

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA
DA COMPUTAÇÃO

Aluna: Giani Carla Ito

BANCOS DE DADOS MÓVEIS:
UMA ANÁLISE DE SOLUÇÕES PROPOSTAS
PARA GERENCIAMENTO DE DADOS

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

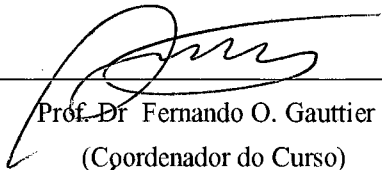
Orientador: Murilo Silva de Camargo

Florianópolis, abril de 2001

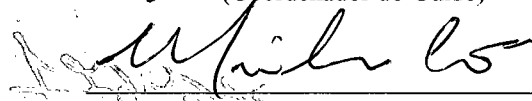
BANCOS DE DADOS MÓVEIS: UMA ANÁLISE DE SOLUÇÕES PROPOSTAS PARA GERENCIAMENTO DE DADOS

Aluna: Giani Carla Ito

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

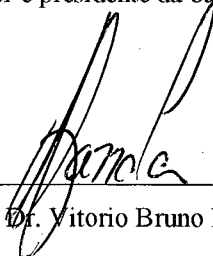


Prof. Dr. Fernando O. Gauttier
(Coordenador do Curso)

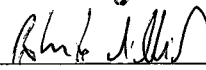


Prof. Dr. Eng. Murilo Silva de Camargo
(Orientador e presidente da banca)

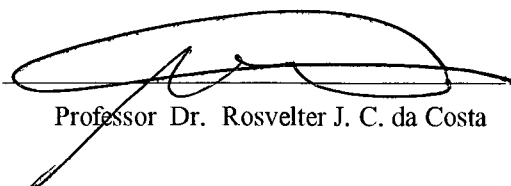
Banca Examinadora



Professor Dr. Vitorio Bruno Mazzola



Professor Dr. Roberto Willrich



Professor Dr. Rosvelter J. C. da Costa

Eu vejo a vida melhor no futuro
Eu vejo isto por cima do muro de hipocrisia
Que insiste em nos rodear
Eu vejo a vida mais clara e farta
Repleta de toda satisfação que se tem direito
Do firmamento ao chão
Eu quero crer no amor numa boa
Que isto valha pra qualquer pessoa
Que realizar a força que tem uma paixão
Eu vejo um novo começo de era
De gente fina, elegante e sincera
Com habilidade
Pra dizer mais sim do que não
Hoje o tempo voa,
Escorre pelas mãos
Mesmo sem sentir
Que não há tempo que volte,
Vamos viver tudo o que há pra viver
Vamos nos permitir...

Lulu Santos

A você *Norio*, um grande homem,
amigo, companheiro, pessoa
indescritível, presente em todos os
momentos...

AGRADECIMENTOS

Amigos são peças fundamentais em nossas conquistas.

Agradeço a todos os amigos que colaboram para que este trabalho se concretizasse.

Meus especiais agradecimentos à orientação e incentivo do Professor *Murilo*;

Ao apoio e compreensão de meu marido *Norio*;

A força sempre presente de minha mãe *Luiza*;

Ao auxílio e colaboração de minha irmã *Vivian*, das amigas *Lúcia*, *Yumi* e *Fabiany*.

Agradeço principalmente a Deus que me fortaleceu durante este processo.

SUMÁRIO

CAPÍTULO I	2
INTRODUÇÃO	2
1.1 APRESENTAÇÃO	2
1.2 METODOLOGIA	3
1.3 ESTRUTURA DA DISSERTAÇÃO.....	3
CAPÍTULO II	5
COMPUTAÇÃO MÓVEL	5
2 ARQUITETURA DA COMPUTAÇÃO MÓVEL	6
2.1 COMPUTADORES MÓVEIS	6
2.2 COMPUTADORES FIXOS	7
2.3 EVOLUÇÃO DA COMPUTAÇÃO MÓVEL	9
2.4 UTILIZAÇÃO PROFISSIONAL	11
2.5 APLICAÇÕES	12
2.6 VANTAGENS DA COMPUTAÇÃO MÓVEL	13
2.7 RESTRIÇÕES.....	13
2.8 TIPOS DE APARELHOS MÓVEIS.....	14
2.8.1 <i>Modelo Cliente/Servidor</i>	16
2.8.2 <i>Modelo Cliente/Agente/Servidor</i>	17
2.8.3 <i>Cliente/Interceptador/Servidor</i>	17
2.8.4 <i>Modelo Peer-to-Peer (P2P)</i>	18
2.8.5 <i>Modelo de Agentes Móveis</i>	19
2.9 ASPECTOS RELACIONADOS À COMPUTAÇÃO MÓVEL	20
2.9.1 <i>Handoff</i>	20
2.9.2 <i>Mobilidade</i>	21
2.9.3 <i>Desconexão</i>	22
2.9.4 <i>Adaptação</i>	26
CAPÍTULO III	28
COMPUTAÇÃO SEM FIO	28
3 WIRELESS COMPUTING (COMPUTAÇÃO SEM FIO)	29
3.1 ARQUITETURA <i>AD HOC</i>	30
3.2 ARQUITETURA CELULAR	31

3.3	ARQUITETURA DE REDE	33
3.3.1	<i>Wireless LANs</i>	34
3.3.2	<i>Wireless WAN</i>	35
3.4	PROTOCOLOS MÓVEIS	39
3.4.1	<i>IP Móvel</i>	39
3.5	PROTOCOLO DE APLICAÇÃO SEM FIO	42
3.5.1	<i>Wireless Markup Language (WML)</i>	43
3.5.2	<i>Wireless Telephony Application (WTA)</i>	44
3.5.3	<i>Funcionamento do WAP</i>	44
CAPÍTULO IV		46
4	ESTRUTURA DE BANCO DE DADOS DISTRIBUÍDO	47
4.1	VANTAGENS E DESVANTAGENS	51
4.2	PROCESSAMENTO DE CONSULTAS	52
4.3	GERENCIAMENTO DE TRANSAÇÕES.....	54
4.3.1	<i>Propriedades ACID</i>	55
4.4	CONTROLE DE CONCORRÊNCIA.....	56
4.5	CONSISTÊNCIA DE DADOS.....	57
CAPÍTULO V		59
5	ARQUITETURA DE BANCO DE DADOS MÓVEIS	61
5.1	GERENCIAMENTO DE DADOS MÓVEIS	63
5.2	PARADIGMAS DE ACESSO AOS DADOS MÓVEIS.....	64
5.2.1	<i>Caching</i>	64
5.2.2	<i>Difusão de dados</i>	66
5.3	GERENCIAMENTO DE TRANSAÇÕES MÓVEIS.....	72
5.3.1	<i>Modelo de Transação Móvel</i>	74
5.3.2	<i>Modelos de Transações Móveis</i>	75
5.3.3	<i>Tipos de Mobilidade de uma Transação</i>	77
5.3.4	<i>Diferentes Plataformas de Execução de uma Transação Móvel</i>	78
5.3.5	<i>Graus de Mobilidade de uma Transação</i>	79
5.3.6	<i>Requisitos de uma Transação Móvel</i>	81
5.4	PROCESSAMENTO DE CONSULTAS MÓVEIS.....	81
5.4.1	<i>Dependência de Localização de Dados</i>	83
5.4.2	<i>Gerenciamento de Localização</i>	85
5.5	REPLICAÇÃO DE DADOS	86
5.5.1	<i>Graus de Replicação</i>	87
5.5.2	<i>Tipos de replicação</i>	88
5.5.3	<i>Modelos de Replicação Otimista de Dados</i>	92
5.5.4	<i>Modelo de Replicação Peer-to-peer</i>	93
5.5.5	<i>Modelo de replicação Cliente/Servidor</i>	94
5.5.6	<i>Modelo WARD</i>	94
5.6	RECUPERAÇÃO DE FALHAS	96
5.6.1	<i>Localização</i>	98
5.6.2	<i>Desconexão</i>	98
5.6.3	<i>Energia</i>	99
5.6.4	<i>Rede</i>	99

5.6.5	<i>Falhas</i>	99
5.6.6	<i>Estratégias de Recuperação em um Ambiente Móvel</i>	100
5.7	SEGURANÇA EM AMBIENTES MÓVEIS	103
5.7.1	<i>Políticas de Segurança</i>	104
5.7.2	<i>Princípios Gerais de Segurança</i>	105
5.7.3	<i>Métodos de Ataque</i>	105
5.7.4	<i>Infiltração</i>	107
5.7.5	<i>Encriptação de Dados</i>	108
5.7.6	<i>Assinatura Digital</i>	111
5.7.7	<i>Firewall</i>	113
CAPÍTULO VI		116
TECNOLOGIAS DE BANCO DE MÓVEIS		116
6	TECNOLOGIAS DE BANCO DE MÓVEIS	117
6.1	ORACLE8I LITE	117
6.1.1	<i>Oracle Lite</i>	118
6.1.2	<i>iConnect</i>	118
6.1.3	<i>Internet Web-to-Go</i>	122
6.2	DB2 EVERYPLACE.....	123
6.2.1	<i>DB2 Everyplace database</i>	125
6.2.2	<i>DB2 Everyplace Sync Server</i>	125
6.2.3	<i>DB2 Everyplace Personal Application Builder</i>	127
6.2.4	<i>Segurança</i>	128
6.3	SYBASE SQL ANYWHERE STUDIO	128
6.3.1	<i>Sybase Adaptive Server Anywhere</i>	129
6.3.2	<i>Adaptive Server Anywhere Ultralite</i>	129
6.3.3	<i>MobileBuilder</i>	132
6.4	SQL SERVER 2000.....	133
6.4.1	<i>O Microsoft SQL Server 2000 Windows CE</i>	133
CAPÍTULO VII		136
CONCLUSÃO		136
7.1	RESUMO DO TRABALHO.....	136
7.2	CONCLUSÕES	138
7.3	RELEVÂNCIA DO TRABALHO	139
7.4	PERSPECTIVAS FUTURAS.....	140
GLOSSÁRIO		141
REFERÊNCIAS BIBLIOGRÁFICAS		147

INDICE DE FIGURAS

Figura 1 - Arquitetura Computação Móvel.....	8
Figura 2 - Modelo Cliente/Servidor.....	16
Figura 3 - Modelo Cliente/Agente/Servidor	17
Figura 4 - Modelo Cliente/Interceptador/Servidor	18
Figura 5 - Modelo <i>Peer-to-Peer</i>	19
Figura 6 - Processo de Handoff.....	21
Figura 7 - Transição de Estados	23
Figura 8 - Modos de Operação de uma Unidade Móvel.....	26
Figura 9 - Modos de Adaptação	27
Figura 10- Rede Ad-Hoc.....	30
Figura 11 - Modelo ideal para Células.....	32
Figura 12 - Arquitetura Celular	33
Figura 13 - Arquitetura de Rede.....	34
Figura 14 - Rede WAN com Satélite.....	37
Figura 15 - Arquitetura básica de um sistema celular	38
Figura 16 - Operações de registro no IP móvel.....	42
Figura 17 - Funcionalidade WAP.....	45
Figura 18 - Banco de Dados Centralizado	48
Figura 19 - Ambiente de um Banco de Dados Distribuído.....	49
Figura 20 - Processamento de Consultas	52
Figura 21 - Arquitetura de Banco de Dados Móveis.....	62
Figura 22 - Difusão de Dados	68
Figura 23 - Comunicação Assimétrica.....	70
Figura 24 - Exemplos de Discos de Difusão.....	71

Figura 25 - Modelo de Transação de Banco de Dados Móveis	75
Figura 26 - Execução de uma transação centralizada e distribuída.....	78
Figura 27 - Execução de uma Transação em Fragmentos	79
Figura 28 - Graus de Mobilidade de uma Transação.....	80
Figura 29 - Exemplo de Dependência de Localização de Dados.....	83
Figura 30 - Reintegração de Dados	87
Figura 31 - Tipos de Replicação de Dados	90
Figura 32 - Exemplo de Tipos de Replicação	92
Figura 34 - Handoff - Estratégia de Recuperação	101
Figura 35 - Modelo de Criptografia.....	109
Figura 36 - Criptografia de Chave Pública	110
Figura 37 - Esquema Assinatura Digital.....	112
Figura 39 - Replicação Avançada.....	120
Figura 40 - Esquema de Funcionamento Consolidator.....	121
Figura 41 - Arquitetura Web-to-Go.....	122
Figura 42 - Funcionamento DB2 Everywhere	124
Figura 43 - Arquitetura DB2 Everyplace.....	126
Figura 44 - Esquema de funcionamento Mobilink	131

INDICE DE TABELAS

Tabela 1 - Tipos de Aparelhos Móveis.....	15
Tabela 2 - Computação Móvel e Redes Sem Fio	29
Tabela 3 – Características do IP Móvel.....	40
Tabela 5 - Tipos de Replicação	91
Tabela 6 - Diferenças SQL Remote e Mobilink.....	132
Tabela 7 - Comparação entre diversos Banco de Dados Móveis.....	135

RESUMO

O volume de dados processados e principalmente armazenados tem crescido de forma exponencial, por este motivo estão surgindo tecnologias novas que possam acompanhar esta evolução. Grandes conquistas tecnológicas do homem estão se miniaturizando em alta velocidade. É importante que o banco de dados busque adaptabilidade à velocidade das transformações que ocorrem em todos os setores computacionais, conferindo uma crescente importância para a habilidade de armazenar, gerenciar e recuperar dados.

O trabalho proposto tem como objeto de estudo banco de dados móveis com ênfase em soluções propostas para gerenciamento de dados, analisando aspectos como gerenciamento de transações, processamento de consultas, replicação de dados, recuperação de falhas e segurança de dados. Além dos aspectos de gerenciamento de dados, o trabalho presente aborda a aplicabilidade dos bancos de dados móveis, sua arquitetura, vantagens, restrições e objetivos, reunindo conceitos sobre computação móvel, comunicação sem fio, protocolos móveis, banco de dados distribuídos e os principais produtos existentes no mercado relacionado à plataforma móvel.

PALAVRAS-CHAVE: Banco de dados móveis, redes sem fio, computação móvel, bancos de dados distribuídos.

ABSTRACT

The volume of processed data and mainly stored data has grown exponentially. Because of this, new technologies able to accompany this revolution are appearing. Mankind's technological conquests are becoming increasingly smaller at high-speed. It is important for databases to be adaptable to the speed of the transformations which occur in all computational sectors, thus conforming the existence of a growing importance of being able to store, manage and retrieve data.

The proposed project aims to study mobile databases with an emphasis on proposed solutions for management of data, analyzing aspects as transaction management, query processing, data replication, failure recoveries and data security. Beyond the aspects of data management, this project addresses the applicability of the mobile databases, their architecture, advantages, restrictions and objectives, gathering concepts about mobile computing, wireless computing, mobile protocols, databases distributed and the main existent products on the market related to the mobile platform.

KEYWORDS: *Mobile Databases, Wireless Computing, Mobile Computing, Distributed Databases*

CAPÍTULO I

INTRODUÇÃO

1.1 Apresentação

“Somos todos nômades e carecemos de suporte de sistemas que nos auxiliem em nossas várias formas de mobilidade”.

Leonard Kleinrock

Com o crescimento da necessidade de mobilidade, surgem novas maneiras de utilização do computador a qualquer hora e em qualquer lugar. A diminuição de tamanho, peso e energia consumida tem tornado os meios de ligação entre computadores cada vez mais flexíveis, onde recursos não necessitam de uma localização fixa e nem precisam estar fisicamente conectados para se comunicar.

Dois avanços na tecnologia colaboraram para esta evolução. O primeiro diz respeito ao desenvolvimento de computadores portáteis com um poder de processamento cada vez maior, tamanho reduzido e preços acessíveis. O segundo trata-se do advento das redes de computadores, que apareceram no início da década de 60, com o projeto e implementação de sistemas centralizados, contendo um grande número de terminais espalhados. Atualmente é possível a comunicação de redes sem fio, através de mecanismos de ondas de rádio, satélite, TV, sistemas de navegação, entre outros.

O avanço da tecnologia de comunicação sem fio e a crescente utilização de computadores portáteis dá início a um novo paradigma denominado computação móvel, onde o usuário pode acessar dados e recursos do sistema de diferentes pontos de acesso a qualquer momento, limitado somente pela condição de que esteja localizado dentro de uma área geográfica determinada. Comparada à computação tradicional, já incorporada naturalmente na vida das pessoas, a computação móvel promete uma revolução mais abrangente, devido à sua maior flexibilidade de localização podendo ocasionar grandes mudanças no comportamento social e corporativo.

Esta pesquisa tem como objeto de estudo banco de dados móveis com ênfase em soluções propostas para gerenciamento de dados. A utilização de banco de dados móveis propicia facilidades de locomoção física que possibilitam aos usuários acesso aos dados a qualquer hora e em qualquer lugar. Usuários têm a possibilidade de participar de congressos, convenções e reuniões de negócios e realizar visitas a clientes e fornecedores, tendo acesso a seus arquivos pessoais armazenados em uma estação de trabalho distante, participando de teleconferências e efetuando, normalmente, suas tarefas computacionais diárias, mesmo quando distantes de sua residência ou local de trabalho.

Levando-se em consideração os elementos apresentados anteriormente, o trabalho proposto tem como finalidade à realização de um estudo dos principais aspectos de gerenciamento de dados móveis, como gerenciamento de transações, processamento de consultas, replicação de dados, recuperação de falhas e segurança de dados.

Além dos aspectos de gerenciamento de dados, o trabalho presente aborda a aplicabilidade dos bancos de dados móveis, sua arquitetura, vantagens, restrições e objetivos, reunindo conceitos sobre computação móvel, comunicação sem fio, protocolos móveis, banco de dados distribuídos e os principais produtos existentes no mercado relacionado à plataforma móvel.

1.2 Metodologia

Os dados obtidos na realização deste trabalho foram através de pesquisa bibliográfica e análise do conteúdo de artigos e relatórios técnicos. Por se tratar de um assunto relativamente novo, constatou-se que a produção científica sobre o tema apresentado é escassa, já que poucos trabalhos semelhantes ao que se apresenta foram encontrados no período estudado (04/2000 a 02/2001), nas fontes pesquisadas.

Este documento utiliza como fontes primárias de pesquisa (PITOURA & SAMARAS, 1998), (MATEUS & LOUREIRO, 2000), (RAMAKRISHNAM, 2000) e documentos oficiais de congressos, *workshops*, etc, artigos, revistas, relatórios técnicos, projetos e *sites* da Internet.

1.3 Estrutura da Dissertação

Os dados obtidos através de levantamento bibliográfico e da análise do conteúdo de artigos e relatórios técnicos possibilitaram organizar este estudo em sete capítulos.

No capítulo II apresenta-se inicialmente a importância da computação móvel para o contexto deste estudo, suas aplicabilidades, vantagens, restrições, arquitetura e modelos. Aspectos como estrutura móvel, *handoff*, desconexão, adaptação e mobilidade fazem parte do ambiente móvel e são abordados neste capítulo.

No capítulo III descreve-se conceitos relacionados à computação sem fio, com ênfase em três tipos de arquiteturas: arquitetura *ad hoc*, arquitetura celular e arquitetura de rede. O capítulo aborda também os principais protocolos móveis: WAP (*Wireless Application Protocol*) e o IP (*Internet Protocol*) móvel.

No capítulo IV estuda-se a definição de banco de dados distribuídos, devido a sua importância para o ambiente móvel. São descritos alguns aspectos relacionados a banco de dados distribuídos, tais como: processamento de consultas, gerenciamento de transações, controle de concorrência e consistência de dados.

No capítulo V aborda-se conceitos relacionados a banco de dados móveis, sua arquitetura, vantagens e restrições em sua utilização e principais aplicabilidades. São apresentados alguns aspectos fundamentais de gerenciamento de dados móveis como: paradigmas de acesso aos dados (*caching* e *broadcasting*), gerenciamento de transações, processamento de consultas, replicação de dados, recuperação de falhas e segurança.

No capítulo VI apresentam-se algumas tecnologias atuais com suporte a banco de dados móveis, como *Oracle 8i Lite*, *SQL Server CE*, *DB2 Everywhere*, e *SQL Anywhere Studio*.

O capítulo VII apresenta um resumo sobre o trabalho desenvolvido, conclusões obtidas e perspectivas futuras relacionadas ao ambiente móvel.

CAPÍTULO II

COMPUTAÇÃO MÓVEL

O mundo social e corporativo depende fortemente do acesso as informações, como ponto-chave na posição das empresas frente ao mercado competitivo. A transmissão, atualização ou acesso aos dados a qualquer hora e de qualquer lugar, torna-se um diferencial importante para a realização de negócios.

Para suprir esta necessidade, a computação móvel utiliza computadores portáteis e comunicação sem fio para permitir aos usuários trabalharem fora de ambientes fixos. O ambiente móvel baseia-se na capacidade que os usuários tem, munidos de um dispositivo móvel (*laptops, handhelds, etc*), se comunicarem com a parte fixa da rede e possivelmente com outros dispositivos móveis, independentemente da sua localização.

Neste contexto, o termo computação onipresente, *Ubiquitous Computing*, cunhado por (WISER, 1991), se refere a ambientes com objetos operados por computador e conectados em redes sem fio, ou seja, mobilidade, comunicação e poder de processamento integrado em vários objetos com finalidades diferentes.

No decorrer deste capítulo são estudados a computação móvel, suas aplicabilidades, vantagens, restrições, arquitetura e modelos. Aspectos como *handoff*, desconexão, mobilidade e adaptação fazem parte do ambiente móvel e também são abordados.

A computação móvel é um tipo de computação distribuída (TEWARI & GRILLO, 1995), onde a visão de diversos computadores autônomos é transparente para os usuários, a distribuição das atividades é feita automaticamente pelo sistema. Na computação distribuída existe a possibilidade de distribuição do processamento entre computadores diferentes. Mais do que a simples subdivisão de tarefas, este paradigma permite a repartição e a especialização das tarefas computacionais, conforme a natureza da função de cada computador.

O modelo de computação móvel é baseado em computadores fixos e computadores móveis conectados por uma rede, permitindo mobilidade. Usuários podem se locomover levando consigo seu trabalho, realizando todas as operações necessárias em qualquer lugar e a qualquer momento, graças à possibilidade de mudanças em sua localização.

Para tanto, em ambientes distribuídos é necessário um sistema de gerenciamento que esteja apto a recuperar dados, realizar uma transação e contornar falhas de comunicação. Sistemas como estes são denominados SGBD-D (Sistema de Gerenciamento de Banco de Dados Distribuídos).

2 Arquitetura da Computação Móvel

A computação móvel consiste em um sistema composto por dois tipos distintos de entidades, denominadas computadores móveis e computadores fixos (DUNHAM & HELAL, 1995), relacionados a seguir:

2.1 Computadores Móveis

As unidades ou *hosts* móveis são computadores portáteis, interligados em rede através de ligações sem fio, que permitem a mobilidade estabelecendo uma conexão virtual de qualquer localização dentro de uma área geográfica, como mostra a figura 1.

Computadores portáteis (tabela 1) são dependentes de bateria, possuindo ainda algumas restrições físicas, como poder de armazenamento e processamento mais baixos se comparados aos computadores pessoais.

2.2 Computadores Fixos

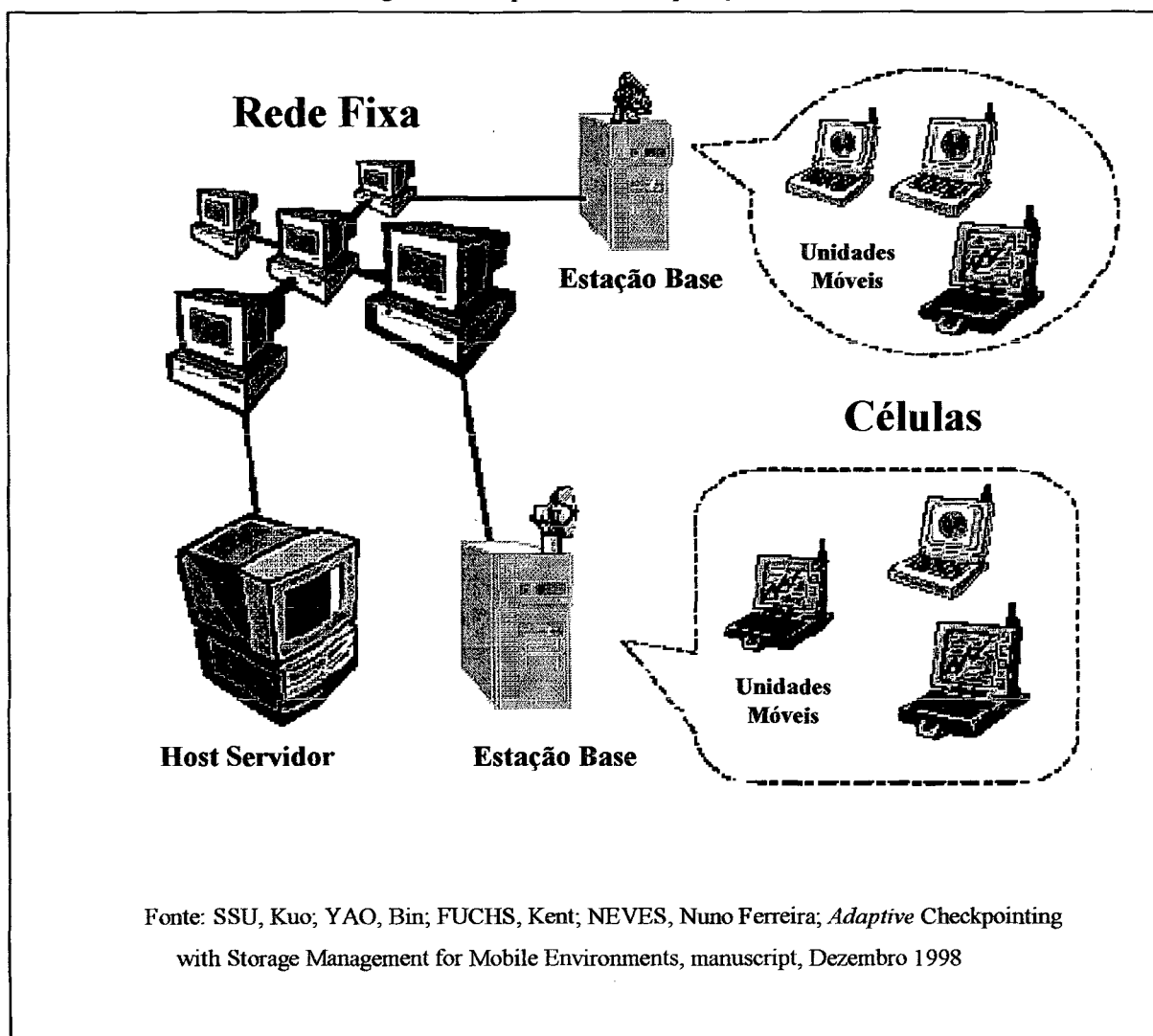
Os computadores ou estações fixos fazem parte de uma rede fixa cuja localização e conectividade não mudam. São utilizados para fornecer vários tipos de informações para os usuários. As mensagens destinadas às unidades móveis são enviadas primeiro para a estação fixa, que mantém a localização da unidade móvel e envia as mensagens através das estações base.

As estações base também conhecidas como *mobile support station* (estações de suporte móveis) realizam a comunicação entre a rede fixa e computadores móveis. São identificadas como servidores e estão localizadas próximas ou em uma torre de antena, que transmitem ou recebem sinais eletromagnéticos dos dispositivos numa área específica. A altura, o *design* e o tamanho da antena são fatores que determinam o alcance do sinal que será transmitido. As estações base são conectadas via linhas fixas para os nós da rede sem fio.

Para gerenciar a mobilidade das unidades móveis o domínio geográfico é dividido em pequenos subdomínios que são cobertos por *interface* sem fio, denominados células. Cada célula é gerenciada por uma estação base, com transmissores e receptores para responder ao processamento de informações necessárias dos clientes localizados na célula. Caso os computadores móveis estejam próximos, a comunicação poderá ser realizada sem a intervenção de uma estação de suporte móvel. As células podem variar de tamanho, cobrindo desde um prédio até grandes áreas geográficas.

O alcance de transmissão de um sinal de uma microonda de rádio é finito e é dependente do poder da estação base e da altura da antena. O caminho a ser percorrido pelo sinal e as obstruções afetam o alcance. Repetidores podem ser utilizados para estender as transmissões a grandes distâncias.

Figura 1- Arquitetura Computação Móvel



Existem dois tipos de ambientes móveis para redes sem fio: o ambiente local e o ambiente global. (SSU et al, 1998). Em um ambiente local, as estações base pertencem às pessoas que utilizam computadores portáteis. Os usuários têm livre acesso e armazenam os dados na estação base. No ambiente global, as estações móveis pertencem às empresas de telecomunicação e os usuários podem alugá-las por um determinado tempo. A empresa pode ou não permitir que os usuários controlem o armazenamento na estação base.

2.3 Evolução da Computação Móvel

A evolução da computação móvel pode ser caracterizada a partir da evolução tecnológica que abrange desde 1820 ao ano de 2000 (MATEUS & LOUREIRO, 1998).

▲ **1820:** Hans Christian Oersted descobre que a corrente elétrica produz um campo magnético; André Marie Ampère quantifica essa observação na Lei de Ampère.

▲ **1830:** Joseph Henry descobre que a variação do campo magnético induz uma corrente elétrica, mas não publica o resultado.

▲ **1864:** James Clark Maxwell modifica a Lei de Ampère, amplia a Lei de Faraday e desenvolve as quatro famosas equações de Maxwell sobre campos magnéticos.

▲ **1876:** Alexander Graham Bell obtém êxito na transmissão de voz através de sinais elétricos. Sua invenção foi chamada inicialmente de *speaking telegraph* (telegrafo falante) e foi a base dos primeiros sistemas telefônicos.

▲ **1887:** Heinrich Rudolph Hertz detecta as ondas eletromagnéticas revistas pelas equações de Maxwell.

▲ **1896:** Guglielmo Marconi inventa o primeiro receptor sem fio prático: o telégrafo sem fio.

▲ **1907:** Início do serviço de radiodifusão comercial transatlântico (estações terrestres imensas: antenas de 30 \Theta 100 m).

▲ **1914:** Início da Primeira Guerra Mundial. Rápido desenvolvimento das comunicações e sua interceptação.

▲ **1921:** A radiodifusão comercial entra em operação nos Estados Unidos.

▲ **1928:** A Polícia de Detroit introduz um sistema de acionamento de carros baseado em radiodifusão (unidirecional) na faixa de 2 MHz.

▲ **1939:** A pesquisa e uso da comunicação via rádio se expande imensamente durante a segunda guerra mundial.

▲ **1945:** AT&T *Bell Labs* inicia experimentos no uso de frequências mais altas com o objetivo de melhorar os serviços móveis.

▲ **1947:** AT&T lança o IMTS (*Improved Mobile Telephone Service*), um sistema de transmissão onde apenas uma torre de alta potência atendia uma grande área ou cidade. Em seguida, AT&T *Bell Labs* propõe o conceito celular.

▲ **Anos 50:** Os sistemas requerem uma elevada largura de banda para transmissão, uma faixa de 120 KHz para transmitir um circuito de voz de apenas 3 KHz. Esta faixa é reduzida pela metade. Com os transistores o tamanho dos equipamentos é reduzido e estes são transportáveis. Os primeiros sistemas de *paging* começam a surgir.

▲ **Anos 60:** Um novo receptor de FM permite reduzir a banda para 30 KHz, abrindo espaço para um maior número de canais de comunicação com o mesmo espectro. *Bell Labs* testa as técnicas de comunicação celular; surgem os primeiros aparelhos portáteis;

▲ **Anos 70:** A AT&T lança o sistema celular conhecido por AMPS (*Advanced Mobile Phone System*). Era considerado um serviço de luxo, destinado para uso em automóveis e de aplicação restrita devido à baixa durabilidade das baterias. Atendiam a uma capacidade limitada de tráfego e a um número reduzido de usuários. A primeira rede celular no mundo foi lançada no Japão, em 1979.

▲ **Anos 80:** O sistema AMPS evoluiu para os padrões atuais. A primeira rede celular americana foi lançada em 1983, em Chicago e Baltimore. Outros sistemas similares entraram em operação no mundo: TACS (*Total Access Communications System*) no Reino Unido (1985), NMT (*Nordic Mobile Telephone Service*) na Escandinávia (1981), NAMTS (*Nippon Advanced Mobile Telephone System*) no Japão. O AMPS, ainda em uso nos EUA, Brasil e grande parte do mundo, é considerado um sistema de primeira geração. A transmissão em FM, reduzida a 25 KHz nos anos 70, entra nos anos 90 na faixa de 10KHz. Também surgem os sistemas de transmissão digital. Pelas técnicas de processamento digital de sinais foi possível reduzir a banda necessária, viabilizando os sistemas móveis digitais.

▲ **1991:** É implantado o primeiro sistema celular de telefonia (AMPS) no Brasil, na cidade do Rio de Janeiro, escolhido pelo Ministério das Telecomunicações como padrão. Validação inicial dos padrões TDMA e CDMA nos EUA. Introdução da tecnologia microcelular.

▲ **1992:** Introdução do sistema celular Pan-Europeu GSM (*Groupe Speciale Mobile*).

▲ **1994:** Introdução do sistema CDPD (*Cellular Digital Packet Data*). Início dos serviços PCS (*Personal Communication Services*) CDMA e TDMA.

▲ **1995:** Início dos projetos para cobertura terrestre de satélites de baixa órbita, como o projeto *Iridium*.

▲ **2000:** Estima-se um número de 100 milhões de assinantes do sistema móvel celular no mundo inteiro.

2.4 Utilização Profissional

A computação móvel tem inúmeras aplicações profissionais, alguns exemplos de serviços são enumerados os seguintes serviços:

- ▲ Escritório portátil e automação para vendas: automação da área de vendas de uma empresa, provendo dispositivos móveis a vendedores de campo, empresários, palestrantes, representantes comerciais e engenheiros de campo;
- ▲ Monitoramento de construções: engenheiros acompanham obras através do computador portátil;
- ▲ Coleta de dados por pesquisadores que viajam pelo mundo;
- ▲ Nos serviços de entrega local de produtos para controle e acompanhamento;
- ▲ Aplicações comerciais: uso de *laptops*, *handhelds* e *PDA*s (*Personal Digital Assistants*) por funcionários de empresas para serviços de atualização e consulta de dados;

- ✧ Aplicações em indústrias de linhas aéreas e estradas de ferro: dispositivos especiais são utilizados para vendas de passagens aéreas, programação de vôos e envio de bagagens, o que reduz as demoras e melhora serviços dos clientes;
- ✧ Comunicação Eletrônica via *Paging*: uma das primeiras aplicações de redes sem fio. *Pagers* podem ler mensagens e números de telefones, *voice pagers*, etc. A comunicação pode ser feita através de um servidor de *paging* ou como um adicional ao *e-mail*.
- ✧ Em serviços de emergência médica para acesso as informações e fornecimento de dados. Médicos e enfermeiros equipados com *notebooks* podem fazer consultas aos bancos de dados para obter informação sobre pacientes, medicamentos, etc.

2.5 Aplicações

A computação móvel tem várias aplicabilidades distintas que estão sendo cada vez mais discutidas, implementadas e aprimoradas. São elas:

- ✧ Telefones celulares;
- ✧ Rádios bidirecionais;
- ✧ Correio eletrônico;
- ✧ Páginas amarelas, contendo informações sobre empresas, produtos, etc;
- ✧ Serviços de informações de tráfego, clima e temperatura;
- ✧ Aplicações de entretenimento: horóscopo, filmes, restaurantes, hotéis, resultados de jogos, trânsito, previsão do tempo, cotações da bolsa de valores, agenda, etc.
- ✧ Aplicações para serviços de viagens: acesso às agências de turismo, informações de vôos *on-line*, confirmação de reservas, cancelamento de vôos, preços, notificação de acidentes, etc..

2.6 Vantagens da Computação Móvel

Podem ser citadas diversas vantagens na utilização de um sistema móvel tais como:

- ⤴ Conforto, pois é possível a utilização do computador em qualquer ambiente;
- ⤴ Flexibilidade, para utilização em diversas aplicações que exijam movimento;
- ⤴ Disponibilidade, independente da localização do usuário;
- ⤴ Portabilidade, facilitando o transporte;
- ⤴ Autonomia de energia, para garantir o funcionamento onde não existe disponibilidade da mesma;

2.7 Restrições

Apesar do grande avanço relacionado à computação móvel, os ambientes móveis apresentam algumas restrições. Como unidade móveis mudam de lugar constantemente, podem encontrar ambientes de rede com interfaces diferentes, o que exige o uso de protocolos de acesso diferentes, causando problemas de interoperabilidade entre as unidades móveis.

Limites físicos de *hardware*, como características de peso, tamanho e ergonomia ocasionam certa desvantagem em termos computacionais referentes à velocidade do processador, tamanho de memória e capacidade de disco, limitando também o poder de processamento e dispositivos. De forma mais específica, a seguir são enumeradas algumas restrições relacionadas à computação móvel:

- ⤴ Restrições econômicas e de segurança;
- ⤴ Computadores portáteis estão mais vulneráveis a roubos, perdas e estragos;
- ⤴ A conexão móvel é altamente variável em desempenho e confiabilidade. Existe a perda temporária de comunicação quando deslocamento entre áreas mantidas por diferentes estações base e renegociação de características de acesso;

- ⤴ O tempo de vida das baterias é relativamente curto. As unidades móveis possuem baterias de duração restrita, demandando estratégias de economia de energia;
- ⤴ A diversidade arquitetural tem grande variabilidade das configurações de *hardware* das unidades móveis;
- ⤴ Devido à localização dos computadores móveis são detectados problemas no processamento de consultas, pois existe a dificuldade de encontrar uma localização ótima para a realização das mesmas;
- ⤴ A mobilidade causa variação na qualidade do canal (largura de banda e aumento da taxa de bits errados, entre outros) e na maioria das vezes resulta em retransmissões e queda do desempenho do enlace de comunicação sem fio, com altas taxas de erros;





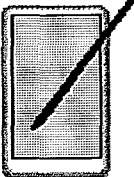

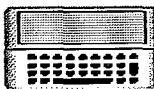
2.8 Tipos de Aparelhos Móveis

Existem vários aparelhos que podem ser utilizados num ambiente de computação móvel. A escolha do equipamento adequado depende de fatores como tipo de atividade, modelo de captura e apresentação das informações e volumes de dados esperados.

Para equipes de profissionais móveis que consomem grande parte do seu tempo trabalhando remotamente, estes equipamentos são versáteis, dedicados, multifuncionais e de uso genérico. Do ponto de vista empresarial, eles são ótimos geradores de informações, podendo ser utilizados desde a automação de processos até a coleta de informações estratégicas.

A tabela 1 apresenta os aparelhos portáteis mais utilizados atualmente (REVISTA INFO EXAME, 1999):

Tabela 1 - Tipos de Aparelhos Móveis

	Smartcard (Cartões inteligentes): Cartões com <i>chips</i> com capacidade de armazenamento e processamento. Através de equipamentos de leitura eletrônica, podem realizar pagamentos, ligações através de telefones públicos, etc.
	Telefone Celular: Além da tradicional comunicação pelo telefone, o celular permite navegação através de um <i>microbrowser</i> pela Internet, troca de <i>e-mails</i> , acesso a serviços bancários, efetuação de pagamentos, acesso a informações sobre previsão do tempo, índices financeiros, etc.
	Pager: Utilizados como agenda telefônica, para leitura de <i>e-mails</i> , para receber mensagens pelo telefone, ondas de rádio ou de <i>sites</i> ; receber informações como previsão do tempo ou notícias sobre o trânsito, etc.
	Relógio Multifuncional: Além das funções de um relógio normal, dependendo do modelo escolhido, tem a função de agenda pessoal; capta sinais de satélite e determina as coordenadas de latitude e longitude; mede a pressão sanguínea e a frequência cardíaca, possui indicador eletrônico de profundidade entre outras utilidades.
	Palmtop: Permite o controle de informações pessoais, como compromissos, tarefas e contatos. Permite acesso a Web para consulta e troca de mensagens. Tem grande utilização na automação da força de vendas e serviços bancários. Possui um <i>scanner</i> de código de barras para gerenciamento de estoques, controle de inventário ou monitoramento de linha de montagem. Não possui teclado, a entrada de dados é feita por meio de reconhecimento de escrita manual.
	Notebook: É um computador portátil com as mesmas funções de um desktop. Tem maior poder de processamento e armazenamento que o <i>palmtop</i> e o <i>handheld</i> . É também mais caro. Possui teclado, <i>mouse</i> , <i>drive</i> para disquete, <i>cd-rom</i> , DVD, <i>modem</i> , etc.
	Handheld: São <i>notebooks</i> menos potentes, mais baratos e com poder de baterias mais longo. Suportam sistemas operacionais mais leves, como o <i>Windows CE</i> . São utilizados principalmente por empresas na área de vendas.

Modelos de Computação Móvel

Ambientes móveis apresentam alguns tipos de restrições e os modelos utilizados em um ambiente móvel devem levar em consideração essas restrições, adequando-se a estas limitações. (PITOURA & SAMARAS, 1998) classificam alguns modelos de computação móvel. São eles: modelo cliente/servidor, modelo *peer-to-peer* e modelo de agentes móveis.

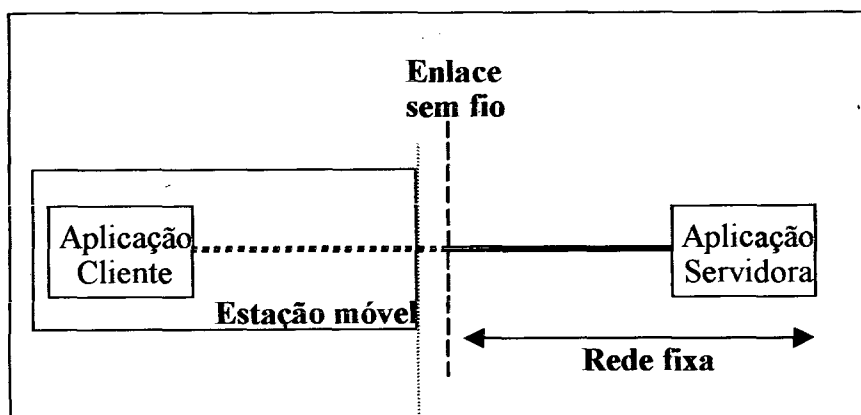
2.8.1 Modelo Cliente/Servidor

Neste modelo, o cliente que é a unidade móvel, se comunica diretamente com o servidor localizado na rede fixa, requisitando serviços, como mostra a figura 2. Em alguns casos, a funcionalidade e os dados são distribuídos entre diversos servidores fixos, que podem ter que se comunicar entre si para atender a requisição de um cliente.

