



PUC

ISSN 0103-9741

Monografias em Ciência da Computação
nº 08/05

ORB para Dispositivos Móveis em Redes sem Fio

Renato Figueiró Maia
Vagner José do Sacramento Rodrigues
Markus Endler

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900
RIO DE JANEIRO - BRASIL

ORB para Dispositivos Móveis em Redes sem Fio

Renato Figueiró Maia, Vagner J. do Sacramento Rodrigues, Markus Endler

{maia,vagner,endler}@inf.puc-rio.br

Abstract. This technical report describes and discusses the recent efforts towards standards and implementations of Object Request Brokers for wireless networks. In particular, it describes the Wireless CORBA standard, an implementation of this standard called MIwCO, as well as other CORBA middlewares to mobile computing.

Keywords: Middleware, CORBA, Wireless CORBA, MIwCO, Wireless Networks.

Resumo. Esta monografia descreve e discute os recentes esforços de padronização e implementação de Object Request Brokers para redes móveis sem fio. Em particular, são descritos o padrão Wireless CORBA, uma implementação deste padrão denominada MIwCO, bem como outros middlewares CORBA para computação móvel.

Palavras-chave: Middleware, CORBA, Wireless CORBA, MIwCO, Redes Wireless.

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3114-1516 Fax: +55 21 3114-1530
E-mail: bib-di@inf.puc-rio.br
Web site: <http://bib-di.inf.puc-rio.br/techreports/>

Sumário

1	Introdução	1
1.1	Ambientes móveis	1
1.2	O padrão CORBA	2
1.2.1	Componentes da arquitetura CORBA	3
1.2.2	Interoperabilidade	3
2	ARQUITETURA CORBA EM REDES SEM FIO	5
2.1	PROBLEMAS DO PADRÃO CORBA PARA AMBIENTES MÓVEIS.	5
2.2	Wireless CORBA	6
2.2.1	Histórico	6
2.2.2	Visão Geral da Especificação	7
2.2.3	Domínios do Wireless CORBA	7
2.2.4	Mobile IOR	8
2.2.5	Tunelamento e Retransmissão de Mensagens na Rede Wireless	10
2.2.5.1	Visão Geral do GTP	11
2.2.5.2	Níveis de Adaptação do GTP	11
2.2.5.3	Estrutura das Mensagens GTP	12
2.2.6	Procedimentos de Handoff	13
2.2.6.1	Participantes do Handoff	14
2.2.6.2	Handoff Inicializado pela Rede	14
2.2.6.3	Handoff Inicializado pelo Terminal	16
2.2.6.4	Access Recovery (Reconstituição do Acesso)	17
2.2.7	Processamento de Mensagens GTP	18
3	MIDDLEWARES CORBA PARA AMBIENTES MÓVEIS.	19
3.1	MIWCO	19
3.1.1	Modelo da Implementação	20
3.1.1.1	Mudanças no Núcleo do MICO	21
3.1.2	Interfaces Bridge CORBA	22
3.2	E*ORB	24
3.2.1	Incorporação dinâmica de protocolos	24
3.2.2	Arquitetura micro-kernel	24
3.2.3	Alocação de memória	25
3.3	Universally Interoperable Core (UIC)	25
4	CONCLUSÃO	26
	Referências Bibliográficas	27

1 Introdução

1.1 Ambientes móveis

A grande difusão das redes wireless é reconhecida como um dos grandes avanços para a indústria que pode, através desse ambiente, melhorar a sua capacidade de produção explorando as características de mobilidade dos usuários e, além disso, diminuir os seus gastos com a instalação da infra-estrutura da rede. No entanto, ao contrário das aplicações nas redes cabeadas, em um ambiente sem fio as aplicações precisam lidar com os problemas da comunicação intermitente e outras questões ligadas a mobilidade do usuário. Dessa forma, fica evidente que a maioria das aplicações existentes na rede fixa não podem ser, simplesmente, portadas para rede sem fio sem considerar as adversidades presentes nesse meio.

Uma rede sem fio é uma tecnologia que permite com que dois ou mais computadores se comuniquem sem a necessidade de alguma espécie de cabeamento entre eles. Uma rede sem fio pode apresentar duas topologias principais. A primeira consiste simplesmente de diversos dispositivos equipados com uma placa de interface com um meio sem fio, que se comunicam diretamente ou por intermédio dos demais dispositivos. Estas redes são chamadas de redes ad hoc e são mais adequadas para redes com poucos computadores e com uma área geográfica reduzida. Outra forma é utilizar um equipamento chamado ponto de acesso (estação base) através do qual os dispositivos móveis podem se conectar. Um ponto de acesso inclui uma interface para uma rede cabeada através da qual este pode se comunicar com outros pontos de acesso. Desta forma é possível difundir o sinal da rede sem fio por áreas maiores e permitir que os dispositivos móveis possam se comunicar com redes cabeadas, como a Internet.

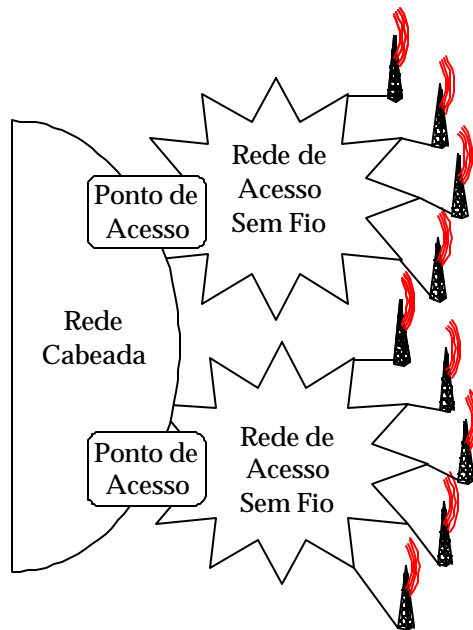


Figura 1 – Modelo de referência de redes sem fio

Utilizaremos neste trabalho, como modelo de referência de redes sem fio, a estrutura de redes composta de pontos de acesso, conforme é ilustrado na **Error! Reference source not found.** Este modelo foi escolhido por ser mais simplificado.

1.2 O padrão CORBA

O desenvolvimento da tecnologia de redes cabeadas criou a possibilidade de se construir aplicações através da composição de conjuntos de componentes distribuídos que executam separadamente através de uma rede. Tais sistemas distribuídos oferecem muitas vantagens em potencial sobre aplicações monolíticas, como escalabilidade, flexibilidade e reusabilidade. Entretanto, é necessário um esquema adequado de comunicação de componentes distribuídos.

CORBA [OMG, 2002] (*Common Object Request Broker Architecture*) é uma especificação proposta pela OMG (*Object Management Group*) de uma infraestrutura para comunicação de objetos distribuídos num ambiente heterogêneo, de forma transparente à aplicação. Esta infraestrutura permite que objetos clientes possam solicitar serviços de outros objetos (chamados objetos servidores) por meio de uma interface bem definida. A Figura 2 mostra os principais componentes da arquitetura CORBA e suas interconexões.

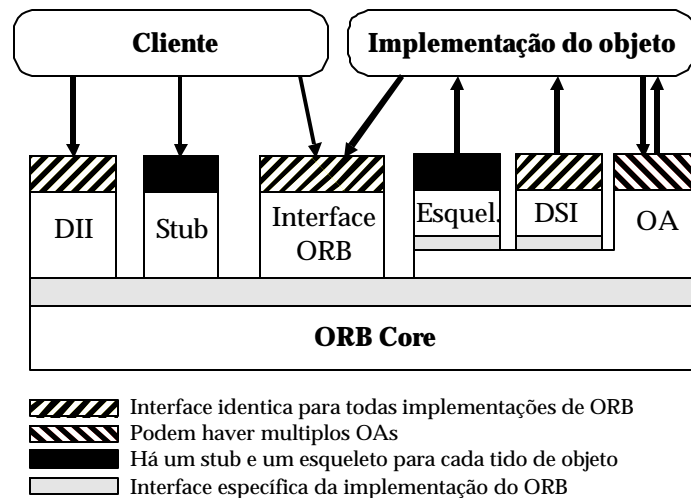


Figura 2 - Estrutura de interfaces de um ORB.

O componente principal da arquitetura CORBA é o ORB (*Object Request Broker*), que encapsula toda a infraestrutura necessária para identificar e localizar objetos, manipular conexões e a entrega de dados. Através do ORB os objetos clientes podem fazer chamadas de métodos de objetos servidores remotos. O ORB é responsável por empacotar os parâmetros da chamada, enviá-los ao objeto servidor e devolver o valor de retorno e os parâmetros de saída ao objeto cliente.

A partir da definição da interface de um objeto servidor, descrita em IDL (*Interface Definition Language*) são gerados dois componentes que conectam os objetos distribuídos ao ORB, são eles o esqueleto do servidor e o *stub* do cliente. O esqueleto do servidor é a interface através da qual as implementações dos objetos recebem as

chamadas remotas, e o *stub* do cliente é a interface através da qual os objetos fazem chamadas a objetos servidores remotos. Para cada tipo de objeto são gerados um *stub* e um esqueleto diferentes.

Também é possível que os objetos façam chamadas através da interface DII (*Dynamic Invocation Interface*), assim como os objetos podem receber chamadas através de um esqueleto dinâmico. Estas interfaces permitem manipular as chamadas remotas de forma dinâmica, como será descrito posteriormente.

1.2.1 Componentes da arquitetura CORBA

A comunicação entre a implementação do objeto e o ORB é feita pelo OA (*Object Adapter*). Ele manipula serviços como a geração e interpretação de referências de objetos, chamadas de métodos, segurança das interações, ativação e desativação de objetos e implementações, mapeamento de referências às implementações de objetos correspondentes e o registro de implementações.

