

Uma Extensão da Plataforma de Distribuição ANSAware para Suportar a Computação Móvel

Adriano Augusto de Souza

Departamento de Informática / Coordenação de Processamento de Dados
Universidade Federal de Pernambuco / Escola Técnica Federal da Paraíba
E-mail: aas@di.ufpe.br / adriano@jpa.etfpb.br

Paulo Roberto Freire Cunha

Departamento de Informática
Universidade Federal de Pernambuco
E-mail: prfc@di.ufpe.br

Resumo

Este artigo propõe extensões à plataforma de distribuição ANSAware para melhorar o desenvolvimento dos sistemas de computação móvel através desta ferramenta. Isto acontece em virtude das plataformas de distribuição atuais não estarem preparadas para funcionar de maneira otimizada em ambientes onde a mobilidade dos *hosts* constitui-se num fator essencial. Logo, após uma introdução sobre o tema de computação móvel, apontamos os pontos problemáticos do funcionamento dos sistemas de computação móvel sobre as plataformas de distribuição. Tais pontos permitiram-nos definir e especificar um modelo que garantisse o suporte adequado da plataforma ANSAware à construção deste tipo de aplicação. Por fim, uma avaliação de desempenho do modelo proposto é apresentada a fim de verificarmos se as extensões propostas alcançaram os resultados desejados.

Abstract

This paper proposes extensions to the ANSAware distributed platform to improve the development of mobile computing systems. This occurs because distributed platforms are not prepared to run in an optimized environment where the host mobility is an essential factor. After an introduction about mobile computing, we present the problem of mobile computing systems running on these platforms. This allows us to define and to specify a model that would guarantee the correct support for the ANSAware platform to create mobile computing applications. Finally, a performance evaluation of the proposed model is realized, in order to verify if the proposed extensions will reach the expected results.

1. Introdução

A fim de prover uma infra-estrutura básica para a construção de sistemas distribuídos, o projeto das plataformas de distribuição propõe diversas soluções para o compartilhamento de recursos, a interoperabilidade, a concorrência, a escalabilidade e a tolerância a falhas. A abordagem adotada por essas plataformas usa a estratégia de transparência a fim de esconder os detalhes de implementação e de desempenho dos usuários das aplicações [Dav96].

Tal tratamento trabalha bem quando o ambiente da rede é estático, pois não sofre muitas mudanças, possibilitando que as informações sejam escondidas dos usuários. Entretanto, quando o ambiente é dinâmico, como é o caso do ambiente de computação móvel, ela se torna pouco produtiva [Kir96]. Neste caso, a informação não pode ser escondida das aplicações e serviços das plataformas, devido ao fato de seu conhecimento possibilitar o desenvolvimento de aplicações mais eficientes que as atuais [Sch95a].

A tendência crescente para o uso das aplicações de computação móvel, tem nos levado a acreditar que a computação móvel deverá tornar-se a forma dominante de computação no próximo século [Duc92]. Nesse contexto, como as aplicações de computação móvel têm se caracterizado por uma natureza eminentemente distribuída [Sou96], pesquisas têm sido realizadas no sentido de definir se será possível a construção dessas aplicações através das plataformas de distribuição atuais. Neste artigo, nos interessamos em explorar a idéia de reestruturação das plataformas de distribuição, como forma de criarmos ferramentas capazes de trabalhar de maneira otimizada no ambiente de computação móvel. Desta forma, procuramos caracterizar o ambiente de computação móvel (seção 2), buscando investigar e classificar, de forma genérica, os problemas decorrentes da mobilidade na construção dos sistemas de computação móvel através dessas ferramentas (seção 3); analisando-os, através de um estudo de caso, sobre a plataforma de distribuição ANSAware (seção 4), e apresentando algumas medidas de desempenho obtidas por simulação (seção 5).

2. Computação Móvel

Um conjunto de fatores tecnológicos tem contribuído para o surgimento de uma nova forma de computação chamada de **Computação Móvel**. A essência da Computação Móvel consiste em se manter a conectividade de usuários com redes de computadores enquanto os mesmos migram entre diferentes localizações, ou seja, estes computadores mantêm sua conectividade e continuam executando suas aplicações independentes de sua localização [Cav96].

Para que possamos utilizar as aplicações de computação móvel, é necessário que tenhamos uma infraestrutura de comunicação móvel adequada. Hoje, pode-se afirmar que tal infra-estrutura é composta basicamente pela rede celular [Roc95], por redes locais sem fio [Roc94], pelas redes baseadas em satélite [Jor94] e pelos sistemas de *paggers* [Wal90]. Além disso, visando a criação de uma RDSI (Rede Digital de Serviços Integrados) em ambientes de redes móveis e a adaptação da Internet ao ambiente de computação móvel, questões como *Wireless ATM (Asynchronous Transfer Mode)* e *mobile TCP/IP (Transmission Control Protocol/Internet Protocol)* vem sendo bastante pesquisadas.

Entretanto, essa rápida expansão da infra-estrutura de comunicação móvel fez surgir novos problemas a serem resolvidos nas técnicas de desenvolvimento de software. Uma questão importante é determinar o papel do *host* móvel num sistema distribuído. Num extremo, os *hosts* móveis estão sendo usados como terminais limitados¹ (*palmtops* e PDAs) e em outro, eles estão sendo usados como estações de trabalho² com capacidade computacional e de memória suficiente para executarem aplicações localmente (portáteis). No primeiro caso, os *hosts* móveis dependerão de alguns *hosts* fixos (não necessariamente os mesmos) para executarem suas aplicações. No segundo, eles serão participantes ativos dos sistemas distribuídos. Entretanto, em qualquer caso, a questão mais relevante é a seguinte: **quais são, se houver alguma, as características que fazem a computação móvel diferente da computação distribuída não móvel [Pit93] ?**

Num sistema distribuído não móvel, um *host* opera em dois modos, ou conectado à rede ou totalmente desconectado dela. Em contraste, num ambiente de computação móvel, podem existir vários graus de desconexão, desde uma **desconexão total** até uma **fraca desconexão** (quando um terminal é conectado ao resto da rede por baixa banda passante). Como resultado, um *host* móvel pode operar em mais de um modo. Adicionalmente, a fim de conservar energia, ele pode operar no estado **conservando energia**³, onde a velocidade do relógio é reduzida e nenhuma aplicação do usuário é executada. O computador móvel ficará aguardando receber qualquer mensagem enviada para ele do resto da rede. Após receber tal mensagem ele retorna ao seu estado normal de execução. A 1ª resume os diferentes modos de operação em que um *host* móvel pode estar [Pit93].

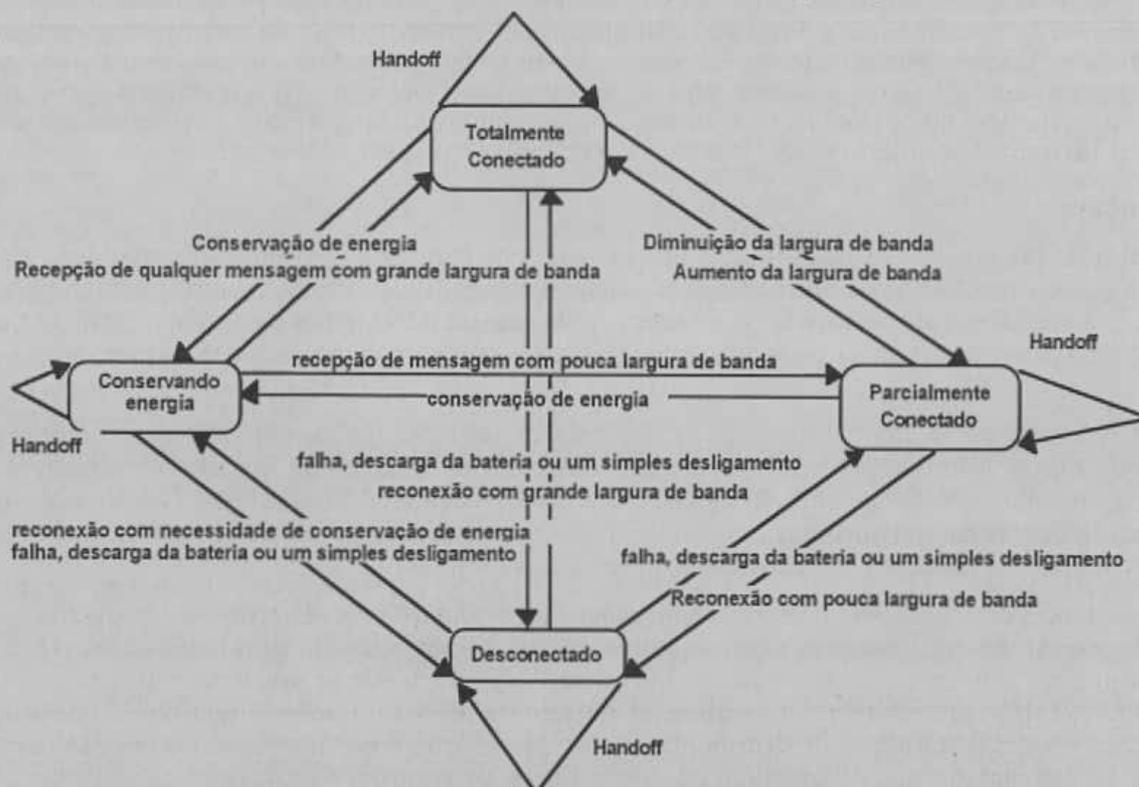


Figura 1 - Modos de Operação de um Host Móvel

Logo, uma questão de alta importância tanto para a comunidade de pesquisa como para a indústria é saber quem são os usuários típicos para os sistemas de computação móvel. Pessoas que sempre viajam com equipamentos transportáveis⁴, tais como empresários e palestrantes constituem os primeiros candidatos. Por outro lado, consumidores podem achar interessante os serviços que usem terminais móveis para entretenimento. O

¹ Que tem falta de uma qualidade característica. No caso acima, sem disco e com capacidade de processamento e de memória limitadas. Também chamados de terminais *dumbs*.