Na sua forma padrão, o modelo Cliente/Servidor apresenta alguns problemas quando utilizado no ambiente móvel, pois a comunicação é feita em baixas velocidades e não é confiável. Outro fator a ser considerado é a desconexão. Todavia, este modelo precisa ser estendido para tratar desconexões, comunicação não estável e não confiável entre clientes e servidores. Em caso de desconexão a unidade móvel deverá emular a função do servidor para que o processamento das operações se concretize.

Duas extensões do modelo cliente/servidor são discutidas a seguir: modelo Cliente/Agente/Servidor e modelo Cliente/Interceptor/Servidor.

Figura 2 - Modelo Cliente/Servidor



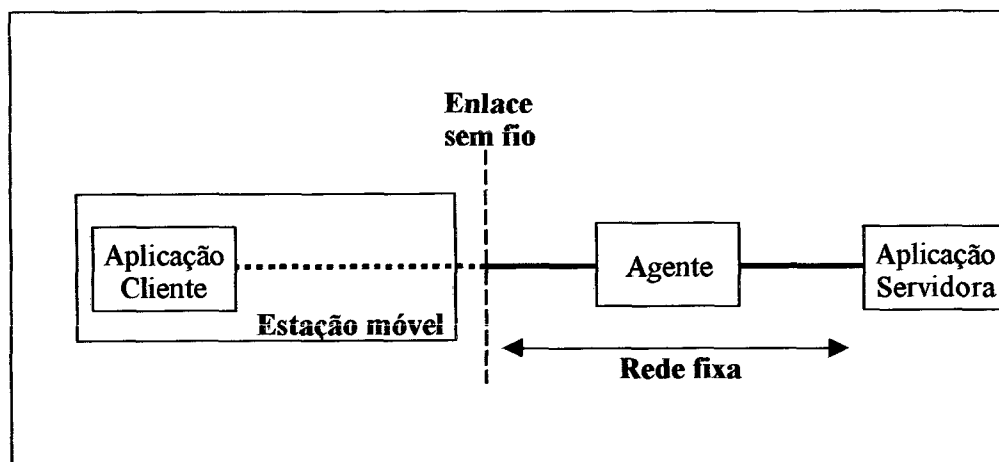
2.8.2 Modelo Cliente/Agente/Servidor

O modelo cliente/agente/servidor, apresentado na figura 3, constitui uma das melhores abordagens para o desenvolvimento de aplicações para computação móvel, pois suporta a desconexão. Quando uma conexão é realizada, tanto o servidor remoto quanto o cliente, executam as atualizações a serem feitas.

Agentes foram acrescentados ao modelo cliente/servidor, sendo que estes podem executar tanto tarefas de comunicação como de aplicação, podendo residir na rede fixa ou na unidade móvel. O agente é um *software* que executa tarefas, processa consultas e retorna a resposta em nome de uma aplicação cliente. Atua como um intermediário, provendo facilidades para armazenar as mensagens que serão trocadas com a unidade móvel.

No contexto de gerenciamento de redes, ele reúne informações sobre os dispositivos da rede e executa comandos em resposta a uma requisição do gerenciador.

Figura 3 - Modelo Cliente/Agente/Servidor



2.8.3 Cliente/Interceptador/Servidor

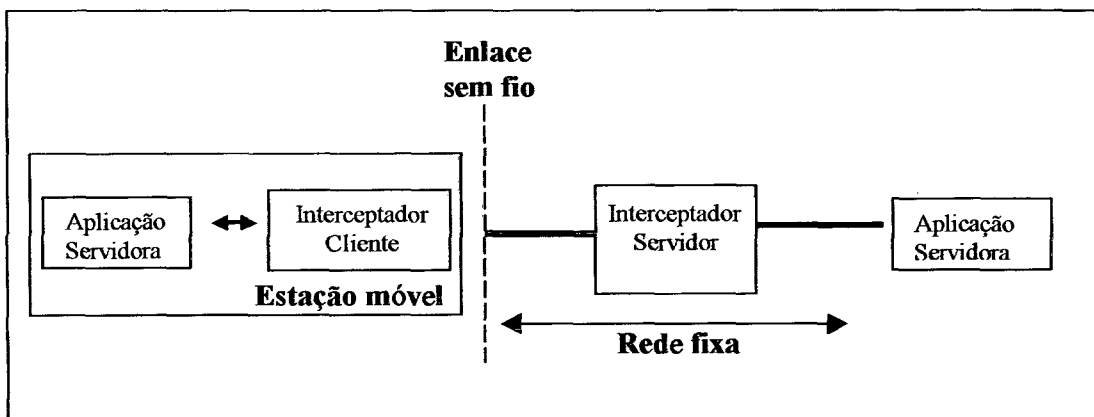
O modelo cliente/interceptador/servidor, divide o agente em duas partes, uma que fica no servidor da rede fixa e outra que fica na unidade móvel. O modelo é transparente

tanto para o cliente como para o servidor, oferecendo flexibilidade e poder de gerenciamento de desconexões.

Este modelo é indicado quando a aplicação exige alto poder de processamento e armazenamento, além de maior autonomia de energia. Melhora a compressão de dados, otimiza a comunicação sem fio e tratamento de desconexões devido à existência dos dois agentes interceptores.

Um dos problemas deste modelo é que cada aplicação requer processamento tanto do cliente como do servidor.

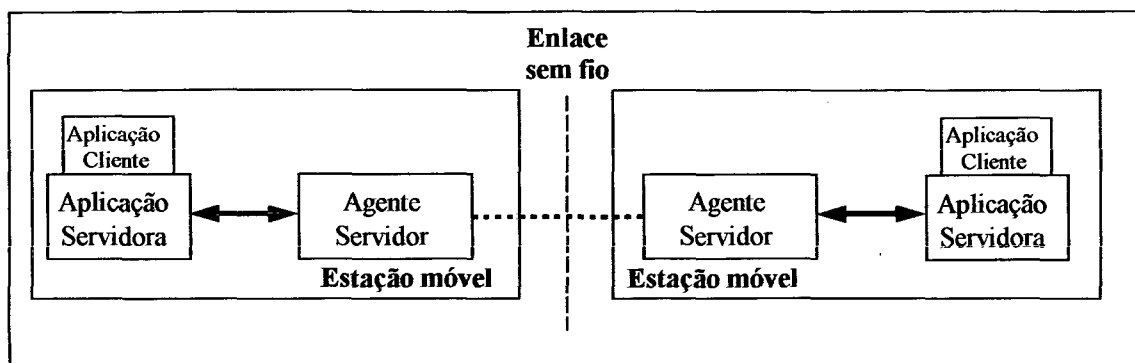
Figura 4 - Modelo Cliente/Interceptador/Servidor



2.8.4 Modelo Peer-to-Peer (P2P)

No modelo *peer-to-peer*, como mostra a figura 5, não existe distinção entre cliente e servidor. Estações móveis podem assumir a função de clientes e a comunicação pode ser realizada com a estação base que estiver disponível.

Este modelo é mais adequado quando há uma forte conexão entre computadores portáteis e pouca ou nenhuma conexão com o servidor, fazendo com que a comunicação seja realizada com mais facilidade e se torne mais acessível do que a realizada com o servidor. Contudo, os custos de comunicação podem ser elevados no caso das unidades móveis estarem fisicamente longe do servidor ou não existir um canal de comunicação entre os mesmos.

Figura 5 - Modelo *Peer-to-Peer*

2.8.5 Modelo de Agentes Móveis

O modelo de agentes móveis trata a questão da mobilidade com naturalidade, permitindo o trabalho desconectado. Pode ser combinado com outros modelos, porém, possui algumas restrições econômicas e de segurança, estudada no capítulo 5, subseção 5.7.

Agentes móveis são programas que podem ser disparados de um computador (cliente) a fim de serem executados remotamente em outro (servidor). Possuem instruções, dados e um estado de execução. Além disso, esses elementos possuem características inerentes ao conceito de multi-agentes, que proporcionam um bom desempenho em sistemas de objetos distribuídos. Desta forma, características como cooperação, autonomia, representatividade, entre outras, foram acopladas neste modelo com a finalidade de suprir as necessidades exigidas para o bom funcionamento de modelos que utilizam esse paradigma. A seguir são descritas algumas características do modelo de agentes móveis (PITOURA & SAMARAS, 1998):

- ▲ **Objetos passantes:** quando um agente móvel é transferido, todo o objeto é movido, ou seja, o código, os dados, o itinerário para chegar ao servidor necessário, o estado de execução, etc.
- ▲ **Assincronismo:** o agente móvel possui seu próprio *thread* de execução e este não precisa ser executado de modo síncrono.

- ^ **Interação local:** o agente móvel interage com outros agentes móveis ou objetos estacionários locais. Se necessário, um agente mensageiro é despachado para facilitar a interação com agentes remotos.
- ^ **Desconexão:** o agente móvel pode executar tarefas mesmo em modo desconectado. Quando se faz necessária transferência de agentes, aguarda-se até que a conexão seja restabelecida.
- ^ **Execução paralela:** múltiplos agentes podem ser disparados para diferentes servidores com o objetivo de executar tarefas em paralelo.

2.9 Aspectos Relacionados à Computação Móvel

Alguns aspectos fundamentais relacionados ao ambiente móvel são abordados a seguir, quais sejam:

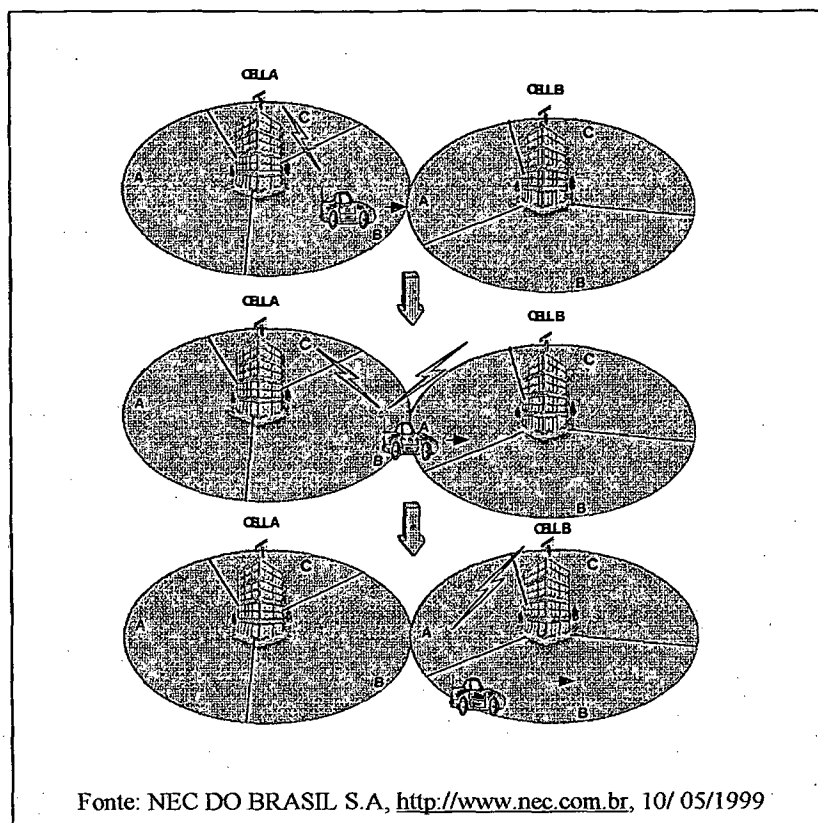
- ^ *Handoff*;
- ^ Mobilidade;
- ^ Desconexão;
- ^ Adaptação.

2.9.1 *Handoff*

Quando os computadores se movimentam dentro de uma célula de cobertura ou entre células vizinhas durante a execução de aplicações, tem-se um movimento que gera um processo chamado *handoff* (figura 6).

Handoff é um processo entre duas estações base adjacentes que garante uma transação enquanto usuários se movimentam de uma célula para outra, mantendo a conectividade de uma unidade móvel com a rede fixa (PITOURA & SAMARAS, 1998). Isto implica que uma estação base pode detectar qualquer entrada e qualquer saída de usuários dentro e fora da sua célula. Durante o processo de *handoff*, as estações base são responsáveis pela comunicação sem fio das unidades móveis, sendo que este processo é realizado de forma transparente ao usuário.

Figura 6 - Processo de Handoff



2.9.2 Mobilidade

Um dos principais objetivos da computação móvel é a mobilidade dos computadores portáteis. Devido às diferenças estruturais de um sistema móvel, assim como as variações de tráfego, o ambiente de operação do usuário passa a ser altamente dinâmico.

A mobilidade introduz problemas e desafios não existentes em ambientes fixos. Vários problemas relativamente bem resolvidos na computação tradicional, permanecem praticamente em aberto em ambientes móveis. Os principais problemas que a mobilidade apresenta vão desde a velocidade do canal, passando por interferências do ambiente e localização da unidade móvel, até a duração da bateria desta unidade. Alguns destes problemas são abordados a seguir (SILVA & ENDLER, 2000):

- ⤴ Devido a ausência de uma topologia fixa da rede com *hosts* móveis, algoritmos distribuídos tradicionais precisam ser reprojatados;
- ⤴ A distribuição de carga em uma rede com elementos móveis pode variar de uma maneira muito mais rápida em redes fixas;
- ⤴ O custo para localizar um elemento móvel contribui de forma significativa para o custo de cada comunicação;
- ⤴ Manter bases de informação sobre a localização de clientes móveis é uma tarefa complexa e requer estruturas de dados e algoritmos eficientes para lidar com a rápida variação desta informação;
- ⤴ A conceitividade de um elemento móvel pode variar muito no que diz respeito ao desempenho (largura de banda e latência da comunicação) e a confiabilidade;
- ⤴ Conjunto de serviços disponíveis para um elemento móvel pode variar de um local para outro.

A mobilidade envolve duas entidades: computadores móveis e computadores fixos. Na rede fixa, unidades móveis podem estabelecer uma conexão de diferentes localizações. A ligação sem fio permite a mobilidade e estabelece uma conexão virtual de qualquer localização dentro de uma célula.

A configuração de um sistema distribuído que inclui elementos móveis é dinâmica. Unidades móveis se comunicam diretamente com as estações base, se movendo dentro e fora das células.

2.9.3 Desconexão

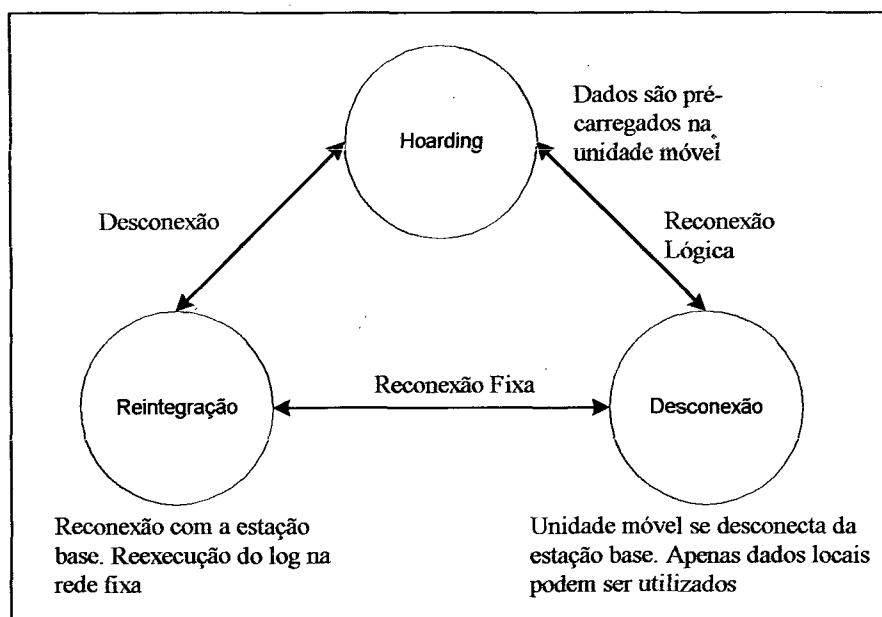
Devido à mobilidade a qualidade do canal de comunicação tende a variar consideravelmente. Como consequência, protocolos de comunicação, sistemas operacionais e aplicativos devem estar preparados para lidar com flutuações na qualidade da conexão. Limitações impostas pelos canais de comunicação devem ser reconhecidas e a funcionalidade da conexão restringida de acordo. Caso contrário, o usuário pode ter sua conexão bloqueada por intervalos prolongados, pois o sistema

operacional ou o sistema de informação insiste em prosseguir com tarefas cujos requisitos de qualidade da conexão são incompatíveis com a condição do canal de comunicação naquele momento.

A desconexão ocorre quando uma unidade móvel se movimenta fora da área de cobertura de todas as células. Pode ser categorizada de duas formas: voluntária e involuntária. A desconexão involuntária ou forçada acontece quando uma unidade móvel entra em uma região onde não há cobertura da rede o qual está registrada. A desconexão voluntária não é tratada como uma falha, pois devido à mobilidade e o poder das baterias, as unidades móveis sofrem de freqüentes desconexões (PITOURA & SAMARAS, 1998). Neste contexto a desconexão é considerada uma falha planejada que pode ser antecipada e preparada. Quando este planejamento acontece, dados são movidos para a unidade móvel para que ela possa ter autonomia de operação durante o período que estiver desconectada. A ação de pré-carregar os dados antecipadamente é chamada de *hoarding* (armazenamento).

A operação de desconexão pode ser descrita como a transição entre três estados (PITOURA & SAMARAS, 1998). São eles: Reintegração, *hoarding* e desconexão como mostra a figura 7.

Figura 7 - Transição de Estados



Antes da desconexão, a unidade móvel entra em um estado de *hoarding*. Neste estado os dados utilizados durante a operação serão pré-carregados para a unidade móvel, tornando-se inacessíveis para outros *sites*. Os dados que serão transferidos para a unidade móvel podem ser determinados pelo usuário, ou de acordo com o passado histórico de acesso aos dados e a necessidade futura que eles terão.

No momento em que a unidade móvel se desliga da estação base ela entra em estado de desconexão. Neste estado, apenas dados locais podem ser utilizados. Caso alguma solicitação seja feita e os dados não estejam disponíveis localmente, uma mensagem de erro é retornada. Estas solicitações podem ser inseridas em uma fila e serem atendidas na reconexão.

A partir do momento em que a unidade móvel se reconecta com a estação base ela entra no estado de reintegração. Neste estado, atualizações efetuadas na unidade móvel são reintegradas com as de outros *sites*. Geralmente esta operação é realizada pela re-execução do *log*¹ na rede fixa.

2.9.3.1 Modos de Operação de uma Unidade Móvel

Na computação tradicional existem dois modos de operação de uma estação, conectada a rede ou completamente desconectada. Uma estação opera em um dos dois modos, respeitando sua conexão com o resto da rede.

Existem vários modos de operação de uma unidade móvel. O grau de conexão é realizado de acordo com a largura de banda disponível. Quando desconexões parciais ou totais acontecem, não devem ser tratadas como falhas. Uma unidade móvel deve ser capaz de operar quando não existe conexão ou quando existe conexão baixa com a rede fixa.

Uma unidade móvel opera em um modo parcialmente desconetado quando a comunicação através da rede sem fio é baixa, podendo utilizar um modo *Doze* (adormecido) para preservar energia.

¹ O termo *log* pode ser traduzido como diário, ou seja, um repositório de anotações sobre atividades, no caso, do banco de dados.

Com relação à forma de como é feita a transição de um modo para outro, existem vários protocolos que podem ser designados para preparar o sistema para a transição entre os modos de operação (conectado, desconectado ou parcialmente desconectado) de uma unidade móvel (PITOURA & BHARGAVA, 1994). São eles (figura - 8):

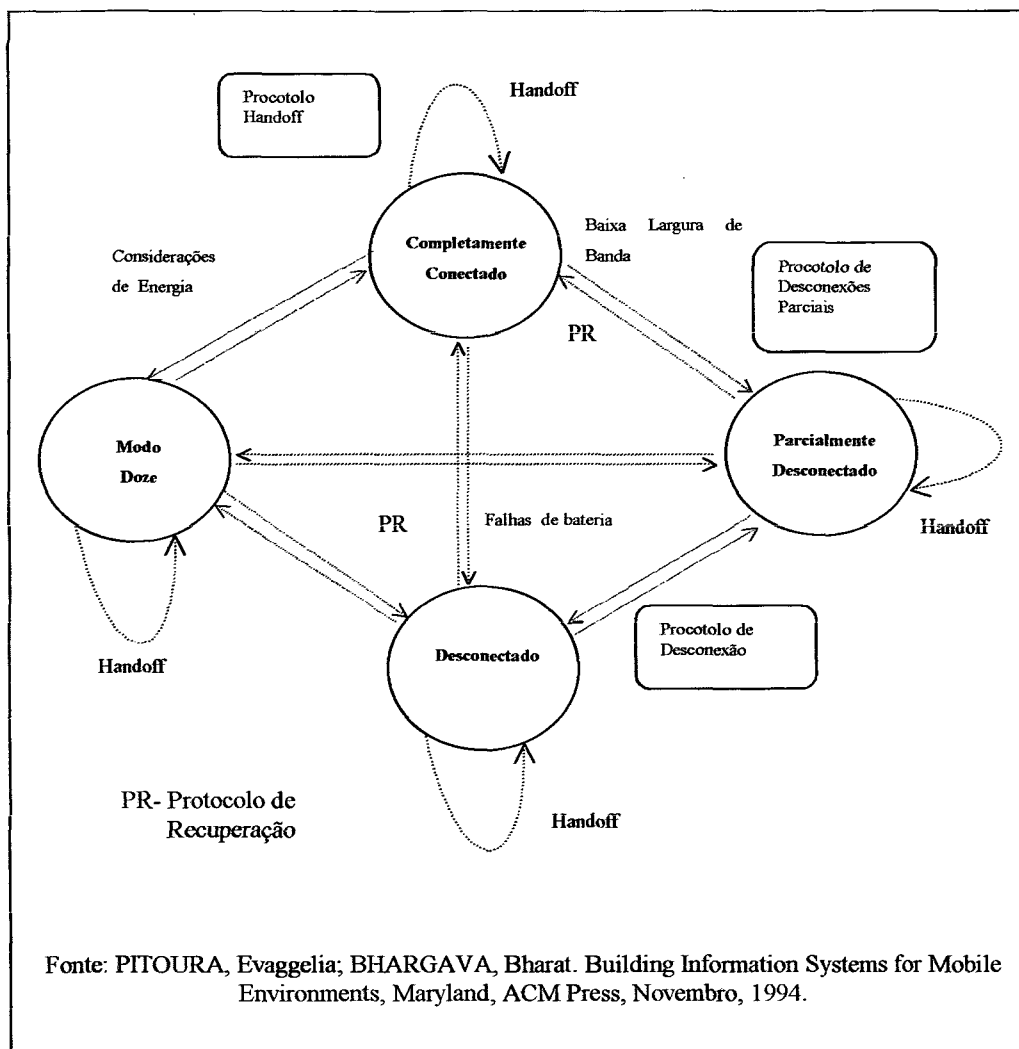
⤴ **Protocolo de desconexão:** é executado antes da unidade móvel ser desligada fisicamente da rede fixa. O protocolo deve garantir informações suficientes para que a unidade móvel opere de forma autônoma durante a desconexão.

⤴ **Protocolo de desconexão parcial:** prepara a unidade móvel para operar em um modo onde toda a comunicação com a rede fixa deve ser tão restrita quanto possível. O armazenamento seletivo pode ser usado para guardar dados cuja presença na unidade deverá minimizar o futuro uso da rede.

⤴ **Protocolo de recuperação:** restabelece a conexão com a rede fixa e prossegue com a operação normal.

⤴ **Protocolos *Handoff*:** refere-se a passagem de uma unidade móvel de uma célula para outra. As informações pertencentes a cada unidade móvel devem ser transferidas para a estação base da nova célula.

Figura 8 - Modos de Operação de uma Unidade Móvel

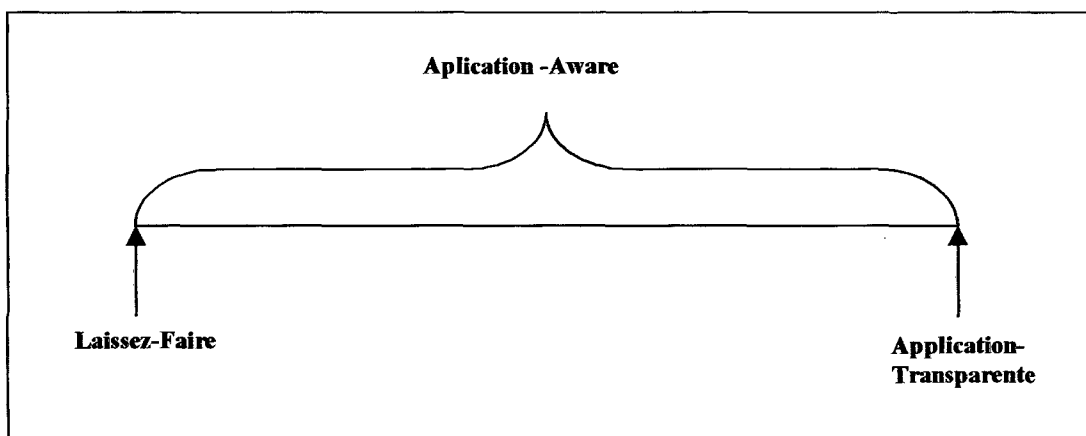


2.9.4 Adaptação

Conforme visto na seção 2.10.2, um ambiente móvel, a mobilidade traz uma série de problemas. No campo das aplicações, muito se tem discutido a respeito daquelas associadas às unidades móveis e de mecanismos que permitam a elas se adaptarem às alterações do ambiente, dividindo a responsabilidade entre cliente e servidor. Para funcionar com sucesso unidades móveis precisam ser adaptativos (HELAL et. al, 1999).

(STAYANARAYANAN, 1996) define uma faixa de estratégias para a adaptação delimitada por dois extremos, *Laissez-Faire* e *Application-Transparent*, como mostra a figura 9.

Figura 9 - Modos de Adaptação



Na estratégia *Laissez-Faire*, a responsabilidade de adaptação é das aplicações individuais, não havendo nenhum suporte pelo sistema. Isto faz com que não haja um árbitro central para gerenciar incompatibilidades nos recursos provindos de diferentes operações.

Em outro extremo, na estratégia *application-transparent*, a responsabilidade de adaptação é inteiramente do sistema, o qual contém um ponto central para gerenciamento e controle de dados. Desta forma, unidades móveis continuam o processamento mesmo estando em movimento. Apresenta a desvantagem de que em muitas situações a adaptação realizada pelo sistema é inadequada com algumas aplicações.

Entre os dois extremos, a estratégia *application-aware* gerencia a adaptação entre aplicações individuais e o sistema operacional. Neste caso, o sistema operacional gerencia fontes (largura de banda, memória ou bateria), notifica aplicações de trocas ambientais relevantes, entre outras tarefas.

CAPÍTULO III

COMPUTAÇÃO SEM FIO

Em 1946, a *Illinois Bell Telephone Company* introduziu um serviço de telefonia móvel, que permitia aos usuários na direção de veículos comunicar-se com o sistema telefônico. Foi a primeira iniciativa de permitir comunicação bidirecional sem fio (COMPUTAÇÃO MÓVEL, 2000). Atualmente temos a popularização do telefone celular, que garante a mobilidade dos usuários do sistema telefônico. A tecnologia precursora foi lançada no Japão em meados da década de 70 e 80, com redes celulares de telefonia móvel.

A tecnologia *Wireless computing* se refere a sistemas conectados via ligações sem fio. A comunicação sem fio pode ser feita através de sistema de ondas de rádio, utilizando antenas para transmissão e recepção. Podem utilizar tecnologias como radiofrequência, infravermelho, satélite, microondas, *laser*, *spread spectrum*, entre outras. Três tipos de arquiteturas de redes sem fio serão abordadas neste capítulo:

- ✧ **Arquitetura *ad hoc***: não existe infraestrutura fixa. Todas unidades móveis são capazes de comunicar-se diretamente, não havendo pontos de acesso, ou seja, não há a necessidade de estações base.
- ✧ **Arquitetura celular**: é composta por áreas chamadas de células, que são ligadas à rede fixa através de vários canais sem fios. Consiste de uma rede fixa que conecta servidores com as estações base.

- ⤴ **Arquitetura de rede:** É composta por três tipos de redes. Uma rede local (LAN - *Local Area Network*) sem fio, uma rede metropolitana (MAN - *Metropolitan Area Network*) sem fio ou uma rede geograficamente distribuída (WAN - *Wide Area Network*) que contenha elementos que utilizem conexões sem fio.

Existem várias tecnologias que podem ser utilizadas para o estabelecimento da comunicação sem fio, destacando-se o *Wireless Application Protocol* e o IP móvel.

- ⤴ ***Wireless Application Protocol:*** É um protocolo de aplicações sem fio que provê um padrão universal para trazer o conteúdo da Internet e serviços agregados para dispositivos móveis.
- ⤴ **IP Móvel:** sobre o *Internet Protocol* móvel são estudados o IPV4 e o IPV6.

3 *Wireless Computing* (computação sem fio)

A mobilidade oferecida pelas redes *Wireless* constitui uma das grandes vantagens em sua utilização, visto que muitas empresas fazem modificações constantes em seu ambiente e a não utilização de cabos físicos elimina algumas dificuldades encontradas, colocando estações de trabalho em lugares estratégicos.

A computação móvel e as redes sem fio possuem muitos pontos em comum, mas não são totalmente semelhantes, como mostra a tabela a seguir (TANEMBAUM, 1997):

Tabela 2 - Computação Móvel e Redes Sem Fio

APLICAÇÕES	REDES SEM FIO	COMP. MÓVEL
Estação de trabalho fixa em escritórios	Não	Não
Utilização de um computador portátil na estrada, em hotéis, aviões, etc	Não	Sim
Redes LANs em prédios sem fiação	Sim	Não
Escritório Portátil; Automação em vendas	Sim	Sim

A comunicação sem fio pode ser feita através de sistema de ondas de rádio, utilizando antenas para transmissão e recepção. Pode utilizar tecnologias como

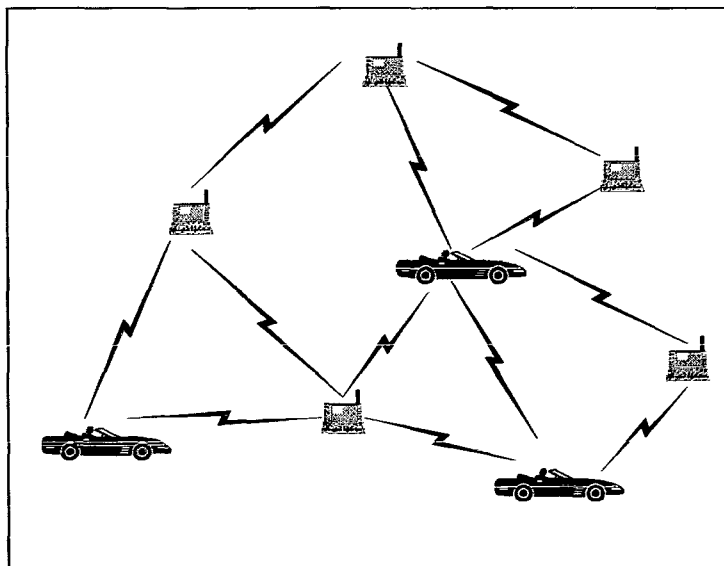
radiofrequência, infravermelho, satélite, microondas, *laser*, *spread spectrum*, entre outras.

Três tipos de arquitetura de redes sem fio são abordadas a seguir. São elas: arquitetura *ad hoc*, arquitetura celular e arquitetura de rede (*wireless LAN*, WAN, MAN).

3.1 Arquitetura *Ad hoc*

Nas redes *ad hoc* não existe infraestrutura fixa. Todas as unidades móveis são capazes de se comunicar diretamente entre si, não havendo pontos de acesso, ou seja, não há a necessidade de estações base. Se uma unidade deseja se comunicar com outra que não está dentro de seu alcance, ela o faz através de outras unidades móveis, que retransmitem os pacotes até que eles alcancem seu destino, independente de onde estão localizados, proporcionando mobilidade, como mostra a figura 10.

Figura 10- Rede Ad-Hoc



As redes *ad hoc* são indicadas para os casos onde a instalação de uma rede fixa é geralmente impossível, como situações trágicas envolvendo terremoto, inundação ou

guerras. Pode ser utilizada também, em reuniões ou aulas interativas, entre outras aplicações.

Os nós de uma rede *ad hoc* se movem de maneira arbitrária, fazendo com que a topologia da rede mude frequentemente e de forma imprevisível. Associado a isto, limitações de largura de banda, taxas de erros nos enlaces sem fio e tempos de vida das baterias dos nós tornam o roteamento, principalmente o multiponto, um desafio. A conectividade entre os nós móveis muda constantemente, requerendo uma permanente adaptação e reconfiguração de rotas.

Apesar destes fatores negativos, as redes *ad hoc* oferecem várias vantagens em sua utilização, tais como:

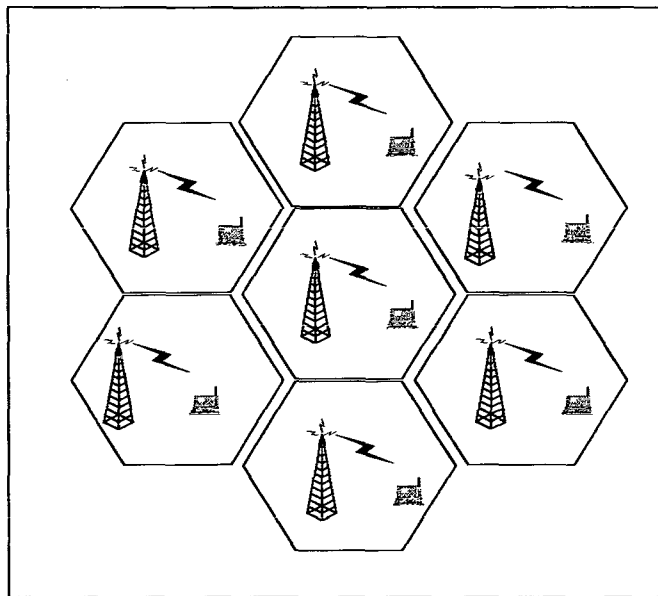
- ⤴ Instalação rápida, pois podem ser estabelecidas dinamicamente em locais onde não haja previamente uma infra-estrutura de rede instalada;
- ⤴ Conectividade, entre dois nós móveis podem se comunicar diretamente, desde de que cada nó esteja dentro da área de alcance do outro;
- ⤴ Tolerância à falhas, devido à facilidade de reconfiguração dinâmica da rede;
- ⤴ Mobilidade, etc.

3.2 Arquitetura Celular

Em uma arquitetura de rede celular, uma área maior é dividida em áreas menores chamadas células, que são ligadas à rede fixa através de vários canais sem fios. Esta arquitetura é composta por terminais móveis, estações base e *switches*.

A área geográfica de cobertura de uma célula é altamente irregular. A figura 11, mostra um modelo ideal para as células, onde as mesmas são divididas em hexágonos (HELAL et. al, 1999), também chamados de “favos de mel”. São gerenciadas por uma estação base que se comunica com a unidade móvel através de alocação de largura de banda de rádio *spectrum*.

Figura 11 - Modelo ideal para Células

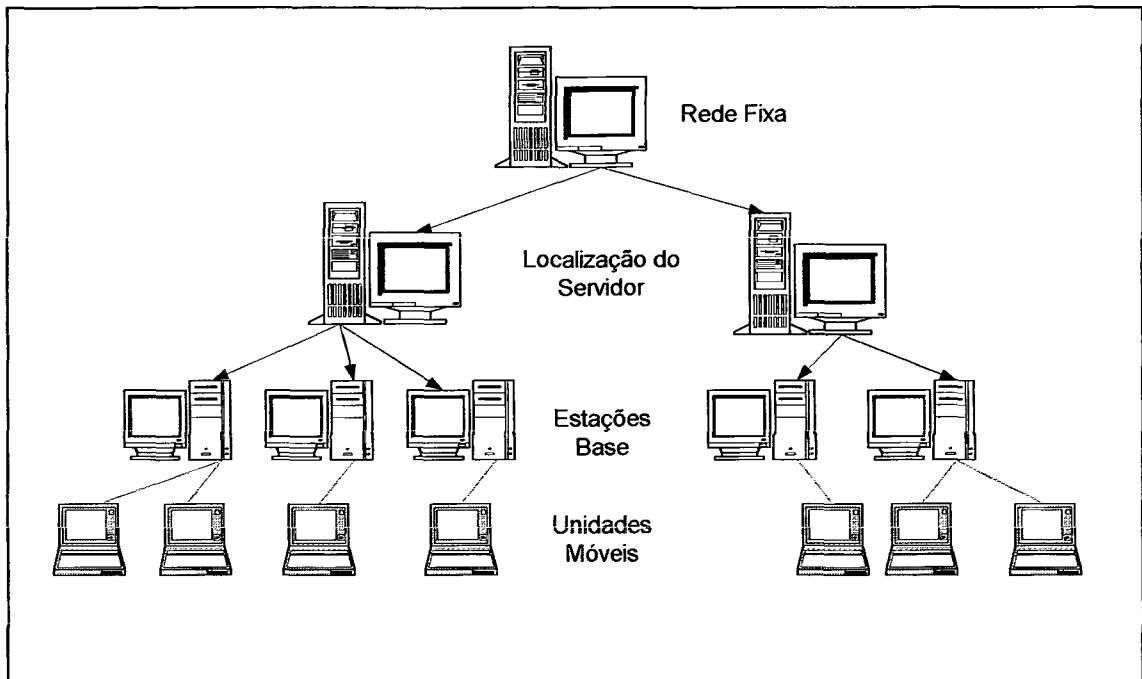


Um conjunto de estações base é controlado por uma estação base controladora denominada BSC (*Base Station Controller*). A função primária da BSC é administrar os recursos de canais de rádio e a execução de *handoffs*. Cada BSC é conectada a um MSC (*Mobile Switching Center*) através da rede fixa. Cada unidade móvel deverá ser permanentemente registrada abaixo do servidor de localização que é chamado de *Home Location Server* HLS, uma unidade móvel pode ser registrada em sua HLR (*Home Location Register*) ou como visitante em uma VLR (*Visitor Location Register*).

A área registrada do usuário móvel é a região na qual ele poderá ser localizado, podendo ser uma única célula ou um grupo de células. Dentro desta área, o caminho percorrido pela unidade móvel é trilhado através de mecanismos citados no capítulo 5.

Os servidores de localização HLS são responsáveis por manter a trilha dos endereços das unidades móveis que estão registradas nele.

Figura 12 - Arquitetura Celular

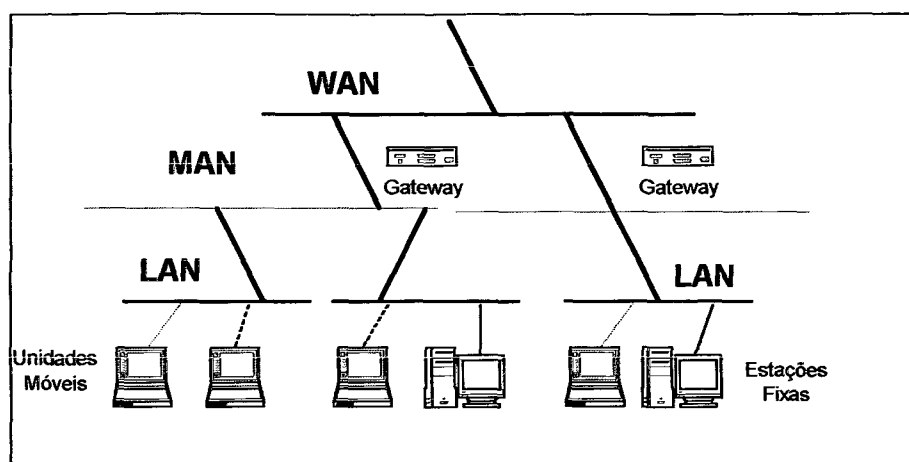


3.3 Arquitetura de Rede

Conforme visto, *Wireless* é o termo utilizado para designar os dispositivos envolvidos em uma rede local (LAN - *Local Area Network*) sem fio, numa rede metropolitana sem fio (MAN - *Metropolitan Area Network*) ou numa rede geograficamente distribuída (WAN - *Wide Area Network*) que contenha elementos que utilizem conexões sem fio (IMIELINSKI & BADRINATH, 1994).

Uma MAN é considerada uma rede metropolitana situada geralmente dentro de uma cidade. As MAN's, por definição, são menores que as redes de longa distância (ou WAN's), porém maiores do que as LANs. Os meios de comunicação geralmente utilizados nas MANs são as linhas telefônicas, os serviços de TV a cabo ou a comunicação sem fio. Devido a maior empregabilidade em ambientes móveis as *Wireless LAN's* e *Wireless WAN's* são abordadas a seguir.

Figura 13 - Arquitetura de Rede



3.3.1 Wireless LANs

A *Wireless LAN* é uma rede local que pode ser utilizada onde haja necessidade de mobilidade dos pontos da rede ou existam dificuldades de cabeamento, como por exemplo, quando a arquitetura de um prédio dificulta (ou impossibilita) a passagem de cabos de rede. Sem o cabeamento físico, a rede torna-se mais flexível, pois uma estação pode ser movida ou retirada com mais facilidade (PITOURA & SARAMARAS, 1998).

As redes sem fio operam com placas de rede que possuem adaptadores para antenas, através dos quais recebem os dados da rede como ondas de rádio. A comunicação entre as estações em uma rede *Wireless LAN* pode ser feita através de um *Hub* central, instalado em uma parte alta do ambiente, transmitindo e recebendo dados das antenas que foram instaladas nas estações da rede.

Existem vários tipos de tecnologias de transmissão que podem ser utilizadas em redes locais sem fio, dentre elas o infravermelho, o rádio-microondas, o *laser* e o *spread spectrum*.

Infravermelho: As LANs baseadas em infravermelho utilizam a mesma tecnologia usada em produtos como telefones sem fio, controles remotos de aparelhos de TV e videocassetes, entre outros. O infravermelho é mais utilizado em ambientes internos, como escritórios, onde as distâncias entre os pontos de conexão são bem menores em relação às encontradas em ambientes externos. A maior vantagem do

infravermelho é a sua habilidade de suportar grande largura de banda. Uma das desvantagens é a facilidade com que ele pode ser obstruído, pois a luz não atravessa objetos sólidos e opacos como paredes, sendo que também pode receber interferências de iluminação do ambiente.