Quando uma chamada é feita, o objeto é ativado pelo OA. Para fazer isso o OA precisa ter acesso a informações sobre a localização do objeto e o ambiente operacional. Esta informação fica armazenada no Repositório de Implementação, que é um componente padrão da arquitetura CORBA.

As interfaces dos objetos podem ser especificadas de duas formas: em IDL ou elas podem ser adicionadas ao Repositório de Interfaces, outro componente da arquitetura. A interface DII permite que clientes especifiquem chamadas a objetos cuja definição e interface são desconhecidas em tempo de compilação. Para utilizar a DII, o cliente tem que construir uma chamada no formato exigido pelo padrão CORBA, incluindo a referência de objeto, a operação e a lista de parâmetros. Estas especificações de objetos e os serviços que eles fornecem são obtidas do Repositório de Interfaces.

O lado do servidor análogo ao DII é o DSI (*Dynamic Skeleton Interface*), com o uso desta interface, a operação não é mais disparada através do esqueleto gerado a partir da especificação da interface em IDL, que é específico para as operações desta interface. Ao invés disso, a operação é disparada através de uma interface que fornece acesso ao nome da operação e seus parâmetros (como na DII especificada acima, a informação pode ser obtida a partir do Repositório de Interface). Portanto, DSI é uma maneira de entregar chamadas de métodos provenientes do ORB a uma implementação de objeto que não se conhece em tempo de compilação.

1.2.2 Interoperabilidade

Há muitas implementações de ORB disponíveis atualmente no mercado. Esta diversidade é problemática, pois permite aos desenvolvedores gerar produtos especificamente para as necessidades do seu ambiente operacional, e cria a necessidade de ORBs diferentes interoperarem. Além do mais, há sistemas distribuídos que não são compatíveis com a arquitetura CORBA, portanto há a crescente necessidade de fornecer mecanismos de interoperabilidade entres estes

sistemas e CORBA. Para tanto, a OMG definiu a arquitetura de interoperabilidade de CORBA.

Diferenças de implementação não são as únicas barreiras que separam objetos, outras razões são fortalecimento de políticas de segurança ou ambientes de teste protegidos para produtos em desenvolvimento. Para fornecer um ambiente completamente interoperável, todas estas diferenças tem que ser levadas em consideração. Por isso que CORBA 2.0 introduz o conceito de domínio, que a grosso modo define um conjunto de objetos que, por alguma razão, seja pela implementação ou por questões administrativas, são separados dos demais objetos. Portanto, objetos de domínios diferentes precisam de mecanismos de interligação para que possam interagir.

As abordagens para interoperabilidade podem ser basicamente divididas em mecanismos de interligações imediatas e mediadas. Com ligações mediadas, as mensagens de um domínio são convertidas da estrutura específica daquele domínio, em outra estrutura mutuamente conhecida entre domínios diferentes. Esta estrutura comum pode ser tanto uma padronizada (especificada pela OMG, por exemplo o IIOP), ou uma estrutura particular conhecida entre as duas partes. Com interligações imediatas, as mensagens são transformadas diretamente de uma forma interna de um domínio para a forma da outra. A segunda solução é potencialmente mais rápida, mas é menos geral, e portanto é necessário poder utilizar ambas.

Para permitir a interoperabilidade entre ORBs é necessário definir um modelo de referência de objetos que possa identificar objetos em diversos ORBs diferentes. Isto é feito através do IOR (*Inter-ORB Object Reference*), que fornece uma estrutura de dados flexível para armazenar as informações necessárias de uma referência de objetos, como a máquina onde o objeto se encontra, o OA responsável e o identificador do objeto no OA.

Além de um modelo de referência de objetos é necessário especificar algum tipo de sintaxe de transmissão de mensagens padrão. Isto é feito pelo protocolo GIOP (*General Inter-ORB Protocol*) definido pela OMG. Ele foi especificamente definido para suprir as necessidades de interação entre os ORBs e foi projetado para trabalhar sobre qualquer protocolo de transporte que se adeque a um conjunto mínimo de características. Obviamente, versões do GIOP implementadas utilizando diferentes protocolos de transporte não serão necessariamente diretamente compatíveis, entretanto sua interação será feita de maneira muito mais eficiente.

Além de definir uma sintaxe de transmissão de mensagens geral, a OMG também especifica um mapeamento obrigatório do protocolo GIOP utilizando o protocolo de transporte TCP/IP, chamado de IIOP (*Internet Inter-ORB Protocol*). A intenção é que o IIOP funcione como uma língua comum e necessariamente compreensível por todas implementações de ORBs. A especificação também permite um conjunto de protocolos específicos para um determinado ambiente, chamados de ESIOPs (*Environment Specific Inter-ORB Protocol*), que servem como mecanismo de interoperabilidade quando existem muitas implementações utilizando um determinado protocolo de transporte.

2 ARQUITETURA CORBA EM REDES SEM FIO

Com o surgimento de tecnologias de redes sem fio, como o GSM, telefonia via satélite, e LANs 802.11 sem fio, se tornou evidente a importância de que CORBA possa também ser estendido para acomodar redes que contêm enlaces sem fio. Primeiramente, o crescimento da classe de dispositivos móveis que utilizam comunicação sem fio faz com que CORBA ou algo similar se mostre necessário, pois dispositivos móveis geralmente tem menos recursos locais do que é possível em dispositivos estacionários, conseqüentemente a habilidade de utilizar serviços de objetos remotos se torna ainda mais importante em dispositivos móveis. Além disso, como os dispositivos móveis se tornam cada vez mais poderosos e sofisticados, espera-se obter acesso a serviços que estão disponíveis através de dispositivos estacionários, através de dispositivos móveis.

Entretanto, a especificação atual de CORBA não se adequa bem ao acesso sem fio. Pelo fato de até agora CORBA ter sido desenvolvido especificamente para redes cabeadas, certas considerações têm sido feitas e que não são válidas para redes sem fio, o que cria dificuldades para aplicações que utilizam acesso sem fio.

2.1 PROBLEMAS DO PADRÃO CORBA PARA AMBIENTES MÓVEIS.

CORBA fornece um framework para construção de sistemas distribuídos através da definição de padrões em três níveis. No nível arquitetural, CORBA define a noção de objeto distribuído e o Object Request Broker, que interage entre eles, e define como os objetos distribuídos se encontram e interagem. No nível de aplicação, CORBA define um conjunto de APIs padrão. No nível de protocolo, CORBA define um pequeno conjunto de protocolos inter-ORB, que permitem com que objetos que utilizem ORBs implementados independentemente se comuniquem.

A arquitetura geral de CORBA mostra-se suficientemente abstrata e flexível para acomodar as diferenças das redes sem fio sem alteração. Entretanto, o nível de protocolo e de aplicação exigem alterações ou extensões.

Apesar de CORBA permitir utilizar protocolos proprietários para comunicação entre objetos de um mesmo tipo de ORB, assim como a definição protocolos especializados para comunicação entre ORBs diferentes que operem em circunstâncias especiais (ESIOPs), o padrão CORBA exige a implementação do protocolo IIOP, que trabalha sobre TCP/IP, como forma de interoperabilidade básica com objetos de outros ORBs. Entretanto, o uso do IIOP é problemático com o acesso sem fio.

Diversos estudos [AMIR, 1995], [Hartenstein, 2001] e [Fladenmuller e Silva, 1999] mostraram que a suíte de protocolos TCP/IP apresenta uma performance pobre sob as circunstâncias adversas presentes na redes sem fio, como latências maiores, banda passante reduzida e variável, perdas e erros de transmissão, etc. O próprio IIOP foi construído assumindo-se que conexões entre objetos são mantidas continuamente. Idealmente, seria necessário que fosse possível encapsular a comunicação dentro de uma abstração que possa manipular desconexões e reconexões automaticamente.

Contudo o IIOP foi desenvolvido de forma que nenhuma informação de estado é preservada entre um objeto cliente e um objeto servidor através de conexões TCP/IP. Se o servidor fica ciente da perda de uma conexão, é esperado que ele cancele quaisquer operações pendentes para aquela conexão cliente.

A interface de invocação dinâmica de CORBA (DII) fornece uma forma de chamada de métodos que permite que o objeto cliente possa fazer requisições com uma chamada não-bloqueante e então recuperar o resultado posteriormente. Entretanto, o resultado de uma chamada assíncrona é armazenado no lado do cliente. A especificação atual do IIOP não define nenhum mecanismo que permita armazenar respostas no servidor até que os clientes estejam hábeis a recebê-las.

Inclusive os identificadores de requisição do IIOP, que são utilizado pelo ORB para identificar as respostas de suas respectivas requisições, só são únicos dentro do contexto e tempo de vida de uma única conexão. Consequentemente, mesmo que um cliente desconectado possa se reconectar antes que o servidor cancele a sua requisição pendente, o IIOP não fornece nenhum mecanismo que permita com que respostas às requisições baseadas em conexões antigas sejam feitas através de uma nova conexão.

Aplicações móveis estão sujeitas às várias alterações no ambiente de rede que inexistem em redes fixas, como a variação de latência, largura de banda e frequência de desconexão. Portanto, é necessário que se possa criar mecanismos que permitam manipular tais circunstâncias de forma flexível e eficiente. Para tanto se faz necessário que o padrão CORBA forneça mecanismos que permitam que aplicações possam determinar as condições de transmissão de rede e talvez modificá-las. Assim como poder definir dinamicamente estratégias para lidar com tais condições de rede que seriam impostas pelo ORB para as aplicações que não fossem “network aware”. Como exemplos de tais métodos e políticas temos:

- Reconexão transparente após desconexões;
- Estratégias para tentativa de conexão (número de tentativas e intervalo entre elas);
- Reserva de largura de banda.