² Também chamadas de *walkstations*.

³ Também conhecido como modo *Doze*, ou seja soneca.

⁴ Como portáteis e agendas eletrônicas.

desafio é descobrir a aplicação “chave” que faça com que os computadores móveis sejam usados em grande escala [Pit93].

Uma das aplicações mais importantes e com grande potencialidade de ser muito usada em situações de mobilidade é a baseada em correio eletrônico, pois permite comunicação interpessoal ou notificação de eventos pré-definidos como um voo cancelado, tráfego congestionado ou liquidações numa certa loja de departamentos. Aplicações simples (baseadas em *paggers*), anotações básicas (agendas), ou entretenimentos (jogos) constituirão os serviços padrões para os sistemas de computação móvel. Informações de páginas amarelas (listas telefônicas ou classificados) também serão largamente utilizadas. Como boa parte dessas aplicações já possuem similares, será necessário diferenciá-las das aplicações já existentes. Um exemplo possível é o caso de uma consulta solicitando qual o hospital mais próximo que trata uma determinada doença.

3. Descrevendo as Mudanças Necessárias

Nesta seção, definiremos de forma genérica as extensões propostas para que as plataformas de distribuição otimizem o suporte à computação móvel. Para isso, consideramos sete aspectos importantes no projeto de uma plataforma de distribuição que permitam atender aos requisitos exigidos para que as aplicações de computação móvel possam ser executadas sobre um ambiente de computação distribuída.

3.1 Ligações dinâmicas

Ligações dinâmicas aos servidores são bastantes úteis para aplicações cliente-servidor em redes cabeadas que realizam tratamento quando ocorrem falhas no servidor. A propriedade fundamental é a de um cliente ser ligado a um serviço em vez de um servidor, permitindo que a infra-estrutura de suporte possa mudar de servidor ou até mesmo empregar múltiplos servidores para proporcionar o mesmo serviço mais eficiente e/ou confiavelmente. Para sistemas de computação móvel onde a localização da informação muda constantemente, ligações dinâmicas para um servidor de informações local faz-se essencial. Pode ser necessário e vantajoso, devido a razões de custos, modificar a ligação cliente-servidor para uma certa aplicação toda vez que um *host* móvel mover-se para uma nova célula [Pop96]. Entretanto, é pouco realista esperar que aplicações de computação móvel mudem as ligações aos servidores após toda movimentação entre as células, como também a preocupação com a mudança da ligação deve ser uma tarefa do sistema de comunicação que deva ser executada automaticamente.

3.2 Protocolo de Transporte Confiável

Muitos clientes móveis desejam acessar serviços em redes cabeadas, fazendo com que a escolha do protocolo de transporte para aplicações distribuídas baseadas em RPC (*Remote Procedure Call*) seja um problema difícil [Jos95]. UDP (*User Datagram Protocol*) é inadequado para ligações sem fio que são bastantes propensas a erros [Cac94]. Já o TCP (*Transmission Control Protocol*), embora proporcione entrega confiável e tenha um bom controle de congestionamento, possui uma enorme perda de desempenho no ambiente de computação móvel por causa de erros e movimentações [Cav96]. Para aplicações num ambiente de computação móvel baseadas na arquitetura das plataformas de distribuição atuais, é desejável um protocolo que proporcione confiabilidade no meio sem fio, mas que ao mesmo tempo tenha a simplicidade do UDP para meios relativamente livres de erro [Cac94].

3.3 Retransmissão de Pedidos

É esperado que a capacidade de vida útil de uma bateria cresça apenas 20% nos próximos dez anos [She92]. Esta é uma taxa bem menor do que as existentes em qualquer outro componente de hardware atual. Daí, surge a necessidade básica de se procurar formas de conservar a energia dos equipamentos portáteis.

Em uma aplicação cliente-servidor normal, as retransmissões de pedidos-resposta perdidas são normalmente coordenadas pelo cliente que usa um *timeout* para reenvio. Como o cliente móvel usa uma ligação com grande probabilidade de erro de transmissão e com pouca capacidade de transmissão, se o reenvio de pedidos for coordenado pelo cliente, então seu resultado poderá ser bastante indesejável devido à pouca largura de banda e ao gasto de energia do *host* móvel, especialmente se o pedido chegar correto à rede cabeada e apenas a resposta for perdida.

3.4 Suporte à Operação de Desconexão

As atuais aplicações baseadas nas arquiteturas das plataformas de distribuição existentes não prevêm a desconexão como uma circunstância normal e muitas destas aplicações encerram-se com uma mensagem de impossibilidade de localização do *host*, no caso de o cliente, o servidor, ou ambos estarem desconectados. Em sistemas de computação móvel entretanto, desconexão e reconexão são processos bastante rotineiros, logo deveriam ser tratados como tal pelas plataformas de distribuição.

Existem duas maneiras de um *host* se desconectar da sua rede [Hil95]. A primeira é através de desconexões súbitas, que ocorre quando o *host* não comunica a saída da área de cobertura da rede ou então simplesmente desliga-se. A outra, é a desconexão planejada, onde o *host* móvel pode informar ao seu roteador da

sua intenção para que alguma ação preventiva possa ser realizada [Gue95]. Tal ação preventiva pode ser a inclusão de uma versão do servidor de nomes no *host* móvel para que este possa continuar a operar mesmo durante o período de desconexão.

3.5 Garantindo a Qualidade do Serviço

A garantia da qualidade do serviço que serve de base ao sistema de comunicação das plataformas de distribuição está diretamente relacionada à liberdade de movimento requerida pelo usuário (2°). Para que este consiga uma maior liberdade de movimentação, ele terá pouca largura de banda disponível. E a medida que o usuário deseje uma maior largura de banda, sua liberdade de movimentação diminuirá.

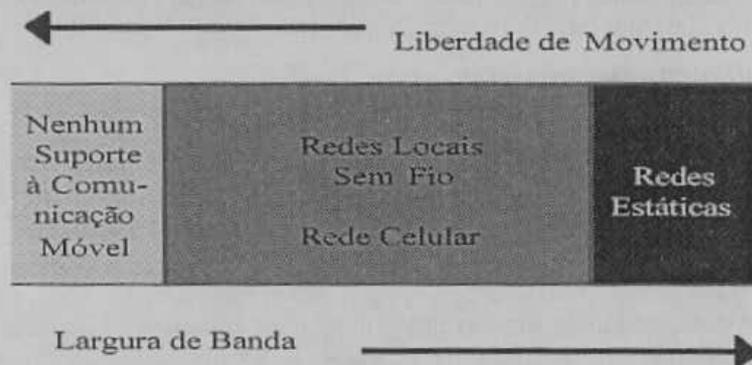


Figura 2 - Relacionamento entre Movimentação e Qualidade de Serviço da Rede

Em um ambiente distribuído estático, os usuários podem contar tipicamente com protocolos de comunicação confiáveis que possuem uma capacidade de transmissão entre 1 Mbps e 5 Gbps. Essa grande largura de banda permite que se projetem aplicações e serviços de sistemas operacionais considerando que a intensidade do tráfego da rede não seja o principal parâmetro. Em redes de longa distância cabeadas, a latência⁵ é o fator dominante na determinação do desempenho da comunicação, ficando a capacidade de transmissão como um parâmetro secundário [Dav96].

Se um portátil move-se de uma conexão de rede fixa para uma conexão de rede móvel, é bastante provável que a qualidade de serviço fornecida seja substancialmente reduzida. Dependendo da tecnologia usada, a capacidade de transmissão disponível cairá para uma taxa entre 1200 bps e 2 Mbps, além da eficiência do canal também ser menor. Por exemplo, o aumento no número de erros de bits e o *handoff*⁶, no caso de sistemas celulares, pode levar a um número relevante de pacotes perdidos.

A informação sobre a qualidade do serviço pode ser usada com grande sucesso pelas plataformas de distribuição. Exemplos como o do projeto MOST (*Mobile Open Systems Technologies*) [Fri96] mostram com sucesso como a qualidade do serviço pode ser usada para que a plataforma ANSAware funcione melhor para suportar sistemas de computação móvel. Esse projeto desenvolveu um protótipo de uma aplicação de computação móvel que possibilita a usuários acessarem um banco de dados remoto. Quando um usuário solicita uma consulta a esse banco de dados, a quantidade de campos necessários ao retorno da consulta é baseada na qualidade de serviço naquele momento. Além disso, a aplicação também fornece informações aos usuários, como, por exemplo, onde os congestionamentos de tráfegos estão ocorrendo, para que estes possam adaptar o seu comportamento de acordo com as características da rede que usam.

