Rede Local Sem Fio: Considerações sobre o Projeto de uma Plataforma de Acesso

Janeiro - 1995

Autor
Nélio Antônio Teodoro de Resende

Orientador
Prof. Dr. João Marcos Travassos Romano
Departamento de Comunicações - FEE/UNICAMP

Co-orientador
Prof. Dr. Michel Daoud Yacoub
Departamento de Comunicações - FEE/UNICAMP

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica
Departamento de Comunicações

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas - FEE/UNICAMP, como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

Este exemplar corresponde à redação final da tese defendida por Nelio Antonio Teodoro de Resende e aprovada pela Comissão Julgadora em 06.01.95.

Orientador

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

R339r

Resende, Nélio Antônio Teodoro de
Rede local sem fio: considerações sobre o projeto de uma plataforma de acesso / Nélio Antônio Teodoro de Resende.-- Campinas, SP: [s.n.], 1995.

Orientador: João Marcos Travassos Romano, Michel Daoud Yacoub
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica.

1.Redes locais de computadores. I. Romano, João Marcos Travassos. II. Yacoub, Michel Daoud.
III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica. IV. Título.

Sumário

Este trabalho aborda a implementação de uma plataforma de acesso para rede sem local sem fio. Os conceitos envolvidos, bem como os parâmetros importantes a uma rede local sem fio são apresentados e discutidos. Protocolos de acesso ao meio físico para este tipo de rede são analisados e, por fim, o protocolo proposto é apresentado detalhadamente. A partir das necessidades decorrentes da implementação deste protocolo, uma plataforma hardware é estabelecida e seus blocos funcionais, relacionados a rede sem fio, são analisados. Descreve-se a implementação do encriptador, do compressor de dados e também das interfaces com o Sistema de Processamento de Dados (interface PCMCIA) e com o Rádio. Assim, integrados, estes blocos funcionais constituem uma plataforma de acesso para redes sem fio.

Agradecimentos

Agradeço ao meu orientador Prof. João Marcos Travassos Romano pelo incentivo, acompanhamento, companheirismo e por acreditar que conseguiríamos chegar a conclusão deste trabalho.

Agradeço ao meu co-orientador Prof. Michel Daoud Yacoub, cuja percepção nos abriu caminho para este trabalho, pelo companheirismo e por nos acompanhar em inúmeras releituras em busca do melhor e melhor.

Aos amigos Marcelo e Marco, que ao me convidarem para trabalhar com redes sem fio, me abriram a oportunidade de desenvolver este trabalho.

Ao Paulo Cesar, Vitor e Amaury pelo companheirismo e por compor comigo a equipe cujo resultado do trabalho apresenta-se nesta Tese.

Ao Ronaldo, Ricardo Barbosa, Celso Brites, Bernardo, Marcelo Ferraz, José Afonso e Eliane pelo intercâmbio de conhecimento técnico durante a elaboração da Tese.

Ao Ernesto, Felipe e Ruth por, nos momentos de grandes dúvidas, me ajudarem no caminho a seguir.

À IBM Brasil pelo patrocínio e apoio e a UNICAMP pela auxílio as pesquisas sem os quais não seria possível o desenvolvimento deste trabalho.

A Deus, que tornou isso tudo possível.

Índice

Capítulo 1. Introdução 1
1.1 Redes Sem Fio - Considerações Iniciais 1
1.2 Tecnologia Acessível 1
1.3 Aplicações das Redes Sem Fio 2
1.4 Custo Como Motivação 2
1.5 Computadores Portáteis 3
1.6 Sumário dos Capítulos 3
1.7 Referências 4

Capítulo 2. Rede Sem Fio: Arquitetura e Parâmetros de Transmissão 5
2.1 Introdução 5
2.2 Topologia Básica 5
2.3 Tipos de Redes Sem Fio 6
2.4 Redes de Comunicação Via Rádio 7
2.4.1 Interferência de Seqüência Direta e Salto em Freqüência 7
2.5 Desempenho, Confiabilidade, Segurança e Salubridade 8
2.6 Compatibilidade entre Redes Sem Fio 9
2.7 Protocolos de Acesso ao Meio Físico 10
2.7.1 Protocolo CSMA/CA 10
2.7.2 Protocolo de Acesso Híbrido: Síncrono e Assíncrono 12
2.8 Arquitetura e Características da Rede Proposta 14
2.8.1 Protocolo de Acesso ao Meio Físico Proposto 14
2.9 Conclusões 15
2.10 Referências 15

Capítulo 3. Protocolo de Acesso ao Meio Físico Proposto 16
3.1 Introdução 16
3.2 As Fases A, B e C 16
3.2.1 Fase A (tráfego Ponto de Acesso - Estações Remotas) 16
3.2.2 Fase B (tráfego Estações Remotas - Ponto de Acesso) 17
3.2.3 Fase C (tráfego assíncrono) 18
3.3 Análise do Protocolo de Acesso Proposto 20
3.3.1 Conjecturas sobre Desempenho do Protocolo Acesso Proposto 20
3.3.2 Pontos Positivos do Protocolo Acesso Proposto 21
3.3.3 Pontos Negativos do Protocolo Acesso Proposto 21
3.4 Conclusões 21
3.5 Referências 22

Capítulo 4. Implementação da Plataforma em Hardware 23
4.1 Introdução 23
4.2 Blocos Funcionais 24
4.3 Fluxo de Dados 25
4.4 Conclusões 29
4.5 Referência 29

Capítulo 5. Encriptação e Autenticação 30
5.1 Encriptador 30
5.1.1 Introdução 30
5.1.2 Gerador de Padrões de Encriptação 31
5.1.3 Implementação 32
5.2 Autenticador 35

5.2.1 Introdução . . . 35
5.2.2 O Mecanismo de Autenticação . . . 36
5.2.3 Implementação . . . 37
5.2.4 Comparações com Métodos de Checagem de Mensagem e Assinatura . . . 37
5.3 Conclusões . . . 39
5.4 Referências . . . 39

Capítulo 6. Compressão de Dados . . . 41
6.1 Introdução . . . 41
6.2 Técnicas de Compressão Disponíveis . . . 41
6.2.1 Algoritmo LZ1 . . . 41
6.2.2 Algoritmo LZ2 . . . 42
6.3 Algoritmo LZ1 Proposto . . . 42
6.3.1 Algoritmo LZ1 Proposto: Compressão . . . 42
6.3.2 Algoritmo LZ1 Proposto: Descompressão . . . 43
6.3.3 Exemplo de Compressão Utilizando-se o Algoritmo LZ1 Proposto . . . 44
6.4 Implementação do Compressor-Descompressor de Dados . . . 47
6.4.1 Compressor . . . 47
6.4.2 Descompressor . . . 48
6.5 Resultados Atingidos pela Compressão . . . 49
6.6 Conclusões . . . 52
6.7 Referências . . . 52

Capítulo 7. Interface PCMCIA . . . 54
7.1 Introdução . . . 54
7.2 Características Físicas . . . 55
7.2.1 Dimensões . . . 55
7.2.2 Conectores . . . 56
7.2.3 Especificações Ambientais do Cartão . . . 56
7.3 Interface Elétrica . . . 57
7.3.1 Operação de um Cartão PCMCIA . . . 58
7.3.2 Descrição dos Sinais da Interface PCMCIA . . . 58
7.3.3 Registros da Interface . . . 65
7.3.4 Memória de Atributos . . . 66
7.4 Camadas de Software para um Cartão PCMCIA . . . 70
7.5 Consumo Racionalizado de Energia . . . 73
7.6 Conclusões . . . 74
7.7 Referência . . . 74

Capítulo 8. Sincronização entre Estações da Rede Sem Fio . . . 75
8.1 Introdução . . . 75
8.2 Sincronismo de Salto em Freqüência . . . 75
8.2.1 Preenchimento da Tabela de Salto em Freqüência . . . 76
8.2.2 Sincronização entre Ponto de Acesso e Estações Remotas . . . 77
8.3 Implementação . . . 79
8.4 Conclusões . . . 81
8.5 Referências . . . 81

Capítulo 9. Conclusão . . . 82
9.1 Resumo . . . 82
9.2 Conclusões e Contribuições . . . 83
9.3 Sugestões para Continuidade deste Trabalho . . . 84

Figuras

1. Topologia básica de uma rede sem fio. 6
2. Esboço dos espectros Seqüência Direta e Salto em Freqüência. 8
3. Exemplo do protocolo ALOHA puro. 11
4. Configuração de acesso direto entre estações. 13
5. Configuração de acesso hierárquico entre estações. 14
6. Estrutura do quadro de tempo no protocolo acesso ao meio físico. 17
7. Tipos de transmissão remota-remota. 20
8. Diagrama das entidades funcionais do sistema de comunicação de dados. 24
9. Interfaces da implementação hardware. 25
10. Diagrama de blocos da implementação hardware. 26
11. Fluxo de dados através do Controlador de Comunicação. 27
12. Organização dos dados durante a transmissão de mensagem. 28
13. Esquema de um Registro de Deslocamento com Realimentação Linear. 32
14. Implementação do circuito de encriptação. 33
15. Implementação do circuito de autenticação. 38
16. Exemplo de conteúdo do dicionário e saída do compressor. 46
17. Diagrama de blocos da entidade de compressão. 48
18. Bloco funcional do compressor para comparação de dados. 49
19. Diagrama funcional da entidade de descompressão. 50
20. Cartão PCMCIA tipo II. 56
21. Perfil de uma embalagem PCMCIA típica. 57
22. Interface PCMCIA (registros e Memória de Atributos). 67
23. Camadas de software e hardware descritas no padrão PCMCIA. 72
24. Quadros de tempo operando com freqüências e durações distintas. 76
25. Diagrama esquemático de um Ponto de Acesso e estação remota. 78
26. Seqüência de eventos para sincronização do rádio (método 1). 79
27. Seqüência de eventos para sincronização do rádio (metódo 2). 80

Tabelas

1. Códigos utilizados pelo LZ1 proposto. 44
2. Compressão da palavra RINTINTIN. 45
3. Mensagem recebida pelo descompressor. 46
4. Comparação entre implementações dos algoritmos LZ1 e LZ2 {6}. 50
5. Comparação entre implementações de LZ1 {6}. 51
6. Comparação da máxima vazão das implementações de LZ1 e LZ2. 51
7. Dimensões dos tipos de cartões PCMCIA. 55
8. Características ambientais dos cartões PCMCIA. 57
9. Possíveis estados de tensão na bateria do cartão PCMCIA. 59
10. Sinais da interface PCMCIA. 60
11. Leitura de Memória Comum. 63
12. Escrita de Memória Comum. 64
13. Leitura de Memória de Atributo. 64
14. Escrita de Memória de Atributo. 64
15. Leitura de I/O. 65
16. Escrita de I/O. 65
17. Formato geral das t-uplas. 66
18. T-uplas de Compatibilidade Básica: informações sobre o dispositivo. 68
19. T-uplas de Compatibilidade Básica: informações complementares. 68
20. T-uplas de Compatibilidade Básica: informações sobre o produto e identificadores JEDEC. 69
21. T-upla de Configuração. 70
22. T-upla da tabela de configuração. 71
23. Consumo típico de energia do adaptador para rede sem fio implementado. 73
24. Tabela de salto em freqüência. 76

Prefácio

A miniaturização de componentes e equipamentos é uma tendência verificada ao longo do processo da evolução tecnológica experimentada pelos setores de telecomunicação e informática. Aparelhos, que há poucas décadas, ocupavam um volume considerável, hoje podem ser colocados no bolso de um paletó ou num canto qualquer da mesa. Neste aspecto, os computadores têm especial destaque. Há algum tempo atrás, haviam sido lançados os "laptops" e com eles a novidade de se dispor de processamento e dados pessoais em qualquer lugar. Ainda há pouco, nos vimos às voltas com os "notebooks" e desfrutamos de um maior processamento, maior capacidade de armazenamento de dados, em ainda menor espaço. Hoje falamos nos "sub-notebooks", menores e mais práticos. Estes microcomputadores, cada vez mais portáteis, vão se integrando mais e mais na vida dos seus usuários.

A necessidade de uma rede de interligação entre estas máquinas, seja para partilhar recursos, seja para facilitar a comunicação entre usuários, é premente. Não há, nos dias de hoje, como evitar de se compartilhar recursos, programas, dados, trocar correspondências, enfim, de se trabalhar em conjunto. Usuários de microcomputadores portáteis, por sua vez, sentem esta mesma necessidade. Conexões, por outro lado, implica ligação física de cabos, restringindo a mobilidade do equipamento, característica marcante dos portáteis.

É possível conectar um "notebook" a uma rede local mas o atrativo da mobilidade se desfaz. Com a vida das pessoas cada vez mais próximas da informática, é de se esperar que a conectividade sem fio complemente os anseios dos usuários de computadores portáteis.

Assim, o desenvolvimento de redes sem fio é atrativo mas requer tecnologias elaboradas para sua implementação. Este trabalho foi feito a partir de vários outros trabalhos isolados, cuja aplicação resultou na plataforma proposta para redes sem fio. Ele tem por objetivo a apresentação dos conceitos utilizados para se implementar uma rede sem fio. A partir destes conceitos, são discutidos protocolos de comunicação e, por fim, uma plataforma hardware para redes local sem fio é apresentada.

Capítulo 1. Introdução

1.1 Redes Sem Fio - Considerações Iniciais

Nos últimos anos, tem havido um crescente interesse por sistemas de comunicação sem fio. Obviamente, a grande motivação para tal interesse é a capacitação tecnológica atingida atualmente que permite implementar redes sem fio a um custo inferior ao das redes convencionais. Outras motivações seriam a facilidade de instalar, manter e utilizar uma rede sem fio em comparação com as redes convencionais.

No entanto, compatibilidade é a palavra chave quando se trata de instalação, uso e manutenção. Para que a rede sem fio seja conectada a uma rede com fio já existente, é imprescindível que exista compatibilidade entre a interface lógica e física, funcionalidade e operação. A interface da rede sem fio deve se alojar dentro dos produtos de computação já existentes, tornando-se assim transparente para os sistemas operacionais de redes e para as aplicações de software. A rede sem fio deve estar em conformidade com os padrões de redes já existentes e, portanto, propiciar métodos automáticos de informação de problemas e controle de dispositivos da rede. Desta forma, o usuário terá que simplesmente "ligar e operar", o que, de fato, constitui um grande atrativo.

O requisito de alto desempenho significa mais do que simplesmente uma alta taxa de transmissão de bits e pequenos atrasos. O ambiente no qual operam as redes sem fio apresenta seus próprios desafios que, geralmente, não são encontrados nos sistemas de rede com fio. Em particular, os sistemas de rede sem fio devem ser capazes de compensar desvanecimentos intermitentes de sinal causados por obstáculos que se movem, tais como pessoas e portas, e interferências devido a múltiplos sinais contendo a mesma informação e chegando ao receptor via caminhos diferentes. Esses dois fenômenos podem causar a degradação em desempenho, chegando inclusive a resultar na perda de informação pelo usuário. Portanto, um sistema sem fio deve se adaptar continuamente às mudanças das instalações em que se encontra.

Outro aspecto que deve ser considerado para que um alto desempenho seja atingido é o que diz respeito à confiabilidade e à segurança. Prover uma taxa de transferência adequada e boas características de atraso é essencial, porém mais fundamental ainda é ter a rede sem fio disponível para uso por todo o tempo. Assim, a tecnologia deve fornecer uma plataforma estável que propicie uma degradação suave em caso de anomalia na rede. A segurança, por sua vez, relaciona-se com a privacidade das informações que trafegam na rede. Assim, uma rede sem fio deve ser ao menos tão segura quanto uma rede com fio em termos de inviolabilidade de informações.

1.2 Tecnologia Acessível

Hoje é evidente que as redes sem fio têm muito a progredir até atingir todo o seu potencial. O custo inicial das redes sem fio ainda é alto, as taxas de transmissão baixas e apenas em 1992 apareceram necessidades tais como conectividade a redes Token Ring e outras redes locais. Mas é interessante notar que muitos dos problemas que se costumam associar às redes sem fio simplesmente não existem, e persistem por desconhecimento dos avanços da tecnologia.

Os telefones sem fio estão no mercado há vários anos, e foi apenas uma questão de tempo para que os usuários sentissem a necessidade de fazerem uso de tal tecnologia para se conectarem às redes sem fio. O fato é que a tecnologia para tal já está presente, inclusive com produtos no mercado.

Ao longo deste trabalho ficará claro que muitas das suposições preconcebidas a respeito de comunicação de dados sem fio não têm fundamento. Itens como confiabilidade, segurança de dados, disponibilidade e mesmo taxa de transmissão, já são atingidos em nível bastante satisfatório com a tecnologia disponível.

1.3 Aplicações das Redes Sem Fio

No caso das redes sem fio, colocar uma estação no ar deve tomar apenas alguns minutos. Na maioria dos casos o usuário simplesmente conecta o dispositivo em seu PC (seja um adaptador ou dispositivo conectado ao PC), alinha a antena e passa a integrar a rede. A habilidade de se instalar e mover componentes da rede sem fio a tornam apropriada para aplicações nas quais as redes com fio seriam problemáticas.

Vejamos a seguir algumas razões que levariam os usuários a optarem por redes sem fio. Alguns prédios podem ser considerados complicados para receberem instalações de cabos, como por exemplo, edificações que podem ter sido tombadas pelo patrimônio histórico. Outro exemplo se aplica a prédios que, contendo altos requisitos de segurança (como paredes auto extinguentes) tornam extremamente cara qualquer tentativa de instalação de cabos. Da mesma forma, um escritório que seria utilizado temporariamente não justificaria semelhante investimento.

Outras razões podem vir exatamente da natureza de determinadas organizações, tais como auditorias no campo, convenções e simpósios que podem tirar grande proveito das redes sem fio.

Resumindo, as redes sem fio oferecem interessantes perspectivas, em particular para:

- *novas instalações onde não existem cabos;*
- *ambientes com grandes dificuldades para instalação de cabos;*
- *instalações temporárias;*
- *rápido crescimento de setores de empresas ou escritórios.*

1.4 Custo Como Motivação

Numa primeira abordagem, o grande atrativo das redes sem fio seria o fator econômico: o custo de uma rede local (LAN[1]) está relacionado diretamente com seu cabeamento. O custo de se conectar um único usuário a uma rede local, incluindo cabos e mão de obra, pode variar entre 200 a 2.000 dólares, sendo 800 dólares o custo médio {3}. Este cálculo não inclui o custo do cartão de interface da rede, sistemas operacionais das redes ou conexões em concentradores.

[1] do inglês "Local Area Network".

O custo inicial é apenas um aspecto dos altos preços de se conectar uma estação a uma LAN. As mudanças de cabos devido a alterações em "layout" podem custar tanto quanto uma nova instalação. As redes com fio necessitam, em geral, de especialistas, para planejar e executar as mudanças e de administradores para documentar estas mudanças e gerenciar a planta dos cabos.

Isto se contrapõe à facilidade de instalação de um usuário da rede sem fio que não tem a necessidade de auxilio de especialistas em redes e também não tem nenhum custo adicional além daquele relativo ao próprio adaptador para a rede.

1.5 Computadores Portáteis

Certamente, os usuários de computadores portáteis, "laptops", "notebooks" e "palmtops", serão bastante beneficiados com o avanço das redes sem fio. Até há pouco tempo, estes computadores tinham sido mantidos fora das redes locais. Para a maioria dos portáteis a única maneira de se conectarem a uma rede seria através de modems. Com as redes sem fio, os usuários podem se conectar às redes com a mesma facilidade que os computadores convencionais e com a possível vantagem adicional da mobilidade.

Um padrão de barramento elaborado especialmente para computadores portáteis permite a conexão de adaptadores de redes (como Token Ring e Ethernet) aos "laptops", "notebooks" etc. Este padrão é o PCMCIA[2] que define as características mecânicas e físicas dos cartões e estrutura de dados.

1.6 Sumário dos Capítulos

Neste primeiro capítulo, tratou-se basicamente de uma introdução às redes sem fio. No Capítulo 2, a abordagem se dá pelos tipos de redes sem fio, incluindo modo de transmissão de sinais, arquitetura da rede e protocolos de comunicação.

O Capítulo 3 trata do protocolo de acesso ao meio físico proposto. A partir da definição deste protocolo, torna-se possível a implementação completa da rede sem fio através da topologia da rede, organização de dados no meio físico e sincronismo de transmissão e recepção.

Uma possível implementação da plataforma hardware de tal proposta é tratada no Capítulo 4, sendo que sua análise detalhada se dará nos capítulos seguintes. Assim, o Capítulo 5 trata da implementação e características do encriptador de dados enquanto que o Capítulo 6 discorre sobre a compressão de dados. O Capítulo 7 trata da interface PCMCIA que é a interface escolhida para o Sistema de Processamento de Dados. A interface com Rádio e sincronismo entre estações da rede são assuntos tratados no Capítulo 8.

A contribuição pessoal do autor reside fundamentalmente na integração dos blocos funcionais citados no Capítulo 4 que compõem a plataforma da hardware do adaptador para rede sem fio. Destaca-se sua participação a frente do projeto e implementação de pioneira interface PCMCIA e da concepção e implementação do consumo racionalizado de energia descrito no

[2] PCMCIA PC CARD Standard, elaborado pela "Personal Computer Memory Card International Association".

Capítulo 7. Como resultado de tal trabalho, apresenta-se um cartão das proporções de um cartão de crédito (PCMCIA tipo II) contendo chips de uso comercial (microprocessador e compressor de dados) e de uso customizado (integração da interface PCMCIA, encriptação, autenticação e funções de suporte ao consumo racionalizado de energia), este último feito sob coordenação do autor deste trabalho.

1.7 Referências

{1} *P. Cripps, "Engineering Choices for Portable Wireless LAN Adapters", Doc: IEEE P802.11/91, pp 1-16, Nov. 1991.*

{2} *G. Berline e E. Perratore, "Portable Affordable, Secure: Wireless LANs", PC Magazine, pp. 291-314, Fev. 1992.*

{3} *C. J. Mathias, "Wireless LANs: The Next Wave", Data Communications, pp. 83-87, Mar. 92.*

{4} *A. Frank, "Networking Without Wires", LAN Technology, pp.53-64, Mar. 1992.*

Capítulo 2. Rede Sem Fio: Arquitetura e Parâmetros de Transmissão

Este capítulo descreve, de maneira sucinta, a arquitetura típica dos sistemas de rede sem fio e os parâmetros envolvidos. O propósito é facilitar a discussão, que virá a seguir, envolvendo topologias, arquiteturas especificas e especificações para rede sem fio. Nele discorreremos sobre topologias das redes sem fio, tipos de redes em relação ao uso do meio físico e características tais como desempenho, segurança, salubridade e compatibilidade. Para finalizar apresentamos os protocolos para redes sem fio apresentados ao grupo IEEE 802.11.

2.1 Introdução

Com o advento das redes, que permitiram aos computadores se comunicarem através de grandes arquivos de dados de maneira rápida e confiável, houve uma revolução na indústria da informática. Durante as últimas décadas, os sistemas de computação conseguiam apenas se comunicar através de linha telefônica a uma taxa de transmissão de centenas de bits por segundo. Mais recentemente as redes de computadores tornaram-se capazes de operar com taxa de transmissão da ordem de dez a cem megabits por segundo onde centenas de estações podem interagir umas com as outras. Cada um dos tipos de rede possui sua própria especificação, a qual define níveis elétricos, organização de dados num pacote e também distribuição espacial das estações que participam da rede.

A seguir descreveremos os parâmetros de transmissão e arquitetura que influenciam a elaboração das especificações para redes sem fio.

2.2 Topologia Básica

A Figura 1 mostra o esquema de uma estrutura básica de uma rede sem fio. O sistema está dividido em áreas distintas de cobertura, chamadas células. Cada uma dessas células opera com um conjunto distinto de freqüências ou com freqüências idênticas mas usadas de maneira conveniente de forma a se minimizarem as interferências.

Em uma célula podemos identificar dois tipos de dispositivos: Ponto de Acesso e estação remota. A estação remota é a unidade móvel, onde o usuário se instala. Ela é, geralmente, um PC ou um "notebook". Através da rede sem fio, este PC pode se conectar, por exemplo, a um servidor de dados, a impressoras ou a qualquer recurso a ser compartilhado pelo sistema. Como a própria topografia sugere, é esperado que uma estação remota possa se movimentar em toda a célula, ou mesmo de uma célula para outra. Isto porque, diferentemente do Ponto de Acesso, na topologia da Figura 1, a remota não está conectada a nenhuma rede local com fio. Torna-se, portanto, muito atrativa a introdução de computadores portáteis que realizem a função de estações remotas.

A função do Ponto de Acesso é gerenciar o transporte de informação de e para as estações remotas. Isto corresponde, por exemplo, a retransmitir os pacotes de dados das, e para as, estações remotas em sua célula, assim como direcionar os pacotes através da rede local, acessando dispositivos além da célula. Os Pontos de Acesso também proporcionam o gerenciamento do tráfego de dados e geram o sincronismo para as estações remotas.

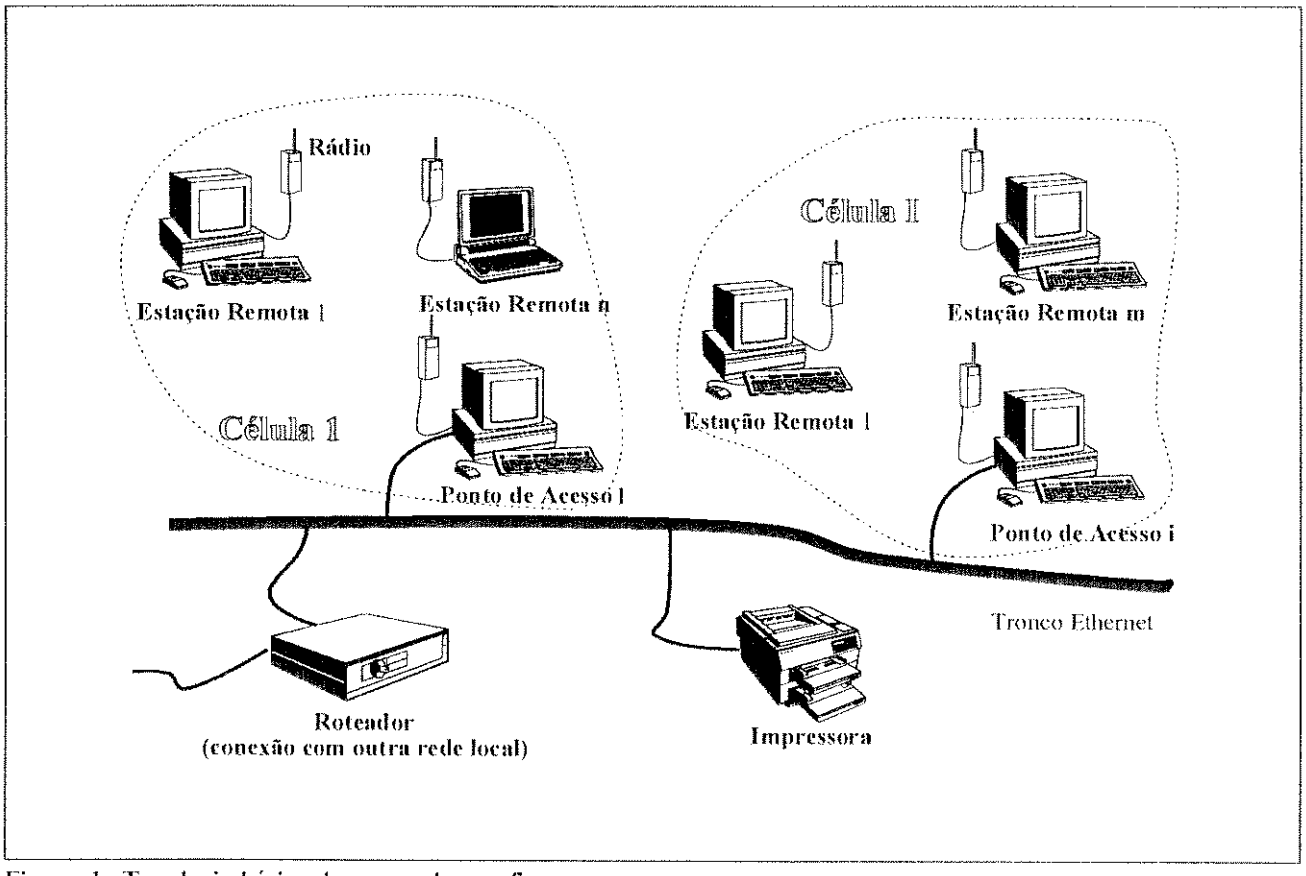


Figura 1. Topologia básica de uma rede sem fio.

2.3 Tipos de Redes Sem Fio

Existem basicamente dois tipos de tecnologia atualmente empregadas nas redes de comunicação sem fio: infra-vermelho e rádio. O infra-vermelho, que já é intensamente utilizado em controles remotos para operar com televisões, video cassetes e aparelhos de som, não é controlado por nenhuma agência governamental em qualquer país. Isto ocorre porque, para as aplicações de infra-vermelho é necessário que haja o que chamamos de linha de visada, ou seja, que o transmissor e receptor tenham livre o espaço entre eles. Além disso, as agências governamentais não se preocupam com o infra-vermelho pelo fato de que, como a luz, ele ser completamente imune a interferências eletromagnéticas. No entanto, para prédios que possuam muitas divisórias ou paredes torna-se inviável a emprego de redes que utilizem de infra-vermelho, uma vez que, como foi dito acima, o mesmo necessita da linha de visada.

Por sua vez, as redes que se utilizam de ondas eletromagnéticas para transmitir seus dados são completamente controladas pelas agências governamentais no que se refere a faixa de operação em freqüência e potência transmitida. No entanto, através de rádio, a mobilidade é completa uma vez que o usuário pode se deslocar ao mesmo tempo em que continua operando na rede sem fio. Neste trabalho será dada ênfase às redes de comunicação via rádio.

2.4 Redes de Comunicação Via Rádio

As redes que utilizam rádio se diferenciam em duas outras tecnologias: transmissão convencional de rádio e transmissão por espalhamento de espectro. A transmissão por espalhamento de espectro surgiu na segunda guerra mundial com dois objetivos: o primeiro era tornar o sinal mais imune a interferências (intencionais ou não), enquanto que o segundo visava a segurança da transmissão de informações. Este tipo de transmissão possui seu espectro espalhado por uma faixa de freqüência maior que a utilizada na transmissão convencional, porém com menor intensidade. Como resultado, o primeiro objetivo é alcançado uma vez que uma interferência para degenerar o sinal teria que também possuir espectro extenso. Por outro lado, o segundo objetivo, o de manter a segurança na transmissão, é também atingido pois, para se espalhar o espectro faz-se uso de códigos apropriados de modo a dificultar a obtenção da informação contida no sinal.

Existem porém duas formas principais de se espalhar o espectro de um sinal sobre uma faixa de transmissão: Salto em Freqüência e Seqüência Direta. No método do Salto em Freqüência a informação é transmitida por um conjunto de freqüências portadoras, ocorrendo individualmente em instantes de tempo diferentes, e numa seqüência que obedece a um padrão pseudo-aleatório.

No segundo método de espalhamento chamado de Seqüência Direta utiliza-se de um código de espalhamento para codificar a informação a ser transmitida. Este código associa a cada bit da mensagem original um número de bits que são finalmente transmitidos. Nota-se que a transmissão de apenas um bit se transforma na transmissão de um número, variável com o código, de bits. Assim, por exemplo, se o código proporciona 10 bits para cada bit da mensagem original a mensagem a ser transmitida terá seu comprimento multiplicado por 10. Continuando o exemplo acima, este método faria uso de uma faixa de freqüência de 10 vezes a necessária para efetuar a transmissão da mensagem original.

2.4.1 Interferência de Seqüência Direta e Salto em Freqüência

Uma das mais importantes diferenças entre Seqüência Direta e Salto em Freqüência pode ser notada quando da sua aplicação na presença de uma ou mais faixas estreitas de interferência.

Tomemos um sistema operando com Seqüência Direta com cada bit de informação sendo codificado em 10 outros bits para transmissão e ocupando uma faixa de 10 MHz. Este sistema será comparado com outro sistema operando com Salto em Freqüência contendo 10 freqüências distintas, cada freqüência ocupando uma faixa de 1 MHz. A interferência de faixa estreita terá a mesma potência total do sinal transmitido em ambos os casos. A Figura 2 mostra os esboços dos espectros em questão.

À saída do receptor, no caso da Seqüência Direta, tem-se um sinal com a potência concentrada em 1MHz. Desta forma a potência do sinal interferente espalhada nos 10MHz aparecerá, à sua saída, com apenas 10% do seu valor. Isso corresponde a uma relação sinal/ruído de entrada e sinal/ruído de saída de 10 dB. Note que se o demodulador necessitar de uma relação sinal/ruído de entrada e saída maior que 10 dB, não haverá sucesso na demodulação. Se por outro lado o demodulador necessitar de uma relação sinal/ruído de entrada e saída menor que 10 dB a interferência não terá efeito algum.