2.2 Wireless CORBA

Nesta seção será apresentada a Wireless CORBA, uma solução padronizada para uso da arquitetura CORBA em terminais móveis que são conectados através de links wireless à rede fixa.

2.2.1 Histórico

Ciente dos problemas relacionados com a mobilidade e comunicações wireless, a DTF (Telecommunications Domain Task Force) do OMG publicou um RFI (Request For Information) em Junho de 1998 para adquirir informações sobre os problemas a serem solucionados e analisar como estes foram resolvidos em outras tecnologias.

Em Maio de 1999, a DTF publicou uma Request for Proposals (RFP) para o acesso sem fio e mobilidade de terminais na arquitetura CORBA [OMGb, 1999], objetivando uma solução padronizada para comunicações móveis e sem fio para o framework CORBA.

Em meados de 2000, o OMG recebeu duas submissões iniciais para a RFP (Request for Proposals), uma da Nokia e Vertel, e outra da Interprise e Highlander. Estes se juntaram posteriormente para preparar uma única submissão, sendo esta submetida ao OMG em fevereiro de 2001. Esta submissão revisada foi adotada pelo OMG como - Wireless Access and Terminal Mobility in CORBA Specification [OMGc, 2001].

Em seguida foi feita uma validação do modelo proposto a partir dos comentários dos implementadores que faziam parte da Wireless CORBA Finalization Task Force (FTF). Várias modificações foram então realizadas na especificação adotada pelo OMG inicialmente, tendo como base os relatos da equipe que integrava o FTF.

2.2.2 Visão Geral da Especificação

Nesta seção serão descritas algumas das partes essenciais da especificação. Durante a descrição são citadas às considerações pertinentes a implementação do MIwCO. Os principais conceitos que são discutidos nesta seção e que definem a arquitetura CORBA móvel são: A Mobile IOR, Home Location Agent, Access Bridge, Terminal Bridge, e GIOP Tunneling Protocol.

2.2.3 Domínios do Wireless CORBA

Na Figura 3 é mostrada uma visão geral da arquitetura Wireless CORBA e seus componentes. O sistema é dividido em três domínios separados, cada qual com o seu componente central.

O home domain para um dado terminal é o domínio que hospeda o seu Home Location Agent. Um visited domain é o domínio que hospeda um ou mais Access Bridges, provendo acesso aos ORBs dos terminais móveis que encontram sob a sua responsabilidade. O terminal domain consiste de um terminal móvel que tem um ORB e uma Terminal Bridge que permite que os objetos em um terminal se comuniquem com objetos em outras redes.

O Home Location Agent em um home domain é responsável por manter a localização de cada terminal pertencente ao seu domínio. Alguns terminais móveis não têm um home domain, e conseqüentemente, não tem Home Location Agent, e portanto são chamados de “homeless¹”. As operações básicas realizadas sobre um Home Location Agent são, consulta sobre a localização de um terminal móvel ou atualização da localização quando um terminal migra de um visited domain para outro.

¹Terminais móveis que não tem *Home Location Agente*, e não oferecem suporte a *handoff*. A conexão GTP entre a *Terminal Bridge* e a *Access Bridge* só permanece válida enquanto durar a conexão.

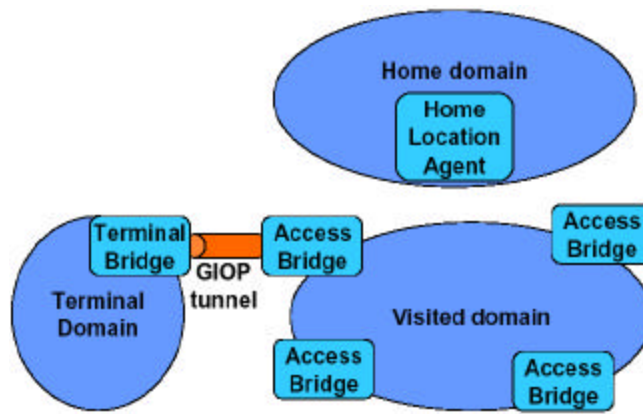


Figura 3 - Arquitetura para mobilidade do terminal em CORBA

Uma Terminal Bridge é essencialmente, um interceptador que viabiliza a comunicação entre todas as aplicações CORBA que estão executando no terminal móvel com a rede wireless. As aplicações se conectam na Terminal Bridge local (executando no terminal móvel), e este por sua vez, se conecta à Access Bridge atual. Os terminais só conseguem acessar objetos na rede fixa através da Access Bridge o qual estão conectados. A Terminal Bridge é responsável por encapsular as mensagens GIOP a serem enviadas para a Access Bridge e desencapsular as mensagens GIOP recebidas da mesma. A Terminal Bridge também pode prover um canal de eventos de mobilidade, que encaminha notificações relacionadas a handoffs e conectividades perdidas para sua Access Bridge. O túnel GIOP é o meio utilizado para transmitir mensagens entre a Terminal Bridge e a Access Bridge.

A Access Bridge faz um papel de proxy de objetos, intermediando a comunicação entre os terminais móveis na rede wireless e as estações da rede fixa. Desta forma, a Access Bridge torna-se o único meio através do qual objetos da rede fixa podem invocar objetos servidores nos terminais móveis e vice-versa. A Access Bridge gera notificações sobre os eventos de mobilidade do terminal para as outras Access Bridges interessadas neste evento.

2.2.4 Mobile IOR

Uma Mobile IOR é uma referência de objeto interoperável especial que oculta a mobilidade de um terminal móvel dos clientes que invocam operações sobre objetos localizados neste terminal. Uma Mobile IOR possui a seguinte estrutura em sua composição: Um IOP Profile normal (TAG_INTERNET_IOP) exigido em um IOR, e um Mobile Terminal Profile (TAG_MOBILE_TERMINAL_IOP). Em uma Mobile IOR podem existir mais de um IOP Profile, mas apenas uma instância do Mobile Terminal Profile.

Quando um objeto CORBA está em um terminal móvel, invocações a este objeto precisam ser encaminhadas de alguma forma para a Access Bridge atual deste terminal. Isto é realizado através da Mobile IOR. Definições da nova IDL para o Mobile IOR são mostradas na Figura 4, que ilustra o profile adicional do terminal móvel (ProfileBody) [OMGc, 2001].

A Mobile IOR é um objeto de referencia realocável. Ela identifica o terminal em que o objeto servidor está sendo executado e a Access Bridge ao qual o terminal está conectado. Além disto, ele identifica o Home Location Agent que mantém uma informação mais atual possível sobre a localização do terminal móvel, podendo assim, informar em qual Access Bridge um dado terminal se encontra conectado.

```
module MobileTerminal {  
  
    typedef sequence<octet> TerminalId;  
  
    typedef sequence<octet> TerminalObjectKey;  
  
    struct Version {  
  
        octet major;  
  
        octet minor;  
  
    };  
  
    struct ProfileBody {  
  
        Version mior_version;  
  
        Octet reserved;  
  
        TerminalId terminal_id;  
  
        TerminalObjectKey terminal_object_key;  
  
        sequence<IOP::TaggedComponent> components;  
  
    };  
  
    struct HomeLocationInfo {  
  
        MobileTerminal::HomeLocationAgent home_location_agent;  
  
    };  
  
};
```

Figura 4 - Nova IDL para a Mobile IOR

Em uma mobile IOR os IOP profiles não contem o endereço IP e porta do terminal em que objeto foi criado, e sim o endereço IP e porta do Home Location Agent ou da Access Bridge atual do terminal móvel. Se o terminal é homeless estas informações sempre serão referentes a Access Bridge. Se o terminal não é homeless, um dos componentes no profile do terminal móvel (Mobile Terminal Profile) é do tipo HomeLocationInfo e identifica o seu Home Location Agent.

Quando um Home Location Agent recebe uma requisição solicitando uma informação sobre a localização de um terminal que faz parte do seu home domain, ele processa o conteúdo do Mobile Terminal Profile contido no IOR alvo da

invocação, e envia uma resposta LOCATION_FORWARD para redirecionar a invocação para a Access Bridge atual do terminal.

Uma Access Bridge pode receber uma invocação destinada a um terminal móvel em qualquer das seguintes possibilidades: o terminal ser “homeless”, e por isso, este utiliza o endereço e porta da Access Bridge em seu IOR, ou pelo fato do Home Location Agent do terminal móvel ter redirecionado a requisição para a Access Bridge. Em qualquer destas circunstâncias a Access Bridge pode identificar o terminal alvo da requisição através do par: identificador do túnel GTP estabelecido e identificador do terminal. Se o terminal migrou para uma outra Access Bridge, a Access Bridge que recebeu a mensagem pode responder com um LOCATION_FORWARD (se a Access Bridge conhece a localização atual do terminal) retornando uma referência da Access Bridge atual ao qual o terminal se encontra associado, ou OBJECT_NOT_EXIST (Esta mensagem fará com que o cliente reenvie uma requisição para o Home Location Agent, caso este exista).

Nas versões 1.0 e 1.1 do GIOP a única informação que possibilita identificar o destinatário de um objeto é o object key, o que impossibilita o uso do Mobile Terminal Profiles para identificar o terminal móvel destinatário da mensagem. Diante desta situação, a especificação Wireless CORBA define um formato Mobile Object Key para ser utilizado em conjunto com os objects keys do objeto servidor em um terminal móvel que tem GIOP 1.0 ou 1.1. Um Mobile Object Key começa com uma string “MIOR” seguido por um encapsulamento da informação contida no Mobile Terminal Profile, exceto os componentes. Isto permite ao Home Location Agent e a Access Bridge rotear requisições para o terminal móvel da mesma forma como seria implementado no Mobile Terminal Profile. Um inconveniente desta solução é a duplicação das informações sobre o terminal móvel dentro do IOR. Esta solução só deve ser utilizada quando for necessário prover compatibilidade com GIOP de versões mais antigas. O MIWCO implementa o Mobile Object Key conforme definido na especificação.