3.6 Contabilização dos Custos

O controle de recursos é o mecanismo que realiza as políticas de gerenciamento de recursos. Geralmente, não é permitido ao cliente solicitar indiscriminadamente qualquer recurso. Alguns recursos são caros, muitos são disponíveis em quantidades limitadas. Políticas de gerenciamento de recursos ditam quando e como um cliente pode utilizar um recurso. Contabilidade e controle de serviços estão intimamente relacionados. Não é possível um controle de serviços sem a existência de mecanismos de contabilidade [Sou94].

Em uma plataforma de distribuição, tem sido bastante difícil projetar e implementar esquemas seguros, eficientes e justos para uso dos serviços. Exceções ocorrem em algumas companhias de telecomunicações, que tem atentado a manter uma clara distinção entre aqueles serviços os quais normalmente residem para dentro da rede, onde podem ser cobrados, e aqueles que residem para fora [Dav96].

⁵ Período de inatividade entre um estímulo e a resposta por ele provocada.

⁶ Passagem de controle da ligação de uma ERB (estação rádio base) para outra ERB. É uma característica muito importante das redes celulares.

3.7 Segurança

À medida que o uso das plataformas de distribuição se torna mais disseminado, a necessidade de manter os dados seguros também aumenta. Enquanto os problemas de segurança na transmissão de dados no enlace móvel são solucionados com algoritmos de criptografia [Azi94] e técnicas de espalhamento espectral, o acesso às diversas redes, devido à mobilidade do *host* móvel necessita ser controlado [Sch95b]. Dessa forma, soluções para o gerenciamento global de contas, grupos de trabalho e senhas necessitam ser repensados [Mol94].

4. Definindo o Novo Modelo de Plataforma de Distribuição

Como o objetivo principal desse artigo é criar uma plataforma de distribuição capaz de trabalhar de maneira otimizada no ambiente de computação móvel, devemos apresentar mecanismos que expressem o comportamento adequado destas ferramentas em situações onde a mobilidade física dos elementos seja um fator essencial.

Porém, em virtude da maioria das plataformas de distribuição não possuir uma infra-estrutura básica homogênea para o desenvolvimento de sistemas distribuídos. Pois cada plataforma apresenta diferentes implementações das funções de distribuição das aplicações (notificação de eventos e suporte à transação, à segurança e à nomeação), além de diferentes estratégias de provisão de transparências de distribuição (acesso, localização, migração, replicação, persistência e falhas), a solução foi identificar quais são as mudanças principais necessárias nas plataformas de distribuição (seção 3), possibilitando que sejam definidos os principais itens a serem questionados, além de permitir que se faça o ataque de cada problema através de uma estratégia específica.

Com isso, para fins de estudo de caso das mudanças necessárias, iremos escolher uma das plataformas de distribuição existentes no mercado. Tomaremos como primeiro critério de escolha, a fim de facilitar a identificação e a especificação de soluções necessárias ao suporte da comunicação móvel, uma plataforma que já seja bastante utilizada no Departamento de Informática da Universidade Federal de Pernambuco e que já tenha sido testada e validada no desenvolvimento de aplicações distribuídas. Além disso, tomaremos como segundo critério uma plataforma que siga o RM-ODP (*Reference Model - Open Distributed Processing*), devido a acreditarmos que tal modelo, por ter uma visão orientada a objetos [Far95], servirá de uma base arquitetural para a modelagem e a construção de sistemas distribuídos mais próximos do mundo real. Com tudo isso, baseado nesses dois critérios, ANSAware torna-se a escolhida, porque encontra-se funcionando dentro do Departamento com pesquisas sendo realizadas e em andamento; e, além disso, está em conformidade com o RM-ODP.

4.1 ANSAware

ANSAware é uma plataforma de distribuição desenvolvida a partir das especificações contidas no modelo de engenharia da arquitetura ANSA (*Advanced Network System Architecture*), em conformidade com o RM-ODP. O objetivo desta plataforma é apoiar o desenvolvimento e a execução de aplicações distribuídas [Ros96].

O RM-ODP é uma estrutura arquitetural para integração de serviços de distribuição, interoperabilidade e portabilidade de aplicações distribuídas. Ele tenta englobar a heterogeneidade dos ambientes, a distribuição física e lógica de equipamentos de computação e de aplicações, a interoperabilidade entre sistemas heterogêneos e a portabilidade das aplicações nos diversos ambientes [Len95].

O RM-ODP receita uma metodologia para o projeto de sistemas distribuídos descrevendo diferentes níveis de abstração chamados **pontos de vista**. Um conjunto de conceitos, estruturas e regras é definido em cada ponto de vista, criando uma linguagem para especificação de sistemas ODP em cada nível, tendo como consequência, um melhor entendimento das questões a serem resolvidas para se desenvolver a aplicação distribuída. Os cinco pontos de vista do modelo RM-ODP são empresa, informação, computacional, engenharia e tecnologia. Destes, apenas o computacional e o de engenharia são implementados na plataforma de distribuição ANSAware.

O ponto de vista computacional lida com o particionamento lógico das aplicações distribuídas independente de qualquer ambiente distribuído específico. Ele procura esconder do projetista da aplicação os detalhes da plataforma de distribuição que suportará a aplicação.

O ponto de vista de engenharia preocupa-se com as questões relacionadas à infra-estrutura das aplicações distribuídas. Ele identifica a funcionalidade da plataforma requerida para prover o modelo computacional.

Os serviços computacionais do ANSAware são compilados para objetos de engenharia, os quais são chamados de cápsulas. Estas são unidades de operação autônoma dentro da plataforma. Cada cápsula tem seu próprio espaço de endereçamento e pode executar concorrentemente. Os recursos de um nó⁷ são gerenciados por uma cápsula ANSAware chamada núcleo. Isto leva a um ambiente uniforme de computação distribuída independente de sistemas operacionais, computadores ou redes [Bau94].

⁷ Refere-se a um computador ou estação de trabalho, mas pode também ser uma máquina virtual num sistema UNIX ou uma rede de computadores gerenciada por um sistema operacional distribuído.

O protocolo de comunicação de dados usado pela ANSA e implementado no ANSAware é baseado no RPC (*Remote Procedure Call*). O protocolos suportados atualmente são UNIX named pipes, UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*).

ANSAware assume que interações com fluxo de dados em rajadas serão mais comuns que interações com fluxo de dados contínuos. Para aumentar o desempenho das aplicações, um escalonador e um protocolo de comunicação com três camadas foi desenvolvido.

As três camadas desse protocolo são tratadas abaixo:

- ◆ **MPS (*Message Passing Service*)**: prover uma interface para cada protocolo da camada de transporte usado pelo sistema operacional.
- ◆ **Protocolo de Execução**: Implementa as chamadas de operações ANSA. Dois protocolos são definidos: REX (*Remote Execution Protocol*) para chamadas ponto-a-ponto e GEX (*Group EXecution protocol*) para chamadas de grupo. O REX já foi completamente implementado, mas o GEX existe somente em partes⁸.
- ◆ **Protocolo de Sessão**: Armazena o estado fim-a-fim para chamadas remotas. Este protocolo possibilita que as aplicações distribuídas possam utilizar uma semântica para os problemas ocasionados no servidor de exatamente uma vez, ou seja, na qual todo pedido REX é executado exatamente uma vez, nem mais nem menos.

A fim de ilustrar o funcionamento do protocolo, a forma como a recepção das mensagens é executada no ANSAware é descrito na 3^o.

O escalonador espera recepções na rede (etapa 1). Quando o escalonador é chamado, ele identifica o protocolo de execução pesquisando-o na tabela de protocolos (etapa 2). Ele então informa ao REX (etapa 3), que executará o MPS apropriado (etapa 4). O MPS então receberá o pacote e decodificará a mensagem contida nele (etapa 5). Se uma conexão é solicitada, o REX (etapa 6) estabelecerá um canal (etapa 7) onde será controlado um ponto de sincronização pelo protocolo de sessão (etapa 8).