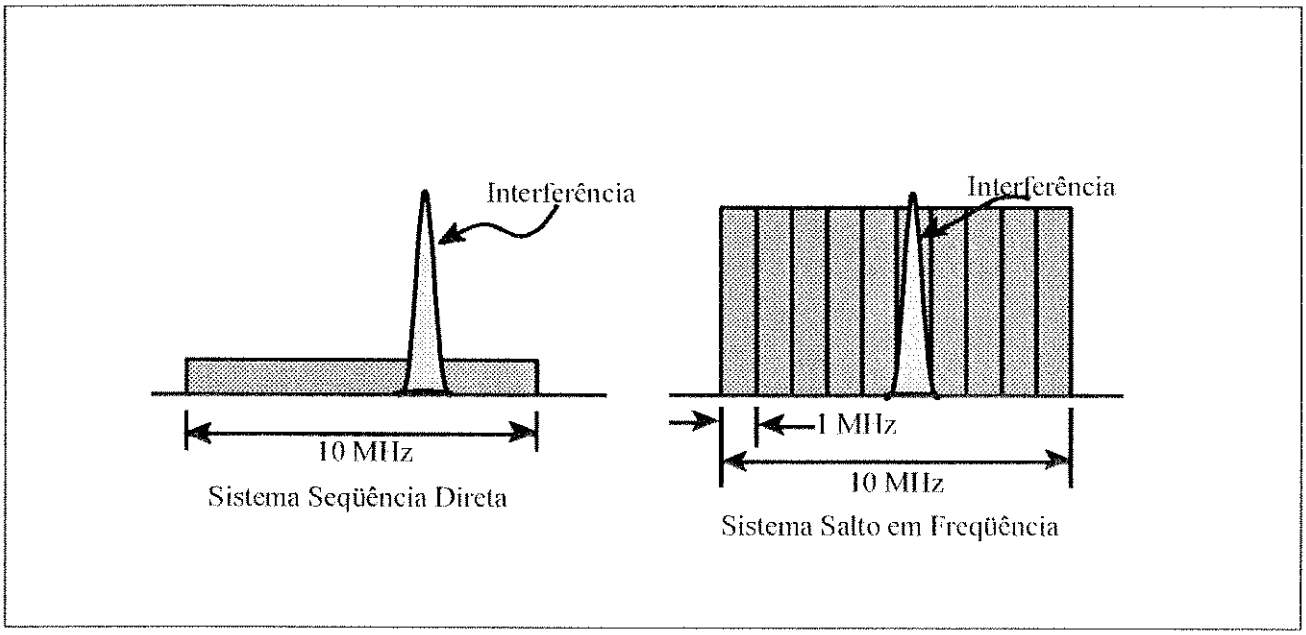


Figura 2. Esboço dos espectros Seqüência Direta e Salto em Freqüência.

Para o sistema de Salto em Freqüência, com a interferência de faixa estreita afetando apenas uma das freqüências de transmissão, tem-se uma razão sinal/ruido na saída do receptor de 0 dB (para uma especifica freqüência). Todos demais freqüências de transmissão não serão afetadas pela interferência. É de se esperar que 90% da taxa de transmissão seja possível utilizando as 9 freqüências restantes. Para um protocolo um pouco mais elaborado, que monitore a utilização das freqüências, a faixa que sofre interferência pode ser eliminada, chegando-se assim aos 100% da taxa de transmissão.

Para finalizar este exemplo, podemos observar que uma conexão usando Seqüência Direta pode se desfazer ou não quando ocorre uma forte interferência de faixa estreita, enquanto que outra conexão operando com Salto em Freqüência pode continuar transmitindo com taxas menores. Assim sendo, ambos as formas de espalhamento espectral apresentam características próprias que possibilitam suas utilizações em diferentes aplicações.

2.5 Desempenho, Confiabilidade, Segurança e Salubridade

Entre as restrições que o mercado ainda faz com relação às redes sem fio encontram-se considerações sobre desempenho, confiabilidade, segurança de informações e salubridade. Sobre performance podemos dizer que os progressos são muitos, basta ver os lançamentos do mercado e suas taxas de operação {2}{3}{5}{6}{7}. Atualmente, não há barreiras tecnológicas que impeçam as redes sem fio de serem compatíveis em velocidade às redes com fio Ethernet, Token Ring ou mesmo FDDI. O uso de compressores de dados, por exemplo, permite a estas redes alcançar desempenhos que se igualam aos das redes com fio.

No que diz respeito à confiabilidade, as redes sem fio possuem condições para superar suas concorrentes com fio. Estudos e pesquisas têm mostrado que a causa mais comum da indisponibilidade da rede relaciona-se com os problemas de cabos e conexões {6}. Obviamente, as redes sem fio levam grande vantagem nesse aspecto em relação às com fio, além de possuírem uma quantidade de conexões muito menor.

Um dos motivadores do uso de espalhamento espectral para transmissões em rádio foi justamente a segurança dos dados. Isto é obtido de imediato para o caso de sistemas operando com Seqüência Direta, pois o próprio código de espalhamento protege a mensagem original. No caso de sistemas por Salto em Freqüência, o uso de encriptadores está se tornando uma prática que determina alto grau de segurança. Por sua vez, a transmissão por infra-vermelho necessita de uma linha de visada entre transmissor e receptor, o que cria uma dificuldade topológica para o interceptador. Além disso, essas redes costumam usar tipos particulares de codificação dos sinais transmitidos.

Preocupações sobre potenciais riscos à saúde, causados por redes sem fio utilizando rádio, parecem ser o maior obstáculo da tecnologia em questão. Conclusões a respeito dos efeitos causados pela exposição do corpo humano a ondas de rádio em alta freqüência ainda não são definitivas. Enquanto isso, indústrias, laboratórios e organizações governamentais têm acordado sobre o uso de transmissores de baixa potência (100 mW no caso dos Estados Unidos e Europa).

2.6 Compatibilidade entre Redes Sem Fio

Ainda não existe disponível uma especificação aceita por toda a indústria de um padrão para rede sem fio. Mesmo assim, produtos que operam em faixas de freqüência que não necessitam de aprovação dos órgãos competentes já existem no mercado. Neste sentido o IEEE 802.11, como parte do projeto de padronização IEEE 802, constitui um grupo de trabalho, operando em forum aberto, designado para as questões relativas às redes sem fio. Criado em 1990, sua missão era a de definir um protocolo de acesso ao meio físico e as especificações para redes operando com rádio e infra-vermelho, como parte de uma proposta submetida à ISO/IEC[3] para um padrão internacional. Este grupo é reconhecido como ponto focal para o desenvolvimento de padrões de redes.

Apesar das redes sem fio incluirem infra-vermelho, a maioria do grupo tem interesse em propagação via rádio, cujos maiores desafios do novo padrão internacional são: restrições ao desempenho da rede sem fio (por freqüentemente se conectarem às redes com fio já existentes), potência radiada (devido às questões de salubridade) e, finalmente, faixas de freqüência de possível utilização na atualidade.

Durante os estudos de técnicas de modulação, o interesse do grupo IEEE 802.11 se voltou para as faixas de freqüência de rádio que não requerem licença do FCC[4]. A única oportunidade para operar nestas freqüências era usar modulação por espalhamento espectral, que é permitido pelo FCC nas seguintes faixas:

- *902-928 MHz;*
- *2.400-2.500 GHz e*
- *5.725-5.875 GHz.*

Estas faixas estão designadas indistintamente para uso industrial, científico e médico.

---

[3] "International Standard Organization".

[4] "Federal Communications Commission", orgão norte americano que determina alocação de freqüências para uso.

Após formular os objetivos de projeto baseados nas necessidades do usuário, o grupo 802.11 trabalhou no sentido de estabelecer uma arquitetura básica da rede e finalmente na elaboração do padrão do protocolo de comunicação desenvolvido. Para tal, o grupo contou com a colaboração de interessados em obter uma padronização para seu protocolo para rede sem fio. Entre estes estão empresas que, através dos seus centros de pesquisas, desenvolveram protocolos e objetivam lançar-se no mercado.

2.7 Protocolos de Acesso ao Meio Físico

Esta seção tem como objetivo descrever os protocolos utilizados para redes sem fio. Serão descritos os protocolos CSMA/CA e Híbrido, ambos propostos no grupo IEEE 802.11. O protocolo de acesso aqui proposto e também apresentado ao grupo IEEE 802.11 é descrito em detalhes no Capítulo 3.

2.7.1 Protocolo CSMA/CA

Este protocolo está sendo utilizado pela compania NCR em seu produto WaveLan. Tem como característica básica a utilização do mesmo CSMA[5] que originou o Ethernet, porém com a preocupação adicional de se evitarem colisões. Descreveremos este protocolo a partir de seu histórico, uma vez que o próprio se origina de seus antecessores.

Protocolo ALOHA

Este protocolo surgiu na década de 70 como uma maneira nova de se resolver o problema de alocação de canal de transmissão compartilhado. A idéia, neste caso, é aplicável a qualquer sistema que possua usuários não coordenados que competem para utilizar um único canal de transmissão compartilhado.

O protocolo ALOHA permite aos usuários transmitir a qualquer instante, sendo que as mensagens que colidirem serão destruídas. Entretanto, o usuário do canal pode sempre verificar se sua mensagem foi destruída utilizando-se de propriedades da rede ou simplesmente mantendo a escuta no canal. Numa rede local, esta informação vem de imediato, pois, praticamente, não há atrasos no canal. Supondo que sua mensagem tenha sido destruída, o emissário apenas espera por um tempo aleatório e re-envia a mensagem. O tempo de espera deve ser aleatório ou as mensagens continuarão a colidir a cada nova tentativa.

Um exemplo genérico da maneira como um sistema ALOHA gera suas mensagens é mostrada na Figura 3. Cada retângulo representa um pacote de dados enviado pelo canal.

Notamos que, mesmo que apenas o último bit de uma mensagem colida com o primeiro bit de outra, ambos se perderão e as mensagens deverão ser transmitidas novamente (haverá, certamente, um erro detectado pela verificação do código de redundância que acompanha as mensagens).

[5] do inglês "Carrier Sense Multiple Access".

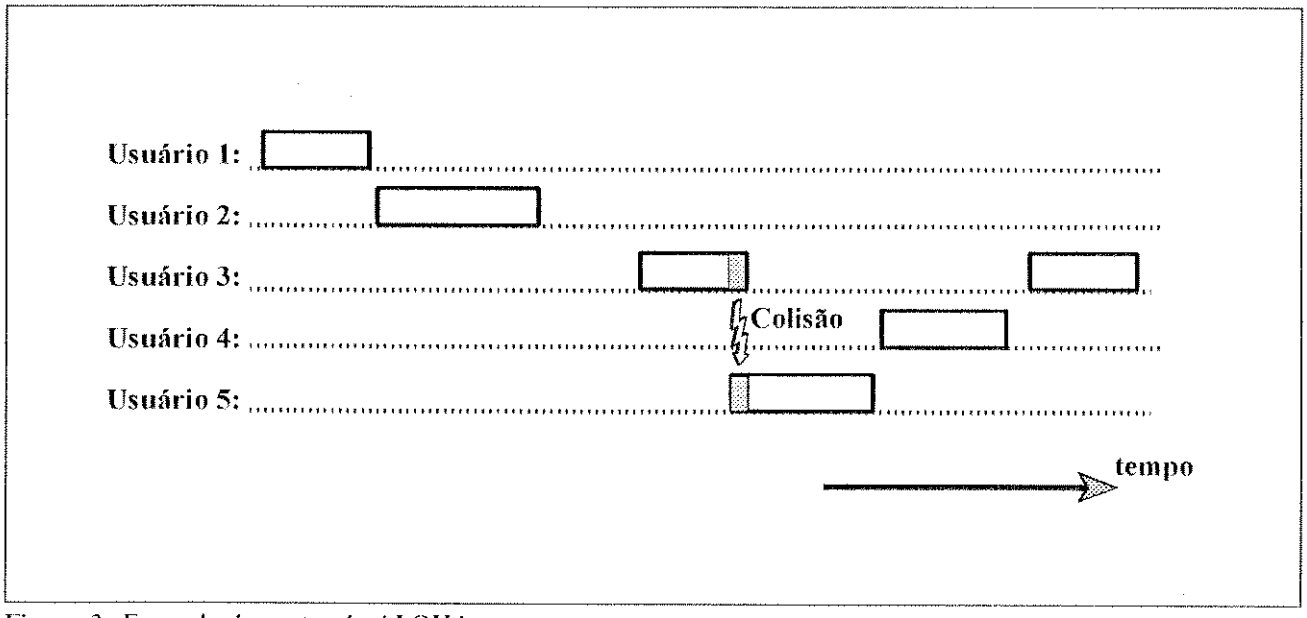


Figura 3. Exemplo do protocolo ALOHA puro.

Um novo método para o ALOHA foi elaborado e com ele poderia se dobrar a capacidade de transmissão. A proposta era dividir o tempo em intervalos discretos e cada intervalo corresponderia a apenas uma mensagem. Uma maneira de se obter a sincronização entre os usuários do canal seria estabelecer uma estação especial que emitisse um sinal a cada início de intervalo. Este método ficou conhecido como ALOHA particionado ("slotted ALOHA"), diferenciado-se do ALOHA puro.

CSMA Persistente e Não Persistente

O protocolo CSMA persistente-1 opera da seguinte forma: quando uma estação tem dados a enviar pela rede, ela observa primeiro o canal de transmissão a fim de verificar se não há outra estação transmitindo. No caso do canal estar disponível, a estação inicia a transmissão. Se o canal está ocupado, a estação irá esperar até que fique disponível, para iniciar a transmissão. Se houver colisão, a estação espera por um tempo aleatório e reinicia o processo de transmissão. Este protocolo é chamado de persistente-1 porque a estação transmite com probabilidade igual a 1 assim que percebe o canal disponível. Haverá colisões se, por exemplo, duas estações prontas para transmitir esperam o término da transmissão de uma terceira para começarem suas transmissões.

Também o atraso devido à propagação exerce um importante papel no desempenho do protocolo. Existe uma pequena chance de que logo em seguida a uma estação começar a transmitir, uma segunda estação fique pronta para enviar uma mensagem pelo canal. Se o sinal da primeira estação ainda não atingiu a segunda, esta perceberá o canal disponível e iniciará sua transmissão resultando em colisão. Conclui-se portanto que, quanto maior o atraso de propagação no canal, maior será a probabilidade de colisão e pior será o desempenho do protocolo.

O protocolo CSMA não persistente difere do CSMA descrito acima por justamente não persistir na espera da liberação do canal por outra estação. Neste caso, antes de enviar uma mensagem, a estação observa o canal. Se este está disponível, a estação começa a transmissão. Entretanto, se o canal está em uso, a estação espera um período aleatório de

tempo e repete o algoritmo. Isto evita que duas estações que estejam prontas para transmitir iniciem transmissão tão logo o canal se torne disponível.

Um pouco mais elaborado é o protocolo CSMA persistente com probabilidade p. Este protocolo é aplicado a canais com acesso restrito a intervalos fixos de tempo ("slotted"). Quando a estação está pronta para transmitir, o canal é observado. Se este está disponível, a estação transmite com probabilidade p. Assim, com probabilidade q = 1 - p a estação adia sua transmissão para o próximo intervalo de acesso ao canal. Se neste próximo intervalo o canal está disponível, a estação transmite ou adia a transmissão segundo o critério acima. Este processo se repetirá até que ou a mensagem seja transmitida ou alguma outra estação inicie sua transmissão. Neste último caso a estação espera um período aleatório de tempo e inicia o processo novamente. Se a estação observa o canal inicialmente ocupado, ela espera até o próximo intervalo de acesso ao canal e inicia o processo descrito acima.

CSMA/CD (Colisão Detectada)

Este protocolo é ainda mais uma melhoria em relação ao ALOHA. Neste caso, se duas estações perceberem que o canal está disponível e iniciarem a transmissão simultaneamente, ambos detectarão que houve colisão e interromperão a transmissão. Desta forma, quanto mais rápido se abortam as mensagens danificadas pela colisão, mais tempo o canal estará disponível.

Em outras palavras, quando o canal está disponível para a transmissão, a estação pode iniciar seu envio de mensagem. Caso ele esteja ocupado por outra estação, as estações que desejam transmitir aguardam o término da transmissão corrente. Caso haja colisão, as estações aguardam um período de tempo aleatório e reiniciam a transmissão.

CSMA/CA (Colisão Evitada)

O principal mecanismo usado nas redes sem fio é o CSMA/CA. Com este protocolo tenta-se evitar a colisão minimizando-se a utilização do meio de transmissão. Da mesma forma que o CSMA/CD, o CSMA/CA faz uma estação transmitir sua mensagem sempre que o canal estiver disponível. No entanto, caso o canal esteja ocupado, a estação que deseja enviar irá esperar até o término da mensagem que está sendo transmitida e enviar sua mensagem com probabilidade p. Desta forma, quando a estação percebe o canal ocupado, ela irá usar o protocolo CSMA persistente com probabilidade p. Uma variação pode ser acrescentada ao CSMCA/CA: a emissão de uma mensagem de recebimento correto ("acknowledgement" - ACK) feito pela estação de destino assim que recebe a mensagem transmitida. Essa variação é chamada de CSMCA/CA ACK.

A eficiência deste protocolo pode ser considerada boa uma vez que atinge aproximadamente 87% da banda disponível para mensagens grandes {10}. Isto é similar à eficiência do CSMA/CD, especificado pelo padrão IEEE 802.3.

2.7.2 Protocolo de Acesso Hibrido: Síncrono e Assíncrono

Este protocolo foi proposto ao Grupo IEEE 802.11 por técnicos da Xircom Inc. Apresenta como característica fundamental a presença de serviços síncronos e assíncronos.

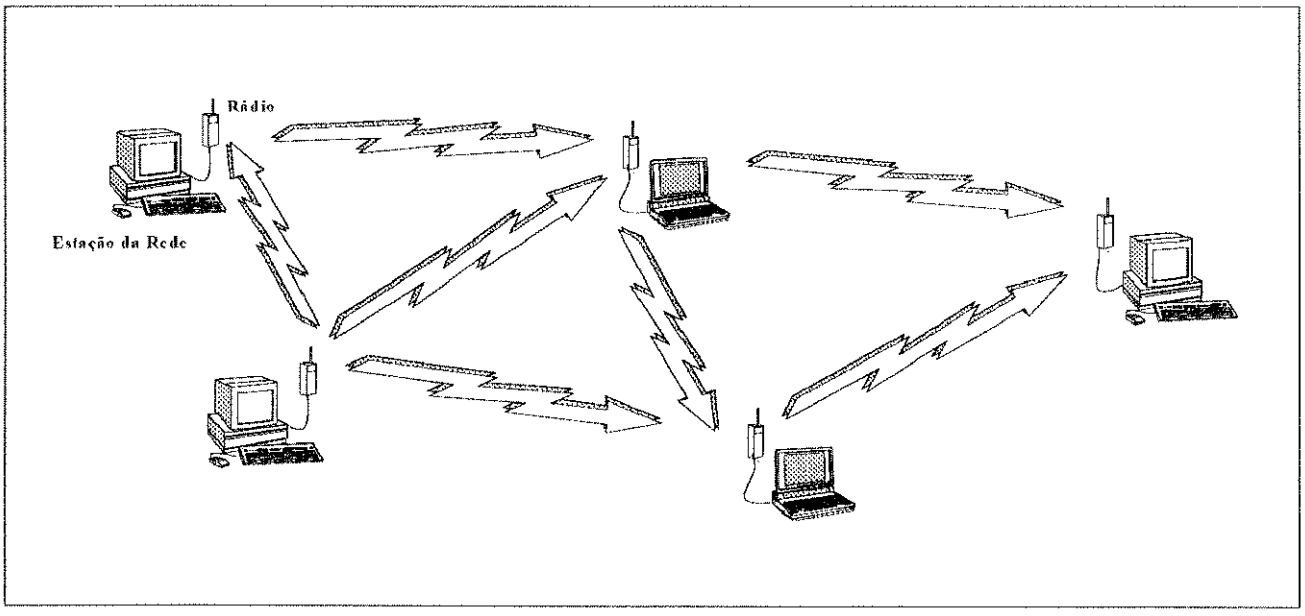


Figura 4. Configuração de acesso direto entre estações.

Tipos de Serviços

Os serviços síncronos são baseados em temporização e baixos atrasos para atendimento dos mesmos. Estes serviços sempre têm prioridades sobre os serviços assíncronos.

Os serviços assíncronos possuem baixos atrasos de acesso ao meio e, pela sua própria natureza, são estocásticos. Estes serviços podem atingir até 80% da taxa de transmissão máxima {11}.

Tipos de Configurações

As configurações dizem respeito à possibilidade das estações se comunicaram umas com as outras:

- ***Configuração de igualdade**, na qual todas as estações têm permissão para transmitir diretamente para outra estação e, quando não conseguem acessar a determinada estação, usam de outra estação para repassar a mensagem (vide Figura 4).*
- ***Configuração hierárquica**, na qual as mensagens entre estações se processam através de um Ponto de Acesso (vide Figura 5), ou seja, o acesso é indireto.*

Descrição Sucinta do Protocolo de Acesso Híbrido

O protocolo é simples e utiliza de partições no tempo para transmitir as mensagens. Inicialmente uma estação que deseja transmitir faz uma requisição para envio de mensagem ("request to send") e observa o meio aguardando, da estação destino, a permissão para transmitir ("clear to send"). Após transmitir a mensagem, a estação destino retorna com uma confirmação de mensagem recebida.

Caso haja colisão, o processo se desencadeia como no CSMA, no qual a estação espera durante um certo tempo escolhido aleatoriamente.

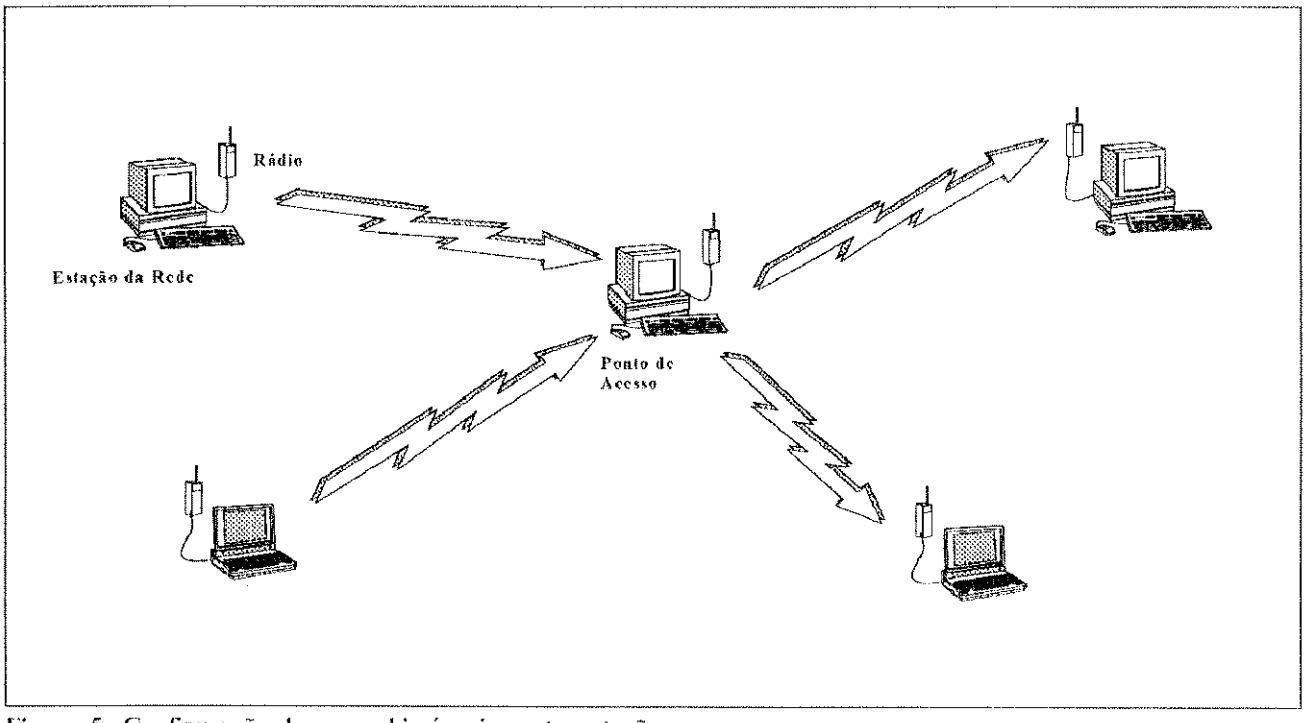


Figura 5. Configuração de acesso hierárquico entre estações.

2.8 Arquitetura e Características da Rede Proposta

Nesta seção colocaremos as características da rede sem fio proposta. A partir das características físicas desta rede se desenvolverão os demais capítulos do trabalho que descrevem o protocolo de acesso e a implementação do adaptador para a rede.

A topologia da rede proposta baseia-se na utilização de um Ponto de Acesso, a qual determina que qualquer troca de mensagens entre estações da rede deve ter a participação do Ponto de Acesso.

Quanto ao desempenho, a taxa de transmissão para a rede proposta deve ser de 1 MBits por segundo. Para sua compatibilidade, em termos de vazão, com as demais redes com fio, o uso de compressor de dados se torna necessário.

Em relação aos tipos de redes quanto ao emprego de tecnologia, a rede proposta faz uso de Salto de Freqüência operando na faixa de 2,4 a 2,484 GHz. A faixa a ser utilizada em cada salto de freqüência é de 1 MHz, o que permite atingir a taxa de transmissão de 1 Mbits por segundo conforme citado acima.

2.8.1 Protocolo de Acesso ao Meio Físico Proposto

A principal motivação para o acesso reservado são as aplicações síncronas tais como voz ou imagem. Com a reserva de faixa de transmissão, possível no protocolo proposto, as aplicações ditas síncronas são suportadas. Assim sendo, garante-se uma certa taxa de transmissão para uma determinada aplicação durante um intervalo de tempo. A descrição detalhada deste protocolo é objetivo do Capítulo 3.

2.9 Conclusões

A topologia, meio físico empregado para transmissão, segurança e parâmetros de desempenho são alguns requisitos importantes para a implementação de uma rede sem fio. É necessário, portanto, que o protocolo de comunicação atenda a todos os requisitos citados acima de modo a permitir a operação desejada da rede.

Neste capítulo, vimos quais são as topologias básicas das redes sem fio, tipos de sinais empregados, necessidades de desempenho, confiabilidade e segurança. Vimos ainda dois tipos de protocolos de comunicação para rede sem fio. No capítulo seguinte, discorreremos sobre o protocolo de acesso ao meio físico proposto que se propõe a compatibilizar todos os requisitos necessários para uma rede sem fio, tornando-a factível.

2.10 Referências

{1} *V. Hayes, "Standarization Effords for Wireless LANs", IEEE Network Magazine pp. 19-20, Nov. 1991.*

{2} *J. E. Mitzlaff, "Radio Propagation and Anti-Multpath Technics in the WIN Environment", IEEE Network Magazine, pp. 21-26, Nov. 1991.*

{3} *P. Cripps, "Engineering Choices for Portable Wireless LAN Adapters", Doc: IEEE P802.11/91, pp 1-16, Nov. 1991.*

{4} *D. Buchhols, P. Odlyzko, M. Taylor e R. White, "Wireless In-Building Network Architeture and Protocols", IEEE Network Magazine, pp. 31-38, Nov. 1991.*

{5} *G. Berline e E. Perratore, "Portable Affordable, Secure: Wirelless LANs", PC Magazine, pp. 291-314, Fev. 1992.*

{6} *C. J. Mathias, "Wireless LANs: The Next Wave", Data Communications, pp. 83-87, Mar. 1992.*

{7} *A. Frank, "Networking Without Wires", LAN Technology, pp.53-64, Mar. 1992.*

{8} *A.S. Tanenbaum, "Computer Networks", Prentice-Hall, Inc., 1989.*

{9} *K.S. Natarajan, C.C. Huang, D.F Bantz, "Medium Access Control Protocol for Wireless LANs", IEEE 802.11/92-39, Mar. 1992.*

{10} *W. Diepstraten, "Wireless Access Method and Physical Layer Specifications", IEEE 802.11/92-51, Maio 1992.*

{11} *K. Biba, "Adaptive Distributed and Centralized Coordination", IEEE 802.11/92-49, Maio 1992.*

{12} *M. Smith, "Further Simulation of the Hybrid MAC Protocol, IEEE 802.11/92-37, Mar. 1992.*

Capítulo 3. Protocolo de Acesso ao Meio Físico Proposto

Este Capítulo desecreve o protocolo de acesso ao meio físico aqui proposto e implementado. Este protocolo é um misto daqueles vistos no capítulo anterior fazendo uso de acessos aleatórios e reservados.

3.1 Introdução

O protocolo de acesso proposto tem como intenção utilizar as vantagens de um protocolo de acesso aleatório e de um protocolo de acesso reservado. Para tal, o protocolo proposto utiliza de um quadro de tempo no qual ora opera com um tipo de acesso, ora com outro.

A fim de proporcionar uma distribuição racional no tempo dos acessos aleatórios e reservados, o quadro de tempo é repetido sincronamente. O controle desta sincronicidade e do tráfego de informações estará a cargo de uma das estações que compõem a rede. Esta estação é denominada de Ponto de Acesso. Assim, o quadro de tempo é dividido em três fases: a primeira, fase A, se presta a acessos reservados com tráfego feito do Ponto de Acesso para as estações remotas; a segunda, fase B, também de acessos reservados mas com tráfego feito das estações remotas para o Ponto de Acesso; finalmente, a terceira, fase C, de acessos aleatórios feitos das estações remotas para o Ponto de Acesso ou para as demais estações da rede. A estrutura dos quadros temporais é mostrada na Figura 6. Na fase C, o meio de transmissão e' disputado pelas estações remotas com o objetivo de obterem alocação nas fases A e B.

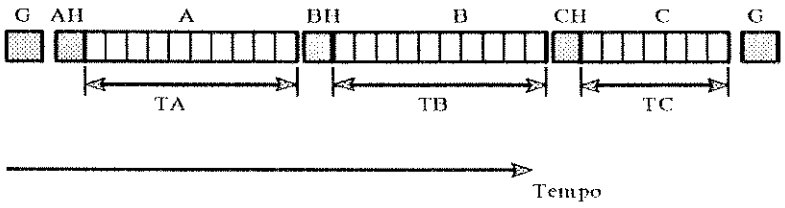
3.2 As Fases A, B e C

3.2.1 Fase A (tráfego Ponto de Acesso - Estações Remotas)

Durante a fase A está presente o chamado cabeçalho AH, que contém informações a respeito de como se apresenta este quadro de tempo e informações essenciais para que as estações remotas possam continuar operando na rede. Estas informações são enviadas simultaneamente para todas as estações em transmissão dita "broadcast".

Além de identificar o início da fase A, o cabeçalho AH contém os campos:

- *TA: duração do período A;*
- *TB: duração do período B;*
- *TC: duração do período C;*
- *TAH: duração do cabeçalho AH;*
- *TBH: duração do cabeçalho BH;*
- *TCH: duração do cabeçalho CH;*
- *Sincronização para salto em freqüência;*
- *Identificação do Ponto de Acesso;*
- *Identificação da rede;*
- *Próxima freqüência para saltar;*
- *Relação das estações receptoras;*
- *Indicação de transmissão "broadcast".*



FASE A: transmissão do Ponto de Acesso para as estações remotas
FASE B: transmissão das estações remotas para o Ponto de Acesso ou para outras remotas
FASE C: acesso aleatório

G: início do quadro de tempo (sincronização)
AH: cabeçalho do período A
BH: cabeçalho do período B
CH: cabeçalho do período C
TA: duração do período A
TB: duração do período B
TC: duração do período C

Figura 6. Estrutura do quadro de tempo no protocolo acesso ao meio físico.

Após o recebimento correto do cabeçalho AH, a estação remota sabe se receberá ou não mensagem na fase A vinda do Ponto de Acesso. Caso não exista mensagem destinada a ela nesta fase, a estação remota, através de um ajuste adequado de seu temporizador, TA, aguarda as informações disponíveis no cabeçalho BH.

A fase A é unicamente utilizada para transmissão de mensagens do Ponto de Acesso para as estações remotas. Caso não exista mensagem alguma a ser transmitida nesta direção, o tempo de duração da fase A (TA) será zero.

A fase A é dividida em intervalos iguais de tempo alocados às estações remotas. A alocação depende da demanda de tráfego e é determinada pelo Ponto de Acesso, evitando-se, desta forma, eventuais colisões.

3.2.2 Fase B (tráfego Estações Remotas - Ponto de Acesso)

O cabeçalho BH é utilizado pelo Ponto de Acesso para informar às estações remotas o fim da fase A e o início da fase B. Além disso, o cabeçalho BH também contém os seguintes campos:

- *TB: duração da fase B;*
- *TC: duração da fase C;*
- *Número de estações remotas que têm intervalos alocados na fase B;*
- *O par I, S(I), indicando à estação I os intervalos de tempo S(I) a ela alocados. Como a lista é ordenada, a seqüência de transmissão das estações remotas é conhecida por todas estações. Assim, qualquer estação remota sabe quando iniciar a transmissão o que evita colisões.*

No cabeçalho BH a estação remota que tenha solicitado transmissão no quadro de tempo anterior verificará se teve sucesso. A fase B, portanto, opera da mesma forma que a fase A.

Agora, porém, o Ponto de Acesso apenas aloca os intervalos (também de igual duração) que serão utilizadas para a transmissão das estações remotas.

Caso não haja nenhuma estação requisitando alocação na fase B, o tempo desta fase (TB) pode ser reduzido a zero. Portanto, no que diz respeito à alocação nas fases A e B, o protocolo é dito dinâmico.

3.2.3 Fase C (tráfego assíncrono)

Como nas outras fases, a fase C também possui seu cabeçalho CH. Ele identifica o fim da fase B, início da fase C, além de possuir os seguintes campos:

- *TC: duração da fase C;*
- *K: o número estimado de estações remotas ativas tentando acesso ao meio físico na fase C (que obviamente é menor ou igual ao número de remotas registradas no Ponto de Acesso).*

Nesta fase, as remotas não necessitam de alocação do Ponto de Acesso para transmitirem podendo haver colisões. O protocolo ALOHA particionado ou mesmo o CSMA/CA são pertinentes para esta fase.

Se a estação remota não deseja transmitir, ela simplesmente ajusta um temporizador para sinalizar o término da fase C, com a informação proveniente do cabeçalho desta fase.

Devido às possíveis colisões, a estação destino (Ponto de Acesso ou estação remota) retorna, a cada transmissão na fase C, um "acknowledgement" acusando o recebimento das mensagens. O não recebimento desta confirmação faz a estação remota retransmitir sua requisição no próximo quadro.