2.2.5 Tunelamento e Retransmissão de Mensagens na Rede Wireless

As bridges se comunicam utilizando GTP (GIOP Tunneling Protocol). Da mesma forma como o GIOP descrito na especificação CORBA, o GTP é um protocolo abstrato modelado para ser mapeado sobre protocolos concretos do nível de transporte. Os requisitos de confiabilidade são os mesmos definidos para o GIOP. Diante desta imposição, a especificação Wireless CORBA define um nível de adaptação na arquitetura do GTP, para implementar a confiabilidade e outras exigências impostas aos protocolos de transporte. Ou seja, o nível de adaptação deve implementar ajustes aos protocolos de transporte que não se adequam aos requisitos definidos pelo GTP, por exemplo, o protocolo UDP (User Datagram Protocol) não é confiável e nem garante entrega ordenada das mensagens, no entanto, o nível de adaptação deve implementar estas funcionalidades para que este protocolo de transporte possa ser utilizado na implementação de um middleware que tenha como base a especificação Wireless CORBA.

2.2.5.1 Visão Geral do GTP

Um túnel GTP é o meio de transmitir mensagens GIOP entre uma Terminal Bridge e uma Access Bridge. A Wireless CORBA estipula que exista um único túnel entre uma dada Terminal Bridge e uma Access Bridge, que deve ser compartilhado entre todas conexões GIOP criadas entre a Terminal Bridge e a Access Bridge. O protocolo de tunelamento, portanto, realiza uma multiplexagem entre as conexões GIOP.

Os componentes definidos na especificação Wireless CORBA da arquitetura do GTP são ilustrados na Figura 5. As conexões GTP são associadas entre a Terminal Bridge do terminal móvel com a Access Bridge na rede fixa. A Access Bridge é responsável por realizar a conversão entre o IIOP utilizado pelo objeto na rede fixa e o GTP utilizado pelo terminal móvel no meio wireless.

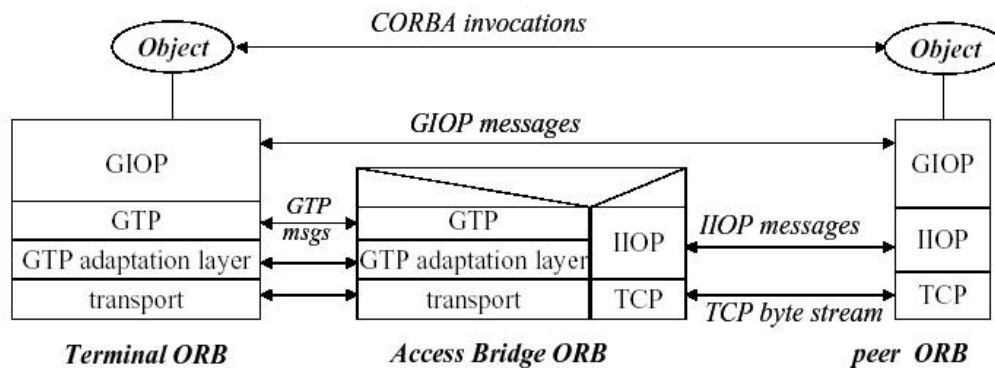


Figura 5 - Arquitetura do protocolo de tunelamento GIOP

2.2.5.2 Níveis de Adaptação do GTP

A especificação Wireless CORBA define três níveis de adaptação diferentes para o GTP, são eles: Transmission Control Protocol (TCP) [Ruggaber e Seitz, 2000], User Datagram Protocol (UDP) [Ruggaber e Seitz, 2000], e o Wireless Datagram Protocol (WDP) [WAP Forum, 2000], todos consistindo de um protocolo de tunelamento e um formato de endereço. O formato de endereço consiste do endereço IP e número da porta do servidor. O endereço IP pode ser dado na notação decimal ou nome DNS, dependendo do protocolo que está sendo adaptado.

Dentre os níveis de adaptação, o TCP é o mais simples, já que este implementa a confiabilidade exigida pelo GTP. O endereço IP ou nome de domínio podem ser utilizados no formato de endereço.

O protocolo de tunelamento UDP (UTP) no nível de adaptação executa sobre o protocolo UDP convencional. O UDP não provê nenhuma confiabilidade para transmissão das mensagens GTP. Sendo assim, o nível de adaptação tem que implementar todos os requisitos exigidos pelo GTP. A especificação do UTP confia nos níveis inferiores para garantir que as mensagens não sejam corrompidas, e por isso, ele não implementa nenhum checksum ou mecanismo similar. No formato de endereço é utilizado a notação decimal para representar o endereço IP e porta da Access Bridge. Não podem ser utilizados nomes de domínios como no TCP.

O protocolo de tunelamento do Wireless Application Protocol (WAP) no nível de adaptação é o WAP Tunneling Protocol (WAPTP), que utiliza o WDP como seu protocolo de transporte. O WDP é similar ao UDP, exceto que ele dispõe de capacidades de segmentação e confiabilidade opcional, que o WAPTP assume está sendo utilizado. Contudo, o WDP não garante o encaminhamento das mensagens de forma ordenada. Portanto, fica sendo responsabilidade da implementação do WAPTP realizar o despacho das mensagens GTP de forma adequada, garantindo sua entrega dentro da seqüência esperada pela outra ponta da comunicação. O nível de adaptação WAP utiliza um formato de endereçamento mais complexo, dado que, o WDP suporta vários níveis de endereços diferentes. Os tipos de endereços são: endereços IP ou números de telefone. Da mesma forma que o UDP o WDP utiliza endereços IP na notação decimal.

A especificação possibilita que outros níveis de adaptação possam ser definidos e implementados. Pesquisadores da Universidade Lappeenranta estão estudando uma forma de como especificar um nível de adaptação para Bluetooth [Merovuo, Valtaoja e Kotchanov, 2002].

2.2.5.3 Estrutura das Mensagens GTP

O GTP consiste de 17 tipos de mensagens, sendo que a maioria destas vem em pares Request-Reply. As mensagens podem ser agrupadas dentro de quatro categorias: gerenciamento de tunel, uso de conexões GIOP, forwarding GTP, e mensagens com propósitos específicos. A tabela 1 mostra o nome dessas mensagens, os respectivos identificadores exemplificando em que versão do GIOP elas se aplicam. Maiores detalhes podem ser encontrados na especificação [OMGc, 2001].

Todas mensagens GTP têm um cabeçalho de 8 bytes. Este é ilustrado na Figura 6.

```

struct GTPHeader {
    unsigned short seq_no;
    unsigned short last_seq_no_received;
    octet gtp_msg_type;
    octet flags;
    unsigned short content_length;
};

```

Figura 6 - Estrutura do cabeçalho GIOP

Nome da Mensagem	Identificador da msg GTP	Versão GTP
IdleSync	0x00	1.0, 2.0
EstablishTunnelRequest	0x01	1.0, 2.0

EstablishTunnelReply	0x02	1.0, 2.0
ReleaseTunnelRequest	0x03	1.0, 2.0
ReleaseTunnelReply	0x04	1.0, 2.0
HandoffTunnelRequest	0x05	2.0
HandoffTunnelReplyCompleted	0x06	2.0
OpenConnectionRequest	0x07	1.0, 2.0
OpenConnectionReply	0x08	1.0, 2.0
CloseConnectionRequest	0x09	1.0, 2.0
CloseConnectionReply	0x0A	1.0, 2.0
ConnectionCloseIndication	0x0B	1.0, 2.0
GIOPData	0x0C	1.0, 2.0
GIOPDataReply	0x0D	1.0, 2.0
GTPForward	0x0E	2.0
GTPForwardReply	0x0F	2.0
Error	0xFF	1.0, 2.0

Tabela 1 - Mensagens GTP

Cada mensagem GTP contém um número de seqüência de 16 bits para identificar as mensagens transmitidas. O algoritmo de geração dos números de seqüência é muito simplório, incrementando em 1 o ID (identificador) de cada mensagem GTP transmitida. Todas mensagens processadas possuem também um número de seqüência acknowledgement para confirmação de que as mesmas foram recebidas pelo seu destinatário. Estes tipos de mensagens são utilizadas para que o emissor possa retransmitir as mensagens não processadas pelo destinatário caso ocorra alguma perda durante handoffs ou durante uma situação inesperada.

2.2.6 Procedimentos de Handoff

A especificação define dois tipos diferentes de handoff: backward handoff e forward handoff. O primeiro ocorre em uma situação em que a conexão é comutada para um novo ponto de acesso na rede wireless. O segundo tipo de handoff ocorre quando a conexão é restabelecida após uma perda de conexão inesperada. A especificação Wireless CORBA se refere a estes como “Handoff” e “Access Recovery”, respectivamente.

A especificação Wireless CORBA adota a classificação comum para o handoff em duas sub-categorias, handoff inicializado pela rede e handoff inicializado pelo terminal.

Os diferentes tipos de handoffs, o access recovery, bem como os principais elementos participantes são descritos na próxima seção.

2.2.6.1 Participantes do Handoff

Em todos handoffs existem quatro componentes participantes: a Terminal Bridge, a Access Bridge atual do terminal (que passará a ser a old Access Bridge após o handoff), a nova Access Bridge e o Home Location Agente (HLA) do terminal. Dentre estes, o Home Location Agent não será considerado em caso de terminais homeless. Um outro detalhe a ser notado é que no Access Recovery, a nova Access Bridge pode ser a Access Bridge atual. Isso se aplica em situações inesperadas em que o terminal móvel encerra a conexão corrente e abre uma nova para a mesma Access Bridge. Isso também pode ocorrer em casos que o terminal móvel é reinicializado.