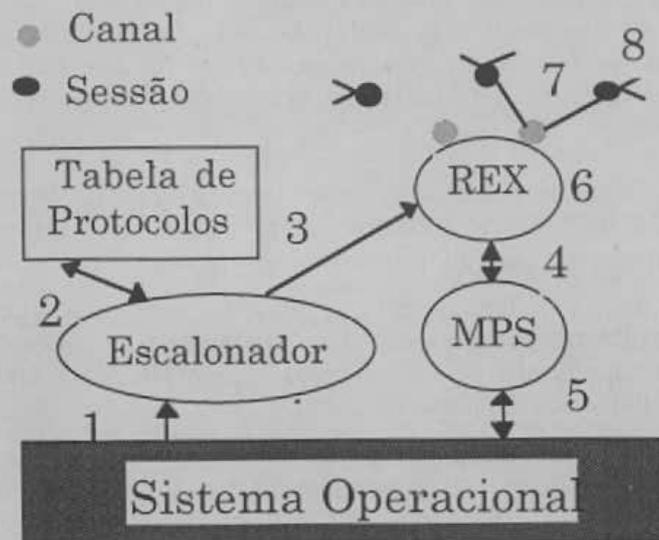


Figura 3 - Recepção de Mensagens no ANSAware

4.2 M-REX

RPC (*Remote Procedure Call*) tem sido muito utilizado como um protocolo básico para estruturação de muitas aplicações distribuídas construídas através de plataformas de distribuição, como de outras formas. Entretanto, implementações de RPC convencionais assumem que todos os *hosts* numa rede são estacionários e que são sempre alcançáveis, exceto em caso de falhas. Para que os sistemas de computação móvel possam ser desenvolvidos em cima dessas implementações, mecanismos que corrijam a perda de desempenho e que suportem as desconexões e os *handoffs* necessitam ser implementados. É com base no M-RPC (*Mobile - Remote Procedure Call*), uma proposta efetuada por Bakre em [Bak95], e no serviço de mediação para clientes móveis num modelo cliente-servidor [Ric96], que desenvolveremos nossas extensões, a qual chamaremos de M-REX (*Mobile - Remote EXecution Protocol*).

M-REX será nossa proposta de mudança para que o serviço REX da plataforma de distribuição ANSAware suporte clientes móveis de maneira otimizada. Nela, um cliente M-REX móvel poderá acessar serviços baseados no servidor REX localizado numa rede cabeada via um agente localizado num MSR (*Mobility*

⁸ Dados válido para a versão 4.1.

Support Router) (4°). Um ponto importante dessa nossa proposta será estender os serviços da camada RPC a fim de tratar os requisitos exigidos pelos sistemas de computação móvel num caminho compatível com o sistema REX existente em ANSAware. Entretanto, como destaque da proposta está a separação entre os diferentes meios de transmissão não ser escondida da camada de transporte, possibilitando assim, a divisão de um enlace, parte cabeado e parte sem fio, em um enlace sem fio e um enlace cabeado independentes. As mudanças propostas serão especificadas nas subseções seguintes.

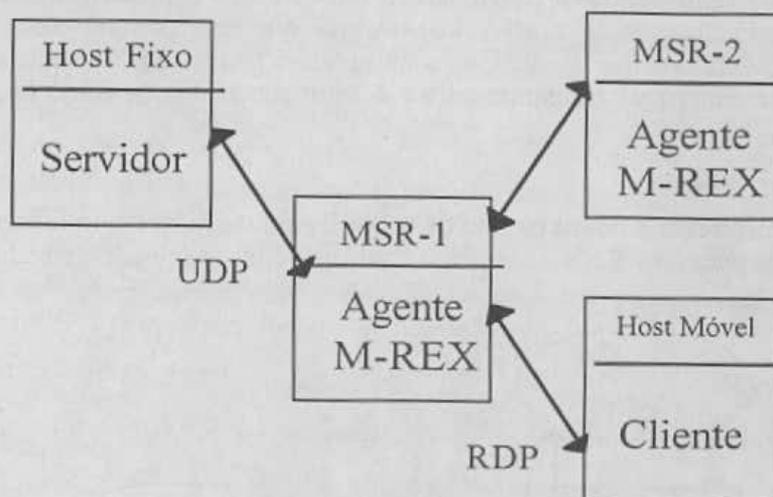


Figura 4 - Visão Geral do M-REX

4.2.1 Ligações dinâmicas

A ligação a um serviço será transparentemente proporcionada pelo M-REX no *host* móvel e pelo agente localizado no MSR. Quando a ligação a um serviço for solicitada por uma aplicação do *host* móvel, o *Trader* do *host* móvel, localizado no MSR é consultado para determinar se o serviço está disponível via o atual MSR. O *Trader* retorna uma instância do serviço para ser usada pelo solicitante da ligação que é guardada no agente M-REX do MSR e no M-REX do cliente. O agente, por sua vez, realiza a ligação entre o cliente e o servidor desejado. Tal ligação poderá ser modificada pelo agente M-REX para outros pedidos da mesma aplicação cliente.

O esquema de ligações dinâmicas funciona bem com servidores *stateless*, pois com servidores *statefull* orientados a conexão se requer a quebra da conexão com o servidor antigo e o estabelecimento de uma outra conexão com o novo servidor, ocasionando a necessidade de manutenção do estado do cliente. As informações mantidas por um servidor sobre o estado das interações em andamento com os clientes são chamadas de informações de estado. Aqueles servidores que não guardam qualquer informação são chamados de servidores *stateless*; os outros são chamados de servidores *statefull*. Logo, como o ANSAware já possui um protocolo de sessão que controla um ponto de sincronização para possíveis falhas na comunicação, deixamos a cargo do programador da aplicação a opção de escolher se o servidor será *stateless* ou não, para que o agente M-REX do MSR possa se adequar ao tipo de operação utilizada.

Outra questão importante, só que dessa vez sobre balanceamento de carga, é que na versão atual do ANSAware (4.1) [APM93], quando um cliente faz um pedido para uma interface, esta pode falhar devido a alguns motivos como interface ilegal, operação ilegal, *timeout* ou migração do servidor. Se uma nova referência da interface for recebida pelo *Trader*, hoje, o procedimento padrão da infra-estrutura do ANSAware quanto a mudança de servidores será retornar uma nova referência de interface para a ligação. Na nossa proposta, o *Trader* do M-REX, não só fará isso, como também, através de uma nova versão do serviço *Relocator*, ficará sempre monitorando se um servidor com perspectiva de melhor eficiência (por exemplo, baseando-se algoritmo de roteamento de menor caminho) para a ligação estará disponível. Pois, caso isso aconteça, o agente M-REX efetuará a troca de servidor. Tal monitoração deverá ocorrer sempre que acontecer um *handoff*. Este é o melhor momento de se realizar a monitoração, pois o agente M-REX aproveita que já ocorre uma perda de dados durante a transmissão, devido à troca de controle da ERB (estação rádio base), para verificar se um servidor mais eficiente para a ligação se disponibilizará.

4.2.2 Protocolo de Transporte Confiável

O M-REX usará um protocolo indireto para a camada de transporte, ou seja, ele não terá o controle fim a fim do caminho entre a origem e o destino, pois este será dividido em duas partes: a primeira, a parte sem fio, entre o *host* móvel (origem) e o agente localizado no MSR, e a segunda, a parte cabeada, entre o agente e o servidor. Tal arranjo (4°) permitirá o uso do protocolo RDP (*Reliable Data Protocol*) [Pat90], que é orientado à conexão e possui uma transmissão e recepção ordenadas de pacotes, na parte sem fio da ligação e do uso do protocolo UDP na parte cabeada da ligação. RDP proporciona proteção contra erros e interrupções na ligação sem fio que não são necessários numa ligação cabeada, que é, pelo menos teoricamente, mais confiável. Ele também assegura a não quebra da ligação da camada de transporte entre o *host* móvel e o agente M-REX do MSR durante

o processo de *handoff*. Se considerarmos o meio de transmissão da parte cabeada da ligação confiável, o uso de TCP na parte cabeada não proporcionará um desempenho melhor do que o UDP. Mas, se mesmo assim nessa parte do enlace for necessário um protocolo orientado a conexão, o uso de TCP seria possível.

Em outras propostas de melhor suportar a comunicação móvel em plataformas de distribuição, como [Sch95a] e [Fri96], é comentado que o protocolo IP deve sofrer mudanças para dar melhor suporte à comunicação móvel, ficando a cargo do IETF (*Internet Engineering Task Force*) escolher a melhor proposta submetida à padronização. Quanto à mudança no IP e a forma pela qual que tem sido estudada, somos de pleno acordo [Rod96]. Entretanto, consideramos que as mudanças não podem se restringir apenas à camada de rede, pois a camada de transporte tem um papel fundamental no desempenho da rede e ela também deve se adaptar à mobilidade.

4.2.3 Retransmissão de Pedidos

A fim de não sobrecarregar a pouca largura de banda do enlace móvel e de conservar energia da bateria do *host* móvel, nossa proposta para o M-REX é colocar o controle de retransmissão de pedidos no agente M-REX do MSR.