Relativamente a fase C, o protocolo estabelece uma duração mínima, de 20% da duração do quadro de tempo. Isto para que as requisições sejam atendidas num curto período de tempo, minimizando o tempo de acesso à rede. Desta forma, TC pode variar de 20% a aproximadamente 100% da duração do quadro de tempo.

A fase C e' usada para:

- *Requisitar do Ponto de Acesso a permissão (chamada de registro) para participar da rede;*
- *Requisitar do Ponto de Acesso faixa para transmissão;*
- *Transmissão de mensagens.*

Registro

É identificado como o processo da estação remota de apresentar-se ao Ponto de Acesso e requisitar seus serviços. O processo de registro deve se iniciar pela estação remota observando o meio de transmissão (chamado de "escuta") e escolhendo um Ponto de Acesso (no caso de haver mais que um na mesma região). Após isso, a estação remota envia ao Ponto de Acesso uma requisição para registro. Ao recebê-la, o Ponto de Acesso verifica a validade e, em caso positivo, concede o registro.

Requisição de Partições para Transmissão

Como já foi dito, na fase C são feitas as requisições para transmissões a serem efetuadas na fase B. A requisição é feita das estações remotas para o Ponto de Acesso utilizando se o protocolo "slotted" ALOHA. O Ponto de Acesso recebe a requisição e aloca, se possível, intervalos de tempo para a estação remota na fase B do próximo quadro de tempo. Caso não haja intervalo de tempo disponível no próximo quadro de tempo, a estação remota deve esperar a possível alocação no quadro de tempo seguinte e assim por diante.

A mensagem de requisição de intervalo de tempo para transmissão deve conter as seguintes informações:

- *endereço da estação remota solicitante;*
- *identificação da rede;*
- *identificação do Ponto de Acesso;*
- *tamanho da reserva solicitada (intervalo de tempo)*
- *tipo de serviço requerido (assíncrono ou síncrono).*

Uma vez reservado um determinado número de intervalos de tempo para um serviço síncrono, então este número estará alocado em cada quadro de tempo até que seu cancelamento seja solicitado por uma nova requisição.

Transmissão Entre Remotas

A transmissão entre remotas numa mesma célula pode acontecer da seguinte forma:

- *Mensagem direta: as estações remotas comunicam-se diretamente através de uma única transmissão.*
- *Modo de repetição: o Ponto de Acesso recebe a mensagem da estação fonte e a retransmite para a estação destino. Apesar deste modo envolver duas transmissões para uma simples comunicação entre duas estações, ele garante que toda a área da célula seja alcançada (note que apenas o Ponto de Acesso tem necessidade de ter todas as estações remotas em seu alcance dentro de uma célula).*

No caso da mensagem direta dois tipos de transmissões são possíveis. No primeiro trata-se de um acesso na fase C no qual já se inclui a mensagem a ser transmitida. É chamada de "mensagem direta por acesso aleatório". Neste caso a estação fonte deve elaborar um pacote contendo informações tais como: tipo de transmissão da mensagem direta, endereço da estação fonte, endereço da estação destino e mensagem propriamente dita. Vide Figura 7, tipo (1).

O segundo tipo é chamado de "mensagem direta por alocação de intervalos de tempo". Neste caso a transmissão é feita na fase B e a alocação é feita pelo Ponto de Acesso através de requisição na fase C. O Ponto de Acesso anexa as seguintes informações no cabeçalho da fase B no qual acontecerá a transmissão: tipo de transmissão direta, endereço da estação fonte, endereço de destino e intervalos de tempo alocadas. Vide Figura 7, tipo (2).

Obviamente os tipos de mensagem direta descritos acima só se realizam quando uma estação remota está em alcance da outra. Em caso de não obter sucesso em algumas tentativas, a estação fonte entra no modo de repetição para retransmitir sua mensagem para a estação destino. Vide Figura 7, tipo (3).

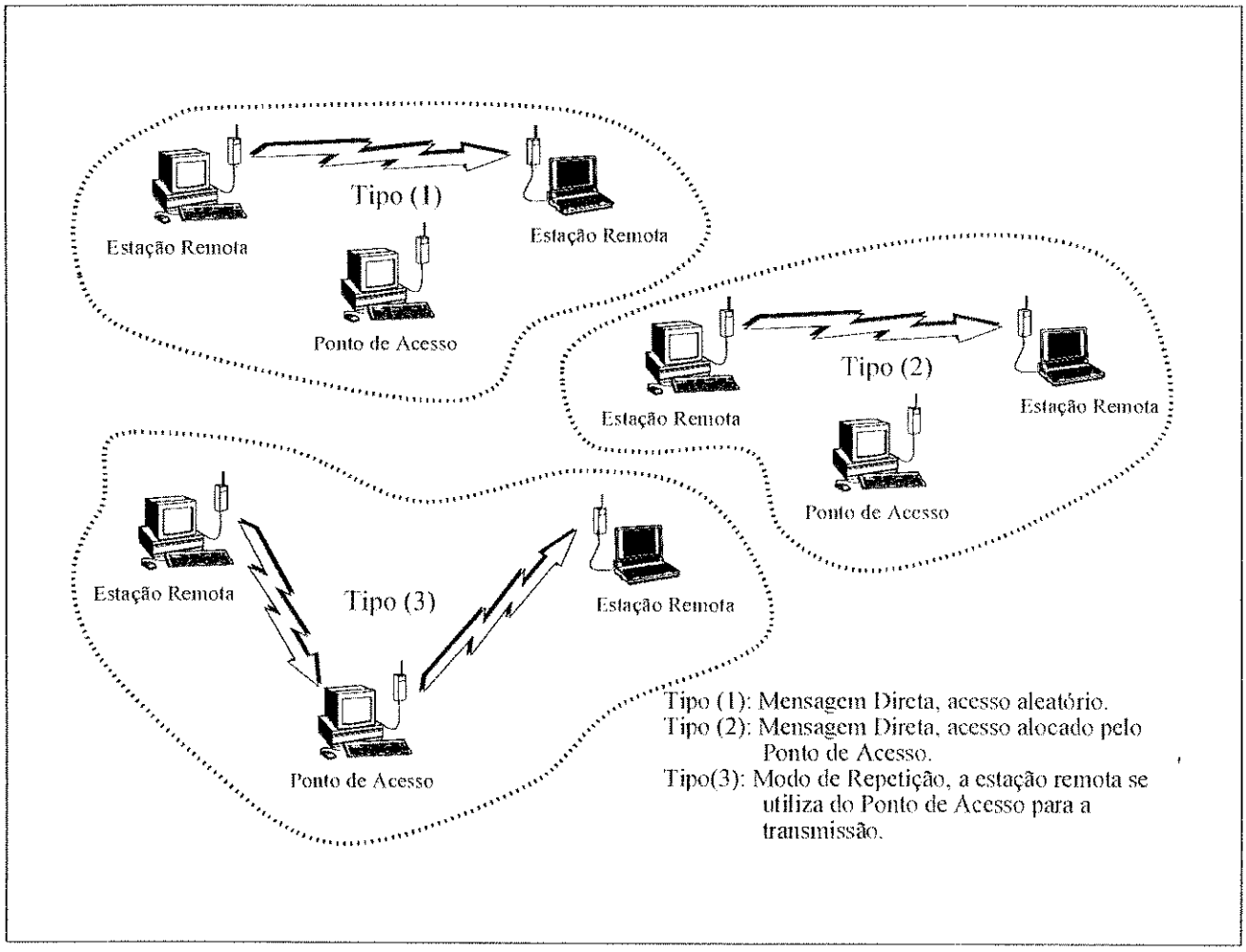


Figura 7. Tipos de transmissão remota-remota.

3.3 Análise do Protocolo de Acesso Proposto

3.3.1 Conjecturas sobre Desempenho do Protocolo Acesso Proposto

Este protocolo é dito hibrido por conter periodos que são previamente alocados (fases A e B) e período sujeito a contenções (fase C). Supondo que a fração do quadro de tempo usado pelas fase A e B seja f, a fração usada pela fase C será (1 - f). Assim a máxima vazão será 100(f + A (1 - f))% onde A é a máxima vazão do "slotted" ALOHA e vale 0,37 {4}. Se considerarmos f = 0,80, como sugerido pelo protocolo proposto, teremos uma máxima vazão de 87,4%. Obviamente a implementação do protocolo influi sobre a performance do mesmo. Na estimativa acima não se considera o quanto se perde em termos de vazão devido às mensagens de "overhead" (tais como cabeçalhos e "acknowlogements"). Assim, é de se esperar uma vazão inferior aos 87,4% citatos acima quando da implementação do protocolo.

3.3.2 Pontos Positivos do Protocolo Acesso Proposto

Citam-se aqui alguns pontos vantajosos do protocolo acesso proposto para redes sem fio.
Serviços Síncronos: *A implementação de transmissão de pacotes isócronos é parte integrante do protocolo.*
Implementação de Salto em Freqüência: *A cada quadro de tempo utiliza-se uma freqüência para transmissão dos dados. Isto permite a operação de várias células de rede sem fio no mesmo espaço físico visto que células adjacentes usam seqüências de freqüências ortogonais. Com a utilização das confirmações de mensagens recebidas é possivel para o Ponto de Acesso perceber se determinada freqüência de operação apresenta niveis de ruído muito elevados no demodulador. Uma vez comprovada a ineficiência de transmissão numa freqüência, ela pode ser retirada da seqüência operada pelo Ponto de Acesso.*
Consumo Racionalizado de Energia: *O fato da estação remota, pela informação saber previamente em que momento deve transmitir ou receber mensagens possibilita uma utilização racionalizada do consumo de energia. Uma vez que "laptops" e "notebooks" operam por baterias, é de grande valia a economia de energia nos periodos os quais o Rádio não está sendo utilizado. Neste caso, a observação do cabeçalho AH pode ser o único tempo em que o Rádio esteja ligado a fim de saber se há ou não mensagens para a estação remota naquele quadro de tempo.*

3.3.3 Pontos Negativos do Protocolo Acesso Proposto

Dois são os principais pontos negativos no protocolo de acesso proposto. O primeiro refere-se ao atraso em se obter acesso ao meio e o segundo refere-se à redução da vazão devido ao protocolo.
Tempo de Acesso ao Meio: *O tempo necessário desde o pedido de alocação de faixa até a transmissão efetiva da mensagem por uma das estações que compõem a rede é denomidado de tempo de acesso ao meio físico. No protocolo proposto, a requisição de faixa é feita na fase C e, na melhor das hipóteses, a alocação de faixa será feita no próximo quadro de tempo. Isto quer dizer que o tempo de acesso ao meio físico está diretamente ligado a duração do quadro de tempo.*
Redução de Vazão Devido ao Protocolo: *Os cabeçalhos das fases A, B e C, o tempo de salto em freqüência, a partição das mensagens nas fases e o recebimento de confirmação (ACK) após cada pacote transmitido são "overheads" impostos pelo protocolo proposto. Estes "overheads" reduzem efetivamente a vazão.*

3.4 Conclusões

Vimos neste capítulo como o protocolo de acesso ao meio físico proposto organiza seu fluxo de dados. Este protocolo também determina períodos de tempos, chamados de fases, que se repetem sucessivamente em quadros temporais. Os dados trocados via Rádio a cada um destes quadros temporais devem ser modulados em freqüências sucessivamente diferentes obedecendo assim a metodologia do Salto em Freqüência descrita no Capítulo 2.

Todas as determinações deste protocolo compõem as especificações para o hardware e software que o implementarão. Nos capítulos seguintes descreveremos uma proposta para o hardware que se proponha a implementar este protocolo.

3.5 Referências

{1} *K.S. Natarajan, C.C. Huang, D.F Bantz, "Medium Access Control Protocol for Wireless LANs", IEEE 802.11/92-39, Mar. 1992.*

{2} *A.S. Tanenbaun, "Computer Networks", Prentice-Hall, Inc., 1989.*

{3} *A.K. Budri, I.S. Bonatti, "Wireless LAN Protocols", 11° Simpósio Brasileiro de Telecomunicações, Natal, RN, Set. 1993.*

{4} *R.O. LaMaire, A. Krishna, H. Ahmadi, "Analysis of a Wireless MAC Protocol with Client-Server Traffic and Capture", IEEE J. Select. Areas on Communication, Out. 1994.*

Capítulo 4. Implementação da Plataforma em Hardware

Tendo visto, nos capítulos anteriores, os parâmetros a serem considerados numa rede local sem fio, assim como a descrição do protocolo de acesso proposto, apresentamos neste capítulo a proposta de um adaptador capaz de implementar este protocolo. Esta solução obviamente não é a única possível mas, por fazer uso de microprocessador, vai torná-la de uso geral possibilitando a implementação do protocolo em microcódigo.

4.1 Introdução

Em nossa proposta de implementação associaremos algumas entidades funcionais, às estações remotas ou Ponto de Acesso, de modo a facilitar a visão global da mesma. Estas entidades são mostradas Figura 8. Contamos, então, com uma entidade denominada Sistema de Processamento de Dados que pode ser associada a um PC ou estação de trabalho, outra entidade chamada de Controlador de Comunicação, associada ao cartão conectável a este PC ou estação de trabalho e, finalmente, o Rádio composto de controlador, transmissor e receptor.

O sistema de comunicação de dados, conforme proposto acima, divide-se em três entidades funcionais que serão descritos a seguir. A primeira entidade, Sistema de Processamento de Dados, origina as mensagens e as transmite para o Controlador de Comunicação. De maneira reversa, o Sistema de Processamento de Dados processa os dados recebidos pelo Controlador de Comunicação.

O Controlador de Comunicação é responsável por exercer o controle do Rádio, determinando os parâmetros para sua operação conveniente. Além disso, o Controlador de Comunicação é responsável por dar formato, pré-estabelecido pelo protocolo de acesso ao meio físico, aos dados recebidos do Sistema de Processamento de Dados a serem enviados ao Rádio. O Controlador de Comunicação também é responsável por interpretar informações recebidas do Rádio que sejam pertinentes ao controle da comunicação.

O Rádio por sua vez, recebe os dados do Controlador de Comunicação e os transmite em uma particular freqüência portadora. Também é responsável por receber dados transmitidos por outra estação e enviá-los ao Controlador de Comunicação. O Rádio não é capaz de interpretar os dados recebidos ou transmitidos que se relacionem com o controle de comunicação. Portanto, estas funções são desempenhadas pelo Controlador de Comunicação, que controla o Rádio a fim de iniciar e terminar as transmissões ou recepções de acordo com o estabelecido pelo protocolo de acesso ao meio.

O Sistema de Processamento de Dados corresponde a estação, seja remota ou Ponto de Acesso, e pode ser um PC, PS, "notebook" ou máquina de maior porte. O Controlador de Comunicação corresponde ao cartão inserido na estação e, o rádio, a uma unidade externa acoplada ao cartão por meio de um cabo.

Assim sendo, o cartão possuirá as seguintes interfaces com o meio externo: interface com o Sistema de Processamento de Dados e interface com o Rádio. A conexão com o Sistema de Processamento de Dados se dará através da interface descrita pelo padrão PCMCIA que, além de ser utilizado em "notebooks", também o é em computadores maiores (PC, PS ou

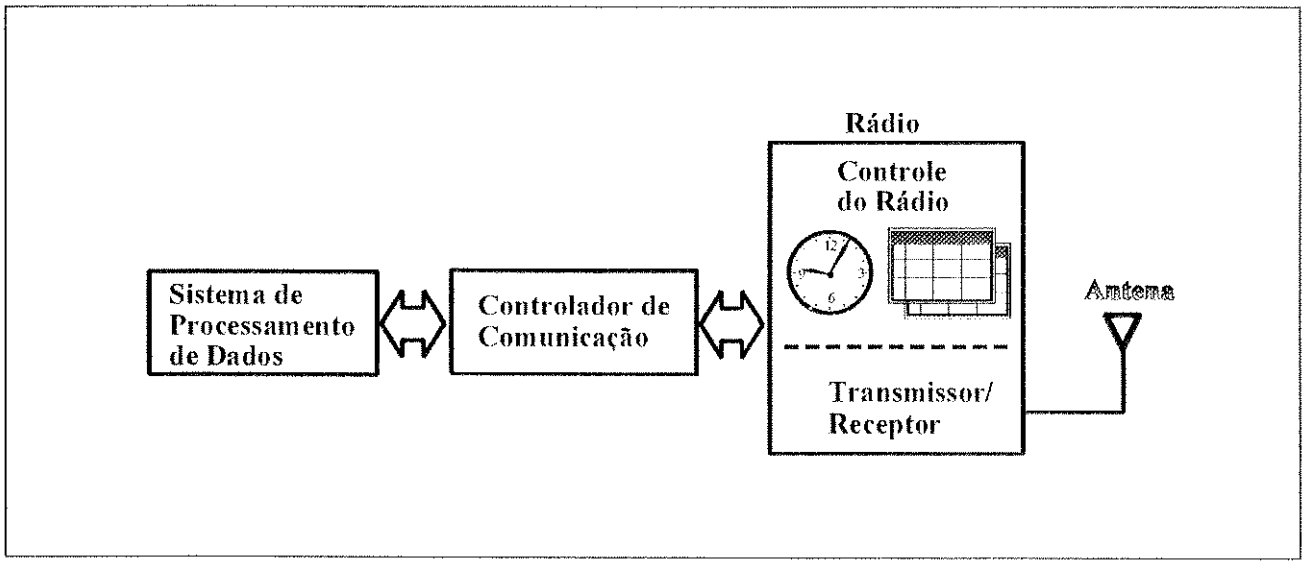


Figura 8. Diagrama das entidades funcionais do sistema de comunicação de dados.

estações de trabalho). A comunicação com o Rádio será realizada através de interface que inclui o caminho de dados e seus controles. A Figura 9 mostra o cartão e suas interfaces.

4.2 Blocos Funcionais

Vejamos agora os blocos funcionais que comporão o cartão. Vimos no item 2.5, que a compressão de dados é uma operação desejável em redes sem fio a fim de superar limitações nas taxas de transmissões impostas pelos rádios. Isto porque os rádios, tecnologicamente convenientes as aplicações de rede sem fio, limitam-se a transmissões de alguns megabits por segundo, enquanto que às redes com fio mais comumente utilizadas operam com 10 ou 16 megabits por segundo. Outra operação desejável é a encriptação e autenticação de mensagem também citadas item 2.5 como segurança de dados. A Figura 10 apresenta uma interligação dos blocos funcionais de compressão de dados e encriptação. Este diagrama de blocos mostra toda a plataforma hardware proposta que será discutida a seguir.

Por uma questão de desempenho, mantemos o processador em um subsistema à parte para que sempre possa executar suas instruções independentemente dos dados estarem sendo transmitidos ou recebidos simultaneamente. O subsistema do processador deve conter ao menos uma memória não volátil (tipo EPROM ou Flash EPROM), onde estará gravado o código a ser executado, e uma memória volátil (tipo RAM), a ser utilizada para armazenamento de variáveis de programa. Os temporizadores que atuam no subsistema do processador têm a função de marcar os tempos necessários à implementação do protocolo de acesso proposto (por exemplo, o fim de uma fase A, B ou C ou mesmo o fim do quadro de tempo).

Os dados vindos do Sistema de Processamento de Dados chegam através da interface PCMCIA e são depositados na memória compartilhada entre o Sistema e o cartão, que tem a função de compensar diferenças de velocidade de acesso entre um e outro. Mais adiante, após estes dados serem comprimidos e encriptados, eles são depositados na memória de dados a fim de esperar pelo tempo correto de envio. Um conversor série-paralelo-série (CSPS) é usado com a função de enviar os dados da memória em forma serial e de recebê-los do Rádio e paralelizá-los para serem depositados na memória de dados. A presença da

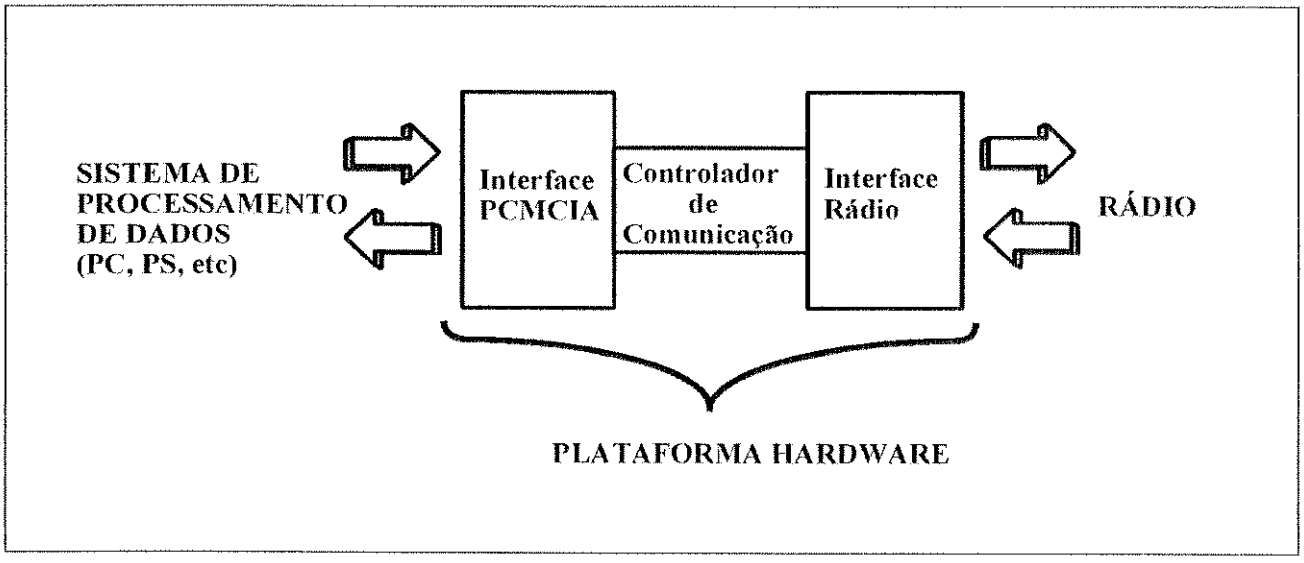


Figura 9. Interfaces da implementação hardware.

UART[6] *(contida no subsistema do processador) é necessária para o envio de comandos ao Rádio, de modo a controlá-lo e também sincronizá-lo ao Controlador de Comunicação. Nota-se a presença de três DMA's*[7]*. O primeiro é responsável pela transferência de dados pelo compressor, o segundo pelo encriptador e o terceiro pelo envio e recepção de dados do CSPS. O último bloco a ser descrito corresponde aos sinalizadores, que são utilizados pelo Sistema de Processamento de Dados e Controlador de Comunicação para troca de status e controles.*

A seguir analisaremos a transmissão e recepção de uma mensagem para melhor compreensão dos blocos funcionais.

4.3 Fluxo de Dados

Iniciaremos a descrição da transmissão de dados supondo que uma aplicação em funcionamento no Sistema de Processamento de Dados requer o envio de uma mensagem através da rede sem fio. Esta aplicação aciona o direcionador de dispositivo[8] *do adaptador e escreve a mensagem na memória compartilhada entre o Sistema e cartão. Após isso, a aplicação (novamente utilizando o direcionador de dispositivo) faz saber ao cartão, mais precisamente ao processador do subsistema, que a mensagem em sua memória compartilhada deve ser transmitida. Esta troca de mensagem é feita através dos sinalizadores. Em termos de organização de dados, esta ou qualquer mensagem trocada entre Sistema e cartão deve conter um cabeçalho com informações a respeito da mensagem. Estas informações são o endereço da estação origem, o endereço da estação destino e o seu comprimento. É justamente através desta última informação que o processador pode programar o DMA do compressor e o DMA do encriptador, a fim de transferir os dados da memória compartilhada para a memória de dados. É importante observar que, ao se transferir*

[6] do inglês "Universal Asynchronous Receiver-Transmiter".

[7] do inglês "Direct Memory Access".

[8] Comumente denominado de "device driver", que é uma entidade software operando com um com certo dispositivo.

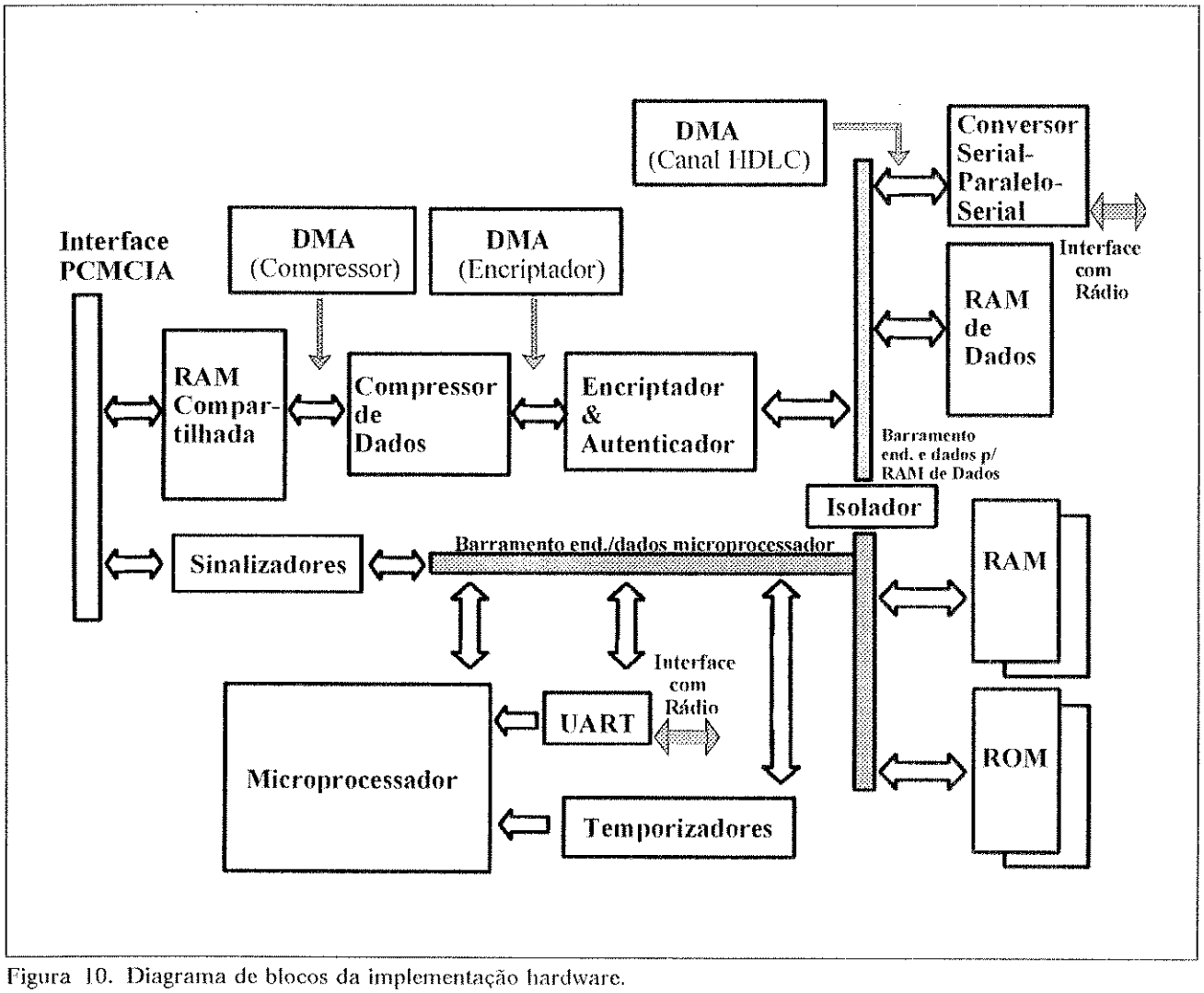


Figura 10. Diagrama de blocos da implementação hardware.

o cabeçalho da memória compartilhada para a memória de dados, o compressor e o encriptador devem estar desativados a fim de preservar as informações inteligíveis ao processador (note que o cabeçalho deve sempre possuir o mesmo comprimento de modo a permitir a programação conveniente dos DMA's de transferência).

Uma vez que o compressor de dados e o encriptador estejam com suas condições iniciais ajustadas, os DMA's são programados e habilitados para que a mensagem seja transferida da memória compartilhada para a memória de dados. Após a transferência, a mensagem está comprimida e encriptada cabendo ao processador anexar à mensagem a autenticação.

Durante a transferência descrita acima, o processador trata de programar os DMA's do encriptador de modo a segmentar a mensagem em pacotes conforme determina o protocolo de acesso proposto. A tarefa de particionar a mensagem em pacotes conforme especificado no protocolo de acesso proposto é, então, feita nesta etapa (vide descrição das fases A, B e C no Capítulo 3). A Figura 11 mostra o fluxo de dados no Controlador de Comunicação, enquanto que a Figura 12 mostra uma mensagem enviada pelo Sistema sendo armazenada na memória de dados já particionada e, finalmente, sendo enviada conforme especifica o protocolo de acesso ao meio físico proposto.

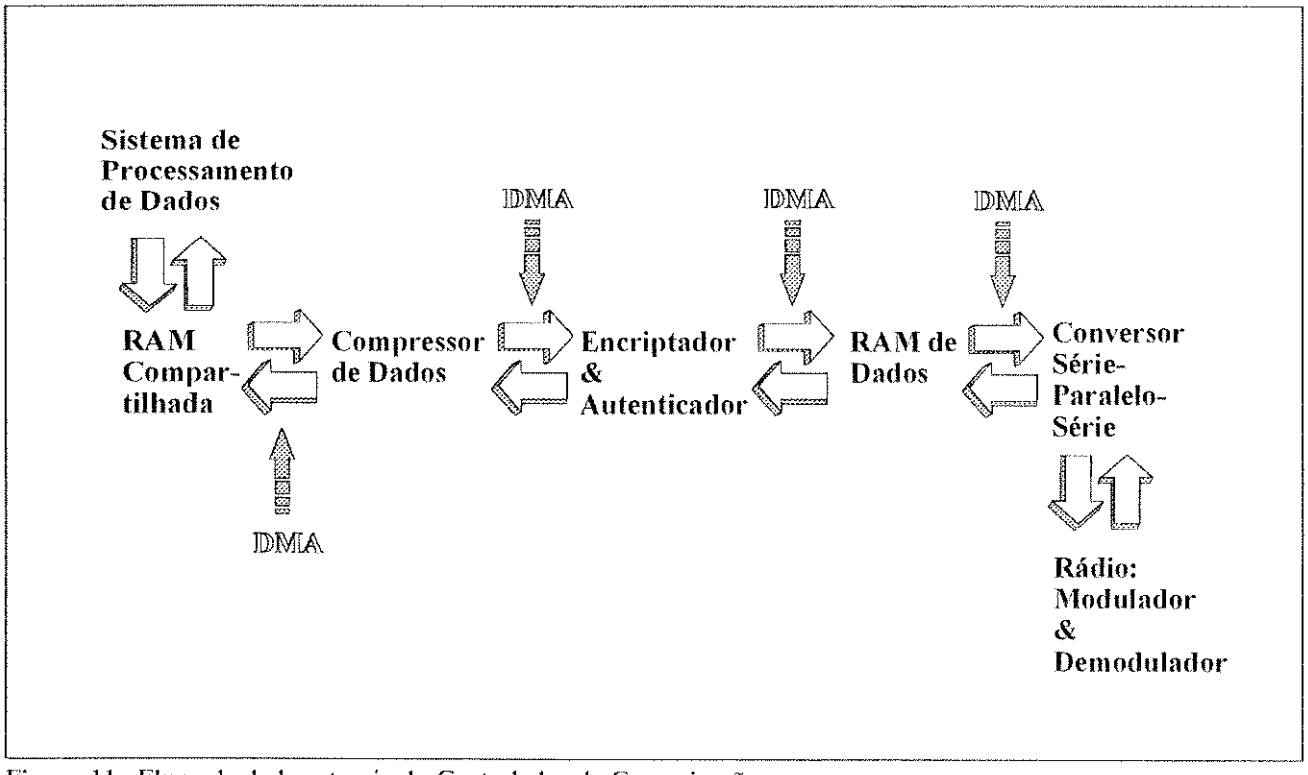


Figura 11. Fluxo de dados através do Controlador de Comunicação.

Fazendo uso dos temporizadores disponíveis no cartão, o processador envia comandos ao Rádio através da UART estabelecendo os períodos de recepção e de transmissão. Isto se faz de acordo com as especificações do quadro de tempo do protocolo. Ao constatar que a mensagem está pronta para ser transmitida na memória de dados, o processador aguarda a fase conveniente de transmissão, programa o DMA do serializador de dados e passa a transmiti-la ao Rádio que a modula e a envia. Note que em cada partição de tempo haverá um pacote de dados independente, com endereços de origem e destino e com código corretor de erro. A remontagem destes pacotes na recepção resultará na mensagem comprimida e encriptada.

Após o envio de cada pacote, o processador sinaliza o Rádio a fim de alterar seu estado de transmissão para recepção. Neste momento, o processador espera receber da estação destino a confirmação de que o pacote foi recebido integralmente. O não recebimento da confirmação acarreta no re-envio do pacote.

