

Capítulo

3

Banco de Dados para um Ambiente de Computação Móvel

Sérgio da Costa Côrtes e Sérgio Lifschitz

Abstract

Portable computers have been used in many different application domains and are, together with communication networks, the underlying technology for mobile computing. Database Systems (DBS) provide reliability, security and availability for data, besides integrity control and efficient data access. These DBS can either be located in conventional (fixed) networks as common data servers for many applications, or may serve global and local data for mobile computers and related technology. The latter defines a new role for Database Management Systems (DBMS), which might have to be adapted to answer new requirements due to the mobile environment. Indeed, these include support for unstable networks, weak connectivity and limited resources. This work is concerned with the main problems one may find when DBS are used in mobile computer systems. We present solutions and approaches used both in academy and at the industry. We give also an overview of ongoing research in this area. Particularly, we focus on systems architectures, transaction models, data management and mobile users location.

Resumo

Computadores portáteis são cada vez mais utilizados em diversos domínios de aplicações. Juntamente com as tecnologias das redes de comunicações provêem a base do hardware para a computação móvel. Os Sistemas de Bancos de Dados (SBD) e seus componentes básicos fornecem todos os mecanismos para confiança, segurança, disponibilidade, integridade e acesso eficiente a suas bases de dados. Assim, os SBDs podem estar localizados tanto nas redes fixas, desempenhando um papel tradicional de servidores de dados para todas as aplicações, como podem, também, desempenhar um novo papel de servidores de dados globais e locais, agora residindo em um equipamento móvel. Este novo papel dos SBDs, exigirá um novo comportamento e adaptações nos Sistemas de Gerência de Bancos de Dados (SGBDs), para que os mesmos possam desempenhar seus papeis na administração dos dados, em face às características do ambiente móvel. Entre estas características se destacam as constantes desconexões voluntárias ou involuntárias dos equipamentos móveis de suas redes, a fraca conectividade das redes sem fio nas quais os equipamentos móveis estão conectados e os poucos recursos dos equipamentos que hospedarão os SGBDs Móveis. Este trabalho apresenta os SBDs para o ambiente de computação móvel, os principais problemas encontrados, as soluções hoje utilizadas acadêmica e comercialmente e algumas das pesquisas em andamento. Vamos dar destaque em particular para assuntos tais como arquiteturas de sistemas, modelos de transações, gerência de dados e localização de usuários móveis.

1. Introdução

Diversas atividades que envolvem alguma interação com um **Sistemas de Bancos de Dados (SBD)** estão presentes no nosso dia-a-dia. O depósito ou a retirada de dinheiro em um banco, reservas em hotéis ou assentos de aviões em companhias aéreas, a compra de produtos em um supermercado ou consultas a catálogos informatizados de uma biblioteca, são bons exemplos da utilização dos SBD na vida das pessoas. Todas essas atividades, para serem eficientes, envolvem o acesso a um banco de dados. As interações acima são exemplos do que podemos chamar de *aplicações de bancos de dados tradicionais (convencionais)* [Elmasri and Navathe, 2002].

Ao longo dos últimos anos, os Sistemas de Bancos de Dados (SBD) também estão sendo utilizados em novas e interessantes aplicações. Entre essas aplicações se destacam as aplicações geográficas, de multimídia, biológicas, de apoio à decisão e processamento analítico on-line, em sistemas de tempo real, na internet e no ambiente de computação móvel.

Recentes avanços na tecnologia da comunicação sem fio (*wireless*) têm transformado os serviços de informações móveis em uma realidade. Há um número cada vez maior de novos sistemas móveis que necessitam de acessos a bancos de dados utilizando equipamentos do tipo *Personal Digital Assistant - (PDA)*, computadores portáteis ou telefones celulares [Phatak and Badrinath, 1999]. A tecnologia de computação móvel traz uma melhoria na distribuição e no fluxo de informações e também provê um aumento considerável nas funcionalidades das aplicações de bancos de dados [Özsu and Valduriez, 1999].

A computação móvel tem se transformado em uma realidade graças à convergência de duas tecnologias: o surgimento de computadores portáteis com maior poder de processamento e o desenvolvimento de redes de comunicação de dados mais velozes e confiáveis. Um número cada vez maior de computadores necessita consultar diversos bancos de dados através de canais de comunicação sem fio. Esses *clientes móveis*, dependendo do contexto, irão frequentemente desconectar-se de suas redes de comunicação de dados por longos períodos de tempo, desconexão esta causada, entre outros motivos, pela pouca capacidade das baterias de seus equipamentos. Além disso, irão se locomover entre pontos distintos de suas redes, se conectando a diferentes servidores de dados em diferentes momentos [Barbará, 1999].

O paradigma da computação móvel tem afetado conceitos e modelos tradicionais em várias áreas da ciência da computação. Por exemplo, na área das redes de comunicação de dados, as redes necessitam estar disponíveis em todos os lugares, pois devem garantir a conectividade aos seus usuários independente de sua localização física. Redes de comunicação que atendam a essa propriedade são chamadas de *ad hoc networks*. Já na área de engenharia de software o paradigma da computação móvel introduziu a noção de código móvel, o que significa a capacidade do código migrar, por exemplo, de um servidor de banco de dados para um cliente móvel, visando executar uma funcionalidade específica. Já na área de banco de dados é introduzido o conceito e a necessidade dos clientes móveis acessarem seus bancos de dados a partir de qualquer lugar (*any-where*) e a qualquer momento (*any-time*) [Brayner and Filho, 2002] e [Satyanarayanan, 2001].

A computação móvel pode ser considerada como uma variação da computação

distribuída [Özsu and Valduriez, 1999]. Assim sendo, questões relativas à gerência de dados distribuídos também podem ser aplicadas aos bancos de dados móveis, tais como distribuição e replicação de dados, modelos de transações, processamento de consultas, recuperação e tolerância a falhas, bem como seu projeto de bancos de dados, todas analisadas com o objetivo de atender os requisitos do ambiente da computação móvel [Frank, 1999], [Gore and Ghosh, 2000] e [Lam et al., 2000].

Entre os principais objetivos deste trabalho destacam-se a utilização dos bancos de dados no ambiente da computação móvel, ressaltando seus novos requisitos e funcionalidades, a apresentação de diversos modelos e arquiteturas que podem ser utilizados com os bancos de dados no ambiente de computação móvel, os principais problemas encontrados no ambiente de computação móvel e como os bancos de dados podem ajudar a resolver estes problemas.

Este trabalho está organizado da seguinte forma: nesta seção é apresentada uma introdução ao ambiente da computação móvel, destacando suas tecnologias, sua arquitetura, algumas aplicações, seus principais problemas e limitações, bem como uma breve revisão dos principais conceitos de um Sistema de Banco de Dados (SBD). Na segunda seção é apresentada uma classificação para os Sistemas de Gerência de Banco de Dados Móveis (SGBDs Móveis), com destaque para as extensões do modelo cliente/servidor e o modelo de agentes móveis. Na terceira seção a ênfase está na apresentação dos principais problemas da computação móvel que envolvem os bancos de dados, com destaque para o processamento de transações, os estados de operação que as unidades móveis podem assumir, que tipo de operações podem executar em cada um desses estados, como os bancos de dados são utilizados nestes estados, os problemas envolvendo a desconexão de suas redes, a fraca conectividade e como tratar as falhas para recuperação dos bancos de dados. A quarta seção apresenta a gerência de dados móveis, destacando a comunicação via *broadcast*, o problema da localização dos clientes móveis e alguns requisitos e funcionalidades necessários para a gerência da informação. Na quinta seção é apresentado o problema da localização dos clientes móveis, enfatizando as arquiteturas de bancos de dados usadas para gerenciar a localização desses clientes. Na sexta seção o destaque é para como os bancos de dados móveis estão disponíveis comercialmente pela indústria e o andamento de alguns seguimentos de pesquisa científica e, finalmente, na sétima seção é apresentada uma conclusão deste trabalho.

1.1. Tecnologia sem fio (*wireless*)

Conexões sem fio podem existir em diferentes partes de um ambiente das redes de comunicação de dados. Um importante papel das redes que suportam conexões sem fio é a sua capacidade de permitir a conexão de um grande número de endereços fixos e sem fio na sua infra-estrutura já existente. Existem muitas tecnologias de acessos sem fio para conexão de usuários móveis ao backbone de redes com grande largura de banda. Tais tecnologias incluem: celular, rádios públicas e privadas, redes locais sem fio, sistemas de paging e sistemas de satélites em órbita da terra [Pitoura and Samaras, 1998].

Celular - Estes sistemas podem ser geralmente caracterizados por proverem um amplo alcance de cobertura, suportarem duas formas (direção) de comunicação sem fio e terem uma sofisticada inteligência para gerenciar a mobilidade dos usuários. Sistemas

de celulares foram projetados para prover serviços de voz de grande alcance a pessoas que estejam em movimento e geralmente usam grande poder de transmissão. Por esta razão, sistemas de celulares podem somente prover serviços reduzidos para um conjunto de equipamentos portáteis que usam um pouco do baixo poder de transmissão e antenas menos eficientes que os veículos dos usuários.

A primeira geração dos sistemas celulares foi baseada na *tecnologia analógica*. Nesta tecnologia a voz é transmitida através de circuitos-switch com estreitas larguras de bandas nos canais, usando frequência modular (FM). O sistema controla para cada célula¹ sob controle de uma Estação de Suporte a Mobilidade (ESM), também conhecida como Estação de Base (EB), a sua localização. Uma ESM está localizada em uma rede fixa. Esta geração de sistemas sem fio se caracteriza por possuir uma baixa taxa de transmissão de dados entre a ESM e o usuário móvel.

A segunda geração de sistemas sem fio emprega *modulação digital* e possui uma capacidade avançada no processando de chamadas. Todos os sistemas celulares digitais provêm serviços de uma *Rede Digital de Serviços Integrados*, padrão ISDN (Integrated Services Digital Network), os quais tornam a transferência de dados muito mais rápida. Portanto, podem oferecer serviços de voz junto com uma variedade de serviços de dados, tais como paging, faxsimile e outros acessos a redes com melhores taxas de transferência de dados. Nesta geração, as estações móveis apresentam e executam mais funções de controles do que nos sistemas da primeira geração, incluindo o controle do processo de *handoff*², codificação e criptografia de dados. Exemplos dessa segunda geração de sistemas sem fio incluem o *Global System for Mobile (GSM)*, desenvolvido principalmente na Europa, o *Time Division Multiple Access (TDMA)* e o *Code Division Multiple Access (CDMA)*, padrão digital dos Estados Unidos da América (EUA), o *Personal Access Communication Systems (PACS)* e o *Digital European Cordless Telephone (DECT)*, padrão Europeu para os telefones sem fio e de escritórios.

Rádios Públicas - Sistemas de Rádios Públicas (SRP) possibilitam um excelente e eficiente canal para transmissão de dados de pequeno tamanho. Os sistemas existentes podem ser geralmente caracterizados como provedoras de alta mobilidade, ampla abrangência e baixa taxa de comunicação de dados digital. Esses sistemas têm atraído muita atenção e servem muitos clientes em suas aplicações verticais, ou seja, específicas de uma empresa.

Rádios Móveis Privadas - Sistemas de Rádios Móveis Privadas (SRMP) são sistemas de rádio tradicionais provendo sistemas de voz sem fio, a uma crescente demanda de serviços de comunicação de dados a organizações, tais como polícias estaduais, serviços de táxis, serviços de ambulância, empresas de transportes, etc. Algumas redes são construídas em forma de consórcios e servem a várias organizações. Tais sistemas são chamados de Sistemas de Rádio Móveis Compartilhados (SRMC). Os SRMP e os SRMC podem ser caracterizados como sistemas sem conexão de voz, uma vez que seus usuários estão, basicamente, competindo por pequeno intervalo de tempo na mesma frequência. Em geral, a tecnologia é proprietária e totalmente fornecida pelo vendedor dos equipamentos.

¹Área geográfica de abrangência de uma estação de suporte a mobilidade ou estação de base.

²Mudança da estação de base em função do movimento entre células por um host móvel.

Redes Locais Sem Fio (RLSF) - As RLSF são geralmente vistas como provedoras de alta velocidade de comunicação e com menor mobilidade dos seus clientes. RLSFs não objetivam suportar a verdadeira mobilidade, mas particularmente fornecer uma ponte entre as redes com fio convencionais e as redes sem fio. São projetadas para operarem em regiões bem definidas, como por exemplo no campus de uma universidade, em departamentos de uma empresa, etc. Na falta de um padrão, esses sistemas empregam uma variedade de tecnologias para suas conexões. Em geral, as RLSFs de alta velocidade se enquadram em duas categorias: as que são desenvolvidas para operarem usando frequência de rádio e as que são desenvolvidas para utilizar regiões infravermelho. Possibilitam altas taxas de transmissão de dados, a partir de centenas de Kbps a mais de 15 Mbps. Alguns exemplos dessas redes incluem as fabricadas pela Motorola, NCR, Proxim, WinData e InfraLan.

Redes de Paging - Paging, na sua forma mais simples de apresentação, nada mais possui do que a capacidade de avisar e alertar que alguém deseja se comunicar com o seu portador. Um sistema de rádio *paging* foi originalmente definido como não discursivo (*non-speech*), de mão única, um sistema seletivo pessoal com alertas, sem mensagens ou com mensagens definidas como alfanuméricas ou numéricas. Já os sistemas mais recentes possuem uma limitada capacidade de comunicação em mão dupla e enviam pequenas mensagens de voz. Um aspecto atrativo dos paging é o baixo consumo de recursos para receber mensagens.

Sistemas de Satélites - Sistemas sem fio com base em satélites representam a maior cobertura de longo alcance, cobrindo amplas áreas como oceanos, continentes e países. Podem ser caracterizadas por proverem uma comunicação em mão dupla, com baixas qualidades de comunicação de voz e/ou com capacidade muito limitada para troca de mensagem ou de dados.

1.2. Arquiteturas Sem Fio

Mobilidade tornou-se um novo paradigma em sistemas distribuídos. Como consequência da mobilidade, destaca-se a complexa e difícil tarefa de manter as bases de dados operacionais atualizadas. Em relação à interface de comunicação, existe o problema da qualidade da conexão sem fio, que varia abruptamente em função do local, as falhas de comunicação entre os terminais remotos e seus hosts, e a segurança da comunicação entre os clientes móveis e seus servidores fixos. Quanto à portabilidade dos dispositivos, destacam-se a pouca memória disponível nos equipamentos móveis, a lentidão de seus processadores e a sua capacidade em se manter ativo em função da energia fornecida por sua bateria [Endler and da Silva e Silva, 2000].

1.2.1. Arquitetura de um Sistema Móvel

A arquitetura genérica de uma plataforma móvel consiste em uma arquitetura distribuída [Özsu and Valduriez, 1999], na qual diversos computadores, geralmente conhecidos como Hosts Fixos (*FS - Fixed Hosts*) e Estações de Base (*BS - Base Stations*), são interligados através de uma rede com fio de alta velocidade. Hosts fixos são computadores de finalidade genérica que não são equipados para gerenciar unidades móveis, mas podem ser configurados de forma a fazê-lo. Estações de Base são equipadas com interfaces para

as redes sem fio e podem comunicar-se com as unidades móveis para suportar o acesso a dados [Elmasri and Navathe, 2002].

Unidades Móveis (*Mobile Units*) são computadores portáteis movidos à bateria, que se movimentam livremente em um domínio geográfico de mobilidade, uma área que é restringida pela limitada amplitude de banda dos canais de comunicação sem fios. Para gerenciar a mobilidade dessas unidades, o domínio de mobilidade geográfica é dividido em domínios menores chamados de *células*. A computação móvel requer que o movimento de unidades seja irrestrito dentro do domínio de mobilidade geográfica (movimento intercelular), ao mesmo tempo em que possuir contigüidade de acesso durante o movimento, garante que o movimento de uma unidade móvel através dos limites das células não terá nenhum efeito sobre o processo de recuperação de dados [Elmasri and Navathe, 2002].

Unidades Móveis e Estações de Base se comunicam através de canais sem fio que possuem largura de banda significativamente menores do que aquelas de uma rede com fio. Um canal de conexão do tipo *downlink* é utilizado para enviar dados da Estações de Base para as Unidades Móveis, e um canal de conexão do tipo *uplink* é utilizado para enviar dados das Unidades Móveis para as Estações de Base [Elmasri and Navathe, 2002]. A Figura 1, extraída de [Dunham and Helal, 1995] apresenta uma arquitetura genérica para um sistema de computação móvel.

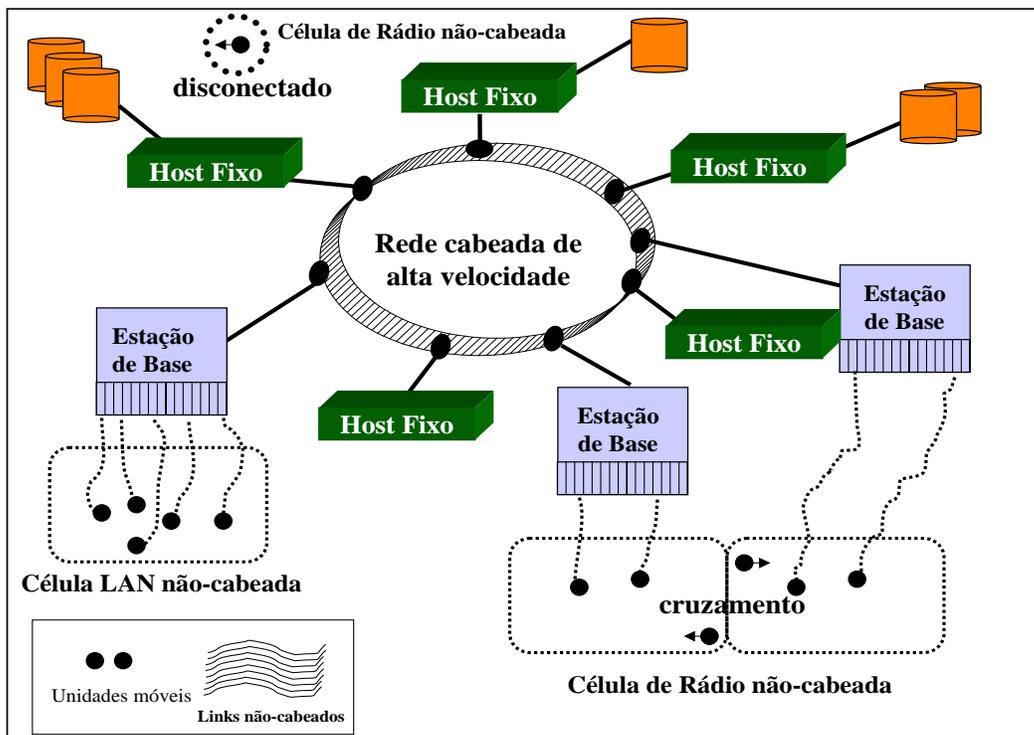


Figura 1: Arquitetura de um Sistema de Computação Móvel

Segundo [Enderl and da Silva e Silva, 2000], entre os principais requisitos impostos pelas características da computação móvel aos sistemas de softwares destacam-se:

1. A capacidade de localizar/endereçar elementos móveis;

2. A capacidade de perceber mudanças no ambiente de execução;
3. A capacidade de se autoconfigurar;
4. A capacidade de migrar funcionalidade;
5. O uso eficiente dos recursos no elemento móvel;
6. O uso eficiente dos recursos do canal de comunicação sem fio;
7. O alto grau de tolerância à falhas; e
8. Os mecanismos para autenticação e encriptografia de dados.