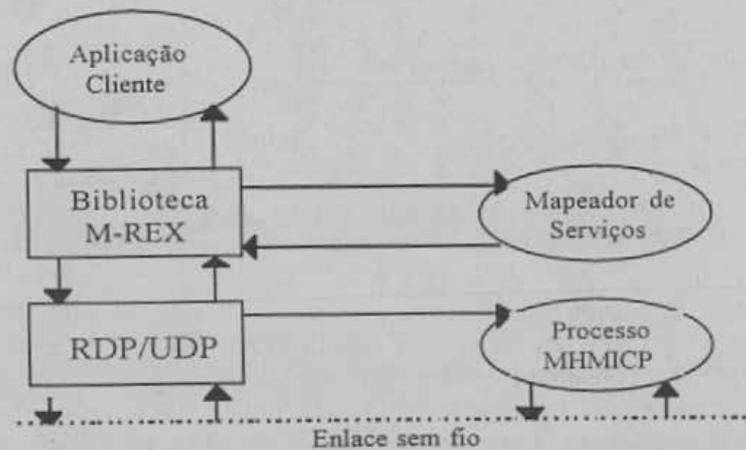


Figura 5 - Stub M-REX do *Host* Móvel

Para suportar este mecanismo, deverá ser montada a seguinte estrutura: os componentes de software do *stub* M-REX do *host* móvel (5º) serão responsáveis pelo processamento do cliente. A biblioteca M-REX fornecerá a API (*Application Programming Interface*) para as funções do M-REX que seja similar a do REX. O processo MHMICP (*Mobile Host Migration Control Process*) deverá controlar o gerenciamento das conexões I-TCP. Os componentes de software do agente M-REX (6º) controlarão o funcionamento do MSR. O processo MSRMICP (*Mobility Support Router Migration Control Process*) será o responsável pelo gerenciamento de todas as conexões I-TCP para todos os *hosts* móveis que estiverem, naquele momento, ligados ao MSR. O agente M-REX fica responsável pela execução de pedidos RPC submetidos pelos *hosts* móveis localizados na sua área de cobertura. Tal pedido é enviado ao agente M-REX pelo *host* móvel, o qual controlará, através das funções do I-TCP, os pedidos de retransmissão⁹, devido a perdas, para o agente; ficando o controle dos pedidos de retransmissão para o servidor¹⁰, também devido a perdas, para o agente M-REX. Isto acontece em virtude de num ambiente móvel, um pedido feito por um cliente móvel para um servidor, cuja arquitetura é apresentada na 7º, usando um protocolo de transporte não orientado a conexão, poderá falhar devido às seguintes razões:

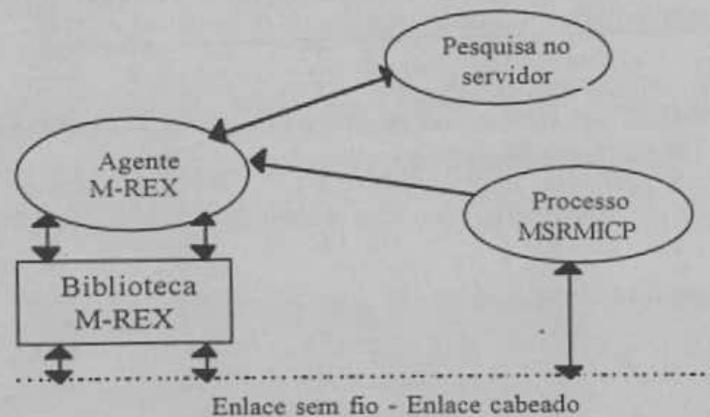


Figura 6 - Agente M-REX no MSR

⁹ Parte sem fio.

¹⁰ Parte cabeada.

- ◆ O pedido é perdido no enlace sem fio;
- ◆ O pedido é perdido no enlace cabeado;
- ◆ O pedido é parado ou atrasado pelo servidor se este estiver ocupado atendendo a outros clientes;
- ◆ A resposta é perdida no enlace cabeado;
- ◆ A resposta é perdida no enlace sem fio.

Usar o REX convencional da plataforma ANSAware faz com que a retransmissão do pedido seja efetuada pelo cliente móvel em todos os casos acima. Por outro lado, usar o M-REX, requer apenas que a retransmissão do pedido do *host* móvel seja efetuada apenas no primeiro caso, ou seja, quando o pedido for perdido no enlace sem fio. Além disso, o agente M-REX poderá guardar no MSR a resposta, caso ela seja perdida na parte sem fio.



Figura 7 - Sistema M-REX do Servidor

Dessa forma, outro fator importante que podemos verificar no caso do servidor da *ligação* dinâmica ser *statefull*, é que parte da necessidade do protocolo de sessão do ANSAware se estabelecer, ocorre de forma diferente da atual (seção 4.1). Se uma conexão for solicitada, para viabilizar uma semântica de **exatamente uma vez**, o M-REX estabelecerá um canal entre a ligação cabeada do MSR e do servidor, onde será controlado um ponto de sincronização pelo protocolo de sessão. Tal canal não precisará envolver a parte sem fio, pois só haverá operações **idempotentes**¹¹ nessa parte da ligação, já que as possíveis atualizações dos dados só ocorrerão no servidor.

Acreditamos também que parâmetros como o número de tentativas de retransmissão e o intervalo entre essas tentativas no enlace sem fio possam causar problemas de congestionamento, acarretando uma queda do desempenho durante a transmissão. Entretanto, não achamos por bem adotar soluções como o protocolo QEX (*Quality-of-service driven Execution protocol*) [Fri96], que analisa as características do meio de comunicação para cada interação e automaticamente ajusta-se para o melhor aproveitamento daquele enlace, porque consideramos que a tarefa do controle de congestionamento já é realizado pelos protocolos do *mobile TCP*, não havendo necessidade de mais um protocolo realizando a mesma tarefa [Awe91].

4.2.4 Suporte à Operação de Desconexão

Para o M-REX, o suporte para a operação de desconexão será proposto em duas formas. Para desconexões rápidas (que ocorrem devido aos *handoffs*, às regiões de sombreamentos e às interferências de campos eletromagnéticos¹²) ficará a cargo da camada de transporte entre o *host* móvel e o MSR a preocupação com a retransmissão dos pacotes. Para desconexões longas (que podem ocorrer quando um *host* móvel move-se para fora da área de cobertura da rede) o M-REX permitirá que caso o *host* móvel deseje desconectar-se após submeter um pedido RPC para o agente M-REX do MSR, o agente poderá recolher e armazenar o resultado do pedido até o *host* móvel reconectar-se novamente.

Entendemos que essa forma de tratar desconexões permitirá que boa parte das aplicações de computação móvel não sejam interrompidas quando ocorrerem desconexões. Mas, para isso, faz-se necessário avaliar qual o tempo ideal para caracterizar uma desconexão rápida, e qual o ideal para uma desconexão longa. Além disso, faz-se necessário também avaliar o desempenho da possibilidade do *Trader* acompanhar seus usuários durante desconexões longas, ou seja, que a estrutura do *Trader* seja replicada no cliente móvel durante desconexões longas, se este assim o desejar.

4.2.5 Garantindo a Qualidade do Serviço

O controle da garantia da qualidade do serviço é aplicável para monitorar o desempenho da informação transmitida entre os componentes de uma aplicação distribuída. Para solucionar esse problema, o qual é uma consequência direta da pouca largura de banda do enlace sem fio, utilizaremos no M-REX a abordagem do projeto MOST (*Mobile Open Systems Technologies*) [Fri96]. Esta proposta implementa com bastante eficiência esse

¹¹ As operações que podem ser repetidas sem nenhum problema são ditas idempotentes.

¹² Devido aos serviços de radiodifusão já existentes.

controle, possibilitando que os usuários adequem o desempenho da rede às suas reais necessidades, principalmente antes de enviar qualquer informação.

A grande mudança realizada pelo projeto MOST para que a plataforma de distribuição ANSAware melhor suportasse a comunicação móvel foi o fato de estender o suporte para as ligações *stream*. Um ótimo resultado desta extensão é que a plataforma ANSAware ficou em perfeita sintonia com o padrão RM-ODP atual no que se refere às ligações. Pois, a fim de encontrar o requisito para uma abstração do fluxo de dados de tempo real, foi adicionado à plataforma o conceito de ligações *stream* explícitas. Estas ligações possibilitam uma abstração fim a fim sobre comunicação de mídia contínua e suportam conexões *m:n*, ou seja, *m* nós fontes sendo conectados a *n* nós destinos [Fri96].

As ligações *stream* serão estabelecidas usando uma operação explícita de ligação, a qual toma como parâmetros as interfaces fonte e destino e um conjunto dos parâmetros de qualidade do serviço desejados que podem ser uma especificação da vazão e da latência desejadas pela ligação. Aos serviços clientes, é retornada uma interface de controle da ligação, que inclui as operações SetQoS() e GetQoS() (ver 8º), a fim de controlar a qualidade do serviço do fluxo. Estas operações tomam como argumentos um conjunto de parâmetros de qualidade de serviço que podem ser passados pela ligação *stream* ao protocolo de transporte básico. Um mecanismo de *callback* é também proporcionado com a finalidade de informar aos serviços clientes das degradações da qualidade do serviço relatadas pelo serviço de transporte.