Na recepção, o processador monitora os temporizadores a fim de comandar o Rádio ao estado de recepção, quando o momento especificado pelo protocolo convier. Assim, o CSPS recebe os dados seriais vindos do Rádio e, para cada pacote, verifica o endereço de destino. Se for igual ao seu próprio endereço, ele passa a paralelizar os dados e os armazena em uma FIFO interna de três bytes. O esvaziamento desta FIFO é feito pelo DMA do CSPS que transfere os dados recebidos para a memória de dados. A programação deste DMA é feita pelo processador com base nas informações recebidas por ele nos cabeçalhos das fases A ou B do protocolo. Digamos, por exemplo, que no cabeçalho da fase A de um determinado quadro de tempo é informado que a n-ésima partição será destinada à estação receptora em questão. Através desta informação, o processador tem condições de, operando com os temporizadores, comandar o Rádio ao estado de recepção na n-ésima partição. Da mesma forma, o processador, através da informação do cabeçalho, tem como programar convenientemente o

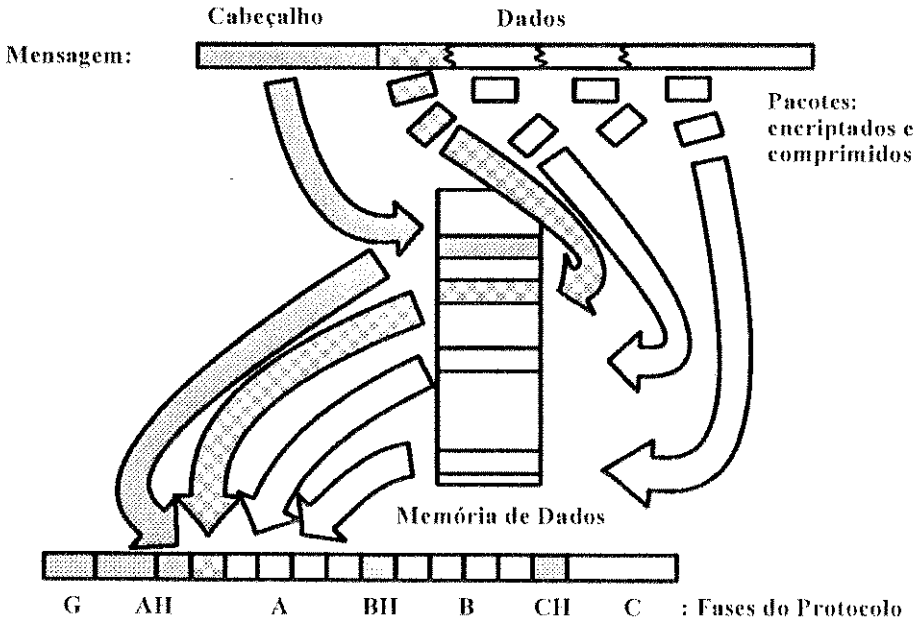


Figura 12. Organização dos dados durante a transmissão de mensagem.

DMA do CSPS. Por uma questão de simplicidade, se cada partição tiver sempre o mesmo número de bytes, a programação do estado do Rádio e do DMA do CSPS é sempre fixa no que diz respeito ao tempo em que o Rádio deve permanecer no estado de recepção e ao comprimento da mensagem a ser transferida.

Uma vez que a mensagem agora está contida na memória de dados, o processador[9] tem acesso ao seu cabeçalho (observe que agora trata-se do cabeçalho da mensagem e não do protocolo). Com estas informações, ele programa o compressor, encriptador e seus DMA's para realizarem a descompressão e desencriptação, respectivamente. A seguir, sinaliza ao Sistema de Processamento de Dados que existe uma mensagem a ser enviada. A aplicação que estiver sendo executada no Sistema será interrompida e responderá ao cartão para iniciar o envio da mensagem. O processador habilita os DMA's e os dados passam a ser colocados na memória compartilhada ao mesmo tempo que são lidos pelo Sistema. No final da transferência a autenticação é verificada pelo processador. Caso a autenticação esteja correta, ela é eliminada e a mensagem é considerada correta. Caso contrário, o processador sinaliza para a aplicação do Sistema que a mensagem recebida deve ser descartada.

[9] Muitas vezes a palavra processador refere-se ao conjunto processador e seu código. Neste caso entenda-se "o código sendo executado pelo processador".

4.4 Conclusões

Neste capítulo vimos uma possível implementação hardware que atende às necessidades do protocolo de acesso proposto e também às exigências de segurança (encriptação) e taxa de transferência (compressão) das redes sem fio.

Esta implementação, definida pelos seus blocos funcionais, determina completamente a arquitetura a ser seguida para a plataforma hardware, onde deve-se ter um ambiente apropriado para suportar tanto o protocolo de acesso proposto, como encriptação e compressão de dados. Ao hardware cabe, portanto, a responsabilidade de dispor dos dados convenientemente formatados no instante de tempo planejado pelo software. Tal implementação hardware proposta tem os méritos de aliviar o processador de tarefas como encripação, compressão e formatação dos dados, dando-lhe mais tempo para execução de tarefas relacionadas com a programação das entidades de hardware.

Nos capítulos seguintes descreveremos cada um dos blocos funcionais que compõem a rede sem fio em análise. Assim, serão descritos o compressor de dados, encriptador, autenticador, interface com Sistema de Processamento de Dados (PCMCIA) e interface com Rádio.

4.5 Referência

{1} *N.A.T.Resende, "Wireless/PCMCIA Card Functional Specification", Publicação Interna IBM, Centre d'Etude et de Recherche - La Gaude, França, Out. 1993.*

*

Capítulo 5. Encriptação e Autenticação

Discutiremos neste capítulo uma possível implementação para encriptação e autenticação de dados em uma rede sem fio. Nosso principal objetivo é a possibilidade de implementação de tais blocos funcionais em hardware com simplicidade e eficiência.

5.1 Encriptador

5.1.1 Introdução

O sistema criptográfico aqui proposto objetiva atender as necessidades de uma rede sem fio no que diz respeito à segurança dos dados e detecção de interferências intencionais na rede.

O sistema consiste em um gerador pseudo-aleatório que gera uma seqüência de bits a ser usada como dados encriptados. Devido ao objetivo de implementá-lo em hardware, este gerador deve ser simples mas também eficiente. Iniciaremos nossa discussão com os tipos comumente usados de encriptadores e no transcorrer do texto vamos particularizá-lo a nossa aplicação. Em geral utiliza-se geradores que nada mais são que combinações de registros de deslocamento realimentados.

A fim de realizar a encriptação no caso de troca de mensagens entre estações, devemos agir sobre o fluxo de dados, de modo a codificá-los antes de serem transmitidos e decodificá-los assim que recebidos. Estes chamados sistemas criptográficos são formados por um dispositivo, o gerador pseudo-aleatório, que gera uma seqüência a partir de um conjunto inicial de bits.

Chamemos de "padrão de encriptação" a seqüência gerada e de "semente" a seqüência inicial. Dada uma mensagem M, um padrão P, com o mesmo comprimento que M, é gerado pelo sistema criptográfico. Para se encriptar a mensagem, para cada bit de M realiza-se um ou-exclusivo com o respectivo bit do padrão P. Em outras palavras, se $M_0,M_1,...,M_k$ são bits da mensagem e $P_0,P_1,...,P_k$ os bits do padrão, o texto cifrado consiste então nos bits $M_0 + P_0, M_1 + P_1,..., P_k + M_k$, onde + simboliza a operação lógica de ou-exclusivo (XOR). A desencriptação é possível através da mesma operação tomando-se o texto cifrado e os mesmos padrões de encriptação.

Para se obter os mesmos padrões de encriptação, o encriptador e desencriptador utilizam a mesma semente inicial. Obviamente, estes geradores pseudo-aleatórios devem ser idênticos e, portanto, gerar os mesmos padrões de encriptação. Desta forma, podemos dizer que a complexidade e qualidade de um sistema criptográfico dependem apenas do seu gerador de padrões.

Considerando-se os padrões gerados como verdadeiramente aleatórios, a encriptação, conforme descrita acima, seria perfeita. Isto porque a obtenção da mensagem original a partir da mensagem encriptada seria impossível. Assim, necessita-se que o padrões sejam o mais próximo possível do aleatório, o que chamamos de pseudo-aleatório. Deseja-se, portanto, que os padrões sejam o mais difícil possível de serem previstos e que alguém, mesmo tendo acesso à mensagem encriptada, tenha o máximo de dificuldade em reconstruir o padrão de encriptação.

Um bom gerador de padrões de encriptação deve, pelo menos, evitar as leis de formação baseadas na linearidade das seqüências geradas. Outras propriedades são também desejáveis no aspecto de implementação. É necessário que ele seja simples para resultar numa fácil e eficiente implementação em hardware e, ao mesmo tempo, suportar altas taxas de transmissão (como, por exemplo, milhões de bits por segundo).

5.1.2 Gerador de Padrões de Encriptação

O gerador aqui proposto compõe-se de estruturas às quais chamamos de Registro de Deslocamento com Realimentação Linear (ou RDRL). Por sua vez, esta estrutura constitui-se de um registro de deslocamento cuja entrada é o resultado da operação lógica de ou-exclusivo entre algumas posições deste registro. Note que chamamos de registro de deslocamento o registro que, a cada pulso de relógio, desloca o bit de sua entrada para a sua posição primeira e desloca sucessivamente todos os demais bits para as posições seguintes. A Figura 13 apresenta o esquema de um RDRL.

As seqüências produzidas por RDRL's apresentam características desejáveis em se tratando de geração de padrões para encriptação. A primeira delas diz respeito ao longo periodo do padrão gerado, o que evita que mensagens diferentes sejam encriptadas com a mesma parte do padrão. Além disso, estas seqüências possuem distribuição de bits balanceada, o que previne que a grande maioria dos bits de uma mensagem não se modifiquem após a encriptação. Como podemos ver na Figura 13, outra característica dos RDRL's é sua fácil implementação. Além disso, este tipo de estrutura permite uma implementação paralelizada, o que quer dizer que em apenas um pulso de relógio podem-se calcular vários bits à frente.

No entanto, devido à linearidade e à periodicidade dos padrões gerados pelos RDRL's, é possível descobrir toda a estrutura de realimentação do registro de deslocamento e a semente utilizada a partir da mensagem encriptada. Este fato é suficiente para não utilizarmos os RDRL's em sua estrutura convencional, mas sim compô-los de modo a obtermos uma robustez maior na encriptação. Isto caracteriza justamente a estrutura de encriptação sugerida e descrita a seguir.

O esquema de encriptação proposto é mostrado na Figura 14. Notamos a presença de dois RDRL's, denominados de C (chave) e S (seletor). Cada qual é carregado com sementes específicas, usualmente diferentes entre si e aleatórias, que nada mais são que os valores iniciais dos registros de deslocamento.

Chamemos as seqüências produzidas pelos RDRL's de seqüência C e seqüência S. Se o n-ésimo bit produzido por S é '1', então o n-ésimo bit produzido por C é concatenado ao padrão de encriptação. Caso contrário, o n-ésimo bit de C é descartado. Desta forma, se a seqüência C é composta dos bits $c_1 c_2 \ldots$ *e a seqüência S é composta dos bits* $s_1, s_2, \ldots$ *então o padrão P gerado é definido por:*

$p_k = c_{ik}$, *onde* i_k *é a posição do k-ésimo bit '1' da seqüência S.*

Omitindo-se os bits gerados por C de maneira pseudo-aleatória destrói-se a linearidade do RDRL utilizado com chave de encriptação. Assim, apesar da destruição da linearidade, mantém-se as propriedade desejáveis do RDRL C.

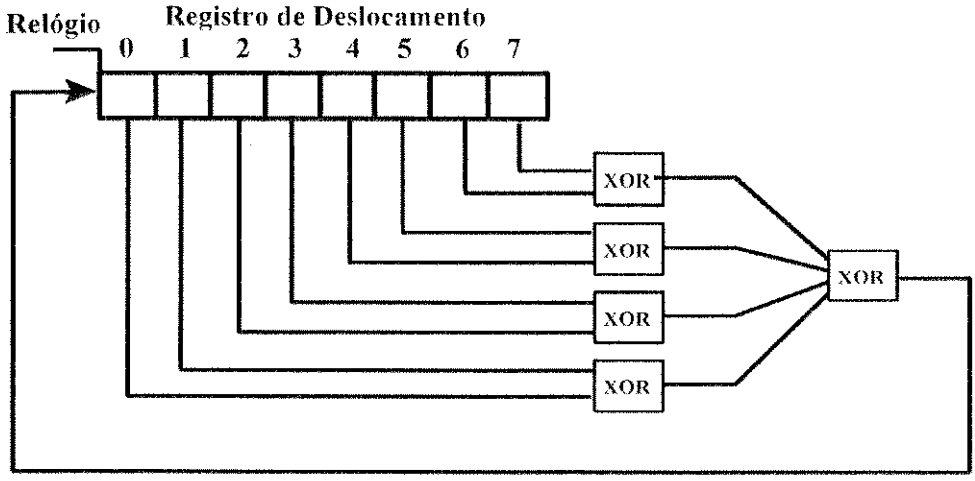


Figura 13. Esquema de um Registro de Deslocamento com Realimentação Linear.

5.1.3 Implementação

Observando a Figura 14, notamos que o registro de controle determina quais posições de cada registro de deslocamento estão habilitadas para compor a realimentação. Estas posições constituem parte da chave de encriptação, em conjunto com as sementes de C e S. Obviamente, a alteração periódica das conexões de realimentação irá requerer um esforço maior para quem deseja descobrir a estrutura de encriptação utilizada. Além disso, mesmo que descoberta a estrutura, as conexões de realimentação podem ser alteradas de forma a não se permitir a obtenção do padrão de encriptação.

As conexões são feitas combinando-se os bits de registro de controle com os bits do registro de deslocamento. Assim, o n-ésimo bit do registro de controle é combinado através de uma porta lógica E com o n-ésimo bit do registro de deslocamento. Todas as saídas dessas portas E são conectadas a uma porta ou-exclusivo que gera o bit de entrada do registro de deslocamento. Pode-se dizer que o registro de controle age como uma máscara para as conexões da realimentação.

Podemos representar as conexões de realimentação dos RDRL como polinômios sobre GF(2)[10]*. Isto pode ser feito através da correspondência entre posições dos registros de controle e os coeficientes do polinômio. Por exemplo, se a n-ésima posição do registro de controle for '1', então o coeficiente de x^n também será '1'.*

Como já foi dito antes, as seqüências geradas pelos RDRL's possuem um período de repetição por não serem verdaderamente aleatórias. Neste caso, tanto melhor será o encriptador quanto maior for o período da seqüência gerada pelo RDRL. Chamamos de RDRL de máximo comprimento aqueles que possuem a seqüência de maior período possível, ou seja, um RDRL

[10] Campo de Galois de ordem 2.

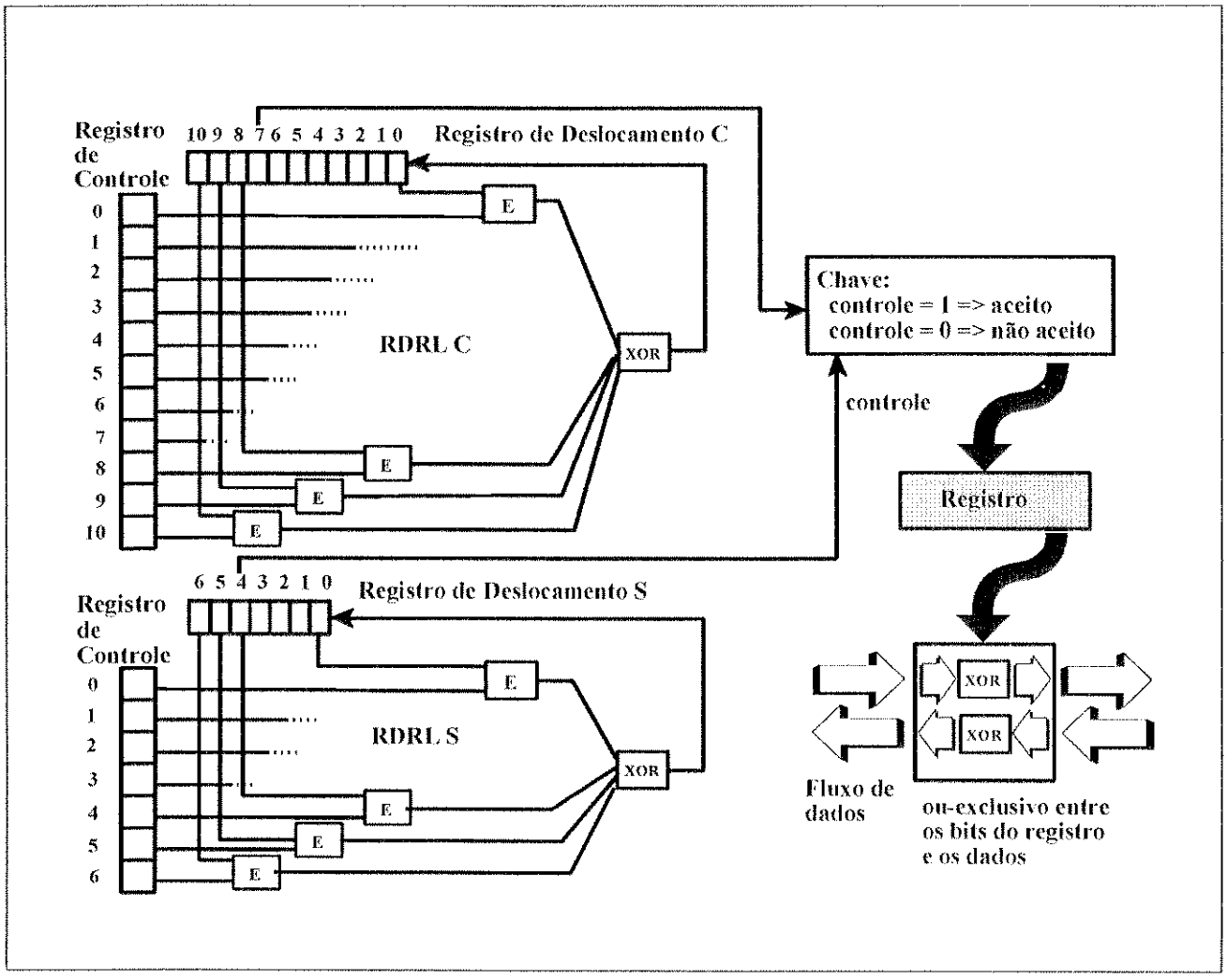


Figura 14. Implementação do circuito de encriptação.

cuja seqüência de saída tenha período $2^n - 1$, sendo n o comprimento do registro. Uma condição necessária e suficiente para o máximo comprimento é que a seqüência de conexão corresponda a um polinômio primitivo (sobre GF(2)) {1}. Uma outra condição para o máximo comprimento, mas nem sempre suficiente, é de se ter um polinômio irredutível. Esta última condição é mais fácil de se testar que a condição de polinômio primitivo. Além disso, para alguns comprimentos de registros ela é suficiente para garantir seqüências de máximo comprimento. Estes comprimentos são chamados de bases de Mersenne, isto é, números primos L tais que $2^L - 1$ também seja primo {1}. As primeiras bases de Mersenne são 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521. Escolhendo-se, portanto, registros com estes comprimentos, e escolhendo-se a conexão polinomial de forma aleatória de maneira que estes polinômios sejam sempre irredutíveis, cumpre-se o objetivo de máximo comprimento para o padrão de encriptação.

De maneira a dificultar um intruso reproduzir a seqüência a partir do mesmo gerador de padrões, sugere-se tomar como saída do RDRL qualquer bit diferente do bit mais significativo ou do bit menos significativo (que é justamente a saída da porta "ou-exclusivo").

Considerações a Respeito das Características dos RDRL's

Sobre o Comprimento dos Registros: *Veremos agora alguns requisitos sobre a escolha dos comprimentos dos registros que compõem os RDRL's. Como vimos anteriormente, os comprimentos de S e C devem ser escolhidos de acordo com as bases de Mersenne, de modo a garantir seqüências de máximo comprimento.*

O objetivo é obter padrões com grande complexidade além de máximo comprimento de seqüência. Notamos que o principal papel de S relaciona-se com a complexidade do padrão gerado, enquanto que C relaciona-se com as propriedades pseudo-aleatórias de uma seqüência RDRL típica. Assim, quanto maior o comprimento de C, em comparação ao comprimento de S, mais estarão presentes as propriedades pseudo aleatórias do padrão gerado.

Sobre Aplicações em Tempo Real: *Observando o funcionamento do gerador de padrões, composto por dois RDRL's C e S alimentados pela mesma base de tempo, constatamos que ambos produzem um bit a cada ciclo do relógio. Salientando-se que o bit gerado por C é aceito ou não como componente do padrão de encriptação, dependendo do bit gerado por S, conclui-se que a geração do padrão é feita a uma taxa que depende do número de 1's gerados por S. Assumindo que a distribuição de 0's e 1's provenientes de S é balanceada, pode-se dizer que, em média, um bit do padrão é produzido a cada dois ciclos de relógio. Isto resultaria numa encriptação cuja taxa é de, em média, metade da freqüência utilizada pelo relógio.*

Nota-se que, na descrição acima, fizemos uso da expressão "em média" de modo que pode haver períodos no qual o RDRL S gere uma seqüência consecutiva de 0's, evitando por determinado tempo a geração do padrão de encriptação. Assim, o que se faz para compensar a ocorrência abundante de 0's é beneficiar-se de ocorrências abundantes de 1's. Desta forma, os bits do padrão de encriptação são armazenados num registro antes de serem utilizados, permitindo assim que seqüências consecutivas de 0's não impeçam a encriptação. Graças ao bom balanceamento da distribuição de 0's e 1's das seqüências geradas pelos RDRL's, este mecanismo garante, com grande probabilidade, que os atrasos causados pela falta de 1's nas seqüências geradas por S sejam pequenos.

Uma outra técnica evita quase que completamente qualquer atraso na geração do padrão de encriptação. Ela é chamada de reciclagem de bits. Supondo um registro adicional no qual sejam armazenados todos os bits de C não utilizados no padrão de encriptação, ao ocorrer a situação de atraso por falta de bits do padrão de encriptação, utiliza-se os bits deste registro. A razão do uso deste registro e não a utilização dos bits consecutivos gerados por C, é justamente evitar que muitos bits consecutivos da seqüência do RDRL C sejam externados. Ou seja, não se compromete a segurança do sistema de comunicação quando utilizam-se os bits gerados por C contidos no registro adicional. Esta técnica visa obter uma taxa de um bit de dado encriptado para cada dois ciclos de relógio.

Por fim, outra forma de se conter os atrasos na geração de padrões de encriptação consiste em fazer uso do paralelismo para a geração dos bits dos RDRL's nas implementações. Por exemplo, pode-se implementar um RDRL que gere 4 bits a cada ciclo de relógio. Pode-se então, combinar a reciclagem de bits com o paralelismo na geração dos bits dos RDRL's e ainda usar um registro para armazenamento dos padrões já gerados. Assim, quando ocorrer uma seqüência de 0's consecutivos em S, a probabilidade de atraso na encriptação será muito pequena.

Sobre o Ajuste na Segurança de Dados: *Pode-se dizer que da maneira com que o gerador é proposto, ele pode ser ajustado para diferentes graus de segurança sem alterações no hardware. Esta característica pode ser utilizada, por exemplo, para deixar o encriptador em conformidade com leis de importação e exportação que vigoram em diferentes países, a fim de se evitar sistemas de comunicação extremamente seguros.*

Observe que as conexões do registro de deslocamento (C ou S) são, por assim dizer, mascaradas pelo conteúdo do registro de controle. Em outras palavras, o conteúdo do registro de controle estabelece a conexão ou desconexão do bits do registro de deslocamento na árvore de realimentação. Selecionar, portanto, a desconexão dos bits mais significativos equivale a reduzir efetivamente o comprimento do registro de deslocamento. Esta redução diminui a complexidade linear e também o período da seqüência gerada.

Sobre as Chaves de Encriptação: *Uma vez conhecida a arquitetura do sistema de encriptação, não se perde necessariamente a segurança dos dados. Os valores iniciais dos registros de deslocamento, que são as sementes dos RDRL's, e os valores contidos nos registros de controle, que são as máscaras, são conhecidos como chaves, às quais se deve a segurança de dados. Obviamente que estas chaves devem ser compartilhadas pelas duas partes em comunicação. A geração das sementes pode ser feita, por exemplo, utilizando-se qualquer gerador pseudo aleatório, porém de baixa correlação entre as seqüências geradas. Esta geração de sementes, no entanto, não é objetivo deste trabalho.*

Na prática, trocam-se mais freqüentemente as sementes que as conexões da estrutura de realimentação. A atualização das sementes pode ser útil às necessidades de sincronização dos geradores pseudo-aleatórios que compõem as partes em comunicação, enquanto que a troca das conexões irão depender das necessidades de segurança do sistema.

5.2 Autenticador

5.2.1 Introdução

A intenção em se autenticar uma mensagem é de, na sua recepção, poder verificar sua integridade. Em outras palavras, garantir, com alta probabilidade, que a mensagem enviada é a mesma recebida. Além disso, uma desejável característica da autenticação é dar uma assinatura a determinada mensagem, de tal forma que apenas componentes autorizados de uma rede de comunicação possam construir e verificar essa assinatura.

Em geral, seqüências de verificação geradas e anexadas a mensagens não possuem propriedades de segurança, sendo simplesmente um mecanismo para detectar alterações involuntárias nas mensagens enviadas. O que se deseja, portanto, é evitar que algum intruso modifique o conteúdo das mensagens sem ser detectado. Os mecanismos padrões, tais como CRC[11]*, não são capazes de fazer esse tipo de detecção uma vez que eles são fixos em sua geração e seu mecanismo extremamente divulgado, permitindo-se assim ao intruso autenticar sua própria mensagem.*

[11] do inglês "Cyclic Redundancy Code".

Métodos de encriptação não resolvem o problema da autenticação uma vez que é muito fácil se alterar a mensagem mesmo após a encriptação. Tomemos, por exemplo, uma mensagem M e sua correspondente encriptação C, ou seja, C = E(M). Então, C + M' é a encriptação de M + M', uma vez que C + M' = E(M + M'), onde + representa a operação lógica de "ou exclusivo". Assim sendo, ao se interceptar a mensagem encriptada M, o intruso pode facilmente modificá-la para M + M' sem ser detectado pelo algoritmo de desencriptação.

Um outro ponto favorável à autenticação, quando realizada antes da encriptação e verificada após a desencriptação, é verificar a correta desencriptação. Isto garante uma perfeita sincronização entre transmissor e receptor em se tratando de sistemas de encriptação dinâmicos (isto é, com alteração de chave de encriptação).

Os requisitos para a autenticação num sistema de transmissão, com ou sem fio, são, primeiro, suportar altas taxas de transmissão e, segundo, ser simples a fim de permitir baixa complexidade e, conseqüentemente, baixo custo na implementação. O método proposto a seguir satisfaz a ambos os requisitos acima por, utilizar registros de deslocamento (também propostos no sistema de encriptação).

É interessante notar que implementações baseadas em CRC possuem limitações, uma vez que são usualmente definidas em padrões. No sistema aqui proposto, estas limitações não estão presentes e, além disso, podemos fazer uso do conhecimento existente para implementações de geração rápida de CRC. Queremos dizer que, por não estar ligada a nenhum padrão previamente definido, a implementação aqui proposta pode fazer uso de técnicas como o paralelismo, a fim de acelerar o computo de resíduos.

5.2.2 O Mecanismo de Autenticação

As estações envolvidas na comunicação compartilham suas chaves de encriptação utilizadas na função de encriptação E(.). A seguir, elas compartilham um polinômio, chamado h(x), de grau n, sobre GF(2). Este polinômio h(x) é escolhido aleatoriamente com probabilidade uniforme entre todos os polinômios irredutíveis de grau n sobre GF(2). O valor de n é escolhido por implementação de acordo com o nível de segurança desejado.

Seja, portanto, M a mensagem de comprimento L constituída da seqüência de bits $M_0, M_1, \ldots, M_{L-1}$. *Seja* $M(x) = x^L + M_{L-1}x^{L-1} + \ldots + M_0$ *o polinômio de grau L sobre GF(2) com o primeiro coeficiente igual a 1 e os demais correspondentes aos bits da mensagem M.*

A transmissão de mensagem e autenticação é descrita a seguir. Calcula-se o resíduo da divisão do polinômio $M(x).x^n$ *por h(x). Seja r(x) este resíduo e r(M) a seqüência de bits composta dos coeficientes deste resíduo. Encripta-se a seqüência de r(M) usando-se a função E(.) para se obter E(r(M)) e, finalmente, transmite-se a mensagem M, encriptada ou não, e a seqüência E(r(M)).*

A partir da transmissão de M e E(r(M)), supomos que o receptor recebe as seqüências concatenadas M' e E'. Note que na desencriptação deve se obter r(M). Chamemos esta seqüência resultante da concatenação de M' e E' de C e associemos a ela um polinômio C(x), cujo primeiro coeficiente 1 e os demais componentes compostos pela seqüência C. Se o resíduo da divisão de C(x) por h(x) for igual a 0, a autenticação está correta, caso contrário, a mensagem deve ser desconsiderada.

Caso a mensagem deva ser encriptada, a autenticação r(M) é computada antes da encriptação. Ao serem encriptadas, r(M) e M são tomadas como uma só seqüência. Da mesma forma, na recepção, a desencriptação é feita sobre a seqüência concatenada e, em seguida, é feito o teste de integridade da mensagem. Neste caso, observa-se que quanto mais seguro é o sistema de encriptação, mais confiável (em termos de detecção de alterações de mensagem) é o mecanismo de autenticação.

5.2.3 Implementação

A seguir descreveremos o funcionamento da geração e da verificação da autenticação. O que pretende-se fazer é a divisão do polinômio associado a $M(x).x^n$ por $h(x)$. Para tal, utiliza-se um registro de deslocamento onde são armazenados os dados ($M(x)$), um registro de controle (que contém os coeficientes de $h(x)$) e, finalmente, um acumulador onde será armazenada a autenticação. Observando a Figura 15, vemos que os bits de dados fluem através do acumulador, porém combinados com os bits de realimentação do mesmo acumulador, caso não mascarados pelo registro de controle. O acumulador deve ser iniciado com 1 em seu bit mais significativo e 0 nos demais. A partir de então, os bits do acumulador são reavaliados $L + n$ vezes, onde L é o comprimento da mensagem e n o comprimento do acumulador e do registro de controle. Após essas $L + n$ iterações, o acumulador conterá o valor da autenticação.

Para se verificar a autenticação, o mesmo hardware é utilizado. Neste caso, o registro de deslocamento passa a receber a mensagem concatenada com a autenticação. O acumulador novamente é iniciado com 1 apenas em seu bit mais significativo. Da mesma forma que na geração, são necessários $L + n$ iterações para se concluir a verificação da autenticação. Após isso, caso o acumulador contenha todas suas posições iguais a 0, a mensagem é considerada como válida.

Para economia de tempo de processamento, o acumulador pode ser inicializado com os $n - 1$ primeiros bits da mensagem recebida. Assim, apenas $L + 1$ iterações são necessárias para se concluir a verificação.

Na implementação proposta, o polinômio $h(x)$ deve ser escolhido por software através de técnicas para teste de irredutibilidade. Deve-se, no entanto, implementar em hardware o acumulador e o registro de controle com comprimento (n) igual a um número primo, a fim de facilitar o teste de irredutibilidade de $h(x)$. Cabe acrescentar que o próprio polinômio $h(x)$ pode ser implementado em hardware (valores do registro de controle sendo fixos), resultando em maior simplicidade e economia. No entanto, uma vez encontrado esse polinômio, a função de segurança proposta na autenticação seria perdida.

5.2.4 Comparações com Métodos de Checagem de Mensagem e Assinatura

O método utilizado combina dois outros métodos conhecidos e utilizados para resolver problemas diferentes. Um é o método CRC (Cyclic Redundacy Code, {4}) utilizado largamente para detecção de erro em redes de comunicação. O outro método é o de função de assinatura proposto por M. Ranbin em {3}. Ambos os métodos, assim como o aqui proposto, executam divisão polinomial como sua operação básica de autenticação.

A seguir colocamos as principais diferenças entre os métodos descritos acima.

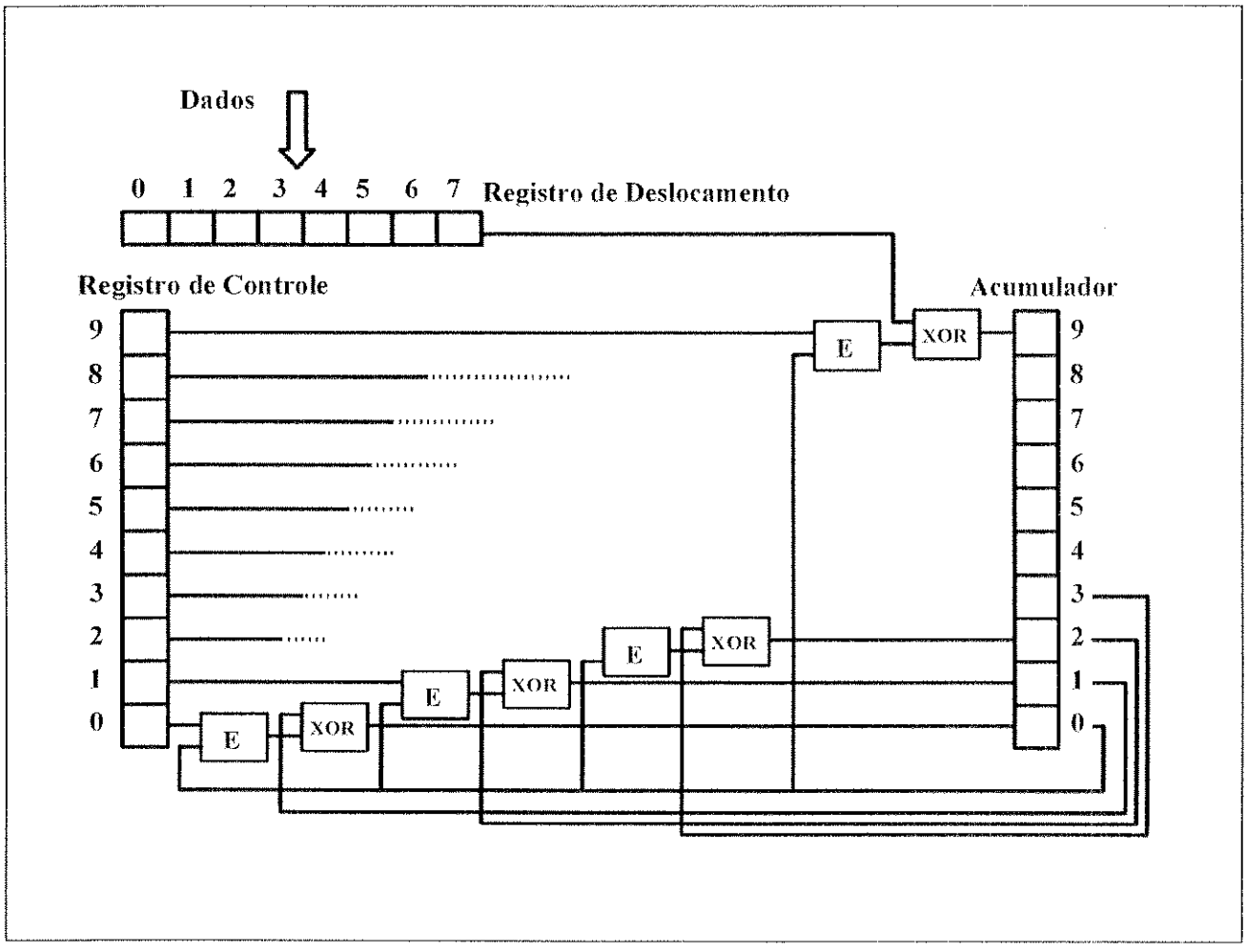


Figura 15. Implementação do circuito de autenticação.