As migrações de um terminal móvel podem interessar outros componentes além do Home Location Agent e da Access Bridge. Para este propósito, a especificação CORBA Wireless define os eventos de mobilidade que podem ser gerados sempre que o terminal move de uma Access Bridge para outra. Tanto a Terminal Bridge quanto a Access Bridge geram estes eventos no momento em que o terminal realiza um handoff ou perde sua conexão corrente.

2.2.6.2 Handoff Inicializado pela Rede

No handoff inicializado pela rede presume-se que exista um agente externo que de alguma forma decide que o terminal deve migrar para uma nova Access Bridge. Este invoca a operação `start_handoff` na interface IDL da Access Bridge corrente do terminal. A especificação Wireless CORBA não deixa explícito a real necessidade de se ter um agente controlador externo que define quando e para onde um terminal móvel deve migrar. Pode-se imaginar algumas circunstâncias em que essa autonomia externa pode ser desejada, por exemplo, a possibilidade de fazer uma melhor distribuição dos terminais entre os diferentes pontos de acesso realizando algum tipo de balanceamento de carga, controle de congestionamento, ou outra necessidade gerencial qualquer. Os argumentos da operação `start_handoff` são: o identificador do terminal móvel e o endereço da nova Access Bridge. A Figura 7 mostra parte da interface IDL da Access Bridge. A old Access Bridge invoca a operação `transport_address_request` na nova Access Bridge, na qual retornará todos os endereços GTP disponíveis para conexão dos terminais móveis.

```
module MobileTerminal {  
  
    interface HandoffCallback {  
  
        void report_handoff_status (  
  
            in HandoffStatus status  
  
        );  
  
    };  
  
    interface AccessBridge {
```

```

...
void start_handoff(
    in TerminalId terminal_id,
    in AccessBridge new_access_bridge,
    in HandoffCallback handoff_callback_target
);
};

```

Figura 7 - Operações utilizadas no handoff inicializado pela rede

Os endereços recebidos pela old Access Bridge são então repassados para o terminal móvel que terá que realizar o handoff através de uma mensagem HandoffTunnelRequest.

O terminal ID especificado como parâmetro precisa ser único globalmente. Um esquema que pode ser utilizado para conseguir isto é concatenar as seguintes informações para gerar cada identificador:

- Versão do IP (1 byte)
- Endereço IP (4 ou 16 bytes), e
- Local_id (definido pelo administrador da rede, sendo que este tem que ser um identificador único dentro da organização)

Qualquer outro esquema pode ser utilizado, desde que ele gere identificadores únicos globalmente.

A seqüência de mensagens para o handoff inicializado pela rede é mostrada na Figura 8.

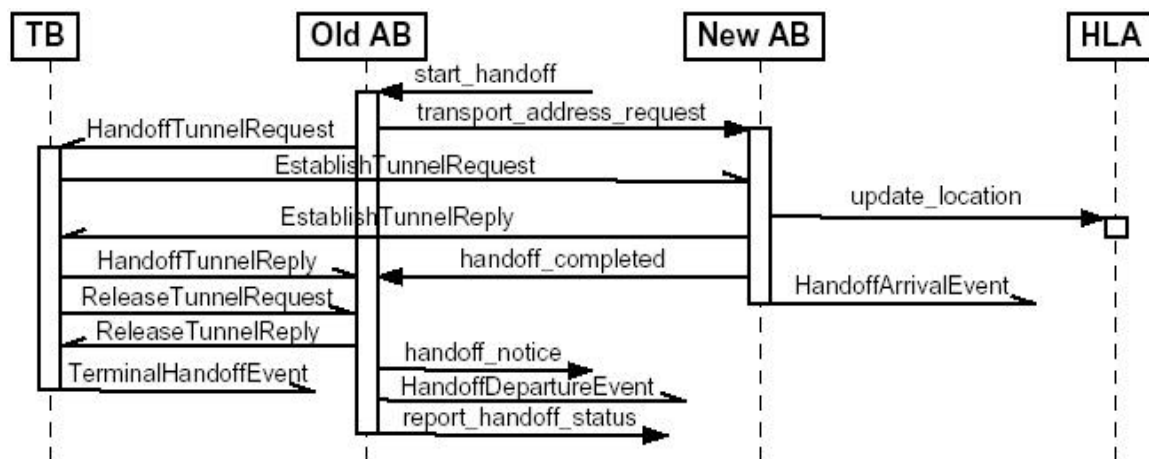


Figura 8 - Handoff Inicializado pela Rede

Ao receber a mensagem HandoffTunnelRequest o terminal tem o endereço GTP (segundo o formato definido de acordo com o protocolo de transporte utilizado pela

Access Bridge) da nova Access Bridge. Primeiro ele estabelece uma conectividade no nível de transporte de acordo com o protocolo utilizado. Depois disso ele envia uma mensagem EstablishTunnelRequest para a nova Access Bridge, contendo a referência da old Access Bridge e uma indicação que o terminal está inicializando uma operação de handoff. Com base nisto, a nova Access Bridge invoca a operação update_location no Home Location Agent do terminal móvel com intuito de atualizar a localização corrente do mesmo.

A nova Access Bridge envia uma mensagem EstablishTunnelReply para o terminal indicando que o handoff foi bem sucedido. Ao receber esta mensagem o terminal envia uma mensagem HandoffTunnelReply para a old Access Bridge indicando o sucesso na operação de handoff. Para finalizar o processo, a nova Access Bridge invoca a operação handoff_completed oferecida pela interface IDL da old Access Bridge para indicar que ela a partir de agora é responsável pelo terminal.

Logo em seguida o terminal envia a mensagem ReleaseTunnelRequest para fechar a conexão GTP com a old Access Bridge. Depois que a conexão é fechada todas as Bridges geram os eventos de mobilidade apropriados. A old Access Bridge notifica as Access Bridges anteriores sobre a migração do terminal, e finalmente a old Access Bridge invoca report_handoff_status sobre o objeto HandoffCallback referente à requisição inicial recebida na invocação start_handoff. Isto irá confirmar o resultado do handoff para o agente externo.

A implementação do MIwCO dispõe de uma ferramenta (nadmin) que pode ser utilizada para ativar handoffs a partir da rede.

2.2.6.3 Handoff Inicializado pelo Terminal

Neste tipo de handoff, assume-se que a Terminal Bridge de alguma forma seja capaz de identificar sua nova Access Bridge. O procedimento de handoff é inicializado através de uma mensagem EstablishTunnelRequest que carrega informações sobre a old Access Bridge e indica que o terminal quer realizar um handoff inicializado pelo terminal. Com base nisto, a nova Access Bridge executa a operação update_location para que o Home Location Agent possa atualizar a localização atual do terminal, e invoca a operação handoff_in_progress na old Access Bridge para que esta possa ficar ciente do handoff daquele terminal. Posteriormente, a nova Access Bridge envia uma mensagem EstablishTunnelReply de volta para a Terminal Bridge indicando que o novo túnel foi estabelecido. A seqüência de mensagens para o handoff inicializado pelo terminal é mostrada na Figura 9.

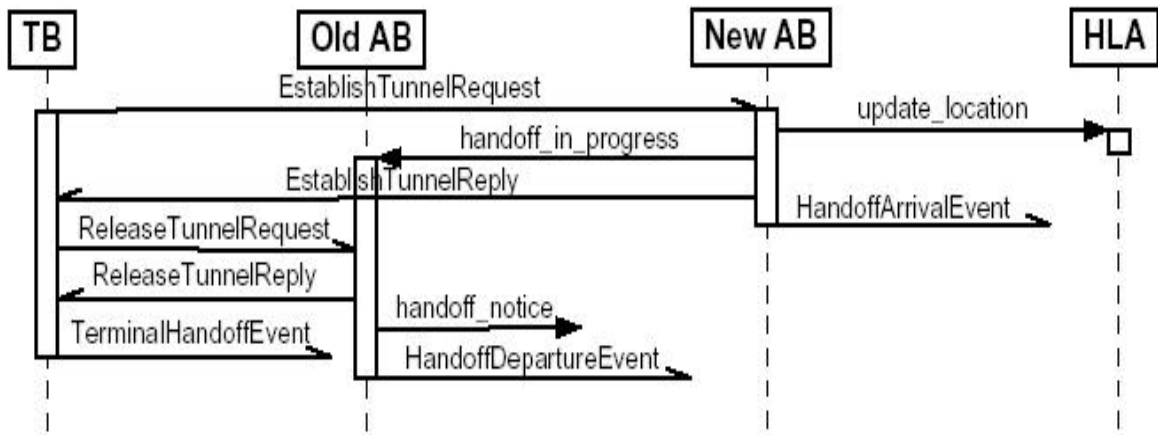


Figura 9 - Seqüências de mensagens do handoff inicializado pelo terminal

Após receber a mensagem EstablishTunnelReply a Terminal Bridge encerra a conexão com a old Access Bridge através da mensagem ReleaseTunnelRequest, todas Bridges (TB e AB) geram eventos de mobilidade, e a old Access Bridge notifica as Access Bridges anteriores sobre o movimento do terminal. O MIwCO disponibiliza a ferramenta tadmin para disparar os handoffs a partir do terminal.

2.2.6.4 Access Recovery (Reconstituição do Acesso)

A Access Recovery, é similar ao handoff inicializado pelo terminal, mas ao invés do terminal invocar a operação handoff_in_progress, a nova Access Bridge executa a operação recovery_request, na qual informa os números de seqüência do túnel GTP de ambos os lados da conexão antiga do terminal, de forma que a old Access Bridge e a Terminal Bridge possam se sincronizar.