Foi também adicionado uma nova classe de ligação explícita para uso com interfaces operacionais. Estas ligações são estabelecidas usando a operação binder\$Bind (ver 10º), mas toma como argumentos interfaces operacionais. A interface de controle da ligação resultante é idêntica àquela usada para ligações *stream*, exceto que é permitido aos clientes especificar e monitorar um conjunto diferente de parâmetros de qualidade de serviço associados com a ligação. Isto habilita, por exemplo, um cliente a pedir para ser informado quando o nível de qualidade do serviço cai a um determinado patamar. De importância particular aos sistemas de computação móvel está a capacidade para monitorar a possibilidade de envio e recebimento de mensagens numa ligação sem ter que explicitamente enviar mensagens de teste a nível de aplicação. As aplicações podem delegar responsabilidades para garantir a qualidade do serviço estipulada para o sistema. Isto é importante, pois possibilita que os sistemas de computação móvel sejam estruturados numa maneira baseada em eventos. Por exemplo, através do uso dessas ligações é possível afirmar que a ausência de mensagens numa dada interface é resultado de nenhum tráfego ocorrendo nela ao invés de falha na comunicação.

```

BindingControl : INTERFACE =
BEGIN
    Callback : TYPE = {
        QoSViolationCallback,
        ClientMemberPolicy,
        ServerMemberPolicy};

    Unbind :
    OPERATION [ ]
    RETURNS [ ];
    SetQoS :
    OPERATION [ NewQoS : QoS ]
    RETURNS [ Status, QoS ];
    GetQoS :
    OPERATION [ ]
    RETURNS [ QoS];
    Register :
    OPERATION [ Callback_Type :
                Callback;
                Callback_Interface :
                Interface ]
    RETURNS [ Status ];
    DeRegister :
    OPERATION [ Callback_Type :
                Callback;
                Callback_Interface :
                Interface ]
    RETURNS [ Status ];
END.

```

Figura 8 - Especificação IDL para Interface de Controle da Ligação

Por fim, a 9º exemplifica uma interface *stream*, a 10º mostra como criar uma ligação explícita nas extensões do ANSAware propostas pelo projeto MOST e adotada pelo M-REX, enquanto a 8º descreve a especificação IDL criada pelo MOST e também adotada pelo M-REX para interface de controle da ligação [Dav94].

```
Telephone: STREAM_INTERFACE =
BEGIN
FLOW_SPEC 1 AUDIO IN      64, 20, 5, 10
FLOW_SPEC 2 AUDIO OUT     64, 20, 5, 10
END.
```

Figura 9 - Um exemplo de Interface *Stream*

```
! {src} <- traderRef$Import("Telephone","context","properties")
! {sink} <- traderRef$Import("Telephone","context","properties")
! {binding_control_ir} <- binder$Bind(Stream,src,sink,QoS)
```

Figura 10 - Criando uma ligação explícita

4.2.6 Contabilização dos Custos

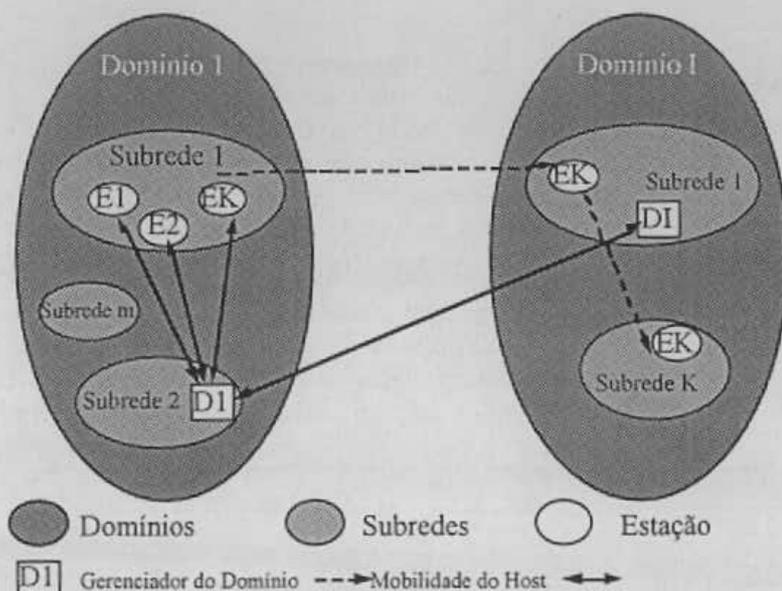
A plataforma ANSAware não apresenta nenhuma política de controle de recursos. O simples fato de definir extensões para suportar a mobilidade não obriga a definição de uma política para controle de recursos. Entretanto, caso esta política não seja definida, os usuários móveis poderão se deslocar para ambientes fora de sua área de atuação e utilizar os recursos desse novo ambiente para onde se deslocaram geralmente num mesmo nível que os usuários originários desse novo ambiente.

Assim, fica como sugestão para trabalhos futuros que o M-REX adote a seguinte política: a princípio, cada usuário recebe (periodicamente) uma quantia em alguma moeda¹³, representando a quantia disponível de um certo recurso. Esta moeda é utilizada para pagar os serviços consumidos. Esses serviços seriam cobrados de acordo com um plano de tarifação discutido previamente. Portanto, a qualquer momento, a diferença entre a soma dada ao usuário e a quantidade disponível representará a quantidade consumida.

4.2.7 Segurança

Devido ao *host* móvel poder se locomover por diversos lugares, usando servidores diferentes, seu acesso a essas redes necessita ser controlado. Para solucionar esse problema, utilizaremos a abordagem em que o ambiente distribuído será dividido em unidades lógicas diferentes chamadas **domínios**. Um exemplo é demonstrado na 11°. Cada domínio pode conter um número diferente de subredes que incluam um determinado número de *hosts* fixos ou móveis. Tal domínio pode também corresponder a um determinado departamento em uma empresa ou instituição. Entretanto, é importante perceber que um domínio antes de tudo será uma construção lógica em vez de uma construção física. Para isso, cada domínio é controlado por um gerenciador de domínio. Este gerenciador de domínio controla uma base de dados para o serviço de registro e de autenticação. O serviço de registro monitorará o registro e a saída do usuário de dentro do domínio. Enquanto o serviço de autenticação implementará a política de acesso aos recursos dos ambientes móveis.

Da mesma forma que para a questão de contabilização de custos, a plataforma ANSAware não implementa nenhuma política de controle de segurança. Logo, para que o M-REX possa funcionar num ambiente seguro, é necessário que um serviço de segurança, como por exemplo a nossa proposta do parágrafo anterior, seja estendido à plataforma de distribuição ANSAware.



¹³ Moeda nesse contexto está sendo colocada com a idéia de criação de uma moeda global e não uma moeda de uma determinada área.

Figura 11 - Estrutura de Domínios para Controle de Segurança

4.3 Algumas Considerações

Neste artigo, descrevemos o modelo M-REX proposto como suporte aos sistemas de computação móvel. As principais características dos ambientes de computação móvel puderam ser adaptadas à plataforma ANSAware através do M-REX, pois tal modelo visa a mudança do ANSAware para que este possa melhor suportar a computação móvel. Para planejar e viabilizar nossa proposta, foram efetuadas as seguintes considerações:

- ◆ A plataforma de distribuição analisada se utiliza do modelo cliente-servidor para a construção de sistemas distribuídos. Tal modelo consiste de um conjunto de processos servidores, que agem como prestadores de serviços, e de um conjunto de processos clientes, que agem como consumidores dos serviços.
- ◆ As aplicações alvos de estudo não foram as que prevêm a mobilidade dos componentes de softwares para onde o host móvel deseja se locomover, mas sim, aquelas que se utilizam da replicação de serviços. Como exemplo, temos a possibilidade do serviço de catálogo telefônico de uma determinada região metropolitana estar replicado em vários locais dessa mesma região. A réplica que estivesse mais próxima do host móvel em um determinado instante, seria a acessada pelo usuário cliente.
- ◆ Os processos servidores e o agente do M-REX só serão executados em *hosts* fixos, enquanto que os processos clientes em *hosts* móveis. Caso os processos servidores possam ser executados em *hosts* móveis, será necessário reavaliar algumas questões definidas aqui, como por exemplo o ponto de sincronização do protocolo de sessão do ANSAware e a garantia da semântica exatamente uma vez.
- ◆ Os protocolos *mobile* IP e *mobile* TCP já foram totalmente especificados, se encontrando disponíveis e em pleno uso. Questões como o *overhead* gerado, a quantidade de ERBs por MSR (*Mobility Support Router*) e os problemas de controle de congestionamento já foram resolvidos.
- ◆ Outras extensões para melhor suportar a computação móvel em plataformas de distribuição levando em consideração as características não tratadas neste trabalho, como o modelo baseado em objetos e pesquisas que levam em conta a mobilidade do software podem ser encontradas em [Kir95] [Bau95]. Entretanto, devido à complexidade que elas trariam ao modelo proposto, pois este foi baseado na plataforma de distribuição ANSAware, que por sua vez se utiliza do modelo cliente-servidor e não do modelo baseado em objetos para a construção de sistemas distribuídos, consideramos que elas seriam mais adequadas para análise em trabalhos futuros.

5. Avaliação de Desempenho

A realização da avaliação de desempenho do M-REX será efetuada através de um experimento de simulação. O objetivo real de avaliar o M-REX é verificar se ele cumprirá de fato as expectativas quando estiver em funcionamento no mundo real. Para isso, teremos que utilizar uma ferramenta que permita criar e executar cenários baseado no modelo proposto da seção 4. Tal ferramenta será o **BONeS Designer** [Alt96].

A fim de proporcionar auxílio na criação da avaliação de desempenho e de garantir a qualidade de uma futura implementação do M-REX, foi realizado uma especificação formal do mecanismo proposto. Tal especificação foi realizada utilizando-se a metodologia OMT (*Object Modelling Technique*) [Pro95] e a técnica de descrição formal LOTOS (*Language of Temporal Ordering Specification*) [Que94], podendo ser encontrada em [Sou96].