Diferenças entre o Método Proposto e o Método de CRC

1. *Utilizamos um polinômio h(x) variável que pode ser programável no registro de controle. O método de CRC tem esse polinômio fixo.*
2. *O resíduo é encriptado no método utilizado, o que não acontece no do CRC.*
3. *O método utilizado pode, desde que mantido o sigilo de encriptação, detectar alterações propositadas nas mensagens enquanto que o método de CRC não permite qualquer detecção.*

Diferenças entre o Método Proposto e o Método de Rabin

1. *O polinômio correspondente ao dado autenticado é multiplicado por x^n antes da divisão na transmissão. Esta operação é essencial para a segurança caso a função de encriptação for simplesmente aditiva binária.*
2. *O resultado da autenticação é encriptado. No método proposto por Rabin, a assinatura não é transmitida e portanto não há necessidade de encriptá-la.*

5.3 Conclusões

Vimos neste capitulo os blocos funcionais de encriptação e autenticação que integram a solução de rede sem fio proposta. Ambos os blocos são concebidos de modo a possuirem uma implementação hardware fácil, o que não necessariamente implica em economia das portas lógicas que os constituem. Assim, dependendo do grau de segurança que se deseja para a rede de comunicação, estes blocos podem assumir tamanhos consideráveis. No entanto, sua implementação em hardware é extremamente favorável se comparada com sua possível implementação por software. Os cálculos da encriptação e autenticação tomariam tanto tempo do processador que tornaria a solução proposta inviável.

Outro ponto importante na implementação destes blocos funcionais é a capacidade da rede sem fio de suportar taxas de transmissão de uma rede convencional. Isto se deve, principalmente, ao fato do encriptador se colocar no caminho do fluxo de dados e, portanto, não poder ser um gargalo para o Controlador de Comunicação. Vimos, para a encriptação, algumas técnicas para combater o atraso na geração dos dados criptografados, tais como reciclagem de bits, paralelismo de geração e registro adicional para armazenamento dos padrões já gerados. Na prática, estas técnicas nos permitem atingir 1 bit encriptado a cada dois ciclos de relógio. Isto nos leva a dimensionar uma freqüência de relógio que atenda à taxa de transmissão requerida. Se, para atender à taxa de transmissão do Rádio, necessitamos de uma vazão de pelo menos 1 Mbps, o relógio deve então utilizar uma freqüência necessariamente maior que 2 MHz.

Diferentemente da encriptação, o método de autenticação proposto não possui atrasos além dos inerentes às portas lógicas envolvidas na implementação hardware. Note que o autenticador não se posiciona no caminho dos dados, mas se utiliza destes para gerar uma seqüência a ser posteriormente anexada à mensagem. Cabe observar que, depois de terminada a mensagem, são necessários mais 2.n (n sendo comprimento do acumulador) pulsos de relógio para se finalizar a autenticação. No entanto, esta demora em dispor da autenticação não é problema para o fluxo de dados, uma vez que ela será anexada à mensagem pelo processador. Assim, o tempo de finalização da autenticação não deve ultrapassar o tempo de um ciclo de leitura do processador, a fim de não provocar atrasos adicionais no fluxo de dados.

Por fim, devemos mencionar a versatilidade da implementação proposta, no que se refere ao grau de segurança e à complexidade de encriptação e autenticação. Os registradores de controle que fazem parte tanto do bloco funcional de encriptação quanto de autenticação são programáveis pelo processador, de modo que está ao alcance do código determinar o grau de segurança desejado. Esta versatilidade pode permitir uma alteração deste grau de segurança até mesmo pelo usuário.

5.4 Referências

{1} *K. Zeng, C. H. Yang, D. Y. Wei, T. R. N. Rao, "Pseudorandom Bit Generators In Stream-Cipher Cryptography", pp. 8-17, IEEE Computers, 1991.*

{2} *H. Beker, F. Piper, "Cipher Systems, John Wiley and Sons, 1982.*

{3} *R. Rueppel, "Stream Ciphers", The Science of Information, pp. 65-134, IEEE Press, 1992.*

{4} *Rabin, M.O., "Fingerprinting by Random Polynomials", Tech. Rep. TR-15-81, Center for Research in Computing Technology, Harvard University, Cambridge, Massachessets, 1981.*

{5} *Tanenbaum, A., "Computer Networks", Prentice Hall Publishing, 1988.*

Capítulo 6. Compressão de Dados

Neste capítulo será abordado o tópico Compressão de Dados com a descrição de alguns algoritmos disponíveis. O objetivo não é ser exaustivo mas apenas prover um embasamento para a descrição de uma possível solução para redes sem fio.

6.1 Introdução

Compressão de dados é o processo pelo qual uma informação contida em N bytes é codificada em M bytes onde M < N. Existem duas classes de codificação para compressão de dados: uma **com perdas** *e outra* **sem perdas**, *onde perda diz respeito à informação. No caso de redes sem fio estamos interessados em compressão de dados* **sem perdas**, *o que pode não ser uma exigência de outras aplicações, como FAX, onde se admite alguma perda sem que a inteligibilidade global seja comprometida. Compressão* **sem perdas** *é aplicada sempre que a informação transmitida não pode ser alterada, como é o caso das redes locais.*

Utilizando-se da compressão, maiores taxas de transmissão são atingidas, o que significa, na maioria das vezes, aumento de velocidade de operação do sistema. Como exemplo de aplicações, podemos citar os Modems, enlaces de micro-ondas e as redes locais.

Outra aplicação para a compressão de dados surge da necessidade de se aumentar a capacidade de armazenamento de informação de um sistema. Para tal, os dados correspondentes são comprimidos antes de serem armazenados em memória (discos, fitas magnéticas, etc.).

6.2 Técnicas de Compressão Disponíveis

A compressão de dados **sem perdas** *é baseada em dicionários, enquanto que a* **com perdas**, *uma técnica estatística. Dentre as técnicas baseadas em dicionários as mais utilizadas são as de Lempel-Ziv, originalmente introduzidas por J. Ziv e A. Lempel em 1977 e 1978 {1} {2}. Os autores propõem duas famílias de algoritmos para compressão* **sem perdas**, *denominados LZ1 e LZ2. De uma maneira geral, os algoritmos baseiam-se em substituir seqüências repetidas por uma referência à primeira ocorrência da mesma. A seguir descreveremos ambos os algoritmos.*

6.2.1 Algoritmo LZ1

O algoritmo LZ1 armazena a seqüência de bytes de entrada em uma memória. A seguir, cada byte entrante é comparado com o conteúdo desta memória. O objetivo é descobrir alguma seqüência que se iguale à nova seqüência que está sendo recebida. Assim, caso seja identificada uma seqüência de entrada de comprimento N com alguma seqüência previamente armazenada na memória, esta é substituída por campos de informações, que correspondem ao código de compressão. Estes campos devem conter o comprimento da seqüência identificada e também a posição de início desta na memória. Assim, a compressão de dados é atingida ao se substituir determinada seqüência pelos campos citados acima. Para o caso em que não haja identificação de um byte de entrada com o conteúdo da memória, o

mesmo byte é externado ao invés do código de compressão. Dizemos que o algoritmo LZ1 constrói seu dicionário à medida que a memória é preenchida.

6.2.2 Algoritmo LZ2

Diferentemente do algoritmo LZ1, o LZ2 não armazena os bytes de entrada, mas ao invés disso, constrói um dicionário das seqüências até então encontradas. Inicialmente seu dicionário é preenchido com todos os caracteres possíveis de serem transmitidos como se fossem seqüências de apenas um byte. A seguir, para cada byte entrante, o dicionário é varrido na procura de alguma identificação. Ao se encontrar esta identificação, o processo de comparação é repetido para o próximo byte de entrada e o byte seguinte no dicionário. Esta repetição é feita até que não haja mais a identificação entre a seqüência do dicionário e a seqüência de entrada incorporada de mais um byte de entrada. Quando isto ocorre, o código da seqüência identificada é externado em substituição à seqüência de entrada. Para finalizar o processo, a seqüência identificada justaposta ao último byte de entrada, para o qual não se atingiu a identificação, é incorporada ao dicionário. Este processo repete-se a cada byte não identificado ou a cada seqüência repetida.

Nota-se que umas das diferenças entre os algoritmos LZ1 e LZ2 é que o segundo possui a necessidade de iniciar o processo de compressão com o seu dicionário contendo valores iniciais, enquanto que o primeiro é capaz de construir seu dicionário a partir de elementos nulos.

6.3 Algoritmo LZ1 Proposto

O algoritmo LZ1 proposto é uma variante do algoritmo LZ1 e é uma possível implementação de compressão de dados para redes sem fio. Descreveremos, a seguir, o processo de compressão e descompressão para este algoritmo.

6.3.1 Algoritmo LZ1 Proposto: Compressão

Como foi descrito acima, no algoritmo LZ1 cada byte de entrada é armazenado em um dicionário que pode ser visto como uma memória FIFO[12]*. Portanto, para cada novo byte, o byte mais antigo do dicionário é excluído deslocando-se todas as demais posições. O dicionário age como uma janela que desliza sobre a "história" dos dados e se estende de algum byte passado até o byte presente.*

A compressão é atingida quando alguma seqüência é identificada com outra que tenha ocorrido antes e que ainda esteja no dicionário. Estas seqüências são então codificadas por dois campos que descrevem a sua localização no dicionário e o seu comprimento. Quanto menos bits forem necessários para esta codificação, melhor será a compressão da seqüência repetida.

Tipicamente os dicionários têm tamanhos de 512 a 4098 bytes. Dicionários maiores, apesar de fornecerem maior probabilidade de identificação de seqüências, também necessitarão de códigos mais extensos para descrever sua localização e, portanto, a razão de compressão N :

[12] do inglês "First In First Out".

M nem sempre é melhorada. Como a codificação tem grande influência na performance da compressão, ela deve ser feita de tal forma que seqüências de comprimento menores sejam codificadas por menor quantidade de bits que outras. Pelo mesmo motivo, o deslocamento pode ser representado por menor quantidade de bits quanto mais próximo se encontrar do início do dicionário. Seguindo este procedimento, implementações do LZ1, como o LZ1 proposto, podem codificar até mesmo 2 bytes e conseguir comprimi-los.

Finalmente, é também necessário estabelecer-se uma codificação para os bytes avulsos que não formem uma seqüência de pelo menos 2 bytes contida no dicionário. Comumente estes bytes são codificados com seu próprio valor e a eles é acrescentado um bit para diferenciá-los das seqüências comprimidas.

A saída do compressor consistirá de bytes, que podemos chamar, de 1) ***naturais*** *e de 2) campos identificadores de deslocamento e de comprimento das seqüências comprimidas. Cada um destes códigos deve possuir um prefixo que os diferencie. Note que o dicionário no início da compressão não deverá conter valor algum e que, portanto, o primeiro dado será sempre um* ***natural****. Após alguns dados de entrada o dicionário começa a preencher suas posições de memória e a comprimir as seqüências que se repetirem.*

6.3.2 Algoritmo LZ1 Proposto: Descompressão

A composição correta do dicionário na descompressão é responsável pela integridade dos dados comprimidos a serem descomprimidos. Isto quer dizer que, mantendo-se os dicionários de compressão e descompressão idênticos, nunca será necessário transmitir este dicionário na intenção de se assegurar a correta descompressão.

Para que esta identidade seja atingida, é necessário, primeiramente, garantir o estado inicial dos dicionários, o que é alcançado, por exemplo, deixando seus elementos todos nulos. A partir de então, os dicionários de compressão e descompressão são preenchidos com bytes que compõem as mensagens e não com os códigos de compressão. Desta forma, na compressão, o dicionário é preenchido com o dados originais e, na descompressão, o dicionário recebe os dados já descomprimidos.

A descompressão consiste inicialmente em identificar se o dado recebido é um ***natural*** *(não comprimido) ou um código de compressão de seqüência. No caso de um dado* ***natural****, o descompressor apenas retira o prefixo identificador de dado não comprimido, insere-o no dicionário e o externa como dado descomprimido. Caso se trate de uma seqüência comprimida, o descompressor obtém, através do código de compressão, sua posição no dicionário e passa a copiar esta seqüência nas posições iniciais do próprio dicionário externado-a simultaneamente.*

Como ilustração descreveremos dois exemplos. Inicialmente, analisemos o caso do recebimento de um dado ***natural****. Notamos que, no exato momento do recebimento deste dado pelo descompressor, o compressor de dados já o contém em seu dicionário e, por assim dizer, o dicionário do descompressor está a um passo atrás. Após processado este dado, ou seja, o byte descomprimido é colocado no dicionário do descompressor, este estará idêntico ao dicionário do compressor. Vejamos, agora, um outro exemplo no qual uma seqüência de dez bytes, a ser transmitida, esteja contida em alguma parte do dicionário do compressor. Desta forma, o compressor irá receber a seqüência mas só irá produzir o dado comprimido quando identificar o fim desta. Neste momento, o dicionário do compressor estará dez passos à frente do dicionário do descompressor.*

Observando o algoritmo para compressão de um byte que não corresponde a seqüência alguma do dicionário, notamos que, na realidade, ocorre uma expansão. Ou seja, aos 8 bits é anexado um bit de prefixo a fim de identificar o byte como ***natural****. Este byte é codificado, portanto, em 9 bits. Como conclusão, podemos dizer que, se os dados originais se constituírem de bytes não encontrados no dicionário de compressão, haverá uma expansão de 12,5%. Na maioria dos casos em que ocorre expansões, os dados de entrada são aqueles que já haviam sido comprimidos e que, ao final, não irão ultrapassar uma expansão de 12,5%.*

Para concluir o processo de compressão e descompressão, é necessário um código que identifique o término dos dados comprimidos. Neste caso, utiliza-se um código especial de dado comprimido que não seja usado normalmente.

6.3.3 Exemplo de Compressão Utilizando-se o Algoritmo LZ1 Proposto

Para melhor entendimento do algoritmo de compressão de LZ1 proposto, vamos apresentar um exemplo no qual realizaremos a compressão dos códigos ASCII da palavra RINTINTIN.

Inicialmente um byte (caracter ASCII) é armazenado num determinado endereço do dicionário. A cada novo byte o compressor procura por uma seqüência já contida em seu dicionário. Se for encontrado um segundo byte no dicionário, na mesma ordem em que aparecem na entrada, o compressor passa a tratá-los com uma seqüência repetida. Novos bytes continuarão a ser analisados até que a seqüência repetida termine. Neste momento, o compressor substitui a seqüência repetida por um dado contendo a posição e o comprimento da seqüência no dicionário.

Ao mesmo tempo, todos os bytes recebidos são inseridos em posições sucessivas do dicionário, independentemente de fazerem parte de seqüências repetidas. Os bytes para os quais não são encontradas repetições no dicionário, são apenas externados com o prefixo de ***natural****.*

Para o nosso exemplo a Figura 16 mostra o conteúdo do dicionário.

Como já vimos, mesmo os bytes literais devem ser acrescidos de um prefixo, assim como os dados que representam seqüência comprimida devem ter especial codificação para posterior descompressão. A Tabela 1 mostra a codificação utilizada pelo algoritmo LZ1 proposto quando se trata de um dicionário de 1 kbytes.

Tabela 1. Códigos utilizados pelo LZ1 proposto.

Tipo de dado	Codificação	Comprimento
byte natural	0nnnnnnnn	9 bits
Seq. comprimida de 2 a 3 bytes	10cppppppppppp	13 bits
Seq. comprimida de 4 a 7 bytes	110ccpppppppppppp	15 bits
Seq. comprimida de 8 a 15 bytes	1110cccpppppppppppp	17 bits
Seq. comprimida de 16 a 31 bytes	11110ccccpppppppppppp	19 bits
Seq. comprimida de 32 a 271 bytes	11111ccccccccpppppppppppp	23 bits

onde:

n: bits do código em ASCII do byte natural
c: bits do comprimento da seqüência comprimida
p: bits da posição no dicionário de 1kbyte

Para um byte ***natural****, apenas o prefixo "0" é colocado antes do caracter ASCII. Como visto anteriormente, há uma expansão de 8 para 9 bits neste caso. Observa-se que qualquer código iniciando-se com "1" representa uma seqüência comprimida.*

Os códigos de seqüências comprimidas se fazem valer do comprimento variável destas. Assim, os códigos de compressão apresentados levam em consideração que, para menores seqüências comprimidas, é interessante se utilizar códigos com menor quantidade de bits para se otimizar a compressão.

Os primeiros 5 bits do código identificam qual dos possíveis códigos de compressão está sendo utilizado e, portanto, determina o comprimento do código. Esta informação é necessária para se descomprimirem os dados a fim de possibilitar a separação dos códigos. Há ainda um código não definido acima que identifica o término da compressão. Este código tem por finalidade indicar ao descompressor o fim dos dados comprimidos.

Vejamos como codificaríamos a palavra RINTINTIN. Para simples clareza de notação, não utilizaremos os códigos ASCII mas indicaremos seus bits com a letra "a". A Tabela 2 mostra a codificação.

Tabela 2. Compressão da palavra RINTINTIN.

Codificação	Descrição
0aaaaaaaa	"0" seguido do código ASCII para "R"
0aaaaaaaa	"0" seguido do código ASCII para "I"
0aaaaaaaa	"0" seguido do código ASCII para "N"
0aaaaaaaa	"0" seguido do código ASCII para "T"
10 0 0000000001	código de compressão para 2 bytes (c=0, p=1)
10 1 0000000011	código de compressão para 3 bytes (c=1, p=3)

O número total de bits da mensagem comprimida é 62 que podem ser representados em 8 bytes. Se comprado com o comprimento original da mensagem, 9 bytes, notamos o efeito da compressão, 9 : 8.

Procederemos agora de modo a exemplificar a descompressão da palavra RINTINTIN. Com a utilização do código de compressão proposto acima, a descompressão é de relativa facilidade. O descompressor deve observar o bit inicial do dado recebido, se este for "0" trata-se de um ***natural*** *e os 8 bits restantes são anexados ao dicionário e externados como dado descomprimido. Caso o bit inicial seja "1" o descompressor irá observar os próximos 5 bits do código a fim de saber seu comprimento, o comprimento da seqüência em questão e a posição dela no dicionário. A seguir, o descompressor passa a copiar esta seqüência do dicionário para as suas posições iniciais e, simultaneamente, externar os dados descomprimidos. Note que no dicionário será armazenada toda a mensagem descomprimida a fim de permitir a*

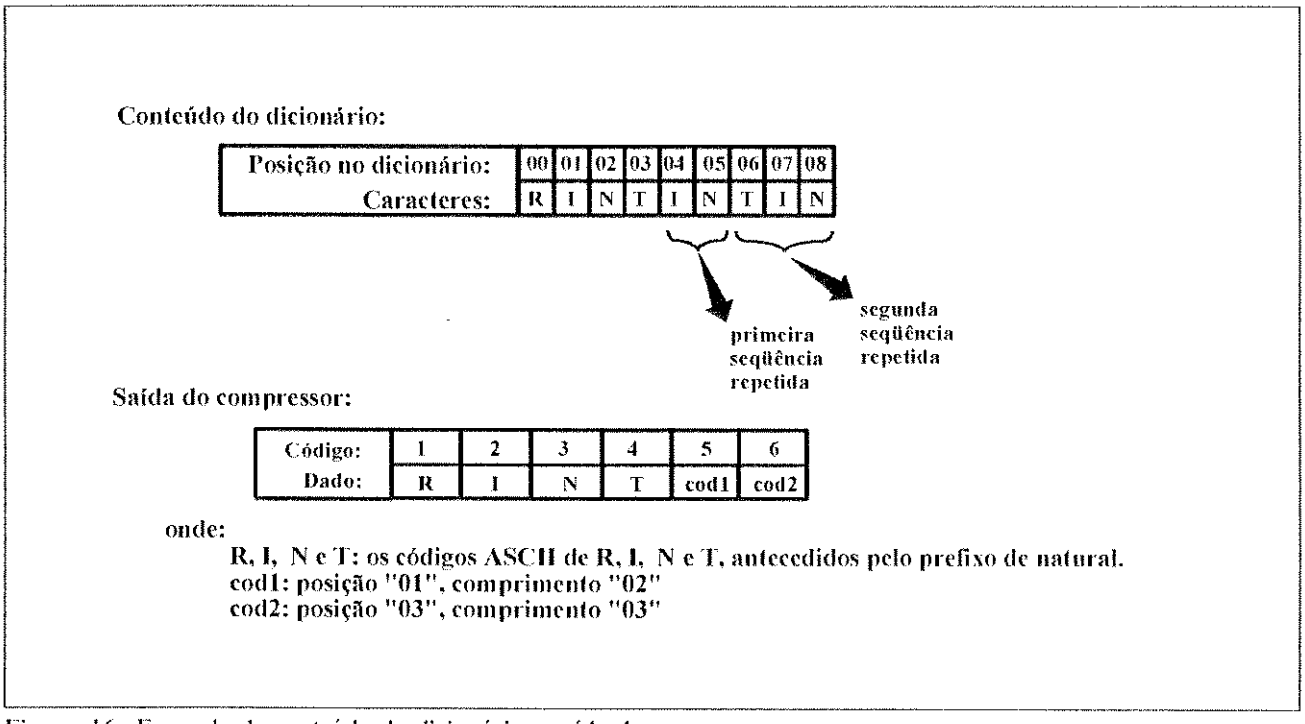


Figura 16. Exemplo de conteúdo do dicionário e saída do compressor.

recursividade na compressão. No exemplo em questão, o segundo TIN é comprimido com referência ao primeiro que já possui a seqüência IN comprimida. É para esse tipo de recursividade que é necessário que o dicionário contenha apenas dados descomprimidos.

Uma dificuldade adicional do descompressor é saber onde inicia cada código da mensagem recebida. A Tabela 3 apresenta a mensagem recebida na entrada do descompressor.

Tabela 3. Mensagem recebida pelo descompressor.

Bits:	**0 1 2 3 4 5 6 7 8**
byte 01	0 a a a a a a a
byte 02	a 0 a a a a a a
byte 03	a a 0 a a a a a
byte 04	a a a 0 a a a a
byte 05	a a a a 1 0 c p
byte 06	p p p p p p p p
byte 07	p 1 0 c p p p p
byte 08	p p p p p p - -

Ao receber o primeiro byte o descompressor percebe tratar-se de um ***natural*** *e que o próximo código inicia-se na posição corrente adicionada de 9 bits (o que significa dizer que o próximo código inicia-se no segundo byte, bit 1). Da mesma forma, para um código de compressão, o descompressor deve conhecer seu comprimento para poder alcançar o*

próximo código. Isto obviamente imprime um caráter serial na operação de descompressão, constituindo-se um gargalo de tempo de processamento.

6.4 Implementação do Compressor-Descompressor de Dados

A seguir é descrita a implementação em hardware para o algoritmo LZ1 aqui proposto. Esta descrição se fará em termos funcionais não entrando em detalhes quanto à sua implementação.

6.4.1 Compressor

Para facilidade de entendimento, dividiremos esta entidade em três blocos funcionais como indicado na Figura 17. O primeiro bloco funcional, "Comparação com Dicionário", é o mais complexo. Ele se constitui de certa lógica e principalmente de uma memória CAM[13]*. O funcionamento de uma CAM é tal que, ao receber um determinado dado em sua entrada, ela sinaliza quais de seus endereços contém este mesmo dado. Esta CAM assume o papel, descrito anteriormente, do dicionário.*

A Figura 18 mostra esquematicamente como se constitui o primeiro bloco funcional do compressor de dados. Acompanharemos agora, uma a descrição da compressão de uma seqüência a fim de estudarmos o seu funcionamento. A CAM, ao receber um dado em seu barramento de entrada, sinalizará quais de suas posições possuem este mesmo dado. A escolha de um dicionário de 1024 posições implica em uma CAM com 1024 posições e um bit sinalizador para cada posição. Este bit é ligado caso o dado de entrada se encontre na respectiva posição da CAM.

Se, para um determinado dado de entrada na CAM, houver a identificação com uma ou mais de suas posições, os bits sinalizadores são armazenados em um registro de deslocamento que, a cada dado de entrada é deslocado conforme o sentido indicado na Figura 18. Desta forma, a cada dado, a posição que havia sido identificada com o dado anterior avança no registro e passa a ocupar uma posição seguinte da CAM. Se para este novo dado houver identificação com esta posição da saida da CAM, este bit do registro de deslocamento permanecerá ativo, caso contrário ele será desligado. Um contador será incrementado a cada dado identificado e fornecerá a informação do comprimento da seqüência identificada. Ao não se identificar um novo dado na CAM, o valor do registro de deslocamento é passado para o registro acumulador. As posições de bits ligados no acumulador, em conjunto com o valor do contador, serão utilizados pelo bloco funcional seguinte enquanto este bloco reinicia o processo de comparação para o próximo dado.

Caso não haja identificação na CAM, o dado de entrada deve ser considerado como ***natural*** *e enviado ao estágio seguinte após ser acrescentado à CAM. O ponteiro citado na Figura 18 é simplesmente o seletor de endereços para a CAM, com a finalidade de preenchê-la com os dados recebidos.*

O segundo bloco funcional, "Seleção das Possíveis Seqüências", necessita das seguintes informações do primeiro: comprimento de seqüência e possíveis seqüências a serem

[13] do inglês "Content Access Memory".

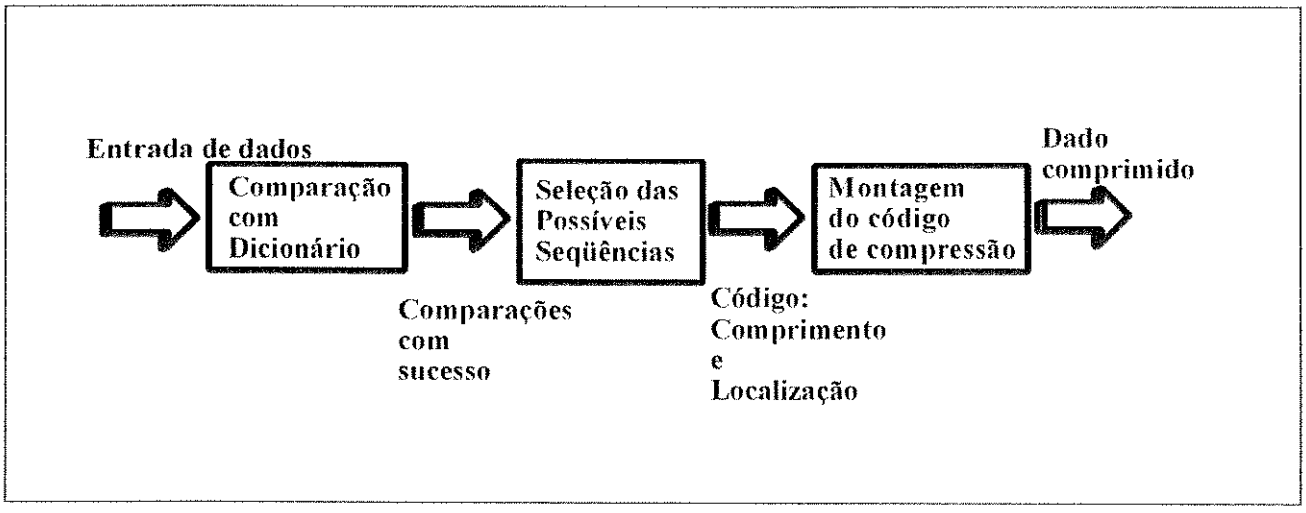


Figura 17. Diagrama de blocos da entidade de compressão.

escolhidas, uma vez que mais de um bit do acumulador pode estar afirmado no fim de cada ciclo do primeiro bloco funcional. Com isso ele é capaz de determinar uma seqüência dentre várias, a fim de ser referenciada pelo código de compressão.

O bloco seguinte, "Montagem do Código de Compressão", compõe o código de compressão e externa o que chamamos de dado comprimido. Note que este bloco, além da montagem do código, é responsável por justapô-los uma vez que os mesmos não se compõem de octetos.

6.4.2 Descompressor

A fim de compreendermos o funcionamento do descompressor, vamos dividi-lo em três blocos funcionais: "Deslocador à Esquerda", "Decodificador" e "RAM de 2 portas" (vide Figura 19). O primeiro bloco chamado de Deslocador a Esquerda é responsável por separar um a um os códigos de compressão. No caso de utilizarmos um dicionário de 1024 posições, estes códigos podem ter comprimentos que variam de 9 a 23 bits, como foi visto no item 6.3.3.

O decodificador separa a porção que descreve o comprimento da seqüência comprimida, armazenando-a em um contador. Também obtém do código de compressão a posição deste no dicionário. A partir de então, passa a fornecer ao terceiro bloco, que é na realidade uma RAM[14] *de duas portas o endereço de acesso ao mesmo tempo em que a RAM externa o dado descomprimido. O decodificador age de forma a escrever o dado descomprimido na RAM, porém na posição indicada pelo ponteiro que, assim como o contador, é atualizado a cada dado processado.*

O processo para a descompressão de um ***natural*** *é simples pois o decodificador, ao identificá-lo, irá escrevê-lo na posição indicada pelo ponteiro na RAM e irá externá-lo como dado descomprimido, excluindo-lhe o prefixo.*

[14] do inglês "Random Access Memory".

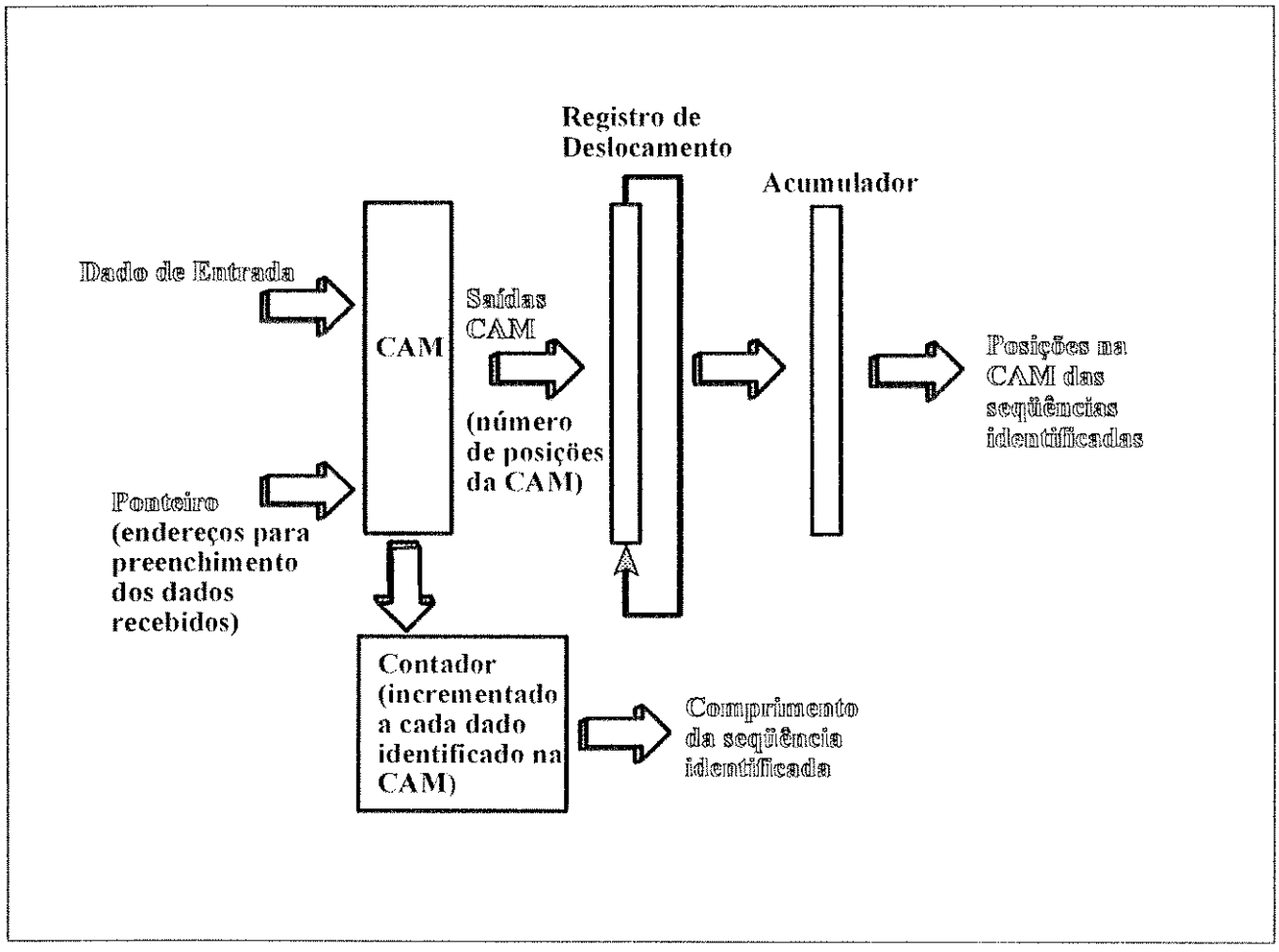


Figura 18. Bloco funcional do compressor para comparação de dados.