1.2.2. Gerência de Dados em um Ambiente Móvel

A computação móvel pode ser considerada como uma variação da computação distribuída. Os *bancos de dados móveis* podem ser distribuídos sob dois aspectos [Özsu and Valduriez, 1999]:

1. Todo o banco de dados é distribuído, principalmente, entre os componentes sob a rede com fio, possivelmente com replicação total ou parcial. Assim, uma estação de base gerencia seu próprio banco de dados com as funcionalidades inerentes aos Sistemas de Gerência de Banco de Dados (SGBDs), com funcionalidades adicionais para *localizar unidades móveis* e características adicionais de *gerência de consultas e transações*, para atender aos requisitos de ambientes móveis;
2. O banco de dados é distribuído entre os componentes sob a rede com fio e com os componentes sob a rede sem fio. A responsabilidade sobre a gerência de dados é compartilhada entre estações de base e unidades móveis.

Segundo [Elmasri and Navathe, 2002], [Frank, 1999], [Lam et al., 2000] e [Gore and Ghosh, 2000], as questões relativas à gerência de dados distribuídos podem também ser aplicadas aos bancos de dados móveis com os seguintes aspectos sendo analisados de forma especial:

1. A distribuição e replicação de dados;
2. Os modelo de transação;
3. O processamento de consultas;
4. A recuperação e tolerância à falhas;
5. O projeto de bancos de dados móveis.

1.3. Aplicações

Com a arquitetura e a infra-estrutura disponíveis surge uma nova classe de aplicações, a qual combina a computação pessoal com clientes e o ambiente da computação móvel. Para que essas novas tecnologias de computação móvel e bancos de dados móveis possam trabalhar em conjunto, é necessário que as mesmas dêem suporte a aplicações que sejam necessárias e úteis a uma comunidade de usuários. A tecnologia sem fio deve ir além de servir a um pequeno número de aplicações verticais (de âmbito específico de empresas) e alcançar um grande número de usuários. Entre as principais características que podem levar essas tecnologias a se popularizarem destacam-se:

1. A possibilidade de acessar a grandes bancos de dados de qualquer lugar e a qualquer momento;

2. A versatilidade de formas de comunicação e a cooperação que poder existir entre seus usuários;
3. A possibilidade de notificação a uma comunidade de usuários a respeito de fatos ou eventos críticos e úteis para seus usuários móveis;
4. A possibilidade de integração de pontos distantes de empresas.

Entre os muitos exemplos de aplicações para o ambiente de computação móvel e bancos de dados móveis destacam-se diversas aplicações que podem ser classificá-las em verticais e horizontais (no âmbito geral de comunidades) [Imielinski and Badrinath, 1992], [Badrinath et al., 1993] e [Pitoura and Bhargava, 1994a]:

- Vendas de produtos por vendedores com acesso a estoques em tempo real;
- Controle de transportes de cargas e localização geográficas de caminhões através de rastreamento;
- Acompanhamento e orientação de procedimentos médicos em ambulâncias;
- Envio e recebimento de arquivos e e-mails em notebooks através de modem de celular;
- Acesso à internet em notebooks e PADs através de modem de celular;
- Computação cooperativa entre usuários móveis, como por exemplo alertas de acidentes em rodovias, catástrofes naturais, situações de perigo, etc;
- Recebimento e armazenamento de fotos de satélites;

1.4. Problemas e Limitações

Diversas novas características e limitações inerentes aos sistemas de computação móveis precisam ser reavaliadas, principalmente no contexto dos SBDs. Segundo [Satyanarayanan, 1996] estas limitações podem ser explicadas pelas seguintes restrições:

- Elementos móveis são relativamente pobres em recursos em relação aos elementos fixos, estáticos;
- A mobilidade é em sua natureza perigosa;
- A conectividade no ambiente móvel é altamente variável em seu desempenho e na confiabilidade da conexão;
- Os elementos móveis contam com uma fonte de energia finita para seus equipamentos.

As características e limitações dos sistemas de computação móveis, descritas em [Pitoura and Samaras, 1998] e [Endler and da Silva e Silva, 2000], podem ser classificadas de acordo com as seguintes propriedades:

1. **Mobilidade dos hosts** - A localização dos elementos móveis e conseqüentemente seus pontos de conexão a rede fixa se alteram como o seu movimento. As principais conseqüências da mobilidade são:
 - (a) A configuração dos sistemas que incluem elementos móveis não são estáticas.
 - i. Os algoritmos de processamento distribuídos tradicionais precisam ser reprojatados devido à ausência de uma topologia fixa da rede com os elementos móveis;

- ii. A distribuição de carga em uma rede com elementos móveis pode mudar de maneira muito rápida, em função da possibilidade de mudança de localização dinâmica desses elementos.
- (b) Gerência de localização.
- i. O custo para localizar um elemento móvel contribui de forma significativa para o custo de cada comunicação;
 - ii. Estruturas de dados, algoritmos e planos de consultas eficientes devem ser planejados e desenvolvidos para representar, gerenciar e consultar a localização dos elementos móveis, uma vez que os dados de localização podem se alterar de forma muito rápida.
- (c) Heterogeneidade.
- i. A conectividade de um elemento móvel pode variar muito em relação ao seu desempenho, principalmente pela largura de banda do canal de comunicação e pela latência³ da comunicação, bem como pela confiabilidade dessa comunicação;
 - ii. O número de serviços disponíveis para um elemento móvel pode variar de uma localização para outra, em função das características da célula e da estação de base na qual ele se conecta;
 - iii. Os recursos disponíveis nos elementos móveis variam, como por exemplo a sua capacidade de memória, o tamanho de sua tela, o tempo de duração de sua bateria, etc.
2. **Interface de comunicação sem fio** - O fato da comunicação entre um elemento móvel e uma estação de base ser através de uma rede de comunicação sem fio, tem como principais conseqüências:
- (a) Conectividade fraca e intermitente⁴.
- i. As redes sem fio são monetariamente mais caras;
 - ii. Oferecem menores larguras de banda;
 - iii. Possuem uma latência maior do que as redes com fio;
 - iv. São menos confiáveis que as redes com fio.
- (b) Conectividade variável.
- i. A qualidade da conexão pode variar abruptamente em função de possíveis interferências na comunicação, da distância do elemento móvel com a estação de base na qual está conectado ou devido ao compartilhamento da estação de base por vários elementos móveis;
 - ii. A tecnologia sem fio se modifica em função do grau da largura da banda de comunicação e da segurança que a rede fornece;
- (c) Facilidade para transmissão de dados (broadcast).
- i. Existe um canal com grande largura de banda para transmissão de dados da estação de base para todos os clientes móveis em sua célula;
 - ii. Possibilidade para disseminação de informações a grupos de clientes móveis específicos;
 - iii. Possibilidade de prestação de serviços específicos para certos grupos de clientes móveis
- (d) Tarifas

³Período de inatividade entre um estímulo e a resposta por ele provocada.

⁴Apresenta interrupções ou suspensões; não contínua.

- i. Em algumas redes o acesso à rede é pago pelo tempo de conexão, como por exemplo nas comunicações via telefones celulares;
 - ii. Em outras redes o acesso à rede é pago por mensagens, como por exemplo nas comunicações via pacotes de rádio.
3. **Portabilidade dos Elementos Móveis** - Por serem portáteis, os elementos móveis precisam ser pequenos e leves, tornando-os inferiores em suas capacidades de processamento aos computadores convencionais. Assim sendo:
 - (a) Elementos móveis são pobres em recursos quando comparados com os elementos estáticos;
 - i. São equipados com pouca memória do tipo RAM (Random Access Memory)⁵;
 - ii. Seus processadores são mais lentos;
 - iii. Possuem pouca memória não volátil (não possuem disco rígido);
 - iv. A interface do usuário é mais limitada;
 - v. O monitor é menor ou tem menor capacidade em seu buffer;
 - vi. Os dispositivos de entrada de dados também são menores e limitados (por exemplo, teclados de aparelhos celulares).
 - (b) Elementos móveis contam com pouca capacidade em suas baterias;
 - i. Dependem da energia fornecida por baterias;
 - ii. Normalmente as baterias possuem capacidades limitadas;
 - iii. Em alguns lugares, dificuldade para recarga da bateria.
 - (c) Elementos móveis são poucos robustos
 - i. Fáceis de serem danificados;
 - ii. Facilmente perdidos ou roubados.

1.5. Sistemas de Bancos de Dados (SBD) - SGBDs, BDs e Aplicações

Bancos de Dados e suas tecnologias vêm tendo um grande impacto no crescimento do uso de computadores. Podemos afirmar que os bancos de dados desempenham um papel extremamente importante em todas as áreas da ciência onde a computação é utilizada. A expressão *banco de dados* tem sido tão amplamente utilizada que precisamos iniciar com a definição do que seja um banco de dados. Segundo [Elmasri and Navathe, 2002], um banco de dados *é uma coleção de dados relacionados*. Dados são definidos como fatos conhecidos que podem ser registrados e que possuem significado implícito. Por exemplo, *o código, nome, valor e a quantidade do produto em estoque* de um supermercado, é uma coleção de dados relacionados com um significado implícito e, portanto, um banco de dados. Um banco de dados possui as seguintes propriedades implícitas:

- Representa algum aspecto do mundo real. Alterações neste mundo real são refletidas no banco de dados;
- É uma coleção lógica e coerente de dados com algum significado inerente. Uma organização aleatória de dados não pode ser referenciada corretamente como um banco de dados;
- É projetado, construído e povoado com dados que possuem um objetivo específico. Ele possui um grupo provável de usuários e algumas aplicações preconcebidas, nas quais esses usuários estão interessados;

⁵Memória de Acesso Aleatório.

- Deve possuir alguma fonte da qual os dados são derivados, algum grau de interação com eventos do mundo real e um público que está ativamente interessado no conteúdo do banco de dados.

Um **Sistema de Gerência de Banco de Dados (SGBD)** é uma coleção de programas que possibilita que os usuários criem e mantenham um banco de dados. O SGBD é portanto um sistema de software de finalidade genérica que facilita o processo de definição, construção e manipulação de bancos de dados para várias aplicações. *Definir* um banco de dados envolve especificar os tipos de dados, as estruturas e as restrições para os dados que serão armazenados no banco de dados. *Construir* o banco de dados é o processo de armazenar os referidos dados em algum meio de armazenamento que seja controlado pelo SGBD. *Manipular* o banco de dados inclui funções como fazer consultas ao banco de dados para recuperar dados específicos, atualizar o banco de dados para refletir as alterações do mundo real e gerar relatórios a partir dos dados [Elmasri and Navathe, 2002].

Um **Sistema de Banco de Dados (SBD)** é formado pelos bancos de dados, pelo SGBD e pelas aplicações necessárias para criação e manutenção do banco de dados, bem como pelas aplicações para atualização e consulta aos dados do banco de dados. A Figura 2, extraída de [Elmasri and Navathe, 2002], apresenta um ambiente simplificado de um SBD.

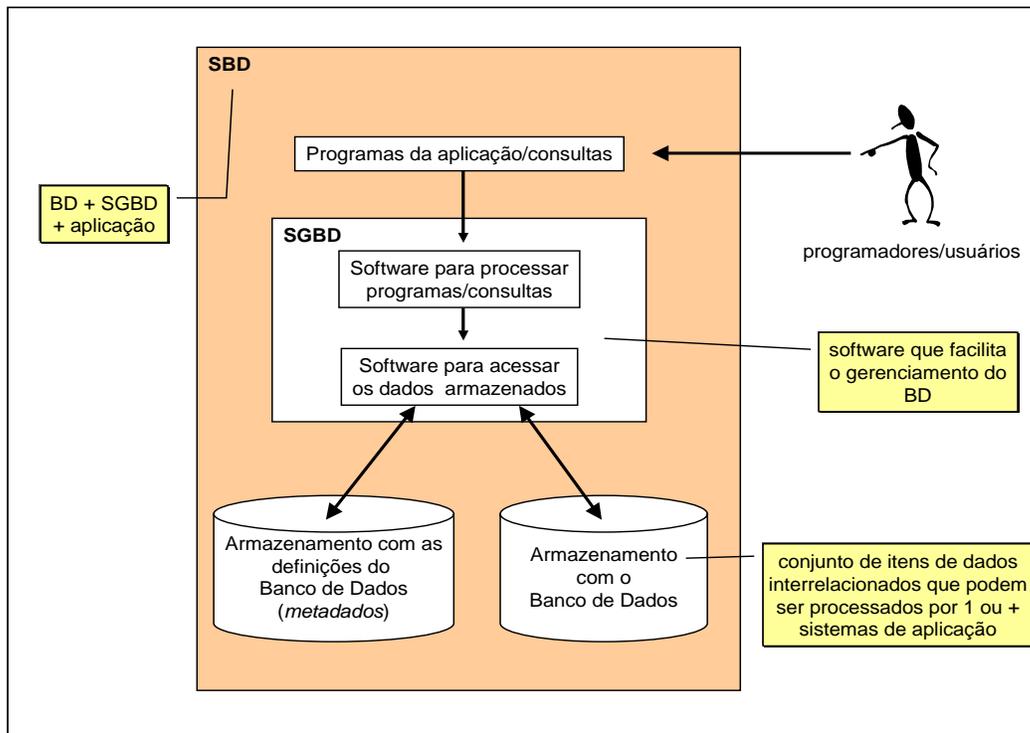


Figura 2: Um ambiente simplificado de um SBD

2. Classificação dos SGBDs Móveis e Arquiteturas de software para computação Móvel

O ambiente de computação móvel é restritivo em vários aspectos. Os elementos móveis possuem poucos recursos e não são muito confiáveis. Sua conectividade com a rede é frequentemente obtida através de links sem fio com pequena largura de banda. Além disso, a conectividade é frequentemente perdida em vários momentos. Essas dificuldades causadas por estas restrições são combinações da mobilidade que induz a variabilidade na disponibilidade tanto dos recursos computacionais como de comunicação. Essas restrições causam um grande impacto no projeto e na estrutura das aplicações para computação móvel e motivam a pesquisa e desenvolvimento de novos modelos computacionais. Esses modelos devem prover eficiência de acesso tanto para as novas aplicações como as já existentes, o que é o requisito básico para uma ampla aceitação da computação móvel.

2.1. Classificação dos Sistemas de Gerência de Banco de Dados Móveis

Podemos ver o ambiente de computação dos SGBDs Móveis como uma extensão dos sistemas distribuídos. [Özsu and Valduriez, 1999] oferece uma excelente classificação para os SGBDs Distribuídos baseado nas características do sistema quanto a *autonomia*, *distribuição* e *heterogeneidade*. A Figura 3 apresenta uma extensão para essa classificação segundo [Dunham and Helal, 1995] para os SGBDs Móveis, os quais acrescentam um ponto no eixo de *distribuição*, caracterizando os sistemas de computação móveis. A lógica para essa extensão está no fato que os sistemas de computação móveis devem ter uma rede fixa (veja figura 1) que é um sistema distribuído.

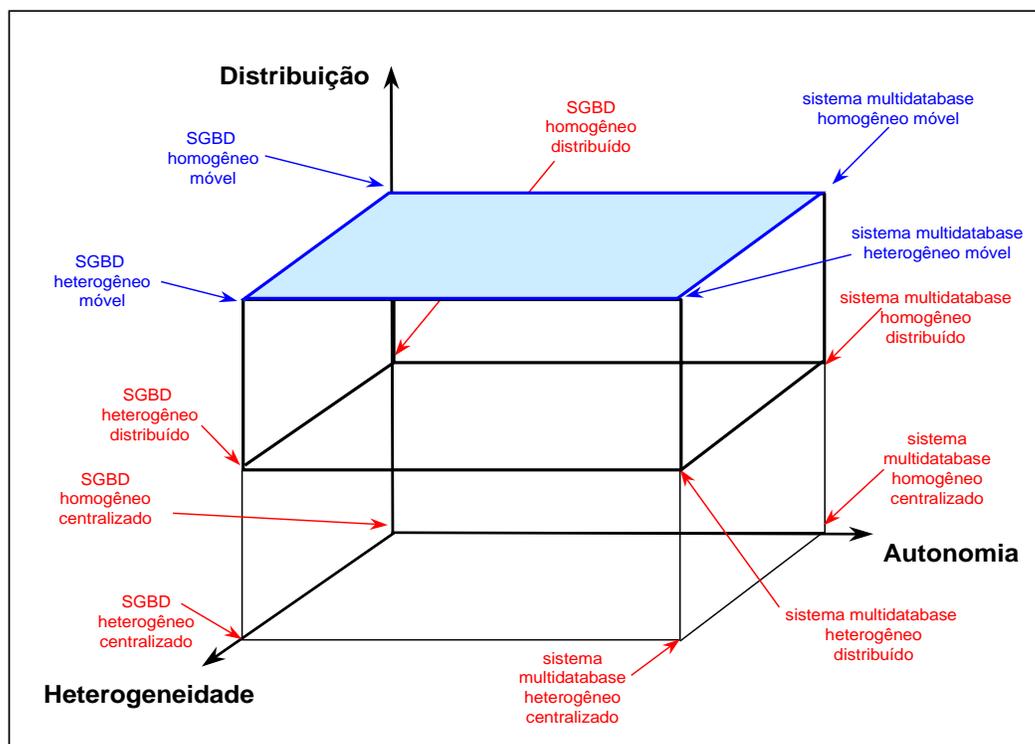


Figura 3: Classificação dos SGBDs Móveis

Considerando os sistemas móveis como um caso especial dos sistemas distribuídos, percebe-se que enquanto muitos problemas são iguais, existem outros problemas bem diferentes. Os sistemas distribuídos têm como principal objeto a *transparência de distribuição*, já os sistemas móveis buscam o *conhecimento da localização*, ou seja, saber onde um cliente móvel está localizado geograficamente (espacialmente) para poder manter uma comunicação constante, sem interrupções, com o mesmo.

Entre as principais características dos SGBDs Móveis podemos destacar:

- Vários bancos de dados interligados por uma rede de comunicação sem fio;
- Acesso de um computador móvel (*host móvel*) a um banco de dados residindo em um em um computador fixo (*host fixo*) ou em um computador móvel;
- O computador móvel pode desempenhar o papel de cliente ou servidor de banco de dados;
- Os bancos de dados são autônomos, distribuídos e provavelmente heterogêneos.

Quanto à gerência e administração dos dados, diversos fatores podem influenciar o funcionamento e a performance dos bancos de dados no ambiente de computação móvel. Entre esses fatores, destacam-se:

- *Velocidade dos links sem fio* - pode ocasionar demora nas consultas;
- *Escalabilidade* - crescimento dos bancos de dados tem impacto nas consultas e limita a quantidade de dados nos bancos de dados residentes em clientes móveis;
- *Mobilidade* - pode causar desconectividades e cancelamento de transações;
- *Localização dos hosts móveis* - implica na necessidade de administrar a localização dos hosts móveis para interação durante as consultas;
- *Limite do poder das baterias* - exigem trabalho em modo desconectado da rede e podem causar desconexões e cancelamento de transações;
- *Desconectividade* (voluntária ou involuntária) - causam o cancelamento de transações, aumentando a necessidade de novos modelos de recuperação (*recovery*) do banco de dados;
- *Replicação/Caching* - limitadas pelo pouco poder de memória dos hosts móveis;
- *Handoff* - aumenta a necessidade de controle e administração da localização dos hosts móveis.

Para facilitar o entendimento de algumas características do ambiente de computação móvel, é importante entender algumas diferenças entre as soluções tradicionais aplicadas aos bancos de dados convencionais para solução de problemas em bancos de dados distribuídos, e esses mesmos problemas no ambiente de computação móvel. A tabela 1, extraída de [Dunham and Helal, 1995] apresenta essas principais diferenças.