Existem dois tipos de Access Recovey: O primeiro é quando o acesso é restabelecido para o mesma Access Bridge reconstituindo a conexão perdida anteriormente. O segundo tipo de Access Recovery é o estabelecimento de acesso para uma nova Access Bridge.

A seqüência de mensagens do Access Recovery para a mesma Access Bridge é mostrada na Figura 10 e para uma nova Access Bridge é mostrada na Figura 11.

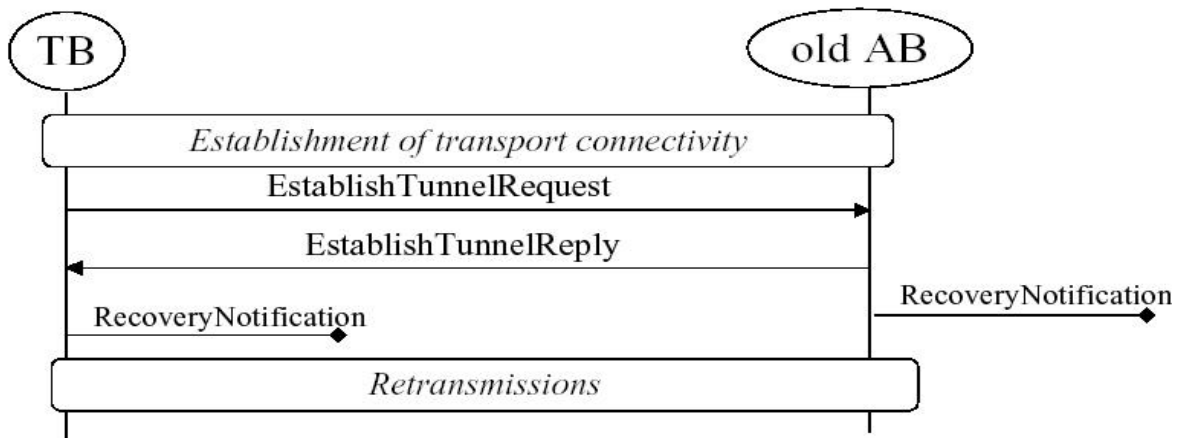


Figura 10 - Seqüências de mensagens para Access Recovery com a mesma Access Bridge

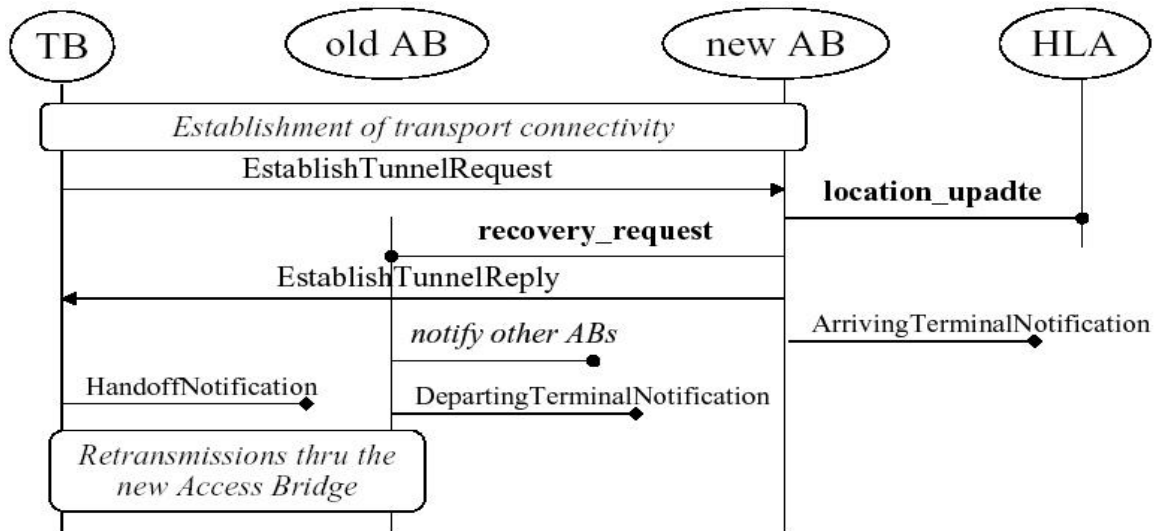


Figura 11 - Seqüência de mensagem do Access Recovery com uma nova Access Bridge

2.2.7 Processamento de Mensagens GTP

O GIOP requer que as respostas sejam enviadas na mesma conexão GIOP que foram enviadas as requisições (durante uma comunicação em andamento). Como a Access Bridge é o ponto final de uma conexão GIOP, as respostas devem obrigatoriamente passar através dela se o terminal tiver se deslocado para uma outra Access Bridge. Entretanto, a interface AccessBridge possui duas operações para serem utilizadas no repasse das mensagens GTP entre a Terminal Bridge e a old Access Bridge através da Access Bridge atual do terminal. A Figura 12 mostra a declaração das operações `gtp_from_terminal`, `gtp_to_terminal` na interface `AccessBridge`.

```

module MobileTerminal {
    ...
    interface AccessBridge {
        ...
        void gtp_to_terminal (
            in TerminalId terminal_id,
            in AccessBridge old_access_bridge,
  
```

```

        in unsigned long gtp_message_id,
        in GTPEncapsulation gtp_message
    ) raises (TerminalNotHere);

    ...

void gtp_from_terminal (
    in TerminalId terminal_id,
    in unsigned long gtp_message_id,
    in GTPEncapsulation gtp_message
);

};
};

```

Figura 12 - Operações utilizadas no processamento das mensagens GTP

Quando uma old Access Bridge recebe uma mensagem GIOP destinada a um terminal que tenha migrado para outra Access Bridge, aquela gera uma mensagem GTP e invoca a operação `gtp_to_terminal` na Access Bridge atual do terminal móvel. A old Access Bridge pode utilizar a operação `query_location` disponível na interface `HomeLocationAgente` para identificar a Access Bridge atual do terminal. A Access Bridge atual utiliza a mensagem `GTPForward` para encaminhar a mensagem GTP para a Terminal Bridge. Quando a Terminal Bridge precisa enviar uma mensagem GTP através da old Access Bridge, esta gera e envia uma mensagem `GTPForward` para a Access Bridge atual. Esta por sua vez, invoca a operação `gtp_from_terminal` na old Access Bridge.

3 MIDDLEWARES CORBA PARA AMBIENTES MÓVEIS.

O desenvolvimento de middlewares para ambientes móveis enfrenta dois problemas principais. Primeiramente o problema da gerência da mobilidade deve ser abordado, ou seja, toda a comunicação deve ser adaptada para o contexto de dispositivos que podem mudar sua localização. Outro problema a ser abordado é referente a computação ubíqua, que provê com que a estrutura do CORBA seja adaptada para dispositivos com escassez de recursos. A seguir são apresentados alguns trabalhos desenvolvidos no sentido de atacar tais problemas.

3.1 MIWCO

O MIwCO [Kangasharju, 2002] é um produto do projeto de pesquisa wCORBA na universidade de Helsinki e está implementado como uma extensão para o ORB de código fonte aberto MICO. A implementação do MIwCO foi feita paralelamente à

definição da especificação para assegurar que esta poderia ser implementada. Para atender esta exigência o MIwCO foi projetado de forma a facilitar o processo de re-implementação de certos componentes de acordo com as modificações ocorridas na especificação.

A implementação pode ser separada em três partes: o Home Location Agent, que é uma aplicação que não precisa estar ciente dos assuntos de mobilidade, as alterações na biblioteca do núcleo do MICO necessárias para prover suporte a implementação das aplicações CORBA no terminal móvel, e por último a implementação da Terminal Bridge e da Access Bridge, que são muito similares e compartilham boa parte dos seus códigos.

O MIwCO é uma implementação completa dos requisitos obrigatórios da especificação Wireless CORBA, e prover facilidades de implementação para os requisitos opcionais não implementados. Por exemplo, protocolos de transporte, UDP e WAP. MIwCO herda do MICO a facilidade de implementação dos níveis de adaptação, considerando que os níveis de adaptação são basicamente novas classes de endereços e transporte que são modeladas para ser facilmente estendidas no MICO. MIwCO adiciona os requisitos de implementação para reconhecimento de novos níveis de adaptação dentro de um código de inicialização da Bridge. Implementação dos níveis de adaptação não foi uma prioridade na implementação do MIwCO, atualmente (11/11/2002) o MIwCO implementa somente o nível de adaptação do TCP. Contudo, este protocolo não fornece os requisitos estimados pela computação móvel, conforme descrito em [AMIR , 1995].

Algumas das características do MIwCO são:

- GIOP Tunneling Protocol (GTP) sobre links wireless asseguram interoperabilidade entre Terminal e Access Bridge;
- Bridges agem como proxies para os objetos CORBA no lado da rede sem fio;
- Suporte a Handoff nas Bridges possibilita o terminal ter conectividade mesmo durante migrações entre células diferentes;
- GTP forward torna o handoff transparente para qualquer invocação CORBA ativa durante um handoff ;
- Suporte a terminais homeless.

3.1.1 Modelo da Implementação

A arquitetura para a implementação do sistema é mostrada na Figura 13. Nesta figura os círculos simbolizam as aplicações CORBA, os retângulos com bordas arredondadas são os componentes do MIwCO, os retângulos são os computadores, as linhas tracejadas são possíveis conexões GIOP e as linhas pontilhadas são possíveis conexões GTP.