Como parâmetros a serem medidos, optamos por avaliar o tempo de transferência de um pedido-resposta de uma RPC em função da variação da quantidade de canais disponíveis das ERBs e da quantidade máxima de conexões simultâneas dos *hosts* fixos (servidores). Desde que tivéssemos pelo menos um protótipo à disposição para realização de medidas, outros parâmetros tais como o tempo de estabelecimento de conexão em função de uma possível disputa pelo canal da ERB, taxa de erros que poderiam ser simuladas por meio de estimativas de probabilidade e disponibilidade de serviços também poderiam ser considerados.

O tempo médio de transferência dentro da camada de enlace será diretamente proporcional à distância entre o cliente (*host* móvel) e o servidor (*host* fixo) e à quantidade de *handoffs*. A capacidade de transmissão do canal a ser considerada na parte sem fio será 4800 bps enquanto que na parte cabeada será 2 Mbps. Os pacotes terão tamanho constante, havendo paralelismo entre eles. Assim, mede-se o atraso médio de cada pacote como resultado de uma variação da quantidade de canais disponíveis em cada ERB e do limite de conexões que cada servidor possa ter.

Parâmetro	Valor
◆ Quantidade de ERBs	36
◆ Quantidade de Servidores	4
◆ Distância entre as ERBs	5 Km
◆ Distância entre os Servidores	15 Km

◆ Quantidade de Canais da ERB	(5,10,15,20,25,30,35,40,45,50) ¹⁴
◆ Limite de Conexões Simultâneas por Servidor	20
◆ Taxa entre Chegada dos Clientes	0,5 s
◆ Capacidade do Canal Sem Fio	4800 bps
◆ Capacidade do Canal Cabeado	2 Mbps
◆ Tamanho do Pacote de Dados	10 KB
◆ Velocidade Mínima do Cliente	0 Km/h
◆ Velocidade Máxima do Cliente	140 Km/h
◆ Tempo de Checagem do Cliente para Ajuste de Seus Atributos, tais como Localização, ERB, Canal, Servidor e Conexão.	$((\text{Distância entre as ERBs} / \text{Velocidade Máxima do Cliente}) / 10)$

Tabela 1 - Parâmetros e Valores Utilizados no Estudo de Caso

O bloco da análise de replicações independentes é utilizado para encerrar a simulação e estimar o intervalo de confiança, com isso temos a propriedade desejável de que as amostras são independentes. Consideramos também que o esquema de alocação de canais das ERBs será estático, não havendo nenhuma priorização aos *handoffs* na questão de distribuição dos canais. Entretanto, isto não significa que em outras simulações não pudéssemos utilizar o esquema dinâmico ou híbrido [Kat96], como também alguma técnica de priorização aos *handoffs* na designação dos canais [Tek92], pois a ferramenta utilizada para simular, o BONEs Designer, tem como característica mais forte a possibilidade de reutilização dos blocos. Outro item importante e que não foi levado em conta, é a possibilidade de existência de distorções no sinal transmitido provocadas por ruído ou eco [Soa95], pois mesmo que tais fenômenos causem prejuízo à qualidade do sinal, já existem processos para contorná-los que já são embutidos na soma do tempo de transferência do sinal.

Tempo de Transferência

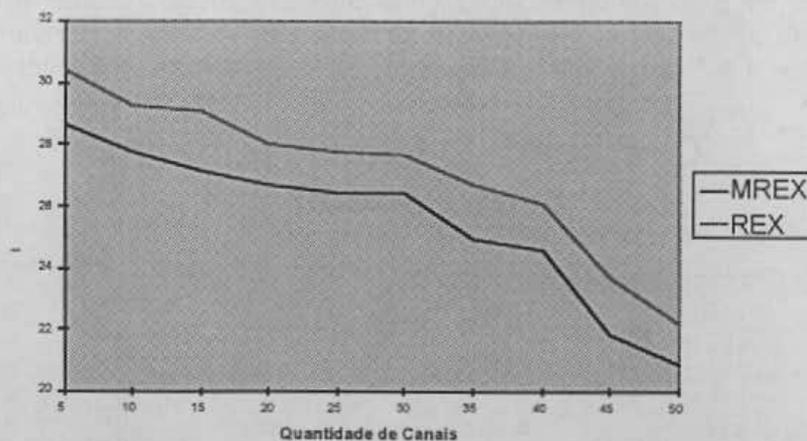


Figura 12 - Tempo de Médio de Transferência das RPCs

A fim de ilustrar o funcionamento do M-REX, já que o programa é inteiramente baseado nele, um estudo de caso foi simulado conforme a 1ª, apresentando os resultados exibidos nos gráficos das 12ª e da 13ª. O estudo de caso completo dessa avaliação de desempenho pode também ser encontrado em [Sou96].

¹⁴ Essa sequência indica que a simulação foi executada variando também a quantidade de canais das estações bases. Na primeira vez todas tinham uma capacidade de 5 canais, na segunda 10, na terceira 15 ... e assim por diante.

Quantidade de Chamadas Atendidas

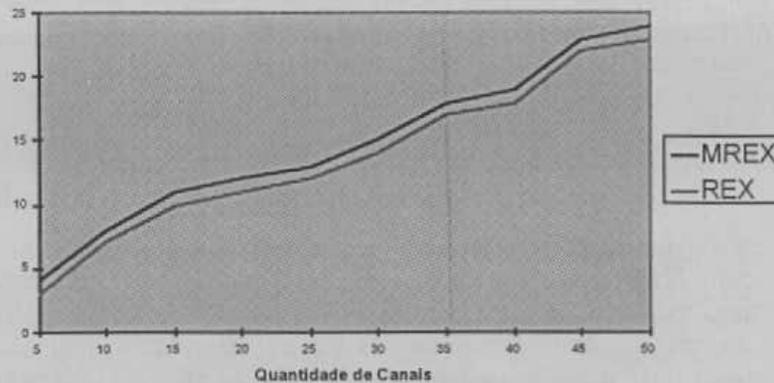


Figura 13 - Quantidade de RPCs Bem Sucedidas

Com estes gráficos foi possível verificamos que de fato a troca para um servidor mais próximo proporcionada pelo M-REX impõe um melhor desempenho à plataforma de distribuição ANSAware no suporte às aplicações de computação móvel, do que o REX. É verdade que o *overhead* gerado (14^o) para realização da pesquisa e troca de servidor produz um custo razoável. Contudo, consideramos que a troca de servidor baseada no algoritmo de menor caminho, utilizando-se como medida a distância geográfica, será sempre vantajosa na situação em que todas as ERBs são homogêneas, ou seja possuem o mesmo alcance e o mesmo número máximo de canais. Por outro lado, em situações onde as ERBs fossem heterogêneas, o uso de medidas baseadas na carga da rede (por exemplo, o menor atraso) poderá ser mais vantajoso. No entanto, deixaremos o estudo desse tipo de cenário, como também a avaliação de outras medidas, como as da seção 4, para trabalhos futuros.

Overhead do Processamento de Handoff

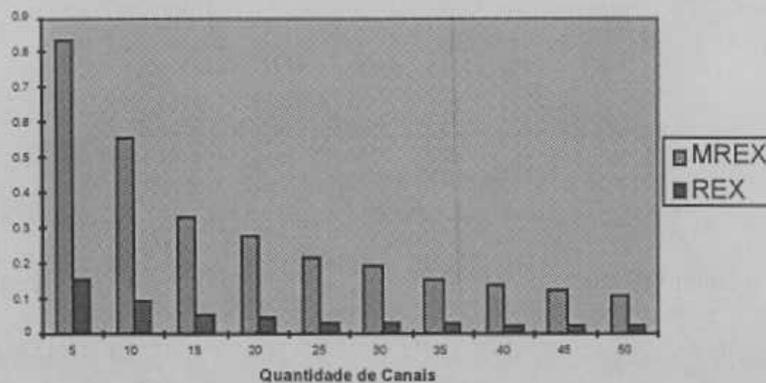


Figura 14 - Overhead de Processamento Durante Handoffs Devido à Troca de Servidor

6. Conclusões e Trabalhos Futuros

Este artigo apresentou extensões que permitem à plataforma de distribuição ANSAware tratar melhor a computação móvel. O mecanismo RPC existente tem sido muito utilizado como um protocolo básico das plataformas de distribuição, assumindo que todos os *hosts* numa rede são estacionários e que são sempre alcançáveis, exceto em caso de falhas. Este artigo possibilitou demonstrarmos que será necessário estabelecer mecanismos que corrijam a perda de desempenho e que suportem as desconexões e os *handoffs* para que os sistemas de computação móvel possam ser desenvolvidos em cima das atuais plataformas de distribuição.

As ligações dinâmicas aos servidores, por exemplo, são muito úteis para aplicações cliente-servidor em redes móveis, pois permitem que se apliquem algoritmos de balanceamento de carga como também um melhor suporte à operação de desconexão quando ocorrem *handoffs* no cliente móvel. Além disso, um controle de retransmissão de pedidos por parte de um agente poderia ser implementado, fazendo com que o cliente móvel economizasse energia.