Rádio-microondas: A principal utilização do rádio-microondas é interconectar redes locais em prédios diferentes. Devem ser utilizados aparatos para microondas em ambos os lados da conexão, chamados de *microwaves dishes*, que possuem o formato de uma antena parabólica. Devem estar na mesma linha de visão para transmitir e coletar os sinais. Os sistemas *Wireless* que utilizam microondas conseguem ultrapassar pequenos obstáculos como por exemplo, paredes finas.

Laser: Os sistemas a *laser* são mais utilizados para conexões de longa distância, como por exemplo à interligação de duas LANs em prédios separados. Podem sofrer interferências climáticas, como chuvas e nevoeiros, que prejudicam a transmissão de dados. Como o infravermelho, os sistemas a *laser* necessitam de visada direta entre os pontos para poder operar, isto é, o receptor deve estar na mesma linha do transmissor para haver comunicação entre os pontos de conexão.

Spread Spectrum: É atualmente uma das tecnologias de transmissão mais utilizadas em *Wireless* LANs, por ser considerada menos sensível a interferências do meio. A transmissão com *Spread Spectrum* atravessa obstáculos com mais facilidade, por utilizar frequências mais baixas, sendo portanto, mais fáceis de ultrapassar barreiras como paredes.

3.3.2 **Wireless WAN**

Uma *wireless* WAN é uma rede que agrega os conceitos de uma WAN e de comunicação sem fio. Assim como a WAN, é uma rede que se estende por uma área geograficamente considerável.

As definições entre os dois tipos de redes se diferenciam quanto à tecnologia. Uma WAN pode utilizar por exemplo, fios, fibra ótica ou *modem* de alta velocidade. Uma *Wireless* WAN usa antenas, transmissores e receptores de rádio, entre outros.

Uma *Wireless* WAN apresenta as seguintes características:

- ⤴ Pode ser utilizada para a comunicação entre vários prédios;
- ⤴ É utilizada por grupos de usuários bem definidos;
- ⤴ Está distribuída por uma extensa área geográfica;
- ⤴ Apresenta um grande número de usuários;
- ⤴ As conexões não dependem de um meio concreto para serem estabelecidas;

Apesar das *Wireless* WANs cobrirem uma extensa área geográfica, em alguns casos podem não prover cobertura em áreas extremamente remotas. Uma cobertura universal para estes tipos de usuários móveis é possível através de redes sem fio baseadas em satélites.

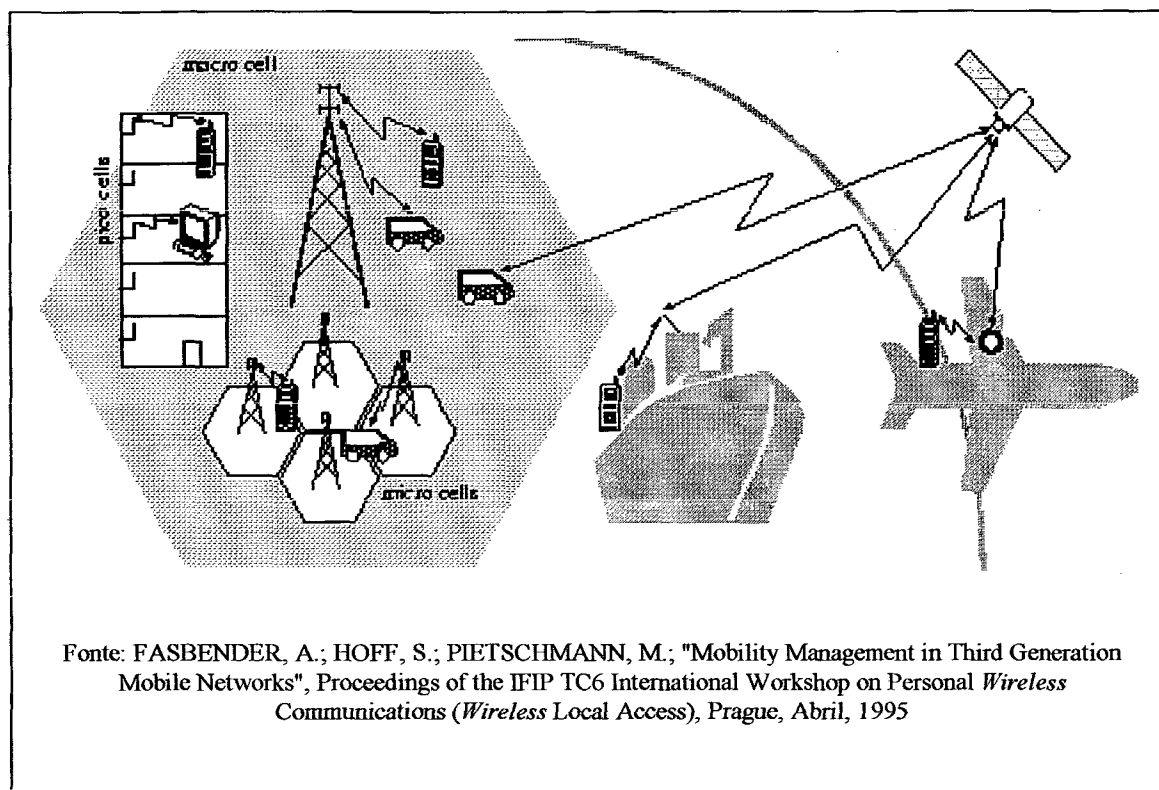
Os satélites são interconectados através de ligações de microondas, formando uma rede global no espaço, estabelecendo uma ligação com a terra através de *gateways*. Os *gateways* são conectados com uma rede pública de telefone e são capazes de realizar conexões com outros usuários que não utilizem serviços de satélite.

Existem vários tipos de satélite que proporcionam comunicação de alturas diferentes da superfície da terra. Eles geralmente podem ser caracterizados por dois modos: baixa comunicação de qualidade de voz e baixa capacidade de transferência de dados.

Sistemas de satélite são classificados em três categorias (MATEUS & LOUREIRO, 1998), figura 14:

- ⤴ **GEO** - *GeoSynchronous Earth Orbit*: satélites de órbita geoestacionária, têm cobertura extensa, possibilitando que um grande número de usuários possa ser alcançado de uma só vez. Estão situados à aproximadamente a 36000 km de altitude e em regiões próximas à linha do Equador.
- ⤴ **MEO** - *Médium Earth Orbit*: satélites de órbita de altura média, posicionados aproximadamente a 10000 km de altitude.
- ⤴ **LEO** - *Low Earth Orbit*: satélites de órbita abaixo da terra, posicionados em torno de 1000 km de altitude, em diferentes posições em relação à terra.

Figura 14 - Rede WAN com Satélite



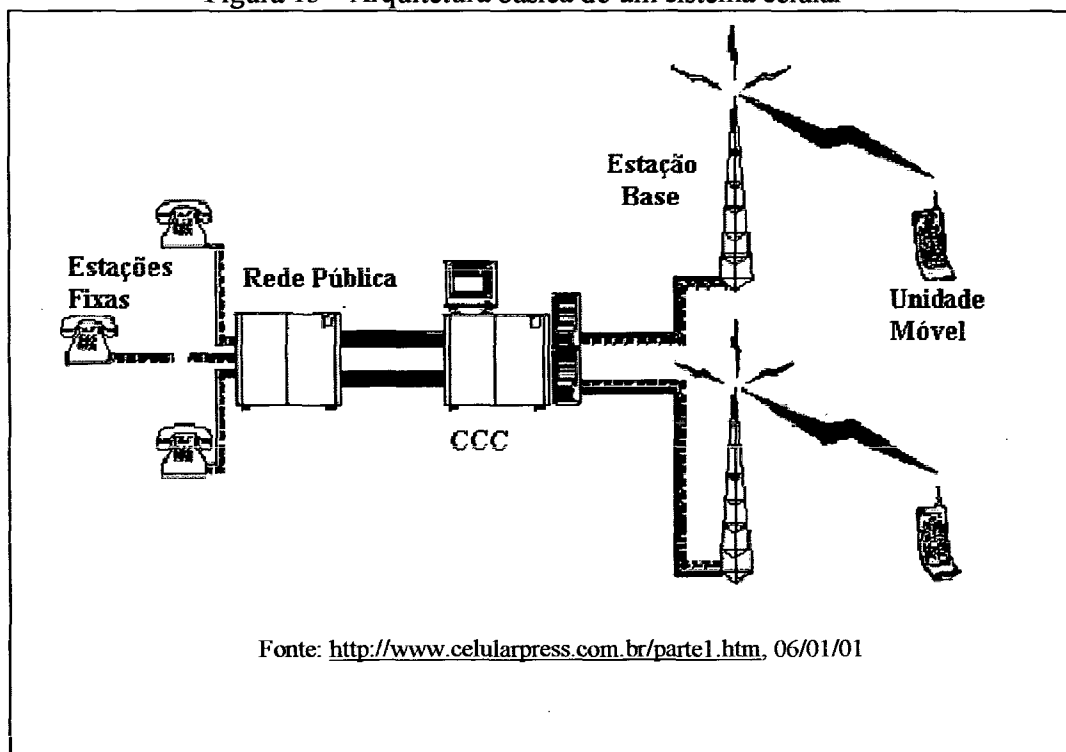
Como exemplo de uma WAN podemos citar a tecnologia de Telefonia Móvel Celular (TMC). Uma célula de telefonia móvel contém diversos canais (frequências), onde um ou mais canais podem ser reservados para a comunicação de dados e compartilhados entre diversos usuários.

Para a implantação da TMC, subdivide-se a área geográfica em pequenas células hexagonais, cada uma dispendo de uma estação-base e de conjuntos de antenas direcionais para supervisão e controle das suas radiofrequências disponíveis. Estações base são conectadas a um centro de comutação e controle (CCC) de rádio-frequência e interligadas com o sistema telefônico convencional (figura 15), este por sua vez, é conectado entre si e com a rede nacional e internacional de telefonia. As principais funções do CCC são:

- ⋄ Realizar a ligação entre a rede telefônica e o sistema móvel celular;
- ⋄ Comunicar-se com outros padrões de sistemas celulares;

- ⤴ Controlar as estações base;
- ⤴ Monitorar e controlar as chamadas;
- ⤴ Interligar várias estações base ao sistema;
- ⤴ Supervisionar o estado do sistema;
- ⤴ Comutar e controlar o *handoff* de sistemas;
- ⤴ Administrar o sistema.

Figura 15 - Arquitetura básica de um sistema celular



Quando um telefone celular deseja fazer uma chamada, envia uma mensagem a estação-base. Essa mensagem, após ser processada e aceita pelo terminal de controle, implica a conexão do telefone celular à estação telefônica através da concessão de uma rádio-freqüência disponível.

O Brasil adotou os padrões norte-americanos de sistemas celulares analógicos e digitais. Para o modo analógico, o padrão utilizado é o AMPS (*Advanced Mobile Phone System* - Sistema de Telefonia Móvel Avançada). Para modo digital, TDMA (*Time Division Multiple Access* - Acesso Múltiplo por Divisão de Tempo) e o CDMA (*Code Division Multiple Access* - Acesso Múltiplo por Divisão de Código). A nível mundial,

são utilizadas outras plataformas digitais, como por exemplo o GSM na Europa e o PDC no Japão.

3.4 Protocolos Móveis

Existem várias tecnologias que podem ser utilizadas para o estabelecimento da comunicação sem fio, dentre elas destacamos o IP móvel e o *Wireless Application Protocol*.

3.4.1 IP Móvel

Em um ambiente Móvel, a entrega correta de pacotes para a unidade móvel depende do endereço IP (*Internet Protocol*) que está sendo usado para conexão com a estação fixa naquele momento. Esse endereço varia em função da mobilidade do usuário. Para mudar o roteamento é necessário um novo endereço IP associado à conexão com a rede fixa. O IP móvel permite o movimento de uma unidade móvel de uma subrede para outra. É igualmente aplicável para mobilidade em meios homogêneos como heterogêneos.

O IP está em sua quarta versão, o Ipv4. Devido à necessidade de endereçamento, roteamento de pacotes, segurança e mobilidade no ambiente móvel, pode-se empregar o IPV6. Neste modelo, o computador móvel possui 2 endereços (MATEUS & LOUREIRO, 1998).

- ♣ O primeiro é o *Home address*, endereço fixo associado a sua estação de origem;
- ♣ O terceiro endereço é o *Care-of address*, endereço móvel. Está associado a cada ponto de acesso à rede e é atribuído pelos métodos de descobrimento de vizinhos do IPv6.

O *Home address* pode ser usado para receber e identificar conexões TCP (*Transmission Control Protocol*). Quando o computador móvel não está conectado a seu endereço fixo, deve existir um mecanismo para receber e enviar todos os pacotes endereçados ao cliente móvel no seu ponto de acesso, que é dado pelo seu *care-of address*. Esta tarefa é executada por um agente chamado de *Home agent* (agente local).

Quando há uma mudança no ponto de acesso à rede fixa, o usuário móvel registra o novo *care-of address* com o seu *home agent*, que se encarrega da entrega dos pacotes para esse novo endereço. Este processo é realizado modificando-se o campo de destinatário no pacote IP do *home address* para o *care-of address*. É chamado de redireção. O nó do endereço *care-of address*, ao receber um pacote para a unidade móvel, aplica o processo inverso, colocando como endereço de destinatário o *home address*. Esta tarefa é executada pelo *Foreign Agent* (agente estrangeiro). (MATEUS & LOUREIRO, 1998).

Tabela 3 – Características do IP Móvel

EXPRESSÕES	CARACTERÍSTICAS
<i>Home address</i>	Endereço associado permanentemente à unidade móvel
<i>Local Link Address</i>	Endereço de ligação local associado a subrede
<i>Care-of Address</i>	Endereço associado à unidade móvel a de cada ponto de acesso
<i>Binding</i>	Associação entre <i>home address</i> e <i>care-of address</i>
<i>Home Agent</i>	Agente local - recebe e envia pacotes endereçados ao cliente móvel
<i>Foreign Agent</i>	Agente estrangeiro - coloca o <i>home address</i> como endereço de destinatário

TANEMBAUM, Andrew S., *Redes de Computadores*, Rio de Janeiro. Campus, 1997

O protocolo IP Móvel é constituído por 3 partes: Identificação *Care-of address*, registro e tunelamento do *Care-of Address* (MATEUS & LOUREIRO, 1998):

Identificar o *Care-of address*: O processo de identificar o *care-of address* no IP móvel é baseado na difusão de dados. O IP móvel acrescenta funcionalidades associadas à mobilidade. *Home agents* e *Foreign agents* transmitem periodicamente divulgações de agentes que podem conter informações para:

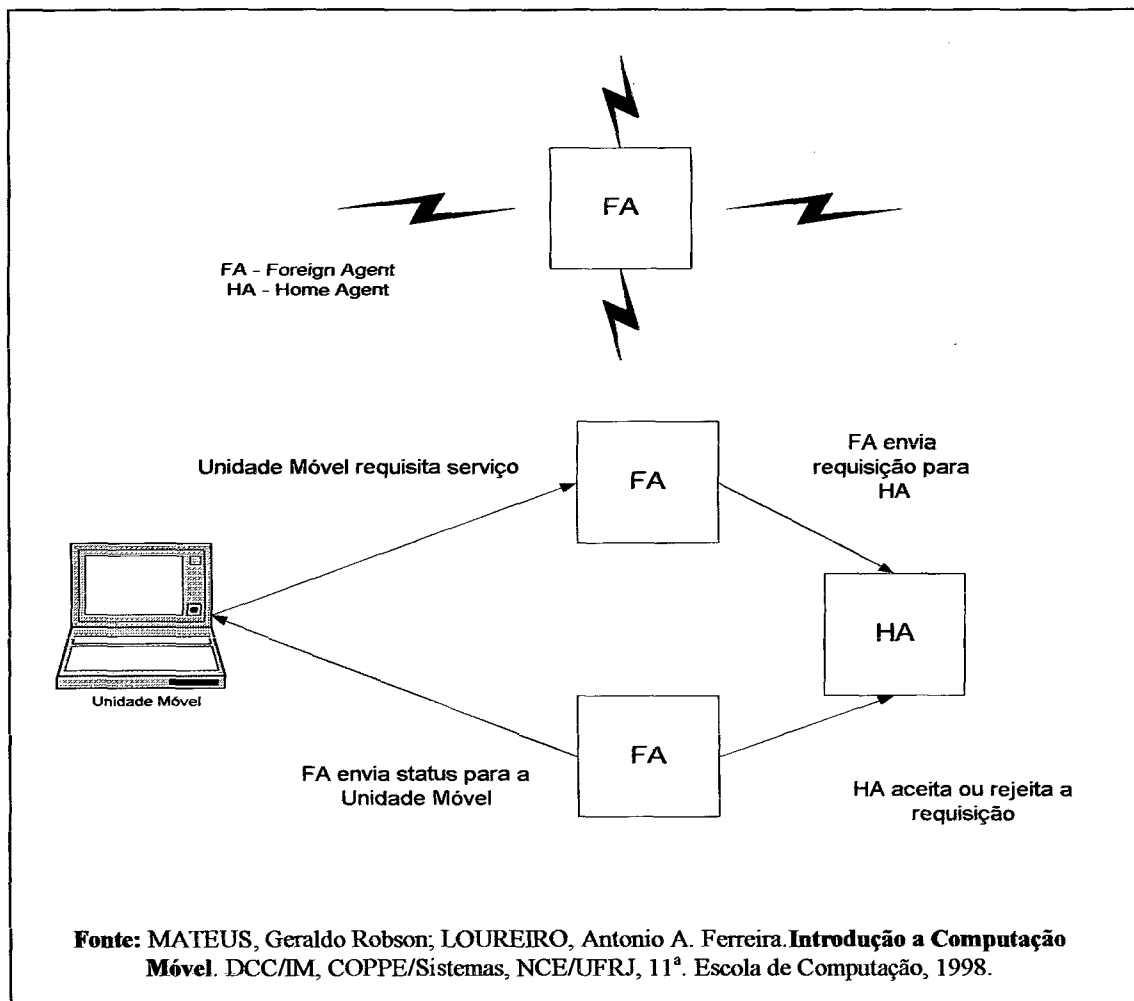
- ⤴ Permitir a identificação dos agentes móveis;
- ⤴ Listar um ou mais *Care-of address* disponíveis;
- ⤴ Informar a unidade móvel sobre funcionalidades oferecidas pelos *foreign agents* como técnicas de encapsulamento;

- ⤴ Permitir a unidade móvel determinar o número da rede e o *status* de seu enlace para Internet;
- ⤴ Permitir as unidades móveis saber se o agente é uma *home agent*, *foreign agent* ou ambos e se, portanto, está na sua rede local ou numa exterior.

Registro do *Care-of Address*: Para registrar o *Care-of address* o cliente móvel envia uma requisição de registro para o seu *home agent* (figura 16), possivelmente com a ajuda de um *foreign agent*. Ao receber a requisição e aprová-la, atualiza a tabela de roteamento e envia uma resposta para a unidade móvel. Os pedidos de registro contêm parâmetros e *flags* que caracterizam o "túnel" que o *home agent* irá usar para enviar pacotes para o *care-of address*. Quando o *home agent* aceita a requisição, é feita uma associação entre o *home address* e o *care-of address*, que é mantido até que termine o tempo de vida do registro. O conjunto das três informações *home address*, *care-of address* e o tempo de vida do registro são chamados de ***binding da unidade móvel***, sendo armazenado em uma estrutura chamada *Binding Cache*.

Tunelamento para o *Care-of address*: O caminho seguido por um conjunto de dados até o *care-of address* é chamado túnel. O mecanismo de encapsulamento padrão, o qual deve ser executado por todos os agentes usando o IP móvel, é o IP dentro de IP (*IP within IP*), onde pacotes IP endereçados para a unidade móvel são primeiramente enviados para o *home agent*, que por sua vez, empacota o IP recebido, insere um novo cabeçalho IP (chamado de cabeçalho de túnel), contendo o *care-of address* e os re-envia para o endereço real da unidade móvel.

Figura 16 - Operações de registro no IP móvel



3.5 Protocolo de aplicação Sem Fio

O WAP (*Wireless Application Protocol*) é um protocolo de aplicações sem fio que provê um padrão universal para trazer o conteúdo da Internet e serviços agregados para telefones móveis e outros dispositivos sem fio, como telefones celulares, PDAs (*Personal Digital Assistants*), rádios, *paggers*, entre outros. O WAP trabalha com as restrições apresentadas por estes aparelhos: processadores e memórias limitadas, telas pequenas e monocromáticas, baixa largura de banda, conexões irregulares, etc.

O WAP foi iniciado pelas empresas Ericsson, Nokia, Motorola e Phone.com <http://www.webproforum.com/wap/topic01.html>, suas especificações são abertas, criando com isso uma padronização nos desenvolvimentos entre estas empresas. Atualmente, a WAP Fórum cuida da administração dessa padronização, fazendo do WAP um padrão global para as empresas que desenvolvem recursos para terminais móveis.

A tecnologia WAP utiliza transmissão digital que garante a compressão de dados através da baixa largura de banda. Pode ser usada pela maioria das redes sem fio e é compatível com diversos tipos de sistemas operacionais. Através do WAP, é possível realizar aplicações como acesso a *e-mails* com interface visual, navegação na Internet e personalização de serviços de mensagens, funções para leitura e resposta de *e-mails* via terminal móvel, serviço pré-pago para transferência de créditos, transações móveis, comércio eletrônico, entretenimento, informações sobre tráfego e temperatura, etc.

3.5.1 *Wireless Markup Language (WML)*

A linguagem adotada pelo WAP é a WML, semelhante a HTML (*hypertext markup language*). É uma linguagem de programação baseada no padrão XML, lido e interpretado por um *microbrowser*² instalado num dispositivo móvel contendo o WAP. A figura 17 apresenta um esquema funcional do protocolo WAP.

O WML possui as seguintes características:

- ⤴ **Suporte para texto e imagens:** O WML disponibiliza aos desenvolvedores meios para especificar textos e imagens que serão apresentados aos usuários. O WML requer que o autor especifique a apresentação e termos gerais e confere ao *user agent* liberdade para determinar como a informação é apresentada.
- ⤴ **Suporte para entrada do usuário:** Suporta vários elementos para solicitar a entrada do usuário. Os elementos podem ser combinados em uma ou mais *cards*³. Todos os pedidos para uma entrada do usuário são realizados em termos

² É uma aplicação contida nos celulares que efetua a navegação em páginas WAP, geradas pela linguagem WML. Ele possui a mesma funcionalidade de um browser de Internet.

abstratos, dando ao *user agent* a liberdade de otimizar o aspecto para um dispositivo particular.

- ⤴ **Navigation e History Stack:** O WML permite vários mecanismos de navegação usando URLs. Incluindo *hyperlinks*, elementos de navegação entre *cards*, tais como elementos do histórico da navegação.
- ⤴ **Suporte Internacional:** O conjunto de caracteres do WML é UNICODE, o que permite a apresentação da maioria das línguas e dialetos.
- ⤴ **Narrow-band Optimisation:** O WML possui tecnologias para otimizar as comunicações em dispositivos móveis. Isto inclui a possibilidade de especificar *cards* múltiplas numa transferência na rede. O WML também inclui outros mecanismos para melhorar o tempo de resposta, minimizar a necessidade de solicitação de serviços aos servidores e as trocas de dados pelo ar.

3.5.2 Wireless Telephony Application (WTA)

O WTA especifica tecnologias, incluindo aplicações para telefonia sem fio, o que permite a combinação de voz e serviços de Internet. O WTA é responsável pela interface dos dispositivos para a rede sem fio. Isto inclui a manipulação de eventos que possibilita que um dispositivo interaja com uma aplicação remota.

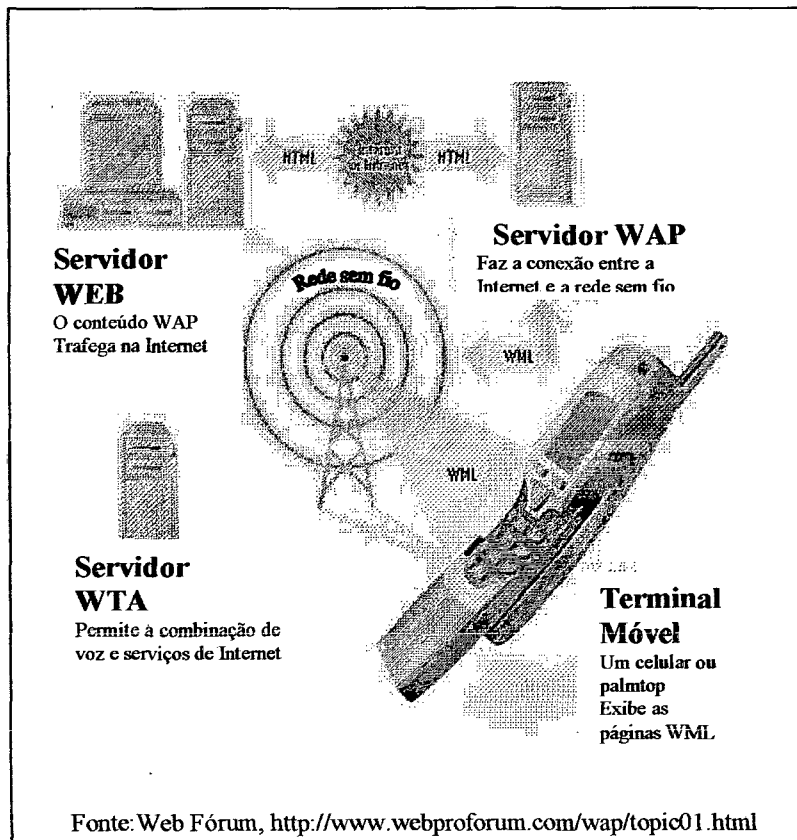
3.5.3 Funcionamento do WAP

O protocolo WAP funciona de modo similar à tecnologia da empregada na Internet, como mostra a figura 17. As etapas envolvidas na utilização do WAP são as seguintes (<http://www.portaisbrasil.com.br/Tecnologia/WAP/wap.html>):

1. O usuário digita o endereço desejado e aperta a tecla de envio do celular;
2. O celular WAP envia um pedido ao servidor WAP (interface entre o WAP e o servidor Web onde se localiza a página desejada) usando o protocolo WAP.
3. O servidor WAP converte o pedido para o protocolo da Internet (HTTP/HTTPS) e o envia para o servidor Web.

4. A chamada é processada pelo servidor Web e é retornada no formato WML ao servidor WAP.
5. O servidor WAP cria uma resposta no formato WAP contendo o conteúdo WML recebido.
6. O usuário do celular WAP recebe o conteúdo WAP exibido na tela do celular.

Figura 17 - Funcionalidade WAP



CAPÍTULO IV

BANCO DE DADOS DISTRIBUÍDOS

A distribuição do sistema permite às aplicações individuais exercerem um controle local sobre os seus dados de maneira que se tornem menos dependentes de um centro de processamento de dados remoto, possibilitando o acesso aos dados de outras localidades. Um banco de dados distribuídos pode ser mais confiável devido à multiplicidade e a um certo grau de autonomia existentes em suas partes. Os dados podem ser armazenados de maneira distribuída, próximos ao seu ponto normal de uso, reduzindo o tempo de resposta e os custos de comunicação.

Um sistema de banco de dados distribuído consiste em uma coleção de nós, cada qual podendo participar na execução de transações que acessam dados em um ou diversos nós.

Neste capítulo aborda-se banco de dados distribuídos devido a sua importância para o gerenciamento de dados móveis. São tratados alguns aspectos relacionados a bancos de dados como: processamento de consultas, gerenciamento de transações, controle de concorrência e consistência de dados.

O processamento de consultas corresponde ao ato de extrair informações de um banco de dados ou a qualquer acesso solicitado a algum repositório de dados. Uma consulta pode ser uma seleção de elementos numa linguagem de alto nível sobre um nó da rede, numa seqüência de instruções elementares, as quais recuperam dados armazenados em um banco de dados. O principal objetivo do processamento de

consultas em um ambiente distribuído é transformar uma consulta de alto nível (cálculo relacional), vista pelos usuários como um único banco de dados, utilizando uma estratégia eficiente de execução, expressa uma linguagem de baixo nível (álgebra relacional) em um banco de dados local.

Uma transação pode ser definida como uma seqüência de operações realizadas sobre um conjunto de dados, delimitada por um comando de inicialização e finalização (MELO et al, 1997) . É uma unidade de controle de acessos a dados solicitados por um programa ou pelas consultas realizadas por usuários. É tratada como um bloco único, atômico e indivisível relacionado à recuperação de dados.

O controle de concorrência é a parte de um SGBD-D capaz de assegurar que transações executadas concorrentemente produzam os mesmos resultados que se fossem executadas em seqüência. O objetivo principal do controle de concorrência é assegurar que todas as transações possam ser executadas em série, isoladamente, uma após a outra, garantindo resultados corretos e um estado consistente.

A consistência é mantida quando cada item do banco de dados satisfaz restrições de consistência de acordo com a modelagem previamente definida. Mecanismos presentes nos SGBD-Ds garantem o estado consistente ao final da transação.

4 Estrutura de Banco de Dados Distribuído

Um sistema distribuído caracteriza-se por um conjunto de múltiplos computadores, interligados em rede e que utilizam tempo compartilhado para a execução cooperativa de programas de aplicação, com controle geral de recursos descentralizados.

Os dados podem ser armazenados de forma distribuída, próximos ao seu ponto normal de uso, reduzindo o tempo de resposta e os custos de comunicação. O paralelismo inerente às redes de localidades múltiplas pode fornecer uma passagem de dados aperfeiçoada e possivelmente, melhorar os tempos de resposta em certas situações.

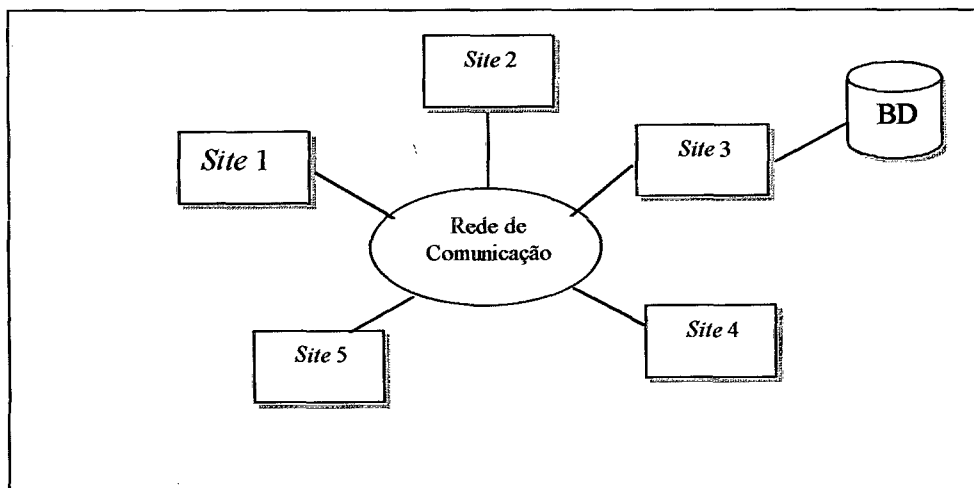
Em um sistema distribuído os processadores podem variar em tamanho e função. Podem incluir microcomputadores, estações de trabalho, minicomputadores, entre

outros. Esses processadores são chamados por diversos nomes, tais como, nós, computadores ou *sites*, dependendo do contexto no qual eles são mencionados. Uma das principais vantagens de compartilhar os dados pela distribuição é o fato de que cada *site* pode reter um grau de controle sobre os dados armazenados localmente.

A principal característica que distingue os sistemas de processamento distribuído dos sistemas de arquiteturas clássicas está na descentralização do controle. Os sistemas centralizados possuem maior acessibilidade com um ambiente homogêneo ao usuário; a congruência de informação em uma operação normal é naturalmente assegurada, uma vez que se trata de uma única máquina. Os sistemas distribuídos são heterogêneos, possuindo maior confiabilidade perante falhas sucessivas de componentes. Um conjunto de máquinas é visto por seus usuários como se fossem um único sistema de tempo compartilhado. No caso de uma dessas máquinas falhar no decorrer de uma operação, o sistema poderá continuar funcionando como um todo. (TANEMBAUM, 1995).

Em um sistema centralizado, o administrador de banco de dados controla todos os dados, sendo que estes são gerenciados em apenas um *site* como mostra a figura-18 (*site 3*, neste caso).

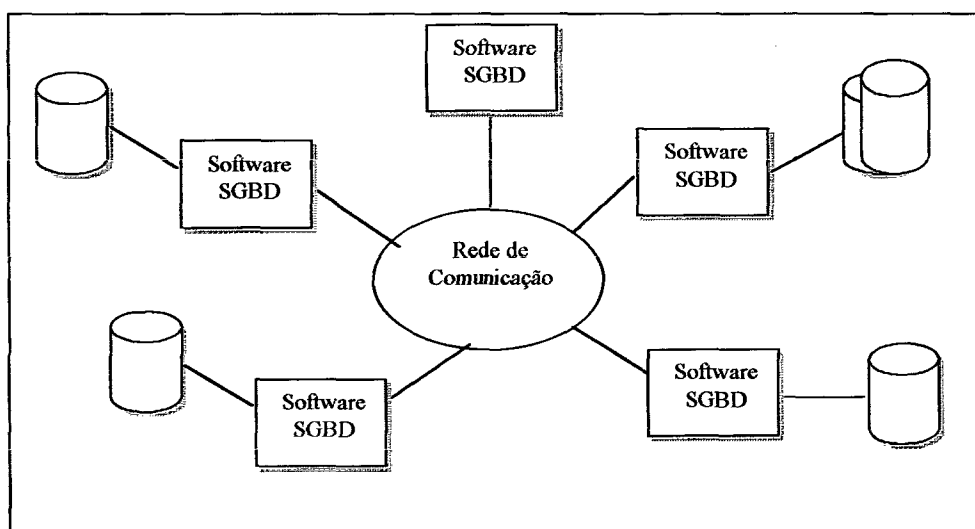
Figura 18 - Banco de Dados Centralizado



Um sistema de banco de dados distribuídos consiste em uma coleção de nós, cada qual podendo participar na execução de transações que acessam dados em um ou diversos nós. Um administrador global é responsável pelo sistema inteiro, mas parte desta responsabilidade é delegada aos administradores locais de cada *site*. A possibilidade de autonomia local é freqüentemente a maior vantagem de bancos de dados distribuídos (BDD).

(OZU & VALDURIEZ, 1999) define um BDD como uma coleção de múltiplos bancos de dados logicamente inter-relacionados, distribuídos sobre uma rede de computadores, como mostra a figura 19:

Figura 19 - Ambiente de um Banco de Dados Distribuído



De acordo com (DATE, 1991), um sistema distribuído é qualquer sistema que envolve múltiplas localidades conectadas juntas em uma espécie de rede de comunicação, na qual o usuário de qualquer localidade pode acessar os dados armazenados em outro local. Cada localidade, por sua vez, pode ser considerada como um sistema de banco de dados em si, que tem seu próprio banco de dados e sua própria função DBA (administrador de banco de dados) seus próprios terminais e usuários, o seu próprio armazenamento local e a sua própria CPU (unidade central de processamento), executando o próprio SGBD local. O mesmo também tem seu próprio

gerenciador local, com a responsabilidade (dentre tantas outras) de controlar a troca de mensagens com outras localidades integrantes do sistema distribuído.

Um sistema de gerenciamento de banco de dados distribuídos (SGBD-D) é um *software* que gerencia o BDD e provê um mecanismo de acesso que torna a distribuição dos dados transparente para os usuários, ou seja, o usuário tem uma visão global do sistema, não percebendo como os dados são acessados. Com base nestas definições, um sistema de banco de dados distribuídos (SBDD-D) pode ser definido como a junção de um BDD e um SGBD.

A maioria dos especialistas, dentre eles (TANENBAUM, 1995), concordam em afirmar que os principais objetivos dos sistemas distribuídos são:

- ⤴ **Transparência:** o usuário não deve perceber a individualidade dos componentes do sistema, que serão vistos como um todo, ou seja, como um sistema indivisível funcionalmente. Os acessos aos recursos locais ou remotos devem ser feitos utilizando as mesmas funções, de forma que o usuário não precise conhecer a localização dos recursos.
- ⤴ **Compartilhamento de recursos:** todos os recursos do sistema (processador, arquivos, periféricos, etc) podem ser utilizados por qualquer usuário, de forma transparente. Isto permite que as estações ociosas sejam utilizadas mesmo que estejam remotas a um dado usuário.
- ⤴ **Alto desempenho:** o sistema deve executar todas as suas tarefas da maneira que melhor lhe convier, aproveitando o compartilhamento de recursos e visando o maior desempenho do sistema.
- ⤴ **Confiabilidade:** o sistema deve ser totalmente confiável, o que acarreta a necessidade de mecanismos que permitam que o sistema seja tolerante a falhas.

4.1 Vantagens e Desvantagens

A utilização de sistemas distribuídos torna-se fundamental quando se deseja obter maior desempenho, pois estes sistemas são capazes de distribuir as tarefas entre vários processadores, reduzindo a ociosidade dos mesmos e agilizando o tempo de execução.

A evolução tecnológica vem propiciando a multiplicidade de recursos a custos acessíveis, enquanto que os mecanismos de cooperação entre processos têm sido usados para aumentar a eficiência, a confiabilidade e a disponibilidade de sistemas. Como principais vantagens dos sistemas distribuídos podemos citar as seguintes (TANENBAUM, 1995):

- ✧ **Crescimento incremental:** é possível adicionar mais processadores ao sistema, permitindo que ele seja expandido gradualmente, conforme a necessidade;
- ✧ **Questão econômica:** a relação custo/performance é maior do que os sistemas centralizados, pois em um sistema distribuído pode-se juntar um grande número de processadores baratos em um único sistema;
- ✧ **Confiabilidade:** sistemas distribuídos podem ser mais confiáveis devido à multiplicidade e a um certo grau de autonomia de suas partes. A falta de um processador pode causar a falha de uma única máquina, não afetando o restante do sistema;
- ✧ **Aplicação distribuída:** algumas aplicações são essencialmente distribuídas, envolvendo máquinas separadas fisicamente. Neste caso, se torna necessário o emprego de um sistema distribuído.

Apesar de todos os pontos positivos em sistemas distribuídos, a implementação e aplicações apresentam algumas desvantagens (TANENBAUM, 1995). São elas:

- ✧ **Ligação em rede:** podem ser perdidas mensagens na rede, o que obriga a utilização de um *software* especial para fazer a manipulação das mensagens. Outro aspecto é a possibilidade da rede ficar sobrecarregada com o tráfego gerado pelo intercâmbio de mensagens;

- ⤴ **Segurança:** dados secretos podem ser facilmente acessados se não houver um mecanismo de proteção adequado para o sistema.

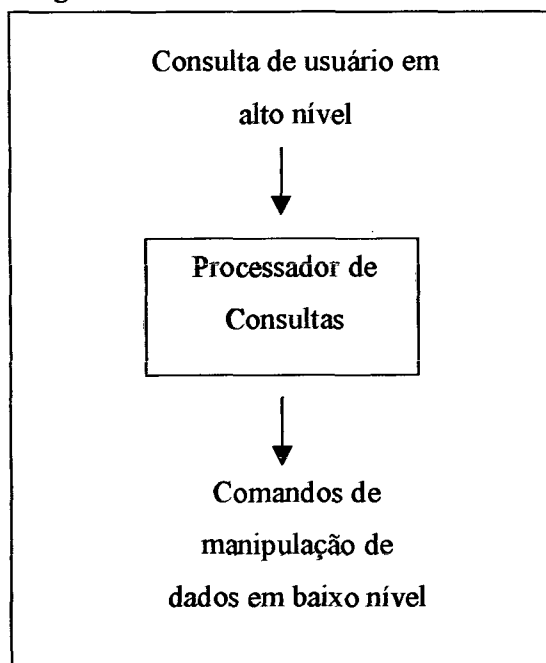
A seguir serão abordados aspectos relacionados a banco de dados distribuídos. São eles: processamento de consultas, gerenciamento de transações, controle de concorrência e consistência de dados.

4.2 Processamento de Consultas

O processamento de consultas corresponde ao processo de extrair informações de um banco de dados ou qualquer acesso solicitado a algum repositório de dados. Uma consulta pode ser uma seleção de elementos numa linguagem de alto nível sobre um nó da rede, numa seqüência de instruções elementares, as quais recuperam dados armazenados em um banco de dados.

De acordo com (OZU & VALDURIEZ, 1999), o principal objetivo do processamento de consultas em um ambiente distribuído é transformar uma consulta de alto nível (cálculo relacional), vista pelos usuários como um único banco de dados, utilizando uma estratégia eficiente de execução, expressa uma linguagem de baixo nível (álgebra relacional) em um banco de dados local, como mostra a figura 20.

Figura 20 - Processamento de Consultas



Em sistemas centralizados, o principal critério para medir o custo de uma consulta é o número de acessos a disco. Em sistema distribuído, além da questão anterior, acrescenta-se o custo de transmissão da rede e o desempenho de diversos nós executando processamento paralelo.

O processamento de consultas em um BDD é considerado muito mais complexo que em um banco de dados centralizado, pois a integridade do sistema precisa ser mantida e os dados consultados devem os mais atualizados. Quando uma consulta é realizada e os dados não são duplicados nem fragmentados, se os dados estiverem no nó local, a consulta será executada rapidamente, porém, se os dados estiverem em outro nó, a execução dependerá do acesso aos dados via rede.

Um fator que deve ser observado diz respeito ao tipo de banco de dado distribuído: heterogêneo ou homogêneo. Se for heterogêneo, haverá um custo de processamento adicional para que a comunicação seja efetuada e, caso haja a replicação de dados, os arquivos a serem consultados estão fragmentados e gravados pelos diversos nós da rede. Isto implica que, de alguma maneira, os dados terão que ser agrupados para a resolução da consulta. Neste caso, a consulta deve ser otimizada, isto é, deve ser dividida em subconsultas de maneira a diminuir o número de acessos a disco e os produtos cartesianos entre as tabelas.

Nas consultas realizadas em um banco de dados homogêneo, tem-se um tradutor em cada nó, o qual traduz as instruções solicitadas. Já nas consultas realizadas em um banco de dados heterogêneo, tem-se também um tradutor em cada nó da rede, porém, com a função de realizar a comunicação entre os BDDs. O mapeamento de tabelas e a duplicação de dados são tarefas complexas e o tradutor deve realizá-las de maneira rápida e eficiente.

A consulta em um BDD é realizada conforme o seu modelo relacional. Divide-se em duas fases:

1. Especificar o domínio da relação a ser recuperada, ou seja, definir quais tabelas serão utilizadas na consulta;
2. Definir os predicados de seleção.