No ambiente distribuído o SGBD necessita se recuperar de eventuais falhas em sites, nos meios de comunicação, de transações e de comunicação. Estes mesmos problemas persistem na computação móvel, entretanto, a frequência pode ser muito maior. Estes problemas complicam a recuperação, devido ao fator de mobilidade nas transações. Transações distribuídas são executadas concorrentemente em múltiplos

Principais diferenças para os BDs móveis	
<i>Implicações na Administração de Dados</i>	<i>Diferenças com a Computação Móvel</i>
Aplicações	<ul style="list-style-type: none"> - Pode ser dependente de localização do host móvel - Necessita adaptar-se às mudanças no contexto do sistema
Transações	<ul style="list-style-type: none"> - Novos modelos para captação da mobilidade
Recuperação	<ul style="list-style-type: none"> - Divisão (particionamento) freqüente da rede - Desligamento voluntário do host móvel não é uma falha do sistema - A mobilidade pode causar inúmeros acessos (logins) ao sistema - Técnicas para recuperação de desconexões durante o handoff
Replicação	<ul style="list-style-type: none"> - Diferentes restrições de consistência - Novas técnicas para os hosts móveis atualizarem suas memórias cache durante as freqüentes desconexões
Processamento de consulta	<ul style="list-style-type: none"> - Dependente da localização do host móvel - Diferentes fatores de custos - Resposta das consultas retornam para diferentes localizações - Necessidade de adaptações das técnicas
Resolução de nomes (endereços)	<ul style="list-style-type: none"> - Estratégia de nome global devido a desconectividade e mobilidade

Tabela 1: Diferenças entre a gerência de dados distribuídos e a gerência de dados móveis

conjuntos de dados e processadores. As execuções das transações distribuídas são totalmente coordenadas pelo sistema, incluindo o controle de concorrência, a gerência de replicação e a atomicidade das transações. Transações móveis, por outro lado, são executadas seqüencialmente através de múltiplas estações de base, em vários conjuntos de dados, dependendo do movimento da unidade móvel. Assim, a execução da transação móvel na estação de base pode ser considerada uma transação distribuída. A execução da transação, é então, não totalmente coordenada pelo SGBD. O movimento da unidade móvel, por uma grande extensão espacial, controla a execução. Somente este fato exige um novo modelo de transações para o SGBD Móvel.

2.2. Agentes de Software

Um *agente* é um sistema de computação situado em algum ambiente computacional, sistema este que é capaz de executar ações autônomas neste ambiente, visando atingir os objetivos para o qual foi projetado e desenvolvido. Um sistema de agentes deve poder agir sem a intervenção direta dos seres humanos ou de outros agentes, como também deve possuir todo o controle de suas próprias ações e estado interno. Um *agente inteligente* é um sistema computacional que é capaz de executar ações autônomas *flexíveis* a fim de atingir os objetivos do projeto para o qual foi projetado e

desenvolvido. Por flexível, entendemos que o sistema deve ser *reativo, proativo e social* [Jennings and Wooldridge, 1998].

A principal característica de um agente é a sua habilidade de agir de forma autônoma. Isto implica que um agente recebe estímulos de seu ambiente, pode executar um conjunto de ações que alteram este mesmo ambiente, decidindo pelas ações que devem executadas baseadas em seus próprios objetivos [Kearney, 1998]. Os agentes de software podem ser classificados por sua (1) *mobilidade*; (2) *capacidade de reação*; (3) *através dos diversos atributos que idealmente devem exibir*; (4) *por seus papéis*; e (5) *pelos agentes híbridos que combinam duas ou mais filosofias do agente em um único agente* [Nwana and Ndumu, 1998].

2.3. O modelo Cliente/Servidor e suas extensões para Computação Móvel

O planejamento de modelos apropriados para aplicações estruturadas que envolvem elementos sem fio é uma questão fundamental no desenvolvimento do software para a computação móvel. A seguir é apresentado um breve resumo sobre o modelo cliente/servidor, bem como algumas extensões desse modelo de software para a computação móvel.

2.3.1. O modelo Cliente/Servidor

A modelo (arquitetura) *cliente-servidor* supõe uma estrutura (*framework*) básica que consiste de muitos microcomputadores (PCs) e estações de trabalho, bem como um número menor de equipamentos de grande porte, conectados através de sistemas de redes locais e outros tipos de sistemas de rede de computadores. Um **cliente** nesta estrutura é geralmente um equipamento de um usuário que possui capacidades de interface e processamento local. Quando um **cliente** solicita acesso a funcionalidades adicionais, como por exemplo acesso a um banco de dados e este não reside neste equipamento, ele se conecta a um **servidor** que fornece a funcionalidade necessária. Um **servidor** é um equipamento que pode fornecer serviços para as estações clientes, como impressão, armazenamento de dados ou acesso ao banco de dados [Elmasri and Navathe, 2000].

A idéia da arquitetura *cliente-servidor* é dividir as funcionalidades que precisam ser fornecidas aos usuários do SGBD em duas camadas, a camada *servidor* e a camada *cliente*, ambas com funcionalidades distintas. Essa arquitetura facilita o gerenciamento da complexidade dos SGBDs atuais [Özsu and Valduriez, 1999]. A arquitetura cliente-servidor está incorporada aos Sistemas de Gerência de Banco de Dados (SGBD). Em SGBDs relacionais, através de sua uma linguagem padrão chamada *Structured Query Language (SQL)*, criou-se um ponto de divisão lógica entre o cliente e o servidor do banco de dados. Nos cliente estão os programas de aplicação e a interface do usuário, enquanto que as funcionalidades de consultas e transações permaneceu no lado do servidor. Neste caso podemos dizer que o servidor é um servidor de consulta ou servidor de transação, pois fornece essas duas funcionalidades. Os programas da interface do usuário e da aplicação podem ser executados no lado do cliente. Quando é necessário acesso ao banco de dados, o programa estabelece uma conexão com o SGBD, que está no lado do servidor, e uma vez que a conexão é estabelecida, os programas no cliente podem se comunicar com o SGBD. Um exemplo de como efetivar essa conexão é através do

padrão chamado Open Database Connectivity (ODBC), o qual fornece uma Application Programming Interface (API), que permite que programas no lado do cliente chamem o SGBD, contanto que os equipamentos do cliente e do servidor tenham instalado os software necessários. Os resultados das consultas são enviados de volta para os programas no cliente, que podem processar ou exibir os resultados conforme necessário. Um outro exemplo de conexão é a utilização de programação JAVA, chamado JDBC, permitindo que programas de clientes escritos na linguagem JAVA acessem o SGBD através de uma interface padrão [Elmasri and Navathe, 2000].

Conforme mostra a Figura 4, extraída de [Özsu and Valduriez, 1999], a camada servidor faz a maior parte do trabalho de gerenciamento de dados, enquanto que a camada cliente cuida da interface do usuário e da aplicação, além de administrar uma memória local para solicitação e armazenamento do resultado das consultas.

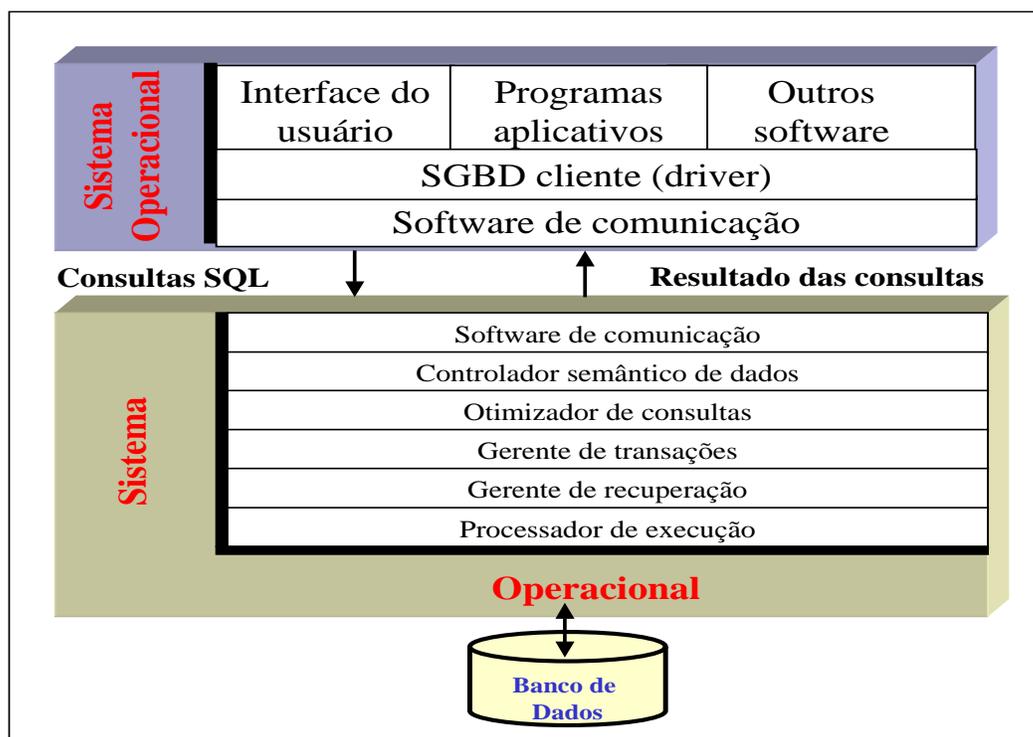


Figura 4: Componentes da modelo/arquitetura Cliente/Servidor para SGBDs

Diferentes tipos de arquiteturas cliente/servidor podem ser implementadas. A arquitetura mais simples é aquela em que vários clientes acessam um único servidor, denominada *múltiplos clientes-servidor único*. A arquitetura mais complexa é aquela em que o ambiente possui *múltiplos clientes-múltiplos servidores* [Özsu and Valduriez, 1999] e [Melo et al., 1997].

2.3.2. O modelo Cliente/Servidor para Ambiente de Computação Móvel

Na computação móvel, o *host móvel* atua como um *cliente* requisitando serviços dos servidores localizados na rede fixa. Nestes casos, os dados e as funcionalidades estão distribuídos através de vários *servidores* em diferentes *hosts fixos* que podem

comunicar-se entre si para atender as solicitações dos clientes. Em muitos casos, o servidor é replicado em diferentes sites na rede fixa para aumentar a disponibilidade nos casos de falhas nos sites ou na rede de comunicação. A Figura 5, extraída de [Pitoura and Chrysanthis, 2000], apresenta um ambiente de computação móvel em uma arquitetura cliente servidor tradicional.

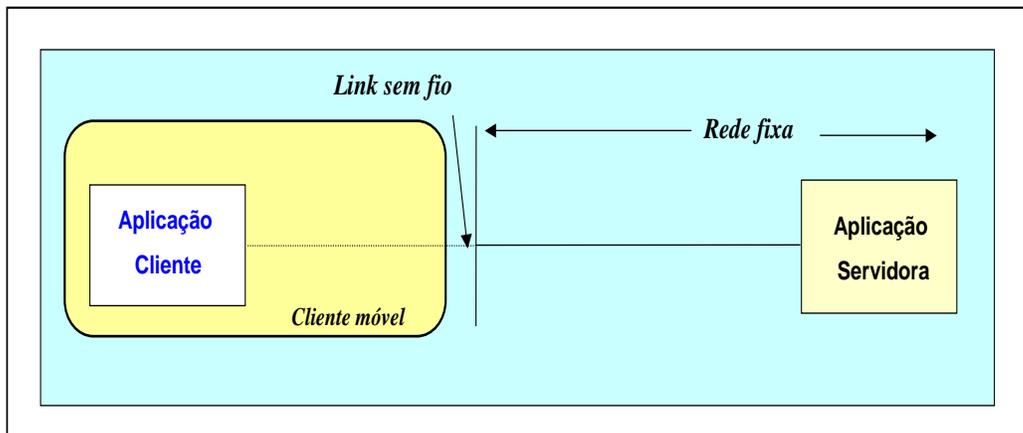


Figura 5: Ambiente de computação móvel em uma arquitetura Cliente/Servidor convencional

A divisão das funcionalidades entre os *clientes móveis* e os *servidores fixos* que estão sob as redes fixas não é muito clara. Em muitos casos o papel de cada um fica confuso, como no caso durante as desconexões em que o cliente móvel necessita emular as funcionalidades de um servidor para continuar operando.

Um importante componente da arquitetura cliente/servidor é o tipo de mecanismo de comunicação que é usado para a troca de informações entre o cliente e o servidor. Uma possibilidade é a troca de mensagens direta entre o cliente e o servidor. Essa abordagem não é adequada para as redes lentas e pouco confiáveis como é o caso da computação sem fio e novos mecanismos de troca de informações estão em estudos.

Extensões do tradicional modelo cliente/servidor, como se enfileirar mensagens RDC (Remote Procedure Call), são necessárias para dar apoio a operações desconectadas e a fraca conectividade. Otimizações adicionais como compressão e filtros de dados também são importantes. Por estes motivos, o tradicional modelo cliente/servidor deve ser estendido para prover sérios componentes para implementação de otimizações adequadas e possibilitar o mínimo possível de mudanças nos clientes e nos servidores [Pitoura and Samaras, 1998].

2.3.3. O Modelo Cliente/Agente-Servidor/Servidor

Este modelo em três camadas (*three-tier*) realiza a comunicação entre o cliente móvel e o servidor, através da troca de mensagens do cliente com o agente e deste com o servidor. Genericamente, podemos dizer que o agente assume o papel de substituto (*surrogate*) do cliente móvel na rede fixa. Esta arquitetura alivia um pouco o impacto da limitação da largura de banda e da pouca segurança do link sem fio, através da presença do cliente

móvel na rede fixa via o agente. Os agentes, tipicamente, dividem a interação entre os clientes móveis e os servidores fixos em duas partes, sendo uma entre o cliente e o agente e a outra entre o agente e o servidor fixo. Assim, diferentes protocolos⁶ podem ser usados para interação em cada parte e essas partes podem executar suas funcionalidades independentemente. A Figura 6, extraída de [Pitoura and Chrysanthis, 2000], apresenta essa arquitetura.

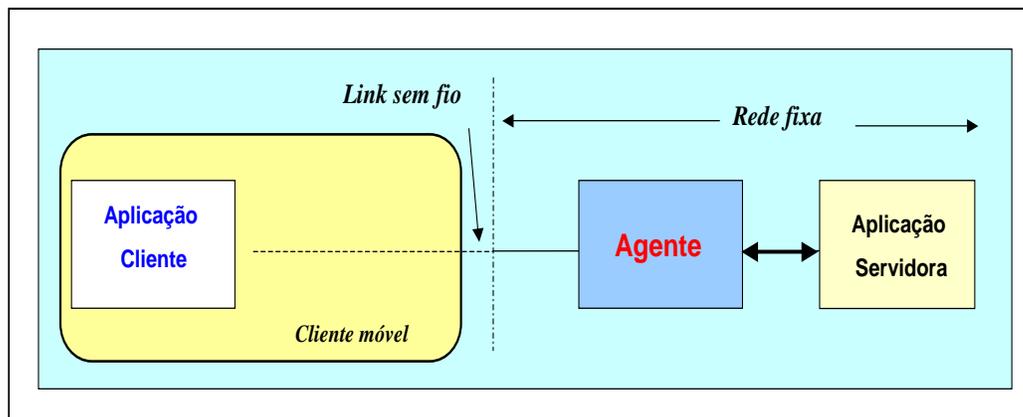


Figura 6: Modelo em três camadas Cliente/Agente-Servidor/Servidor

Este modelo é mais apropriado para clientes móveis com limitado poder de recursos e poder computacional. Assim, diversas responsabilidades dos clientes móveis migram para os agentes. Por exemplo, uma consulta com grande quantidade de dados como resposta pode ser gerenciada pelo agente e somente ao seu final ter seus resultados transmitido ao cliente móvel. A localização do agente na rede fixa possibilita o acesso à rede com grande largura de banda e grande poder computacional, podendo usar esses recursos em favor dos clientes móveis. Por exemplo, a compressão dos dados pode ser executada no agente, antes da transmissão dos dados aos clientes móveis.

Para lidar com a *desconexão* os clientes móveis podem submeter suas solicitações aos agentes e esperar a devolução dos resultados quando a conexão for restabelecida. Durante a desconexão, várias solicitações dos clientes móveis podem ser processadas e enfileiradas para serem transmitidas aos clientes móveis logo após a nova conexão. Esta estratégia também pode ser usada pelos clientes móveis para preservar o tempo útil de suas baterias. Os clientes móveis submetem suas solicitações, entram em modo de *cochilo (doze)* e aguardam a espera das respostas dos agentes. A fraca conectividade também pode ser efetivamente manuseada pelos agentes de várias formas, empregando inúmeras técnicas para minimizar o tamanho dos dados a serem transmitidos aos clientes móveis, dependendo do tipo de dados e das aplicações. O agente também pode manipular os dados antes da sua transmissão, alterando a sua ordem de transmissão em função da prioridade dos mesmos para os clientes móveis.

Os agentes podem ser usados de várias formas e desempenhando diversos papéis nesta arquitetura. A exata posição dos agentes na rede fixa depende do papel que o mesmo

⁶Conjunto de regras, padrões e especificações técnicas que regulam a transmissão de dados entre computadores por meio de programas específicos

venha a desempenhar. Colocando-o na extremidade da rede fixa, como por exemplo nas *estações de bases*, trás algumas vantagens quando o agente atua como substituto (*surrogate*) do cliente móvel em sua área de cobertura. Assim, fica mais fácil reunir informações das características do link sem fio; utilizar um protocolo especial entre o cliente móvel e o agente; personalizar informações sobre o cliente móvel que podem ser disponibilizadas localmente.

O modelo *Cliente/Agente-Servidor/Servidor* oferece um grande número de vantagens, porém, falha no suporte da operação dos clientes móveis durante o período de desconexão. Quando acontece uma desconexão, o cliente móvel não continua a operar ininterruptamente. Finalmente, o agente somente consegue otimizar a transmissão de dados do agente para o cliente móvel, mas não no caminho inverso [Pitoura and Samaras, 1998].

2.3.4. O Modelo Cliente/Agente-Cliente/Servidor

Este modelo apresenta uma extensão do modelo cliente/servidor, também em três camadas, porém, com a inclusão do *agente de software* junto ao host móvel, ou seja, no cliente móvel. A Figura 7, extraída de [Pitoura and Chrysanthis, 2000], apresenta essa extensão. Genericamente, podemos dizer que o agente assume o papel de ampliar as funcionalidades nos clientes móveis, geralmente pobres em recursos computacionais. Entre as muitas atividades específicas de uma aplicação que os agentes podem desempenhar, podemos destacar, de uma forma mais genérica, as seguintes atividades:

- Administrar a memória *caching* no cliente móvel;
- Disponibilizar memória progressivamente para o cliente móvel durante o pouco tráfego da rede (*prefetching*⁷);
- Copiar parte do banco de dados para a memória do cliente móvel (*hoarding*⁸), caracterizando um caso especial de replicação;
- Otimizar a comunicação entre o cliente móvel e sua estação de base.

2.3.5. O Modelo Cliente/Agente-Agente/Servidor

Para sanar as deficiências dos dois modelos anteriores, nos quais os agentes de software residem e executam em apenas um dos lados da arquitetura, ou seja, no cliente móvel ou no servidor da rede fixa, [Samaras and Pitsillides, 1997] propuseram este modelo, também chamado de *Cliente/Intercept/Server* no qual o agente situado no cliente móvel executará em conjunto com o agente situado no servidor. O agente situado no cliente móvel detecta (intercepta) as solicitações do cliente e juntamente com o agente situado no servidor executam otimizações para redução da quantidade de dados transmitidos na rede sem fio, melhoram a segurança na transferência dos dados e sustentam a não interrupção da computação móvel, entre outras possíveis atividades. A Figura 8, extraída de [Pitoura and Chrysanthis, 2000], apresenta este modelo.