6.5 Resultados Atingidos pela Compressão

Os resultados mostrados a seguir são apresentados com o propósito de comparação entre os métodos LZ1 e LZ2. Quatro algoritmos implementados em hardware atualmente comercializados em produtos são analisados. A característica comum destes algoritmos é a menor complexidade se comparados a outros disponiveis.

Os algoritmos que implementam o LZ2 são licenciados pela Advanced Hardware Architectures Inc. (Moscou), que se denomina DCLZ, e pela Infochip System Inc. (California). Os algoritmos que implementam o LZ1 são licenceados pela Stac Electronics (California) e IBM, que é aqui proposto. A Tabela 4 apresenta um resumo dos resultados obtidos por diversos tipos de arquivos comprimidos versus algoritmo utilizado.

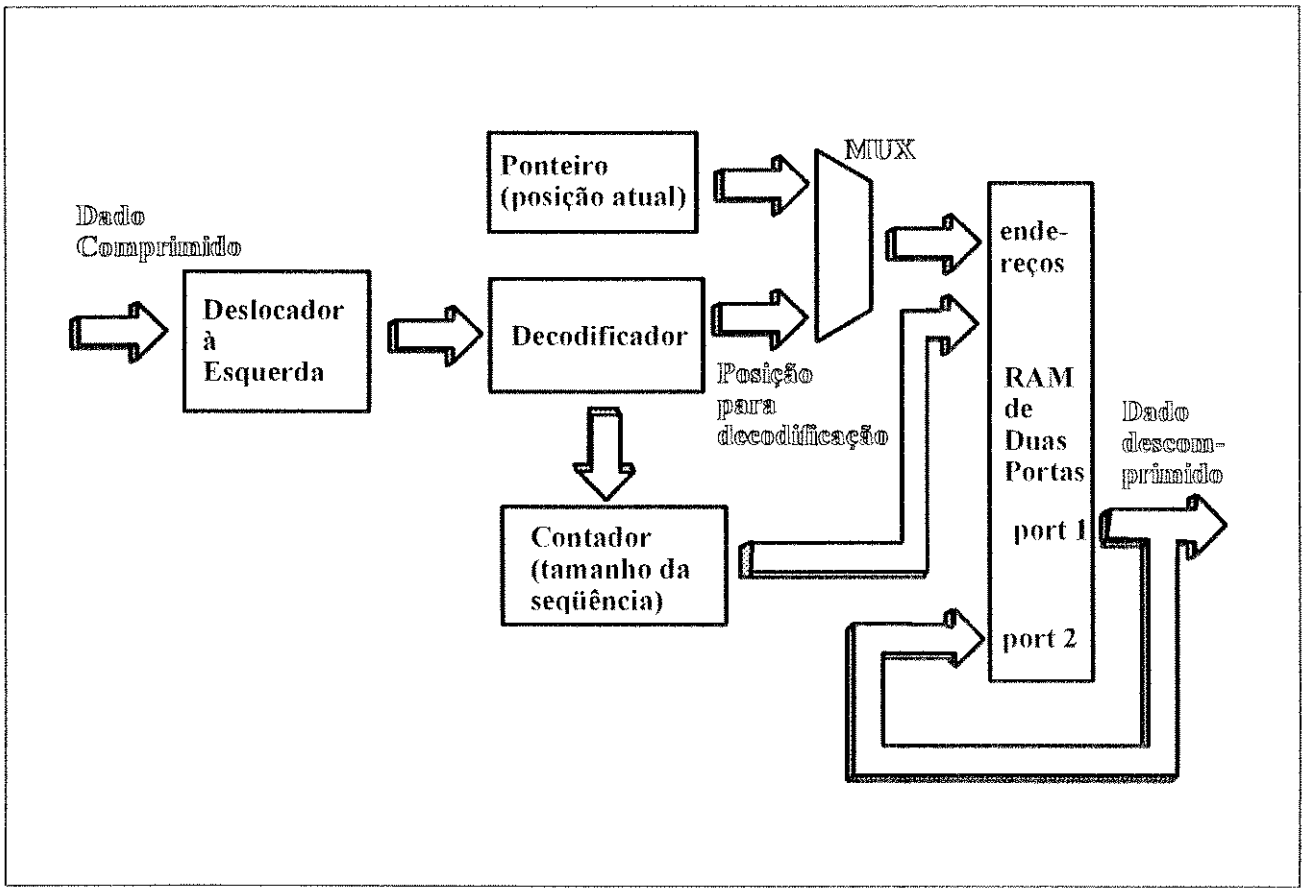


Figura 19. Diagrama funcional da entidade de descompressão.

Tabela 4. Comparação entre implementações dos algoritmos LZ1 e LZ2 {6}.

Tipo de arquivo	LZ2		LZ1
	DCSZ	ISI	STAC
Dados de teste ISI	13,7%	13,6%	19,3%
Texto em Inglês (leis)	39,8%	39,8%	33,4%
Lotus 123	45,3%	45,0%	35,8%
S/370 código objeto	39,6%	41,7%	36,7%
Inglês técnico	41,4%	42,5%	40,4%
RS/6000 código objeto	65,9%	65,5%	55,2%
80x86 código objeto	71,0%	71,9%	58,1%

Notas:

ISI: Infochip System Inc.
S/370: Sistema IBM 370
RS/6000: Estação de trabalho IBM Risc 6000
80x86: Microprocessadores Intel da família 8086

Notamos que o algoritmo LZ1 desempenha consideravelmente melhor em relação às implementações do LZ2. Os números mostram, por exemplo, que 100 Mbytes de inglês técnico é compactado em 42.5 Mbytes pela implementação do algoritmo LZ2 e em 40.4 Mbytes pelo algoritmo LZ1. É interessante observar que, para os arquivos de teste fornecido pela ISI, o algoritmo LZ2 tem sua performance acima do LZ1. Isto demonstra como se pode construir arquivos que influenciam nos resultados da compressão. Numa visão global, no entanto, o algoritmo LZ1 tem uma performance melhor.

Já na Tabela 5 comparamos duas implementações do algoritmo LZ1. Uma feita pela Stac Electronics e a outra é a LZ1 proposto. A fim de buscar parâmetros de comparação utilizamos o algoritmo Stac com 2 kbytes de dicionário, enquanto o confrontamos com o LZ1 proposto utilizando 2 kbytes, 1 kbytes e 512 bytes de dicionário.

Tabela 5. Comparação entre implementações de LZ1 {6}.

Algoritmo e tamanho de dicionário	**Taxa de compressão**	
STAC (2k)	2,862:1	34,9%
LZ1 Proposto (2k)	3,056:1	32,7%
LZ1 Proposto (1k)	2,926:1	34,2%
LZ1 Proposto (512b)	2,810:1	35,5%

Os arquivos aqui compactados são parte do conjunto representado na tabela anterior por "S/370 código objeto". Notamos que a implementação proposta pode atingir a mesma performance que a da Stac Electronics, porém utilizando-se de dicionários de 1 kbytes e 512 bytes ao invés de 2 kbytes. Isto significa simplicidade, o que implica em redução de custo de implementação.

Para completar a análise dos resultados atingidos pelos compressores de dados, falta apresentar a vazão máxima permitida para cada algoritmo implementado. O dados dos manuais fornecidos pelos fabricantes de compressores de dados estão na Tabela 6. Note que são mostradas as máximas taxas de compressão e descompressão, indistintamente do tipo de arquivo utilizado. Além disso, convém observar que tais resultados são obtidos das implementações hardware dos algoritmos.

Tabela 6. Comparação da máxima vazão das implementações de LZ1 e LZ2.

Implementação	**Vazão compressão**	**Vazão descompressão**
STAC (chip 9703)	1 Mbyte/seg.	5 Mbyte/seg.
ISI (chip IC-105)	2 Mbyte/seg.	4 Mbyte/seg.
DCLZ (chip AHA3110)	2,5 Mbyte/seg.	2,5 Mbyte/seg.
LZ1 Proposto (chip ALDC1-5S)	5 Mbyte/seg.	5 Mbyte/seg.

6.6 Conclusões

Vimos neste capítulo uma proposta de algoritmo de compressão de dados para redes sem fio implementada em hardware. Este algoritmo proposto foi inspirado nos algoritmos propostos por Lempel e Ziv {1}. O algoritmo LZ1 proposto caracteriza-se pela utilização de códigos de compressão particulares o que aumenta sua eficiência de compressão se comparado com outras implementações do LZ1.

A comparação dos resultados de compressão, entre implementações de algoritmos LZ1 e LZ2, mostra o melhor desempenho do primeiro. Outra comparação feita foi entre implementações do algoritmo LZ1 aqui proposto e da Stac Electronics. Com observância da mesma taxa de compressão para ambas as implementações, a solução aqui proposta mostra-se mais simples e econômica de ser realizada.

Com relação à arquitetura da implementação proposta, nota-se a complexidade do compressor e descompressor, por fazer uso de máquinas de estados elaboradas e entidades hardware que necessitam de grande número de registros (CAM, RAM). O aumento de complexidade é compensado pelo ganho na taxa de transmissão tornando a velocidade de operação da rede compatível com as das redes convencionais. A compressão-descompressão de dados poderia também ser realizada em software, poupando hardware de implementação. No entanto, estas tarefas requerem um processamento grande, nem sempre compatível com a capacidade das CPU's em uso.

Por fim, um parâmetro importante no caso dos compressores de dados é a vazão de compressão e descompressão. O compressor de dados deve suportar as taxas de transmissão requeridas pelo sistema como um todo, o que engloba o caminho de dados a partir do Sistema de Processamento de Dados até o Rádio (vide Capítulo 4). Este requisito faz com que o compressor de dados seja um possível gargalo no caminho da transmissão e, portanto, sua taxa de compressão e descompressão pode limitar a vazão da estação na rede sem fio. No Capítulo 2 (seção 2.8) vimos que a taxa de transmissão do Rádio proposto é de 1 Mbit/s, o que é muito inferior às taxas dos compressores de dados citados na Tabela 6.

6.7 Referências

{1} *Ziv J., Lempel A., "A Universal Algorithm for Sequential Data Compression", IEEE Transaction on Information Theory, IT-23(3), pp 337-343, 1977.*

{2} *Ziv J., Lempel A., "Compression of Individual Sequences Via Variable Rate Coding", IEEE Transaction on Information Theory, IT- 24(5), pp 530-536, 1978.*

{3} *Stac Electronics, "9703/9704 Product Family", 126 W. Del Mar Vldv., Pasadena, CA 91105, 1990.*

{4} *Karnin E. D., "Evaluation and Enhancement of LZ1 Based Data Compression Systems", IBM Science and Technology Haifa Research Group, Israel, Abr. 1991.*

{5} *Compression Subsystem Development - IBM Coorporation, Technology Products Division, "ALDC1-5S Product Specification", 1000 River Road, Essex Junction, Vermont 05452, Ago. 1993.*

{6} *D.J.Craft, "Inexpensive Vendor Hardware Data Compression Algorithms - Comparison", Publicação Interna IBM, IBM Austin Development Laboratory, Ago. 1993.*

Capítulo 7. Interface PCMCIA

Este capítulo descreve a interface PCMCIA[15], que compõe a solução proposta no que diz respeito em sua interligação com o Sistema de Processamento de Dados. Como vimos no Capítulo 4, a interface PCMCIA se encarrega de integrar o adaptador a um Sistema de Processamento de Dados que possua essa interface disponível. A opção pelo padrão PCMCIA se deve ao fato desta ser comumente encontrada nos "notebooks", "laptops" e "palmtops". Assim, os usuários destes computadores se beneficiam da mobilidade permitida pela utilização das redes sem fio. No entanto, os usuários de máquinas de maior porte, PC ou PS, poderão fazer uso dos cartões PCMCIA, uma vez que existem no mercado placas adaptadoras dos diversos barramentos comerciais, tais como ISA[16] e Microcanal[17], que tornam disponível o barramento PCMCIA. Além disso, já se observa a tendência de máquinas de maior porte trazerem de fábrica soquetes para cartões PCMCIA.

7.1 Introdução

O objetivo de se criar o padrão PCMCIA foi prover a compatibilidade entre vários microcomputadores e adaptadores eletrônicos que, por diversas razões, necessitavam de características especiais relativamente às suas dimensões físicas. Em outras palavras, a necessidade era a de se dispor adaptadores eletrônicos com formato de cartão de crédito para aplicações em "notebooks", "palmtops" e mesmo estações convencionais de trabalho. Estes cartões poderiam ter aplicações como memória ou dispositivos de expansão periférica (chamados comumente de cartões de I/O).

As bases deste padrão PCMCIA foram lançadas pela JEIDA[18] em 1985. A primeira versão do padrão PCMCIA foi liberada em setembro de 1990. Até então, projeto e comercialização de muitos adaptadores, tipo cartões de crédito, eram feitas utilizando-se interfaces proprietárias e incompatíveis umas com as outras. Assim, o objetivo inicial deste padrão foi permitir que os fabricantes de adaptadores tipo cartões de crédito e PC's pudessem anunciar produtos compatíveis. Para tal, as necessidades elétricas, mecânicas e de software foram analisadas de forma a poder contemplar numerosas tecnologias de memórias e dispositivos periféricos.

A seguir, descreveremos brevemente os principais blocos funcionais e características deste padrão, que se mostraram relevantes na implementação do adaptador em questão. Discorreremos sobre definições físicas, características elétricas e de programação que compõem o padrão PCMCIA.

---

[15] "Personal Computer Memory Card International Association".

[16] "Industry Standard Association".

[17] Barramento patenteado pela IBM.

[18] "Japan Electronics Industry Development Association".

7.2 Características Físicas

Relativamente às características físicas, o padrão PCMCIA especifica dimensões, tolerâncias mecânicas e conexões a serem aplicadas aos adaptadores. Por exemplo, um determinado tamanho de pinos do conector é definido para que a tensão no cartão seja fornecida antes da conexão dos demais pinos de sinais, quando da inserção deste. Da mesma forma, na remoção do cartão, a tensão sobre o mesmo é retirada após a desconexão dos pinos de sinais. No que se refere às características físicas, também são feitas considerações sobre confiabilidade mecânica tais como número mínimo de possíveis inserções dos cartões, condições climáticas de operação e métodos de teste de rigidez mecânica e de flexibilidade.

7.2.1 Dimensões

O padrão PCMCIA especifica três tipos de cartões conforme sua espessura. A Tabela 7 ilustra as dimensões especificadas.

Tabela 7. Dimensões dos tipos de cartões PCMCIA.

Tipo	Dimensões (mm)		
	Comprimento	Largura	Espessura
Tipo I	85,6	54,0	3,3
Tipo II	85,6	54,0	5,0
Tipo III	85,6	54,0	10,5

O tipo empregado em nossa implementação é o II (vide Figura 20). Esta escolha se deve ao fato do Rádio ser implementado externamente ao cartão PCMCIA. Assim sendo, a espessura máxima de 5 mm deve ser respeitada pelas partes que compõem o cartão, tais como, embalagem, placa de circuito impresso e componentes. Um dado adicional é que a placa de circuito impresso não estará necessariamente posicionada no centro geométrico do cartão PCMCIA, conforme mostra a Figura 21. Esta assimetria é propositada, para permitir a utilização de componentes mais espessos na face superior da placa de circuito impresso. Ou seja, utilizando-se uma placa de circuito impresso com ambos os lados para componentes eletrônicos, a altura máxima para os componentes da face superior é de 2,5 mm enquanto que na face inferior é de 1,6 mm.

Dadas as restrições de empacotamento do padrão PCMCIA, surge a dificuldade de se encontrarem componentes eletrônicos que possam ser montados em tais cartões. Hoje existem tecnologias para a confecção de tais empacotamentos para componentes eletrônicos ativos e passivos. No entanto, muitas vezes, eles só são encontrados se requisitados sob medida. Como exemplo de encapsulamento de chips podemos citar os disponíveis no mercado conforme a especificação abaixo:

***TSOP:** do inglês "Thin Small Outline Package". Componentes TTL[19] com altura máxima de 1,4 mm.*

[19] do inglês "Transistor-Transistor Logic". Circuitos digitais integrados construídos com a tecnologia bipolar.

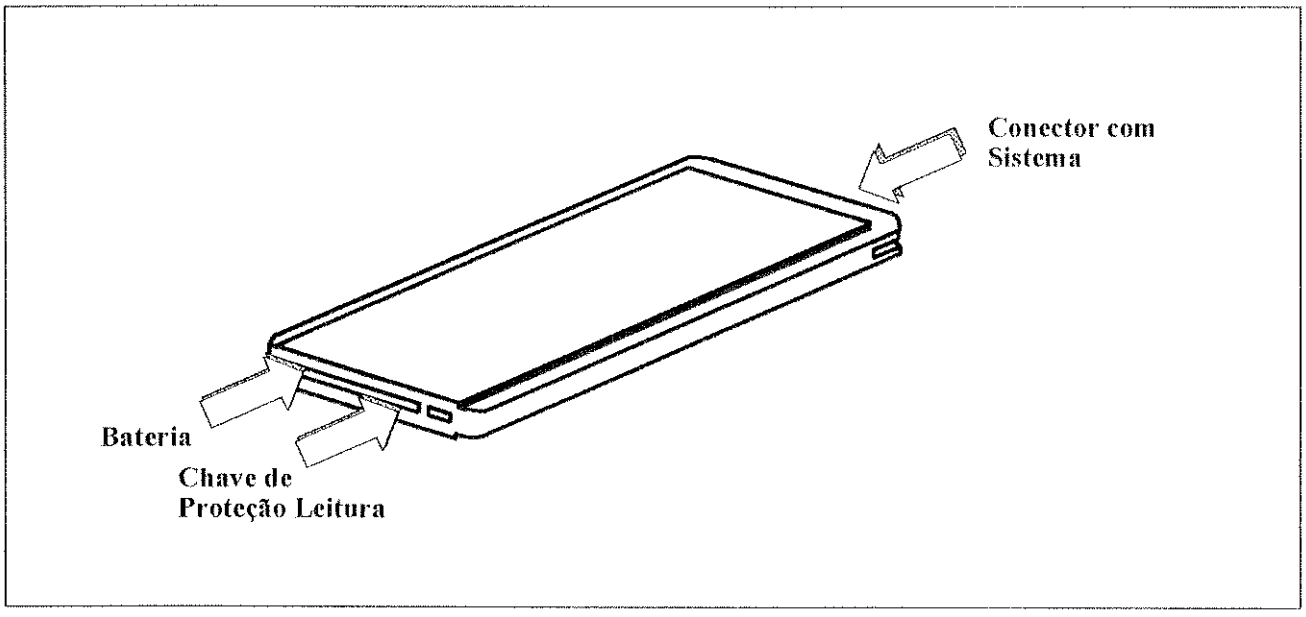


Figura 20. Cartão PCMCIA tipo II.

TQFP: *do inglês "Thin Quad Flat Package". Componentes com mais de 20 pinos em formato quadrado com altura máxima de 1,6 mm.*

Quanto aos componentes passivos, como capacitores cerâmicos e resitores de carbono, eles são comumente conhecidos no mercado pela classificação SMT0806 e são de tecnologia SMT[20] *com altura máxima de 0,55 mm.*

7.2.2 Conectores

O cartão PCMCIA liga-se ao Sistema de Processamento de Dados através de um conector de 68 pinos. Este conector deve ser do tipo fêmea, enquanto que o soquete no Sistema deve ter pinos cujo comprimento mínimo é especificado em 4,21 mm. Por razão descrita anteriormente, os pinos de alimentação do cartão devem ter comprimento de 5 mm.

Além dos desenhos apropriados para a implementação de tais conectores, o padrão PCMCIA especifica a confiabilidade necessária, força de inserção e de-inserção, limites para vibração mecânica e choque mecânico, além das caracteristicas do desempenho elétrico e ambiental.

7.2.3 Especificações Ambientais do Cartão

O padrão PCMCIA determina as caracteristicas ambientais de operação e armazenamento do cartão conforme a Tabela 8.

[20] do inglês "Surface Mount Technology".

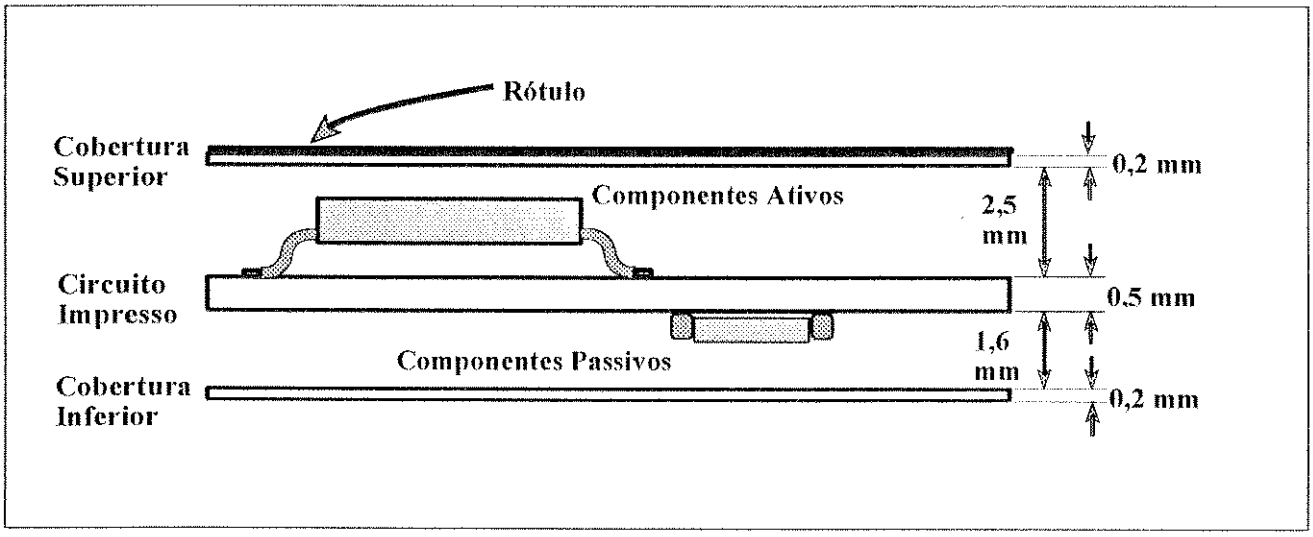


Figura 21. Perfil de uma embalagem PCMCIA típica.

Tabela 8. Características ambientais dos cartões PCMCIA.

Condição do cartão	Faixa de temperatura	Faixa de humidade
Armazenamento	-20 a 65 graus Celsius	0 a 95%
Em operação	0 a 55 graus Celsius (96 horas no mínimo)	0 a 95%

O padrão também cita limites para choques térmicos, descarga eletrostática, interferência de campos eletro-magnéticos, vibração mecânica, choque mecânico, torção e queda. É, portanto, de responsabilidade do fabricante produzir seu próprio cartão PCMCIA de acordo com as especificações do padrão.

7.3 Interface Elétrica

Nesta seção, descreveremos como o padrão PCMCIA especifica os sinais elétricos da interface. Cabe aqui lembrar o que foi dito na introdução deste capítulo sobre a existência de dois possíveis tipos de cartões PCMCIA: um referenciado como cartão de memória e outro como cartão de expansão periférica ou simplesmente cartão de I/O. Para cada tipo de cartão há uma definição de sinais na interface elétrica. Além disso, nesta seção serão descritas informações sobre funcionalidade e operação dos sinais.

A interface PCMCIA especifica um barramento de 8 ou 16 bits de dados, selecionáveis pelo próprio Sistema de Processamento de Dados. É possível, portanto, endereçar bytes (8 bits) ou words (16 bits) através da interface. A presença de registros em separado, que compõem o que se denomina de memória de atributos, permite ao Sistema obter informações altamente detalhadas sobre o cartão sem necessitar que o usuário as forneça. Utilizamos acima a expressão "registros em separado" para ressaltar que seus acessos são feitos através do mesmo barramento de dados e endereços da interface PCMCIA, porém com a ativação de um sinal especial em diferenciação aos registros de operação do cartão.

A interface PCMCIA tem capacidade de endereçar 64 Megabytes e foi concebida para suportar várias tecnologias de memórias, tais como ROM ("Read Only Memory"), OTPROM ("One Time Programmable ROM"), UV-EPROM ("Ultra-Violet Erasable Programmable ROM"), FLASH EPROM (memórias EPROM apagáveis e escritas através de algoritmos de software) e SRAM ("Static Random Access Memory"). Em particular, os cartões de I/O possuem acesso a uma interrupção do Sistema, ciclos de 8 ou 16 bits no barramento de dados, acessos assíncronos ou síncronos.

Como foi dito acima, fazem parte da interface elétrica registros em separado que constituem a memória de atributos. O padrão PCMCIA define a organização das informações nesta memória. Estas informações são referenciadas pelo padrão como CIS ("Card Information Structure") e obedece a uma organização de lista ligada. A necessidade de se ter as informações contidas na CIS disponíveis para o Sistema é, por exemplo, o fato de diferentes tecnologias de memórias terem específicos procedimentos para apagamento e escrita das mesmas (ditos algoritmos de operação).

7.3.1 Operação de um Cartão PCMCIA

Ao se energizar um cartão PCMCIA, ele deverá proceder uma autoiniciação e assim obter o valor "default" para todos os seus registros internos incluindo os que constituem a interface PCMCIA. O "default" da iniciação de qualquer cartão PCMCIA é apresentar-se como cartão de memória ao Sistema (mesmo que seja um cartão de I/O). Neste estado inicial, os sinais válidos para a interface são os descritos para os cartões de memória. Através de leituras na memória de atributos, o Sistema obtém informações sobre o cartão conectado a seu soquete PCMCIA identificando, inclusive, que se trata de um cartão de I/O ou de memória.

Um registro alocado na mesma área da memória de atributos e denominado de COR ("Configuration Option Register") deve ser acessado pelo Sistema a fim de configurar o cartão como I/O, se este for o caso. O valor a ser escrito neste registro, assim como o seu endereço, estão disponíveis nas informações contidas na memória de atributos. Uma vez acessado registro COR, um cartão de I/O passa a operar como tal, alguns de seus pinos da interface trocam de função (vide Tabela 10). Neste caso, os registros de I/O, que antes não estavam disponíveis para o Sistema, agora podem ser acessados.

7.3.2 Descrição dos Sinais da Interface PCMCIA

Através da descrição dos sinais da interface PCMCIA muito será dito da funcionalidade do cartão. Inicialmente convém enfatizar que o padrão PCMCIA prevê dois níveis de alimentação para o cartão: 5 Volts +/- 5% ou 3,3 Volts +/- 5%. A seguir, a descrição dos sinais da interface PCMCIA.

Endereços: *A0-A25. Permitem o endereçamento de até 64 Megabytes, seja para acessos de memória comum, memória de atributos ou I/O.*
Dados: *D0-D15. Barramento de dados bi-direcional que é utilizado para escrita e leitura.*
Habilitadores do Cartão: *CE1 e CE2. Quando ativados, estes sinais habilitam o cartão ao acesso de escrita ou leitura. CE1 habilita os endereços pares e CE2 os endereços ímpares.*
Habilitador de Saída: *OE. Quando ativado pelo Sistema faz o cartão externar os dados selecionados pelos endereços e habilitadores do cartão. Este sinal é utilizado nas operações de leitura.*

Habilitador de Escrita: *WE. Este sinal é ativado pelo Sistema durante acessos de escrita. Os dados presentes no barramento são armazenados no endereço convenientemente selecionado.*
Pronto/Ocupado: *RDY/BSU. O cartão ativa este sinal para indicar ao Sistema que está processando informações, ficando indisponível para outras operações. É utilizado, por exemplo, quando um cartão com memória FLASH recebe um comando de apagamento e está processando o mesmo.*
Cartão Detectado: *CD1, CD2. São sinais provenientes do cartão para o Sistema informando de sua presença.*
Proteção de Escrita: *WP. Este sinal protege o cartão PCMCIA de qualquer escrita do Sistema. O cartão pode possuir uma chave para proteção de escrita como um disquete convencional. Quando ativada esta chave, este sinal também é ativado.*
Seleção da Memória de Atributos: *REG. Este sinal permanece ativo quando ocorre um acesso à memória de atributos ou aos registros de I/O. Durante um acesso à memória comum, este sinal não é ativado.*
Detecção de Tensão de Bateria: *BVD1, BVD2. Informam ao Sistema as condições da bateria interna do cartão (se houver) conforme a Tabela 9.*

Tabela 9. Possíveis estados de tensão na bateria do cartão PCMCIA.

BVD1	BVD2	Estado da bateria
Ativo	Ativo	Bateria operacional
Ativo	Inativo	Bateria deve ser trocada. Dados íntegros.
Inativo	Ativo	Bateria deve ser trocada. Dados não estão íntegros.
Inativo	Inativo	Bateria deve ser trocada. Dados não estão íntegros.

Tensão de Programação e Periférica: *VPP1, VPP2. De acordo com as informações contidas na memória de atributo, o Sistema deve fornecer a estes pinos as tensões de 0, 5 ou 12 Volts. Estas tensões são utilizadas para a programação de memórias ou mesmo alimentação adicional dos circuitos.*
Tensão e Terra *Vcc, Gnd. Estes pinos estão distribuídos em posições simétricas (dois pinos Vcc e quatro pinos Gnd). Através das informações disponíveis na memória de atributos, o Sistema saberá se o cartão possui tensão dual, ou seja, opera também com 3,3 Volts além de 5 Volts no pino Vcc. Em caso afirmativo, o Sistema deve fornecer 3.3 Volts no pino Vcc e ajustar os "timings" convenientes para seus acessos devido a tal alteração.*
"Refresh": *RFSH. Sinal utilizado pelo cartão para renovar a carga elétrica das memórias pseudo-estáticas ou dinâmicas.*
"Reset": *RESET. Este sinal é utilizado para a iniciação do cartão. Quando ativado pelo Sistema leva o cartão para sua condição inicial de cartão de memória com todos os seus registros contendo valores iniciais.*
Ciclo de Barramento Estendido: *WAIT. Este sinal é utilizado para estender o tempo de acesso de um registro interno ao cartão ou mesmo um acesso à memória.*
Leitura de I/O: *IORD. Este sinal tem a mesma função que o sinal OE (habilitador de saída) porém aplicado aos registros de I/O.*
Escrita de I/O: *IOWR. Da mesma forma que o sinal IORD age para a leitura, este sinal age como habilitador para a escrita de registros de I/O.*
Reconhecimento de Acesso: *INPACK. O cartão ativa este sinal quando qualquer registro existente no cartão é selecionado pelo barramento de endereços com a conveniente ativação*

dos sinais de controle. Este sinal pode ser utilizado, por exemplo, para ativar as portas direcionadoras ("drivers") de dados do Sistema.

Acesso de I/O com 16 bits: *IOIS16. Quando configurado como cartão de I/O, o sinal WP (Proteção de Escrita) cede lugar ao sinal IOIS16, que tem por função identificar ao Sistema um registro que deve ser acessado através dos 16 bits do barramento de dados.*

Requisição de Interrupção: *IREQ. Este sinal substitui o sinal RDY/BSY (Pronto/Ocupado) quando o cartão está configurado como cartão de I/O. É, então, utilizado pelo cartão I/O para solicitar ao Sistema algum tipo de serviço.*

Sinal Digital de Áudio: *SPKR. Este sinal substitui o sinal de estado de bateria BVD2 quando o cartão está selecionado como I/O. Ele provê um sinal binário que deve ser direcionado opcionalmente ao auto-falante do Sistema.*

Troca de Estado: *STSCHG. O uso deste sinal é opcional e ele substitui o sinal de bateria BVD1 quando o cartão está selecionado como I/O. Se ativo, este sinal indica que um dos estados Pronto/Ocupado, Proteção de Escrita ou Tensão de Bateria teve alteração. Note que, para um cartão de memória, estes estados eram simbolizados por sinais designados aos pinos da interface e, para um cartão de I/O, estes estados estão contidos no registro de realocação de sinais. Vide seção 7.3.3, "Registros da Interface".*

A Tabela 10 apresenta os sinais da interface PCMCIA incluindo sua posição no conector (pino), tipo de sinal (entrada ou saída - "In", "Out" ou "In/Out") e função. Observa-se nesta tabela a diferença de função de determinados pinos caso o cartão esteja configurado como cartão de memória ou cartão de I/O.

Tabela 10. Sinais da interface PCMCIA.