A complexidade da consulta aumentará de acordo com a quantidade de operadores lógicos e operadores de comparação existentes no banco de dados. O acesso aos nós da rede de BDD pode ser dividido em acesso centralizado e acesso distribuído. Na estratégia distribuída, são executadas todas as subconsultas nos diversos nós da rede, juntamente com as operações lógicas e comparativas que puderem ser resolvidas em cada nó. A resposta final é gerada como a união das diferentes respostas de todos os nós locais da rede.

A estratégia centralizada consiste em agrupar para uma determinada consulta todas as subconsultas e em seguida, executar todas as operações lógicas em um simples nó. A principal característica deste tipo de consulta é que na fase final inclui-se um processamento único de consultas no local da consulta.

Caso uma consulta seja realizada onde os dados estão fragmentados em diversos nós, se torna necessário reunir todas as partes através de junções e uniões para reconstituir a tabela original. Se os dados estão duplicados, deve-se escolher a réplica mais atualizada para fazer a consulta.

4.3 Gerenciamento de Transações

Uma transação pode ser definida como uma seqüência de operações realizadas sobre um conjunto de dados, delimitada por comandos de inicialização e finalização (MELO et. al, 1997). É uma unidade de controle de acessos a dados solicitados por um programa ou pelas consultas realizadas por usuários. É tratada como um bloco único, atômico e indivisível relacionado à recuperação de dados.

As transações devem ser executadas concorrentemente, de forma transparente e confiável para a aplicação. A transparência, neste contexto, refere-se à manutenção do banco de dados em um estado consistente, mesmo na presença de várias transações executadas simultaneamente. A confiabilidade refere-se à capacidade do sistema de banco de dados em resistir aos diversos tipos de falhas que podem ocorrer no ambiente durante a execução de uma transação.

Como produto da sua execução, uma transação pode mudar o estado do banco de dados ou não provocar qualquer alteração, dependendo dos efeitos de sua execução

serem aceitos com sucesso (*commit*)⁴ ou cancelados (*abort*)⁵. Quando uma transação é cancelada, todas as mudanças feitas por ela são desfeitas (*rollback*).

Um exemplo de aplicação é um programa que realiza a transferência de valores entre contas. Tal programa realiza duas transações num banco de dados compartilhado:

- ⤴ A primeira verifica se o usuário possui saldo suficiente para pagar um montante;
- ⤴ A segunda realiza a transferência do montante, debitando de uma conta e creditando na outra. (FERREIRA & FINGER, 2000). O sucesso desta transação significa que o montante saiu de uma conta e entrou em outra. O fracasso, que as duas contas permanecem com estavam.

A seguir serão descritas as propriedades básicas do modelo de transações e que devido as suas iniciais ficaram conhecidas como propriedades ACID.

4.3.1 Propriedades ACID

A integridade transacional diz respeito às transações que satisfazem certas características, tradicionalmente agrupadas em quatro propriedades referenciadas como ACID - **A**tomicidade, **C**onsistência, **I**solamento e **D**urabilidade (FERREIRA & FINGER, 2000).

⤴ **Atomicidade:** refere-se aos efeitos das transações sobre o estado do banco de dados como unidades de operação únicas e indivisíveis, ou seja, todos ou nenhum dos efeitos são considerados.

⤴ **Consistência:** proporciona o estado consistente do sistema ao final da transação. É expressa de duas maneiras: a consistência da transação em si mesma e a consistência do banco de dados face à execução concorrente de transações.

⤴ **Isolamento:** garante que cada transação deve ver um estado consistente do banco de dados sempre que for executada. Em execuções paralelas ou simultâneas, cada transação será executada livre das interferências de outras,

⁴ A operação *commit* informa que a transação foi realizada com sucesso e todos os seus efeitos deverão ser permanentes no sistema.

⁵ A operação *abort* informa ao sistema que uma transação não pode ser completada com sucesso.

fazendo com que cada transação não torne suas atualizações visíveis até que seja confirmada.

^ **Durabilidade:** os efeitos de uma transação, uma vez completada, tornam-se permanentes no banco de dados, ou seja, não podem ser desfeitos.

Os modelos de transação consideram dois fatores importantes: a serialização e a confiabilidade. A serialização é suportada pelo mecanismo de controle de concorrência que garante executar cada transação isoladamente, uma após a outra. A confiabilidade assegura que as consultas feitas pelo usuário são realizadas por completo e os efeitos de suas ações são refletidas no banco de dados. Geralmente as propriedades de atomicidade e isolamento suportam a serialização no gerenciamento de transação, enquanto que o isolamento e a consistência garantem a confiabilidade (MELO et. al, 1997).

4.4 Controle de Concorrência

O controle de concorrência é à parte de um SGBD-D capaz de assegurar que transações executadas concorrentemente produzam os mesmos resultados como se fossem executadas em seqüência. O objetivo principal do controle de concorrência é assegurar que todas as transações possam ser executadas em série, isoladamente, uma após a outra, garantindo resultados corretos e um estado consistente para o banco de dados.

Caso uma transação termine com sucesso, todas as alterações aos recursos compartilhados serão permanentes. Caso contrário (termine sem sucesso), ela não tem nenhum efeito, ou seja, todas as alterações feitas pela transação devem ser desfeitas. (FERREIRA & FINGER, 2000)

Como exemplo considere um banco de dados que gerencia reservas de avião. Usualmente, usuários estão reservando assentos e verificando poltronas vazias. Quando o banco de dados é acessado e possivelmente modificado ao mesmo tempo, o SGBD-D pode ordenar seus pedidos, evitando desta forma conflitos entre as operações.

Essa função de gerenciar acessos concorrentes ao banco de dados, é muito importante num SGBD-D e faz com que cada usuário ignore o fato de que estão feitos outros acessos aos dados concorrentemente.

O nível de concorrência indica o número de transações que são executadas de forma concorrente sendo um dos parâmetros mais importantes em sistemas distribuídos. O mecanismo de controle de concorrência procura manter a consistência e a integridade do banco de dados, bem como, o alto nível de concorrência.

O controle de concorrência possibilita o compartilhamento de dados de forma completamente transparente aos usuários. No caso de dois ou mais clientes acidentalmente acessarem o mesmo arquivo ao mesmo tempo, podem ocorrer problemas tratados através de bloqueios. Esta solução permite aos clientes bloquearem arquivos antes de os utilizarem. São usadas duas espécies de bloqueios: para leitura e para gravação.

- ⤴ os bloqueios para leitura são chamados de compartilhados;
- ⤴ os bloqueios para escrita são chamados de exclusivos;

Duas ou mais transações podem deter bloqueios de leitura sobre um mesmo objeto de dado, simultaneamente. Apenas uma transação, no entanto, num determinado instante, pode deter o bloqueio de gravação sobre um objeto de dado específico (MELO et. al, 1997).

O controle de concorrência, na maioria dos sistemas distribuídos baseia-se em bloqueios para garantir a seriabilidade de execução de transações concorrentes.

4.5 Consistência de Dados

A consistência de um banco de dados é mantida quando cada item satisfaz restrições de consistência de acordo com a modelagem previamente definida. Mecanismos presentes nos SGBD-Ds garantem um estado consistente do banco de dados ao final de uma transação. Por exemplo, em um sistema de reservas de avião, uma restrição de consistência pode ser a de que cada poltrona seja reservada por apenas um passageiro.

A integridade e consistência em bancos de dados distribuídos são baseadas no conceito fundamental de transações, como nos bancos de dados centralizados. Isso significa que um SGBD-D é responsável por manter o banco de dados em um de dois estados consistentes: antes do começo da transação de atualização e depois da execução com sucesso da transação de atualização. O objetivo de uma transação é deixar o banco de dados num estado consistente ao final de sua execução.

Pelo fato dos clientes fazerem atualizações locais em conjuntos de dados, pode haver a possibilidade de uma consistência fraca devido a alguns problemas que possam ocorrer. Diferentes clientes podem tentar atualizar os mesmos dados de várias maneiras. Existem duas aproximações para problemas de consistência e desconexões intermitentes (BADRINATH & PHATAK, 1998). A primeira aproximação é pessimista e requer sincronização explícita antes da desconexão para manter a consistência. Esta aproximação tenta eliminar conflitos. Em contrapartida requer mais informações, reduzindo acessos ao sistema.

A segunda aproximação é otimista e permite o conflito nas atualizações em todas as máquinas desconectadas resolvendo os conflitos na reconexão. Esta aproximação é difícil de implementar e pode haver problemas devido ao alto número de conflitos, contudo provém alta acessibilidade ao sistema.

CAPÍTULO V

BANCO DE DADOS MÓVEIS

O volume de dados processados e principalmente armazenados tem crescido de forma exponencial, por este motivo estão surgindo tecnologias novas que possam acompanhar esta evolução. Grandes conquistas tecnológicas do homem estão se miniaturizando em alta velocidade. É importante que o banco de dados busque adaptabilidade à velocidade das transformações que ocorrem em todos os setores computacionais, conferindo uma crescente importância para a habilidade de armazenar, gerenciar e recuperar dados.

A utilização de banco de dados móveis propicia facilidades de locomoção física que possibilitam aos usuários acesso aos dados a qualquer hora e em qualquer lugar. Usuários têm a possibilidade de participar de congressos, convenções e reuniões de negócios e realizar visitas a clientes e fornecedores, tendo acesso a seus arquivos pessoais armazenados em uma estação de trabalho distante, participando de teleconferências e efetuando, normalmente, suas tarefas computacionais diárias, mesmo quando distantes de sua residência ou local de trabalho.

Vendedores de campo são exemplos de trabalhadores com alta mobilidade que necessitam acessar bases de dados remotas e executar operações diversas. No seu computador portátil, carregam um banco de dados, contendo os produtos que são vendidos, podendo entrar em contato com a estação base de sua empresa repassando as vendas realizadas, atualizando o cadastro com novos clientes, realizando consultas de

preços, de quantidades em estoque, do valor da comissão a ser recebida, da requisição de mercadorias, entre outras informações.

Outro exemplo da utilização de banco de dados móveis é o de pesquisadores que estão viajando regularmente pelo mundo e precisam entrar em contato com laboratórios para atualização de dados. Através dos recursos de banco de dados móveis, o pesquisador não precisa enviar dados coletados para a rede fixa a toda hora. Poderá fazer isso somente quando for necessário.

Um ambiente móvel é altamente dinâmico. Entre as considerações sobre bancos de dados móveis está o equipamento utilizado, a alocação de canais para a comunicação, os tipos de comunicação, as falhas que possam ocorrer, o tratamento de desconexões, etc.

Neste capítulo estuda-se banco de dados móveis e vários aspectos de gerenciamento de dados que são comentados a seguir:

- ⤴ Paradigmas de acesso aos dados (*caching* e *broadcasting*);
- ⤴ Gerenciamento de transações;
- ⤴ Processamento de consultas;
- ⤴ Recuperação de falhas;
- ⤴ Replicação de dados;
- ⤴ Segurança.

O mecanismo de *caching* (armazenamento) em um ambiente móvel tem como objetivo o acesso mais eficiente aos dados, permitindo o trabalho desconectado, possibilitando uma economia de energia e disponibilidade de recursos.

A difusão de dados ou *broadcasting* consiste num sistema, onde a mesma mensagem é enviada de uma estação base para todas as unidades móveis que estejam em uma célula. Os dados difundidos podem ser recebidos por um grande número de unidades móveis de uma só vez, sem custos adicionais.

Uma transação móvel pode ser definida como uma transação distribuída, onde algumas partes do processamento são executadas em uma unidade móvel e outras partes em unidades fixas.

Em se tratando do processamento de consultas em bancos de dados móveis exige a localização precisa da unidade móvel. O uso de computadores em navios, aviões, carros entre outros meios de transporte, dificulta esta localização, devido às constantes mudanças de lugar. Consultas em banco de dados móveis possui maior complexidade se comparado à consultas em banco de dados centralizados, pois tem um maior número de parâmetros que afeta o desempenho dessas consultas.

A replicação de dados faz com que um determinado objeto de dados possa ter fisicamente várias cópias armazenadas em diversas localidades diferentes. Isto proporciona um aumento da disponibilidade de dados, melhorando o desempenho do sistema.

A recuperação é responsável por preservar a consistência do banco de dados após falhas do sistema, de transações ou dos meios de comunicação. Uma parte essencial do sistema é um esquema de recuperação responsável pela detecção de falhas e pela restauração do banco de dados para um estado consistente que existia antes da ocorrência da falha.

A segurança em aparelhos móveis é um dos maiores problemas no ambiente móvel. As vulnerabilidades dos sistemas e dos dados tornam-se maiores e a troca de mensagens pode ser passivamente capturada por pessoas não-autorizadas.

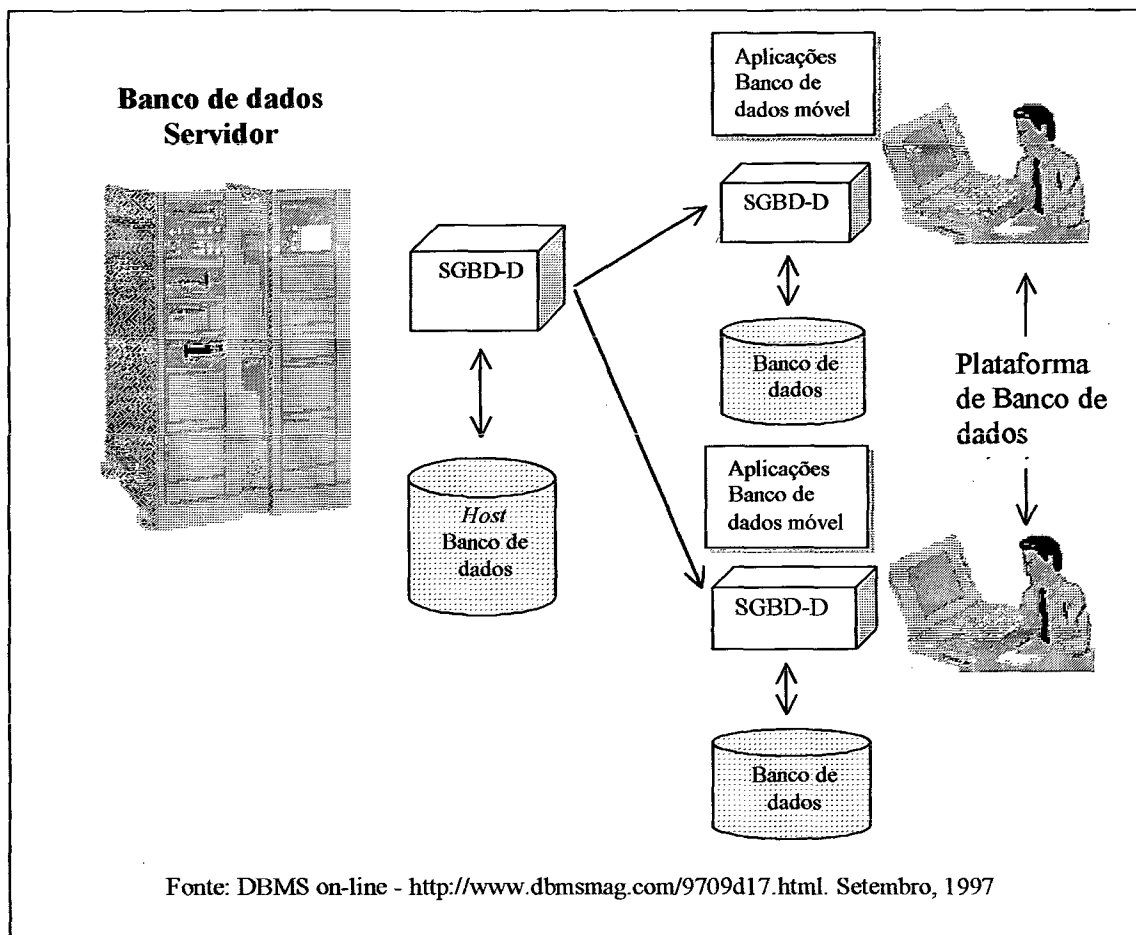
5 Arquitetura de Banco de dados Móveis

A arquitetura de um banco de dados móveis, consiste de plataformas de aplicação, um *host* servidor e a comunicação realizada entre eles, como mostra figura 21.

A plataforma de aplicação de bancos de dados móveis é um computador portátil que contém o banco de dados, onde são armazenados informações, algumas camadas de software, o sistema operacional, o software de comunicação, o software responsável pelo gerenciamento do sistema e a aplicação de banco de dados móvel.

Os usuários interagem com a aplicação do banco de dados e acessam informações armazenadas através de um SGBD-D. Os *softwares* para um ambiente distribuído devem tratar problemas de comunicação, desconexão, topologia dinâmica, entre outros fatores.

Figura 21 - Arquitetura de Banco de Dados Móveis



O banco de dados servidor normalmente é um computador de médio ou grande porte que é coordenado por um sistema de gerenciamento. O *host* que armazena o banco de dados contém dados dinâmicos que podem ser usados pelos usuários móveis. A posição do *host* na organização depende do tipo de informações que está sendo processado.

O banco de dados móvel troca informações com o computador que contém o banco de dados, de modo a manter a atualização, realizar consultas, etc. A comunicação entre ambos acontece com interrupções e em intervalos irregulares, por curtos períodos de tempo, sendo que, algumas vezes, não estão conectados em uma mesma rede.

A tabela 4 resume alguns tópicos relacionados ao gerenciamento de dados móveis e características que estão abordados neste capítulo.

5.1 Gerenciamento de Dados Móveis

Alguns aspectos fundamentais de gerenciamento de dados móveis serão abordados neste trabalho. A tabela a seguir resume os aspectos de gerenciamento de dados e suas principais características:

Tabela 4 - Aspectos de Gerenciamento de Dados Móveis

ASPECTOS DE GERENCIAMENTO DE DADOS	CARACTERÍSTICAS
PARADIGMAS DE ACESSO AOS DADOS	<ul style="list-style-type: none"> ⤴ <i>Caching</i> ⤴ Difusão de dados (<i>Broadcasting</i>)
GERENCIAMENTO DE TRANSAÇÕES	<ul style="list-style-type: none"> ⤴ Modelos de transações móveis ⤴ Graus de mobilidade ⤴ Alocação de canais
PROCESSAMENTO DE CONSULTAS	<ul style="list-style-type: none"> ⤴ Dependência de localização ⤴ Diferentes fatores de custo ⤴ Respostas de consultas retornam para lugares diferentes ⤴ Necessidade de técnicas adaptativas
REPLICAÇÃO DE DADOS	<ul style="list-style-type: none"> ⤴ Graus e tipos de replicação ⤴ Modelos de replicação
RECUPERAÇÃO DE FALHAS	<ul style="list-style-type: none"> ⤴ Particionamento freqüente da rede ⤴ Desconexões voluntárias não são consideradas falhas do sistema ⤴ Técnicas de recuperação de desconexão durante <i>handoff</i>
SEGURANÇA	<ul style="list-style-type: none"> ⤴ Encriptação de dados ⤴ Assinatura Digital ⤴ <i>Firewall</i>

5.2 Paradigmas de Acesso aos Dados Móveis

Os computadores portáteis oferecem um grande número de limitações. Devido a este fator, existe a necessidade de formas de acesso aos dados que reduzam o número de acessos aos servidores de arquivos, diminuindo o tráfego de informações na rede de comunicação. Neste trabalho aborda-se o método *caching* e a difusão de dados.

5.2.1 *Caching*

A comunicação sem fio utiliza a desconexão por períodos substanciais como meio de economia, onde a largura de banda é limitada e variável. Mesmo estando desconectadas, algumas unidades móveis podem permanecer em operação, permitindo que os dados sejam utilizados resultando em diminuição de carga tanto no servidor quanto no canal de comunicação. O uso de memória *cache* para consultas e atualizações pode reduzir o número de acessos aos servidores de arquivos, diminuindo o tráfego de informações na rede de comunicação. Um cliente móvel armazena dados acessados freqüentemente para melhorar o desempenho das consultas durante a desconexão.

O mecanismo de *caching* (armazenamento) em um ambiente móvel tem como objetivo o acesso mais eficiente aos dados. É utilizado a fim de aumentar a probabilidade do dado procurado estar armazenado localmente na unidade móvel, fazendo com que permaneçam em memória dados que serão possivelmente reutilizados, de modo a reduzir a contenção de dados e o tempo de repostas a consultas e conseqüentemente, melhorar o desempenho do sistema.

Por outro lado, o mecanismo de *caching* pode ocasionar inconsistência nos dados utilizados pelos usuários, tornando-os desatualizados, pois algumas vezes é praticamente impossível atualizar os mesmos itens em diferentes servidores simultaneamente. Além disso, canais sem fio sofrem de baixa largura de banda e estão vulneráveis a freqüentes desconexões. Neste caso, a consistência dos dados também pode ser afetada, pois, as mensagens enviadas entre servidor e unidade móvel podem ser invalidadas.

A consistência dos dados pode ser aperfeiçoada, desde que as transações realizadas no sistema sejam feitas utilizando a serialização. A manutenção de

consistência do *caching* é uma tarefa complexa, que aumenta a carga na rede, comprometendo o desempenho do sistema.

A natureza (conteúdo que será armazenado) do *caching* pode ser estática ou dinâmica. A natureza estática diz respeito ao conteúdo que é definido com antecedência pelo sistema, sendo solicitado no início de sua execução. A natureza dinâmica se refere ao conteúdo que é definido em tempo de execução, sendo realizado através do armazenamento dos resultados obtidos das consultas.

O armazenamento automático é uma solução utilizada para suportar operações desconectadas, onde arquivos remotos são armazenados localmente nos clientes. A dificuldade está em definir quais arquivos devem ser selecionados para armazenamento local. Entre as soluções possíveis está em escolher os arquivos utilizados mais recentemente ou fazer com que o usuário participe do gerenciamento dos conteúdos armazenados. Para coordenar a classificação dos dados quanto a seu uso, são usadas duas regiões distintas, definidas como região quente e região fria (ARAÚJO & FERREIRA, 2000):

- ⤴ **Região Quente:** Contém o conjunto de dados que são utilizados mais frequentemente.
- ⤴ **Região Fria:** Contém o conjunto de dados que são utilizados com menos frequência.

As duas regiões podem variar dinamicamente, dependendo das consultas que serão geradas pelo sistema. Uma região fria pode passar a ser quente e vice-versa.

A complexidade e funcionalidade do *cache*⁶ são influenciadas pelas operações permitidas em um conjunto de dados. (DESPANDE et. al., 1998) divide em três categorias:

- ⤴ **Dados somente para consulta:** O *cache* não realiza atualizações locais sobre os dados, permitindo somente operações de consulta. Desta forma, os dados não necessitam ser reintegrados ao banco de dados e o *cache* pode ser estruturado de

⁶ O *cache* de banco de dados não deve ser confundido com *cache* de hardware, que é um dispositivo de memória ultra-rápida usada para agilizar a comunicação entre um processador e a memória primária do sistema. No entanto, em ambos casos o conceito de *cache* é o mesmo: aumentar a eficiência no acesso aos dados (FERREIRA & FINGER, 2000).

forma mais simples, sem a necessidade de controle de alteração de seu conteúdo. Este é o tipo mais conhecido de *cache*.

- ⤴ **Dados não concorrentes:** As atualizações locais são permitidas, mas os dados replicados ficam bloqueados no servidor, evitando atualizações conflitantes.
- ⤴ **Dados concorrentes:** Ocorre quando os dados replicados podem ser alterados tanto no servidor quanto no *cache*. Para permitir que possíveis conflitos de atualização sejam tratados no servidor, o *cache* deve armazenar informações sobre as alterações ocorridas nos seus dados.

Quando o *caching* não consegue responder a uma consulta realizada pela aplicação, a solicitação de dados ao servidor é necessária, podendo ser realizada de forma total ou parcial. (ARAÚJO & FERREIRA, 2000):

- ⤴ **Solicitação Total de Dados:** Acontece quando os dados são obtidos do *cache* local ou do servidor. Caso o *cache* contenha apenas parte dos dados solicitados, irá requerer todos os dados do servidor, tornando irrelevantes os dados que já estão em memória.
- ⤴ **Solicitação Parcial dos Dados:** Acontece quando os dados são obtidos parcialmente do *cache* ou parcialmente do servidor, com o objetivo de melhorar a utilização dos dados, permitindo que possam ser utilizados mesmo quando não respondam completamente a consulta.

5.2.2 Difusão de dados

Alguns sistemas de comunicação fazem a transmissão dos dados utilizando fios de cobre ou fibra ótica. Devido a necessidade de comunicação sem fio, dados são transmitidos pelo ar, não utilizando qualquer tipo de meio físico, como é o caso da transmissão por raios infravermelhos, *laser*, microondas, satélite, rádio entre outros.

O ar constitui-se de um meio natural para a propagação de sinais eletromagnéticos, pois provê uma interconexão completa, e permite uma grande flexibilidade na localização das estações. Contudo, existem alguns inconvenientes com relação a este sistema:

- ⤴ Os equipamentos de radiodifusão ainda são caros e complexos, devido aos níveis de potência de sinal na transmissão;
- ⤴ Utilização de técnicas analógicas de modulação e multiplexação;
- ⤴ Variedade de fontes de interferência e ruído;

Os sistemas dependem da regulamentação pública para a utilização do espectro de frequências. A implantação de novos serviços de comunicação têm sido limitado pela regulamentação pública, que permite o compartilhamento do ar.

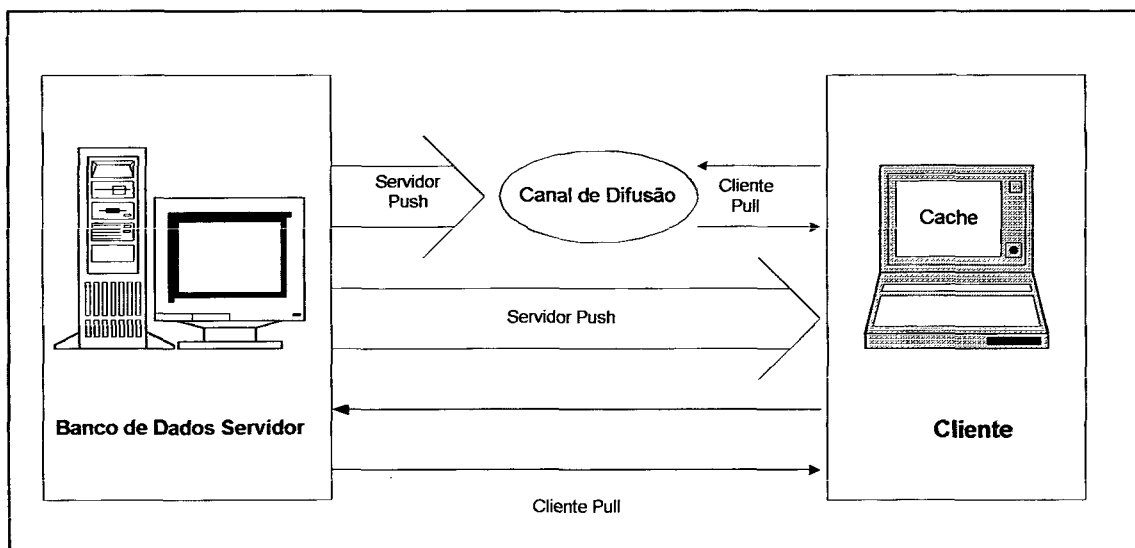
Uma maneira utilizada para disseminação de dados em ambientes móveis é a difusão de dados ou *broadcasting*, que consiste em um sistema de envio de mensagens, onde a mesma mensagem é enviada de uma estação de suporte móvel para todas as unidades móveis que estejam em uma célula.

Existem duas razões para a utilização da difusão de dados segundo (KORTH & SILBERSCHATZ, 1999). Primeiro, a unidade móvel evita o gasto de energia com a transmissão de solicitações de dados. Segundo, os dados difundidos podem ser recebidos por um grande número de unidades móveis de uma só vez, sem custos adicionais. Desse modo, a banda de transmissão tem utilização mais efetiva. As unidades móveis podem receber os dados quando eles forem transmitidos, em vez de consumir energia transmitindo uma solicitação, permitindo a disponibilidade local de armazenamento não-volátil para armazenar estes dados no momento de seu recebimento.

O servidor utiliza a largura de banda para difundir dados para múltiplos clientes. (MATEUS & LOUREIRO, 1998). Este método utiliza uma arquitetura chamada *pushed-based*, onde os dados são “empurrados” repetidamente do servidor para os clientes, como mostra a figura 22. Uma das limitações desta estratégia de entrega de mensagens é que as informações são entregues aos clientes de forma sequencial, fazendo com que o cliente tenha que monitorar o canal de comunicação, esperar até que os dados solicitados apareçam no canal de difusão e selecionar o item desejado. Por isso essa abordagem, é mais indicada nos casos em que a informação deve ser transmitida a um grande número de unidades móveis.

Outra estratégia de entrega de mensagens é denominada *pull-based*, que, ao contrário da *pushed-based*, faz com que o cliente exerça papel ativo, iniciando a transferência de dados pelo envio de pedidos ao servidor. A combinação destas duas estratégias é denominada de **entrega de dados híbrida**.

Figura 22 - Difusão de Dados



5.2.2.1 Alocação de canais para disseminação de dados

A alocação de canais consiste na distribuição de um conjunto de canais entre as estações base, com os seguintes objetivos:

- ✦ Alocar frequências para diversos pares transmissor/receptor com alta qualidade de serviço e baixa interferência;
- ✦ Alocar o maior número de antenas em uma mesma torre;
- ✦ Reduzir do espectro utilizado;
- ✦ Otimizar o uso de canais e ajustar a potência de transmissão.

A comunicação entre unidades móveis e sua respectiva estação base não é constante devido as freqüentes mudanças de lugar das mesmas. Com limitada largura de banda disponível na célula, há a necessidade de alocar canais de difusão para melhorar o

desempenho da comunicação global na célula. De acordo com [LEE et. al., 1999], existem quatro métodos de alocação de canais:

- ⤴ **Demanda exclusiva:** Todos os canais estão no modo de demanda, na qual todas as solicitações e resultados são entregues através de conexões ponto a ponto. Este método é indicado quando o número de consultas é pequeno comparado ao número de canais disponíveis e quando há energia suficiente para transmitir os pedidos.
- ⤴ **Difusão exclusiva:** Ao contrário do método anterior, todos os canais estão em estado de difusão. Os dados são disseminados periodicamente em canais de difusão. Este método é indicado quando um pequeno número de dados é de interesse a um grande grupo de usuários.
- ⤴ **Alocação Híbrida:** Este método mistura a difusão e a demanda exclusiva, utilizando o sistema de largura de banda de forma mais eficiente. A idéia principal é fazer a complementação dos dois métodos.
- ⤴ **Alocação dinâmica:** Este método aloca dinamicamente os canais de difusão e demanda para encontrar um desempenho ótimo para o acesso dos dados. Em contraste com o método híbrido, o método dinâmico aloca canais baseando-se em diferentes transmissões de dados para a estação base.

5.2.2.2 Comunicação Assimétrica

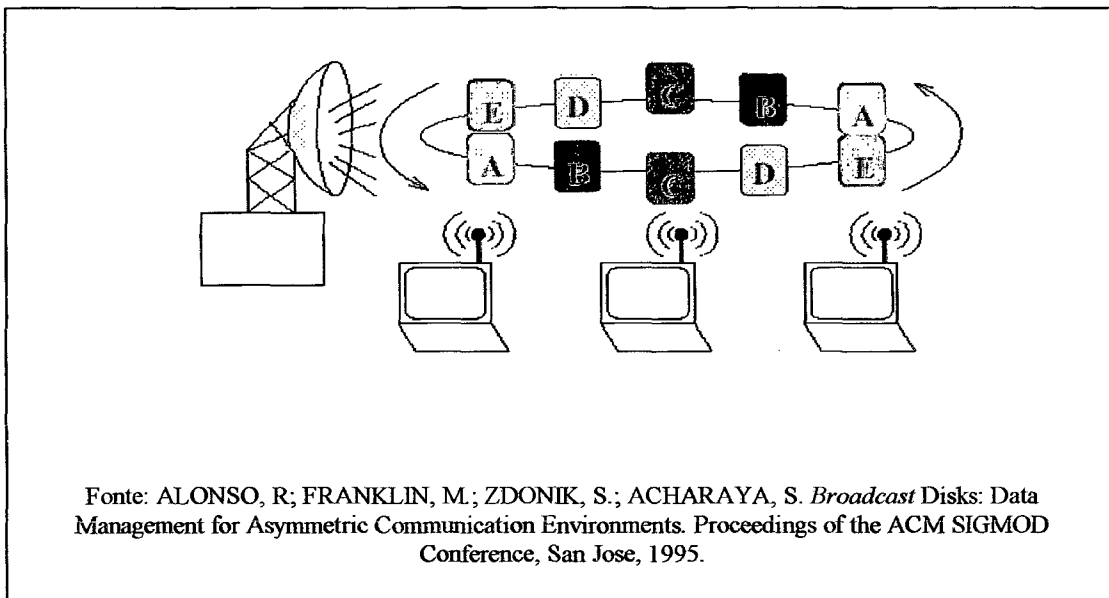
A comunicação assimétrica possui um ambiente onde a capacidade de comunicação dos servidores para clientes (*downstream*) é muito maior que a capacidade de comunicação de clientes para servidores (*upstream*). Um exemplo dessa situação ocorre quando o servidor tem a capacidade de transmitir dados em alta largura de banda e o cliente não consegue responder ou consegue fazê-lo apenas em baixo nível.

A comunicação assimétrica pode ser originada de duas formas, segundo (ALONSO et. al., 1995). A primeira é devido às limitações da largura de banda de um sistema de comunicação. Um exemplo seria a comunicação assimétrica entre cliente e servidor. A segunda é relacionada com o fluxo da informação na aplicação. Como exemplo, pode-se citar um sistema de recuperação de informação onde o número de clientes é bem maior que o número de servidores, gerando problemas de gerenciamento,

pois múltiplos clientes podem realizar pedidos simultâneos e o servidor pode não conseguir atendê-los.

Depois que o servidor tem a indicação dos dados que serão difundidos para cada cliente, ele faz a transmissão do conjunto de dados ciclicamente, como mostra a figura 23.

Figura 23 - Comunicação Assimétrica



Existem pelo menos três maneiras de organizar a difusão de dados. São elas: horizontal, *broadcast disks* ou discos de difusão e indexação. Na organização horizontal os dados são difundidos para os clientes via servidor. Este processo é realizado repetidamente até que o cliente selecione os dados que foram solicitados. A seguir são abordados os métodos *broadcast disks* e indexação.

Broadcast disks: É uma técnica para aumentar a hierarquia de memória de clientes em ambientes de comunicação assimétrico, difundindo dados de um servidor fixo para clientes móveis. Na arquitetura de *broadcast disks*, o servidor difunde dados para os clientes de maneira contínua e repetida, fazendo com que os canais de difusão se tornem discos de onde os clientes podem recuperar dados. Estes são carregados dentro dos discos de difusão de diferentes tamanhos e velocidades, baseando-se nas

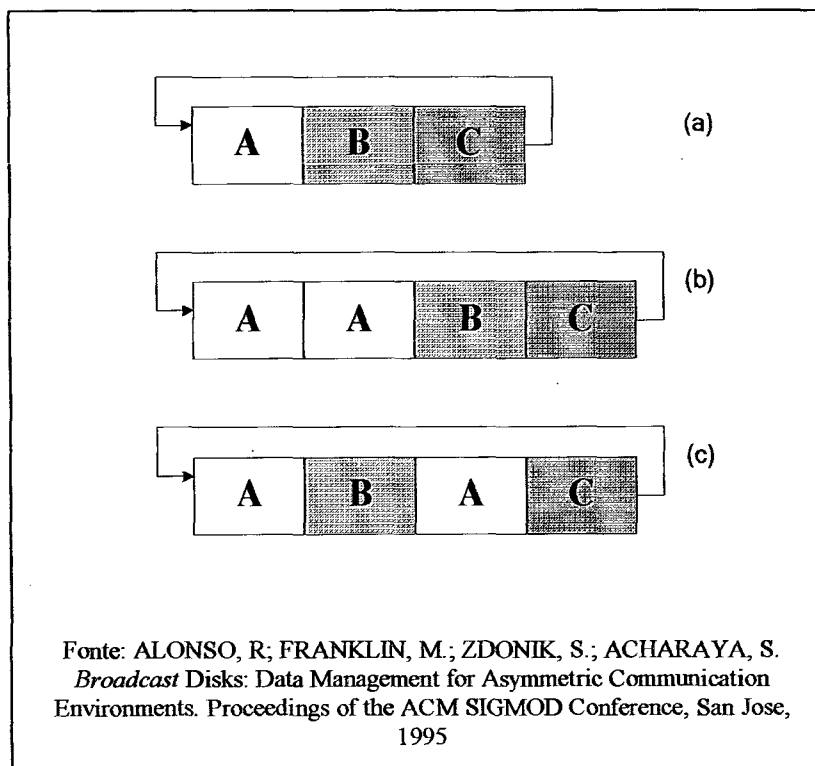
freqüências de acesso, sendo que dados que são transmitidos com a mesma freqüência, são considerados pertencentes a um mesmo disco.

Quando uma aplicação está sendo executada no cliente e precisa de um item de dado, primeiro ele tenta recuperar o dado da memória local. Caso não seja encontrado, então o cliente monitora a difusão e espera a chegada do dado.

Alternativamente, o servidor pode difundir vários itens com diferentes freqüências. Um programa de difusão pode enfatizar os dados mais solicitados enquanto pode dar menos prioridade para os menos solicitados. O servidor pode determinar um percentual ótimo de difusão de largura de banda que deve ser alocado para cada item de dado.

A figura 24 mostra três organizações diferentes de *broadcasts disks* contendo o mesmo tamanho mas utilizando estratégias diferentes.

Figura 24 - Exemplos de Discos de Difusão



Em 24 (a) é ilustrada a estrutura de uma difusão horizontal, também conhecida como estrutura plana. São considerados apenas os dados a serem transmitidos, independentes da probabilidade de acesso às informações.

A figura 24 (b) tem-se a difusão randômica, na qual difusões subseqüentes de A são armazenadas juntas, provavelmente de forma aleatória.

A figura 24 (c) exibe um tipo de difusão regular, não havendo variação no tempo de chegada de cada página (A,B ou C). É conhecida como difusão de multi-discos sendo que A está armazenado em um disco que tem duas vezes a velocidade do disco de B e C.

Indexação: (IMIELINSKI et. al., 1994) descreve uma situação em que o cliente está interessado em parte dos dados que serão transmitidos, podendo ser identificados por uma chave. Para que isto seja possível, junto com os dados deve-se enviar um índice ou diretório descrevendo a organização e a ordem das informações que serão difundidas, para que o cliente possa recuperar o item desejado. Desta forma, os clientes economizam energia, permanecendo no modo *doze* (adormecido), citado no capítulo 2, subcessão 2.10.3.1.

Para tanto, (IMIELINSKI et. al., 1994) sugere uma forma de difusão chamada indexação, onde todo índice é transmitido a cada fração ($1/m^7$) de dados enviados. Anexado a cada dado é transmitida a quantidade de itens que vão aparecer no canal antes de uma nova difusão do índice. Quando o cliente desejar acessar um registro, por exemplo, deverá sintonizar o canal de transmissão, descobrir o número de itens a serem transmitidos antes do índice, entrar no modo *doze*, sintonizar novamente no momento da difusão do índice, identificar quando o item desejado será transmitido, voltar para o modo *doze* e sintonizar novamente quando o dado é transmitido.

5.3 Gerenciamento de Transações Móveis

⁷ O valor ótimo de m é igual à raiz quadrada de d/i , onde d é o tamanho médio dos dados e i o tamanho do índice.

Para garantir a consistência de dados compartilhados propensos à falhas e desconexões, usuários de unidades móveis e fixas atualizam e recuperam dados através de transações.

Em um ambiente móvel uma transação não é totalmente gerenciada pelo sistema, sendo que o deslocamento de uma unidade é que controla a execução da transação (DUNHAM & KUMAR, 1998). Desta forma, ela é executada sequencialmente através de várias estações base e em possíveis conjuntos de dados, dependendo do movimento da unidade móvel. Em contrapartida, uma transação distribuída é completamente gerenciada pelo sistema sendo executada concorrentemente por múltiplos processadores e conjuntos de dados.

Devido à mobilidade as transações podem ser executadas de diversas localizações e conseqüentemente de diferentes servidores. Diferentemente de transações realizadas em uma rede fixa, as transações móveis podem acessar dados que estão em constantes mudanças de lugar, sendo assim dependente da localização dos dados.

A mobilidade resulta em transações que acessam sistemas de informações heterogêneas. A utilização de redes sem fio acarreta transações de longa duração, devido à sobrecarga de dados na rede, sendo mais propensas a erros, pelo fato das freqüentes desconexões e a duração das baterias. Podem ser baseadas em sessões para evitar transações demasiadamente demoradas.

Devido ao longo tempo da duração das atividades, vários refinamentos de transações são necessários. A idéia básica é tratar cada transação como um conjunto de subtransações.

Uma unidade móvel pode ter autonomia local para processar transações apesar das freqüentes desconexões. Na reconexão, os efeitos das transações móveis encerradas durante a desconexão deverão ser incorporadas ao banco de dados, garantindo desta forma, o sucesso da transação (WALBORN & CHRYSANTHIS, 1995).

Em sistemas de banco de dados tradicionais, usuários interagem com o banco de dados por transações atômicas, consistentes, isoladas e duráveis. Em aplicações móveis estas propriedades são consideradas restritivas, por serem limitadas, não suportando as desconexões e operações parciais (*commit* e *abort*).

5.3.1 Modelo de Transação Móvel

A figura 25 mostra o modelo genérico da arquitetura de banco de dados móveis. Neste modelo o servidor (suporta operações básicas como *commit*, *abort* e alocação do meio) e o banco de dados estão anexados em cada computador fixo.