⁷*Prefetching* é um processo progressivo de transferir dados para a memória cache do host móvel o mais

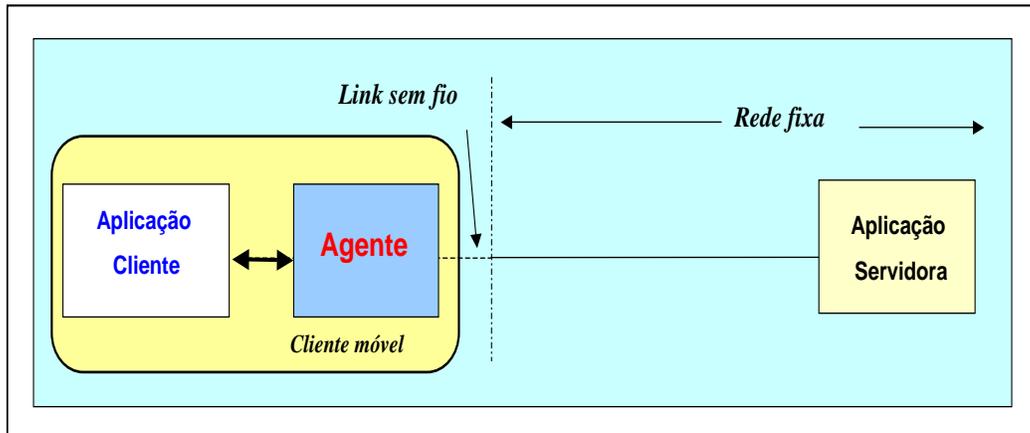


Figura 7: O modelo em três camadas Cliente/Agente-Cliente/Servidor

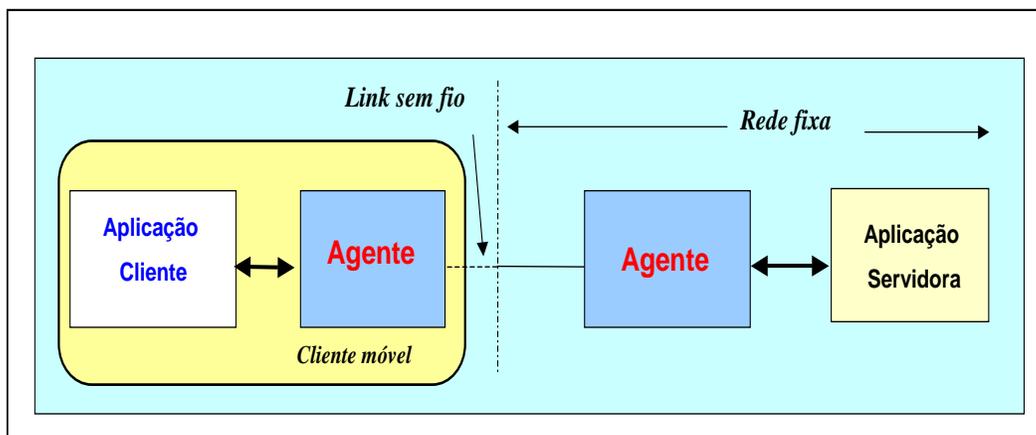


Figura 8: O modelo em três camadas Cliente/Agente-Agente/Servidor

Este modelo é transparente tanto para o cliente como para o servidor. Conseqüentemente, o par de agentes pode ser empregado em várias aplicações clientes. O protocolo de comunicação entre os dois agentes pode proporcionar uma alta eficiência na redução (compressão) dos dados e na otimização de protocolos, sem limitar as funcionalidades ou interoperabilidade do cliente. A cooperação entre os dois agentes permite uma maior otimização nos links sem fio, beneficiando diferentes aplicações. Além disso, otimizações específicas para algumas aplicações podem ser efetivamente realizadas pelos agentes.

Este modelo oferece flexibilidade no manuseio da desconexão. Por exemplo, a memória *cache* dos clientes móveis pode ser mantida pelo seu agente. Esta memória pode ser usada para atender as necessidades e solicitações dos clientes durante o período de desconexão. Na falta de memória cache, o agente pode enfileirar solicitações para serem enviadas ao servidor quando da reconexão. Similarmente, o agente situado no servidor pode acumular e transferir requisições do cliente, quando de uma nova reconexão. A

cedo possível durante os períodos de baixo tráfego na rede.

⁸*Hoarding* é um processo de carga antecipada dos dados para garantir a sobrevivência do host móvel durante a desconectividade que está por vir.

fraca conectividade também pode ser manuseada de varias formas, como por exemplo, por uma relocação computacional do agente do cliente ao agente do servidor ou vice-versa. O retardo de gravação e disponibilidade de memória progressivamente ao cliente móvel (*prefetch*) também reduz a comunicação durante a o período de fraca conectividade ou nos momentos de inatividade da rede.

Este modelo é apropriado para clientes móveis com suficiente poder computacional e que possua memória secundária. O ponto fraco deste modelo é que cada aplicação necessita de trabalho de desenvolvimento tanto no servidor quanto no cliente. Porém, não é necessário o desenvolvimento do par de agentes para cada nova aplicação. Construindo-se as funcionalidades e otimizações suficiente mente genéricas, somente será necessário o desenvolvimento de pares de agentes diferentes para aplicações do tipo de acesso a um banco de dados, aplicações Web, etc.

O Modelo Cliente/*Agente-Agente*/Servidor provê uma clara distinção entre os agentes do cliente e do servidor, bem como divide a responsabilidade da execução das tarefas necessárias ao ambiente de computação móvel entre os mesmos. Um exemplo da aplicação deste modelo é apresentado em [Villate et al., 2000].

2.4. O Modelo de Agentes Móveis

Agentes Móveis são processo enviados de um computador fonte para realizar uma tarefa específica em outro computador. São processos que migram associados com um itinerário (direção) e com uma distribuição dinâmica do seu código e do seu estado. Após a sua submissão, o agente móvel processa de forma autônoma e independente de seu emissor. Quando alcança um cliente ou servidor, se integra a um ambiente de execução de agentes. Possui a capacidade de se transportar para outro cliente ou servidor, gerando novos agentes ou interagindo com outros agentes. Quando concluído, entrega seus resultados ao cliente ou servidor emissor ou a outro cliente ou servidor.

A diferença entre os agentes móveis e os agentes estáticos não é, entretanto, demasiadamente grande. Sistemas que empregam agentes móveis, também empregam agentes estáticos associados com seus ambientes computacionais. A forma de enviar os agentes móveis para um outro ambiente para recolher as informações necessárias é análoga à forma como um agente estático emite um pedido declarativo a um outro agente, o qual contem a informação desejada. Um sistema que usa agentes móveis requer um ambiente comum para execução do agente durante toda a execução do sistema ou o mínimo possível de compatibilidade para sua execução [Kearney, 1998]. Além de seus componentes computacionais, os agentes móveis são equipados com inteligência para reagir, tomar decisões e resolver problemas. Alem desses aspectos, os fatores principais para o paradigma dos agentes móveis são:

- Sua habilidade de interagir e cooperar com outros agentes;
- Sua autonomia, ou seja, sua capacidade de executar processos com pouca ou nenhuma intervenção com o seu emissor;
- Sua capacidade de interoperabilidade, possibilitando a sua execução em diversas plataformas de software e hardware;
- Seu poder de reação, dando-lhe a habilidade de responder a eventos externos;
- Sua mobilidade, dando-lhe a capacidade de se movimentar pela rede de comunicação com e sem fio;

Dois fatores principais motivam a utilização dos agentes móveis na computação móvel. Em primeiro lugar, os agentes móveis oferecem um método assíncrono eficiente para procurar informações ou serviços, de forma rápida em uma rede. Clientes móveis são lançados na rede e procuram em toda parte pela informação ou serviço desejado. Em segundo lugar, os agentes móveis suportam a conectividade intermitente, as redes lentas e os equipamentos com poucos recursos. Este segundo fator torna a utilização dos agentes móveis muito atraente para a computação móvel.

Durante um *período de desconexão* o cliente móvel envia um agente móvel para a rede fixa e este agente age como um substituto da aplicação na rede fixa, permitindo interação durante a desconexão. De forma oposta, o agente móvel é carregado no cliente móvel, oriundo da rede fixa, quando acontece uma nova conexão. Quando em *fraca conectividade*, todo o tráfego de comunicação através do link sem fio reduz a possibilidade de submissão de um grande número de mensagens pelos agentes fixos e na obtenção de seus resultados. Por este motivo, os agentes móveis deslocam a carga de computação dos clientes móveis, pobres em recursos, para a rede fixa [Pitoura and Samaras, 1998].

Um dos principais obstáculos para a utilização e aceitação dos agentes móveis em aplicações comerciais é a sua segurança, como por exemplo, protegê-los contra vírus.

3. Os SGBDs e o ambiente da Computação Móvel

Entre os principais problemas do ambiente de computação móvel que afetarão diretamente as *transações* dos bancos de dados estão as constantes *desconexões* dos clientes móveis de suas redes fixas, a *fraca conectividade* das redes sem fio, a *mobilidade dos clientes móveis* por um amplo espaço geográfico e as constantes *falhas de recuperação* das transações. Alguns desses aspectos serão discutidos nesta seção, com o objetivo de enfatizar a importância para a gerência de dados dos SGBDs.

3.1. Transações nos SGBDs Convencionais e Distribuídos

O conceito de **transação** fornece um mecanismo para descrever unidades lógicas de processamento do banco de dados. Uma **transação** é uma unidade lógica de processamento do banco de dados que inclui uma ou mais operações de acesso ao banco de dados. Essas **transações** podem incluir operações de *inclusão*, *exclusão*, *modificação* ou *recuperação* de dados. As operações do banco de dados que formam uma transação podem se embutidas dentro de um programa da aplicação ou podem ser especificadas de maneira interativa através de uma linguagem de consulta de alto nível como a SQL. Se as operações do banco de dados em uma **transação** não atualizarem o banco de dados, mas somente recuperarem dados, a **transação** é denominada como transação somente de leitura (*read-only transaction*).

Para assegurar a integridade dos dados, o SGBD deve garantir as seguintes propriedades das transações, denominadas de propriedades **ACID**:

- **Atomicidade** - ou todas as operações da *transação* são refletidas corretamente no banco de dados ou nenhuma será;
- **Consistência** - a execução de uma *transação* isolada, ou seja, sem a execução concorrente de outra transação, preserva a consistência do banco de dados;

- **Isolamento** - embora diversas *transações* possam ser executadas de forma concorrente, o sistema garante que, para todo par de *transações* T_i e T_j , T_i tem a sensação de que T_j já terminou sua execução antes de T_i começar, ou que T_j começou a sua execução após T_i terminar. Assim, cada transação não toma conhecimento de outras transações concorrentes no sistema;
- **Durabilidade** - depois da *transação* completar-se com sucesso, fazer seu *commit*, as mudanças que ela fez no banco de dados persistem, até mesmo se houver falhas no sistema.

O acesso a diversos itens de dados em um **sistema distribuído** é normalmente acompanhado de transações que são obrigadas a preservar as propriedades ACID. Existem dois tipos de transações que devem ser consideradas nos sistemas distribuídos: as *transações locais* e as *transações globais*. As *transações locais* são aquelas que mantêm acesso e atualizam somente ao banco de dados local. Já as *transações globais* são aquelas que mantêm acesso e atualizam diversos bancos de dados locais. A garantia das propriedades ACID nas *transações globais* é bem mais complicada, uma vez que diversos sites podem participar de sua execução. Maiores detalhes sobre as transações de banco de dados nos sistemas centralizados e distribuídos podem ser encontrados em [Elmasri and Navathe, 2000], [Elmasri and Navathe, 2002], [Silberschatz et al., 1997] e [Özsu and Valduriez, 1999].

3.2. Processamento de Transações no Ambiente de Computação Móvel

Uma computação (um conjunto de instruções) que acessa dados compartilhados em um banco de dados é comumente estruturada como uma *transação atômica*, com o objetivo de preservar a *consistência dos dados* na presença de concorrência e falhas no sistema. Entretanto, uma *computação móvel* que acessa dados compartilhados não pode ser estruturado usando uma transação atômica. Isto se dá porque *transações atômicas* assumem o pressuposto de executarem isoladas, o que impede as mesma de dividirem a sua computação e de compartilhar seus estados e resultados parciais. Já na computação móvel, considerações práticas únicas deste ambiente, exigem que as computações nos hosts móveis sejam sustentadas e apoiadas por um host de suporte a mobilidade (estação de base), tanto para computação como para comunicação de dados. Isto significa que a computação móvel precisa ser estruturada como um conjunto de transações, algumas que executam no host móvel, enquanto outras executam no host de suporte a mobilidade [Chrysanthis, 1993].

Alem da necessidade da divisão de suas transações, a computação móvel também está envolvida, em primeiro lugar com a mobilidade de suas fontes de dados e de seus consumidores, e em segundo lugar por sua natureza interativa, como por exemplo pausas para entrada de dados por seus usuários. Assim, uma outra exigência que as transações atômicas não podem satisfazer é a sua habilidade de lidar com falhas parciais e prover diferentes estratégias de recuperação, desta forma minimizando os efeitos das falhas [Chrysanthis, 1993].

Uma *transação móvel* é uma transação distribuída, onde alguma parte da computação é executada no host móvel e outra parte em um host fixo, não móvel. O uso do meio sem fio e a mobilidade resultante dos consumidores e produtores de dados afetam o processamento das transações de várias formas [Pitoura and Bhargava, 1994b].

O emprego de conexões em fio resulta em transações longas, em função dos longos atrasos da rede. Além disso, *transações móveis* envolvem sites que são intermitentemente conectados ao resto da rede. Por esta razão é que os usuários de hosts móveis evitam freqüentemente o uso dos meios sem fio, uma vez que estas conexões são caras, tanto monetariamente quanto em termos de poder de consumo dos hosts móveis, ou porque os recursos da rede não estão disponíveis na sua localização geográfica. Adicionalmente, para muitas tecnologias, como os modem celulares, há uma alta carga inicial para cada comunicação. O custo efetivo da gerência de transação para cada tecnologia pode adotar a abordagem do suporte a poucas transações longas ou invés de muitas transações curtas. Finalmente, a computação móvel é mais propensa a erros, em função das freqüentes desconexões e serem mais suscetíveis a acidentes do que os hosts fixos.

A mobilidade resulta em transações que acessam sistemas de informações heterogêneas. Além disso, enquanto em projetos estáticos a localização dos usuários é fixa, no ambiente móvel mudam constantemente. Conseqüentemente, as transações móveis acessam dados de posição (localização) dos clientes que mudam rapidamente, em muitos casos em posições imprecisas. Finalmente, as transações podem envolver dados que estão dinamicamente sendo mudados de lugar [Pitoura and Bhargava, 1994b].

Nos últimos anos, diversos modelos de transações foram propostos. Entre estes modelos tantos modelos podemos citar: o modelo de *Transações Isolation-only*, proposto por [Lu and Satyanarayanan, 1994]; o modelo *Two-tier replication*, proposto por [Gray et al., 1996]; o modelo *Mobile Open Nested Transactions*, proposto por [Chrysanthis, 1993]; o modelo *Kangaroo*, proposto por [Dunham et al., 1997]; o modelo de *Clustering*, proposto por [Pitoura and Bhargava, 1995, Pitoura and Bhargava, 1999]; o modelo *Pro-motion*, proposto por [Walborn and Chrysanthis, 1997]; o modelo *Transaction Processing Agent - TXAgent*, proposto por [Buchholz et al., 2000]; o modelo desenvolvido para o projeto *Mobisnap*, proposto por [Preguica et al., 2000]; o modelo desenvolvido para o projeto *MDRTDBS*, proposto por [Lam et al., 1999]. Uma comparação interessante de alguns destes modelos pode ser encontrada em [Serrano-Alvarado et al., 2001].

3.3. Estados de Operação de um Host Móvel

Em um *sistema distribuído não móvel* as operações em seus hosts ou são totalmente conectadas ou totalmente desconectadas. Já no ambiente móvel, existem vários graus de desconexão que vão desde a desconexão total até a fraca desconexão⁹. Como resultado desta característica do ambiente móvel, um host móvel pode operar em vários estados. Adicionalmente, para conservar bateria um host móvel pode operar no estado *cochilo (doze mode)*. A Figura 9, extraída de [Pitoura and Bhargava, 1994a] resume os diferentes estados de operação que um host pode se encontrar,

Uma outra característica dos estados de operação é que eles são previsíveis. Enquanto nos ambientes distribuídos não móveis esta característica de desconexão não existe, no ambiente móvel a maioria das desconexões podem ser detectadas. Assim sendo, um protocolo especial pode ser projetado para manusear a desconexão. Além disto, visto que as desconexões são freqüentes, idealmente um host móvel deve operar de forma autônoma mesmo durante a desconexão total.

⁹Quando um terminal está conectado ao resto da rede via uma pequena largura de banda.

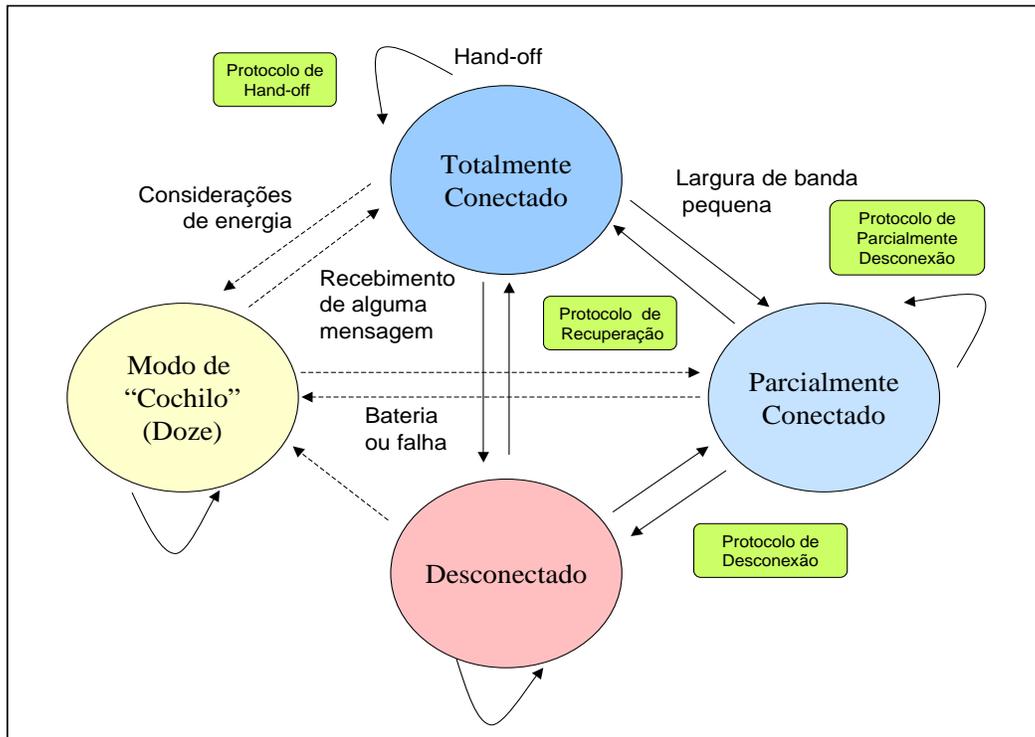


Figura 9: Estados de operação de um Host Móvel

Um host móvel pode executar um protocolo de desconexão antes de se desligar fisicamente da rede. Este protocolo visa garantir a transferência suficiente de dados para que o mesmo continue a operar durante o estado de desconexão. Um host móvel pode trocar seu estado de *parcialmente conectado* para executar no estado *parcialmente desconectado*. Enquanto permanecer neste estado, toda a comunicação com a rede deve ser limitada. Finalmente, enquanto um host móvel esta operando em um modo ele pode entrar em uma nova *célula*. Neste caso, toda a informação pertinente ao estado do host móvel deve ser transferida a estação de base da nova célula, através da execução de um protocolo de *hand-off*.