A utilização do algoritmo de menor caminho para troca de servidores permitiu que questões ligadas à implementação do mecanismo REX, da plataforma ANSAware, pudessem ser tratadas de forma simples. Com isso, através de uma avaliação de desempenho, pode-se verificar que nosso modelo, M-REX, obterá um certo ganho no desempenho da plataforma ANSAware em ambientes de computação móvel.

Um primeiro trabalho futuro consiste na implementação das extensões propostas. Assim, poderemos avaliar de forma concreta como se comportam todas as extensões descritas, além de poder validar as já simuladas. Uma sugestão é que, numa primeira fase, as extensões fossem implementadas para os tópicos de ligações dinâmicas, protocolo de transporte confiável, retransmissão de pedidos, suporte à operação de desconexão e garantia da qualidade do serviço, pois isto já permite uma avaliação geral das propostas e não muda drasticamente a filosofia da plataforma de distribuição ANSAware. Só depois da consolidação dos resultados da primeira fase, é que deveríamos iniciar a segunda etapa, que visaria expandir o ANSAware para os tópicos relacionados à segurança e à contabilização dos custos.

Por fim, pesquisas devem ser motivadas no sentido de verificar o comportamento das extensões diante de algumas outras mudanças, como a possibilidade de mobilidade dos componentes de *softwares* para onde o *host* móvel desejar se locomover, e também, a possibilidade de utilização de uma plataforma que se utilize do modelo de objetos, como CORBA (*Common Object Request Broker Architecture*), para construção de sistemas distribuídos, em vez de um modelo cliente-servidor, como ANSAware. Com essa mudança de direcionamento, pode-se pensar também em criarmos extensões para a IDL (*Interface Definition Language*) e/ou para a DPL (*Distributed Programming Language*) de ANSAware para melhor tratar a programação das aplicações de computação móvel.

7. Bibliografia

- [Alt96] Alta Group of Cadence Design Systems. BONEs Designer Release 3.6. Modelling Reference Guide, March 1996.
- [APM93] ANSAware 4.1 Application Programming ANSAware. Architecture Projects Management Limited, February 1993.
- [Awe91] Awerbuch, Baruch and Peleg, David. Concurrent Online Tracking of Mobile Users. ACM, 1991.
- [Azi94] Aziz, Ashar and Diffie, Whitfield. Privacy and Authentication for Wireless Local Area Networks. IEEE Personal Communications, First Quarter 1994.
- [Bak95] Bakre, Ajay and Badrinath, B. R. M-RPC: A Remote Procedure Call Service for Mobile Clients. Rutgers WINLAB TR-98, June 1995
- [Bau94] Baumgarth, Markus. Management of QoS Attributes with ANSAware in the Multimedia Systems PRISM. ENST, 1994.
- [Bau95] Baumgarten, Uwe and Maier, Joachim. Using Mobile Objects with a PDA. Munich University of Technology, February 1995.
- [Cac94] Cáceres, Ramón and Iftode, Liviu. The Effects of Mobility on Reliable Transport Protocols. Proc. Of the 14th ICDCS, June 1994.
- [Cav96] Cavalcanti, Johnny Wilson Araújo. Desafios e Propostas de Suporte à Computação Móvel em Ambientes TCP. UFPE, Agosto 1996.
- [Dav94] Davies, N., Blair, Gordon S., Chervest, Keith and Friday, Adrian. Supporting Adaptive Services in a Heterogeneous Mobile Environment. Proc. of 1st Workshop on Mobile Computing Systems and Applications, December 1994.
- [Dav96] Davies, Nigel. The Impact of Mobility on Distributed Systems Platforms. Proc. Of 1st International Conference on Distributed Platforms, February 1996.
- [Duc92] Duchamp, Dan. Issues in Wireless Mobile Computing. Columbia University, 1992.
- [Far95] Farooqui, Kazi; Logrippo Luigi and de Meer, Jan. The ISO Reference Model for Open Distributed Processing: an introduction. Computer Networks and ISDN Systems 27 (1995) 1215-1229.
- [Fri96] Friday, A. J., Blair, G. S., Chervest, K. W. J. and Davies, N. Extensions to ANSAware for Advanced Mobile Applications. Proc. of 1st International Conference on Distributed Platforms, February 1996.
- [Gue95] Guedes, Vitor and Moura, Francisco. Replica Control in MIO-NFS. Proceedings of ECOOP'95 Workshop on Mobility and Replication, 1995.
- [Hil95] Hild, Stefan G. Disconnected Operation for Wireless Nodes Position Statement. IBM UK Laboratories, June 1995.
- [Jor94] Jordan, Larry. Comunicações e Redes com o PC. Axel Books, 1994.
- [Jos95] Joseph, Anthony D.; deLepinasse, Alan F.; Tauber, Joshua A.; Gifford, David K. and Kaashoek, M. Frans. Rover: A Toolkit for Mobile Information Access. Proceedings of the Fifteenth Symposium on Operating Systems Principle, December 1995.
- [Kat96] Katzela, I. and Naghshineh, M. Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey. IEEE Personal Communications, June 1996.
- [Kir95] Kirste, Thomas. An Infrastructure for Mobile Information Systems Based on a Fragmented Object Model. Computer Graphics Center Wilhelminenstr, January 1995.
- [Kir96] Kirste, Thomas; Heuer, Andreas; Schumman, Heidrun; Kehrer, Bernd and Urban, Bodo. Concepts for Mobile Information Visualization: The MOVI-Project. Darmstadt Technical University, 1996.
- [Len95] Lento, Luiz Otávio Botelho e Madeira, Edmundo Roberto Mauro. Um Esquema para Acessar Objetos em Ambientes Distribuídos. 13º Simpósio Brasileiro de Redes de Computadores, págs 301-317, Maio 1995.
- [Mol94] Molva, Refik; Samfat, Didier and Tsudik, Gene. Authentication of Mobile Users. IEEE Network, March/April 1994.
- [Pat90] Partridge, C. Request for Comments: 1151. BBN Communications Corp., April 1990

- [Pit93] Pitoura, Evaggelia and Bhargava, Bharat. Consistency and Transaction Management in a Mobile Computing Environment. Purdue University, 1993.
- [Pop96] Pope, S. Application Migration for Mobile Computers. Third International Workshop on Services in Distributed and Networked Environments (SDNE), 1996.
- [Pro95] Protosoft Inc. Paradigm Plus v3.0b, Reference Manual. Release 3.0 for UNIX and Windows Workstations, 1995.
- [Que94] Queiroz, José Antônio Monteiro de e Cunha, Paulo Roberto Freire. Sistemas Distribuídos: De Especificações LOTOS a Implementações. IX Escola de Computação, Julho 1994.
- [Ric96] Richter, K.; Rudolf, St. and Irmscher, K. Mediator Services for Mobile Clients. Proceedings of International Conference on Distributed Platforms, March 1996.
- [Roc94] Rocha, Helder e Martins, Paulo. Redes Sem Fio: Alternativas e Tendências. Departamento de Sistemas de Computação, UFPB, Outubro 1994.
- [Roc95] Rochol, Juergen; Barcelos, Marcelo Boeira e Pufal, Henrique. Comunicação de Dados em Redes Celulares de Telefonia Móvel. XIII Simpósio Brasileiro de Redes de Computadores, Maio 1995.
- [Rod96] Rodrigues, Roberto W. S., Souza, Adriano A. de e Monteiro, José A. S. Simulando um Protocolo da Camada de Rede de Suporte a Computação Móvel. XIV SBRC, Maio 1996.
- [Ros96] Rosa, Nelson S. e Cunha, Paulo R. F. Implementando especificações LOTOS na plataforma de distribuição ANSAware. 14º Simpósio Brasileiro de Redes de Computadores, págs: 239 a 252, Maio 1996
- [Sch95a] Schill, A., Bellmann, B., Böhmak, W. and Kummel, S. System Support for Mobile Distributed Applications. IEEE SDNE Workshop, 1995.
- [Sch95b] Schill, A. and Kummel S. Design and Implementation of a Support Platform for Distributed Mobile Computing, Dresden University of Technology, Faculty of Computer Science, 1995.
- [She92] Sheng, Samuel; Chandrasekaran, Ananth and Broderon, R. W. A Portable Multimedia Terminal for Personal Communications, IEEE Communications Magazine, December 1992, pp 64-75.
- [Soa95] Soares, Luiz Fernando G. Redes de Computadores: das LANs, MANs e WANs às redes ATM. Campus, 1995.
- [Sou94] Souza, Adriano A. e Batista, Daniela C. F. Aspectos de Nomeação e Segurança em Sistemas Distribuídos. Departamento de Informática UFPE, Outubro 1994.
- [Sou96] Souza, Adriano Augusto de. Uma Extensão da Plataforma de Distribuição ANSAware para Suportar a Computação Móvel. Dissertação de Mestrado, Universidade Federal de Pernambuco, Dezembro 1996.
- [Tek92] Tekinay, Sirin and Jabbari, Bijan. A Measurement-Based Priorization Scheme for Handovers in Mobile Cellular Networks. IEEE Journal on Selected Areas in Communications, October 1992.
- [Wal90] Walker, John. Mobile Information Systems. Artech House Inc, 1990.