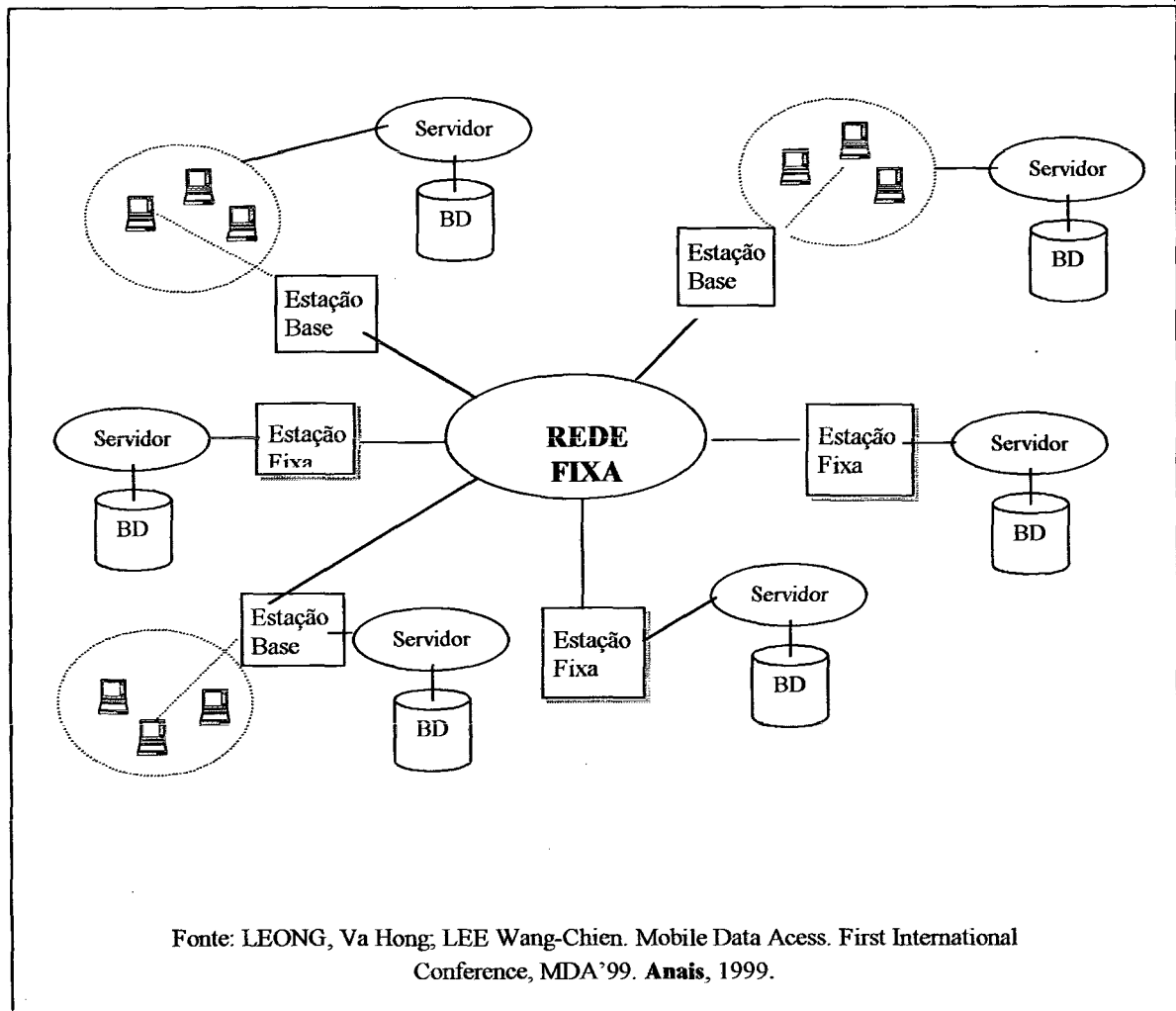
Cada estação base tem um coordenador que recebe as operações das transações solicitadas pelas unidades móveis e monitora sua execução no banco de dados servidor dentro da rede fixa. As transações são entregues pela unidade móvel ao coordenador em sua estação base, o qual as envia para o servidor de banco de dados distribuídos na rede fixa para a execução (ELMAGARMID & RUSINKIEWICZ, 1995).

Uma unidade móvel pode realizar dois tipos de transações:

- ⤴ A transação completa pode ser submetida em uma única mensagem de solicitação. A unidade móvel entrega o controle da execução para seu coordenador e aguarda o retorno dos resultados para a execução da transação;
- ⤴ As transações podem ser submetidas em múltiplas mensagens de solicitação. A unidade móvel interativamente submete as operações da transação para o seu coordenador e a operação subsequente só pode ser submetida após os resultados terem retornado do coordenador.

Enquanto a primeira abordagem envolve um único coordenador para todas as operações de uma transação, a segunda pode envolver múltiplos coordenadores devido à mobilidade da unidade móvel.

Figura 25 - Modelo de Transação de Banco de Dados Móveis



5.3.2 Modelos de Transações Móveis

A seguir serão abordados alguns modelos de transações móveis. São eles (DUNHAM & KUMAR, 1998):

- ⋈ Modelo *Kangaroo*;
- ⋈ Modelo de Fundamento Semântico;
- ⋈ Modelo de *Reporting* e Co-transações;
- ⋈ Modelo MDSTPM;
- ⋈ Transação *Proxy*.

Modelo *Kangaroo*: O modelo de transação *Kangaroo* é o único em redes móveis que inclui movimento em sua estratégia de execução, capturando o movimento natural de uma unidade móvel, suportando transações longas.

O modelo é construído assumindo um multi-banco de dados onde o gerenciamento da transação é desempenhado pela estação base. Quando a unidade móvel se movimenta, o gerenciamento da transação também se movimenta. Cada transação é dividida em subtransações, as quais serão executadas independentemente do sistema de banco de dados da rede fixa. Quando a solicitação de uma transação é feita por uma unidade móvel, a estação base associada cria uma transação móvel.

Modelo de Fundamento Semântico: Este modelo assume o processamento de uma transação móvel como um problema de conexão e concorrência, podendo ter longos atrasos no processamento de dados, bem como, desconexões. Esta abordagem utiliza a organização dos objetos para dividir grandes e complexos objetos em pequenos fragmentos. O servidor do banco de dados disponibiliza os fragmentos a pedido de uma unidade móvel. Para completar a transação, a unidade móvel retorna os fragmentos para o servidor.

Modelo de *Reporting* e Co-transações: Neste modelo, a origem das transações é representada em termos de *reporting* e co-transações, as quais podem ser executadas tanto na estação base como na unidade móvel. Uma transação *reporting* pode compartilhar seu resultado parcial com a transação de origem a qualquer hora e pode realizar a operação *commit* independentemente.

Uma co-transação é uma classe especial da transação anterior, na qual pode ser forçada a esperar por outra transação. Depois de entregar seu resultado, pode continuar a executar outras operações.

Modelo MDSTPM: Neste modelo, é pressuposta a existência de um MDSTPM (*Multidatabase Transaction Processing Manager*) em cada unidade móvel. Cada componente do sistema de banco de dados é responsável pelo gerenciamento das transações locais. Para facilitar a execução de transações globais, uma camada de *software* adicional pode ser implementada, permitindo o gerenciamento e coordenação

de transações através dos múltiplos componentes do sistema de gerenciamento (YEO & ZASLAVSKY,1994).

Transação Proxy: Uma transação *proxy* é utilizada para gerenciar a recuperação de dados. Pode ser considerada como uma subtransação da transação original. Alternativamente, podem ser executadas no modo *off-line*, quando o tráfego da rede ou a carga na estação base é baixa. Neste caso, transações *proxy* correspondem a *backups* periódicos do processamento desempenhado em uma unidade móvel .

5.3.3 Tipos de Mobilidade de uma Transação

Existem quatro tipos de mobilidade da transação definidos por (DUNHAM & KUMAR ,1998):

- ⤴ A transação é originada em uma unidade móvel, onde é executada completamente, não existindo qualquer tipo de movimento nesta operação.
- ⤴ A transação inteira completa sua execução localmente, para minimizar custos de comunicação e sobrecarga na rede. Qualquer dado solicitado que não está na unidade móvel é movido de outros *sites* (unidade móvel, estação base ou qualquer outro nó da rede fixa), sendo que, durante sua execução, a unidade móvel pode se mover para uma célula estrangeira. Este tipo de execução é chamada de **local atômica**.
- ⤴ Uma transação pode saltar por um número de células estrangeiras durante seu tempo de execução. Este processo é caracterizado como **migração de transação**. Uma unidade móvel pode entrar em uma nova célula enquanto está em operação. Neste caso, pode ser necessário migrar parte do processamento que foi executado em uma unidade fixa para outra unidade fixa. Uma motivação para a migração é a melhora de desempenho. Além disso, movendo o processamento finalizado para uma unidade móvel, custos de comunicação podem ser minimizados. A migração de uma transação pode ser feita através de dupla relocação de dados, balanceando o carregamento entre estações base, de modo que, uma transação executa parcialmente em uma estação base e posteriormente, transfere os dados processados para a nova estação base.

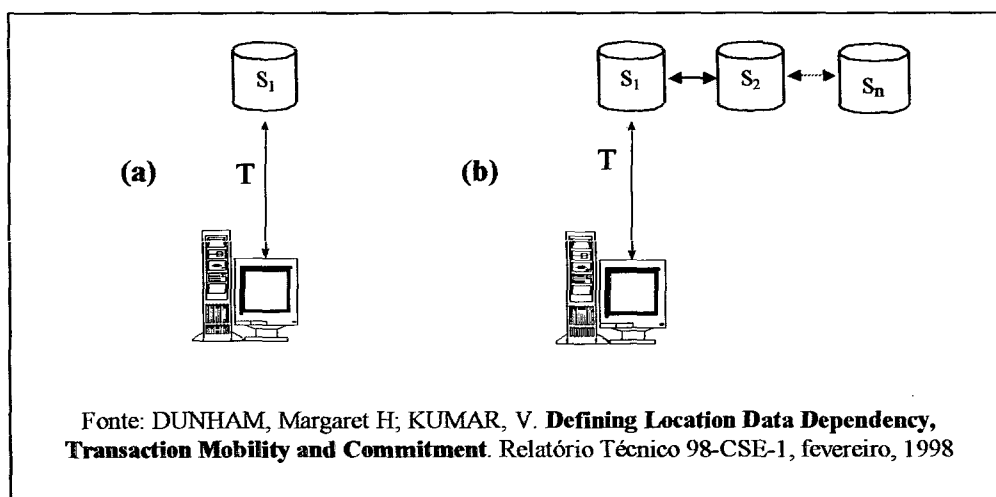
- ^ Uma transação móvel é definida como um conjunto de fragmentos, que podem “visitar” apenas um subconjunto pré-definido de unidades móveis. Este tipo de restrição de movimento é dependente da distribuição do banco de dados entre as unidades móveis.

5.3.4 Diferentes Plataformas de Execução de uma Transação Móvel

Em um ambiente centralizado, a transação é solicitada, executada e o resultado é mostrado no mesmo *site*, como mostra a figura 26 (a).

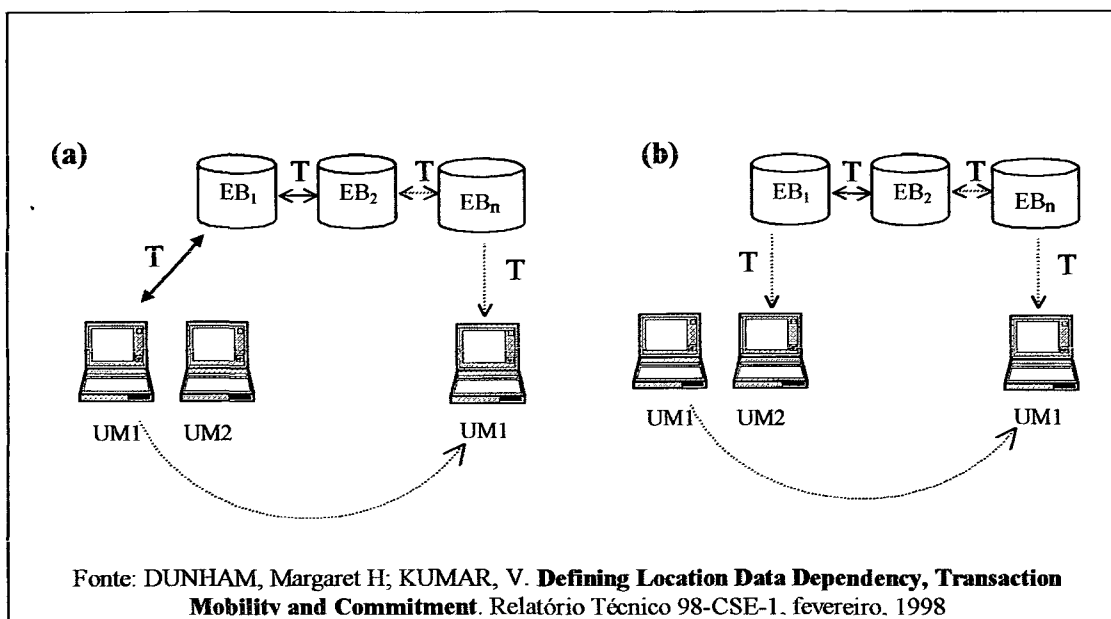
Se a transação for executada em um ambiente distribuído ela será dividida em fragmentos e distribuída em diversos *sites* para a realização da sua execução, figura 26 (b).

Figura 26 - Execução de uma transação centralizada e distribuída



Na figura 27 (a) a UM1 (unidade de processamento) é mostrada movendo-se para uma nova localização geográfica na qual é gerenciada pela estação base (EB_n) juntamente com a execução dos fragmentos.

Figura 27 - Execução de uma Transação em Fragmentos



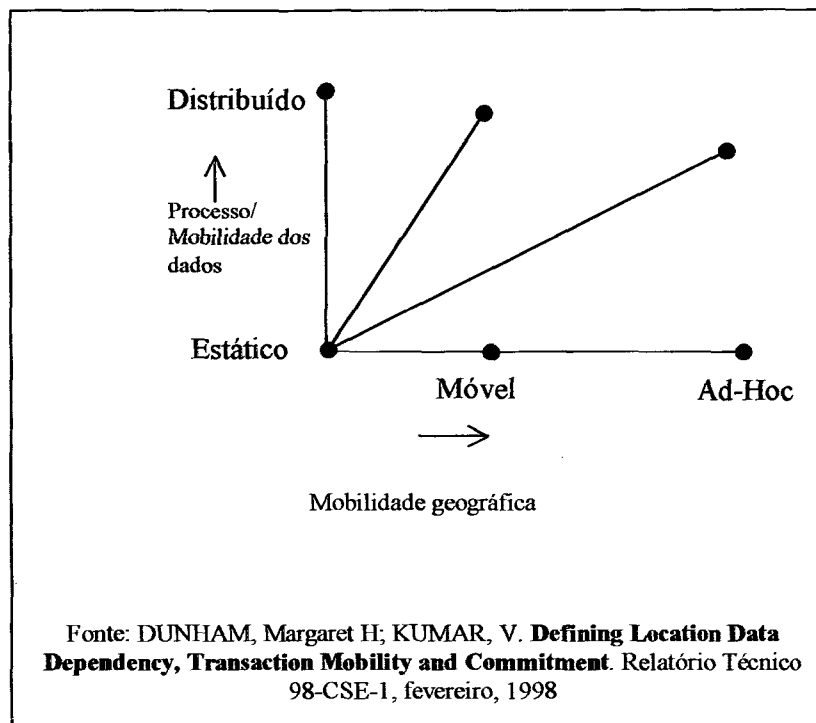
Já na figura 27 (b), apenas a UM_1 se move para a EB_n deixando a execução dos fragmentos. Isto pode acontecer se T tem que ser executada na célula que a originou porque a EB tem dependência de localização dos dados que a transação necessita.

5.3.5 Graus de Mobilidade de uma Transação

De acordo com (DUNHAM & KUMAR, 1998), existem seis tipos ou graus de mobilidade. Estas categorias podem ser aplicadas em uma arquitetura ou transação. O grau mais alto suportado por uma arquitetura é o grau permitido por qualquer T executada nesta arquitetura.

A classificação é dividida em dois eixos (figura 28). O eixo horizontal representa a mobilidade geográfica e o eixo vertical representa a mobilidade dos dados.

Figura 28 - Graus de Mobilidade de uma Transação



O grau 0 (estático) assume que os processadores não se movem. O grau 1 (distribuído) assume que as unidades se movimentam. O grau 2 (*Ad-Hoc*) assume que qualquer nó pode se mover e comunicar diretamente. A seguir são descritos os seis graus de mobilidade:

- ⤴ (0,0) – Ambiente centralizado com um único processador. Não há movimento geográfico.
- ⤴ (0,1) – Ambiente distribuído tradicional. O movimento dos processadores e dos dados é permitido mas não há movimento geográfico.
- ⤴ (1,0) – Esta categoria permite o deslocamento da unidade móvel, mas não há movimento dos dados nem dos processadores.
- ⤴ (1,1) – Tradicional ambiente da computação móvel. As unidades móveis se movem mas não podem se comunicar diretamente uma com a outra.
- ⤴ (2,0) – Neste caso qualquer nó pode se comunicar, mas não pode haver movimento dos dados ou processadores.

- ⤴ (2,1) – Tipo de mobilidade genérica, onde se movimentam dados e processadores.

5.3.6 Requisitos de uma Transação Móvel

Para descrever o processamento de transações em uma unidade móvel, é viável a união das propriedades ACID e os conceitos de visibilidade, consistência, permanência e recuperação (WALBORN & CHRYSANTHIS, 1998).

A visibilidade refere-se a habilidade da transação para ver os efeitos causados nos dados por outra transação. Para reduzir os custos de recuperação, os efeitos da transação não estão visíveis até que a transação termine e as trocas sejam feitas permanentemente no banco de dados. Permitindo as novas transações a visão das operações não terminadas, o resultado pode gerar dependências de dados e encerramentos não desejados. Desde que, as atualizações de uma unidade móvel desconectada não sejam feitas no servidor, as transações subsequentes usando os mesmos dados normalmente não podem continuar até que a conexão ocorra e a transação termine.

Um sistema de gerenciamento de banco de dados que suporte o processamento de transações em ambiente móvel desconectado deve incluir, segundo (WALBORN & CHRYSANTHIS, 1998):

- ⤴ Uma forma de comunicação com o banco de dados servidor;
- ⤴ Um subsistema de processamento da transação para executar na unidade móvel;
- ⤴ Um método para reintegrar transações processadas enquanto desconectadas com o banco de dados servidor;
- ⤴ *Logging* suficiente, *checkpoint* e sistema de recuperação para aliviar falhas do sistema;
- ⤴ Um método para gerenciar a replicação e a consistência de dados.

5.4 Processamento de Consultas Móveis

O processamento de consultas em um ambiente móvel é diferenciado de um ambiente centralizado. No ambiente móvel, fatores como largura de banda limitada,

consumo de energia, mobilidade, entre outros, são determinantes para que a consulta seja realizada. No ambiente centralizado, os acessos à memória principal e a disco são os principais critérios para a otimização das consultas.

Para que uma consulta seja realizada em banco de dados móveis, é necessário conhecer a localização precisa da unidade móvel. O uso de computadores em navios, aviões, carros entre outros meios de transporte, dificulta esta localização, devido as constantes mudanças de lugar. Consultas ao posto de gasolina mais próximo, hotéis, restaurantes, hospitais e outros tipos de serviços necessários enquanto se está em movimento, são exemplos de consultas móveis.

Em um banco de dados móveis, quando um usuário faz uma consulta, diversos fatores contribuem no tempo de resposta do sistema, tais como (NASSU & FINGER, 2000):

- ⤴ O tempo de transmissão de dados pode ser alto. A baixa taxa de transmissão dos meios sem fio, a baixa confiabilidade, que pode gerar retransmissões frequentes, influem nesse tempo.
- ⤴ A execução da consulta ao banco de dados pode ser demorada, principalmente no caso em que as cidades são grandes, com grande número de ruas e informações. O grande uso de forma concorrente também pode retardar o tempo de resposta do banco de dados.
- ⤴ O usuário pode estar desconectado da rede, por vontade própria ou não. O tempo de desconexão pode ser longo e a resposta pode demorar mais para chegar ao usuário que a solicitou.

A localização dos dados no processamento de consultas pode incluir dados transientes e dados contínuos. Os dados transientes trocam de valores de acordo com o processamento das consultas. Dados contínuos são aqueles que são continuamente atualizados durante o movimento da unidade móvel.

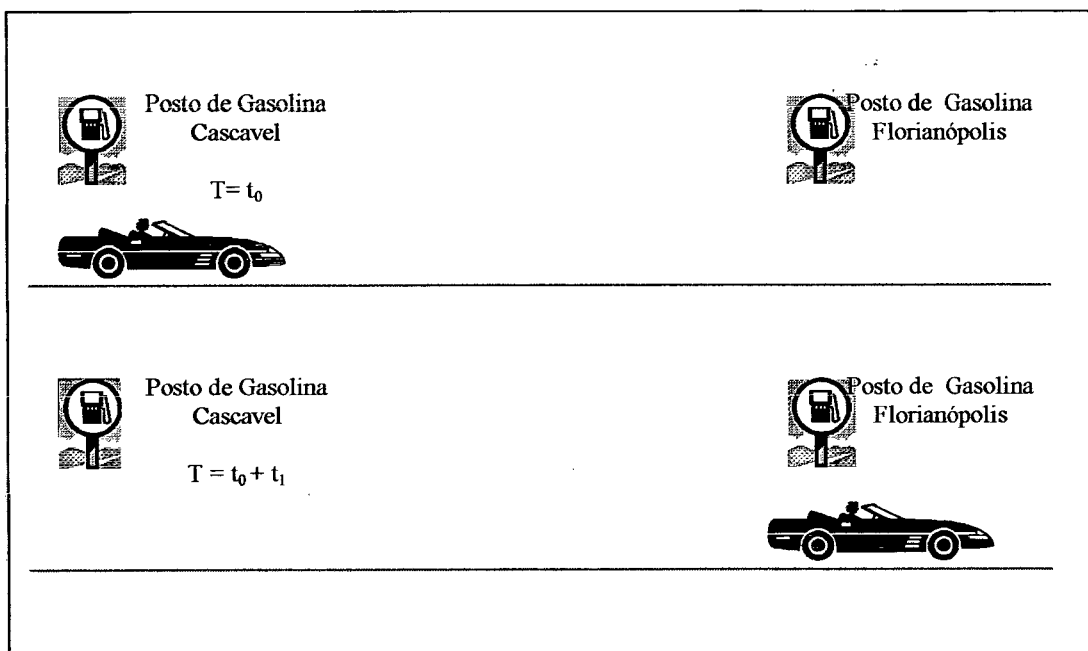
5.4.1 Dependência de Localização de Dados

Em sistemas de gerenciamento de banco de dados tradicionais a localização geográfica dos dados e a localização na qual os dados são processados normalmente não possuem nenhuma relação entre si. Em computação móvel, uma unidade móvel está em constantes mudanças de lugar. A localização⁸ depende dos dados existentes e o conjunto de dados e seus valores são ligados com uma localização particular, onde respostas para consultas dependem de onde são originadas.

A dependência de localização contém dados cujos valores são determinados pelo domínio geográfico⁹ das unidades móveis ou estações base onde as consultas são realizadas.

Como exemplo, pode-se citar uma pessoa viajando de Cascavel a Florianópolis, como mostra a figura 29.

Figura 29 - Exemplo de Dependência de Localização de Dados



⁸ Localização: é um ponto preciso dentro de um domínio geográfico. Ela representa a menor posição identificável do domínio. Pode ser representada pela coordenada latitude/longitude.

⁹ Domínio Geográfico: É a área coberta por uma plataforma de computação móvel, onde uma estação móvel pode se mover livremente.

Se esta pessoa fizer uma consulta do posto de gasolina mais próximo de onde está, a resposta dependerá da coordenada geográfica de seu computador portátil, ou seja, dependerá da localização de onde foi feita a consulta. Se a mesma consulta fosse realizada várias vezes, a partir de lugares diferentes, várias respostas diferentes seriam obtidas.

Caso a mesma pessoa fizesse uma consulta para saber qual o posto de gasolina mais barato para abastecer a resposta dependeria das características do lugar onde o posto de gasolina é localizado, pois em uma cidade a mesma gasolina pode possuir valores diferentes. O valor da gasolina em Cascavel pode ser diferente se comparada ao valor de Florianópolis, devido a taxas, custos locais, etc. Neste exemplo, tem-se uma **localização de dados dependente**.

Se a consulta fosse realizada em Cascavel e o sistema demorasse algum tempo para responder, ao receber a resposta, o usuário poderia ter passado do posto mais próximo que o sistema tenha indicado. Neste caso, seria desejável que ele recebesse a resposta a tempo de ir ao local mais próximo e atual. A posição do usuário após alguns momentos é dependente de muitos fatores, dentre eles (NASSU & FINGER, 2000):

- ⤴ Velocidade;
- ⤴ Aceleração;
- ⤴ Tipo do movimento, como uniforme, uniformemente variado, etc;
- ⤴ Tempo que a transação leva para ser executada;
- ⤴ Direção, sentido, trajetória, ou seja, que tipo de movimento o agente está executando, como por exemplo, linear, circular, etc.
- ⤴ Tempo de processamento de consulta, demora na transmissão de dados, desconexões ou quedas de sinal.

Outro exemplo, seria o de uma consulta realizada para saber a data de nascimento de uma pessoa em Cascavel ou em Florianópolis. A resposta obtida seria idêntica para as duas consultas. Neste caso, tem-se a **localização de dados livre**, sem dependência de localização.

5.4.1.1 Consistência em Dependência de Localização de Dados

Um banco de dados está em um estado mutuamente consistente se todas as cópias têm exatamente o mesmo conteúdo e se todas as restrições de integridade identificadas para o banco de dados são obedecidas.

A consistência mútua é usada para indicar que todos os dados convergem para um mesmo valor. Existem dois tipos de consistência relacionadas à dependência de localização, consistência temporal e consistência espacial.

- ⤴ **Consistência Temporal:** indica que todos os valores dados podem satisfazer a um dado conjunto de restrições de integridade, independentemente da localização.
- ⤴ **Consistência Espacial:** indica que todos os valores dos dados de uma replicação espacial são associados à apenas uma região de dados e são satisfeitas restrições de integridade definidas pela região.

5.4.2 Gerenciamento de Localização

O gerenciamento da localização das unidades móveis é um dos temas de grande pesquisa em gerenciamento de dados. A administração da mobilidade trata de localizar, armazenar, manter e recuperar informação de unidades móveis.

Foram propostas várias estratégias para reduzir custos e sobrecarga de dados na rede. O desempenho de várias propostas de gerenciamento de localização dos usuários móveis depende de diversos fatores como mobilidade, números de usuários, características da rede e a topologia do banco de dados utilizado.

A localização é tratada como um fragmento de dados que pode ser consultado e atualizado, onde a informação da mesma é mantida apenas por unidades que estão ativas.

Um sistema pode ser fisicamente móvel, mas logicamente estático, ou quase estático. Isto acontece quando apesar de todos os componentes serem móveis, suas relativas posições são conhecidas todo o tempo. Por exemplo, quando dois usuários viajam de avião, apesar de estarem em movimento, suas posições relativas são

conhecidas e não se alteram durante o período de vôo. Neste caso, o sistema tem uma informação completa sobre a localização do usuário (BADRINATH & IMIELINSKI, 1992).

Como exemplo, suponha que a unidade X deseja saber a localização da unidade Y, em uma arquitetura celular. Nesta arquitetura, quando o usuário se move, ele deve informar ao servidor HLR (*Home Location Register*) sobre seu novo endereço. Para enviar uma mensagem ou chamada a cada usuário, o servidor deve ser contactado para obter seu endereço atual.

A maior desvantagem deste esquema é que as unidades móveis não podem sair de suas respectivas áreas geográficas de origem, realizando apenas movimentos dentro de sua HLR .

5.5 Replicação de Dados

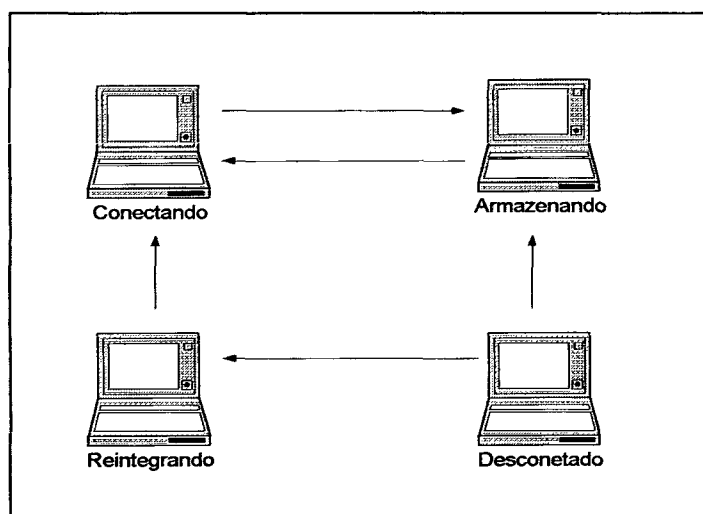
Em termos técnicos, a replicação é o processo no qual transações executadas em um banco de dados são propagadas assincronamente para um ou mais banco de dados de forma serial. A serialização significa que as transações, assim como todas as operações, são replicadas na mesma ordem na qual elas foram solicitadas. Caso isto não aconteça, podem ocorrer inconsistências entre a unidade móvel e o servidor. A propagação de forma assíncrona significa uma forma de armazenar e enviar replicação, ou seja, as operações que serão replicadas do servidor podem ser armazenadas em um banco de dados fonte até que sejam propagadas para as unidades móveis, no momento da sincronização, pois algumas vezes a conexão se torna impossível (DBMS ONLINE, 1997).

Em um ambiente móvel é importante que estações móveis acessem o banco de dados de maneira que a comunicação seja rápida e reduzida. A replicação é necessária para sincronizar os dados e as operações no banco de dados. Usuários móveis necessitam da replicação de dados para armazenar dados utilizados mais frequentemente durante o período de desconexão fraca ou total. O item é copiado para a estação móvel, onde a leitura é realizada localmente, sem a necessidade de comunicação da estação móvel com o servidor.

A replicação completa é melhor que o método *caching*, pois, utiliza a funcionalidade completa nos dados do computador portátil e os dados são generalizados para mais de um usuário. Com a replicação, se pelo menos uma de suas cópias estiver disponível, a confiabilidade aumenta, uma vez que o usuário não depende apenas dos dados disponibilizados em uma única localidade. No caso de acontecer algum problema no sistema, a réplica mais próxima dos dados é solicitada pelo usuário.

Em um sistema de arquivos replicados desconectado, o cliente armazena conjuntos de arquivos específicos em diretórios locais, podendo desconectar e realizar atualizações localmente, ou seja, o cliente armazena dados e trabalha localmente nos dados em um modo desconectado. Quando o cliente fizer a reconexão, todos os arquivos atualizados serão reintegrados, como mostra a figura 30.

Figura 30 - Reintegração de Dados



5.5.1 Graus de Replicação

A replicação de dados pode ser dividido em três graus: replicação total do banco de dados, inexistência de replicação do banco de dados e replicação parcial do banco de dados (BADRINATH & PHATAK, 1998).

- ⤴ **Replicação Total do Banco de Dados:** O banco de dados tem todos os seus dados replicados em todos os seus *sites*. A vantagem desta operação é que a disponibilidade de dados aumenta consideravelmente, na medida em que o sistema pode operar enquanto pelo menos um *site* estiver funcionando, sendo que as consultas são feitas localmente em cada *site*. Por outro lado, isto pode afetar as operações de atualizações, uma vez que estas devem ser feitas em cada um dos *sites* que contém uma cópia do banco de dados, para evitar inconsistências. Além disto, torna as operações de controle de concorrência e técnicas de recuperação de dados mais pesadas do que se não houvesse replicação.
- ⤴ **Inexistência de Replicação no Banco de Dados:** Cada fragmento está armazenado em exatamente um *site*, ou seja, em apenas uma posição. Neste caso, todos fragmentos devem sofrer uma disjunção, exceto pela repetição de chaves primárias entre fragmentos verticais (ou mistos). Este esquema também é chamado de **alocação não-redundante**.
- ⤴ **Replicação Parcial do Banco de Dados:** Apenas alguns objetos de dados ou fragmentos de dados são replicados. O número de objetos ou fragmentos a serem replicados e os seus números de réplicas é variável (de acordo com as solicitações dos usuários).

A escolha do melhor tipo/grau de replicação para um sistema é bastante difícil e depende das metas principais a que ele se propõe, das principais transações a que ele é submetido e a localização dessas transações.

Dados nos quais a frequência das alterações é muito alta, a replicação de apenas alguns objetos nas posições onde eles são mais solicitados, ou a inexistência de replicação, são as duas opções mais recomendadas.

Em sistemas que exigem alta performance, os graus de replicação devem ser muito bem estudados para uma escolha que não leve a resultados indesejados, como tempos de resposta ou atualização muito longos.

5.5.2 Tipos de replicação

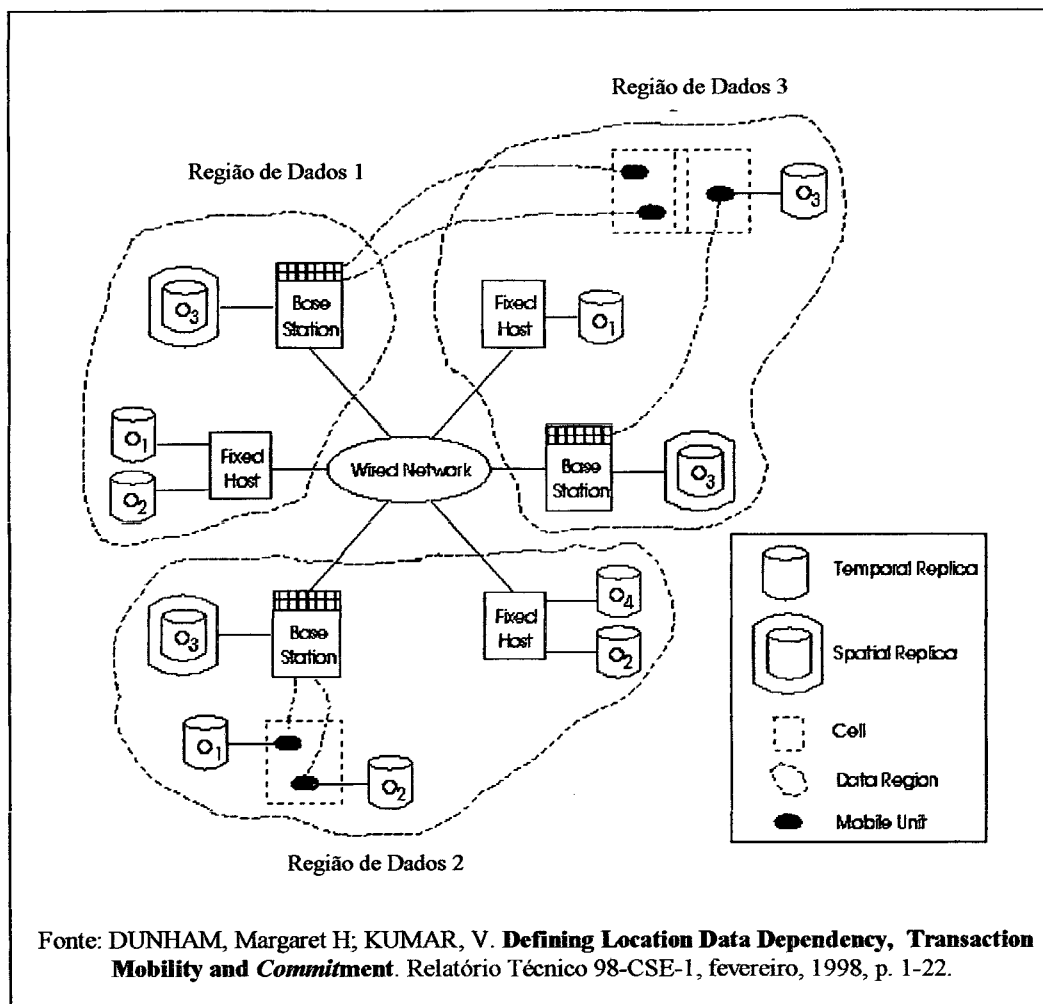
Existem três tipos básicos de replicação (DUNHAM & KUMAR, 1998):

- ⤴ Não replicação de dados, sendo estes concentrados em um único local;
- ⤴ Replicação Temporal;
- ⤴ Replicação Espacial;

A replicação temporal é a tradicional replicação distribuída, onde réplicas de dados podem temporariamente ter valores diferentes com a responsabilidade de implementar um controle de réplicas para manter a consistência mútua.

A replicação espacial é a replicação utilizada na computação móvel, onde múltiplas cópias de dados podem ter diferentes valores corretos a qualquer momento. O valor correto é determinado pela localização atual da unidade móvel. Diferente de réplicas temporais, não há a necessidade de que réplicas espaciais de um objeto tenham o mesmo valor.

Figura 31 - Tipos de Replicação de Dados



A figura 31 ilustra os três tipos de replicação distribuídas em três regiões de dados¹⁰ diferentes.

- 1) A não replicação acontece na região 2, no objeto O₄.
- 2) A replicação temporal acontece nas regiões 2 e 1, representada pelos objetos O₁ e O₂.
- 3) A replicação espacial acontece em todas as regiões representadas pelo objeto O₃. Cada região tem um valor correto para O₃, podendo ter réplicas temporais nesta

¹⁰ Uma região de dados é uma área na qual existe um valor correto para uma réplica espacial.

região, como mostra a região 2. O valor do objeto O_3 da região 3 é replicado da estação base para uma unidade móvel, ambas contendo o mesmo valor.

Na tabela 5 temos as principais diferenças entre os tipos de replicação de dados:

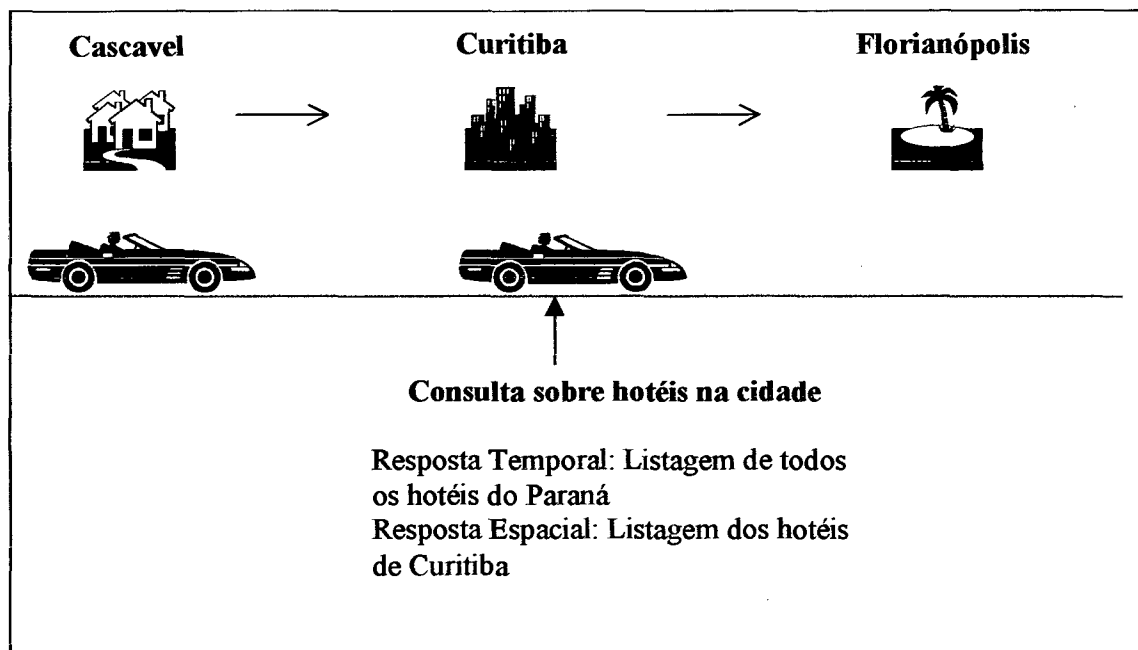
Tabela 5 - Tipos de Replicação

	Não Replicação	Replicação Temporal	Replicação Espacial
Cópias	Uma	Múltiplas	Múltiplas
Valores Corretos	Um	Uma por vez	Uma por localização e por vez
Arquitetura	Centralizada	Distribuída	Móvel
Mobilidade	Não tem	Não tem	Tem

Fonte: DUNHAM, Margaret H; KUMAR, V. **Defining Location Data Dependency, Transaction Mobility and Commitment**. Relatório Técnico 98-CSE-1, fevereiro, 1998, p. 1-22.

Como exemplo de tipos de replicação de dados pode-se citar uma pessoa que está viajando de carro de Cascavel a Florianópolis, como mostra a figura 32. Em determinado momento da viagem ela pára para descansar em um hotel na cidade de Curitiba. Realiza uma consulta com a listagem de todos os hotéis para hospedagem na cidade.

Figura 32 - Exemplo de Tipos de Replicação



Caso a consulta fosse realizada em um ambiente distribuído tradicional com réplicas temporais, a resposta seria os nomes de todos os hotéis do domínio geográfico completo, ou seja, uma listagem de todos os hotéis do Paraná, por exemplo.

Caso a consulta fosse realizada com réplicas espaciais, seriam listados apenas os nomes dos hotéis situados na região correspondente, ou seja, em Curitiba.

A replicação temporal realiza cópias mútuas, as quais podem ter valores diferentes temporariamente, mas apenas um valor é correto. A replicação espacial tem múltiplas cópias e múltiplos valores corretos, determinados pela localização

5.5.3 Modelos de Replicação Otimista de Dados

A replicação otimista tem como objetivo fazer que com todas as suas réplicas fiquem disponíveis para uso completo dos usuários, inclusive para a atualização, bem como o acesso de dados compartilhados, perante uma conexão variável, largura de banda e latência.

A abordagem “otimista” significa que o modelo permite que réplicas sejam atualizadas independentemente e detectando atualizações concorrentes quando existirem.

Existem três modelos de replicação otimista: *peer-to-peer*, cliente servidor e o modelo WARD (RATNER, 1998).

5.5.4 Modelo de Replicação *Peer-to-peer*

Neste modelo todas as réplicas são iguais e existe a possibilidade de comunicação de estação para estação. Permite a qualquer réplica de um dado trocar e atualizar informações com qualquer outra réplica, desde que estejam conectadas.

Este modelo é mais adequado quando há uma forte conexão entre computadores portáteis e existe pouca ou nenhuma conexão com o servidor, fazendo com que a comunicação seja realizada mais facilmente e se torne mais acessível do que a conseguida com o servidor. Uma outra vantagem do *peer-to-peer* é que a comunicação pode ser realizada com a estação que estiver disponível. Uma desvantagem deste modelo é a escalabilidade, só possível realizar pequenos números de réplicas.

O modelo *Bayou* (PITOURA & SAMARAS, 1998) é um exemplo de sistema de banco de dados replicado que utiliza o modelo *peer-to-peer*, com computadores portáteis fracamente conectados entre si. Neste modelo, cada conjunto de dados é replicado completamente em vários servidores.