3.4. Estados de Operações Desconectadas de um Host Móvel

Os *hosts móveis* (ou clientes móveis) são constantemente desconectados de suas redes de comunicação. Um host móvel desconectado não pode enviar nem receber mensagens. Assim sendo, um host móvel que se desconecte no meio da execução de um processo (algoritmo), pode causar a suspensão da tarefa em execução até a sua nova conexão com a rede fixa. Desconexão no ambiente móvel é diferente de falha, pois a mesma pode ser voluntária ou involuntária. As desconexões são eleitas (*elective*). Assim sendo, um host móvel pode informar ao sistema, a priori, uma iminente desconexão, e então executar um protocolo específico para desconexão. Desconexões são constantemente esperadas, tornando-as uma característica típica do ambiente móvel. Com a possibilidade de prever uma desconexão, o host móvel pode executar um protocolo de desconexão antes do seu desligamento físico da rede de comunicação [Badrinath et al., 1993].

Uma nova operação nos hosts móveis com o objetivo de reduzir o consumo das baterias, principalmente em palmtops e laptops é o modo *cochilo* (*doze mode*).

Neste modo a velocidade do *clock* do processador é reduzida e nenhuma operação do usuário é executada. O host móvel simplesmente aguarda passivamente o recebimento de mensagens enviadas dos demais componentes da rede, retornando ao modo normal (de execução) quando uma mensagem é recebida. Como na desconexão, esta é uma operação voluntária. Entretanto, as implicações são diferentes. No modo *cochilo* o host móvel é alcançável pelo resto do sistema, e assim, pode ser induzido pelo sistema a retomar o seu modo de operação normal. Diferentemente dos dois tipos de desconexões, a conexão com o sistema só pode ser retomada *somente* pelo host móvel [Badrinath et al., 1993].

A idéia por trás do suporte as operações desconectadas é simples. Quando uma desconexão é percebida antecipadamente, os itens de dados são transferidos para o cliente móvel (host móvel), para possibilitar a sua operação durante o período de desconexão. A carga antecipada dos dados para garantir a sobrevivência durante a desconectividade que está por vir é chamada de **hoarding**. Segundo [Pitoura and Samaras, 1998] a desconexão pode ser descrita como a transição entre três estados:

1. Carga Antecipada de Dados (*Hoarding*);
2. Operações Desconectadas; e
3. Reintegração.

A Figura 10, extraída de [Pitoura and Chrysanthis, 2000] resume os diferentes modos de operação que um host móvel pode se encontrar.

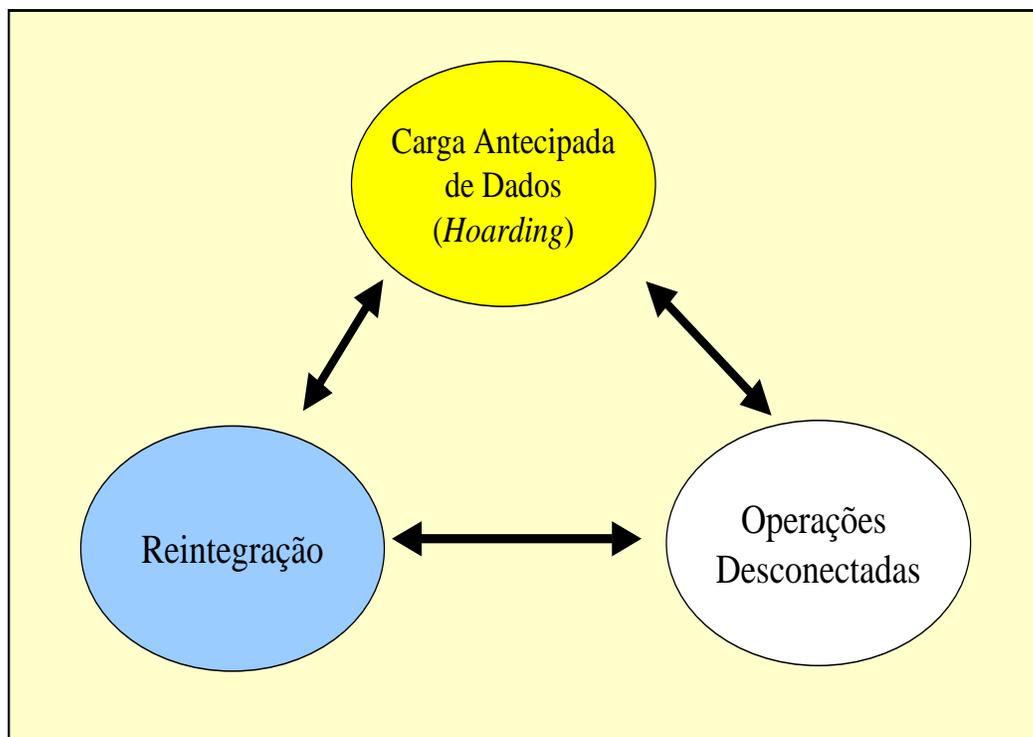


Figura 10: Estados de Operações Desconectadas de um Host Móvel

3.4.1. Carga Antecipada de Dados

Antes da desconexão o host móvel está no estado de *Carga Antecipada de Dados (Hoarding)*. Neste estado os itens de dados necessários para operação são carregados na unidade móvel. Os itens podem ser simplesmente realocados (mudados de endereço) ou movimentados do host fixo para a unidade móvel. Entretanto, fazendo assim, esses itens de dados se tornarão inacessíveis para outros sites. Como alternativa, os itens de dados podem ser replicados ou mantidos na memória cache da unidade móvel. O tipo de dados transferidos para o host móvel depende da aplicação e do modelo de dados que o sustenta. Por exemplo, no caso de *sistemas de arquivos* os dados podem ser arquivos, diretórios ou volumes de disco. No caso de *SGBDs*, os dados podem ser relações ou visões. No caso de sistemas de *browsing na web* os dados podem ser páginas. No caso do *modelo de objetos* os objetos de dados podem carregar com eles informações adicionais, tais como o conjunto de operações permitidas. No caso do *modelo baseado em agentes móveis*, os agentes podem ser transmitidos ao longo da rede para serem executados no cliente móvel. Para desconexões previsíveis, como por exemplo, a motivada pela entrada do cliente em uma região fora da sua área original de cobertura, e portanto financeiramente mais cara, os dados podem ser carregados imediatamente antes da desconexão [Pitoura and Samaras, 1998].

Uma questão complicada quando um cliente móvel está neste estado é como antecipar suas necessidades futuras de dados. Uma abordagem simplista é permitir que o usuário especifique explicitamente que itens de dados necessita carregar para suas aplicações. Uma outra abordagem é usar as informações sobre o histórico dos acessos aos dados para prever a provável necessidade futura dos dados. Que dados carregar depende também da aplicação que o sistema irá usar. Por exemplo, pode precisar de arquivos para um processador de texto ou uma tabela atualizada de preços. Além das questões anteriores, uma outra questão complexa para a *carga antecipada dos dados* ocorre quando os dados carregados são compartilhados por vários clientes ou sites que necessitam continuar acessando e modificando os dados com o host móvel desconectado. Assim, avaliando a probabilidade de operações conflitantes quando se decide que itens de dados carregar antecipadamente pode-se melhorar a efetividade das operações desconectadas. Em [Saygn et al., 2001] existe uma abordagem muito interessante para especificação da carga antecipada de dados, utilizando-se uma técnica de mineração de dados (*data mining*) chamada de regras de associação.

3.4.2. Operações Desconectadas

Quando acontece uma ruptura de conexão com a rede fixa a unidade móvel entra no estado de *desconexão*. Enquanto estiver neste estado, somente pode usar dados locais. Solicitações de dados que não estão localmente disponíveis não podem ser atendidas e retornam com indicação de erro. Estas pendências podem ser organizadas em uma fila para serem atendidas quando ocorrer à próxima conexão. Aplicações com solicitações de dados não atendidas também podem suspender sua execução ou continuar trabalhando em outro processamento que não dependa dos dados solicitados.

Existem duas abordagens para reparar as atualizações dos dados compartilhados.

A **abordagem pessimista** e a **abordagem otimista**. Na abordagem pessimista as atualizações são executadas somente em um site utilizando bloqueio (*lock*) ou alguma forma de verificação dos dados de entrada e saída (*check-in/check-out*). Na abordagem otimista as atualizações são permitidas em mais de um site com a possibilidade perigosa de conflito nas operações. As atualizações nas unidades móveis são registradas (*logged*) na memória estável do cliente. As informações são armazenadas no *log* (*diário*) do sistema. O tipo de informação afeta a eficácia da reintegração das atualizações quando da reconexão, assim como a eficácia da otimização do *log*. A otimização do *log*, mantendo-o em um pequeno tamanho, é muito importante pois este *log* é armazenado na memória do cliente móvel e reduz o seu tempo para propagação das atualizações e reintegração durante a reconexão. Operações de otimização no *log* podem ser realizadas durante as operações quando a unidade móvel está desconectada, incrementalmente cada vez que uma nova operação é inserida no *log* ou como um passo de pré-processamento antes da propagação ou aplicando o *log* na próxima reconexão [Pitoura and Samaras, 1998].

3.4.3. Reintegração

Quando acontece uma nova conexão com a rede fixa a unidade móvel entra no estado de *reintegração*. Neste estado, as atualizações do host móvel são reintegradas com as atualizações de outros sites, re-executando o seu *log* no host fixo. Questões referentes a concorrência e seriação das transações são efetivadas em função de cada sistema em particular, visando resolver o problema da atualização sobre o mesmo objeto.

A tabela 2, extraída de [Pitoura and Samaras, 1998], apresenta os principais problemas e questões relacionadas ao estado de desconexão de uma unidade móvel.

3.4.4. Desconexão e os SGBDs

Existem muitos problemas que permanecem abertos quanto a questão da carga de dados antecipada (*hoarding*) em banco de dados. A primeira questão em aberto é determinar qual deve ser a granularidade dos dados para a carga antecipada. Por exemplo, no caso dos SGBDs relacionais vai desde uma tupla, um conjunto de tuplas ou toda uma relação. Uma abordagem lógica seria carregar os dados em função das consultas realizadas, por exemplo, carregando antecipadamente de forma progressiva (*prefetching*) apenas os dados que constituem a resposta a uma dada pergunta. Isso corresponde a carregar na unidade móvel uma *visão materializada*. Uma outra questão é como decidir que dados carregar. Em termos de visões, significa identificar as visões materializadas ou como especificar as consultas para as cargas que definem as visões. Usuários podem explicitamente especificar quais as consultas para a carga antecipada. Uma outra alternativa, é pela avaliação do comportamento das consultas anteriormente realizadas, para que o próprio sistema tente identificar futuras prováveis necessidades dos usuários. Em ambos os casos o sistema carregaria automaticamente o conjunto de dados mais comumente usado ou por último referenciado nas consultas dos usuários.

Existem algumas particularidades que se deve considerar quando se rever o problema de ruptura (partição, divisão) das redes, particularmente na identificação do

Estado	<i>Problemas e Questões</i>	<i>Abordagens</i>
Hoarding	Unidade de carga	Depende do sistema (por exemplo, arquivos ou fragmentos de BD)
	Que itens carregar ?	- Especificado pelo usuário - Usar as informações sobre o histórico dos acessos aos dados - Depende na aplicação para que o sistema é usado
	Quando executar a carga dos dados ?	- Antes da Desconexão - Em uma base regular
	Solicitação para dados não disponíveis localmente	- Retorna uma exceção/erro - Solicita uma fila para futuros serviços
Desconexão	O que por no LOG ?	- Valores dos dados - Timestamps - Operações
	Quando otimizar o LOG ?	- Nas operações locais (incrementação) - Antes da integração
	Como otimizar o LOG	Depende do sistema
Reintegração	Como integrar ?	Reexecutando as operações do LOG
	Como resolver os conflitos ?	- Usar semântica das aplicações - Soluções automáticas - Ferramentas para ajudar ao usuário

Tabela 2: Principais problemas e questões relacionadas ao estado de desconexão de uma unidade móvel

local ideal para se executar algum processo para a computação móvel. As *transações* de banco de dados executadas em um sistema distribuído são de igual importância, de mesmo nível, seja ela executada no servidor ou no cliente distribuído. Na computação móvel, entretanto, as transações executadas nos host móveis são frequentemente consideradas de *segunda classe*. Uma outra questão é a frequência da desconexão na computação móvel. A ruptura da rede corresponde a um comportamento de falha, visto que desconexões são comuns na computação móvel. Somente este fato já justifica a construção de sistemas para tratar as transações de banco de dados. Finalmente, muitas desconexões na computação móvel podem ser consideradas previsíveis [Pitoura and Samaras, 1998].

Uma tendência comum é fazer o (*commit*) da transação de forma temporária na unidade móvel, fazendo com que seus resultados sejam visíveis apenas para uma próxima transação na mesma unidade móvel. Por exemplo, um vendedor de bebidas pode ter carregado em seu host móvel um valor de estoque para cada bebida e a cada venda realizada esta quantidade é atualizada, neste caso, diminuindo a sua quantidade disponível para venda para este vendedor. Na reconexão, ocorre um processo de certificação, durante o qual toda transação temporariamente confirmada (*committed*) no host móvel é validada contra uma aplicação ou por um critério de exatidão definido pelo sistema. Se o critério for satisfeito, então a transação é confirmada. Por exemplo, o critério no nosso exemplo anterior, poderia ser existir bebidas em estoque para entrega no prazo combinado

entre o vendedor e o comprador, uma vez que o estoque total da empresa é fornecido diariamente a todos os vendedores. Caso o critério não seja satisfeito, a transação deve ser cancelada, reconciliada ou compensada. Isto realmente se torna um problema, pois estas ações podem ter efeito em cascata sobre as transações temporariamente confirmadas no host móvel. Para resolver este problema [Lu and Satyanarayanan, 1994] e [Lu and Satyanarayanan, 1995] introduziram o conceito de transações Isolation-only (IOTs) e [Gray et al., 1996] o conceito de transações no esquema de *Two-tier Replication* para computação móvel.

3.5. Fraca Conectividade

Fraca conectividade é a conectividade fornecida por redes de comunicação de dados lentas e ou monetariamente caras para seu usuário. Além disso, a conectividade nessas redes são oferecidas em pequenos períodos de tempo. A fraca conectividade causa várias limitações que não estão presentes quando a conectividade é normal e estimula a revisão de vários protocolos. Uma característica adicional da fraca conectividade na computação móvel é a variação de sua intensidade. Conectividade em computação móvel varia no custo, fornece largura de banda e segurança/confiança. Muitas propostas para tratamento da fraca conectividade examinam estas características e fornecem suporte para operações que se adaptam ao corrente grau de conectividade. Neste sistemas, operações desconectadas são exatamente a forma de operação no extremo total da falta conectividade. O objetivo da maioria das propostas para tratamento da fraca conectividade é a prudência no uso da largura de banda. Elas aceitam trocar a confiança (exatidão) do dado transmitido pela a redução dos custos na comunicação [Pitoura and Samaras, 1998].

3.6. Fraca Conectividade e os SGBDs

Abordagens para tratar a fraca conectividade em **SGBDs** objetivam minimizar a comunicação e sobreviver a pequenas desconexões. Entretanto, devido à complicada dependência entre os itens de dados do banco de dados, o problema é complexo. Um host móvel pode desempenhar vários papéis em sua ambientação em um banco de dados distribuído. Por exemplo, ele pode simplesmente submeter operações para serem executadas em um servidor ou para um agente residentes na rede fixa.

Uma abordagem diferente no papel do host móvel é reservada ao processamento local no mesmo. Assim, é necessário permitir operações autônomas durante a desconexão, porém, complicando a gerência dos dados e podendo causar um inaceitável aumento no tempo de processamento (*overhead*) na comunicação. Colocando os dados no host móvel causa novos problemas no projeto físico do banco de dados. Tais problemas incluem, entre outros, como fragmentar de forma apropriada o banco de dados e como alocar os fragmentos no host móvel e fixo. Quando o fragmento é alocado ele pode, também, ser replicado em alguns sites, ou em todos os sites ou mantido como uma única cópia. Em geral, manter os dados como uma única cópia dos dados no host móvel não é apropriada em função da disponibilidade e segurança dos mesmos [Pitoura and Samaras, 1998].

O controle de concorrência nos casos de transações que envolvem tanto host móveis quanto fixos é complicado. Para transações que acessam dados tanto nos hosts móveis quanto nos fixos via conexões sem fio, certamente, irão impor um maior *overheads*. Por exemplo, no caso do protocolo de controle de concorrência pessimista

que requer que a transação adquira bloqueios (*locks*) em vários sites, as transações podem bloquear se pedirem bloqueios em sites que estão desconectados ou pedirem bloqueios a sites bloqueados por transações desconectadas. Por outro lado, técnicas tais como (*timestamps*) podem levar a um grande número de transações canceladas (abortadas) em função da lentidão das operações na rede. Para evitar os atrasos pela lentidão das conexões sem fio, um modelo de transações chamado de *Mobile Open Nested Transactions* foi proposto por [Chrysanthis, 1993].

3.7. Mobilidade

Um sistema distribuído que inclua elementos móveis é dito dinâmico. Unidades móveis se comunicam diretamente com diferentes estações de base, como também se movimentam dentro e para fora de sua área, em uma nova célula. A *mobilidade* também significa que os clientes se movem e executam atividades de leitura/gravação para realizar suas funcionalidades. Por esta razão, também alguns dados ou algumas tarefas precisam se movimentar juntamente com as unidades móveis, principalmente, por questões de performance. Também, muitos sites se tornam indisponíveis durante a desconexão. Além disso, o custo da comunicação na computação móvel é diferente do custo da computação fixa. Primeiramente porque existe a necessidade de se determinar exatamente a localização do host móvel e segundo porque pode ser baseado na duração da conexão e não no número das mensagens enviadas. Assim, os algoritmos para distribuição de dados e tarefas dos sistemas distribuídos precisam ser revisados para a computação móvel.

Muitos parâmetros se modificam com o tempo no ambiente da computação móvel. Assim sendo, deixar os dados localizados em locais fixos pode não ser apropriado. Em [Huang and Wolfson, 1994] é feita uma distinção entre os custos de replicação na computação fixa e na computação móvel, baseado no pressuposto de que na computação móvel, o custo de entrada/saída (*input/output*) de dados se torna insignificante, em função da carga da comunicação sem fio. Também é introduzido um algoritmo de alocação dinâmica de dados que tem a propriedade de sempre que um processador lê uma cópia, salva essa cópia em um banco de dados local. Então, este algoritmo dinâmico é comparado ao esquema de alocação estática e demonstra ser superior.

Em [Huang et al., 1994] é estudado o caso em que computadores móveis acessam dados de forma on-line em hosts fixos. O problema discutido é se é benéfico, em termos de custo de comunicação, guardar uma cópia dos dados localmente no host móvel. Intuitivamente, guardar uma cópia local é apropriado para itens de dados que são frequentemente lidos, quando comparados com a sua taxa de gravação.

3.8. Falhas de Recuperação

Recuperação em ambiente móvel tem se concentrado em gravar *checkpoints* (pontos de verificação) de consistência global de uma aplicação distribuída móvel. O processamento de recuperação para transações móveis dentro do contexto desses algoritmos de *checkpoints* não estão ainda bem amadurecidos. As características do ambiente móvel e sem fio que são levados em consideração na definição dos esquemas de *checkpoints* são as seguintes:

- A *mobilidade dos hosts móveis de célula para célula*, em que o problema é encontrar o melhor lugar para armazenar o próximo local do *checkpoints*;

- A *disponibilidade de armazenamento estável no host móvel*, que determina o grau de participação do host móvel no processo de *checkpoints*;
- A *largura de banda*, que juntamente com a disponibilidade de armazenamento estável, refina a participação do host móvel no *checkpoints* e afeta a frequência em fazer o *checkpoints*.

3.8.1. *Checkpoints no ambiente Móvel*

Em sistemas distribuídos, a recuperação de falhas é baseada nos protocolos de *checkpoints*. Esses protocolos periodicamente armazenam o estado da aplicação em armazenamento estável (seguro). Após uma falha, a aplicação usa esses *checkpoints* para desfazer (*roll back*) até o último ponto salvo e reiniciar a execução. Um estado global é armazenado e usado pelo protocolo para recuperar a aplicação. O estado global inclui o estado de cada processo participante na aplicação distribuída e possivelmente muitas mensagens. Para uma correta recuperação, o protocolo deve conservar recuperável o estado global da aplicação. Maiores detalhes sobre protocolos em sistemas distribuídos podem ser encontrados em [Özsu and Valduriez, 1999].

