MAC5701 - Tópicos em Ciência da Computação

Um arcabouço para simulação de protocolos de entrega de mensagens em redes móveis.

Weslley Emmanuel Martins Lima
Orientador: Alan Mitchell Durham
Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

25 de junho de 2002

Resumo

Nos últimos anos houve um enorme crescimento na utilização de dispositivos móveis, principalmente em serviços de recuperação de informação. No entanto, a inserção da mobilidade trouxe importantes desafios aos pesquisadores na adaptação das aplicações existentes e na construção de novas aplicações para tal ambiente. Dentre esses desafios, está o de garantir confiabilidade a aplicações que possuem essa característica como requisito fundamental, apesar da intrínseca imprevisibilidade do ambiente de computação móvel.

Esta monografia descreve alguns dos protocolos de entrega confiável de mensagens em redes móveis e analisa as situações a que cada um melhor se adequa. Partindo dessas informações, é feita uma análise das semelhanças e diferenças tentando separar as funcionalidades internas de cada protocolo em módulos de funcionalidade bem definida. Partindo disso, um arcabouço é proposto para a construção de protocolos com as mesmas características.

1 Introdução

Os grandes avanços tecnológicos ocorridos nos últimos anos propiciaram a criação de um novo paradigma de comunicação, no qual computadores portáteis podem comunicar-se com uma rede fixa ou até mesmo com outros computadores portáteis independente da sua localização física. Esse novo paradigma é chamado de **computação móvel**.

Outro fator importante que tem contribuído para a viabilidade do ambiente de computação móvel, também decorrente dos avanços tecnológicos, é o aumento do poder computacional dos dispositivos móveis. Aumentos consideráveis tem ocorrido no poder de processamento e na quantidade de memória desses dispositivos.

No entanto, a mobilidade introduz vários problemas e restrições na construção de aplicações. Além disso, surgiram importantes desafios na adaptação dos sistemas e serviços existentes. Um fator relevante que dificulta a adaptação desses sistemas e serviços é a diversidade tecnológica entre o ambiente de computação móvel e as redes fixas existentes.

Um dos desafios é localizar uma unidade móvel dentro do sistema. Existem duas possibilidades, manter a informação atualizada da localização da unidade móvel ou buscá-la no sistema quando necessário. Porém essas técnicas possuem restrições que devem ser consideradas no projeto de uma rede sem fio.

Outros desafios importantes introduzidos pela mobilidade são garantir a entrega de mensagens a máquinas que mudam sua localização(roteamento) e permitir que as unidades móveis possam realizar *operações desconectadas*. A necessidade de operações desconectadas decorre da instabilidade que há na comunicação sem fio ou da possibilidade das unidades móveis migrarem para regiões não cobertas por células.

A energia limitada das unidades móveis é um outro problema relevante na construção de aplicações que devem utilizar técnicas de tolerância a falhas para recuperar-se de repentinas faltas de energia.

A largura de banda do meio sem fio é pequena quando comparada com a rede fixa. Dessa forma, as aplicações devem considerar esse fator durante o processo comunicação com a unidade móvel, buscando minimizar o tráfego pelo meio sem fio. Além disso, o meio sem fio é muito suscetível a interferências(ruído, atenuação) que afetam diretamente a taxa de erro. Segundo [ML98], a taxa de erro na comunicação sem fio é de cinco a dez ordens de grandeza maior que na rede fixa.

Diante de todos os desafios e problemas que surgem com a introdução da mobilidade, fica clara a necessidade de se construir novos protocolos e adaptar os já existentes, de forma que considerem todos os problemas advindos da mobilidade. Este monografia descreve alguns protocolos confiáveis de entrega de mensagens para o ambiente de computação móvel e faz uma análise comparativa entre os protocolos. Por fim, descreve uma proposta inicial de um arcabouço para a construção de protocolos de entrega de mensagens.