As aplicações executadas nos clientes interagem com os servidores através de uma interface denominada API (*Application Programming Interface*), que suporta duas operações básicas: leitura e escrita. As operações de leitura permitem a consulta dos dados, enquanto que as operações de escrita permitem modificar, inserir e apagar dados. Clientes podem ler e escrever em qualquer cópia do servidor, com os quais eles podem se comunicar.

O modelo *peer-to-peer* é necessário em uma grande variedade de ambientes, porém, oferece menos flexibilidade do que o modelo cliente-servidor. (RATNER et. al, 1997).

5.5.5 Modelo de replicação Cliente/Servidor

No modelo de replicação do modelo cliente/servidor, todas as estações contém o mesmo conjunto de dados, possuindo um controle superior para gerenciar a replicação dos dados. Os clientes armazenam apenas os dados mais relevantes. A atualização dos dados é realizada diretamente do servidor para todos dos clientes, não permitindo a troca de informações entre clientes.

CODA (PITOURA & SAMARAS, 1998) é um exemplo de sistema de arquivo replicado que utiliza o modelo cliente/servidor. Os clientes não podem se comunicar diretamente devido às restrições do modelo utilizado. CODA oferece maior largura de banda, possibilitando uma comunicação otimizada e sincronizada, especialmente em ambientes com conexão fraca.

5.5.6 Modelo WARD

O modelo WARD (*Wide Area Replication Domain*) é a combinação do modelo *peer-to-peer* e do modelo cliente/servidor (RATNER et. al, 1998). É denominada como uma arquitetura híbrida, com mais flexibilidade de replicação, permitindo reconfiguração dinâmica e topologia de sincronização. É utilizada para o gerenciamento de réplicas e atualizações distribuídas.

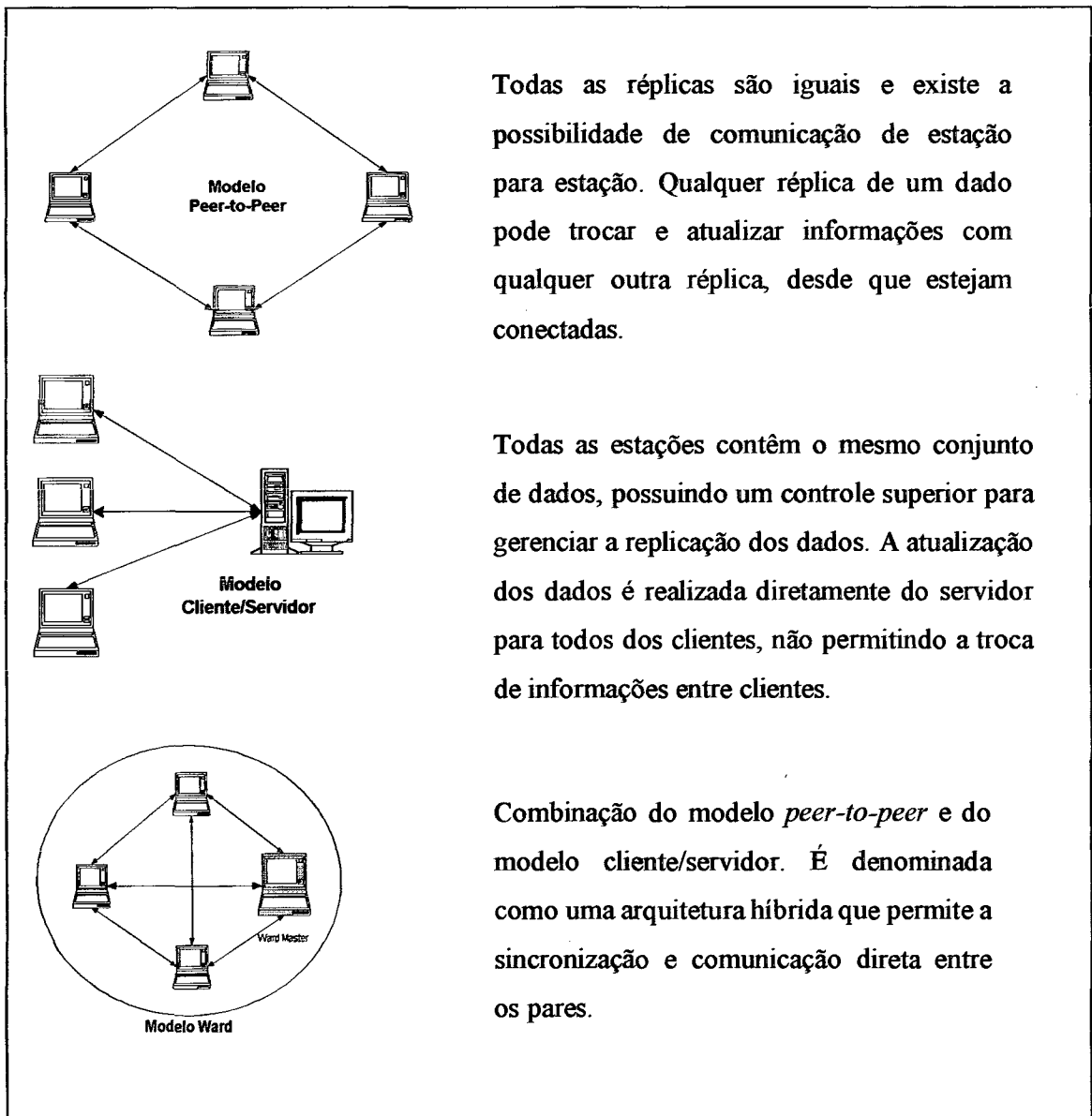
O modelo pode agrupar mecanismo de WARD que é um conjunto de estações vizinhas conectadas em uma mesma área geográfica. Parâmetros como localização geográfica, largura de banda, custos e modelos de compartilhamento devem ser considerados para formar WARDs. Todos os WARDs são pares iguais, com exceção do WARD *master*, que possui características semelhantes ao servidor do modelo cliente/servidor. Como todos os membros são pares, qualquer um deles pode ser designado como WARD *master*, podendo realizar sincronização direta entre seus membros.

O modelo WARD pode ser aplicado em ambientes de grande escala de estações que compartilham características comuns, como IP móvel, protocolos de roteamento de rede e segurança na Internet. A escalabilidade deste modelo é relacionada com o grau de flexibilidade da replicação.

O ROAM (RATNER et. al, 1998) é um sistema de replicação para computação móvel baseado no modelo WARD, que mantém o modelo *peer-to-peer* em todas as réplicas, permitindo a sincronização e comunicação direta entre elas.

A figura 33 faz uma comparação entre os três modelos citados.

Figura 33 - Modelos de Replicação Otimista



5.6 Recuperação de Falhas

Uma das grandes diferenças entre um ambiente móvel e um ambiente centralizado está na capacidade do primeiro oferecer mobilidade e recuperação de dados. O processo de recuperação é responsável por preservar a consistência do banco de dados após falhas do sistema, de transações ou dos meios de comunicação.

Um sistema em ambiente móvel está sujeito à falhas tanto de *hardware* como de *software*. Em cada um destes casos, informações que se referem ao sistema de banco de dados podem ser perdidos. Uma parte essencial do sistema de banco de dados é um esquema de recuperação responsável pela detecção de falhas e pela restauração do banco de dados para um estado consistente que existia antes da ocorrência da falha.

Para que o sistema não seja prejudicado devido às falhas em seus componentes, erros devem ser detectados o mais rápido possível, através de um diagnóstico apropriado. Para recuperar os dados, informações relevantes são armazenadas em um local fixo durante o processamento de transações.

Em sistemas distribuídos a recuperação de falhas é baseada em pontos de recuperação, conhecidos como *checkpoints*. No caso de falhas e desconexões, a aplicação usa o último *checkpoint* salvo para reiniciar sua execução (PITOURA & SAMARAS, 1998).

A unidade móvel deve estar sempre informada sobre qual célula se encontra e quando o sistema entrará em modo de desconexão, pois ela é responsável por gerar um *checkpoint* caso mude de célula ou haja uma desconexão.

Um *checkpoint* global consiste de um conjunto de *checkpoints* locais. Os estados globais são armazenados e usados pelo protocolo para recuperar falhas das aplicações. Um estado global inclui o estado de cada processo que faz parte da aplicação distribuída e possivelmente algumas mensagens. Para uma recuperação correta, o protocolo pode salvar um estado consistente global.

Os protocolos *checkpoint* são classificados em duas categorias: protocolos coordenados e protocolos não coordenados (NEVES & FUCHS, 1997).

Protocolos Coordenados: A criação dos *checkpoints* é realizada em modo conectado, para garantir a consistência e recuperabilidade do *checkpoint* global. O protocolo requer o envio de mensagens para diferentes unidades móveis, para que ocorra a sincronização no processo de *checkpoint*. Alguns problemas podem ocorrer durante o envio das mensagens, como por exemplo:

- ⤴ Desconexões;
- ⤴ Dificuldade para localizar uma unidade móvel para o envio das mensagens;
- ⤴ As unidades móveis podem se mover para outra célula antes que o processo de *checkpoint* seja concluído.

Protocolos Não Coordenados: Permitem a criação de *checkpoints* independentes, sem a necessidade de comunicação, desse modo, o processo pode continuar durante as desconexões. No entanto, mensagens precisam ser trocadas durante a recuperação para localizar o *checkpoint* global, exigindo a coleta de informações sobre outras unidades móveis. Durante a recuperação, uma coordenada é selecionada para escolher um *checkpoint* de cada unidade para criar um *checkpoint* global consistente.

Devido às características como mobilidade, baixa largura de banda e desconexões, protocolos coordenados e não coordenados são considerados inadequados para o ambiente móvel (PITOURA & SAMARAS, 1998).

(NEVES & FUCHS, 1997) propõem um novo tipo de protocolo com características especiais para ambientes móveis. Este protocolo armazena o estado global consistente da aplicação sem troca de mensagens. Os processos de *checkpoint*, podem ser armazenados no servidor ou localmente nos *hosts*. O armazenamento local não consome largura de banda e utiliza menos tempo para ser criado, contudo, podem ocorrer perdas durante desconexões da unidade móvel.

Durante a execução da aplicação, o protocolo mantém o estado global em um local fixo para recuperar falhas permanentes (falhas de *hardware*) e outro em unidades móveis para recuperar falhas temporárias (falhas de *software*).

(BADRINATH & ARUP, 1994) propuseram um protocolo baseado no armazenamento do estado global consistente para ambientes distribuídos, composto por duas fases, uma fase de enviar e outra de receber. O protocolo cria um *checkpoint*

sempre que a unidade móvel recebe uma mensagem na fase de envio. Todas as mensagens enviadas e recebidas são armazenadas. As mensagens das unidades móveis e os *checkpoints* são salvos em um armazenamento fixo da estação base corrente. Devido à mobilidade das unidades móveis para diferentes células, os *checkpoints* são distribuídos entre as estações base.

Algumas características de um esquema de *checkpoint* em um ambiente móvel são analisados a seguir, quais sejam: localização, desconexão, energia, rede e falhas.

5.6.1 Localização

Para que exista a comunicação entre a rede fixa e a unidade móvel, é necessário saber a localização exata desta última. A localização de uma unidade móvel dentro de uma rede fixa não é constante devido a sua mobilidade. Uma unidade móvel precisa ser localizada para depois determinar seu *checkpoint*.

O protocolo *checkpoint* tem a função de armazenar o estado do processo em um *site* conhecido ou em um computador próximo da localização atual da unidade móvel. O protocolo pode também manter a trilha dos lugares onde o estado do processo foi salvo.

5.6.2 Desconexão

Uma unidade móvel é dita desconectada quando está fora da faixa de transmissão dos emissores. Durante a desconexão, todas as informações armazenadas em *sites* remotos não estão acessíveis às unidades móveis. Desta forma, uma unidade móvel fica impossibilitada de enviar ou receber mensagens.

A criação de *checkpoints* por processos enquanto a unidade móvel está desconectada é possível. Por outro lado, os protocolos têm a função de salvar uma grande quantidade de informações que garantam sua execução após uma falha. Estas informações incluem a ordem da recepção e os conteúdos das mensagens. Quando as informações precisam ser salvas nas unidades móveis, poderá haver limitação de memória e de armazenamento.

Devido às freqüentes desconexões das unidades móveis, o local do *checkpoint* pode não estar disponível.

5.6.3 Energia

Devido às restrições oferecidas pelo tempo das baterias utilizadas pelos computadores portáteis, o consumo das mesmas deve ser reduzido.

O protocolo *checkpoint* tem a função de reduzir a quantidade de informações que são adicionadas para as mensagens e evitar o envio de mensagens extras, realizando o menor número possível de acessos ao servidor.

5.6.4 Rede

A diversidade de tecnologias utilizadas em redes *wireless* faz com que as características da rede não sejam constantes, tais como, custos, velocidade e latência. O protocolo *checkpoint* deverá se adaptar as características da rede onde está atuando.

5.6.5 Falhas

Unidades móveis podem apresentar falhas de *software* e de *hardware*. Falhas de *software* podem ocorrer quando acontecem problemas com o sistema operacional, descarga de bateria, entre outros. São gerenciadas por *soft checkpoints*, que são armazenados localmente na unidade móvel. Falhas de *hardware* ocorrem quando por exemplo, o computador portátil é roubado, danificado, etc. São gerenciadas por *hard checkpoints* que são armazenados na rede fixa.

O protocolo *checkpoint* deve fornecer mecanismos para tolerar os tipos de falhas que possam vir a acontecer com as unidades móveis.

5.6.6 Estratégias de Recuperação em um Ambiente Móvel

(KRISHNA, 1996) sugere estratégias de recuperação composta por dois componentes: *handoff* e *state-saving* (estado armazenado).

5.6.6.1 State Saving (Estado Armazenado)

Esta estratégia é baseada em técnicas de *checkpoint* e *logging* de mensagens, onde o *host* salva periodicamente seu estado em um local fixo. Como as unidades móveis estão em constante movimento, não são consideradas locais de armazenamento estável. A estação base a qual a unidade móvel pertence armazena seu estado. Caso ocorram falhas, a execução pode ser reiniciada do último *checkpoint* salvo.

As mensagens são primeiramente enviadas para a estação base, esta em contrapartida, as envia para a unidade móvel ou fixa correspondente.

Existem duas estratégias para salvar o estado do processo: *Logging* e *Não-Logging*.

Não-Logging: O estado do processo pode ser alterado, ou por mensagem recebida de outro *host* ou por entrada do usuário. O estado da unidade móvel é salvo na estação base em cada mensagem recebida ou entrada do usuário. Quando a unidade móvel reinicia, após uma desconexão, o *host* envia a mensagem para a estação base, a qual transfere o último estado para a unidade móvel. Neste esquema, há a necessidade de freqüentes transmissões de estado.

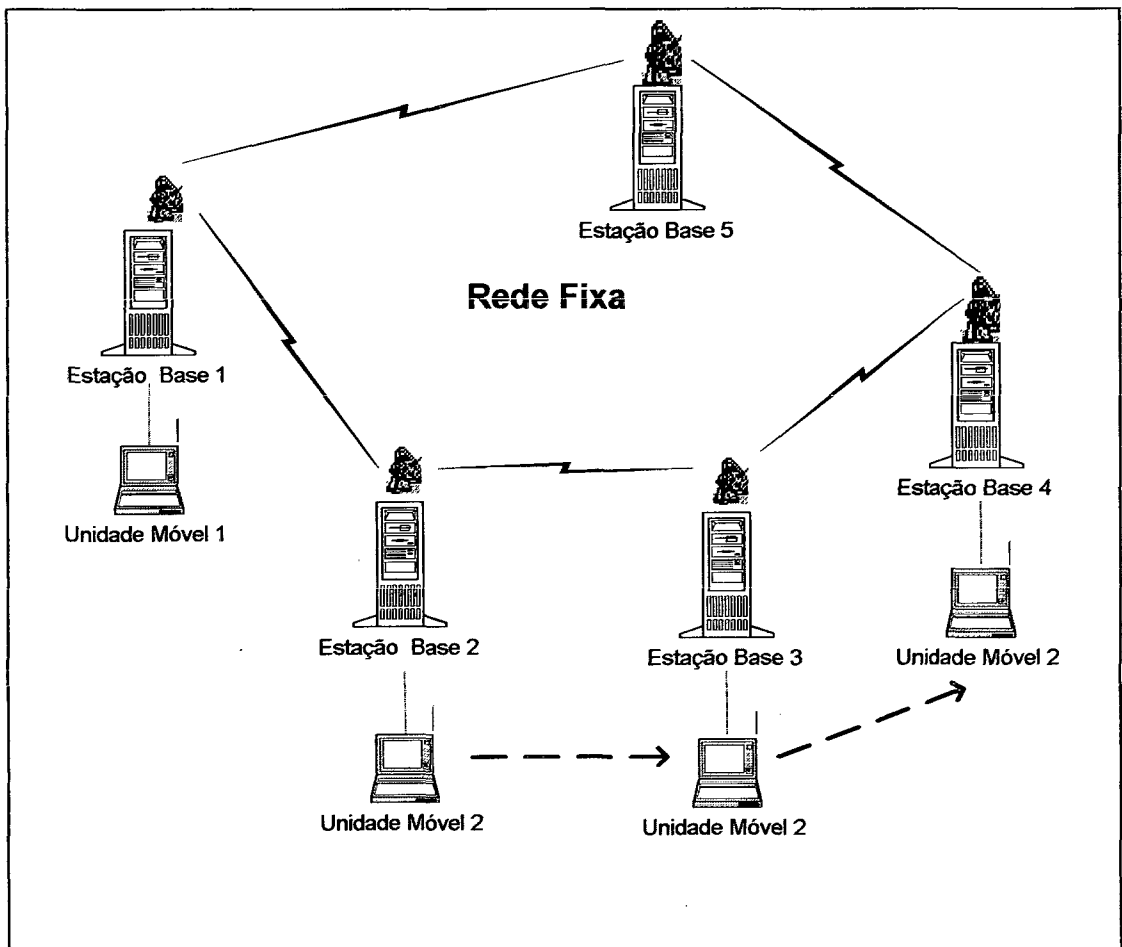
Logging: A unidade móvel verifica seu estado periodicamente. As mensagens e entradas que modificam o estado da unidade móvel são armazenadas. Caso uma mensagem seja recebida de outro usuário, a estação base primeiro a *loga*, e depois a envia para a unidade móvel, para ser executada. Em uma entrada do usuário, a unidade primeiro envia uma cópia para a estação base para *logging* e depois do reconhecimento da unidade móvel, processa a entrada e envia uma resposta para o usuário. Este procedimento garante que as mensagens ou entradas do usuário não sejam perdidas devido às falhas que possam ocorrer.

5.6.6.2 Handoff

Um grande problema para recuperação de dados é a mobilidade de estações móveis de célula para célula, não sendo possível a definição exata de sua localização.

A figura 34 representa a execução de um *handoff*. As unidades móveis 1 e 2 executam um algoritmo distribuído.

Figura 34 - Handoff - Estratégia de Recuperação



Na execução de um *handoff* a estação base 2 armazena o *checkpoint* e as mensagens de *log* da unidade móvel 2. Durante a execução, a unidade 2 se desloca para a célula da estação 3, e depois para a célula da estação 4.

O processo de *handoff* ocorre entre os limites das estações 2 e 3, e entre as estações 3 e 4. Se uma falha ocorrer entre a unidade 2 até sua chegada na estação 4, as informações poderão ser recuperadas devido ao seu armazenamento na estação 2. Caso este processo não seja realizado, isto é, as informações não sejam armazenadas, a estação base onde se encontra a unidade móvel deverá identificar onde o *checkpoint* foi salvo. Este procedimento poderá ser baseado em uma transferência durante o processo de *handoff* sobre o estado da unidade móvel.

Existem três modos de transferir informações durante o *handoff pessimistic* (pessimista), *lazy* (preguiçoso) e *trickle* (fluxo) (KRISHNA, 1996).

- ♣ ***Pessimistic***: Em um descolamento de uma unidade móvel de uma célula para outra, o *checkpoint* é transferido durante o *handoff*. Caso a estratégia de *logging* esteja sendo usada, a mensagem de *log* é transferida juntamente para a nova célula da estação base. Uma vez recebido o *checkpoint* ou o *log*, a estação base envia um reconhecimento para a estação anterior e esta cancela estas informações, desde que a unidade móvel não esteja mais em sua célula. A principal desvantagem é que esta aproximação requer que um grande volume de dados seja transferido durante o *handoff*
- ♣ ***Lazy***: Durante o *handoff* não há transferência de *log* e *checkpoint*. A estratégia cria uma lista de estações base de células visitadas por uma unidade móvel que pode utilizar as estratégias de *Logging* e *Não-Logging* descritas anteriormente. A desvantagem desta estratégia é o armazenamento de *checkpoints* e *logs* em diferentes estações base.
- ♣ ***Trickle***: Garante que *logs* e *checkpoints* correspondentes a uma unidade móvel, estejam sempre na estação base mais próxima. Durante o *handoff*, uma mensagem é enviada para a estação precedente para transferir qualquer *log* ou *checkpoint* que tenha sido armazenado em uma unidade móvel. Semelhante à estratégia *Lazy*, a estação base corrente envia uma mensagem de controle para a

nova célula da estação base, identificando a localização da célula da unidade móvel, que grava a autenticação da unidade móvel precedente.

5.7 Segurança em Ambientes Móveis

A segurança está relacionada à necessidade de proteção contra o acesso ou manipulação, intencional ou não, de informações confidenciais por elementos não autorizados, e à utilização não autorizada do computador ou de seus dispositivos periféricos.

A proposta de um sistema de segurança é restringir o acesso a informações e recursos para apenas alguns usuários autorizados. Para produzir um sistema seguro contra ameaças específicas é necessário classificar as ameaças e os métodos pelos quais cada uma delas pode ser produzida.

Com a necessidade de mobilidade, a utilização de computadores portáteis e de outros equipamentos móveis tem crescido gradualmente e conseqüentemente, a necessidade de proteção dos dados. A segurança em aparelhos móveis é um dos maiores problemas em redes sem fio. Neste ambiente em que a mobilidade é intensamente explorada, a vulnerabilidade do sistema e dados sensíveis torna-se maior, a troca de mensagens pode ser capturada por alguém não-autorizado. Ondas de rádio viajam abertas pela atmosfera onde podem ser interceptadas por pessoas que estão conectadas constantemente, podendo ser violadas e utilizadas de maneira incorreta. Números de cartões de crédito, números de conta bancária e de número de celulares podem ser *clonados*, sendo utilizados sem que se pague pelo serviço.

Em um ambiente centralizado, o gerenciamento da segurança não é escalável quando o tamanho ou a complexidade da rede cresce. Neste ambiente, uma falha em um ponto da rede ou no gerente pode deixá-la sem gerenciamento. Em um ambiente móvel, os agentes móveis descentralizam o processamento e o controle, reduzindo o tráfego da estação de gerenciamento, tornando a comunicação assíncrona, de modo a permitir um balanceamento de carga de processamento e aumentar a flexibilidade do comportamento dos agentes de gerenciamento.

No decorrer do capítulo são abordados os principais temas relacionados a segurança de dispositivos móveis: encriptação de dados, assinatura digital e *firewall*.

A encriptação de dados consiste em escrever mensagens em cifra ou em códigos, de forma a permitir que somente o destinatário a decifre e compreenda. É uma das maneiras de se evitar o acesso indevido a informações confidenciais através de uma mistura digital de bits com algoritmos matemáticos.

A assinatura digital é um mecanismo elaborado para garantir a autenticação e integridade de informações, permitindo assim que se prove quem é o autor de um determinado documento e, se este não foi alterado ou forjado por terceiros. São importantes para o comércio eletrônico e para os métodos de autenticação.

Um *firewall* é uma combinação de *hardware* e *software* com o objetivo de impedir o acesso de usuários não-autorizados a informações do sistema. Pode ser inserido entre a rede fixa, nos servidores de segurança sem fios e o servidor de banco de dados. A aplicação mais comum de um *firewall* é proteger uma rede contra o acesso de usuários mal-intencionados.

5.7.1 Políticas de Segurança

O primeiro passo para garantir a segurança em um banco de dados consiste no desenvolvimento de uma política de privacidade e segurança. A política de segurança define precisamente os requisitos que deverão ser implementados no *software* e no *hardware* do sistema, assim como aspectos de controle humano, físico e procedimental. Para tanto, são necessários mecanismos para implementar os requisitos da política de segurança, que realizem as funções pretendidas. Outro fator a ser considerado é quanto aos aspectos de garantia que devem prover diretrizes que garantam que os mecanismos adotados cumpram com um alto grau de confiabilidade a política de segurança.

As ameaças à segurança de sistemas móveis podem ser ocasionadas por diversos fatores (http://www.mobileinfo.com/mobileinfo_security_issues.htm):

- ⌘ Intercepção da autorização de cartões de crédito através de redes sem fio;

- ⤴ Lacunas físicas de segurança não gerenciadas pela estação base ou outra comunicação;
- ⤴ Interceptação de mensagens de *e-mail* nas conexões sem fio da Internet;
- ⤴ Roubo de ondas de rádio da telefonia celular;
- ⤴ Aquisição de informações por pessoas não autorizadas;
- ⤴ Falsificação através da alteração de informações (incluindo programas) sem autorização;
- ⤴ Interferência nas operações próprias de um sistema, sem ganho por parte do perpetrador.

5.7.2 Princípios Gerais de Segurança

(FEDERRATH et. al, 1995) divide os princípios gerais de segurança em três partes:

- ⤴ **Privacidade:** Tem como objetivo prevenir o acesso de informações não autorizadas a integrantes do grupo, ou seja, proteger o banco de dados contra intrusos.
- ⤴ **Integridade:** Garante que modificações não autorizadas de informações ou de processos não sejam realizadas no banco de dados e falsificações do conteúdo das mensagens sejam detectadas.
- ⤴ **Disponibilidade:** Tem como finalidade fazer com que a comunicação da rede esteja apta a se comunicar com todas as partes da rede que desejarem fazê-la.

5.7.3 Métodos de Ataque

As principais ameaças à segurança em sistemas distribuídos derivam da abertura dos canais e de sua conseqüente vulnerabilidade. Todo canal de comunicação em qualquer nível de *hardware* ou *software* do sistema está sujeito a essas ameaças. Violadores em potencial (humanos ou programas) não são facilmente identificáveis.

Os ataques às redes de computadores podem gerar alguns tipos de irregularidades, como destruição de informação ou de outros recursos; modificação ou deturpação da

informação; roubo, remoção ou perda da informação ou de outros recursos; revelação de informações; interrupção de serviços, entre outros prejuízos.

Os ataques são classificados em dois tipos: ataques passivos e ataques ativos. Os ataques passivos não alteram conteúdo, apenas extraem informações. São mais difíceis de detectar. Dividem-se em: *Eavesdropping* e *masquerading*.

- ⤴ ***Eaveadropping***: É caracterizado pela obtenção de cópias de mensagens sem autorização. Isto pode ser feito diretamente pela rede ou examinando informações que estejam protegidas inadequadamente durante a armazenagem.
- ⤴ ***Masquerading***: É caracterizado por enviar ou receber mensagens usando a identidade de outro usuário, sem autorização. Isto pode ser feito utilizando a senha e a identidade desse usuário ou usando um *token* de acesso depois que a autorização para o uso tenha expirado.

Ataques ativos são os que intervêm no fluxo normal de informação, quer alterando o seu conteúdo, quer produzindo informação não fidedigna, normalmente com o intuito de atentar contra a segurança de um sistema. São mais difíceis de prevenir. Os ataques ativos se dividem em : *Message Tampering* e *replaying*.

- ⤴ ***Message tampering***: Tem como intenção interceptar mensagens ou alterar seu conteúdo antes que elas cheguem ao destinatário. É difícil conseguir evitar esse tipo de violação dentro de um meio *broadcast*, uma vez que a camada física de comunicação garante a entrega de mensagens para todas as estações.
- ⤴ ***Replaying***: Tem a intenção de armazenar mensagens e enviá-las depois de algum tempo, depois do esgotamento do prazo para utilização de um recurso, por exemplo. O *replaying* não pode ser combatido por simples encriptação, uma vez que o reenvio de mensagens pode ser utilizado para consumir tempo de utilização de recursos e vandalismo, mesmo quando as mensagens não podem ser interpretadas pelo invasor.

5.7.4 Infiltração

Para infiltrar-se em um sistema distribuído, o atacante deve ter acesso ao sistema para poder executar o programa que é implementado no sistema. A maioria dos ataques é lançada por usuários de dentro do sistema.

Para usuários de fora do sistema, um método simples é o de adivinhação de senhas, ou do uso de programas para descobrir a senha de um usuário conhecido. Tais ataques podem ser prevenidos pelo uso de senhas bem escolhidas e de tamanho adequado.

Além dessas formas diretas de infiltração, existem vários métodos que são citados a seguir:

- ⤴ **Vírus:** Programa que é anexado a um programa ou arquivo hospedeiro e se instala no ambiente alvo sempre que o hospedeiro é executado. Uma vez instalado, o vírus executa ações que danificam informações. Tem o poder de replicar-se, anexando-se a todos os programas ou arquivos que encontrar no ambiente alvo. Eles se transferem para outras máquinas sempre que um programa ou arquivo hospedeiro é movido, seja através de comunicação de rede ou através de armazenamento físico.
- ⤴ **Worm:** é um programa que explora as facilidades para execução remota de processos em sistemas distribuídos. Tais facilidades podem existir tanto acidentalmente quanto propositadamente.
- ⤴ **Cavalo de Tróia:** Nem sempre é considerado um vírus porque não tem a capacidade de se replicar. É um programa que vem acoplado a um arquivo executável aparentemente inofensivo e que é instalado no computador sem que o usuário perceba. É ativado quando se executa este programa infectado
- ⤴ **Clonagem:** é uma forma utilizada para falar ao celular sem débitos em conta. O que os invasores fazem é programar um telefone com os mesmos parâmetros que um telefone que esteja sendo utilizado por um usuário que pague sua conta corretamente. Esta clonagem pode ser feita de diversas formas, porém, uma das mais comuns é a cópia do aparelho registrado quando o mesmo é levado em assistências técnicas não-autorizadas.

Existem vários meios de prevenção para garantir a segurança em ambientes móveis em diferentes aspectos ou propósitos, dentre eles: encriptação de dados, assinaturas digitais e *firewalls*;

5.7.5 Encriptação de Dados

A criptografia (*kriptós* = escondido, oculto; *grápho* = grafia) consiste em escrever em cifra ou em códigos, de forma a permitir que somente o destinatário decifre a mensagem e a compreenda. É uma das maneiras de se evitar o acesso indevido a informações confidenciais através de uma mistura digital de bits com algoritmos matemáticos. As informações são codificadas ou cifradas, sendo acessíveis apenas às pessoas às quais estas informações são destinadas.

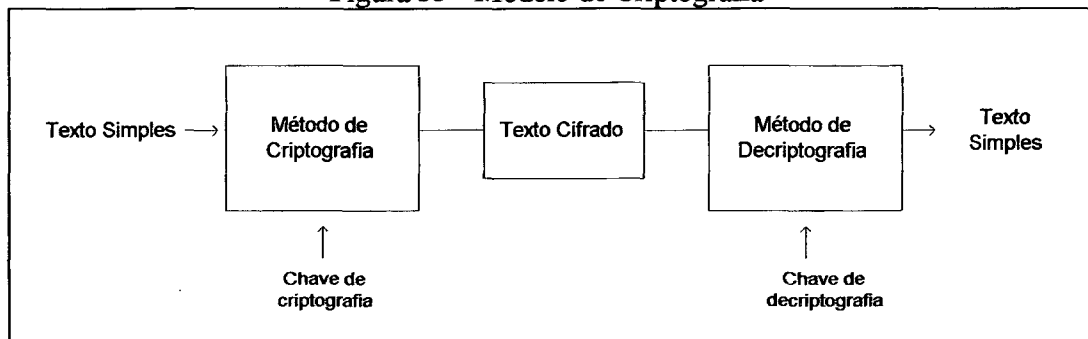
A criptografia fornece técnicas para codificar e decodificar dados de modo que os mesmos possam ser armazenados, transmitidos e recuperados sem sua alteração ou exposição, se tornando um método para segurança de informações confidenciais durante troca de mensagens. Ainda como recurso de segurança, pode-se misturar ou codificar os dados de forma que, ao serem armazenados no disco rígido ou transmitidos por alguma linha de comunicação, não passem de bits ininteligíveis para aqueles que os acessam por um meio que não o oficial. A criptografia é importante nos bancos de dados, pois a informação fica armazenada por um longo tempo em meios de fácil acesso, como por exemplo, discos rígidos.

As técnicas de criptografia podem ser usadas como um meio efetivo de proteção de informações suscetíveis a ataques. Seu principal objetivo é prover uma comunicação segura, garantindo serviços básicos de autenticação, privacidade e integridade dos dados.

A figura 35 descreve um modelo para a criptografia. Primeiramente as mensagens que serão criptografadas, conhecidas como texto simples, são transformadas por uma função parametrizada por uma chave. A chave consiste em um *string* que pode ser alterada sempre que necessário. Na seqüência, o texto simples passa por um processo de criptografia, sendo transformado em um texto cifrado (pode então ser transmitido por difusão de dados ou em forma de mensagens). A seguir, o texto cifrado passa pelo

método de deciptação, se transformando novamente no texto original. Para deciptografar o texto, é necessário conhecer a chave de deciptografia.

Figura 35 - Modelo de Criptografia



Existem várias técnicas para encriptação de dados, mas as mais eficientes devem obedecer aos seguintes princípios:

- ⌘ A técnica de encriptação utilizada deve ser relativamente simples para um usuário autorizado encriptar e deciptar dados;
- ⌘ O esquema de encriptação não deve depender do algoritmo utilizado, mas de um parâmetro chamado **chave de encriptação**;
- ⌘ Deve ser extremamente difícil para um intruso determinar a chave de encriptação.

De um modo geral, os métodos de encriptação baseiam-se em métodos divididos em duas categorias (TANEMBAUM, 1997):

Encriptação por substituição: Neste método de encriptação, cada letra ou grupo de letras é substituído por outra letra ou conjunto de letras. As letras são "disfarçadas", contudo, a ordem é preservada. Este método é muito vulnerável, uma vez que os ataques tomam vantagem das propriedades estatísticas das línguas naturais. Temos o seguinte exemplo:

Texto original :	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Texto encriptado:	Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

Encriptação por transposição: Em contraste ao método da substituição, é feita uma troca do texto através da transposição. Neste método não há troca de um caracter por outro, havendo, no entanto, uma reordenação dos mesmos, isto é, eles são escritos fora de ordem.

A seguir são descritos dois métodos de criptografia: por chave pública e simétrica.

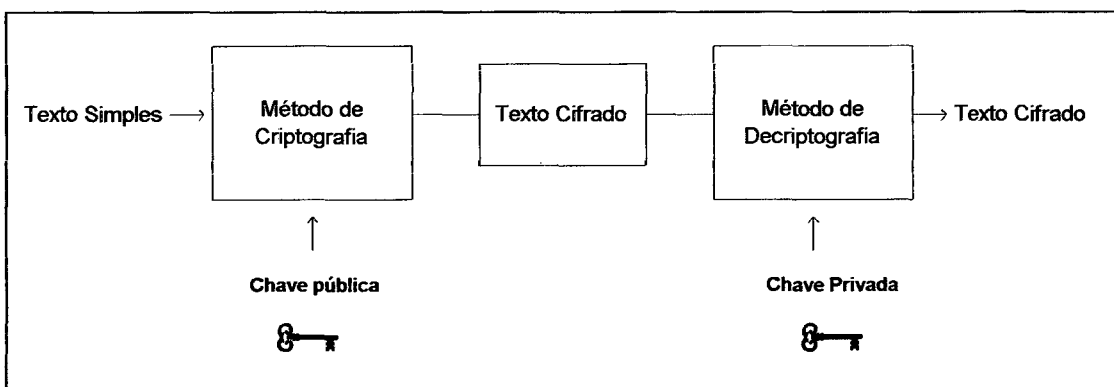
5.7.5.1 Criptografia de chave pública

O processo de encriptação/decriptação depende de duas chaves, uma pública e outra privada. Como o próprio nome diz, a chave privada é de conhecimento único e exclusivo. Já a pública deve estar disponível a quem quiser enviar informações encriptadas. Os dados são criptografados com a chave pública, mas só é possível decriptografar com a privada, ou vice-versa.

Quando uma chave pública é criada, cria-se também uma chave privada. Essas duas chaves estão relacionadas entre si. A chave pública pode e deve ser distribuída livremente.

Quando uma pessoa deseja enviar uma mensagem encriptada para esse usuário, ela usa a sua chave pública para encriptar a mensagem. E quando ele recebe, usa sua chave privada para desencriptar a mensagem, conforme o esquema da figura 36.

Figura 36 - Criptografia de Chave Pública



A chave privada é o método de encriptação que utiliza uma mesma chave (senha), tanto para encriptar o arquivo, quanto para decriptá-lo. É medida em bits e, quanto maior o número de bits, mais segura é a encriptação. A chave pública é a mais recomendada para o envio de mensagens encriptadas.

5.7.5.2 Criptografia simétrica

Uma única chave é utilizada para criptografar e decriptografar os dados. Ela é relativamente pequena e rápida, contudo, como desvantagem, não só o transmissor deve conhecer a chave, como também o receptor. Além disso, o volume total dos dados transmitidos é limitado pelo tamanho da chave. É usada frequentemente para comunicação entre duas máquinas ou para armazenamento da informação em um disco rígido.

5.7.6 Assinatura Digital

As assinaturas digitais são mecanismos elaborados para garantir a autenticação e integridade de informações, permitindo que se prove quem é o autor de um determinado documento e se este não foi alterado ou forjado por terceiros. A assinatura digital é uma garantia que o documento é uma cópia verdadeira e correta do original. Assinaturas digitais, como as convencionais, podem ser forjadas. A diferença é que a assinatura digital pode ser matematicamente verificada.

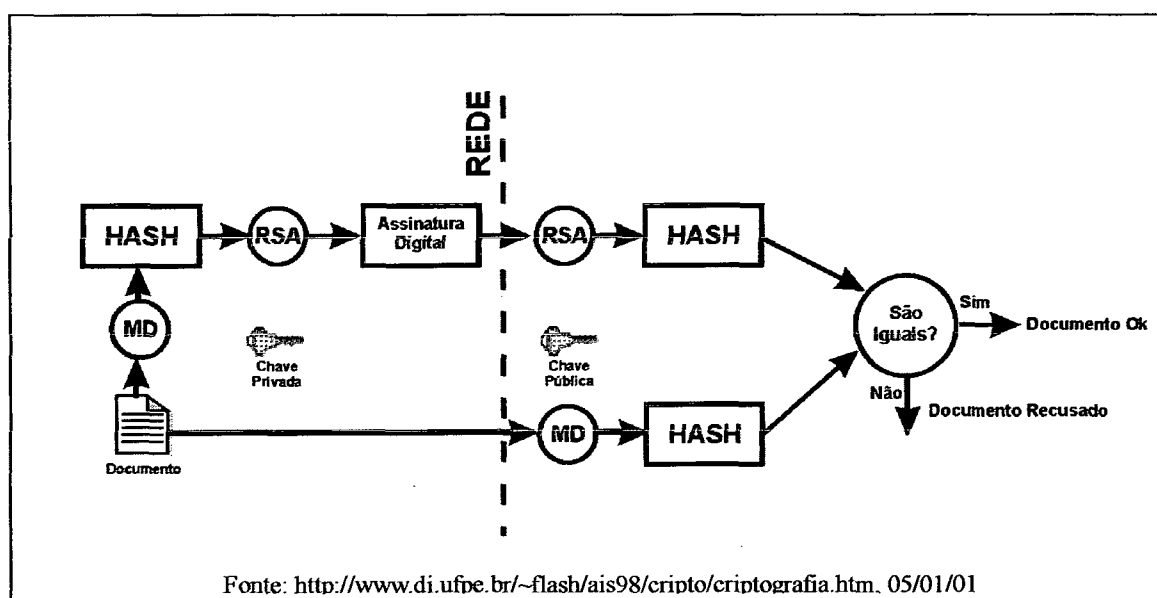
As assinaturas digitais são importantes principalmente para o comércio eletrônico e para os métodos de autenticação. São divididas em três tipos:

- ⤴ Assinaturas digitais que fazem uso de chave única;
- ⤴ Assinaturas que fazem uso de chaves públicas e
- ⤴ Assinaturas que fazem uso de MD (*Message Digest*), que é uma função matemática que refina toda a informação de um arquivo em uma única porção de dados de tamanho fixo.

A figura 37 mostra um esquema de geração e verificação de assinatura digital. Primeiro, executa-se a função MD (*Message Digest*), obtendo assim um *hash*¹¹ para aquele documento, e posteriormente, decifra-se a assinatura digital com a chave pública do remetente. A assinatura digital decifrada deve produzir o mesmo *hash* gerado pela função MD executada anteriormente. Se estes valores são iguais, é determinado que o documento não foi modificado após a assinatura do mesmo, caso contrário, o documento ou a assinatura, ou ambos, foram alterados. Infelizmente, a assinatura digital pode apenas indicar que o documento foi modificado, mas não o que foi modificado e o quanto foi modificado. Todo o processo de geração e verificação de assinatura digital utiliza o algoritmo de criptografia de chave pública RSA (Rivest, Shamir e Adleman, nome de seus criadores).

O RSA usa duas chaves criptográficas, uma chave pública e uma privada. A segurança desse método se baseia na dificuldade de fatorar números extensos. Segundo seus pesquisadores, a fatoração de um número de 200 dígitos requer 4 milhões de anos para ser processada. Fatorar um número de 500 dígitos exige 1025 anos¹².