Em [Acharya and Badrinath, 1994] um *checkpoint global* é definido como um *checkpoint local* consistente para cada host móvel/processo participante na aplicação. Considerando que estamos lidando com hosts móveis que enviam e recebem mensagens, [Ssu et al., 1998] traduziu nas seguintes condições de recuperação e consistência:

- **Consistência:** Um *checkpoint global* é consistente se as seguintes condições são satisfeitas: para cada mensagem m , se o *recedor da mensagem* (m) é incluído no *checkpoint global*, então o *emissor da mensagem* (m) também é incluído no *checkpoints global*.
- **Recuperabilidade:** Para evitar a perda de mensagens *em trânsito* que foram enviadas, porém, não recebidas por nenhum outro processo, se o *checkpoint global* conter o *emissor da mensagem* (m), mas não conter o *recedor da mensagem* (m), então o *protocolo de checkpoint* deve salvar a mensagem (m) também.

Os algoritmos de *checkpointing* são classificados em duas categorias: *coordenados* e *não-coordenados*. Os *protocolos coordenados* necessitam que cada participante coordene seus *checkpoints locais* para garantir a recuperação e consistência do *checkpoint global*. Os *protocolos não-coordenados*, por sua vez, permitem aos participantes *checkpoints* independentes em seu estado local. Durante a recuperação, um esforço de coordenação é necessário para selecionar um *checkpoint* para cada participante criar o *checkpoint global* consistente.

Devido à mobilidade, pequena largura de banda e desconexões, ambos os tipos de protocolos, em sua forma original, são inadequados para a computação móvel. *Protocolos coordenados* requerem controle de mensagens enviadas a diferentes hosts móveis para sincronizar o processo de *checkpointing*. Isto envolve um custo de pesquisa para encontrar a localização do host móvel. O processo pode ser ainda mais complicado, pois o host móvel pode se movimentar para outra célula antes do processo de *checkpointing* ter sido concluído. Os esquemas coordenados também são afetados pela desconexão. Durante a desconexão, os *checkpoint locais* dos hosts móveis ficam inacessíveis para o

algoritmo de sincronismo do *checkpointing*, tornando assim a execução problemática. *Protocolos não-coordenados* permite aos participantes móveis o estado de *checkpointing* sem troca de mensagens. Isto é adequado para o ambiente móvel porque permite que o processo de *checkpointing* continue durante a desconexão. Entretanto, mensagens necessitam ser trocadas durante a recuperação para encontrar o *checkpointing global* ou pegar informações sobre outros participantes [Pitoura and Samaras, 1998].

3.8.2. Protocolos de Checkpoints Móveis

Um pequeno número de adaptações para os protocolos de *checkpointing* coordenados e não-coordenados tem sido desenvolvidos para manusear as limitações do ambiente de computação móvel. Estes protocolos podem ser classificados baseados no seu grau de adaptabilidade e se os armazenamentos estáveis são considerados um lugar relativamente seguro para armazenar o estado local da aplicação. Assim sendo, os *checkpointing* são classificados como *soft* ou *hard* baseados nos *tipos de falhas* que podem ocorrer. *falhas soft* são aquelas que não causam danos permanentes aos host móveis, como por exemplo descarga de bateria ou problemas no sistema operacional. Já as *Falhas hard* são aquelas que causam danos permanentes aos host móveis. *Falhas hard* são manuseadas pelo chamado *checkpoint hard* que são armazenados na rede fixa, enquanto as *falhas soft* são distribuídas por um chamado *checkpoint soft* que são armazenados localmente no host móvel. *Checkpoints* armazenados localmente, são fáceis de serem criados e sobretudo, permite que o host móvel continue funcionando durante a desconexão [Pitoura and Samaras, 1998].

A tabela 3, extraída de [Pitoura and Samaras, 1998] apresenta características desejadas dos grupos de protocolos em torno do grau de conectividade.

	Conectado	Desconectado	Fraca Conectividade
Tipo de Login	Imediato ou periódico	Periódico	Periódico
Razão do Checkpoints	<i>hard = soft</i>	Somente soft	<i>soft > hard</i>
Tipo de Coordenação	Coordenado ou não-coordenado	Não-coordenado	Não-coordenado
Clientes com poucos recursos (armazenamento estável inseguro)	Equilibrado ou minimizado o uso do cliente	Maximizado o uso do cliente (somente memória do cliente)	Maximizado o uso do cliente
Clientes com muitos recursos (armazenamento estável seguro)	Equilibrado ou maximizado o uso do cliente	Maximizado o uso do cliente (somente memória do cliente)	Maximizado o uso do cliente

Tabela 3: Conectividade e Checkpointing

Entre os trabalhos de destaque podemos citar [Acharya and Badrinath, 1994] e [Pradhan et al., 1996], os quais apresentam três protocolos não-coordenados que consideram os armazenamento em disco dos hosts móveis como instáveis e

conseqüentemente inapropriados para armazenar o estado de um participante. [Ssu et al., 1998] e [Prakash and Sihghal, 1996], entretanto, assumem que os Hosts móveis tem um armazenamento estável relativamente seguro e assim podem participar no processo de checkpoint como um host fixo.

4. Gerência de Dados Móveis

Esta seção está focada nas técnicas de gerência de informação e recuperação em sistemas móveis. Destaca-se a importância das técnicas de *Disseminação ou Difusão de Dados (Broadcast)* como meio de entregar a informação aos hosts móveis e o impacto da computação móvel na gerência de consultas.

A maioria das tecnologias sem fio suportam a disseminação de dados *broadcast* para todos os elementos móveis que estão em uma célula. No modelo cliente/servidor, o servidor ou o agente localizado na estação de base pode tirar vantagem desta característica para disseminar informações para todos os clientes de sua célula. Assim, é estabelecida uma nova forma de enviar dados aos clientes, diferente da forma tradicional de envio de dados somente por solicitação (sob demanda) dos clientes. A Figura 11, apresenta uma arquitetura genérica para utilização da técnica de *broadcast*.

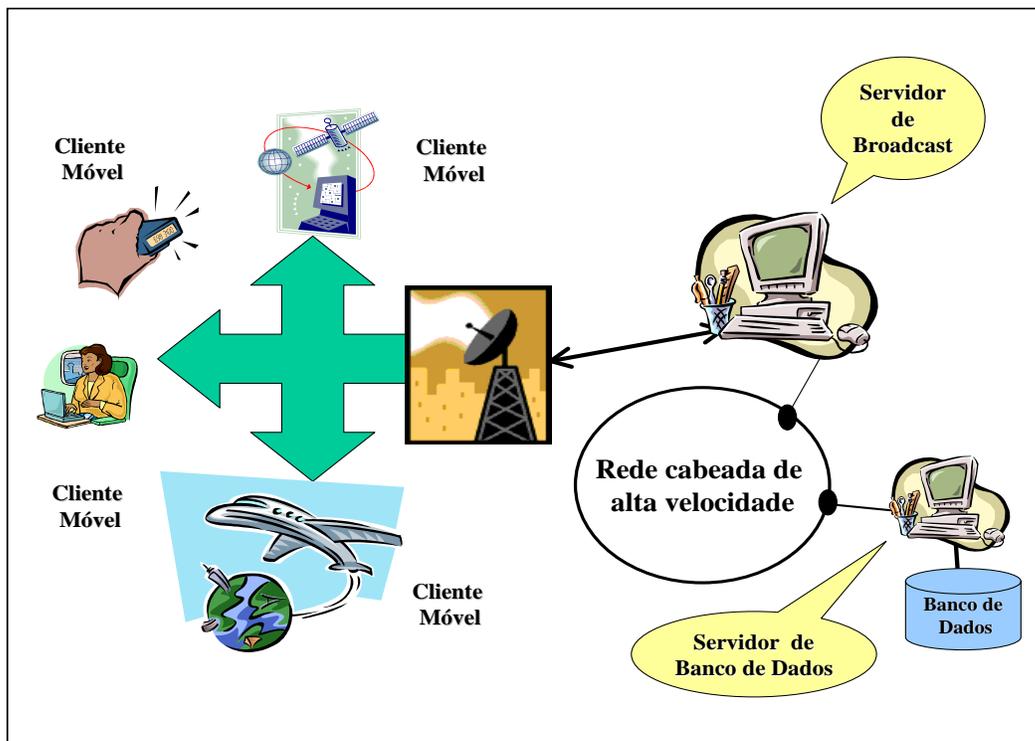


Figura 11: Arquitetura genérica para utilização da técnica de *broadcast*

A *gerência de consultas* para a computação móvel deve tratar de forma eficiente consultas que envolvem a localização do usuário móvel. A localização do usuário móvel introduz vários novos problemas referentes ao armazenamento e atualização desta informação. A gerência de consulta também deve trabalhar com os recursos limitados dos hosts móveis, nos quais destacam-se o pequeno tamanho de sua tela, o escasso poder de

suas baterias, o longo período de desconexão dos clientes, as transações longas de banco de dados, o pouco recurso de memória, entre outros [Brayner and Monteiro, 2000].

4.1. Disseminação ou Difusão de Dados (*Broadcast*)

No modelo cliente/servidor tradicional, os dados são enviados para o cliente somente sob demanda, ou seja, atendendo a uma solicitação do cliente. O servidor recebe a solicitação, processa-a e retorna ao cliente a informação solicitada. Esta forma de entrega dos dados é chamada de *pull-based* (puxar/buscar na base). Na computação sem fio, os servidores fixos são providos de canais com relativa largura de banda que suportam a entrega dos dados por *broadcast* ao clientes móveis em suas células. Esta facilidade de infra-estrutura propiciou a uma nova forma de entrega de dados chamada *push-based*, que consiste do servidor, repetitivamente, disseminar dados ao universo de clientes sem uma solicitação específica por parte dos clientes. Os próprios clientes monitoram a difusão dos dados (*broadcast*) e recuperam os itens de dados que eles precisam, da forma como eles chegam na transmissão. A Figura 12, extraída de [Pitoura and Chrysanthis, 2000], apresenta a forma de *broadcast push-based*.

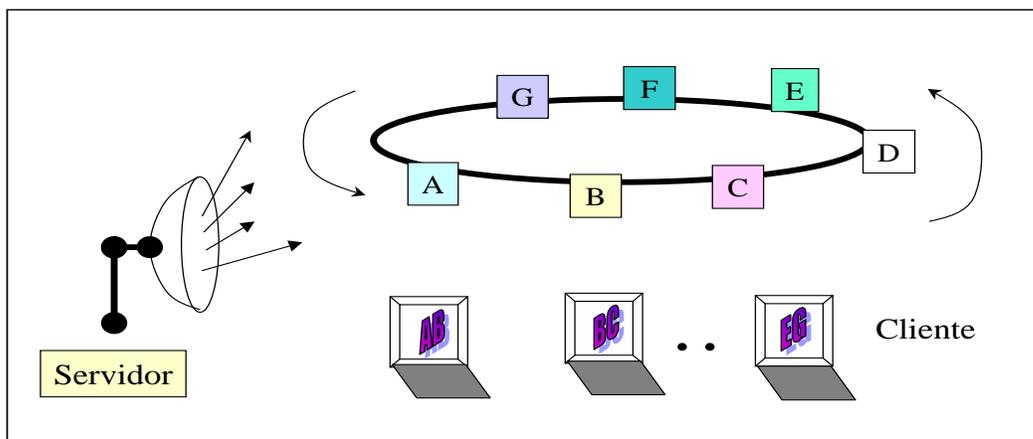


Figura 12: Forma de *broadcast push-based*

A entrega de dados na forma *broadcast push-based* (impulsionada pela base) é extremamente importante para um grande conjunto de aplicações que necessitam disseminar informações para um grande número de clientes. Entre estas aplicações podemos citar as aplicações comunicados de emergência a policiais de uma determinada área, controle de tráfego de automóveis, informações sobre as condições de tempo, renovação e novas cotas de estoques de vendedores, entre outras. A idéia da entrega de dados pela técnica de *broadcast* não é nova, os primeiros trabalhos são originários das tecnologias de Tele-Texto e Vídeo-Texto. Recentemente o *broadcast* ganhou mais atenção nos sistemas sem fio, devido ao suporte físico para *broadcast* tanto nas redes de celulares como nas redes de satélites. Nas redes de celulares constantemente recebemos comunicados das operadoras com ofertas, ou mesmo os chamados *torpedos* coletivos.

A entrega dos dados na forma de *broadcast push-based* (impulsionada pela base) é apropriada para os casos em que a informação precisa ser transmitida para um grande número de clientes que possuem interesses comuns. Neste caso, os servidores guardam

diversas mensagens que nos sistemas com a forma de entrega dos dados *pull-based* teriam que enviá-las individualmente. Além disso, o servidor é impedido de ser *bombardado* por inúmeras solicitações dos clientes. A entrega de dados na forma *push-based* é escalável em seu desempenho, pois não depende do número de clientes ouvindo os canais de *broadcast*. Por outro lado, a entrega dos dados na forma *pull-based* não pode ser escalável além da capacidade do servidor ou da rede. Uma limitação na entrega de dados em *broadcast* é que o acesso é somente seqüencial. Os clientes precisam esperar até que a solicitação de dados esteja (apareça) no canal de comunicação para então pegá-la. Assim, a latência¹⁰ degrada com o volume de dados existente no *broadcast*, que é compatível com o tamanho do banco de dados. Na forma de entrega *pull-based*, os clientes podem ter um papel mais ativo e explicitamente solicitarem seus dados ao servidor.

As formas de entrega *pull-based* e *push-based* podem ser combinadas, considerando os sistemas em que além do canal de *broadcast*, os clientes possuem um canal de alta velocidade (*uplink*) para comunicação com o servidor, também chamado de *backchannel*, no qual os usuários enviam suas mensagens. Esta abordagem é também conhecida como entrega de dados *híbrida*. Uma característica importante desta abordagem, está no fato de que se o mesmo canal do servidor até os clientes é usado para as duas formas de *broadcast* e também para transmissão de respostas as consultas sob demanda dos clientes, então existe a necessidade do uso de técnicas para um eficiente compartilhamento do canal de comunicação. Os clientes podem usar o *backchannel* de várias formas, como por exemplo, para solicitar dados diretamente ao servidor, como no casos em que os dados requerem sigilo e portanto não podem aparecer no canal de *broadcast*. A Figura 13, extraída de [Pitoura and Chrysanthis, 2000], apresenta a forma de *broadcast* com a combinações das formas *push-based* e *pull-based*, na qual existe um compartilhamento de canal [Pitoura and Samaras, 1998].

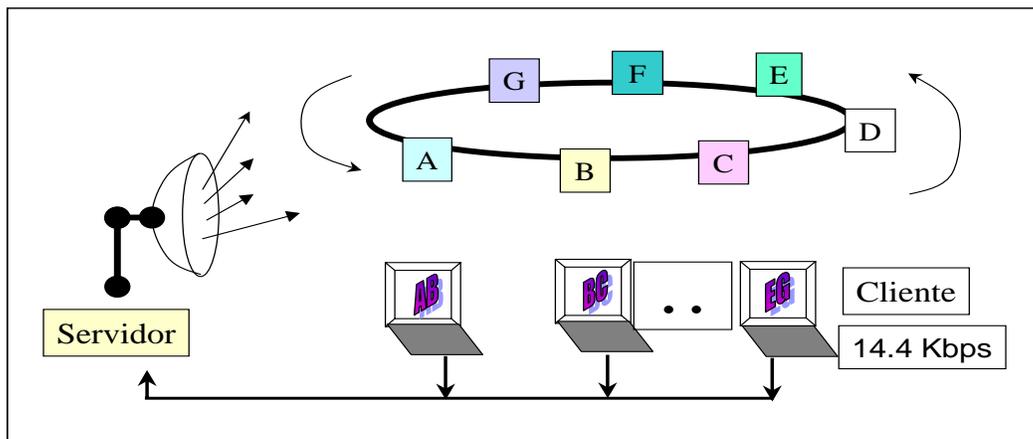


Figura 13: Forma de *broadcast* com as combinações *push-based* e *pull-based*

Em vez de *broadcasting* de todos os itens de dados do banco de dados, um caminho para realizar a entrega dos dados na forma híbrida é particionar os itens de dados em

¹⁰Período de inatividade entre um estímulo e a resposta por ele provocada, neste caso entre uma solicitação ao servidor e sua resposta.

dois conjuntos: um para ser usado nas transmissões de *broadcasting* e o outro para ser disponibilizado por demanda dos clientes. Determinar que subconjunto de dados poderá ser transmitido via *broadcasting* é uma tarefa complicada, uma vez que esta decisão depende de muitos fatores, incluindo o padrão de acesso dos clientes e a capacidade dos servidores de atender aos serviços por demanda dos clientes.

A mobilidade dos usuários também é um fator crítico na determinação de que conjunto de dados selecionar para o *broadcast*. As células podem diferir em seus tipos de estrutura de comunicações, e assim, em sua capacidade de prestação de serviços as solicitações dos servidores. Além disso, como os usuários estão em constantes movimentos entre as células, a distribuição das solicitações para um específico conjunto de dados muda em cada célula. Em [Datta et al., 1997] são apresentadas duas variações adaptáveis de um algoritmo que leva em conta a mobilidade dos usuários entre as células. Este algoritmo seleciona estatisticamente os dados do *broadcast* em função do perfil e registros dos usuários em cada célula.

4.2. Uso de Caches em *Broadcast*

O armazenamento de dados na memória dos hosts móveis (*caching*) pode ser desenvolvido junto com o *broadcast* nos sistemas de bases para disseminação (textitbroadcast-based systems). Clientes podem colocar em sua memória *cache* itens de dados para diminuir a sua dependência com a escolha do servidor sobre que itens de dados priorizar para o *broadcast*. Esta prioridade de escolha do servidor pode não ser ótima para um cliente específico, uma vez que o servidor pode basear sua prioridade de escolha na frequência média dos itens de dados dos *broadcast* anteriores, os quais cobriram uma população de clientes com necessidades diferentes. Além disso, as necessidades de um cliente podem se modificar ao longo do tempo. Em alguns casos, o armazenamento de itens de dados na memória *cache* reduzem a expectativa de atraso para acesso a esses dados. Em um outro contexto, o *broadcast* pode ser usado para propagar atualizações de dados na memória *cache* em alguns sistemas cliente/servidor, independentemente de ser a transmissão, na entrega dos dados, sob demanda, ou uma combinação das duas formas de *broadcast*, *push-based* e *pull-based*.

Políticas de substituição especificam as estratégias para substituir os itens de dados na memória *cache* dos hosts móveis. Nos sistemas de gerência de cache tradicionais, os clientes armazenam na memória *cache* seus dados mais importantes para melhorar a taxa de acesso aos mesmos. Em geral, em tais sistemas, o custo de obtenção de uma página¹¹ perdida na memória *cache* é considerado constante, e assim não é contabilizado durante a substituição (troca) da página. Entretanto, nos sistemas de *broadcast*, o custo da prestação de serviços de manutenção de uma página em falta depende de quando a página solicitada irá aparecer no próximo *broadcast*. Assim sendo, [Acharya et al., 1995a] e [Acharya et al., 1995b] criaram o custo básico de substituição de página, onde o custo de se obter uma página ausente na memória *cache* deve levar em consideração a decisão de que página substituir.