	Cartão de Memória			Cartão de I/O		
Pino	**Sinal**	**I/O**	**Função**	**Sinal**	**I/O**	**Função**
1	GND		Terra	GND		Terra
2	D3	I/O	Bit de Dado (3)	D3	I/O	Bit de Dado (3)
3	D4	I/O	Bit de Dado (4)	D4	I/O	Bit de Dado (4)
4	D5	I/O	Bit de Dado (5)	D5	I/O	Bit de Dado (5)
5	D6	I/O	Bit de Dado (6)	D6	I/O	Bit de Dado (6)
6	D7	I/O	Bit de Dado (7)	D7	I/O	Bit de Dado (7)
7	CE1	I	Habilitador do Cartão	CE1	I	Habilitador do Cartão
8	A10	I	Bit de Endereço (10)	A10	I	Bit de Endereço (10)
9	OE	I	Habilitador de Saída	OE	I	Habilitador de Saída
10	A11	I	Bit de Endereço (11)	A11	I	Bit de Endereço (11)
11	A9	I	Bit de Endereço (9)	A9	I	Bit de Endereço (9)

	Cartão de Memória			Cartão de I/O		
Pino	**Sinal**	**I/O**	**Função**	**Sinal**	**I/O**	**Função**
12	A8	I	Bit de Endereço (8)	A8	I	Bit de Endereço (8)
13	A13	I	Bit de Endereço (13)	A13	I	Bit de Endereço (13)
14	A14	I	Bit de Endereço (14)	A14	I	Bit de Endereço (14)
15	WE/PGM	I	Habilitador de Escrita	WE/PGM	I	Habilitador de Escrita
16	RDY/BSY	O	Pronto/Ocupado	IREQ	O	Requisição de Interrupção
17	Vcc		Alimentação: +5 Volts	Vcc		Alimentação: +5 Volts
18	Vpp1		Tensão de Programação (1)	Vpp1		Tensão de Programação (1)
19	A16	I	Bit de Endereço (16)	A16	I	Bit de Endereço (16)
20	A15	I	Bit de Endereço (15)	A15	I	Bit de Endereço (15)
21	A12	I	Bit de Endereço (12)	A12	I	Bit de Endereço (12)
22	A7	I	Bit de Endereço (7)	A7	I	Bit de Endereço (7)
23	A6	I	Bit de Endereço (6)	A6	I	Bit de Endereço (6)
24	A5	I	Bit de Endereço (5)	A5	I	Bit de Endereço (5)
25	A4	I	Bit de Endereço (4)	A4	I	Bit de Endereço (4)
26	A3	I	Bit de Endereço (3)	A3	I	Bit de Endereço (3)
27	A2	I	Bit de Endereço (2)	A2	I	Bit de Endereço (2)
28	A1	I	Bit de Endereço (1)	A1	I	Bit de Endereço (1)
29	A0	I	Bit de Endereço (0)	A0	I	Bit de Endereço (0)
30	D0	I/O	Bit de Dado (0)	D0	I/O	Bit de Dado (0)
31	D1	I/O	Bit de Dado (1)	D1	I/O	Bit de Dado (1)
32	D2	I/O	Bit de Dado (2)	D2	I/O	Bit de Dado (2)
33	WP	O	Proteção de Escrita	IOS16	O	Acesso de I/O com 16 bits
34	GND		Terra	GND		Terra
35	GND		Terra	GND		Terra
36	CD1	O	Cartão Detectado (1)	CD1	O	Cartão Detectado (1)
37	D11	I/O	Bit de Dado (11)	D11	I/O	Bit de Dado (11)

	Cartão de Memória			Cartão de I/O		
Pino	**Sinal**	**I/O**	**Função**	**Sinal**	**I/O**	**Função**
38	D12	I/O	Bit de Dado (12)	D12	I/O	Bit de Dado (12)
39	D13	I/O	Bit de Dado (13)	D13	I/O	Bit de Dado (13)
40	D14	I/O	Bit de Dado (14)	D14	I/O	Bit de Dado (14)
41	D15	I/O	Bit de Dado (15)	D15	I/O	Bit de Dado (15)
42	CE2	I	Habilitador do Cartão (2)	CE2	I	Habilitador do Cartão (2)
43	RFSH	I	"Refresh"	RFSH	I	"Refresh"
44	RUF		Reservado	IORD	I	Leitura de I/O
45	RUF		Reservado	IOWR	I	Escrita de I/O
46	A17	I	Bit de Endereço (17)	A17	I	Bit de Endereço (17)
47	A18	I	Bit de Endereço (18)	A18	I	Bit de Endereço (18)
48	A19	I	Bit de Endereço (19)	A19	I	Bit de Endereço (19)
49	A20	I	Bit de Endereço (20)	A20	I	Bit de Endereço (20)
50	A21	I	Bit de Endereço (21)	A21	I	Bit de Endereço (21)
51	Vcc		Alimentação: +5 Volts	Vcc		Alimentação: +5 Volts
52	Vpp2		Tensão de Programação (2)	Vpp2		Tensão de Programação (2)
53	A22	I	Bit de Endereço (22)	A22	I	Bit de Endereço (22)
54	A23	I	Bit de Endereço (23)	A23	I	Bit de Endereço (23)
55	A24	I	Bit de Endereço (24)	A24	I	Bit de Endereço (24)
56	A25	I	Bit de Endereço (25)	A25	I	Bit de Endereço (25)
57	RUF		Reservado	RUF		Reservado
58	RESET	I	"Reset"	RESET	I	"Reset"
59	WAIT	O	Ciclo de Barramento Estendido	WAIT	O	Ciclo de Barramento Estendido
60	RUF		Reservado	INPACK	O	Reconhecimento de Acesso
61	REG	I	Seleção da Memória de Atributos	REG	I	Seleção da Memória de Atributos
62	BVD2	O	Detecção de Tensão da Bateria (2)	SPKR	O	Sinal Digital de Áudio

	Cartão de Memória			Cartão de I/O		
Pino	**Sinal**	**I/O**	**Função**	**Sinal**	**I/O**	**Função**
63	BVD1	O	Detecção de Tensão da Bateria (1)	STSCHG	O	Troca de Status
64	D8	I/O	Bit de Dado (8)	D8	I/O	Bit de Dado (8)
65	D9	I/O	Bit de Dado (9)	D9	I/O	Bit de Dado (9)
66	D10	I/O	Bit de Dado (10)	D10	I/O	Bit de Dado (10)
67	CD2	O	Cartão Detectado (2)	CD2	O	Cartão Detectado (2)
68	GND		Terra	GND		Terra

As Tabelas 11, 12, 13, 14, 15 e 16 ajudam a identificar os sinais ativos para as diversas operações de acesso à interface PCMCIA. Elas são apenas esquemáticas em relação à polaridade dos sinais durante os acessos do Sistema. A simbologia utilizada é a seguinte: 'X' significa indiferença quanto ao estado do sinal, '1' simboliza o "um" lógico e '0' simboliza o "zero" lógico. O termo "alta imped." (alta impedância) refere-se ao barramento não ser alimentado pelo cartão PCMCIA naquele acesso, "byte ímpar" refere-se ao byte acessado cujo endereço é ímpar, assim como "byte par" refere-se ao byte cujo o endereço é par. "Byte indet." (byte indeterminado) identifica a impossibilidade de se prever o conteúdo do barramento no acesso realizado.

Tabela 11. Leitura de Memória Comum.

Modo de Acesso	REG	CE2	CE1	A0	OE	WE	D15-8	D7-0
Sem acesso	X	1	1	X	X	X	Alta imped.	Alta imped.
Acesso de byte par (1)	1	1	0	0	0	1	Alta imped.	Byte par
Acesso de byte par (2)	1	1	0	1	0	1	Alta imped.	Byte ímpar
Acesso de byte ímpar	1	0	1	X	0	1	Byte ímpar	Alta imped.
Acesso de palavra	1	0	0	X	0	1	Byte ímpar	Byte par

Tabela 12. Escrita de Memória Comum.

Modo de Acesso	REG	CE2	CE1	A0	OE	WE	D15-8	D7-0
Sem acesso	X	1	1	X	X	X	Alta imped.	Alta imped.
Acesso de byte par (1)	1	1	0	0	1	0	Alta imped.	Byte par
Acesso de byte par (2)	1	1	0	1	1	0	Alta imped.	Byte ímpar
Acesso de byte ímpar	1	0	1	X	1	0	Byte ímpar	Alta imped.
Acesso de palavra	1	0	0	X	1	0	Byte ímpar	Byte par

Tabela 13. Leitura de Memória de Atributo.

Modo de Acesso	REG	CE2	CE1	A0	OE	WE	D15-8	D7-0
Sem acesso	X	1	1	X	X	X	Alta imped.	Alta imped.
Acesso de byte par (1)	0	1	0	0	0	1	Alta imped.	Byte par
Acesso de byte par (2)	0	1	0	1	0	1	Alta imped.	Byte indet.
Acesso de byte ímpar	0	0	1	X	0	1	Byte ímpar	Alta imped.
Acesso de palavra	0	0	0	X	0	1	Byte ímpar	Byte par

Tabela 14. Escrita de Memória de Atributo.

Modo de Acesso	REG	CE2	CE1	A0	OE	WE	D15-8	D7-0
Sem acesso	X	1	1	X	X	X	Alta imped.	Alta imped.
Acesso de byte par (1)	0	1	0	0	1	0	Alta imped.	Byte par
Acesso de byte par (2)	0	1	0	1	1	0	Alta imped.	Byte ímpar
Acesso de byte ímpar	0	0	1	X	1	0	Byte ímpar	Alta imped.
Acesso de palavra	0	0	0	X	1	0	Byte ímpar	Byte par

Tabela 15. Leitura de I/O.

Modo de Acesso	REG	CE2	CE1	A0	IORD	IOWR	D15-8	D7-0
Sem acesso	X	1	1	X	X	X	Alta imped.	Alta imped.
Acesso de byte par (1)	0	1	0	0	0	1	Alta imped.	Byte par
Acesso de byte par (2)	0	1	0	1	0	1	Alta imped.	Byte indet.
Acesso de byte ímpar	0	0	1	X	0	1	Byte ímpar	Alta imped.
Acesso de palavra	0	0	0	0	0	1	Byte ímpar	Byte par

Tabela 16. Escrita de I/O.

Modo de Acesso	REG	CE2	CE1	A0	IORD	IOWR	D15-8	D7-0
Sem acesso	X	1	1	X	X	X	Alta imped.	Alta imped.
Acesso de byte par (1)	0	1	0	0	1	0	Alta imped.	Byte par
Acesso de byte par (2)	0	1	0	1	1	0	Alta imped.	Byte ímpar
Acesso de byte ímpar	0	0	1	X	1	0	Byte ímpar	Alta imped.
Acesso de palavra	0	0	0	0	1	0	Byte ímpar	Byte par

7.3.3 Registros da Interface

Para que o Sistema possa acessar os registros da interface, ele deve inicialmente consultar a memória de atributos para saber seus endereços. Os cartões, por outro lado, podem ter vários conjuntos dos mesmos registros dependendo de suas possíveis configurações.

A começar pelo Registro de Configuração de Opção (COR - "Configuration Option Register"), que deve ser implementado em todos os cartões PCMCIA, o Sistema pode, através de escritas apropriadas, re-iniciar o cartão (Reset), selecionar o modo de ativação da requisição de interrupção (por nivel ou por pulso) e também selecionar uma das possíveis configurações do cartão que estejam definidas pelo conteúdo da memória de atributos.

Outro registro, denominado Configuração do Cartão e Status ("Card Configuration and Status Register"), possibilita ao Sistema identificar alguma alteração nos estados do cartão, habilitar alterações a serem registradas neste registro, garantir o acesso de 8 bits ao cartão, habilitar

informações de áudio no pino BVD2, colocar o cartão no modo de economia de energia e observar se o cartão necessita que se atenda à requisição de interrupção.

Notamos na seção anterior que vários sinais cedem seus pinos a outros quando o cartão é configurado em modo I/O. Desta forma, o Registro de Re-alocação de Pinos ("Pin Replacement Register") fornece as informações antes constantes nesses pinos ao Sistema, tais como estado da bateria, da chave de proteção de escrita e da disponibilidade do cartão para acessos (Pronto/Ocupado).

Por fim, um último registro que completa a interface é o denominado Registro de Soquete e Cópia ("Socket and Copy Register"). Este registro é utilizado para distinguir cartões similares instalados no mesmo Sistema. Ele permite ao Sistema dar um número de cópia ao cartão e ao soquete que os mantém unicamente identificados para compartilhar espaço de memória ou I/O com os demais.

7.3.4 Memória de Atributos

O conteúdo da memória de atributos deve fornecer informações suficientes para o Sistema a respeito das características do cartão. O padrão PCMCIA trata de organizar os dados contidos na memória de atributos e a denomina de Estrutura de Informação do Cartão ("Card Information Structure" - CIS). Esta estrutura é uma lista ligada de blocos de dados, denominados pelo padrão de t-uplas. Todas as t-uplas têm o mesmo formato, que compreende seu código (1 byte) e dados específicos da t-upla. Por definição, a primeira t-upla deve se localizar no endereço 0 (zero) da memória de atributos. A Tabela 17 mostra o formato padrão das t-uplas.

Tabela 17. Formato geral das t-uplas.

Byte	Conteúdo
0	Código da t-upla
1	Conexão para próxima t-upla (ou seja, comprimento da t-upla: m-1)
2...m	Bytes específicos da t-upla

Vejamos a seguir, as possíveis t-uplas definidas. Começando com as t-uplas de controle, temos seis tipos principais. "T-upla Neutra", identificada pelo código 00H[21], simplesmente preenche um byte da memória de atributos. "T-upla de Controle para Conexão Longa", identificada pelo código 11H ou 12H, indica o próximo endereço da memória de atributos (composto de 4 bytes) do qual será extraída a próxima t-upla. "T-upla Alvo para Conexão" possui código 13H e é utilizada no endereço de destino para toda conexão da lista das t-uplas. "T-upla de Controle de Não-Conexão", código 14H, é utilizada em exclusão com a T-upla de Controle para Conexão Longa (a ausência de uma é registrada pela presença da outra). "T-upla de Fim de Lista" é utilizada para identificar fim da estrutura de informação; seu código é FFH. "T-upla de Cheque de Soma", cuja função é garantir a integridade dos dados que compõem a estrutura de informação, possui o código 10H.

[21] H significa notação hexadecimal.

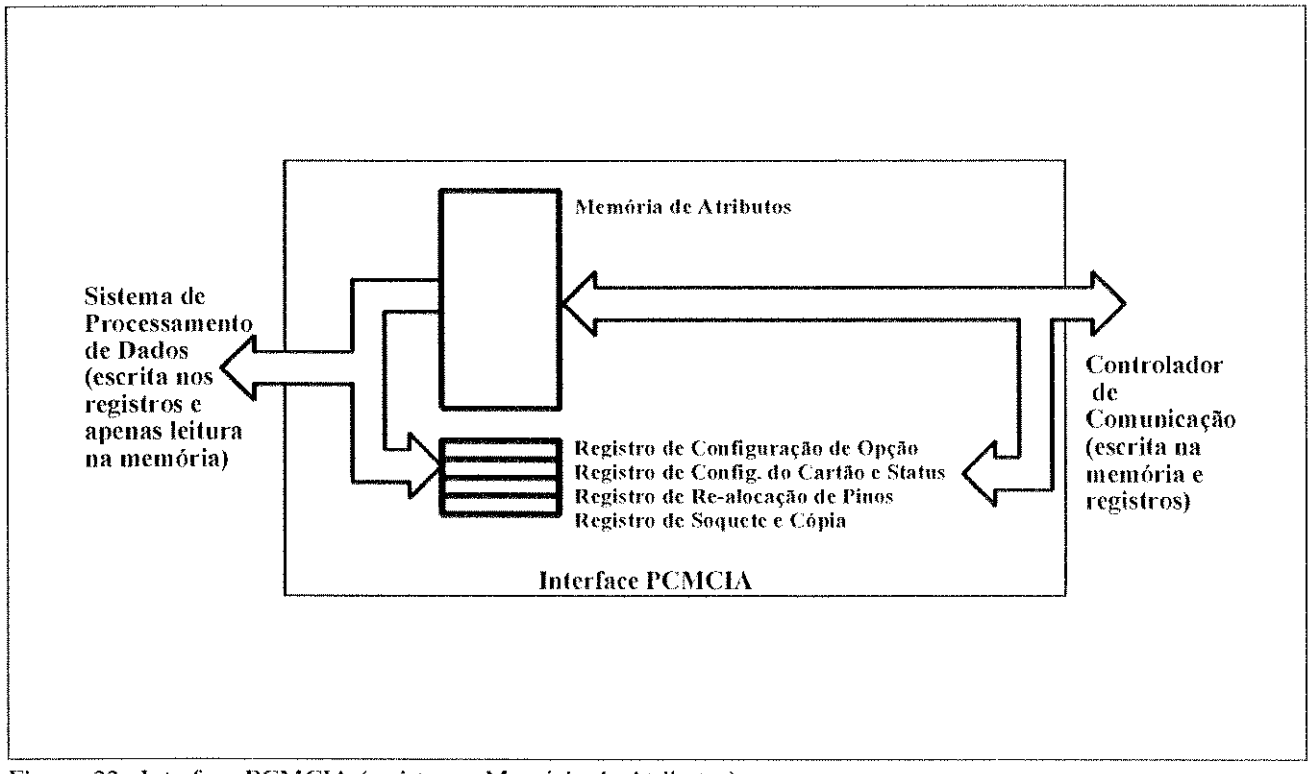


Figura 22. Interface PCMCIA (registros e Memória de Atributos).

A seguir abordaremos as t-uplas de informação sobre o cartão. Classificamos as t-uplas de informação em duas classes: T-uplas de Compatibilidade Básica e T-uplas de Configuração. A primeira classe contém informações sobre o dispositivo no que diz respeito às características da memória de atributos e da memória comum, enquanto que a segunda classe descreve as características configuráveis do cartão.

As T-uplas de Compatibilidade Básica citam, por exemplo, especificações de velocidade de acesso ao dispositivo, tamanho das memórias, tipo de dispositivo (tais como ROM, OTPROM, EPROM, Flash EPROM, SRAM, DRAM ou dispositivo de I/O). Além disso, as T-uplas de Compatibilidade Básica dão informações sobre a versão do padrão PCMCIA utilizado e informações do fabricante do cartão (como, por exemplo, nome do produto e fabricante, local de manufatura, número de série e outras informações em formato texto). Note que todas as informações das t-uplas descritas acima devem obedecer aos campos especificados pelo padrão PCMCIA para cada uma delas. As Tabelas 18, 19 e 20 mostram como são organizadas estas informações.

Tabela 18. T-uplas de Compatibilidade Básica: informações sobre o dispositivo.

Byte	Descrição
0	Código da t-upla: 01H para memória comum e 17H para memória de atributos.
1	Conexão para próxima t-upla.
2...5	Código do dispositivo: ROM (1), OTPROM (2), EPROM (3), EEPROM (4), Flash EPROM (5), SRAM (6), DRAM (7), I/O (D) ou outros (requer byte 5). Velocidade do dispositivo: 100, 150, 200, 250 ns ou outros (requer bytes 3 e 4).
6	Número de bytes do dispositivo (tamanho da memória comum ou da memória de atributos).
m	Conteúdo FFH marca o fim da lista.

Tabela 19. T-uplas de Compatibilidade Básica: informações complementares.

Byte	Descrição
0	Código da t-upla: 1CH para memória comum e 1DH para memória de atributos.
1	Conexão para próxima t-upla.
2	Suporta a operação com 3 Volts de alimentação e suporta à utlilização do sinal WAIT.
3...	Informações não "default" sobre condições de operação do dispositivo. Obedece ao mesmo formato da t-upla de informações sobre o dispositivo.
m	Conteúdo FFH marca o fim da lista.

Tabela 20. T-uplas de Compatibilidade Básica: informações sobre o produto e identificadores JEDEC.	
Byte	**Descrição**
0	Código da t-upla: 15H.
1	Conexão para próxima t-upla.
2 e 3	Número de versões do padrão que são compatíveis.
4...	Nome do fabricante, nome do produto, informações adicionais sobre o produto (em formato texto).
m	Conteúdo FFH marca o fim da lista.
	Identificadores JEDEC:
0	Código da t-upla: 18H para memória comum e 19H para memória de atributos.
1	Conexão para próxima t-upla.
2 e 3	Identificadores JEDEC.
4...	Outros identificadores JEDEC se necessário.
m	Conteúdo FFH marca o fim da lista.

Como mencionado, as T-uplas de Configuração dão informações sobre as características configuráveis do cartão. Estas características podem ser:

- *tipo de cartão (memória ou I/O);*
- *uso de acessos assíncronos (pela utilização do sinal WAIT), chave de proteção de escrita presente, bateria presente e possibilidade de atuar no sinal de interrupção;*
- *configuração de Vcc e Vpp;*
- *designação das portas de I/O e mapeamento de memória necessário;*
- *níveis de interrupção para o Sistema e*
- *identificação única dos cartões.*

Na Tabela 21 podemos ver como estas informações são inseridas na T-upla de Configuração. Assim, esta t-upla determina os endereços dos registros de configuração da interface PCMCIA e a presença ou ausência destes. Uma tabela contendo dados sobre a configuração do cartão é utilizada como T-upla de Configuração, a fim de apresentar informações compactadas em um determinado formato especificado pelo padrão PCMCIA. Esta tabela de configuração possui como primeiro byte seu código identificador (1BH). A seguir, como qualquer outra t-upla, ela indica o número de bytes que compõem a tabela e finalmente, como primeira informação, o byte que chamamos de índice de configuração. Este byte deve ser utilizado pelo Sistema para ser escrito no Registro de Configuração do Cartão (COR) a fim de fazê-lo operar conforme determinam as informações apresentadas na tabela. Estas informações são: campos opcionais presentes, tais como tipo de interface (memória ou I/O), utilização do bit de Pronto/Ocupado e suporte a acessos assíncronos utilizando-se o sinal WAIT. Além disso, a tabela apresenta informações sobre utilização de Vcc e Vpp (valores máximos, mínimos e corrente de alimentação), tempo de acessos assíncronos, número de linhas de endereço necessárias para acessos aos registros de I/O, largura do barramento (8 ou 16 bits de dados), estrutura de interrupção, espaço em memória para alocação no Sistema e informações adicionais, como possibilidade de operação com áudio e de economia de energia. A Tabela 22 lista as posições disponíveis na tabela de configuração. É possível haver

uma tabela de configuração com vários índices acompanhados pelas informações necessárias à tabela.

Tabela 21. T-upla de Configuração.

Byte	Descrição
0	Código da t-upla: 1AH.
1	Conexão para próxima t-upla.
2	Número de bytes para identificação do endereço do endereço dos registros de configuração.
3	Número de índices disponíveis na tabela de configuração.
4...7	Endereço dos registros de configuração na memória de atributos (até 4 bytes).
8 e 9	Presença dos registros de configuração (até 2 bytes).
10...	Informações adicionais sobre a interface.
m	Conteúdo FFH marca o fim da lista.

7.4 Camadas de Software para um Cartão PCMCIA

A operação de um cartão PCMCIA envolve a interação do hardware com as seguintes camadas software:

- *Serviços de soquete;*
- *Serviços de cartão;*
- *Direcionadores específicos para diferentes tecnologias de cartões e*
- *Programas de aplicações que são executados pelo Sistema de Processamento de Dados.*

A Figura 23 mostra a interligação destas camadas. A camada de hardware refere-se aos cartões de crédito, seus soquetes e o hardware do Sistema que atua com estas entidades. Os cartões de crédito, que estão em conformidade com o padrão PCMCIA, são referenciados como "PCCards" ou simplesmente "cartões PCMCIA". Estes cartões são conectados a soquetes do Sistema que, por sua vez, são agrupados em adaptadores compondo o hardware do Sistema, que atua com os soquetes e os interliga à CPU. Note que os Sistemas podem ter um ou mais soquetes e que estes são agrupados nos adaptadores. Como exemplo de Sistema com adaptadores e soquetes podemos citar um PC convencional ou PS que possua um adaptador para seu barramento (ISA, EISA[22], MCA[23]) contendo dois soquetes PCMCIA.

[22] "Extended ISA".

[23] Barramento Microcanal IBM".

Tabela 22. T-upla da tabela de configuração.

Byte	Descrição
0	Código da tabela: 1BH
1	Conexão para próxima t-upla.
2	Índice da tabela de configuração: valor a ser escrito no Registro de Configurações de Opções.
3	Campo de definição da interface: memória ou I/O, presença dos sinais BDV, WP, Pronto/Ocupado, WAIT.
4	Estrutura de descrição presentes: alimentação, "timing", espaço de endereçamento de I/O.
5	Estrutura de descrição de alimentação do cartão (se presente).
6..13	Descrição de tensão nominal, mínima, máxima, corrente de alimentação estática, corrente de alimentação média, pico de corrente, corrente em modo de economia de potência (1 byte para cada item, se presente).
14	Descrição de "timing"(1 byte, se presente).
15	Descrição de espaço de endereçamento de I/O (1 byte, se presente).
16...18	Descrição da estrutura de interrupção (3 bytes, se presente).
19	Descrição do espaço de endereçamento de memória (1 byte, se presente).
20	Mixto de informaçoes: áudio, economia de energia, cartão de apenas leitura, máximo de cartões idênticos (1 byte, se presente).

A camada de software apresenta inicialmente os serviços de soquete. Estes serviços são executados pelo processador do Sistema e provê uma interface de software padrão para a operação dos cartões PCMCIA. Para cada soquete há uma entidade de serviços. A razão destes serviços é possibilitar que todos os acessos aos cartões PCMCIA, soquetes e adaptadores sejam feitos através destes serviços. Dentre as suas atribuições, está a descrição das características do conjunto cartão e soquete ao Sistema, de modo sempre efetivo. Em outras palavras, os serviços de soquete sempre acessam os cartões mesmo que estes não estejam ainda mapeados na memória do Sistema. Isto permite que, por exemplo, antes de mapear o cartão em seu espaço de memória, o Sistema obtenha a estrutura de informações do cartão contida em sua memória de atributos.

A camada de serviços de cartões está posicionada imediatamente acima da camada de serviços de soquete. Estes serviços coordenam os acessos de software de camadas mais altas aos cartões PCMCIA. Ou seja, os serviços de cartões são responsáveis por receber as solicitações de múltiplos processos executando no Sistema e acionar os serviços de soquete de uma forma racional. Assim, os serviços de cartões fazem todos os acessos à camada hardware via serviços de soquete. Note que há apenas uma implementação serviços de cartões operando com diversos serviços de soquetes. A cada alteração de estado nos cartões PCMCIA, a camada de serviços de soquete irá encaminhar esta alteração ao serviço de cartões. Este, por sua vez, dependendo da situação, pode ou não notificar aos seus clientes interessados no específico cartão que teve alguma alteração de estado. Estes clientes poderão

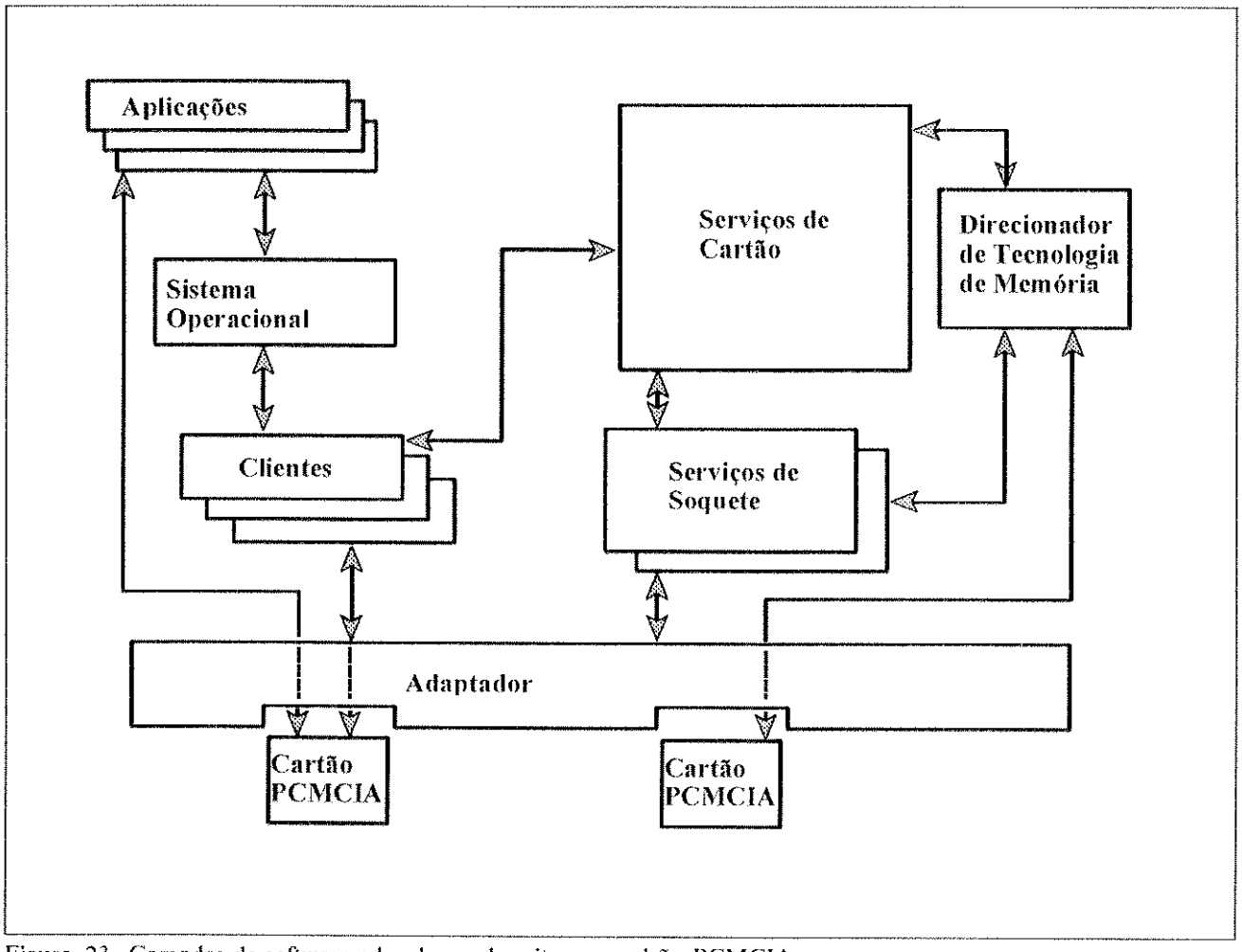


Figura 23. Camadas de software e hardware descritas no padrão PCMCIA.

ser direcionadores dos dispositivos ("device drivers") residentes ou não na memória do Sistema, programas de aplicação ou utilitários do Sistema. Por isso, a fim de evitar conflitos entre clientes, os acessos ao serviço de soquetes são feitos exclusivamente pelo serviço de cartões.

O padrão PCMCIA tem por finalidade operar com todos os tipos de memórias implementadas nos cartões de crédito. Cada qual contém seu próprio algoritmo de escrita e apagamento de memória. Para proporcionar a compatibilidade com todas as tecnologias, o serviço de cartões e o serviço de soquetes foram excluídos desta função. Assim, os direcionadores de tecnologia de memória operam no mesmo nível que os serviços de soquete e são dependentes da tecnologia das memórias por serem responsáveis por operá-las.

A última camada de software acima dos serviços de cartões pode ser uma das várias possibilidades: programas de aplicações, direcionadores de dispositivos ou utilitários do Sistema. Todos estas entidades são consideradas clientes dos serviços de cartões mas após terem configurado os cartões e os terem mapeado na memória do Sistema podem fazer acessos diretos aos mesmos.

7.5 Consumo Racionalizado de Energia

O padrão PCMCIA prevê a possibilidade de operação de cartões com economia de energia. Isto se deve aplicações em "notebooks", "laptops" e "palmtops", onde a alimentação é feita através de baterias, cuja carga deve ser otimizada.

Na solução de rede sem fio proposta, o consumo de energia tanto do adaptador quanto do Rádio é considerável, o que acarreta uma drástica redução do tempo de bateria disponível. Para atenuar este efeito faz-se uso do consumo racionalizado de energia, que é permitido pelo protocolo de comunicação citado na seção 3.3.2. Assim, a estação remota passa a observar o cabeçalho AH e, caso não haja mensagens para ela no quadro de tempo vigente, ela pode economizar energia até o momento de receber o próximo cabeçalho AH.

Neste caso, economizar energia significa reduzir as atividades do cartão PCMCIA e do Rádio que não comprometam a comunicação já estabelecida na rede. Em outras palavras, a configuração, os dados e o sincronismo com o Ponto de Acesso não podem ser perdidos.

Em se tratando do cartão PCMCIA, a economia de energia é implementada bloqueando-se o relógio dos circuitos do compressor de dados, encriptador, DMA's, UART e CSPS e mantendo-se operacionais os sinalizadores com o Sistema e os temporizadores, que garantem o sincronismo com o Ponto de Acesso. Por ser implementada em CMOS-IV[24], a interrupção do relógio das máquinas de estado resulta em considerável economia de energia. Além deste artifício, o processador residente no cartão PCMCIA pode colocar-se no estado de "Halt", aumentando assim a economia de energia. Desta forma, o temporizador, responsável por garantir o sincronismo da estação remota, faz o processador retornar ao seu estado normal de operação. O estado "Halt" do processador também se baseia na interrupção do relógio para economia de energia.

O Rádio possui três estados de operação: transmitindo, recebendo e disponível. O terceiro estado identifica, na realidade, que o Rádio não está transmitindo nem recebendo dados. É, portanto, uma forma de economizar energia.

A Tabela 23 mostra os resultados obtidos da implementação de economia de energia do cartão e do Rádio.

Tabela 23. Consumo típico de energia do adaptador para rede sem fio implementado.

Modo de Operação	Normal		Economia
	Transmitindo	Recebendo	
Cartão	1,87 W	1,87 W	0,10 W
Rádio	1,75 W	0,75 W	0,25 W
Total	3,62 W	2,62 W	0,35 W

[24] Tecnologia de circuitos integrados.

7.6 Conclusões

Neste capítulo vimos como a plataforma de rede sem fio implementada se conecta através da interface PCMCIA ao Sistema de Processamento de Dados. Para tal, discorremos sobre várias características físicas, elétricas e de software determinadas pelo padrão PCMCIA. O cumprimento de todas as especificações garante a compatibilidade entre o cartão implementado e os diversos Sistemas de Processamento de Dados disponíveis.

A padronização das especificações PCMCIA em 1990 (re-editada em 1991 com maior volume de informações) resultou em imediatas implementações por fabricantes de computadores e de adaptadores tipo cartão de crédito. Estas implementações pioneiras se atentaram principalmente para compatibilidades mecânicas, deixando a elétrica e a de software para segundo plano. Essas implementações acabaram sendo colocadas em desuso pela própria indústria e várias reuniões da associação PCMCIA aconteceram a fim de resolver os problemas resultantes de diferentes interpretações da especificação. No entanto, algumas das implementações pioneiras deram sentido a vários itens do padrão e, por assim dizer, tornaram-se exemplos concretos destes. Este trabalho de implementação de um padrão recém editado requereu grande esforço em equipe para sua compreensão e posterior implementação. Desta forma, a implementação hardware da plataforma de rede sem fio, sendo uma das implementações pioneiras do padrão PCMCIA, tornou-se uma referência para posteriores implementações.