Figura 37 - Esquema Assinatura Digital



¹¹ Pequena porção de dados

¹² <http://www.di.ufpe.br/~flash/ais98/cripto/criptografia.htm>

5.7.7 Firewall

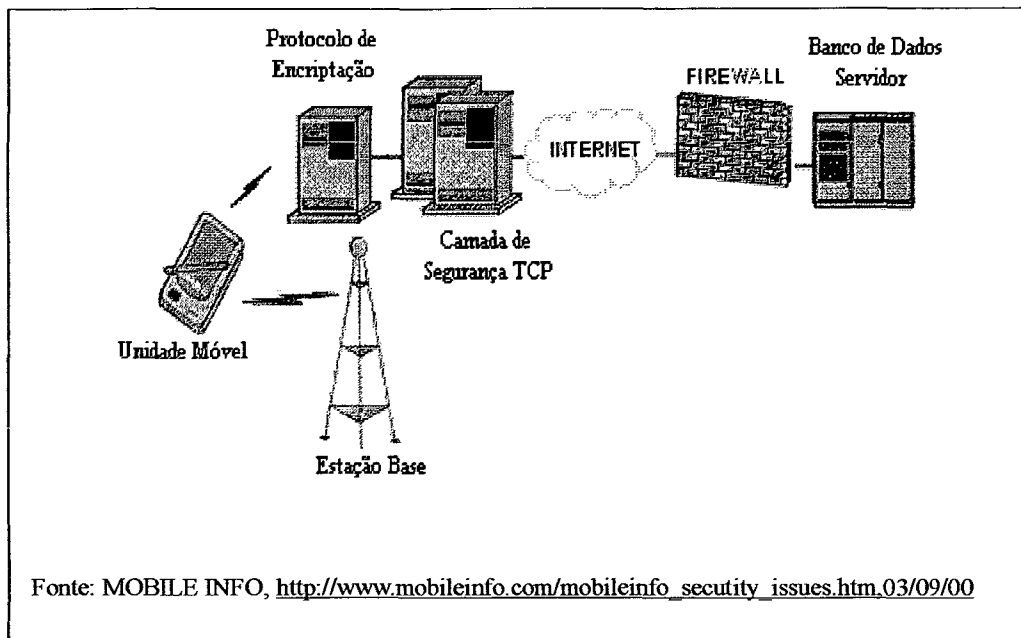
Os *firewalls* são uma adaptação moderna de uma antiga forma de segurança medieval: cavar uma poça profunda em torno do castelo. Esse recurso forçava todos que quisessem entrar ou sair do castelo a passar por uma única ponte levadiça, onde poderiam ser revistados por guardas (TANEMBAUM, 1997).

Um *firewall* é considerado um método de defesa inserido entre a rede fixa nos servidores de segurança sem fio e o servidor de banco de dados, como mostra a figura 38. Em algumas situações em que a comunicação é realizada via rádio onde os pacotes passam pelo *firewall* em ambas as direções, deixando o sistema vulnerável a ataques.

A aplicação mais comum de um *firewall* é proteger uma rede contra acesso de usuários mal-intencionados. São utilizados para reconhecer e neutralizar ataques efetuados da utilização maliciosa de características específicas dos protocolos de comunicação usados na rede, mas pouco pode fazer contra ataques baseados em dados. O *firewall* não pode ser considerado uma solução isolada para a questão de segurança na comunicação com a Internet. Para ser eficaz, deve fazer parte de uma política global de segurança.

Fisicamente o *firewall* é um dispositivo de *hardware* dotado de duas placas de rede (uma ligada à rede fixa e a outra ligada na unidade móvel) executando um *software* específico de análise e roteamento de pacotes. Como todo pacote enviado de uma rede a outra passa obrigatoriamente pelo sistema, o *firewall* tem a chance de analisá-lo, determinar se ele representa algum risco e, se for o caso, descartá-lo antes que ele possa alcançar seu destino. Um *firewall* será inútil se existirem outras entradas para a rede, como por exemplo, através de *modems*.

Figura 38- Esquema de Segurança Firewall



Os critérios utilizados para decidir se determinado pacote de dados oferece ou não riscos, fazem parte da política de segurança praticada. Os *firewalls* podem ser implementados de várias maneiras, por exemplo, operando na camada de rede, com a função de analisar pacotes IP e operando na camada de aplicação, analisando os dados dentro dos pacotes .

Máquinas bem configuradas só concedem acesso a computadores conhecidos, em geral, localizados numa mesma rede fixa. Assim, para obter acesso a uma estação bem configurada, é necessário convencê-la de que a máquina do invasor é de confiança, criando um processo denominado *spoofing*. Para isso, o invasor precisa descobrir o endereço legítimo de um nó da rede interna e enviar à vítima pacotes que apresentam tal endereço como origem. A vítima, crendo ser o atacante uma máquina de confiança, responderá enviando pacotes para o seu endereço.

Um *firewall* de rede sofisticado não se limita em rotear os pacotes para seus destinos: mantém também informações sobre o estado das conexões e sobre o conteúdo dos pacotes, o que lhe permite perceber que um pacote cujo endereço de origem pertence à rede interna não pode provir da Internet. Se isto ocorre, configura-se o

spoofing e o *firewall* descarta o pacote e aciona o alarme. Similarmente, comunicações normais não devem gerar pedidos de *source-routing*, de modo que, se um pedido surge, o *firewall* deve considerá-lo um ataque e agir de acordo. Independente de sua sofisticação, *firewalls* de rede roteiam tráfego diretamente, o que os torna rápidos e transparentes, mas os impede de analisar o conteúdo efetivo dos pacotes trafegados e exige que as máquinas na rede interna tenham endereços IP válidos (o que pode ser um risco).

Quanto ao comércio eletrônico, existe um custo associado à implementação e manutenção de *firewalls*, uma vez que limitam o acesso a certas operações como compra e venda de produtos. Para minimizar estas desvantagens, pode-se optar por implementar um *firewall* simples, com poucas regras de restrição de acesso, quando uma organização pretende fazer publicidade de produtos ou serviços ou outro tipo de operações comerciais.

CAPÍTULO VI

TECNOLOGIAS DE BANCO DE MÓVEIS

Alguns fornecedores de bancos de dados, como a *Oracle*, *IBM*, *Microsoft* e *Sybase*, lançaram ou anunciaram versões de seus bancos de dados corporativos para dispositivos portáteis, proporcionando a funcionalidade de banco de dados em três camadas: servidores corporativos, *desktops* e unidades móveis. A *IBM*, com o *DB2E* seguiu um rumo um pouco diferente, introduzindo um banco de dados relacional pequeno, separado, projetado especificamente para PDAs e *handheld*.

Neste capítulo analisa-se algumas das tecnologias móveis com suporte a banco de dados móveis, quais sejam:

- ♣ *Oracle 8i Lite* (http://technet.Oracle.com/products/8i_Lite/pdf/Ceds99.pdf);
- ♣ *SQL Server CE* (<http://www.microsoft.com/brasil/sql/productinfo/prodover.stm>);
- ♣ *DB2 Everywhere* (www.developer.ibm.com/devcon/novcc00.htm);
- ♣ *SQL Anywhere Studio* (www.hallogram.com/sybase-sql-anywhere).

A *Oracle* lançou o *Oracle8i* com aplicações para a computação móvel, e integração com o *Oracle8i Lite*, podendo explorar as suas aplicações *Web* através de um *browser* na Internet ou na Intranet, utilizando uma única aplicação na rede fixa ou em unidades móveis, gerindo dados em diferentes sistemas operacionais e com ferramentas de sincronização incluídas no pacote. O principal enfoque da empresa era oferecer um produto que permitisse a integração de estruturas de bancos de dados com a Internet e proporcionasse o acesso aos dados independentemente da plataforma utilizada.

Com a tecnologia *SQL Anywhere Studio* a *Sybase* apresenta um pacote para gerenciamento e sincronização de dados que permite o *design* e distribuição das informações corporativas para grupos de trabalho, dispositivos móveis e sistemas com bancos de dados embutidos.

O *SQL Server 2000* da *Microsoft*, lançado em outubro de 2000, é o primeiro produto da empresa a fornecer aplicações para Internet com suporte ao padrão XML (*Extensible Markup Language*). Possui uma versão para equipamentos sem fio baseada no sistema operacional *Windows CE*, chamado *SQL Server CE*.

O *DB2 Everywhere*, o *SQL Server CE*, o *Oracle 8i Lite*, ou o *SQL Anywhere Studio* tratam aspectos de gerenciamento de dados como sincronização, replicação, estruturação e segurança do banco de dados.

6 Tecnologias de Banco de Móveis

Dado o atual estado da arte em banco de dados móveis existentes no mercado, a seguir é feita uma análise de alguns produtos que empregam conceitos de gerenciamento de dados móveis que foram estudados no decorrer deste trabalho.

6.1 Oracle8i Lite

O *Oracle8i Lite* é um *software* de gerenciamento de bancos de dados que integra a linguagem Java e acesso via Internet. Tem a seguinte filosofia: qualquer plataforma, a qualquer momento em qualquer lugar. Possui plataforma para desenvolvimento e gerenciamento de aplicações móveis em *Web*.

O *Oracle8i* possui um servidor de banco de dados da *Oracle* com uma *Java Virtual Machine* (JVM) incluída, para permitir aos programadores, desenvolver, arquivar e executar código Java dentro da própria base de dados, possibilitando assim aplicações com mais rapidez e mais segurança. A flexibilidade da linguagem Java permite aos programadores escreverem código uma vez e disponibilizá-lo desde *mainframes* a dispositivos *handheld*. Também incluído com o *Oracle8i* está o *Oracle WebDB*, uma ferramenta de desenvolvimento pronta para a Internet, que permite aos seus usuários construir e colocar a disposição *sites Web* dinâmicos e de base de dados, utilizando um *browser* padrão.

É composto por três componentes principais: *Oracle Lite*, *iConnect* e *Internet Lite/Web-to-Go* (SHORES, 2000).

6.1.1 Oracle Lite

O *Oracle Lite* possui tamanho entre 50K a 750K, dependendo de plataforma utilizada. É compatível a linguagem Java, podendo ser executado em PC's, *laptops*, PDA's, telefones celulares, entre outros. Apresenta as seguintes características:

- ⤴ Suporte a SQLJ;
- ⤴ Suporte a *stored procedures* e *triggers*;
- ⤴ Acesso a classes Java;
- ⤴ Suporte a múltiplas interfaces: ODBC, JDBC, SQLJ, OKAPI, JAC;
- ⤴ Possibilidade de utilização de múltiplas ferramentas de desenvolvimento como: *C++ Visual*, *Visual Café*, *Oracle JDeveloper* e *Developer*, *Delphi*, *Satellite Forms*, e *CodeWarrior*, etc;
- ⤴ Suporte a vários sistemas operacionais como: *Windows CE*, *Windows 95/98/NT*, *PalmOS*, *EPOC*;
- ⤴ Compatibilidade com o *Oracle database servers*.

6.1.2 iConnect

É um conjunto de serviços que viabilizam o sincronismo dos dados bilaterais, entre o servidor central e o *Oracle Lite* instalado nos equipamentos móveis. O *iConnect* permite desempenhar replicação com cópias locais ou replicar dados de um banco de dados remoto conhecido como *site master*.

O *iConnect* possui as seguintes características: replicação avançada, *AQ Lite* e *Consolidator*.

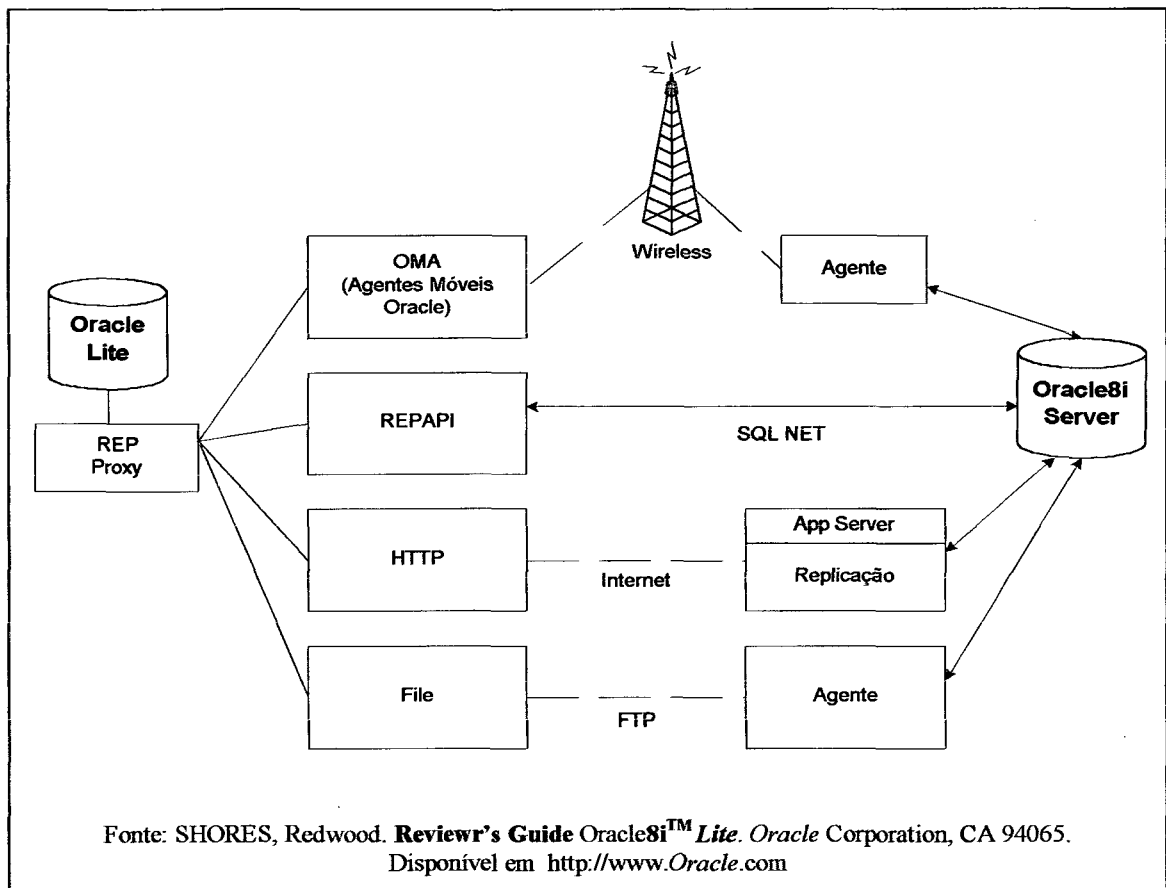
6.1.2.1 Replicação Avançada

Este recurso do *iConnet* suporta a replicação bidirecional com integridade de transações, garantindo replicação apenas de dados necessários para cada unidade móvel. Possui detecção e resolução de conflitos que podem ser pré-definidas ou escritas pelo usuário.

A replicação avançada suporta sincronização múltipla, protocolos de rede para LANs, Internet, transferência de arquivos e *wireless*. A seguir são descritos tipos de replicação realizadas no Oracle8i Lite (figura 39).

- ⤴ **Replicação baseada em conexão:** é configurada apenas no cliente. Desta forma o Centro de Replicação *Oracle* (REPAPI) e seus componentes também são instalados no cliente, exigindo maior quantidade de memória no mesmo.
- ⤴ **Replicação baseada na Internet:** Permite ao usuário móvel, enviar as alterações de um *site Snapshot* para uma aplicação *Web*, HTTP ou protocolos MIME, minimizando o número de conexões;
- ⤴ **Replicação baseada em arquivos:** pode ser feita em arquivo ou baseada em disco. Requer configuração, administração e instalação de componentes, tanto no cliente como no servidor.
- ⤴ **Replicação baseada em OMA:** Utiliza infraestrutura dos Agentes Móveis *Oracle* (OMA) para realizar a replicação bidirecional sobre uma LAN, *dial-up* ou redes *wireless*. Os dados são transmitidos utilizando a estrutura *store-and-forward* (armazenar e enviar).

Figura 39 - Replicação Avançada



6.1.2.2 AQ Lite

O *AQ Lite* é um serviço de mensagem para computação móvel que provê a comunicação entre unidades móveis e servidores *Oracle*. Aplicações que estão sendo executadas em uma unidade móvel podem enviar e receber (*Store-and-Forward*) mensagens de outras aplicações que estão sendo executadas no servidor. Devido a este fato, não há a necessidade de constante conexão da unidade móvel com o servidor. Mensagens são armazenadas em seus próprios bancos de dados e enviadas posteriormente para o local de destino. O *AQ Lite* apresenta as seguintes características:

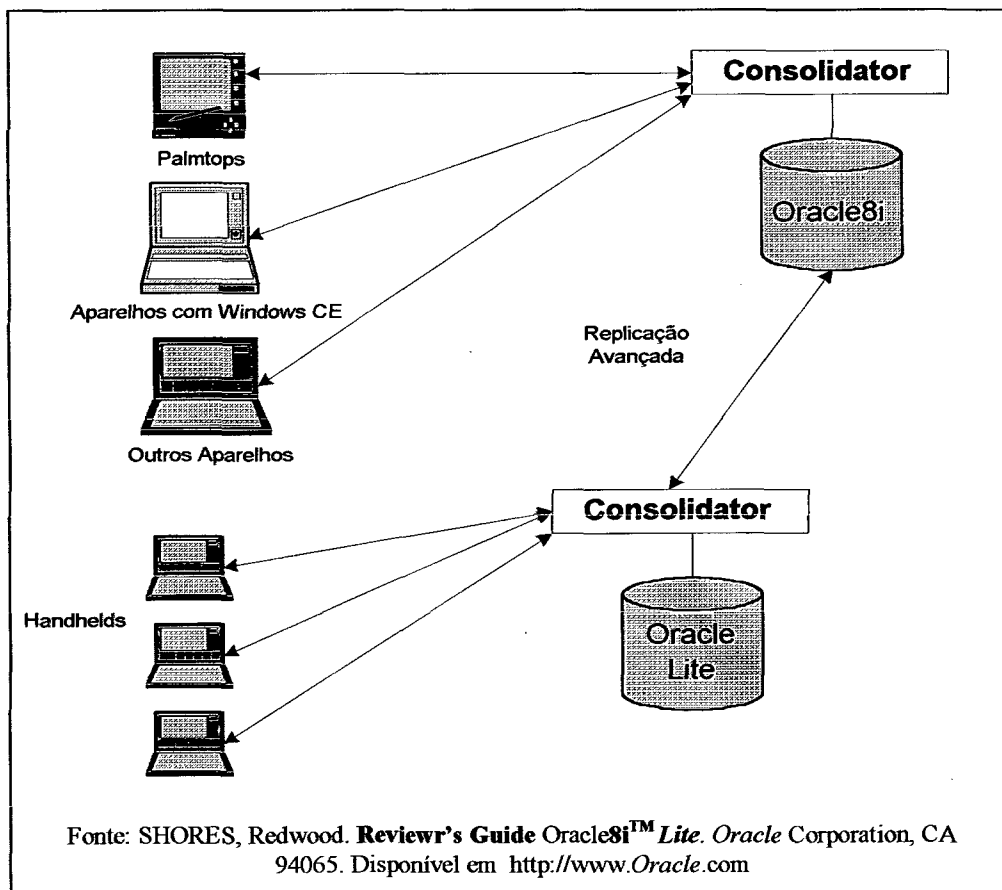
- ✦ Apresenta mensagem compatível com API, permitindo o desenvolvimento de mensagens baseadas em aplicação;

- ✧ Sincronização múltipla e protocolos de rede, tais como: LAN, Internet, *wireless* e baseada em arquivo;

6.1.2.3 Consolidator

O *Consolidator* possui suporte a tradução e sincronização de unidades móveis dentro do servidor de banco de dados *Oracle*. Permite formatar registros de dados para serem replicados, sincronizados e compartilhados com o servidor *Oracle8i*. O *Consolidator* tem arquitetura *two-tier*, como mostra a figura a seguir. De um lado estão os aparelhos móveis, representado a fila 1. De outro, o servidor *Oracle8*, representando a fila 2. Ambos devem ser diretamente sincronizados. O *Consolidator* também pode ser utilizado em uma arquitetura *three-tier*. Neste caso, a segunda fila é um banco de dados *Oracle Lite* e a terceira fila um servidor *Oracle*. A função do *Consolidator* é sincronizar dados entre as filas 1 e 2. A replicação avançada *Oracle* sincroniza os dados entre as filas 2 e 3.

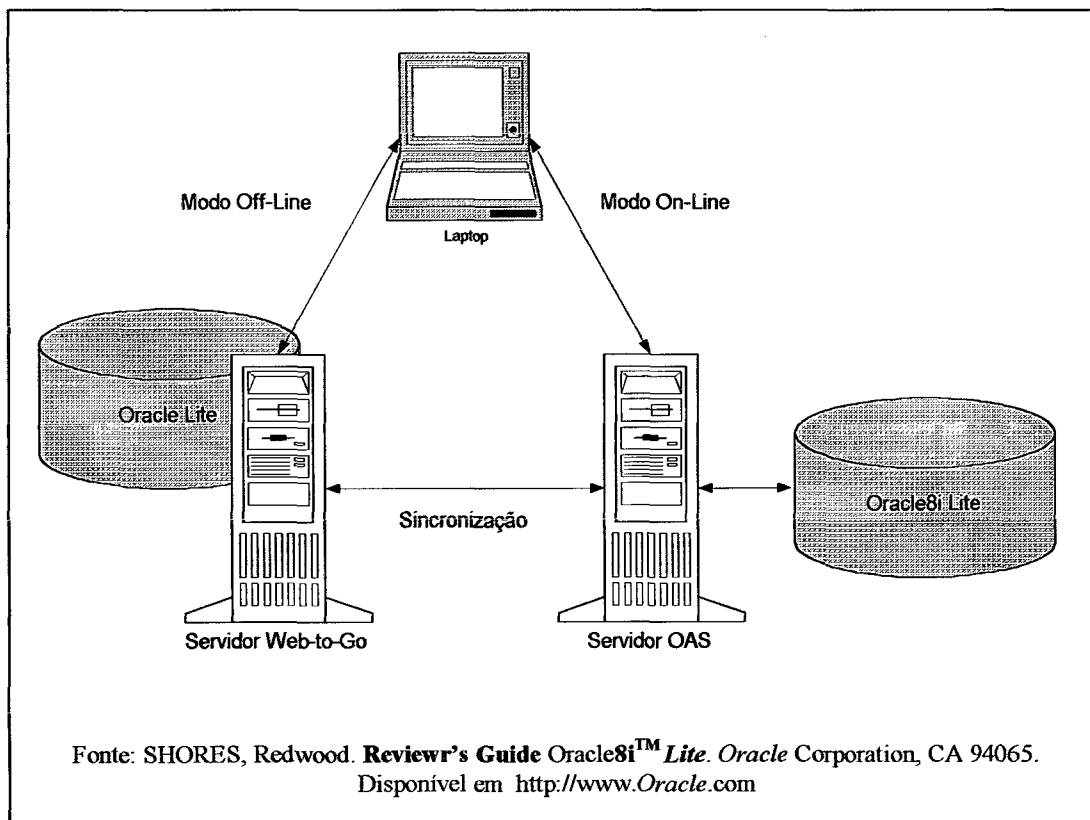
Figura 40 - Esquema de Funcionamento Consolidator



6.1.3 Internet Web-to-Go

São componentes que facilitam o desenvolvimento e gerenciamento de aplicações móveis para *Web*. O *Web-to-go* controla a sincronização de dados e a transparência de distribuição de *software* a usuários sem exigir codificação especial. Desta forma, uma única base de código pode executar em qualquer arquitetura, desde Internet, aplicação móvel, ou modelo cliente/servidor. Possui aplicações e dados armazenados em um servidor centralizado, evitando “ilhas de informação”.

Figura 41 - Arquitetura Web-to-Go



A aplicação é desenvolvida em HTML ou em Java, sendo disponibilizada pelo *Oracle Application Server* (OAS) na Internet. Enquanto o usuário está conectado utiliza

a aplicação no modo *On-line*¹³ (figura 41), ou seja, as modificações que forem realizadas estarão sendo espelhadas no banco de dados que é executado no OAS. Quando o usuário estiver se desconectando, no modo *Off-line*,¹⁴ os dados são espelhados no banco de dados, que é acessado pelo OAS, no *Oracle Lite Web Server*, instalado na unidade móvel. Deste modo, a aplicação se apresenta transparente para o usuário, como se estivesse o tempo inteiro *On-line*.

Uma vantagem desta abordagem é o fato de incluir um mini Web Server dentro da unidade móvel.

6.2 DB2 Everyplace

O IBM DB2 *Everyplace* (DB2E) é um banco de dados relacional que possibilita poder de processamento às áreas descentralizadas de uma empresa que estão fora do alcance de *desktops* e *laptops*. Foi projetado para PDAs, *handheld*, permitindo acessar e executar atualizações a um banco de dados de um dispositivo móvel. A idéia geral é armazenar uma quantia pequena de dados na unidade móvel que é sincronizada depois para outras unidades através do Banco de Dados Universal (UDB) executando em Unix, *Windows NT*, ou plataformas de OS/390.

DB2E vem com aplicativos e o *Query By Example* (QBE), utilizado para consultar ou modificar tabelas de bancos de dados. O QBE é uma ferramenta rudimentar e seu uso deve limitar-se a exibição de informações simples ou a teste de aplicativos.

A seguir são descritas algumas características do DB2E:

- ♣ Tamanho reduzido (100 a 150 Kb);
- ♣ Sincronização de dados e aplicativos via *Mobile Connect*, software que oferece sincronização de informações entre a plataforma móvel e sistemas corporativos,

¹³ O modo *On-line* consiste na concepção de que o usuário precisa estar conectado à rede para utilizar seus serviços. A idéia básica é centralizar as informações em um servidor. Esse servidor tem uma conexão com a rede. A partir do momento em que o usuário se conecta com a rede ele pode fazer uso da aplicação que deseja, ou que lhe é permitida nesse servidor.

¹⁴ No modo *off-line*, as aplicações são instaladas na unidade móvel e executam totalmente fora da rede, atualizando processos em arquivos em lote.

sem intermediação de um PC. O *Mobile Connect* é um complemento natural para DB2E porque oferece conectividade para qualquer banco de dados relacional e sincronização com servidores *Lotus Notes* e *Microsoft Exchange*. O *Mobile Connect* também tem a função de implementação de aplicativos para dispositivos móveis;

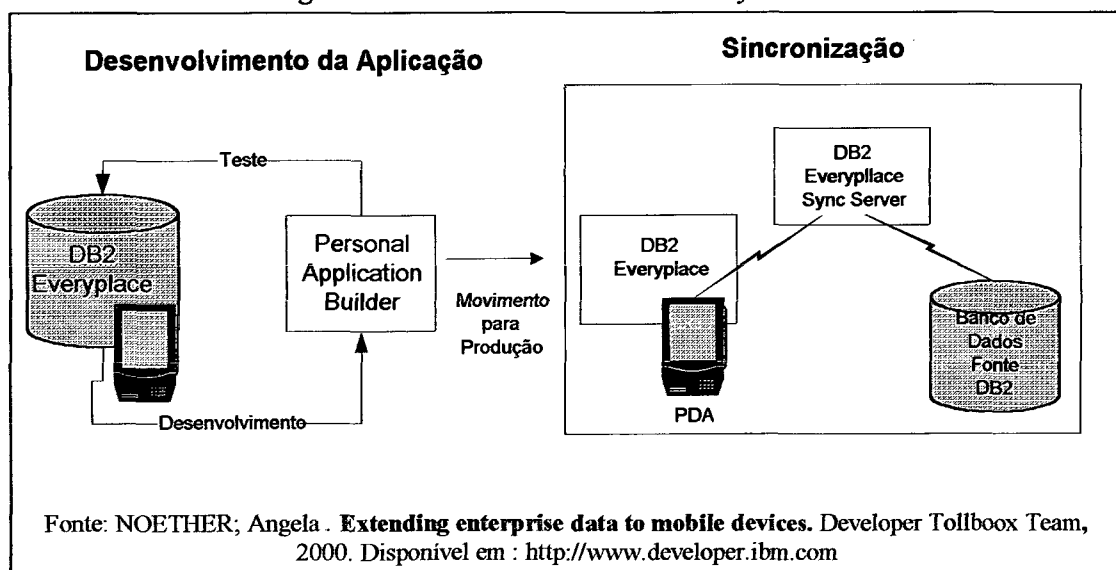
- ⤴ A instalação requer algum conhecimento técnico;
- ⤴ Possui APIs de banco de dados padrões;

O DB2 *Everyplace* é constituído por três componentes principais (NOETHER, 2000):

- ⤴ **DB2 *Everyplace* database:** sistema de banco de dados relacional projetado para dispositivos *handheld* e suas aplicações.;
- ⤴ **DB2 *Everyplace* Sync Server:** servidor de sincronização bidirecional que move dados do DB2 *Everyplace* database e os dados localizados na empresa e vice-versa;
- ⤴ **DB2 *Everyplace* Personal Application Builder:** É uma ferramenta de desenvolvimento de aplicação rápida para criar aplicações DB2 *Everyplace* que executam em dispositivos móveis.

A figura 42, mostra como os três componentes citados acima interagem entre si.

Figura 42 - Funcionamento DB2 Everywhere



6.2.1 **DB2 Everyplace database**

É um sistema de banco de dados relacional projetado para dispositivos *handheld* e suas aplicações. Tem suporte a SQL básico, funções agregadas, operações relacionais, como *Join*, *Group by*, *Order by*, entre outras. *DB2 Everyplace* prove armazenamento local de dados na unidade móvel ou na empresa. Os dados relacionais podem ser sincronizados para a unidade móvel de outras fontes como *DB2 Universal Database for UNIX*, *OS/2* e *Windows NT*, *DB2 for OS/390* e *DB2 for AS/400*.

Não inclui funções como subconsultas, criação de visões, *triggers*, *stored procedures* e funções definidas pelos usuários. Oferece suporte as linguagens *C/C++* e *Java*. O *DB2 Everyplace database* pode ser executado com os seguintes sistemas operacionais:

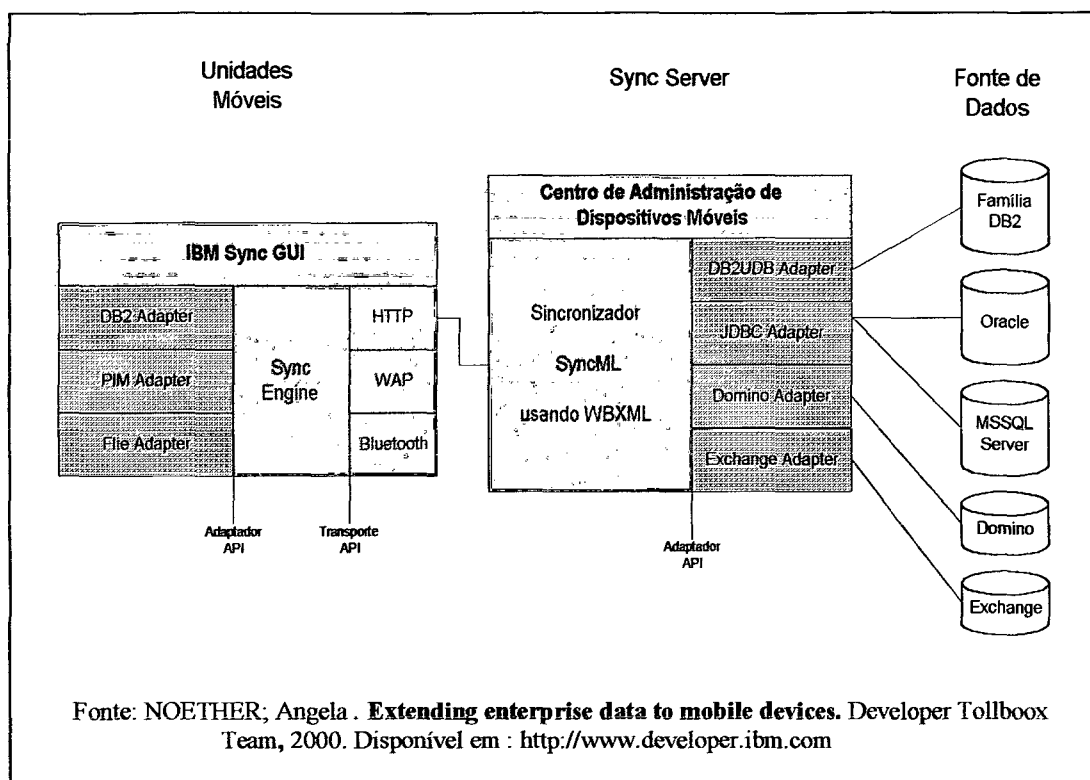
- ♣ *Palm OS* (www.palm.com);
- ♣ *Windows CE/Pocket PC* (www.PocketPC.com);
- ♣ *Psion EPOC* (www.symbian.com);
- ♣ *QNX Neutrino* (www.qnx.com);
- ♣ *Linux*.

6.2.2 **DB2 Everyplace Sync Server**

É um servidor de sincronização bidirecional que utiliza a tecnologia *Java Servlet*. Tem a finalidade de mover dados entre o *DB2* banco de dados de *Everyplace* e o banco de dados da empresa. Sua arquitetura consiste de três partes principais, como mostra a figura 43:

- ♣ *IBM Sync* na unidade móvel ou na empresa;
- ♣ Servidor *DB2 Everyplace Sync* e
- ♣ Fontes de dados na empresa

Figura 43 - Arquitetura DB2 Everyplace



Na figura 43, as áreas azuis da figura mostram as partes de sincronização, uma no servidor e outra na unidade móvel. As áreas cinza representam os adaptadores para as fontes de dados. A área amarela representa as interfaces com o usuário e os cilindros representam as fontes de dados na empresa. Existem também várias camadas API na arquitetura, que permitem a extensão e o acesso aos programas.

O *Sync Server* é uma parte da arquitetura construída com base em uma máquina de sincronização chamada *SyncML*. A *SyncML* é uma iniciativa de indústria patrocinada por várias companhias, como IBM, Lotus, Nokia, Palm, Psion, Motorola, Starfish e Ericsson. Produtos baseados na *SyncML* são designados para interoperar um com o outro, usando uma linguagem de padronização e estrutura de sincronização para replicar e sincronizar dados entre aparelhos.

O IBM *Sync* cliente se comunica com o servidor *DB2 Everyplace Sync Server*, que realiza a comunicação com a fonte de dados na empresa. Para sincronizar os dados, o *Sync Server* inicia uma sessão de sincronização a qual tem dois processos:

1. Usuários móveis submetem mudanças que foram feitas em cópias locais de dados da fonte, realizando a sincronização do cliente para o servidor.
2. Usuários móveis recebem mudanças que foram feitas a dados de fonte que residem no servidor desde a última vez que os dados foram sincronizados, realizando a sincronização do servidor para o cliente.

6.2.3 DB2 Everyplace Personal Application Builder

É uma ferramenta para criar aplicações *DB2 Everyplace* que executam em dispositivos móveis utilizando a XML e a XSL. Suporta pequenas aplicações para *handheld* que têm acesso ao DB2 e ambiente visual para diferentes aparelhos.

O PAB (*Personal Application Builder*) pode ser integrado com outras ferramentas para aplicações, como teste e depuração. (NOETHER,2000) enumera algumas das características:

- ✧ Gerar aplicações em linguagem C que podem acessar tabelas do DB2 Everyplace;
- ✧ Acessar banco de dados sem escrever qualquer código;
- ✧ Criar formulários diretamente das definições das tabelas do banco de dados;
- ✧ Capacidade de prototipação rápida;
- ✧ Editar formulários selecionando controles de uma paleta;
- ✧ Suporte a *scripting* global;
- ✧ Gerar código através de botão e compilação;
- ✧ Integrar com Palm OS e Ferramentas de GNU para compilação, teste e depuração;
- ✧ Exemplos de uso do PBA;
- ✧ Suporte completo a banco de dados relacional e
- ✧ Banco de dados em um único formulário.

6.2.4 Segurança

O DB2 *Everyplace* permite a especificar um grupo de usuários para acessar as aplicações, realizar *downloads* de arquivos e assinatura digital.

A identificação e senha do usuário são autenticadas durante a sincronização. O banco de dados fonte é protegido contra unidades móveis que tentam realizar mudanças de inserção, atualização e exclusão de dados, tendo acesso permitido de acordo com os privilégios que lhe foram oferecidos. O DB2 *Everyplace* usa HTTP para transportar os dados para sincronização, assim como um *firewall* é uma porta especial podem ser ativadas para a sincronização.

6.3 Sybase SQL Anywhere Studio

O Sybase SQL *Anywhere Studio* é um pacote para gerenciamento e sincronização de dados que permite o *design* e distribuição das informações corporativas para grupos de trabalho, dispositivos móveis e sistemas com bancos de dados embutidos. É composto pelo *Adaptive Server Anywhere*, que é banco de dados para aplicações móveis, por um sistema de replicação otimizado para ambientes ocasionalmente conectados e por um conjunto de ferramentas de desenvolvimento e produtividade. A seguir são enumeradas algumas de suas características:

- ⤴ Banco de dados relacional para cada *site* no sistema, usando *Sybase Adaptive Server Anywhere* ou banco de dados *Ultralite*;
- ⤴ Duas formas de sincronização de dados entre o servidor e as unidades móveis;
- ⤴ Entrada de dados em qualquer *site* do sistema;
- ⤴ Operação das unidades móveis em modo desconectado;
- ⤴ Suporte para um grande número de unidades móveis;
- ⤴ Administração central, sem necessidade de administração nas unidades móveis.

O *Sybase SQL Anywhere Studio* é constituído por três componentes principais (SYBASE, 2000): *Sybase Adaptive Server Anywhere*, *Adaptive Server Anywhere Ultralite* e *MobileBuilder*.

6.3.1 Sybase Adaptive Server Anywhere

Denominado anteriormente como *SQL Anywhere*, o *Adaptive Server* foi projetado para aplicações móveis e distribuídas. A replicação nativa possibilita o acesso a redes corporativas e Intranets, de modo que os dados de uma empresa podem ser replicados em máquinas locais, mesmo quando o usuário estiver acessando os dados fora da empresa.

O *Sybase Adaptive Server Anywhere* tem suporte para os seguintes sistemas operacionais:

- ✧ *Windows 95/98, NT, 2000, CE;*
- ✧ *Novell NetWare;*
- ✧ *Solaris/SPARC;*
- ✧ *Solaris/Intel;*
- ✧ *HP-UX;*
- ✧ *IBM AIX;*
- ✧ *Linux e*
- ✧ *UNIX.*

6.3.2 Adaptive Server Anywhere Ultralite

O *Adaptive Server Anywhere Ultralite* é uma tecnologia desenvolvida e patenteada pela *Sybase* que gera bancos de dados otimizados para ambientes móveis e embutidos com recursos limitados, para PDA's, *palmtops*, entre outros. Permite a sincronização e a replicação do banco de dados em modo desconectado. Tem a capacidade de gerar módulos nas unidades móveis contendo apenas o código

necessário para suportar uma aplicação específica, de forma automática. Gera um banco de dados que é acessível através do SQL ANSI.

O *ultralite* é compilado diretamente no banco de dados. Possui duas tecnologias diferentes, mas complementares de sincronização: *SQL Remote* e *MobiLink*.

6.3.2.1 SQL Remote

SQL Remote é uma tecnologia de sincronização baseada em mensagem. Uma instalação de *SQL Remote* consiste em um banco de dados central que armazena uma cópia mestra dos dados e de muitos bancos de dados remotos em dispositivos móveis. O servidor central pode estar executando no *Adaptive Server Anywhere* ou no servidor de banco de dados da empresa e as unidades móveis executam em um *Adaptive Server Anywhere Databases*. A entrada dos dados pode ocorrer no banco de dados central ou em unidades móveis. A sincronização acontece baseada em um sistema de mensagem, como *e-mail* ou transferência de arquivo.

6.3.2.2 MobiLink

MobiLink é uma tecnologia de sincronização bidirecional desenvolvido para complementar *SQL Remote* expandido para sistemas de informação distribuídos.

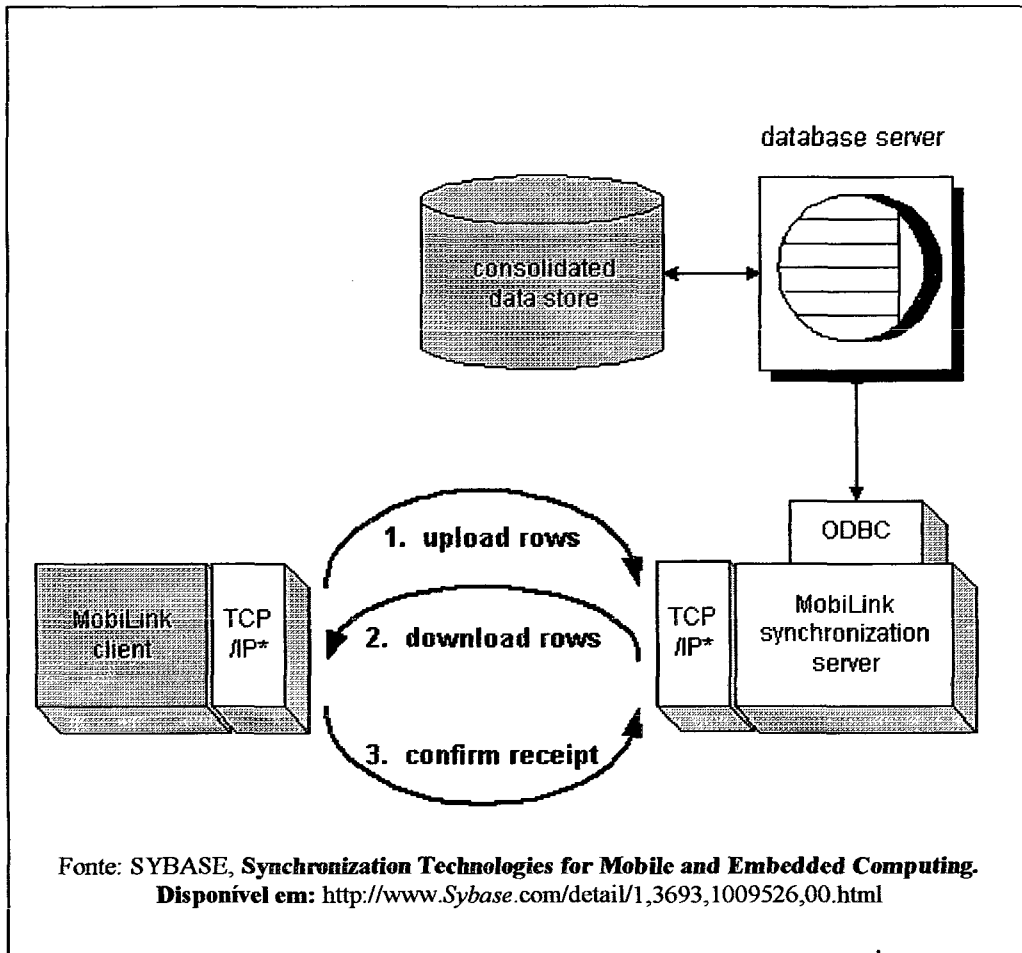
Tem como principais características:

- ✧ Consolidar bancos de dados heterogêneos;
- ✧ Particionar os dados em subconjuntos;
- ✧ Detectar conflitos e solução;
- ✧ Encriptação de dados;
- ✧ Gerenciar grandes bancos de dados;
- ✧ Suporte a *handheld* e dispositivos sem fio.

O *Mobilink* tem em uma forma de sincronização baseada em sessão, como mostra a figura 44, requerendo uma conexão direta com o banco de dados consolidado ou o *Anywhere Server*, utilizando os protocolos TCP/IP, HTTP e *HotSync* ou ondas de rádio.