Um cliente pode disponibilizar (reservar) páginas (*prefetch*) em sua memória *cache* prevendo acessos futuros. Usando *prefetching* ao invés de substituição de página, pode-se reduzir o custo da ausência de um conjunto de dados em uma página. Diversas

¹¹Parte da RAM de tamanho fixo.

estratégicas para *prefetching* podem ser usadas. Usando-se a heurística de reserva *PT* de *prefetching*, busca-se decidir se a página corrente transmitida é mais valiosa que alguma outra página que já está na memória cache. Esta heurística calcula a probabilidade de acesso (*P*) multiplicada pelo tempo (*T*), que é o tempo decorrido antes que a página apareça de novo no *broadcast*. Com este valor, uma página transmitida substitui na memória cache a página com o menor valor de *PT* [Acharya et al., 1995a].

4.3. Consulta aos Dados de Localização (posição) dos Clientes

A gerência de consultas que envolve objetos que se movem é um problema desafiador, uma vez que os dados de localização (posição) de um cliente móvel, seu contexto, pode mudar rapidamente. Assim, o resultado da consulta pode depender tanto da localização do usuário quanto do tempo para este usuário chegar a um outro local. Por exemplo, em cliente gostaria de saber se, em função da sua posição, ele está perto de um determinado prestador de serviço ou se o tempo até que ele possa chegar a este prestador de serviço é menor do que uma hora. Consultas podem não diretamente incluir a localização, mas podem solicitar seguir ao encalce (*tracking*) de objetos móveis. Por exemplo, consultas que envolvem a produção de dados e a localização dos hosts móveis, como é o caso de um grupo de vendedores de produtos alimentícios consultando e atualizando os seus estoques em um servidor localizado na rede fixa. Consultas de posição podem ser impostas tanto por usuários móveis ou estáticos e podem incluir bancos de dados localizados tanto em sites móveis ou estáticos.

Consultas de localização podem incluir dados transientes, que são dados nos quais seus valores mudam enquanto a consulta estar sendo processada. Por exemplo, um usuário em movimento querendo saber o quanto ele está próximo de um hospital. Outro tipo possível de consulta de localização são as *contínuas consultas*. Por exemplo, uma ambulância em movimento perguntando por um local de um acidente em um raio de 5 quilômetros e solicitando que a resposta a esta consulta seja continuamente atualizada.

A posição (*localização*) física atual de um usuário em movimento muda rapidamente. Controlar o volume das atualizações da posição, atualizando a posição armazenada de um usuário em movimento pode ser demorada. Uma outra forma para reduzir o tráfego de atualizações é a manutenção de informações aproximadas da posição, especificando, por exemplo, somente a zona ou partição onde o usuário reside naquele momento. Neste caso, para responder a posição, um processamento extra será necessário para calcular a exata posição do usuário. Entretanto, em uma variedade de consultas de posição, conhecer a exata posição de alguns usuários pode não ser necessário. Assim, um novo problema que aparece no *processamento de consultas* é como obter um *plano de execução da consulta ótimo* para a consulta da posição, que adquirirá somente esta informação faltante para esta resposta. Na próxima seção a *gerência de localização* será tratada com mais detalhes.

4.4. Novos requisitos e funcionalidades para a gerência de informação

Além dos itens citados anteriormente nesta seção, também são necessários para a gerência de informações a *textit*manutenção de visões, o *processamento de consultas com um consumo eficiente de energia* nos hosts móveis e *as interfaces com o banco de dados*.

A **manutenção de visões** na rede nas estações móveis e fixas introduz um número de novos parâmetros. Primeiro, visões podem conter dados de posição e tempo,

constantemente modificados. Segundo, as transmissões por *broadcast* é uma nova opção. Um caminho possível para incorpora o *broadcast* na manutenção de visões é por transmissão de relatórios inválidos por *broadcast* e atualizações de visões em uma base de demanda. Finalmente, exames de desconexão é também um fator importante [Wolfson et al., 1995].

O **processamento de consultas com um consumo eficiente de energia** nos hosts móveis busca levar a vida útil da bateria a um extremo. Um critério de otimização é desenvolvido que minimize a energia gasta pelo host móvel, enquanto não permite que o *throughput*¹² total degrade além dado de um ponto inicial.

O projeto das **interfaces do banco de dados** para a computação móvel enfrenta várias restrições. Essas restrições incluem o tamanho reduzido das telas, a necessidade para uma interface simples, a lenta e cara comunicação sem fio e o limite das baterias [Alonso et al., 1992]. Um facilitador para o processamento de consulta, chamado *Query by Icons (QBI)*, que contempla as restrições citadas anteriormente está descrito em [Massari et al., 1996]. As funções do QBI em uma interface com uma linguagem visual icônica, permitem ao usuário compor uma consulta ao banco de dados por manipulação de ícones. Um ponteiro, do tipo *light-pen*, é usado para manipular a limitação de tamanho. Além disso, o modelo semântico de dados do QBI esconde do usuário os detalhes do banco de dados e uma ferramenta de meta-consulta ajuda na formulação da consulta durante a desconexão.

5. Gerência de Localização dos Clientes Móveis

Com o crescimento do número de usuários utilizando equipamentos móveis e o aumento da quantidade de serviços oferecidos a estes usuários, é esperado que o tráfego da movimentação desses usuários e os bancos de dados para a gerência da localização (posição) desses usuários cresçam de forma bastante rápida. Assim sendo, desenvolver estratégias para a gerência da localização dos clientes móveis é uma questão central para a computação móvel [Pitoura and Samaras, 1998].

5.1. O problema da localização

A gerência de localização é um importante problema na computação móvel. Esta gerência consiste, basicamente, em *atualizar, procurar e procurar/atualizar* a localização do usuário móvel. Uma *atualização* ocorre quando o host móvel o altera a sua localização. Uma *procura* (busca) ocorre quando um host, por exemplo um servidor de banco de dados, deseja se comunicar com um host móvel cuja localização é desconhecida. Uma *procura/atualização* ocorre, após uma procura com sucesso, quando o host solicitante atualiza a informação referente à posição do host móvel procurado. A meta de um bom esquema de gerência de localização é oferecer um método eficiente para procura e atualização. O custo da procura e atualização de uma localização é caracterizada pelo número de mensagens e a distancia que a mensagem necessita viajar. Uma gerência de localização eficiente deve tentar minimizar essas quantidades [Krishna et al., 1994].

¹²Representa o número de processos (tarefas) executados em um determinado intervalo de tempo. Quanto maior, maior o número de tarefas executadas. Sua maximização é sempre desejada

As abordagens para armazenar a informação da localização de um host móvel oscilam entre dois extremos. De um lado, a informação atualizada da exata localização de todos os usuários é mantida em cada e em todas as localizações da rede. Neste caso, localizar um usuário se reduz a uma consulta no banco de dados. Por outro lado, toda vez que a localização do usuário se altera, um grande número de bancos de dados de localizações deve ser atualizado. No outro extremo, nenhuma informação é armazenada em qualquer site. Neste caso, para localizar um cliente em uma busca global, todos os sites da rede devem ser consultados. Entretanto, não há custo associado com a atualização dos bancos de dados de localização [Pitoura and Samaras, 2001].

Entre esses dois extremos, várias abordagens que tentem balancear o custo da procura (*lookups*) contra o custo da atualização são cabíveis. Essas abordagens comprometem a disponibilidade, a precisão ou a "corrência" (*currency*) da informação armazenada para cada usuário. A Figura 14, extraída de [Pitoura and Chrysanthis, 2000], apresenta essas abordagens.

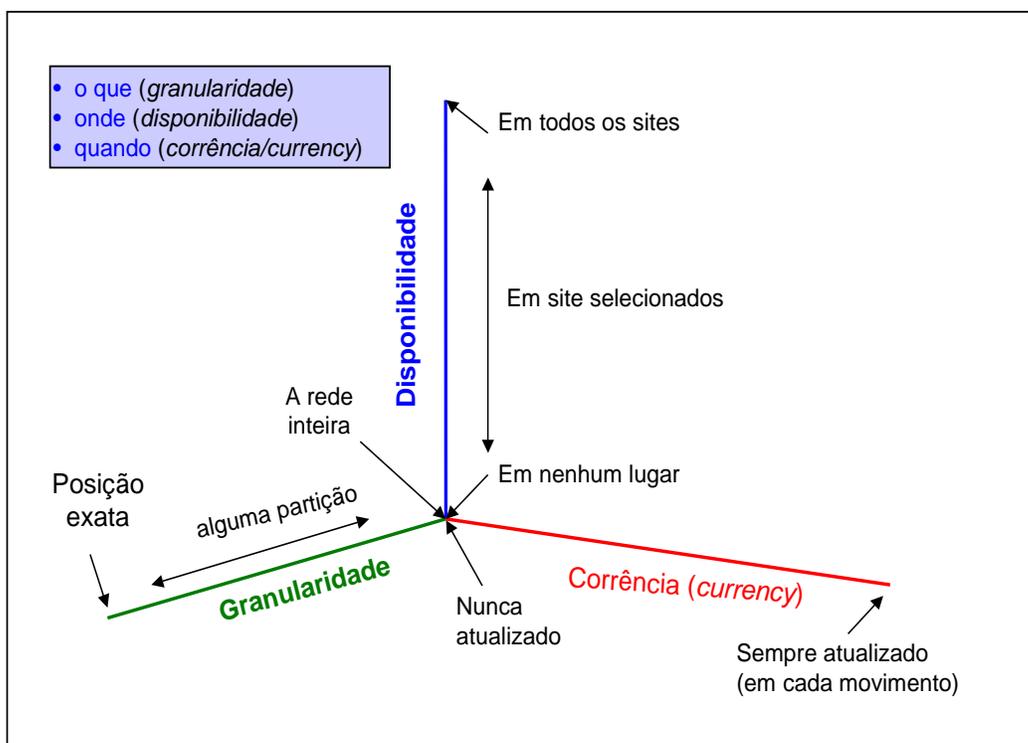


Figura 14: Abordagens para armazenamento de informações da localização dos usuários móveis

Em termos de *disponibilidade*, a escolha varia entre salvar a localização dos usuários móveis em todos os sites da rede até não armazenar em nenhum local. Entre essas duas abordagens extremas, a informação de localização pode ser mantida seletivamente em sites específicos da rede. Existe uma grande quantidade de critérios de seleção para indicar os sites que irão armazenar a informação de localização para cada usuário. Por exemplo, a escolha pode ser salvar a localização dos usuários no site de sua maior frequência de chamadas. A *imprecisão* na informação da posição pode tomar várias formas. Por exemplo, ao invés de manter a exata posição do usuário, uma ampla região ou um conjunto possível de localizações é mantido. Isto se dá em função da

”granularidade¹³” da informação armazenada. A ”*corrência*” (*currency*) se refere a quando a informação da localização armazenada é atualizada. Por exemplo, para usuários com alta mobilidade pode fazer sentido adiar a atualização da informação armazenada sobre a sua localização toda vez que ele se movimentar. Quando a informação precisa e corrente sobre a posição do usuário não estiver disponível, localizar o usuário envolve combinar algum procedimento da busca e um número de consultas que propõem as localizações armazenadas no banco de dados [Pitoura and Samaras, 2001].

5.2. Arquiteturas dos Bancos de Dados de Localização

As duas abordagens mais comuns para arquitetura dos bancos de dados de localização são os **Esquema em Duas Camadas** (*Two-tier schemes*), similar ao modelo utilizado na telefonia celular, no qual a localização corrente para cada usuário em movimento é salvo em duas localizações da rede e o **Esquemas Hierárquico** em que o espaço é hierarquicamente decomposto em sub-regiões. Nesta seção trataremos apenas dessas duas abordagens. Diversas outras abordagens para os bancos de dados de localização foram pesquisadas e desenvolvidas. Maiores detalhes sobre essas abordagens podem ser encontrados em [Pitoura and Samaras, 2001], [Pitoura and Fudos, 1998], [Pitoura and Samaras, 1998], [Cayirci, 2002], [Krishna et al., 1994], [Li et al., 2000], [Li et al., 1999] e [Shivakumar and Widom, 1995].

5.2.1. Esquemas em Duas Camadas

No esquema em duas camadas (*two-tier schemes*), o banco de dados de localização local, chamada de *Home Location Register (HLR)*, é associado com cada usuário móvel. O HLR é localizado em uma localização da rede (zona) pré-especificado para cada usuário. Ele armazena a posição corrente de cada usuário em movimento como parte do perfil do usuário. Os procedimentos de busca e atualização são bastante simples. Para localizar o usuário x , a HLR de x é identificada e a consulta é realizada. Quando o usuário x se movimenta para uma nova zona, a HLR de x é contatada e atualizada para manter a nova localização.

Como aprimoramento deste esquema, um *Visitor Location Register (VLR)* também é mantido em cada zona. O VLR da zona armazena as cópias dos perfis dos usuários que não são da sua área e estão atualmente localizados dentro de sua zona. Quando uma chamada é estabelecida da zona i para o usuário x , a VLR da zona i é primeiramente consultada e somente se o usuário não for encontrado lá, a HLR de x é contatada. Quando x se movimenta da zona i para a zona j , em adição à atualização da HLR de x , o registro de x é excluído da VLR da zona i e um novo registro para x é adicionado na VLR da zona j .

Os dois padrões existentes prevalecem para a tecnologia celular. A Electronics Industry Association Telecommunications Industry Associations (EIA/TIA) Íterim Standard 41 (IS-41), muito comum na América do Norte e a Global System for Mobile Communications (GSM) usada na Europa, ambas utilizam as estratégias de HLR e VLR [Pitoura and Samaras, 2001].

¹³Granularidade se refere ao nível de detalhe em que as informações são mantidas no banco de dados. Quanto maior o nível de detalhes, menor o nível de granularidade.

5.2.2. Esquemas Hierárquico

O *Esquema de Localização Hierárquico* estende o esquema em duas camadas mantendo o banco de dados de localização em uma hierarquia. Nesta hierarquia, um banco de dados de localização contém as informações de localização de nível mais alto para todos os usuários localizados nos níveis inferiores. Normalmente esta hierarquia é uma estrutura em árvore. Neste caso, o banco de dados armazena em uma folha uma única zona (célula) e contém as entradas para todos os usuários registrados nesta zona. Em seus nós internos mantém informações sobre os usuários registrados no conjunto de zonas das sub-árvores. Para cada usuário móvel, estas informações são também um ponteiro para as entradas de nível mais baixo do banco de dados ou a localização atual do usuário. Os bancos de dados são normalmente conectados por conexões da rede de sinalização inteligentes, por exemplo uma *Common Channel Signaling (CCS)*. Por exemplo, em telefonia, o banco de dados pode ser colocado nos *switches* da rede. É freqüente o caso em que a única maneira que duas zonas podem se comunicar seja através da hierarquia. Não existe outra conexão física entre elas. A Figura 15, extraída de [Pitoura and Chrysanthis, 2000], apresenta alguns exemplos de armazenamento e busca nesta hierarquia. A figura *a* é um exemplo de registro de um usuário móvel na hierarquia. A figura *b* apresenta uma simulação de caminhamento na hierarquia para encontrar a localização de um usuário que recebeu uma chamada. A figura *c* apresenta o registro de uma nova localização de um cliente em movimento para fora de sua zona e a figura *d* apresenta a técnica de partição, a qual busca reduzir o custo de busca na hierarquia.

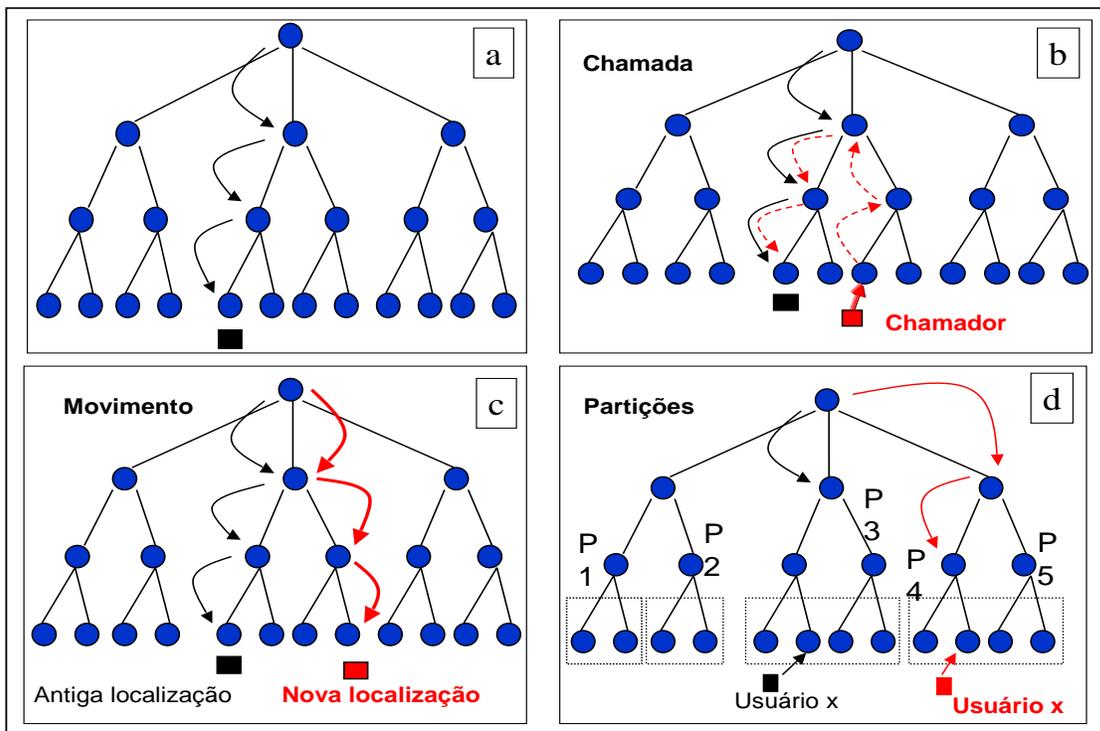


Figura 15: Armazenamento e busca utilizando um banco de dados de localização hierárquico

6. Estágio Atual dos SGBDs Móveis na Indústria e nos Centros de Pesquisas Científicas

A computação móvel é uma área em franca expansão, tanto pelo lado da pesquisa científica como pelo lado da indústria. Cada vez mais novas aplicações necessitam ser apoiadas pela tecnologia sem fio, requerendo o lançamento de novos produtos e motivando a pesquisa, tanto no contexto da comunicação de dados, da engenharia de software como do armazenamento e gerenciamento de dados. Esta seção tem por objetivo apresentar alguns produtos de SGBDs comerciais para a computação móvel, bem como algumas linhas de pesquisas científicas em bancos de dados móveis.

6.1. Estágio atual dos SGBDs Móveis na Indústria

Os maiores fabricantes de SGBDs atuam no mercado da computação móvel, oferecendo um conjunto de produtos que incluem software para o desenvolvimento de aplicações e SGBDs adaptados para este ambiente, os quais buscam uma integração dos dados destas aplicações com os servidores de bancos de dados localizados na rede fixa. Uma característica comum desses produtos está no fato de serem desenvolvidos, principalmente, para atuarem na desconexão dos clientes, como bancos de dados locais, sempre necessitando de uma carga antecipada dos dados (hoarding) para dar suporte as aplicações. Quando conectados a rede fixa, executam a sincronização com seus servidores, evitando trabalharem conectados como se estivessem em um ambiente de banco de dados distribuído.

Segundo uma pesquisa do Gartner Dataquest de 2002, publicada na [Magazine, 2002], a Sybase com o seu produto *Sybase SQL Anywhere* dominava, aproximadamente, 65% do mercado de banco de dados móveis. Outros produtos também relevantes no mercado são o Oracle Lite da Oracle, o DB2 Everyplace da IBM, o SQL Server 2000 CE da Microsoft e o SQLBase da Gupta Technologies. Todos esses produtos são considerados como *Small Footprint DBMS*, ou seja, SGBDs projetados para minimizar a utilização da memória dos hosts móveis. A seguir são apresentados alguns desses produtos, sem o objetivo de comparação entre os mesmos.