A arquitetura inclui dois componentes, nadmin e tadmin, que não fazem parte da especificação Wireless CORBA. O propósito destes programas é desempenhar funções administrativas que na maioria das vezes estão relacionadas ao gerenciamento das conexões GTP. Tanto a Access Bridge quanto o Terminal Bridge

são objetos CORBA que possuem interfaces IDL com operações que podem ser executadas para realizar o gerenciamento da Bridge em questão. O nadmin e o tadmin utilizam estas interfaces IDL para desempenhar as suas funções.

Uma característica da arquitetura que deve ser ressaltada é o papel desempenhado pela Terminal Bridge. No sistema implementado, a Terminal Bridge executa como um processo separado. Tal decisão de projeto foi definida porque o MICO é implementado como uma biblioteca, sendo assim, a partir da necessidade de transmitir todo o tráfego do terminal móvel (possivelmente de várias aplicações CORBA), através de um única conexão GIOP, é necessário que este tráfego seja enviado para a Terminal Bridge, e esta irá retransmiti-lo para Access Bridge. Esta implementação requer que o sistema operacional do terminal móvel seja multitarefa.

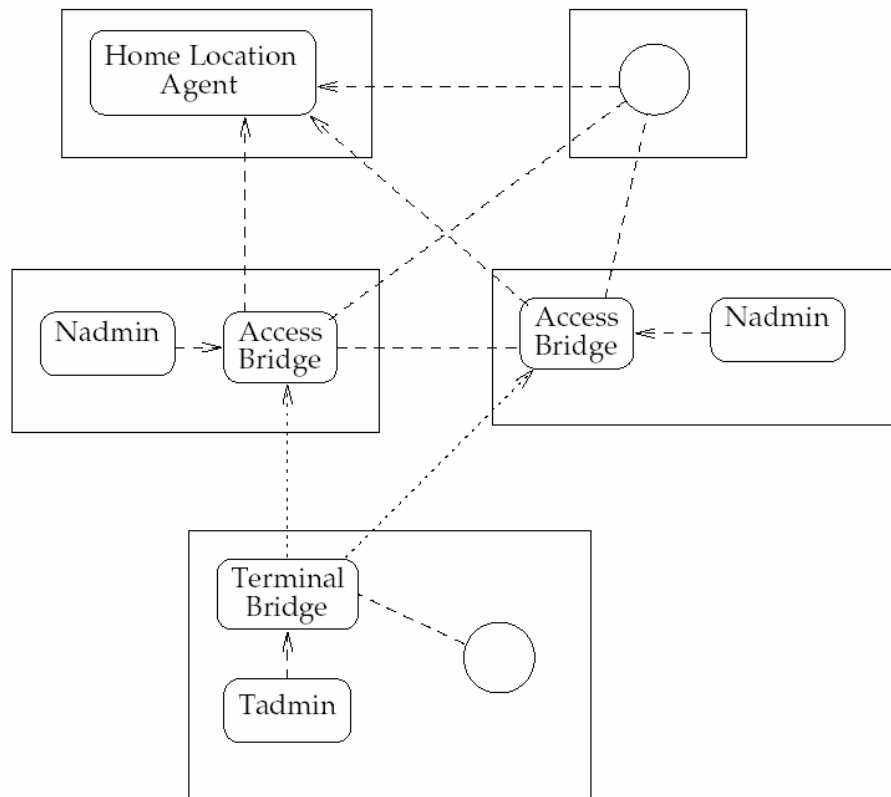


Figura 13 - Modelo da arquitetura do MIwCO

3.1.1.1 Mudanças no Núcleo do MICO

Quando uma aplicação CORBA executa em um terminal móvel, ela necessita desempenhar outras atividades que não são implementadas nas aplicações CORBA que executam na rede fixa. Para possibilitar isso, foram realizadas inúmeras modificações no núcleo do MICO para atender algumas necessidades específicas. Algumas dessas necessidades são: Um terminal móvel deve transmitir suas mensagens GIOP para o Terminal Bridge ao invés de enviá-las diretamente ao seu destinatário. Um servidor no dispositivo móvel precisa gerar Mobile IORs, que contêm o Mobile Terminal Profile e o endereço do Home Location Agent ou do Access Bridge atual, ao invés do seu próprio endereço. Em adicional, um servidor

necessita registrar seu object key na Terminal Bridge para que esta saiba redirecionar as mensagens destinadas ao objeto servidor.

Além das alterações realizadas no núcleo do MICO, um outro objetivo foi o de manter a implementação que dá suporte a mobilidade separada da implementação CORBA que as aplicações convencionais utilizam. A idéia era permitir às aplicações verificar se estão em um terminal móvel, mas caso não estivesse, nenhuma operação adicional deve ser realizada. Desta forma, a compatibilidade com as aplicações CORBA convencionais é preservada.

Os recursos que dão suporte ao padrão Wireless CORBA introduzidos no código do MICO, só são ativados durante a sua compilação se as opções de configuração abaixo forem selecionadas:

```
./configure --enable-wireless --enable-wireless-home --enable-wireless-terminal  
--enable-wireless-access --enable-debug
```

Durante a inicialização do ORB duas opções adicionais [Kangasharju, 2002] foram inseridas para configurar o identificador do terminal e o endereço do Terminal Bridge para onde as mensagens GIOP deveriam ser redirecionadas. Ao configurar estas opções, o ORB identificará que está executando em um terminal móvel.

Um outro aspecto importante da implementação foi a introdução do envio da mensagem NEEDS_ADDRESSING_MODE. O sistema Wireless CORBA desempenha melhor suas funcionalidades quando os clientes e servidores utilizarem a versão 1.2 do GIOP. Isso se dá devido à necessidade dos clientes enviarem suas requisições informando o IOR completo do objeto (IP do Home Location Agent ou da Access Bridge, Mobile Profile Terminal, Terminal ID, Object key). Infelizmente, o MICO não suportava o GIOP 1.2 quando sua implementação foi iniciada. Sendo assim, não foi possível utilizar o modo de endereçamento baseado em referência, utilizando somente o modo de endereçamento padrão baseado nos identificadores do objeto. Para contornar este problema, foi implementado o envio da mensagem NEEDS_ADDRESSING_MODE, para requisitar ao cliente CORBA na rede fixa, uma nova requisição utilizando o formato de endereçamento baseado na referência do GIOP 1.2. Esta mensagem será enviada para o cliente se a primeira requisição recebida tiver sido gerada por um GIOP inferior a 1.2.

A principal modificação concebida foi a geração de Mobile Object Keys. Isto foi estritamente necessário para tratar questões de compatibilidade com o GIOP 1.0 e 1.1.

3.1.2 Interfaces Bridge CORBA

No lado do terminal existem duas operações especiais que implementam os procedimentos de estabelecimento de conexão à uma Access Bridge e registro de objetos CORBA a uma TB. Uma delas é a operação ensure_connectivity utilizada para abrir uma conexão GTP com uma dada Access Bridge. Esta operação é utilizada no acesso inicial e em um access recovery. O endereço da Access Bridge a ser contactada deve ser especificado como parâmetro. A outra operação importante

presente na Terminal Bridge, `register_object`, é utilizada pelos servidores CORBA para registrar seus objetos na Terminal Bridge do terminal móvel, a partir disso, este poderá redirecionar as invocações recebidas para o objeto servidor corretamente.

A especificação Wireless CORBA define uma interface IDL para a Access Bridge e uma para a Terminal Bridge. Estas interfaces IDL facilitam realizar certas ações administrativas que podem vir a ser necessárias. Por exemplo, a aplicação `nadmin` utiliza as operações da interface da Access Bridge para disparar handoffs a partir da rede (handoff inicializado pela rede) nos terminais móveis conectados a ela.

Com propósitos gerenciais, foi implementado no MIwCO uma interface administrativa para a Terminal Bridge. Assim seria possível forçar determinadas circunstância no comportamento do terminal móvel, como por exemplo, conexões perdidas e handoffs. A interface IDL do Terminal Bridge é mostrada na Figura 14.

```
interface TerminalBridge {  
    ObjectIdList list_initial_services ();  
    Object resolve_initial_references (  
        in ObjectId identifier  
    ) raises (InvalidName);  
    string register_profile(  
        in unsigned long tag,  
        in IORProfile profile  
    ) raises (AddressNotAvailable);  
  
    void connect (in string address);  
    void release (in boolean notify_ab);  
    void do_handoff (in string address);  
};
```

Figura 14 - Interface IDL da Terminal Bridge

As operações de gerenciamento das conexões GTP da interface da Terminal Bridge foram originalmente projetadas com propósitos de realizar certos experimentos, mas elas podem ser utilizadas para outras finalidades. Por exemplo, desenvolver uma aplicação que monitora a qualidade de conectividade do link e que, ao detectar um nível de baixa qualidade, pode invocar a operação `do_handoff`, caso esta conheça uma Access Bridge com conectividade melhor, ou que simplesmente encerre a conexão se nenhuma outra Access Bridge foi encontrada.

3.2 e*ORB

O e*ORB [Vertel, 2001] é um ORB comercial desenvolvido pela empresa Vertel, projetado para sistemas embutidos e de tempo-real. O e*ORB foi desenvolvido especificamente para o ambiente de telecomunicações e muitas de suas características são adequadas às exigências de redução de recursos e confiabilidade de ambientes móveis. Apesar do e*ORB não seguir a especificação Wireless CORBA da OMG, este se mostra como uma alternativa viável ao desenvolvimento de aplicações CORBA para ambientes móveis.