Além do *Adaptive Server Anywhere (ASA)* e *Adaptive Server Enterprise (ASE)*, o *mobilink* suporta *Oracle*, *Microsoft SQL Server*, e *IBM DB2*.

Figura 44 - Esquema de funcionamento Mobilink



Grandes empresas podem necessitar de mais do que um banco de dados central. Neste caso, a combinação do *SQL remote* e *mobilink* se torna uma boa solução, através de uma instalação *mult-tier*.

A tabela 6 faz uma comparação entre as duas tecnologias de sincronização: *SQL Remote* e *Mobilink*.

Tabela 6 - Diferenças SQL Remote e Mobilink

Características	SQL Remote	MobiLink
Método de Sincronização	Baseado em mensagem	Baseada em sessão
Sincronização bidirecional	Sim	Sim
<i>Sybase</i> ASA ou ASE como banco de dados central	Sim	Sim
Suporte a <i>Oracle</i> , Microsoft, IBM	Não	Sim
<i>Sybase</i> ASA como banco de dados remoto	Sim	Sim
<i>Sybase Ultralite</i> como banco de dados remoto	Não	Sim
Opera em modo desconectado	Sim	Não

Fonte: SYBASE, **Synchronization Technologies for Mobile and Embedded Computing**.
Disponível em: <http://www.Sybase.com/detail/1,3693,1009526,00.html>

6.3.3 **MobileBuilder**

É uma ferramenta visual para desenvolvimento de aplicações, possui uma tecnologia de integração com o *Ultralite* e pode ser utilizado por vários sistemas operacionais como:

- ✧ *Palm OS*;
- ✧ *Pocket PC*;
- ✧ *WinCE*;
- ✧ *Win32 e Win16*;
- ✧ *DOS & DPMI*.

6.4 SQL Server 2000

O SQL Server 2000 da Microsoft, lançado em outubro de 2000, é o primeiro servidor corporativo da empresa a fornecer aplicações para Internet com suporte ao padrão XML, *e-commerce*, *data warehousing* entre outros¹⁵.

O sistema é um gerenciador de dados e de análise para aplicativos e serviços. Possui suporte a outros bancos de dados, como *Oracle* e *Sybase*, recurso não disponível na versão SQL Server 7.0. Apresenta-se em quatro versões: *standart*, *enterprise*, *personal* e *desktop edition*. O sistema conta ainda com uma versão para equipamentos sem fio baseado no sistema operacional *Windows CE*, denominada SQL Sever CE.

6.4.1 O Microsoft SQL Server 2000 Windows CE

O Microsoft SQL Server 2000 Windows CE Edition (SQL Server CE), é um banco de dados compacto voltado para o desenvolvimento de aplicações corporativas baseadas em equipamentos móveis. Trabalha com ferramentas, interfaces de programação de aplicação (APIs) e comandos da linguagem SQL, que visam diminuir o tempo de desenvolvimento de soluções de *software*. Apresenta as seguintes características¹⁶.

- ⤴ Através do Internet *Information Services*, o SQL Server CE viabiliza acesso seguro com padrões Internet ao banco de dados SQL Server, nas versões 6.5, 7.0 e 2000;
- ⤴ Integridade referencial e operações em cascata;
- ⤴ Manipulação de vários tipos de dados;
- ⤴ Índices por tabela e índices multicolumna;
- ⤴ Suporte a transações;
- ⤴ Suporte a replicação de dados;
- ⤴ Comandos SQL (*inner/outer join*, *group by*, *having*);

¹⁵ http://www.microsoft.com/brasil/pr/ms_sql2000.htm

¹⁶ <http://www.microsoft.com/sql/productinfo/ceoverview.htm>

- ⤴ Compressão de dados em tempo de sincronização;
- ⤴ Encriptação de dados.

O SQL *Server CE* oferece uma série de características de bancos de dados relacionais, como processamento otimizado de *queries* e suporte a transações, garantindo um consumo mínimo que preserva recursos valiosos para o sistema operacional *Windows CE*¹⁷.

Além de acesso remoto, a replicação e a proteção de dados sobre o HTTP (*HyperText Transfer Protocol*), o SQL *Server CE* permite que as informações sejam trabalhadas em modo desconectado, para serem sincronizadas com o servidor central quando for necessário. O SQL *Server CE* possui suporte a gerenciamento de dados e possibilidade de conexão, tanto para grandes servidores como para estações de trabalho *PC Desktop*, além de possuir um modelo de programação e um modelo de operação consistente, igualmente aos demais produtos da família SQL *Server*. No SQL *Server CE*, o consumo de memória é pequeno, sendo que todas as suas funcionalidades ocupam aproximadamente 1MB de memória¹⁸.

A tabela tal faz uma comparação com as principais características dos produtos citados anteriormente, destacando aspectos de gerenciamento de dados móveis, como replicação de dados, suporte a transações, segurança, sincronização e principais sistemas operacionais suportados pelos bancos de dados.

¹⁷ <http://www.microsoft.com/sql/productinfo/ceoverview.htm>

¹⁸ <http://www.microsoft.com/sql/productinfo/SQLServerCEEvalguide.doc>

Tabela 7 - Comparação entre diversos Banco de Dados Móveis

BANCOS DE DADOS	EMPRESA	PERFIL	SUPORE À TRANSAÇÕES	REPLICAÇÃO DE DADOS	SEGURANÇA	SINCRONIZAÇÃO	SISTEMA OPERACIONAL SUPORTADO
ORACLE 8I LITE	Oracle	Software de Gerenciamento	Sim	Sim	Encriptação Autenticação	I-Connect	<ul style="list-style-type: none"> ^ Windows CE ^ Windows 95/98/NT ^ PalmOS ^ EPOC
SQL SERVER CE	Microsoft	BD compacto com características	Sim	Sim	Encriptação Autenticação	Acesso Remoto aos dados	<ul style="list-style-type: none"> ^ Windows 2000 DataCenter ^ Windows 2000 Advanced Server ^ Windows 2000 Server/Professional ^ Windows 98/CE/NT
SYBASE SQL	Sybase	Pacote de gerenciamento	*	Sim	Encriptação Autenticação	SQL Remote Mobilink	<ul style="list-style-type: none"> ^ Windows CE/NT ^ Windows 95/98/2000 ^ Novel NetWare ^ Solaris/Intel ^ HP-UX ^ IBM AIX ^ Linux ^ Unix
DB2 EVERY WHERE	IBM	BD relacional	*	*	Autenticação Firewall	Sync Server	<ul style="list-style-type: none"> ^ Palm OS ^ Windows CE/Pocket PC ^ Psion EPOC ^ QNX Neutrino ^ Linux

* Nas fontes pesquisadas não houve referência quanto ao assunto abordado.

CAPÍTULO VII

CONCLUSÃO

7.1 Resumo do trabalho

A computação móvel consiste em um sistema composto por dois tipos distintos de entidades, denominadas computadores móveis e computadores fixos.

Os computadores, unidades ou *hosts* móveis são computadores portáteis, interligados em rede através de ligações sem fio, que permitem a mobilidade estabelecendo uma conexão virtual de qualquer localização dentro de uma área geográfica. Os computadores fixos fazem parte de uma rede fixa cuja localização e conectividade não mudam.

A *wireless computing* se refere a sistemas conectados via ligações sem fio. A comunicação sem fio pode ser feita através de sistema de ondas de rádio, utilizando antenas para transmissão e recepção. Podem utilizar tecnologias como radiofrequência, infravermelho, satélite, microondas, laser, *spread spectrum*, entre outras.

A distribuição do sistema permite às aplicações individuais exercerem um controle local sobre os seus dados de maneira que se tornem menos dependentes de um centro de processamento de dados remoto, possibilitando o acesso aos dados de outras localidades. Um banco de dados distribuídos pode ser mais confiável devido à multiplicidade e a um certo grau de autonomia existentes em suas partes. Os dados podem ser armazenados de maneira distribuída, próximos ao seu ponto normal de uso, reduzindo o tempo de resposta e os custos de comunicação.

O mecanismo de *caching* (armazenamento) em um ambiente móvel tem como objetivo o acesso mais eficiente aos dados, permitindo o trabalho desconectado, possibilitando uma economia de energia e disponibilidade de recursos utilizado a fim de aumentar a probabilidade do dado procurado estar armazenado localmente na unidade móvel, fazendo com que permaneça em memória dados que serão possivelmente reutilizados, pois desta forma, reduzirá a contenção de dados, reduzindo o tempo de repostas a consultas e conseqüentemente melhorará o desempenho do sistema.

A difusão de dados ou *broadcasting* consiste em um sistema de envio de mensagens, onde a mesma mensagem é enviada de uma estação base para todas as unidades móveis que estejam em uma célula. Os dados difundidos podem ser recebidos por um grande número de unidades móveis de uma só vez, sem custos adicionais. Desse modo, a banda de transmissão tem utilização mais efetiva. As unidades móveis podem receber os dados quando eles forem transmitidos, em vez de consumir energia transmitindo uma solicitação, podendo ter disponibilidade local de armazenamento não-volátil para armazenar estes dados no momento de seu recebimento.

Uma transação móvel é solicitada por uma unidade móvel e enviada para a estação base na qual a unidade está associada, não sendo totalmente gerenciada pelo sistema. O movimento de uma unidade móvel controla a execução de uma transação. Sendo assim, é executada seqüencialmente através de várias estações base e em possíveis conjuntos de dados, dependendo do movimento da unidade móvel. Diferentemente de transações realizadas em uma rede fixa, as transações móveis podem acessar dados que estão mudando constantemente de lugar, sendo assim dependente da localização dos dados.

O processamento de consultas em banco de dados móveis exige a localização precisa da unidade móvel. O uso de computadores em navios, aviões, carros entre outros meios de transporte, dificulta esta localização, devido as constantes mudanças de lugar. Consultas ao posto de gasolina mais próximo, hotéis, restaurantes, hospitais e outros tipos de serviços necessários enquanto se esta em movimento, são exemplos de consultas móveis.

A replicação de dados faz com que um determinado objeto de dados (um registro, por exemplo) possa ter fisicamente várias cópias armazenadas em diversas localidades diferentes. Isto proporciona um aumento da disponibilidade de dados melhorando o desempenho do sistema, fazendo com que os dados estejam localizados o mais próximo possível do local são solicitados com alta frequência.

Um sistema em ambiente móvel está sujeito à falhas tanto de *hardware* como de *software*. Em cada um desses casos, informações que se referem ao sistema de banco de dados podem se perder. Uma parte essencial do sistema de banco de dados é um esquema de recuperação responsável pela detecção de falhas e pela restauração do banco de dados para um estado consistente que existia antes da ocorrência da falha. A recuperação é responsável por preservar a consistência do banco de dados após falhas do sistema, de transações ou dos meios de comunicação.

A segurança em aparelhos móveis é um dos maiores problemas em redes sem fio. As vulnerabilidades dos sistemas e dos dados tornam-se maiores e a troca de mensagens pode ser passivamente capturada por alguém não-autorizado. Ondas de rádio viajam abertas pela atmosfera onde podem ser interceptadas por pessoas que estão conectadas constantemente, podendo ser violadas e utilizadas de maneira incorreta.

As tecnologias com suporte a gerenciamento de dados móveis, estudadas neste trabalho, tratam aspectos como sincronização, gerenciamento de transações, replicação, estruturação e segurança do banco de dados, visando facilitar a comunicação realizada entre a unidade móvel e a rede fixa, inclusive durante períodos de desconexão. Dentre elas destacam-se o *DB2 Everywhere*, o *SQL Server CE*, o *Oracle 8i Lite*, e o *SQL Anywhere Studio*.

7.2 Conclusões

O mundo social e corporativo depende fortemente do acesso as informações, como ponto-chave na posição das empresas frente ao mercado competitivo. A utilização de banco de dados móveis propicia facilidades de locomoção física que possibilitam aos usuários a transmissão, atualização ou acesso aos dados a qualquer momento, limitado pela condição de que esteja localizado dentro de uma área geográfica determinada.

Por se tratar de um assunto relativamente novo, o ambiente móvel representa um desafio para mundo tecnológico. No campo dos desafios técnicos, um dos grandes problemas é a capacidade de armazenamento de energia, ainda pequena nos dias atuais. Outro fator é a confiabilidade da utilização da computação sem fio em operações que requerem elevado grau de segurança, pois, as informações são transmitidas pelo ar, através de ondas eletromagnéticas, fazendo com que a confiabilidade seja questionada. A mobilidade também ocasiona uma série de problemas, tais como, altas taxas de erros, baixa velocidade de comunicação, desconexões freqüentes, baixa largura de banda, heterogeneidade de tecnologias sem fio, ausência de integração entre as redes móveis, etc.

Embora a capacidade dos equipamentos móveis esteja crescendo consideravelmente, a memória disponível e o poder de processamento são recursos muito valiosos na computação móvel. Desta forma, um sistema de banco de dados deve ser o mais compacto possível, para que possa ser utilizado em dispositivos móveis.

No campo das aplicações e equipamentos, muito se tem discutido a respeito das unidades móveis e mecanismos que permitam a elas se adaptarem às alterações do ambiente, através de SGBD-D adequados, protocolos de acesso homogêneo, acesso uniforme à informação sobre localização, endereço universal para unidades móveis, alocação automática de largura de banda, baterias mais leves com mais tempo de energia, entre outros fatores.

7.3 Relevância do Trabalho

Sistemas de gerenciamento de bancos de dados móveis tem uma relevância destacada para trabalhadores com alta mobilidade que necessitam acessar bases de dados remotas e executar operações de diversas localizações. A possibilidade de acessar dados em qualquer ambiente, traz a flexibilidade para utilização em diversas aplicações que exijam movimento, fazendo com que exista uma disponibilidade dos dados, independente da localização do usuário.

A contribuição deste trabalho para a solução dos problemas de gerenciamento de dados móveis foi um estudo realizado sobre os problemas e soluções existentes sob a

visão de diferentes autores. As alternativas propostas para lidar com sistemas de banco de dados móveis são de importância vital para o mundo corporativo. A partir dos problemas e soluções apresentadas, poderão surgir novas propostas tratando as diversas restrições impostas pelo ambiente móvel.

7.4 Perspectivas Futuras

Acredita-se que futuramente os preços de computadores pessoais e portáteis cairão até que se tornem objetos de compras por impulso em qualquer supermercado. Tornarão-se dispositivos multifuncionais e inteligentes oferecendo segurança e qualidade de comunicação. A interatividade, personalização de serviços, trabalho cooperativo e universalização de acesso a qualquer hora e qualquer lugar, serão constantes na vida das pessoas. Aparelhos móveis, estarão mais acessíveis sendo conectados a todos os lugares possíveis, oferecendo serviços de qualidade, dissolvendo fronteiras entre países, empresas, fusos horários, instituições e pessoas.

GLOSSÁRIO

ATM - *Asynchronous Transfer Mode* - tecnologia de redes, dados, voz e imagem voltada a aplicações que necessitam de alta taxa de transmissão e/ou multimídias. As velocidades mais comuns para transmissões ATM são de 25 Mbps, 155 Mbps e 622 Mbps.

API - *Application Programming Interface* - é um conjunto normalizado de rotinas que podem ser referenciadas por um programa aplicativo para acessar serviços essenciais de uma rede.

BANDWIDTH Largura de Banda - é a velocidade máxima que com os dados são transmitidos entre computadores por meio de uma rede.

BINDING- Associação entre *home address* e *care-of address*

BROADCAST- Sistema de envio de mensagens, onde a mesma mensagem é enviada de uma estação base para todas as unidades móveis que estejam em uma célula.

BSC- *Base Station Controller* - Tem a função de administrar os recursos de canais de rádio e execução de *handoffs*.

CACHING – Tem como objetivo o acesso mais eficiente aos dados, sendo utilizado a fim de aumentar a probabilidade do dado procurado estar armazenado localmente na unidade móvel, fazendo com que permaneça em memória dados que serão

possivelmente reutilizados, de modo a reduzir a contenção de dados e o tempo de repostas a consultas, melhorando com isso, o desempenho do sistema.

CARE-OF ADDRESS – Endereço associado à unidade móvel a cada ponto de acesso

CDMA - Tecnologia de telefone celular digital. Sigla de *Code-Division Multiple Access*, ou Acesso Múltiplo por Divisão de Código, no qual a transmissão é feita na faixa de 9,6 a 14,4 Kbps, em diferentes frequências. O CDMA da terceira geração, também chamado CDMA 2000, permitirá o tráfego de dados a até 384 Kbps.

CDPD - *Cellular Digital Packet Data* - Técnica que permite ampliar a capacidade de um sistema celular analógico, permitindo envio de dados a 19,2 Kbps.

CHECKPOINT - Ponto de Recuperação – Tem a função de armazenar o estado do processo em um *site* conhecido ou em um computador próximo da localização atual da unidade móvel. Pode também manter a trilha dos lugares onde o estado do processo foi salvo.

EPOC - É um sistema operacional utilizado principalmente em pequenos telefones portáteis computadorizados, e na linha de *handhelds* conhecida por *Psion*, da *Symbian*, uma parceria com a Ericsson, Nokia e Motorola. Possui compatibilidade com comunicação *wireless* e capacidade de adicionar novos programas. É um sistema operacional de 32 bits com multitarefa e utiliza uma caneta como GUI – *Graphic User Interface* (Interface gráfica de interação com o usuário).

FIREWALL - Uma combinação de *hardware* e *software* desenvolvida especialmente para impedir o acesso de usuários não-autorizados a informações do sistema. É utilizado para separar da Internet a rede de uma empresa.

GATEWAY - É o ponto de acesso a uma rede de comunicação ou interligação entre duas redes. Equipamento utilizado para estabelecer a conexão e fazer a conversão entre duas redes, tanto em termos de *hardware* quanto de *software*.

FOREIGN AGENT- Agente estrangeiro - coloca o *Home address* como endereço de destinatário

HANDOFF- É um processo entre duas estações base adjacentes que garante uma transação enquanto usuários se movimentam de uma célula para outra mantendo a conectividade de uma unidade móvel com a rede fixa. Isto implica que uma estação base pode detectar qualquer entrada e qualquer saída de usuários dentro e fora da sua célula.

HLR - *Home Location Register* - É o servidor de localização, onde cada unidade móvel deverá ser permanentemente registrada.

HOARDING - É um processo que acontece quando os dados utilizados durante uma operação são pré-carregados para a unidade móvel, tornando-se inacessíveis para outros sites.

HOME ADDRESS - É o endereço local associado permanentemente à unidade móvel

HTML - *Hyper Text Markup Language* - Linguagem usada para construir páginas Web.

HUB - é um ponto de conexão central para o cabeamento de uma rede. Permite conectar dois ou mais segmentos *Ethernet*. Capta sinais, repete-os e amplifica-os para todas as portas que lhe estão conectadas.

LOCAL LINK ADDRESS- Endereço de ligação local associado à sub-rede

MSC - *Mobile Switching Center* - Provê a troca de funções, coordena o registro de localização entre outras funções.

PDA - *Personal Digital Assistants* - São computadores lançados a partir da década de 90, como *handhelds* e *palmtops*.

QoS - *Quality of Service* - Termo que especifica uma garantia de desempenho na rede. A grande vantagem da tecnologia ATM é de suportar níveis de QoS. As redes *Ethernet* atuais já contemplam protocolos e campos que atendem a essa necessidade.

REDES AD-HOC- Consiste em redes onde não existe infraestrutura fixa. Todas as unidades móveis são capazes de se comunicar diretamente entre si, não havendo pontos de acesso, ou seja, não há a necessidade de estações base.

SQL - *Structured Query Language* - é uma linguagem específica para comunicação com Bancos de Dados Relacionais. Seu objetivo é fornecer uma interface de alto nível ao usuário.

SQLJ - SQL in Java.

SITE – Representa um microcomputador, uma estação de trabalho, um minicomputador, entre outros. Esses processadores são chamados por diversos nomes, como nós, computadores ou *sites*, dependendo do contexto no qual eles são mencionados.

STORED PROCEDURES - São porções de código SQL que ficam armazenados no servidor de banco de dados para serem utilizados quando necessário. Estas porções de código podem ser chamadas a qualquer momento, podem receber parâmetros ou não e podem retornar valores ou não.

SPREAD SPECTRUM - É tecnologia de transmissão mais utilizada em *Wireless* LANs, por ser considerada menos sensível a interferências do meio que as outras tecnologias. Atravessa obstáculos com mais facilidade, por utilizar frequências mais baixas, sendo portanto, mais fácil de ultrapassar barreiras como paredes

SWITCHES - Aparelho que tem a função de conectar diferentes redes. Mapeia endereços Ethernet dos nós que residem em cada segmento da rede e permite apenas a

passagem do tráfego necessário. Quando recebe um pacote, determina qual o destino e a origem deste, e encaminha-o para a direção certa. O *Switch* abandona o pacote se a origem e o destino são no mesmo segmento de rede. Existem duas arquiteturas básicas de *switches* de rede: *cut-through* e *store-and-forward*. Os *Cut-through* apenas examinam o endereço de destino antes de reencaminhar o pacote enquanto que o *store-and-forward* aceita e analisa o pacote inteiro antes de o reencaminhar.

TCP – *Transmission Control Protocol* - É um padrão de protocolos que governa o funcionamento da Internet.

TDMA - Tecnologia para serviços digitais sem fio. Sigla de *Time-Division Multiple Access*, ou acesso múltiplo por divisão de tempo, baseia-se na divisão de uma frequência de rádio em frações de tempo, as quais são usadas em diferentes conexões.

TRIGGERS - São como *stored procedures*, porções de código que ficam armazenadas no servidor, mas não podem nem receber parâmetros nem retornar valores, nem ser chamadas a qualquer momento. As triggers são chamadas pelo próprio banco de dados no momento de um evento pré-definido.

VLR - *Visitor Location Register* - São os registros de localização de visitante feitos em cada célula quando o usuário se mover para um novo local.

XML - *Extensible Markup Language* – É linguagem uma *meta-markup language* que provê um formato para descrever dados estruturados. Ela facilita declarações mais precisas do conteúdo e resultados mais significativos de busca através de múltiplas plataformas. O XML permite a definição de um número infinito de *tags*.

WAP - *Wireless Application Protocol* - Protocolo para aplicações sem fio. Provê um padrão universal para trazer o conteúdo da Internet e serviços agregados para telefones móveis e outros dispositivos sem fio, como telefones celulares, PDAs (*Personal Digital Assistants*), rádios, *paggers*, entre outros.

WARD – É um modelo que faz a combinação do modelo *peer-to-peer* e do modelo cliente/servidor na replicação de dados.

WINDOWS CE - é uma categoria de sistema operacional, lançada pela Microsoft, para ser utilizada em computadores portáteis.

WML- *Wireless Markup Language* - É a linguagem de programação adotada pelo WAP, semelhante à linguagem utilizada pela Internet. É baseada no padrão XML, lido e interpretado por um *microbrowser* (micro-navegador) instalado num dispositivo móvel contendo o WAP.

WIRELESS- refere-se a sistemas conectados a um ambiente de trabalho via ligações sem fio.

WIRELESS LAN - *Local Area Network* - é uma rede local que pode ser utilizada onde haja necessidade de mobilidade dos pontos da rede ou existam dificuldades de cabeamento

WIRELESS MAN - *Metropolitan Area Network* - é uma rede metropolitana situada geralmente dentro de uma cidade. As MAN's são menores que as redes de longa distância (ou WAN's), porém maiores do que as locais ou LANs.

WIRELESS WAN - *Wide Area Network* – É uma rede de comunicação sem fio que se estende por uma área geograficamente considerável.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 ALONSO, Rafael; FRANKLIN, M.; ZDONIK, Stanley; ACHARAYA, Swarup. **Broadcast Disks: Data Management for Asymmetric Communication Environments**. Proceedings of the ACM SIGMOD Conference, San Jose, CA, Maio 1995.
- 2 _____. **Are "Disks in the Air" Just Pie in the Sky?**, Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, Dezembro, 1994
- 3 ANINDYA, Datta; DEBRA, E; VANDER, Meer; ASLIHAN, Celik; VIJAY, Kumar. **Broadcast Protocols to Support Efficient Retrieval from Databases by Mobile Users**. Electronic Edition. V. 24, N. 1, Março, 1999, p. 1-79.
- 4 ARAÚJO, Luciano V. de; FERREIRA, João Eduardo. **Cachê Semântico para Computação Sem Fio Baseado na Abstração de Composição dos Dados**. WorkSIDAM, Workshop de Sistemas de Informação Distribuída de Agentes Móveis. São Paulo, Outubro, 2000. p. 83-89
- 5 APICELLA, Mario. **DB2 da IBM equipa PDAs**. InfoWorld Test Center, 2000. Disponível em: <http://www.uol.com.br/computerworld/technology/9909/9909db2e>
- 6 BADRINATH, B. R.; PHATAK, Shirish. H. **An Architecture for Mobile Databases**. Relatório Técnico DCS-TR-351, 1998, p. 1-8.
- 7 BADRINATH, B. R.; ARUP Acharya. **Checkpointing distributed applications on mobile computers**. 3rd IEEE International Conference on Parallel and Distributed Information Systems, Outubro, 1994, p.1-4.
- 8 BADRINATH, B. R.; IMIELINSKI, T.; **Replication and mobility**, 2nd IEEE Workshop on Management of replicated data, Novembro.1992, p. 1-12
- 9 _____. **Querying in highly mobile distributed environments**, Proc. of the 18th Int'l Conf. on VLDB, 1992, p. 41-52.
- 10 _____. **Multiversion reconciliation for mobile databases**, In Proceedings of the ICDE 99, Março, 1999, p. 582-589.

- 11 BARBARA, Daniel; IMIELINSKI, T; **Sleepers and Workholics, Caching Strategies in Mobile Environments**. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 1994, p. 1-12.
- 12 BATES, Regis J; GREGORY, Donald; **Voice and Data Communications Handbook** – McGraw-Hill Series on Computer Communications, 1997
- 13 BERNSTEIN, Philip, A; HADZILACOS, Vassos. **Concurrency Control and Recovery in Database Systems**. ISBN 0-201-1071 5-5, Addison-Wesley 1987.
- 14 BOSZORMENYI, Laszlo; CARSTEN, Weich; **Simple and Efficient Transactions for a Distributed Object Store**, In: Proceedings of Database and Expert Systems Application (DEXA'98),IEEE Computer Society, Vienna,1998, p. 693-698.
- 15 Bibliografia sobre Computação Móvel,
<http://www.di.ufpe.br/~cak/movel/movel.html>, 05/01/01
- 16 CHAN, Boris Y.; SI, Antonio; LEONG, Hong V. **Cache Management for Mobile Databases: Design and Evaluation**. 14th International Conference on Data Engineering. Orlando, (ICDE)., Florida, Fevereiro,1998, p. 54-63.
- 17 CHIU, Lin; LIU, Ming T.. **High level specification of concurrency control in distributed database systems**. Relatório Técnico, IEEE., 1988, p. 309-311.
- 18 COMPUTAÇÃO MÓVEL,
<http://www.lecom.dcc.ufmg.br/~sergiool/telefonia/compmovel.htm>, 01/07/00
- 19 COMPUTER WORLD, <http://www.computerworld.com/home/features.nsf>,
10/04/00
- 20 DESPANDE, P.M; RAMASAMY, K.; SKUKLA, A.; NAUGHTON, J. F. **Caching Multidimensional Queries Using Chunks**. Procedente de SIGMOD, 1998, p. 259-270.
- 21 ELMAGARMID, Ahmed; JING, Jin; BUKHRES, Omran. **An Efficient and Reliable Reservation Algorithm for Mobile Transactions**. Proceedings of the 4th International Conference on Information and Knowledge Management (CIKM), 1995, p. 1-7 .
- 22 DBMS ONLINE - <http://www.dbmsmag.com/9709d17.html>. Setembro, 1997
- 23 DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. Rio de Janeiro: Ed. Campus, 1991.
- 24 DUNHAM, Margaret H; Helal A. **Mobile Computing and Databases: Anything New?**. SIGMOD Record, v. 24, n. 4, dezembro, 1995, p. 5-9.

- 25 DUNHAM, Margaret H; KUMAR, V. **Defining Location Data Dependency, Transaction Mobility and Commitment**. Relatório Técnico 98-CSE-1, fevereiro, 1998, p. 1-22.
- 26 _____ **Location Dependent Data and its Management in Mobile Databases**. In Proceedings of IEEE Computer Society, DEXA Workshop, 1998, p.1-18.
- 27 _____ **Impact of Mobility on Transaction Management**. In Proc. of MobiDE /Mobicom99 Workshop, ACM, Agosto 1999. p. 14-- 21
- 28 COMO FUNCIONAM OS APARELHOS CELULARES.
http://www.gradiente.com.br/celular/agregado/a_respostas.asp, 06/01/01
- 29 COMPUTAÇÃO MÓVEL, <http://www.lecom.dcc.ufmg.br>, 01/07/00
- 30 ELMAGARMID, Ahmed; RUSINKIEWICZ, Marek; SHETH, Amit. **Management of Heterogeneous and Autonomous Database Systems**. Morgan Publishers, 1999
- 31 FASBENDER, A.; HOFF, S.; PIETSCHMANN, M.; **Mobility Management in Third Generation Mobile Networks**, Proceedings of the IFIP TC6 International Workshop on Personal Wireless Communications, Prague, Abril,1995,p.1-8. Disponível em: <http://www-i4.informatik.rwth-aachen.de/research/publications/publications.html>
- 32 FEDERRATH, Hannes; **Protection in Mobile Communications**. in: Günter Müller, Kai Rannenber (Ed.): *Multilateral Security in Communications*, Addison-Wesley-Longman, 1999, p. 349-364.
- 33 FEDERRATH, Hannes; JERICHOW, Anja; KESDOGAN, Dogan; PFITZMANN, Andreas; **Security in Public Mobile Communication Networks**, Proc. Of the IFIP TC 6 International Workshop on Personal *Wireless* Communications, Verlag der Augustinus Buchhandlung Aachen, 1995, p. 105-116.
- 34 FERREIRA, E. João; FINGER, Marcelo. **Controle de concorrência e distribuição de dados: a teoria clássica, suas limitações e extensões modernas**. Escola de computação 2000.
- 35 FIRST INTERNATIONAL CONFERENCE, MDA'9; (1748: 1999:china). **Anais. Mobile Data Access**: Springer, 1999.
- 36 FRANKLIN, M; ZDONIK, S.; **Balancing Push and Pull for Data Broadcast**, Proceedings of the ACM SIGMOD International Conference on Management, Tuscon, 1997

- 37 _____ **Disseminating Updates on Broadcast Disks**, Proceedings of the 22nd VLDB Conference,), India, 1996
- 38 GRAY, Jim; HELLAND, Pat; NEIL, O' Patrick; SHASHA, Dennis; **The dangers of replication and a solution**, In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, 1996, p. 173-182.
- 39 HELAL, Abdelsalam; HASKELL, Bert; CARTER, Jeffery, Carter L.; BRICE, Richard; **Any Time, Anywhere Computing**, Kluwer Academic Publishers, EUA, 1999.
- 40 HOLLIDAY, J.; AGRAWAL, D.; ABBADI, A. **Exploiting Planned Disconnections in Mobile Environments**, 10th International Workshop on Research Issues on Data Engineering (RIDE 2000), Fevereiro, 2000, p. 1-5.
- 41 HUANG, Y.; SISTLA, A. P.; WOLFSON, O. **Data Replication for Mobile Computers**. In SIGMOD Conference, 1994, p. 13-23.
- 42 IMIELINSKI, T.; BADRINATH, B. R. **Data Management for Mobile Computing**, In SIGMOD RECORD, V. 22, N. 1, p. 34-39 , Março, 1993.
- 43 _____ **Wireless Mobile Computing : Solutions and Challenges in Data Management**. In Communications of the ACM, v. 37, n.10, Outubro, 1994, p. 5-30.
- 44 IMIELINSKI, T.; BADRINATH, B. R; VISWANATHAN, S. **Energy Efficient Indexing on Air**. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 1994, p. 25-36
- 45 _____ **Power Efficient Filtering of Data on Air**. In Proceedings of the 4th International Conference on Extending Database Technology, março, 1994.
- 46 JING, Jing; HELAL, A; ELMAGARMID, A. **Client-Server Computing in Mobile Environments**. ACM Press, V.31, N. 2, Junho, 1999, p. 125-135.
- 47 KAYAN, Ersan; ULUSOY, Ozgur. **Real-time Transaction Management in Mobile Computing Systems**. Proceedings of the 6th International Conference on Database Systems for Advanced Applications, 1998
- 48 KRISHNA, P. **Performance Issues in Mobile Wireless Networks**. Dissertação–Texas University, Agosto,1996.
- 49 KORTH, Henry F.; SILBERSCHATZ, Abraham. **Database Systems Concepts**. New York: McGraw-Hill, 1999.
- 50 KOTTKAMP, Hans-Erich; ZUKUNFT, Olaf; **Location-Aware Query Processing in Mobile Database Systems**; . 13th Symposium on Applied Computing (ACM SAC98)., Atlanta, USA. ACM Press., 1998, p. 416-423.

- 51 LAUZAC, Susan W; CHRYSANTHIS, Panos K. **Programming Views for Mobile Clients**. Dexa Workshop, p. 1-6, 1998.
- 52 LEE, Wang-Chien; HU, Qinglong; LEE, Dik Lun. **A study on channel allocation for data dissemination in mobile computing environments**. 1999, p.177-129.
- 53 LEONG, Va Hong; LEE Wang-Chien. **Mobile Data Access**. First International Conference, MDA'99. **Anais**, 1999.
- 54 MATEUS, Geraldo Robson; LOUREIRO, Antonio A. Ferreira. **Introdução a Computação Móvel**. DCC/IM, COPPE/Sistemas, NCE/UFRJ, 11^a. Escola de Computação, 1998.
- 55 MELO, Rubens N.; SILVA, Sidney Dias da; TANAKA Aterio K.. **Banco de dados em aplicações cliente-servidor**. Rio de Janeiro: Infobook, 1997.
- 56 MICRSOSOFT SQL SERVER.
<http://www.microsoft.com/brasil/sql/productinfo/prodover.stm>, 17/01/01
- 57 MOBILE INFO, http://www.mobileinfo.com/mobileinfo_wireless_lans.htm,
21/06/00
- 58 MOBILE INFO, http://www.mobileinfo.com/mobileinfo_secutity_issues.htm,
03/09/00
- 59 MULLINS,Craig S.. **You Can Take It With You: Mobile Databases**.2000.
Disponível em:
http://www.db2mag.com/db_area/archives/2000/q3/mullins.shtml
- 60 NEC DO BRASIL S.A, <http://www.nec.com.br>, 10/ 05/1999
- 61 NOETHER; Angela . **Extending enterprise data to mobile devices**. Developer Tollbox Team, 2000. Disponível em : <http://www.developer.ibm.com>
- 62 NASSU, Eugenio Akihiro; FINGER, Marcelo. **O Significado de “Aqui” em Sistemas Transacionais Móveis**, I Workshop SIDAM (Sistemas de Informação Distribuída de Agentes Móveis), outubro, 2000, p. 55-63.
- 63 NEVES, Nuno; FUCHS, W. Kent. **Adaptive Recovery for Mobile Environments**. Communications of the ACM, v.1, n. 40, Janeiro, 1997, p. 69-74.
- 64 O DB2 EVERYPLACE. <http://www-ibm.com/software/data/db2/everyplace.html>,
30/12/2000
- 65 ORACLE. <http://www.Oracle.com>, 31/12/2000

- 66 ORACLE8i LITE FOR WINDOWS CE,
http://technet.Oracle.com/products/8i_Lite/pdf/Ceds99.pdf, 15/12/2000
- 67 OZU, M. Tamer; VALDURIEZ, Patrick. **Principles of distributed databases systems**. Upper Saddle River, ed. Prentice-Hall, New Jersey, 1999.
- 68 PRADHAN, D. K.; KRISHNA, P.; VAIDYA, N. H. **Recovery in Mobile Environments: Design and Trade-Off Analysis**, 26th International Symposium on Fault-Tolerant Computing (FTCS-26), June 1996.
- 69 PITOURA, Evaggelia; BHARGAVA, Bharat. **Building Information Systems for Mobile Environments**. Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, Maryland, ACM Press. p. 371-378, Novembro, 1994.
- 70 PITOURA, Evaggelia; SAMARAS, George. **Data Management for Mobile Computing**, Kluwer Academic Publishers, 1998.
- 71 PITOURA, Evaggelia; BHARGAVA, Lambda Bharat. **Revising Transaction Concepts for Mobile Computing**. 1st IEEE Workshop on Mobile Computing Systems and Applications (MCSA94), Dezembro, 1994, p. 164-168.
- 72 PFITZAMNN, Andreas; PFITZMANN, Birgit; SCHUNTER, Matthias; WAIDER, Michael; **Mobile User Devices and Security Modules: Design for Trustworthiness**, Relatório Técnico, RZ 2784, Germany, 1996, p. 1-13.
- 73 RAKOTONIRAINY, Andry. **Adaptable Transaction Consistency for Mobile Environments**. Ninth International Workshop on Database and Expert Systems Applications, Vienna, Proceedings. IEEE Computer Society, ISBN 8186-8353-8, Austria, 1998
- 74 RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Database Management Systems**. Mcgraw-Hill, 2000
- 75 RATNER, David; REIHER, Peter; POPEK, Gerald. **Roam: A scalable Replication System for Mobile and Distributed Computing**. Relatório Técnico UCLA-CSD-970044, Los Angeles, janeiro, 1998, p.1-8.
- 76 RATNER, David; REIHER, Peter; POPEK, Gerald; KUENING, Geoffrey. **Replication Requirements in mobile environments**. First Dial M for mobility. outubro, 1997, p.1-4.
- 77 _____ . **Peer Replication with Selective Control**. Relatório Técnico, CSD-960031, Los Angeles, Julho, 1996.
- 78 REVISTA INFO EXAME. São Paulo: Editora Abril, 2000. Maio.
- 79 REVISTA INFO EXAME. São Paulo: Editora Abril, 2000. Junho.

- 80 REVISTA INFO EXAME. São Paulo: Editora Abril, 1999. Setembro
- 81 SHORES, Redwood. **Reviewr's Guide Oracle8i™ Lite**. Oracle Corporation, CA 94065, 2000. Disponível em : <http://www.Oracle.com>
- 82 SILVA, Francisco José da Silva e; ENDLER, Markus. **Requisitos e Arquiteturas de Software para Computação Móvel**. I Workshop SIDAM (Sistemas de Informação Distribuída de Agentes Móveis), outubro 2000.
- 83 STAYANARAYANAN, M. **Fundamental Challenges in Mobile Computing**. Relatório Técnico. CMU-CS-96-111, Philadelphia, 1999, p. 1-7.
- 84 _____. **Mobile Information Access**. IEEE Personal Communications, V. 3, N. 1, Pittsburgh, 1996, p.1-8
- 85 SATYANARAYANAN, M.; NARAYANAN, D.; TILTON, J.; FLINN, J.; WALKER, K. **Agile Application-Aware Adaptation for Mobility**, Proceedings of the Sixteenth ACM Symposium on Operating Systems., 1997
- 86 SSU, Kuo; YAO, Bin; FUCHS, Kent; NEVES, Nuno Ferreira; **Adaptive Checkpointing with Storage Management for Mobile Environments**, manuscript, Dezembro 1998. Disponível em <http://citeseer.nj.nec.com/context/582387/0>
- 87 SYBASE ANYWHERE. <http://www.Sybase.com/solutions/mobilewireless>, 31/12/2000
- 88 SYBASE, **Synchronization Technologies for Mobile and Embedded Computing**. Disponível em: <http://www.Sybase.com/detail/1,3693,1009526,00.html>, 29/12/2000
- 89 SYBASE, **SQL Anywhere Studio 7.0 Data Sheet**. Disponível em: <http://www.Sybase.com/detail?id=100262>, 30/12/2000
- 90 TANEMBAUM, Andrew S., **Redes de Computadores**, Rio de Janeiro. Campus, 1997
- 91 _____. **Sistemas operacionais modernos**, Rio de Janeiro: Prentice – Hall do Brasil, 1995.
- 92 TELEFONIA CELULAR NO BRASIL.
<http://www.celularpress.com.br/partel.htm>, 06/01/01
- 93 TEWARI, R.; Grillo, P.. **Data Management for Mobile Computers in Internet**, Proceedings of the 23rd ACM Computer Science Conference, Março, 1995, p. 246-252.

- 94 ZUKUNFT, Olaf; SAUER, Stefan; **Event processing in mobile and active database systems using broadcasts**, Proc. 9th Intl. Workshop on Database and Expert Systems Applications (DEXA 98), Austria, IEEE Computer Society, 1998, p. 420-426.
- 95 WALBORN, D. Gary; CHRYSANTHIS, K. Panos. **Transaction Processing in PRO-MOTION**. Proc. Of the Symposium on Applied Computing. ACM SAC V. 11, Março, 1998, p.389-398.
- 96 _____ **Supporting Semantics-Based Transaction Processing in Mobile Database Applications**. Proceedings of the 14th Symposium on Reliable Distributed Systems (SRDS-95), Setembro, 1995.
- 97 WEISER, Mark. **The Computer for the Twenty-First Century**. Scientific American Ubicomp Paper after Sci Am editing one more final edit from me to go, 1991.
- 98 WEB FÓRUM, <http://www.webproforum.com/wap/topic01.html>, 20/05/00
- 99 WONG, Man-Hon; LEUNG, Wing-Man; **A Caching Policy to Support Read-only Transactions in a Mobile Computing Environment**, Relatório Técnico CS-TR-1995-07 Maio, 1995, p. 1-12. disponível em <http://citeseer.nj.nec.com/159417.html>
- 100 WORKSIDAM, WORKSHOP DE SISTEMAS DE INFORMAÇÃO DISTRIBUÍDA DE AGENTES MÓVEIS (1.: 2000:São Paulo). **Anais. Métodos e Serviços para Computação Móvel**. São Paulo: IME-USP, 2000.
- 101 WU, Shioh-yang; CHANG, Yu-tse. **An Active Replication Scheme for Mobile Data Management**. Proceedings of the Sixth International Conference on Database Systems for Advanced Applications (DASFAA), IEEE Computer Society. Hsinchu, Taiwan, 1999, p. 143-150.
- 102 UFRGS - Grupo de Redes. **Transmissão de Dados sem Fio**. <http://penta.ufrgs.br/redes.94-2/lisianeh/wireless.html>. UFRGS, Porto Alegre, 1996.
- 103 YEO, L. H.; ZASLAVSKY, UM. **Submissão de Transações de Workstation Móvel em um Multidatabase Processing Cooperativo Ambiente**. Em Procedimentos da 14ª Conferência Internacional em Sistemas de Computação Distribuídos, Junho, 1994, p. 1-6.