6.1.1. Sybase UltraLite

O produto *SQL Anywhere Studio* da Sybase é um pacote de software que provê a gerência de dados e possui uma solução de banco de dados embutida chamada de *Sybase UltraLite*. Além do seu sincronismo com os SGBDs da Sybase, pode se comunicar com produtos de outros fabricantes. Suas principais características são:

- Pode ser utilizado nas plataformas Windows 95/98/Me, NT, 200, XP, Windows CE (Pocket PC/Handheld PC), Palm Computing Platform, Wind River VxWorks;
- Acesso aos dados através de JDBC, SQL embutida e API baseada em C++;
- As funcionalidades da SQL incluem processamento de transações, integridade referencial e operações de junções de várias tabelas;
- Suporta Binary Large Object (BLOB);
- Provê serviço de encriptografia para segurança dos dados;
- O tamanho máximo do banco de dados é de 2Gb;

- O tamanho máximo de cada linha é de 4k;
- O número de linhas de cada tabela depende do tamanho do banco de dados;
- pode ter até 1.000 tabelas por banco de dados;
- Pode ter até 65.535 linhas por tabela;
- Possui suporte para sincronização em redes sem fio.

Maiores detalhes sobre este produto da Sybase podem ser encontrados em [Sybase, 2003]

6.1.2. Oracle Lite Mobile Server

O produto *Oracle Lite Mobile Server* está construído sob o produto *Oracle9i Application Server*. Possui uma estrutura XML e suporta o desenvolvimento de aplicações que podem utilizar voz, troca de mensagens e o acesso sem fio. As aplicações móveis desconectadas podem acessar seus dados localmente. O versão *Oracle9i Lite* possui dois componentes principais:

1. *Oracle9i Lite Mobile Server*, responsável pela gerência dos bancos de dados móveis e pelo sincronismo das aplicações móveis desconectadas com um amplo número de equipamentos móveis;
2. *Oracle9i Lite Mobile Development Kit*, produto para desenvolvimento de aplicações para várias plataformas;

Suas principais características são:

- Pode ser utilizado nas plataformas Palm, CE, Symbian EPOC e Windows 95/98/NT/2000;
- Possui suporte para sincronização em redes sem fio;
- Suporta as funcionalidades da SQL padrão;
- Possui serviços de mensagens em aparelhos de telefone, pagers e computadores portáteis;
- Possui serviços de mensagens Push-based e Pull-based para broadcast;
- Possui serviços de voz;
- Provê serviço de encriptografia para segurança dos dados.

Maiores detalhes sobre este produto da Oracle podem ser encontrados em [Oracle, 2003]

6.1.3. DB2 Everyplace

O DB2 Everyplace é um banco de dados relacional com aproximadamente 150K. Pode ser utilizado como um banco de dados local quando seu host está desconectado ou como um cliente acessando o servidor durante a conexão com a rede fixa. Possui sincronismo bi-direcional com os SGBDs corporativos. Suas principais características são:

- Pode ser utilizado nas plataformas PalmOS, Microsoft Windows CE/Pocket PC, Symbian EPOC, embutido no Linux, QNX Neutrino e Microsoft Win32;
- Possui suporte para sincronização em redes sem fio;

- Pode ser sincronizado como cliente com o IBM DB2 Universal Server nas plataformas Unix, Windows, OS/2, OS/390 e AS/400;
- Suporta as funcionalidades da SQL padrão;
- Possui interface Query-By-Example (QBE) como interface de consulta;
- Não suporta sub-consultas, visões, triggers, stored procedure, tipo de dados LOBs e funções definidas pelo usuário.

Maiores detalhes sobre este produto da IBM podem ser encontrados em [IBM, 2003]

6.1.4. Microsoft SQL Server CE

O Microsoft SQL Server CE é um banco de dados relacional para o desenvolvimento de aplicações para os equipamentos móveis. Pode ser utilizado como um banco de dados local quando seu host está desconectado ou como um cliente acessando o servidor durante a conexão com a rede fixa. Suas principais características são:

- Suporta a sintaxe de consulta da (SQL);
- Fornece uma API para acesso ao SGBD Microsoft SQL Server;
- Possui otimização de consultas e processamento de transações;
- Projetado para ser integrado com a plataforma Microsoft .NET;
- Possui seu tamanho em torno de 1 Mb;
- Provê serviços de encriptografia para segurança dos dados;
- Mantém interoperabilidade com os produtos da IBM, Oracle e Sybase;
- Suporta as plataformas Win32 e Windows CE, Linux, Palm OS, QNX Neutrino e Symbian EPOC.

Maiores detalhes sobre este produto da Microsoft podem ser encontrados em [Microsoft, 2003].

6.2. Estágio Atual dos SGBDs Móveis nos Centros de Pesquisas Científicas

Diversos trabalhos estão sendo propostos em várias áreas da computação móvel. Particularmente na área de banco de dados móveis podemos citar:

Em [Satyanarayanan, 2001] são discutidos os desafios nos sistemas computadorizados para um novo campo de pesquisa chamado de *Pervasive Computing* (computação penetrante, envolvente, presente em todos os lugares), no qual é examinado o relacionamento deste novo campo de pesquisa com seus predecessores, os sistemas distribuídos e de computação móvel. Maiores detalhes sobre este campo de pesquisa podem ser encontrados em <http://www.computer.org/pervasive>.

O sistema Coda, uma referência em pesquisa sobre sistemas móveis, é apresentado em [Satyanarayanan et al., 1990] e [Kistler and Satyanarayanan, 1991]. Diversos tópicos de pesquisa em bancos de dados móveis, tais como operações desconectadas, segurança, performance, replicação e escalabilidade foram e estão sendo desenvolvidos neste projeto. Junto com o projeto Coda é desenvolvido o projeto Odyssey, com ênfase nas aplicações adaptáveis para a computação móvel. Esses dois projetos deram origem ao projeto Aura, cujo objetivo principal é criar um ambiente chamado de *Ubiquitous Computing* (computação que está em todos os lugares), cujo objetivo

é fornecer a cada usuário uma *órbita* invisível de serviços computacionais e de informações que persistem não obstante a posição do usuário. Informações sobre esses projetos podem ser encontradas em <http://www.coda.cs.cmu.edu/>, <http://www-2.cs.cmu.edu/afs/cs/project/coda/Web/coda.html> e <http://www-2.cs.cmu.edu/aura/>

[Vlach, 2001] apresenta o projeto MDBAS que é um protótipo de um sistema de gerência de múltiplos bancos de dados baseado em agentes móveis, integrando um conjunto de bancos de dados autônomos distribuídos sob a rede, possibilitando a criação de esquemas globais de bancos de dados, gerenciando a transparência de execução para os clientes móveis.

Em [Preguica et al., 2000] é apresentado um modelo de transação para o projeto MobiSnap, o qual é um projeto que tem por objetivo dar suporte ao desenvolvimento de aplicações baseadas em um banco de dados relacional para o ambiente móvel.

Em [Peng and Chen, 2003] é proposto um algoritmo de data mining para descoberta de padrões na movimentação dos clientes. O interessante desta pesquisa é que combina técnicas de data mining de reconhecimento de padrões com o objetivo de para melhorar desempenho total de um sistema móvel.

Diversos outros trabalhos de pesquisas científicas foram citados ao longo deste trabalho, destacando o processamento de transações, a consistência e a recuperação dos bancos de dados para um ambiente de computação móvel.

7. Conclusão

O objetivo deste trabalho foi apresentar o papel dos Sistemas de Bancos de Dados (SBDs) no ambiente da computação móvel, destacando os principais problemas deste ambiente e como os SBDs podem contribuir e serem utilizados no sentido de expandir, cada vez mais, a utilização dos sistemas de informações, propiciando que em qualquer lugar e em qualquer hora o usuário móvel possa ter acesso à informação.

O texto é apresentado buscando contribuir, principalmente, no entendimento do ambiente da computação móvel, para os leitores interessados na área de banco de dados, não se aprofundado nas questões das redes de computadores, sua infra-estrutura, protocolos e outros requisitos para esta área tão complexa.

A inserção dos SBDs na computação móvel não é algo trivial. Os pressupostos para garantia da consistência e integridade dos dados, principal função dos SGBDs, não pode ser totalmente aplicado no ambiente móvel, uma vez que suas operações por dependerem de dois ambientes, em muitos momentos desconectados, não permitem que as transações de banco de dados possam ser executadas como nos seus modelos centralizados ou distribuídos. Assim, constantes pesquisas se fazem necessárias no sentido de desenvolver novos modelos de transações que possam garantir as operações móveis, mantendo os requisitos básicos dos SGBDs em relação aos dados que armazenam.

Diversas pesquisas em bancos de dados para computação móvel acontecem em vários centros de pesquisas científicas e são absorvidas pela indústria. Este trabalho também pode servir de um bom referencial bibliográfico para os interessados na área de bancos de dados para computação móvel.

Referências

- Acharya, A. and Badrinath, B. R. (1994). Checkpoint distributed applications on mobile computers. In *3 rd International Conference on Paralell and Distributed Information Systems*, pages 73–80.
- Acharya, S., Alonso, R., Franklin, M., and Zdonik, S. (1995a). Broadcast disks: Data management for asymmetric communication environments. In *ACM SIGMOD*, pages 199–210.
- Acharya, S., Franklin, M., and Zdonik, S. (1995b). Dissemination-based data delivery using broadcast disks. In *IEEE Personal Communication*, pages 50–60.
- Alonso, R., Haber, E. M., and Korth, H. F. (1992). A database interface for mobile computers. In *Proceedings of IEEE GLOBECOM 92 Workshop on Networking of Personal Communications Applications*.
- Badrinath, B. R., Acharya, A., and Imielinski, T. (1993). Impact of mobility on distributed computations. *Operating Systems Review*, 27(2):15–20.
- Barbará, D. (1999). Mobile computing and databases - a survey. *Knowledge and Data Engineering*, 11(1):108–117.
- Brayner, A. and Filho, J. A. M. (2002). Amdb: An approach for sharing mobile databases in dynamically configurable environments. In *Anais XVII Simpósio Brasileiro de Banco de Dados*, pages 12–26.
- Brayner, A. and Monteiro, J. M. (2000). Temporal serialization graph testing: An approach to control concurrency in broadcast environments. In *Anais XVII Simpósio Brasileiro de Banco de Dados*, pages 287–30.
- Buchholz, S., Zeigert, T., Schill, A., and Held, A. (2000). Transaction processing in a mobile computing environment with alternating client hosts. In *RIDE*, pages 1–8.
- Cayirci, E. (2002). User mobility pattern scheme for location update and paging in wireless systems. *IEEE Transactions on Mobile Computing*, 01(3):236–247.
- Chrysanthis, P. K. (1993). Transaction processing in mobile computing environment. In *IEEE Workshop on Advances in Parallel and Distributed Systems*, pages 77–83, Princeton, New Jersey.
- Datta, A., Celik, A., Kim, J. G., VanderMeer, D. E., and Kumar, V. (1997). Adaptive broadcast protocols to support power conservant retrieval by mobile users. In *ICDE*, pages 124–133.
- Dunham, M. H. and Helal, A. (1995). Mobile computing and databases: Anything new? In *SIGMOD Record*, pages 5–9.
- Dunham, M. H., Helal, A., and Balakrishnan, S. (1997). A mobile transaction model that captures both the data and movement behavior. *Mobile Networks and Applications*, 2(2):149–162.
- Elmasri, R. and Navathe, S. B. (2000). *Fundamentals of Database Systems*, volume Third Edition. Addison-Wesley.

- Elmasri, R. and Navathe, S. B. (2002). *Sistemas de Banco de Dados - Fundamentos e Aplicações*, volume Tradução da terceira edição. Livros Técnicos e Científicos Editora - LTC.
- Endler, M. and da Silva e Silva, F. J. (2000). Requisitos e arquiteturas de software para computação móvel. In *I Workshop de Sistemas de Informações Distribuídas de Agentes Móveis*, São Paulo - Brazil. USP.
- Frank, L. (1999). Atomicity implementation in mobile. In *Proceedings of the DEXA Conference*, pages 105–113.
- Gore, M. and Ghosh, R. (2000). Recovery of mobile transaction. In *Proceedings of the DEXA Conference*, pages 23–27.
- Gray, J., Helland, P., O’Neil, P., and Shasha, D. (1996). The dangers of replication and a solution. In *ACM SIGMOD*, pages 173–182, Montreal, Canada.
- Huang, Y., Sistla, P., and Wolfson, O. (1994). Data replication for mobile computers. In *SIGMOD Conference*, pages 13–24.
- Huang, Y. and Wolfson, O. (1994). Object allocation in distributed databases and mobile computers. In *ICDE*, pages 20–29.
- IBM (2003). Db2 everyplace. Web address (<http://www-3.ibm.com/software/data/db2/everyplace/>), IBM.
- Imielinski, T. and Badrinath, B. R. (1992). Mobile wireless computing: Solutions and challenges in data management. Technical report, Rutgers University, U.S.
- Jennings, N. R. and Wooldridge, M. J. (1998). *Agent Technology - Foundations, Applications, and Markets - Applications of Intelligent Agents*, volume part 1, pages 3-28. Springer.
- Kearney, P. (1998). *Agent Technology - Foundations, Applications, and Markets - Applications of Intelligent Agents*, volume part 1, pages 125-136. Springer.
- Kistler, J. J. and Satyanarayanan, M. (1991). Disconnected operation in the coda file system. In *Thirteenth ACM Symposium on Operating Systems Principles*, volume 25, pages 213–225, Asilomar Conference Center, Pacific Grove, U.S. ACM Press.
- Krishna, P., Vaidya, N. H., and Pradhan, D. K. (1994). Location management in distributed mobile environments. In *3rd International Conference on Parallel and Distributed Information Systems (PDIS)*, pages 81–88, Washington, DC. IEEE Computer Society.
- Lam, K.-Y., Kuo, T.-W., Tsang, W.-H., and Law, G. C. K. (1999). Transaction shipping approach for mobile distributed real-time databases. In *Database and Expert Systems Applications*, pages 932–941.
- Lam, K.-Y., Kuo, T.-W., Tsang, W.-H., and Law, G. C. K. (2000). Concurrency control in mobile distributed real-time database systems. *Information Systems*, 25(3):261–286.
- Li, G.-H., Lam, K.-Y., and Kuo, T.-W. (1999). Location update generation in cellular mobile computing systems.
- Li, G.-H., Lam, K.-Y., Kuo, T.-W., and Lo, S.-W. (2000). Location management in cellular mobile computing systems.

- Lu, Q. and Satyanarayanan, M. (1994). Isolation-only transactions for mobile computing. *Operating Systems Review*, 28(2):81–87.
- Lu, Q. and Satyanarayanan, M. (1995). Improving data consistency in mobile computing using isolation-only transactions. In *Fifth IEEE HotOS Topics Workshop*, Orcas Island, WA, USA.
- Magazine, P. (2002). On the road to mobile databases. Web address (<http://www.pcmag.com/article2/0,4149,4180,00.asp>), PC Magazine.
- Massari, A., Weissman, S., and Chrysanthis, P. K. (1996). Supporting mobile database access through query by icons. *Distributed and Parallel Databases*, 4(3):249–269.
- Melo, R. N., Silva, S. D., and Tanaka, A. (1997). *Banco de Dados em Aplicações Cliente-Servidor*. Editora Infobook.
- Microsoft (2003). Db2 everyplace. Web address (<http://www.microsoft.com/sql/ce/default.asp/>), IBM.
- Nwana, H. and Ndumu, D. (1998). *Agent Technology - Foundations, Applications, and Markets - Applications of Intelligent Agents*, volume part 1, pages 29-47. Springer.
- Oracle (2003). Oracle9i application server in mobile. Web address (<http://www.oracle.com/ip/deploy/ias/mobile/index.html/>), Oracle.
- Peng, W.-C. and Chen, M.-S. (2003). Developing data allocation schemes by incremental mining of user moving patterns in a mobile computing system. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):70–85.
- Phatak, S. H. and Badrinath, B. R. (1999). Data partitioning for disconnected client server databases. In *MobiDE*, pages 102–109.
- Pitoura, E. and Bhargava, B. (1994a). Dealing with mobility: Issues and research challenges. Technical report, Purdue University, USA.
- Pitoura, E. and Bhargava, B. (1994b). Revising transaction concepts for mobile computing. In *First IEEE Workshop on Mobile Computing Systems and Applications*, pages 164–168, Santa Cruz, CA, US.
- Pitoura, E. and Bhargava, B. K. (1995). Maintaining consistency of data in mobile distributed environments. In *International Conference on Distributed Computing Systems*, pages 404–413.
- Pitoura, E. and Bhargava, B. K. (1999). Data consistency in intermittently connected distributed systems. *Knowledge and Data Engineering*, 11(6):896–915.
- Pitoura, E. and Chrysanthis, P. K. (2000). Mobile and wireless database access for pervasive computing.
- Pitoura, E. and Fudos, I. (1998). An efficient hierarchical scheme for locating highly mobile users. In *CIKM*, pages 218–225.
- Pitoura, E. and Samaras, G. (1998). *Data Management for Mobile Computing*, volume 10. Kluwer Academic Publishers.
- Pitoura, E. and Samaras, G. (2001). Locating objects in mobile computing. *Knowledge and Data Engineering*, 13(4):571–592.

- Pradhan, D., Krishna, P., and Vaidya, N. (1996). Recovery in mobile wireless environment: Design and trade-off analysis.
- Prakash, R. and Sihghal, M. (1996). Low-cost checkpointing and failere recovery in mobile systems. In *IEEE Transactions on Parallel and Distributed Systems*.
- Preguica, N. M., Baquero, C., Moura, F., Martins, J. L., Oliveira, R., Domingos, H. J. L., Pereira, J. O., and Duarte, S. (2000). Mobile transaction management in mobisnap. In *ADBIS-DASFAA*, pages 379–386.
- Samaras, G. and Pitsillides, A. (1997). Client/intercept: a computational model for wireless environments. In *4 th International Conference on Telecommunications (ICT'97), Melbourne, Australia*.
- Satyanarayanan, M. (1996). Fundamental challenges in mobile computing. In *Symposium on Principles of Distributed Computing*, pages 1–7.
- Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications*, pages 10–17.
- Satyanarayanan, M., Kistler, J. J., Kumar, P., Okasaki, M. E., Siegel, E. H., and Steere, D. C. (1990). Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on Computers*, 39(4):447–459.
- Saygn, Y., Özgür Ulusoy, and Elmagarmid, A. K. (2001). Association rules for supporting hoarding in mobile computing environments.
- Serrano-Alvarado, P., Roncacio, C. L., and Adiba, M. (2001). Mobile transaction supports for dbms: An overview.
- Shivakumar, N. and Widom, J. (1995). User profile replication for faster location lookup in mobile environments. In *Mobile Computing and Networking*, pages 161–169.
- Silberschatz, A., Korth, H. F., and Sudarshan, S. (1997). *Database System Concepts*, volume Third Edition. Mc Graw Hill.
- Ssu, K.-F., Yao, B., Neves, N. F., and Fuchs, W. K. (1998). Adaptive checkpointing with storage management for mobile environments.
- Sybase (2003). Sql anywhere studio8 database. Web address (<http://www.sybase.com/products/anywhere/>), Sybase.
- Villate, Y., Pitoura, E., Illarramendi, A., and Elmagarmid, A. K. (2000). Extending the data services of mobile computers by external data lockers. In *DEXA Workshop*, pages 160–164.
- Vlach, R. (2001). Mobile database procedures in MDBAS. In *DEXA Workshop*, pages 559–563.
- Walborn, G. D. and Chrysanthis, P. K. (1997). PRO-MOTION : Management of mobile transactions. In *Selected Areas in Cryptography*, pages 101–108.
- Wolfson, O., Sistla, A. P., Dao, S., Narayanan, K., and Raj, R. (1995). View maintenance in mobile computing. *SIGMOD Record*, 24(4):22–27.
- Özsu, M. T. and Valduriez, P. (1999). *Principles of Distributed Database Systems*. Prentice Hall.