2 Modelo de Sistema

Em computação móvel, existem dois possíveis modelos de sistema: *redes ad-hoc e redes móveis estruturadas*. Em redes ad-hoc, unidades móveis podem comunicar-se diretamente com outras unidades móveis que encontram-se na sua área de cobertura ou utilizar outra unidade móvel para intermediar a comunicação. Em redes móveis estruturadas, a comunicação de uma unidade móvel com o restante do sistema é intermediada pela rede fixa.

Nesta monografia, o modelo de sistema adotado é o modelo de redes móveis estruturadas descrito em [BBIM93]. Estruturalmente, esse modelo pode ser dividido em duas partes: móvel e fixa. A porção fixa do sistema é composta por:

1. **Servidores** que são máquinas cuja funcionalidade é específica para cada aplicação e que encontram-se conectados fisicamente à rede;

2. **Estações de suporte a mobilidade(Mss)** responsáveis de intermediar toda e qualquer comunicação das unidades móveis com o sistema.

A cada Mss está associado a uma região geográfica chamada **Célula**. Toda unidade móvel localizada nesta região é de responsabilidade do Mss associado à Célula.

A porção móvel do sistema é composta apenas pelas unidades móveis que a partir de agora serão chamadas de Mh por questões de comodidade. Um Mh possui um identificador único em todo o sistema e pode encontrar-se em dois estados: *ativo ou inativo*. Quando ativo, o Mh pode enviar e receber informações do sistema. Quando encontra-se inativo, não pode estabelecer qualquer tipo de comunicação com o sistema.

Um Mh utiliza o meio sem fio para comunicar-se com o sistema. Dessa forma, pode mudar de célula em qualquer instante de tempo. A esse fato dá-se o nome de *migração*. Ao entrar em uma nova célula, o Mh detecta uma mudança de localização e envia uma mensagem ao Mss responsável(*respMss*) pela nova célula, a mensagem **Greet**. Essa mensagem carrega consigo o endereço do Mh e outras informações particulares de cada protocolo.

A figura 1 mostra um cenário que exemplifica o modelo de sistema adotado.

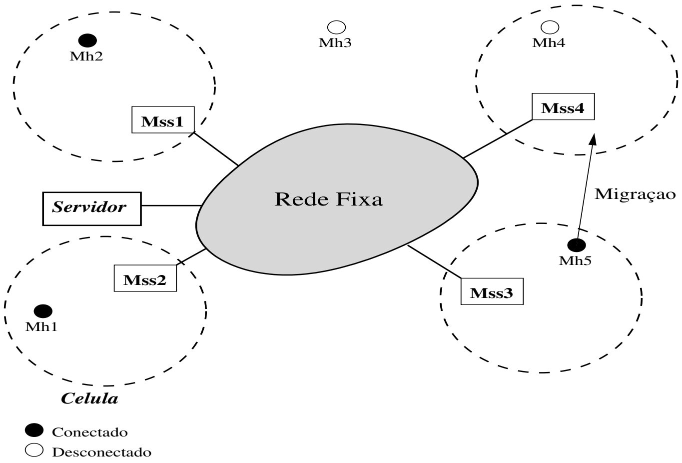

Figura 1: Modelo de Sistema

3 Descrição dos protocolos

Nesta seção são apresentados alguns protocolos de entrega de mensagens em redes móveis. São descritas as suposições que devem ser consideradas para garantir a entrega confiável, as mensagens trocadas entre os elementos de rede do sistema, as estruturas de dados envolvidas e como se dá o funcionamento de cada protocolo. Finalmente, alguns cenários exemplificam funcionamento dos mesmos.

3.1 RDP

O RDP(Result Delivery Protocol) [ESO00] é um protocolo não orientado a conexão adequado a serviços baseados no estilo requisição-resposta de comunicação que, levando em consideração algumas suposições descritas a seguir, garante a entrega de mensagens a clientes móveis. Foi desenvolvido para dar suporte a operações de *consulta* e *assinatura* dentro do projeto SIDAM [ESS+00].

