C.; VELOSO, H. H.; GINEFRA, P.; **Antigos e Novos Conceitos sobre a Onda U do Eletrocardiograma.** *Revista da SOCERJ*, Rio de Janeiro, v.17, n.3, p.192-194, jul./ago./set. 2004.
- [12] TOMPKINS, W. J.; 1995. **Biomedical Digital Signal Processing.** 1. ed. Upper Saddle River, New Jersey : Prentice Hall Ptr.
- [13] FERGUSON, Paul; HUSTON, Geoff. **Quality of Service: Delivering QoS on the Internet and in Corporate Networks.** 1 ed. New York: John Wiley & Sons, 1998.

- [14] HIMONAS, S.D. GELMAN, A.D. **Performance evaluation of a real-time transport protocol with deterministic and statistical QoS guarantees.** In: ICC 97 Montreal, 'Towards the Knowledge Millennium'. 1997 IEEE International Conference. Date: 8-12 June 1997. Volume: 1 Page(s): 256 - 260 vol.1.
- [15] KUROSE, James F.; ROSS, Keith W. **Computer networking: a top-down approach featuring the internet.** Boston: Addison-Wesley, 2001. 712p.
- [16] SOCOLOFSKY, T.; KALE, C. **RFC 1180 - A TCP/IP Tutorial.** Internet Engineering Task Force, 1991.
- [17] TANENBAUM, Andrew S. **Computer networks.** 3 rd ed Upper Saddle River: Prentice-Hall PTR, c1996.813p.
- [18] STEVENS, W. Richard. **TCP/IP Illustrated - The Protocols.** Vol. 1, Addison-Wesley,1994.
- [19] POSTEL,J. **RFC 768 - User Datagram Protocol.** Internet Engineering Task Force, 1980.
- [20] JACOBSON,V, et al. **RFC 1889 - RTP: A Transport Protocol for Real-Time Application.** Internet Enginnering Task Force, 1996.
- [21] BONHO, Samir; KOLM, Daniel; BAGGIO, João F. R; MORAES, Raimés. **Módulo micro-montrolado para comunicação de sinais de ECG pela Internet.** IV Congresso IBERDISCAP: Tecnologias de apoio a portadores de deficiência. Fevereiro,2006. Vitória - ES, Brasil.
- [22] BRAY, J.; STURMAN, C. F; 2001. **Bluetooth: Connect without Cables.** 1. ed. Upper Saddle River, New Jersey : Prentice Hall Ptr.

- [23] SCHULZRINNE, H; CASNER, S. **RFC 3551 - RTP Profile for Audio and Video Conferences with Minimal Control**. Internet Engineering Task Force, 2003.
- [24] MENARD, A, et all. **Dual microprocessor system for cardiovascular data acquisition, processing and recording**. In: Proceedings of IEE International Conference of Industrial Elect. Contr. Intrumentation. 1981. Page(s): 64 - 69.
- [25] BALENSON, D. **RFC 1423 - Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers**. Internet Engineering Task Force, 1993.
- [26] PARK, Seung-Hun, et al. **Real-time monitoring of patients on remote sites**. In: Proceedings of the 20th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society. Volume 3, 29 Oct.-1 Nov. 1998 Page(s): 1321 - 1325 vol.3
- [27] KYOSO, M., UCHIYAMA, A., **ECG Data Reduction Meted for Medical Telemetry System**, Congresso Internacional de Eng. Biomédica. Chicago, 2000.
- [28] ISHIJIMA, M., SHIN S.B., HOSTETTER G.H., SKLANSKY J., **Scan along polygonal approximation for data compression of electrograms**. IEEE Trans. Biomed. Eng. vol. 30, pp. 723-729, Nov. 1983.
- [29] ABENSTEIN, J.P., TOMPKINS, W.J., **A new data-reduction algorithm for real-time ECG analysis**. IEEE Trans. Biomed. Eng. vol. 29, pp. 43-48, Jan. 1982.
- [30] OBJECT MANAGEMENT GROUP (OMG). **UML Resource Page**. Disponível em: < <http://www.uml.org/>>. Acesso em: 20 abril 2006.

- [31] WEBSTER, John G. **Medical Instrumentation: Application and Design**. 3 ed. New York: John Wiley & Sons, 1998.
- [32] SPECIALSKI, Elizabeth. **Arquiteturas de Redes de Computadores**. Curso de Pós-graduação em Ciência da Computação. Florianópolis – SC, 2000. Universidade Federal de Santa Catarina.
- [33] COMER, D., **Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture**, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [34] COMER, D. and Stevens, D., **Internetworking with TCP/IP Volume II: Design, Implementation, and Internals**, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [35] COMER, D. and Stevens, D., **Internetworking with TCP/IP Volume III: Client-Server Programming and Applications**, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [36] YU, G.B.; CHAO, I.P.; SU, J.L.; **The research of wireless application protocol applied in the home care system**. Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE , Volume: 4 , 25-28 Oct. 2001
- [37] LIESENBORGS, Jori. **Voice over IP in networked virtual environments**. Maastricht, 2000. Thesis - School for Knowledge Technology , Maastricht University.
- [38] MALMIVUO, J.; PLONSEY, R.; **Bioelectromagnetism. Principles and Applications of Bioelectric and Biomagnetic Fields**. 1 Ed. New York : Oxford University Press, 1995.

[39] REYNOLDS, J; POSTEL,J. **RFC 1700 - Assigned numbers.** Internet Engineering Task Force, 1994.