7.7 Referência

{1} *"PC Card Standard, release 2.0", Personal Computer Memory Card International Association, Set. 1991.*

{2} *N.A.T.Resende, "Wireless/PCMCIA Card Functional Specification", Publicação Interna IBM, Centre d'Etude et de Recherche - La Gaude, França, Out. 1993.*

Capítulo 8. Sincronização entre Estações da Rede Sem Fio

Neste capítulo trataremos da sincronização dos relógios de sistemas de comunicação de dados. Em particular analisaremos os sistemas de rede de comunicação sem fio que fazem uso de rádio operando em Salto em Freqüência. O objetivo é descrever um método de sincronização de relógios das estações remotas e Ponto de Acesso.

8.1 Introdução

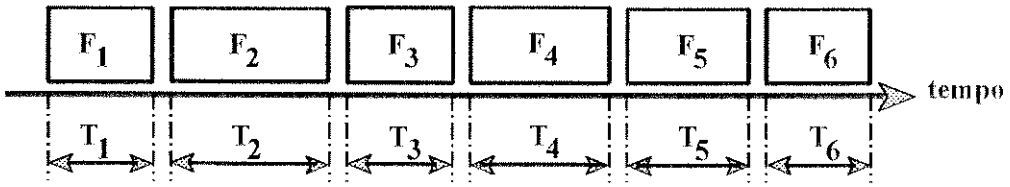
Como vimos no Capítulo 3, o protocolo de acesso ao meio físico proposto para redes sem fio estabelece o uso de Salto em Freqüência. Assim, cada quadro de tempo terá uma portadora distinta do quadro anterior e posterior. Estabelecidas pelo Ponto de Acesso, a freqüência da portadora de determinado quadro de tempo e o instante exato de salto em freqüência devem ser seguidos pelas estações remotas. Para tal, os relógios do Ponto de Acesso e estações remotas devem estar em sincronismo e ambos devem compartilhar a mesma seqüência de freqüências de operação.

O sincronismo de uma célula da rede sem fio é determinado pelo Ponto de Acesso através dos cabeçalhos das fases A, B e C (vide Capítulo 3). Estas informações são interpretadas pelo Controlador de Comunicação que, através de comandos enviados ao Rádio, garantirá o sincronismo de operação. É justamente sobre como garantir este sincronismo que trataremos a seguir.

8.2 Sincronismo de Salto em Freqüência

Para um sistema operando com Salto em Freqüência, com a arquitetura proposta no Capítulo 4, é de se esperar que a responsabilidade da mudança de portadora pelo Rádio seja do Controlador de Comunicação. Para tal, o Controlador de Comunicação necessitaria dispor de um comando a ser enviado ao Rádio para que este realizasse tal salto em freqüência. No entanto, é possível que, em algumas situações, o Controlador de Comunicação não possa diretamente comandar a troca da portadora. Isto acontece quando, por exemplo, ele está ocupado com outras atividades do protocolo de comunicação, ou de transferência de dados com o Sistema de Processamento de Dados no momento requerido. Se restringíssemos o salto em freqüência a comandos do Controlador de Comunicação, poderíamos provocar variações na duração de cada período dos quadros de tempo, o que acarretaria em perda de sincronismo.

A Figura 24 mostra a seqüência dos quadros de tempo, cada qual com uma freqüência portadora apropriada e duração respectiva. Para que tal seqüência ocorra independentemente das atividades que requerem processamento do Controlador de Comunicação, é desejável que o controle do Rádio possua a capacidade de comandar por si só o salto em freqüência, numa seqüência previamente determinada pelo Controlador de Comunicação. Esta seqüência a ser seguida está contida numa tabela constituída pelos seguintes parâmetros: número de entrada na tabela, freqüência da portadora para esta entrada e duração do respectivo quadro de tempo. Um outro parâmetro para o funcionamento correto de tal tabela de salto em freqüência é o seu comprimento, que determinará o ciclo das freqüências portadoras (vide Tabela 24).



onde:

F_j : freqüências j, j = 1, 2, ... , i

T_j : duração do salto em cada freqüência, j = 1, 2, ... ,i

Figura 24. Quadros de tempo operando com freqüências e durações distintas.

Tabela 24. Tabela de salto em freqüência.

Número de Entrada	Freqüência da Portadora	Duração do Quadro de Tempo
1	F_m	T_m
2	F_n	T_n
3	F_o	T_o
...	...	...
i	F_{m+i}	T_{m+i}

A interpretação da tabela de salto em freqüência feita pela controle do Rádio é um processo cíclico e sem fim. Ela se inicia pela primeira entrada da tabela, obtém a nova freqüência e duração do salto, e programa o Rádio com os parâmetros respectivos. A seguir, aguarda o fim do tempo respectivo e volta a realizar a mesma operação para a entrada seguinte da tabela. O processo se repete até a última entrada na tabela que é seguida pela primeira entrada reiniciando a seqüência.

É conveniente se ter mais que uma tabela de salto em freqüência para aplicações de multicélulas, nas quais os saltos em freqüências são funções de qual célula o Rádio está operando em determinado momento. A troca de uma tabela por outra é, obviamente, mais rápida e menos trabalhosa que preenchê-la novamente.

8.2.1 Preenchimento da Tabela de Salto em Freqüência

Vimos que a construção da tabela de salto em freqüência resolve diferenças de sincronismo entre estações numa rede sem fio. Discorreremos nesta seção sobre o preenchimento da tabela de salto em freqüência de modo que seu conteúdo garanta o sincronismo entre estações.

Para conveniente operação da tabela de salto em freqüência, o Controlador de Comunicação necessita carregar as informações da tabela no controlador do Rádio. Estas informações, como já vimos, são as freqüências de operação, duração destas e o número total de freqüências utilizadas (ou comprimento da tabela). Os valores a serem carregados na tabela podem ser armazenados pelo Controlador de Comunicação em memória não volátil ou mesmo serem carregados pelo Sistema de Processamento de Dados.

Esta é, portanto, uma forma de garantir que Ponto de Acesso e estação remota operem com a mesma tabela de salto em freqüência e se mantenham em sincronismo. Em outras palavras, deve-se preencher a tabela de uma estação remota antes de inseri-la em uma célula de rede sem fio com uma réplica da tabela do Ponto de Acesso. Este preenchimento inicial pode ser feito como descrito acima, através do Sistema de Processamento de Dados ao Controlador de Comunicação e que, por sua vez, carrega a tabela no controle do Rádio.

Se por algum motivo (por exemplo, interferência de sinais), o Ponto de Acesso decida evitar algumas freqüências, a presença de uma segunda tabela de salto em freqüência é conveniente. Para proceder tal supressão de freqüências, o Ponto de Acesso pode utilizar da conexão sem fio já estabelecida para transmitir a nova tabela. Esta será interpretada pelo Controlador de Comunicação da estação remota e armazenada na segunda tabela de salto em freqüência do controlador do Rádio. Finalizado o processo de preenchimento da segunda tabela pelo Controlador de Comunicação, Ponto de Acesso e estação remota trocam de tabela de operação simultaneamente e permanecem em sincronismo. Como foi dito antes, a vantagem da possibilidade de se ter uma duplicidade de tabelas está na rapidez de troca de seqüências de freqüências de operação. Uma tabela necessitaria ser preenchida antes de iniciar sua operação enquanto que, com a duplicidade da tabela, pode-se preencher a segunda fazendo-se uso da primeira a fim de manter-se a conexão já estabelecida.

8.2.2 Sincronização entre Ponto de Acesso e Estações Remotas

A seguir discorremos sobre como é realizada o sincronismo entre o Ponto de Acesso e estações remotas no Salto em Freqüência. Note que ambos as classes de estações possuem relógios distintos em seus controladores de Rádio e que necessitam, para que se garanta o sincronismo, serem iniciados simultaneamente ou ajustados no decorrer da operação. A Figura 25 mostra um desenho esquemático do Ponto de Acesso e estação remota representando seus blocos funcionais.

Nosso objetivo é sincronizar o relógio da estação remota com o relógio do Ponto de Acesso. Inicialmente o Ponto de Acesso e a estação remota já possuem a mesma tabela de salto em freqüência e, através de uma destas freqüências, a estação remota recebe uma mensagem tipo "broadcast" (cabeçalho AH, citado no Capítulo 3) do Ponto de Acesso. Ao interpretar esta mensagem, o Controlador de Comunicação da estação remota dispara o relógio do Rádio iniciando, assim, a seqüência de freqüências a ser seguida que constam na tabela. Como já vimos, o Rádio não interpreta mensagens recebidas pela rede sem fio. Além disso, o relógio que controla os saltos em freqüência reside no próprio Rádio, resultando em certo atraso entre os instantes em que o Ponto de Acesso salta de freqüência e a estação remota o acompanha.

Descreveremos agora um primeiro método de ajuste do sincronismo entre Controlador de Comunicação e controlador do Rádio baseado na leitura do relógio do segundo pelo primeiro e sua comparação com a chegada da mensagem "broadcast" (cabeçalho AH). Inicialmente o Controlador de Comunicação requer do Rádio (entenda-se controlador do Rádio) uma leitura

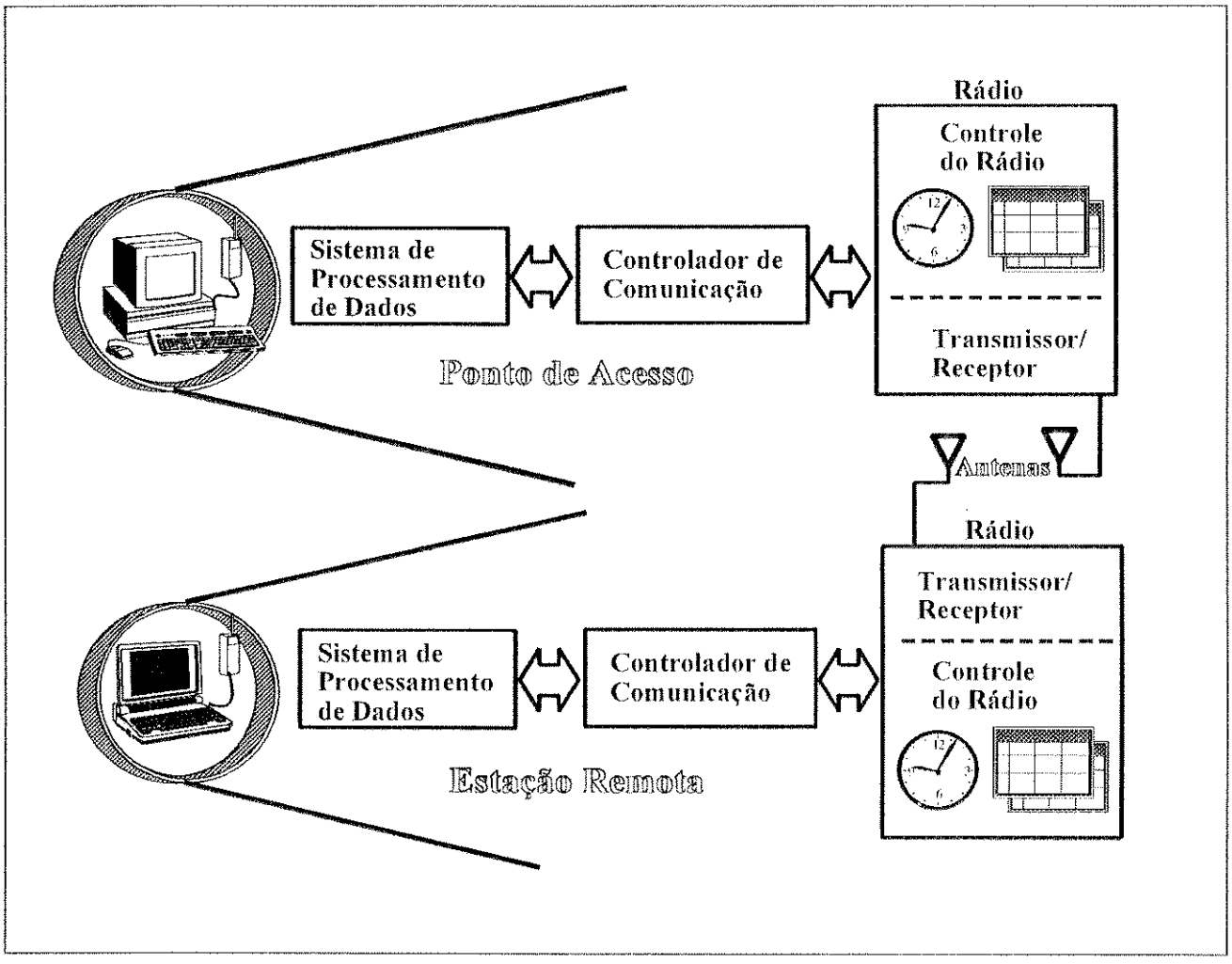


Figura 25. Diagrama esquemático de um Ponto de Acesso e estação remota.

de seu relógio informando em que instante acontecerá seu salto em freqüência. Este envio de comando e posterior recebimento de resposta possui um atraso inerente ao Rádio e ao Controlador de Comunicação, que chamaremos de T_{d1} (atraso na leitura do relógio). Ao valor informado pelo controlador do Rádio denominaremos de T_R (tempo que resta para o salto em freqüência) e T_1 será o momento no qual o Controlador de Comunicação recebe a resposta do Rádio. Através destas informações, o Controlador de Comunicação consegue determinar que o salto em freqüência do Rádio ocorrerá em $T_{Salto} = T_1 + T_R - T_{d1}$, que é representado por T_{Salto} na Figura 26. O instante no qual o Controlador de Comunicação interpreta a mensagem "broadcast" (cabeçalho AH) é assinalado como T_2. Assim, o Controlador de Comunicação é capaz de calcular o tempo no qual o Ponto de Acesso saltou de freqüência subtraindo de T_2 o atraso na transmissão e recepção da mensagem "broadcast". Este atraso denominado de T_{d2} inclui o tempo de propagação, que pode ser desprezado para pequenas distâncias como é o caso de rede sem fio, e o tempo gasto no recebimento da mensagem. Esta última parcela refere-se ao recebimento do sinal pelo Rádio, de-serialização e interpretação da mensagem. que é dependente da implementação e pode ser estimado com certa acuidade.

Uma vez que o Controlador de Comunicação possa estimar o tempo no qual o Ponto de Acesso saltou em freqüência ($T_{SaltoPA}$), ele também pode calcular a diferença de sincronismo entre o relógio de seu Rádio e do relógio do Ponto de Acesso ($T_{SaltoPA} - T_{Salto}$). Esta diferença

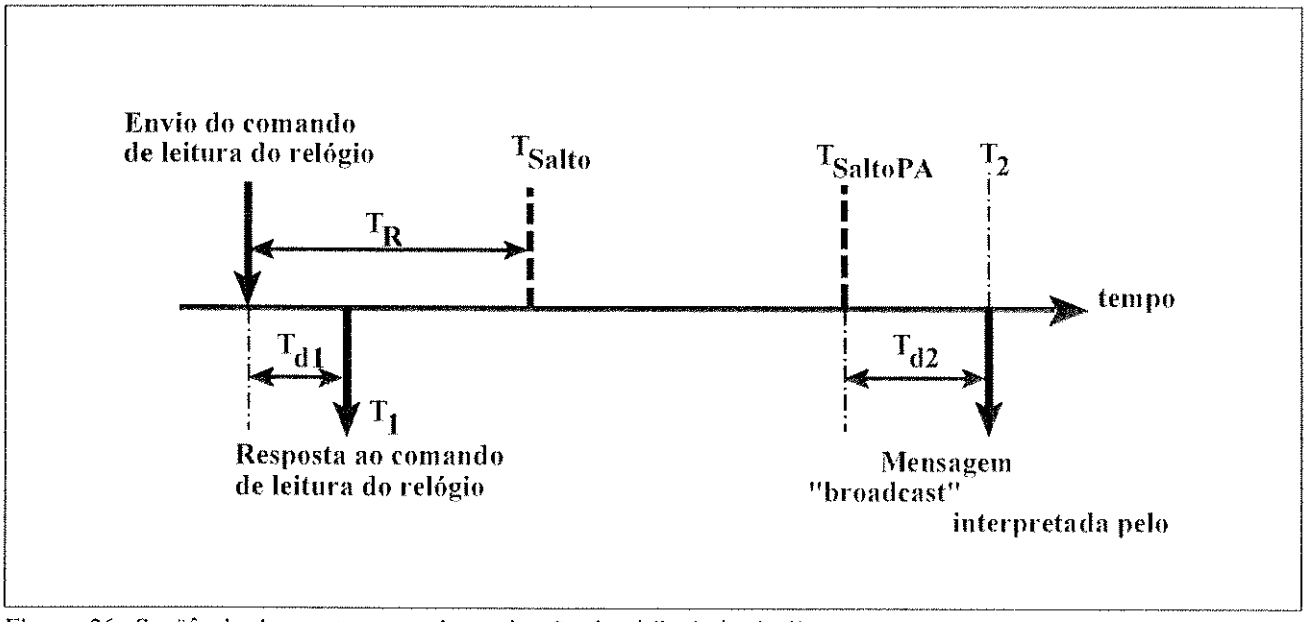


Figura 26. Seqüência de eventos para sincronização do rádio (método 1).

pode ser incorporada ao relógio do controlador do Rádio de duas formas: ajustando-se diretamente o relógio do Rádio pelo valor desejado ou pará-lo pelo por este mesmo valor.

Um segundo método para se alcançar o ajuste entre os relógios do Ponto de Acesso e estação remota, consiste em obter um aviso em adiantado do salto em freqüência a ser executado pelo controlador do Rádio. Desta forma, o Controlador de Comunicação envia um comando ao Rádio para que seja notificado com certo tempo de antecedência do salto em freqüência. Assim, com um atraso T_{d3}, o Rádio informa ao Controlador de Comunicação que está a T_A do salto em freqüência. Este atraso T_{d3}, assim como T_{d1}, é um atraso inerente ao Rádio que pode ser informado pelo mesmo. O momento de recebimento da mensagem é indicado por T_1 na Figura 27.

Com estes dados, o Controlador de Comunicação pode calcular o tempo de salto da estação remota, que ocorrerá em $T_{Salto} = T_1 + T_A - T_{d3}$.

Ao interpretar a mensagem "broadcast" enviada pelo Ponto de Acesso, o Controlador de Comunicação da estação remota, da maneira descrita anteriormente, consegue avaliar o instante do salto em freqüência do primeiro ($T_{SaltoPA} = T_2 - T_{d2}$). Da mesma forma que anteriormente, a diferença de tempo entre o salto em freqüência do Ponto de Acesso e estação remota ($T_{SaltoPA} - T_{Salto}$) pode ser incorporada ao relógio da estação remota através de um ajuste ou mesmo de uma parada temporária deste relógio.

8.3 Implementação

Através do método de sincronismo descrito chegamos à implementação de um conjunto de comandos a serem enviados ao Rádio pelo Controlador de Comunicação. O canal para envio destes comandos deve ser diferenciado do canal de comunicação dos dados, uma vez que comandos podem ser enviados independentemente da transmissão ou recepção simultânea. Assim, a interface entre o Controlador de Comunicação e o Rádio possui um conversor serial-paralelo-serial (CSPS), o qual só opera com dados transmitidos e recebidos pelo Rádio e uma

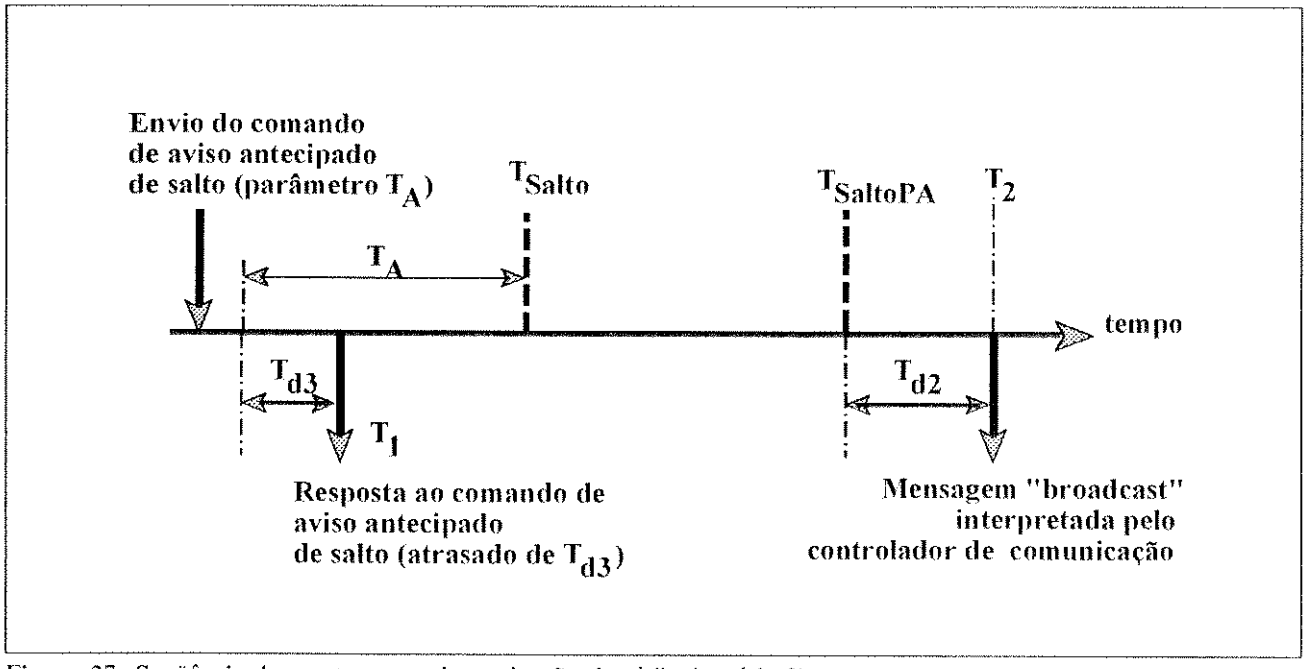


Figura 27. Seqüência de eventos para sincronização do rádio (metódo 2).

UART[25]*. Esta UART será justamente utilizada para enviar os comandos ao Rádio e para receber do mesmo as repostas.*

A fim de possibilitar a operação do método descrito, os seguintes comandos são implementados:

1. *SELECIONAR TABELA 1/2: seleciona-se com qual tabela vai se operar (tabela 1 ou 2). A partir deste comando o relógio de salto em freqüência começa a operar.*
2. *SELEÇÃO DO COMPRIMENTO DA TABELA 1/2: seleciona-se o comprimento a ser utilizado pela tabela 1 ou 2.*
3. *PREENCHIMENTO DA TABELA 1/2: escrita do número de entrada, freqüência e duração na tabela 1 ou 2.*
4. *VALIDAÇÃO DA TABELA 1/2: cálculo da adição simples dos dados que constam na tabela 1 ou 2 e retorno do resultado ao Controlador de Comunicação para verificação do preenchimento das tabelas.*
5. *LEITURA DOS ATRASOS INERENTES DO RÁDIO: após este comando o Rádio retorna os valores T_{d1} e T_{d3}.*
6. *PARADA DO RELÓGIO: este comando pode ser utilizado para se ajustar o relógio de salto em freqüência ou simplesmente parar a operação de salto em freqüência.*
7. *MODIFICAR VALOR DO RELÓGIO: altera o valor do relógio do Rádio a fim de ajustá-lo com o relógio do Ponto de Acesso.*
8. *LEITURA DO RELÓGIO: após este comando o Rádio retorna com o valor residual do relógio, o que significa o tempo que resta para o salto em freqüência.*

[25] do inglês "Universal Asynchronous Receiver/Transmiter".

9. *SELEÇÃO DE NOTIFICAÇÃO DE SALTO COM ANTECEDÊNCIA: ao enviar este comando, o controlador de comunicação anexa o valor de antecedência para o qual necessita a informação de salto em freqüência.*

10. *NOTIFICAÇÃO DE SALTO: após este comando, o Rádio utiliza do valor do comando 9 e notifica ao Controlador de Comunicação do salto em freqüência com a antecedência requerida.*

8.4 Conclusões

Neste capítulo foi descrito um método de sincronização para redes sem fio operando com Salto em Freqüência. O método descreve a sincronização entre o Ponto de Acesso e estações remotas, discorrendo também sobre como efetuar a sincronização entre as entidades de controles numa estação (Controlador de Comunicação e controlador do Rádio).

O método de sincronização descrito apresenta como proposta a utilização de tabelas de salto em freqüência a serem seguidas pelo Ponto de Acesso e estações remotas. Para garantir o sincronismo na seqüência contida nas tabelas, apresentamos como implementação um conjunto de comandos que estabelece uma interface entre o Controlador de Comunicação e o Rádio.

Esta interface pode ser suficientemente especificada para se tornar um padrão. Desta maneira, adaptadores (Controladores de Comunicação) e Rádios tornam-se compatíveis, mesmo quando desenvolvidos por diferentes fabricantes.

8.5 Referências

{1} *K.S. Natarajan, C.C. Huang, D.F Bantz, "Medium Access Control Protocol for Wireless LANs", IEEE 802.11/92-39, Mar. 1992.*

{2} *E. Antunes, D. Bantz, E. Dal Bello, M. Ferraz, B. Tavares, "Command Set And Procedure For Synchronization of Frequency Hopping Control Clocks", IBM Internal Publication, Mar. 1994.*

Capítulo 9. Conclusão

9.1 Resumo

Este trabalho apresentou um dispositivo que implementa uma plataforma hardware para redes sem locais sem fio. Iniciamos com discussões gerais sobre redes sem fio e finalizamos apresentando uma implementação hardware para "notebooks" ou qualquer computador que possua a interface PCMCIA disponivel.

Discorremos sobre os conceitos e parâmetros envolvidos na implementação de uma rede sem fio. Descrevemos os protocolos de comunicação utilizados hoje para rede sem fio, cujas especificações foram apresentadas ao Comitê IEEE. Em especial, apresentamos o protocolo de acesso ao meio físico que motivou a elaboração da plataforma hardware aqui estudada. Este protocolo de acesso, com transmissões síncronas e assíncronas, apresenta vantagens únicas em relação aos demais, apesar de mais elaborado.

No Capítulo 4, apresentamos uma possível plataforma hardware capaz de implementar o protocolo proposto. Como parte integrante desta plataforma, foram citados os blocos funcionais que a compõem. Dentre estes, o encriptador, compressor de dados, interface com o Sistema de Processamento de Dados e interface com Rádio foram discutidos nos capítulos posteriores.

O encriptador é uma implementação inédita de algoritmos desenvolvidos para aplicações em rede sem fio. Ao encriptador incorporou-se a autenticação dos dados, que garante a integridade da mensagem transmitida e a segurança adicional no sistema de comunicação sem fio. O autenticador também é uma implementação inédita do algoritmo apresentado.

Descrevemos como se processa a compressão de dados nos algoritmos LZ1 e LZ2 propostos por Lempel e Ziv. Apresentamos uma particularização do algoritmo LZ1 e descrevemos sua implementação hardware. Resultados de taxa de compressão e vazão completaram a análise sobre compressão de dados.

A implementação hardware do adaptador para redes sem fio tem como premissa sua utilização em "notebooks", por se acreditar que os usuários de computadores portáteis serão os mais beneficiados com este tipo de rede. Assim, para a implementação de sua interface com o Sistema de Processamento de Dados, observou-se o cumprimento das especificações do padrão PCMCIA, amplamente utilizado em computadores portáteis. Apresentamos as características físicas, elétricas e de software deste padrão.

Graças ao protocolo de acesso proposto, é possível um consumo racionalizado de energia, o que é realmente útil em se tratando de computadores portáteis que operam a maior parte do tempo com baterias. No Capítulo 7 é mostrada a maneira como a plataforma consegue implementar um modo de economia de energia que pode ser utilizado para racionalizar o consumo como especificado pelo protocolo de acesso.

Finalmente, é apresentada uma interface com o Rádio que permite a sincronização das estações da rede sem fio. É também mostrado como esta sincronização é efetuada e os blocos funcionais que devem compor o Rádio para tal implementação.

9.2 Conclusões e Contribuições

O desenvolvimento da plataforma hardware foi feito com base em certas premissas que direcionaram o trabalho. Estas premissas são: utilização de Salto em Freqüência operando nas faixa de freqüências permitidas pelo FCC norte-americano, taxa de transmissão e segurança de dados comparáveis às das redes convencionais e utilização em computadores portáteis. Todos esses requisitos são compatíveis com o protocolo de acesso ao meio físico proposto, o que conduziu ao desenvolvimento de uma plataforma hardware capaz de implementá-lo.

A utilização do Salto em Freqüência em redes sem fio traz vantagens se comparado com o método de Seqüência Direta (Capítulo 2). Através do Salto em Freqüência pode se implementar um uso seletivo de freqüências de modo a se evitar interferências. Além disso, este método permite a presença de várias células de rede sem fio no mesmo espaço físico através da utilização de freqüências ortogonais. No entanto, o Salto em Freqüência aumenta a complexidade do Rádio, que deve possuir, além do transmissor e receptor operando em faixas adjacentes de 1 MHz, a capacidade do salto em freqüência, o que implica em várias freqüências de operação para o mesmo Rádio e na necessidade de manter o sincronismo com o Controlador de Comunicação e com o Ponto de Acesso. Esta manutenção de sincronismo exige inteligência no Rádio para interpretar os comandos do Controlador de Comunicação e saltar de freqüência com precisão. Vimos, então, que o Rádio se cerca de várias restrições, tornando-se um componente complexo. Porém graças a ele a rede proposta pode desfrutar das vantagens de operar com Salto em Freqüência.

Vimos que o protocolo de acesso ao meio físico proposto é mais elaborado que os demais protocolos apresentados ao Comitê IEEE. Isto se justifica pela possibilidade de se operar com serviços síncronos e assíncronos, além da importante racionalização de energia em se tratando de computadores portáteis. É de se esperar, no entanto, que a implementação do protocolo acesso proposto tenha uma vazão inferior à dos demais protocolos propostos ao Comitê IEEE. A segmentação de pacotes dentro de um quadro de tempo necessita de campos adicionais, tais como endereços de destino e de origem e código detetor de erro (ou CRC) que se repetem a cada pacote e não contém informação propriamente dita. A isto podemos chamar de "overhead" do protocolo, que efetivamente reduz a vazão. Soma-se ao "overhead" descrito acima, o fato da comutar-se o Rádio do estado de transmissão para recepção com o propósito de receber a confirmação (ACK) da estação de destino que o pacote chegou integralmente. A própria confirmação, além dos tempos de comutação, é mais um "overhead" ao protocolo. No entanto, estes "overheads" se justificam para que se tenha uma conexão confiável. Reduz-se a vazão mas se ganha em confiabilidade e tempo de retransmissão. Ou seja, como cada pacote é acompanhado da confirmação (ACK) da estação de destino, garante-se, pacote a pacote, a transmissão integra da mensagem. Se, por ventura, um dos pacotes apresentar discrepâncias na verificação do seu código detector de erro, apenas o referido pacote deve ser retransmitido e não toda a mensagem. A comunicação via Rádio de baixa potência, como é o caso (100 mW), permite interferências. Assim, o desempenho do protocolo, conforme implementado, apresenta grandes ganhos relativos à vazão em ambientes ruidosos.

Com o objetivo de deixar a taxa de transmissão de dados da rede sem fio a mais próxima possível das taxas de redes com fio, apesar dos "overheads" descritos acima e da limitação do Rádio em operar com faixa de 1 Mbps, incluiu-se a compressão de dados na plataforma hardware. O compressor proposto apresenta boas características em termos de taxa de

compressão e velocidade de operação em relação a outros compressores. No entanto, sua implementação, assim como as dos demais compressores, não é simples. O uso de CAM, RAM e máquinas de estado aumentam a complexidade e a área de silício empregada na confecção do hardware.

Assim como o compressor, o encriptador, que apresenta vantagens pela simplicidade da implementação, não é pequeno em termos de hardware gasto para sua realização. Observa-se que os encriptadores serão apenas úteis com registros maiores que 61 ou 89 posições. Entretanto, tanto o compressor quanto o encriptador, quando implementados em hardware, passam a funcionar de modo transparente ao usuário. Ou seja, não há queda de performance do sistema ou atrasos adicionais pelo fato dos dados na rede estarem sendo comprimidos e encriptados.

É interessante notar que, durante o desenvolvimento deste trabalho, citamos duas classes de estações da rede sem fio, Ponto de Acesso e estação remota que, por fim, têm a mesma plataforma hardware. Isto traz benefícios ao usuário que, por ventura, queira desmembrar sua rede sem fio em outras. Com a implementação proposta, qualquer estação pode se tornar Ponto de Acesso ou remota.

Finalmente, o mais importante atrativo das redes sem fio é alcançado pela implementação da interface PCMCIA. Junto com os computadores portáteis, as redes sem fio ganham o real sentido de mobilidade. Com a implementação proposta, os computadores portáteis podem se conectar a sistemas e ainda assim operar "sem fios".

9.3 Sugestões para Continuidade deste Trabalho

As análises de vazão disponíveis para o protocolo de acesso ao meio físico não levam em consideração os "overheads" de segmentação em pacotes das mensagens, de confirmação (ACK) nem o tempo de comutação do Rádio. Um estudo que contemplasse estes fatores poderia dar a idéia exata dos fatores que mais contribuem para a redução de vazão.

No sistema de encriptação utilizado, espera-se que as sementes (valores iniciais dos registros de deslocamento) sejam fornecidos pelo Ponto de Acesso, de tempos em tempos. A geração das sementes por geradores pseudo-aleatórios poderia ser analisada com o rigor necessário.

O atraso ocasionado pelo protocolo para se estabelecer uma comunicação para uma estação remota está diretamente relacionado com o número de usuários da rede e com a duração do quadro de tempo. Uma vez que a estação necessita transmitir dados, ela faz um acesso assíncrono (fase C) e aguarda a alocação de faixa conveniente pelo ponto de acesso. Seria interessante conhecer as figuras de atraso em relação ao número de usuários e duração do quadro de tempo.