3.2.1 Incorporação dinâmica de protocolos

Para satisfazer as restrições de tempo de resposta dos ambientes de tempo real, se faz necessária a exigência de um certo grau de QoS. Para tanto o e*ORB fornece mecanismos que permitem incorporar ao ORB protocolos de transporte eficientes e mais adequados às necessidades da aplicação. Em tal mecanismo o protocolo é dividido em duas partes conceituais, o protocolo de mensagens (e.g. GIOP) e o protocolo de transporte (e.g. TCP/IP)

Desta forma, o protocolo IIOP é desmembrado em seus componentes básicos: um formato de mensagem GIOP sobre uma conexão TCP/IP. Um outro exemplo de protocolo que poderia ser incorporado ao ORB seria o protocolo composto por um formato de mensagem GIOP sobre uma conexão ATM, o que resultaria em um protocolo com um grau de QoS que poderia ser utilizado em aplicações de tempo real.

O e*ORB fornece uma implementação open-source dos componentes do protocolo de mensagem e do protocolo de transporte que compõe o IIOP (i.e GIOP e TCP/IP), para ser utilizada como exemplo para o desenvolvimento de outros protocolos incorporáveis ao ORB.

3.2.2 Arquitetura micro-kernel

O núcleo do ORB do e*ORB é reduzido, o que o torna adequado para plataformas de hardware com recursos limitados. Entretanto outros módulos opcionais podem ser carregados dinamicamente e adicionados ao ORB a medida que surge a necessidade de uma determinada funcionalidade. Desta forma, é possível utilizar desde um ORB reduzido (como especificado pelo padrão minimum-CORBA da OMG) até um ORB totalmente compatível com o padrão CORBA.

Entretanto esta funcionalidade de incorporação de módulos adicionais somente é disponível em plataformas e sistemas que admitem módulos dinâmicos, como Windows, Solaris e Linux. Nos demais, é necessário compor o ORB estaticamente no momento da compilação da aplicação.

3.2.3 Alocação de memória

Os recursos das plataformas de software convencionais são normalmente melhor administrados pelo sistema operacional. Entretanto, em plataformas móveis, que dispõe de recursos escassos, muitas vezes é mais adequado tratar o uso dos recursos de uma forma mais sistemática.

Para tanto, o e*ORB permite a definição de alocadores de memória, que permitem alterar todo o modelo de alocação e liberação de memória utilizando tanto pela aplicação quanto pelo próprio ORB. O e*ORB disponibiliza o alocador de memória padrão do C++ para as versões de debug e um alocador de alta performance para versões de distribuição da aplicação.

3.3 Universally Interoperable Core (UIC)

O UIC [Roman e Campbell, 2001] é uma infraestrutura de middleware desenvolvida para cenários de computação ubíqua, que se propõe a tratar os problemas encontrados nas plataformas de middleware existentes. O UIC define um esqueleto baseado em componentes que encapsulam funcionalidades padrões comuns a maioria dos ORBs, dentre as quais podemos citar as seguintes:

- Protocolos de rede e transporte;
- Estabelecimento de conexões;
- Estratégias de empacotamento e desempacotamento;
- Chamadas remotas de métodos;
- Geração e reconhecimento de referências de objeto;
- Registro de objetos;
- Interface de clientes e servidores;
- Gerência de memória;
- Estratégias de concorrência.

Este esqueleto pode ser adaptado, estaticamente ou dinamicamente, para as necessidades dos mais diversos dispositivos e ambientes portáteis, permitindo assim manter uma estrutura simples e que contenha apenas os recursos necessários para uma determinada aplicação. O projeto modular do esqueleto, permite a separação de conceitos e encapsula os mecanismos necessários para cada tarefa específica em diferentes componentes, minimizando dependências indesejáveis e facilitando a montagem de esqueletos adaptáveis.

Apesar do UIC definir um esqueleto padrão projetado para middlewares orientados a objetos, como CORBA, Java RMI e DCOM, nada impede que o desenvolvedor da aplicação altere a estrutura de forma a atender outros requisitos, como por exemplo plataformas de chamada remota a procedimento não orientadas a objetos.

Uma determinada configuração do esqueleto é denominada personalidade. Personalidades podem ser classificadas como cliente (envia pedidos e recebe respostas), servidora (recebe pedidos e envia respostas), ou ambas. O UIC pode ser classificado como de personalidade única ou de multipersonalidade. Entretanto, um UIC de multipersonalidade não é o mesmo que uma coleção de UIC de personalidade única. No primeiro caso a aplicação utiliza o único UIC e a mesma interface de chamada de métodos independentemente do objeto remoto, já no segundo caso, a aplicação é responsável por selecionar a instância do UIC adequada.

As personalidades do UIC podem ser configuradas estaticamente, montando todos os componentes no momento da compilação, o que permite uma redução no tamanho o UIC; ou dinamicamente, carregando as bibliotecas necessárias em tempo de execução; ou uma forma híbrida, onde se define um conjunto de componentes estáticos e outros dinâmicos.

Os principais benefícios do UIC são: a infraestrutura homogênea que esconde as diferenças dos dispositivos portáteis; a sua estrutura baseada em componentes, que permite a especificação de um esqueleto específico para uma plataforma, tanto através da composição de um esqueleto adaptado, quando pela especialização de determinados componentes para as necessidades da aplicação; e a capacidade de se adaptar dinamicamente às mais diversas plataformas, através da reconfiguração da personalidade do UIC.

4 CONCLUSÃO

A necessidade de middlewares de comunicação para ambientes móveis se tornou evidente em face à evolução e a popularização de tecnologias de rede sem fio. Por isso, diversos esforços tem sido feitos para adaptar a tecnologia CORBA ao contexto de aplicações móveis. Tratando tanto a questão da gerência de mobilidade nos ambientes de rede sem fio, quanto o problema de adaptação da aplicação à escassez de recursos e variação de QoS encontradas nas plataformas de dispositivos móveis.

Diversas propostas para o suporte da gerência da mobilidade em CORBA foram feitas, como é o caso da proposta de adaptação do modelo CORBA sobre o protocolo WAP e do projeto e*ORB da Vertel. Entretanto, estas são soluções proprietárias ou dependentes de uma tecnologia, além de apresentar problemas na integração com ORBs não-móveis. Contudo, a especificação do modelo Wireless CORBA proposta pela OMG é uma solução eficiente, independente de plataforma, não-proprietária e compatível com o padrão CORBA, além de já estar disponível através da extensão proposta pelo ORB MIwCO.

O problema referente à computação ubíqua é abordado no projeto do UIC, que fornece um framework adaptável, através do qual é possível construir uma infraestrutura para um middleware de acordo com as necessidades do ambiente, permitindo a construção de ORBs adaptáveis e de tamanho reduzido. Outra abordagem similar é utilizada pelo projeto e*ORB, que implementa um núcleo adaptável de um ORB CORBA. Apesar das soluções propostas serem eficazes e compatíveis, ainda não existe nenhuma solução única que aborde ambos os

problemas da adaptação do modelo de CORBA a ambientes móveis de forma satisfatória.

Referências Bibliográficas

[OMG, 2002] Object Management Group, Needham, Massachusetts; *Common Object Request Broker Architecture and Specification*; versão 3.0; Junho de 2002.

[AMIR, 1995] Amir, E.; Balakrishnan, H.; Seshan, S.; Katz, R. H.; *Efficient TCP over networks with wireless links*; Proceedings of the Fifth IEEE Workshop of Hot Topics in Operating Systems, Maio de 1995.

[Hartenstein, 2001] Hartenstein, H.; Jonas, K.; Liebsch, M.; Schmitz, R.; Stiernerling, M.; Westhoff, D.; *Performance of TCP in the Presence of Mobile IP Handoffs*; ICT 2001 IEEE International Conference on Telecommunications, Junho de 2001.

[Fladenmuller e Silva, 1999] Fladenmuller, A.; De Silva, R.; *The effect of Mobile IP handoffs on the performance of TCP*; Journal on Mobile Networks and Applications, vol. 4, 1999.

[OMGb, 1999] Object Management Group, Needham, Massachusetts; *Wireless Access and Terminal Mobility: Request For Proposal*, May 1999. OMG document telecom/99-05-05.

[OMGc, 2001] Object Management Group, Needham, Massachusetts; *Wireless Access and Terminal Mobility in CORBA*; Adopted specification. OMG Document dtc/01-06-02, June 2001.

[Postel, 1981] Postel, J.; *Transmission Control Protocol*, September 1981. RFC 793 <ftp://ftp.internic.net/rfc/rfc793.txt>.

[Ruggaber e Seitz, 2000] Ruggaber, R.; Seitz, J.; *Using CORBA Applications in Nomadic Environments*. Proceedings of the 3rd IEEE Workshop on Mobile Computing Systems and Applications, 2000.

[WAP Forum, 2000] WAP Forum; *Wireless Datagram Protocol Specification*, February 2000. WAP Forum document WAP-201-WDP.

[Merovuo, Valtaoja e Kotchanov, 2002] Merovuo, S., Valtaoja, A., and Kotchanov, A.; *Accessing CORBA services from bluetooth mobile terminals*. In Smart Networks: 7th International Conference on Intelligence in Networks, pages 149–159. International Federation for Information Processing, Kluwer Academic Publishers, April 2002.

[Kangasharju, 2002] Kangasharju, J.; *Implementing the Wireless CORBA Specification*, Laudatur Thesis, Department of Computer Science, University of Helsinki, <http://kotisivu.mtv3.fi/ashar/software/miwco/laudatur-jjk.pdf> (último acesso em Março 2005), May 20, 2002.

[Vertel, 2001] Vertel Corporation; *e*ORB™ Technical Description*, Março de 2001.

[Roman e Campbell, 2001] Roman, M.; Kon, F.; Campbell, R. H.; *Reflective Middleware: From Your Desk to Your Hand*. IEEE Distributed Systems Online Journal, Special Issue on Reflective Middleware, Julho de 2001.