O RDP é baseado na noção de "solicitador de requisições", por comodidade chamado de **proxy**. O proxy é um objeto que tem como principal função intermediar a comunicação entre o Mh e os servidores na rede fixa. É criado no Mss responsável pela célula onde encontra-se o Mh no momento em que novas requisições são emitidas. No proxy são mantidas todas as requisições pendentes e os resultados das requisições a serem repassados ao Mh. Este objeto permanece no sistema somente até que todas as requisições pendentes sejam atendidas. A partir de então, um novo proxy será criado na atual localização do Mh quando uma nova requisição for realizada. É importante ressaltar que em qualquer instante cada Mh está associado a um único proxy.

3.1.1 Suposições

As suposições de modelo são importantes para a corretude no funcionamento, são elas:

1. Comunicação confiável entre *Mss's*;
2. Todos os *Mss's* são confiáveis e não falham;
3. Cada *Mh* está associado a um único Mss;
4. Um *Mh* só deixa o sistema após confirmar o recebimento de todas mensagens;
5. O *Mh* detecta retransmissões.

3.1.2 Estruturas de Dados

A seguir são listadas as estruturas de dados mais importantes existentes nas entidades participantes:

1. No Proxy:
   (a) `currentLoc` - Variável contendo o endereço do atual respMss do Mh correspondente. É atualizada a cada migração do Mh.
   (b) `requestList` - Lista de todas as requisições pendentes emitidas por um Mh.
2. No respMss:
   (a) `pRef` - Referência para o atual proxy associado ao Mh. Permite que o proxy seja notificado de toda migração do Mh. Possui uma flag chamada `RKpR`*(Ready to Kill pRef)* que indica se o proxy encaminhou o resultado da última requisição pendente.

3.1.3 Especificação das Mensagens

A Tabela 1 especifica todas as possíveis mensagens trocadas entre duas máquinas(móveis ou fixas) do sistema.

Mensagens	Participantes	Significado
greet	Mh → Mss	Mudança para a célula controlada por Mss
request	Mh → Mss; Mss → Server	Envio de requisição a Server
result	Server → Mss	Envio de resposta a Mss
forwardRes	Mss → Mss	Envio do resultado ao atual respMss
forwardResWl	Mss → Mh	Envio do resultado ao Mh
ack	Mh → Mss Mss → Mss	Confirmação do recebimento de um resultado
forwardAck	Mss → Mss	Encaminhamento de confirmação ao Proxy
deReg	Mss → Mss	Desregistro de um Mh
deRegAck	Mss → Mss	Confirmar o desregistro de um Mh
updateCurrL	Mss → Mss	Atualizar a localização de um Mh no proxy

Tabela 1: Mensagens do RDP

3.1.4 Funcionamento

Sempre que um Mh envia uma requisição, seu respMss verifica o valor de `pRef`. Caso ele seja nulo, um novo proxy é criado e a requisição é encaminhada ao proxy que se encarregará de repassá-la ao servidor. O servidor processa a requisição e retorna a resposta ao proxy que a encaminhará a atual localização do Mh.

No entanto, um Mh pode mudar sua localização antes que as respostas cheguem até ele. Portanto é importante que, ao ocorrer uma migração, ocorra também a transferência das informações relacionadas ao Mh do antigo para o novo Mss(protocolo de Hand-off). No caso do RDP, o valor de `pRef` é transferido para o novo Mss. Terminado o Hand-off, o novo respMss envia a mensagem `update currentLoc` ao proxy que atualiza a localização do Mh correspondente na variável `currentLoc` e, logo a seguir, qualquer resultado pendente é encaminhado à nova localização do Mh.

Todo resultado de requisições pendentes que chegam ao proxy são encaminhados ao respMss e repassados ao Mh através do meio sem fio. O Mh deve confirmar o recebimento de todos os resultados o que permitirá ao proxy remover de `requestList` todas as requisições já atendidas. A confirmação do recebimento é essencial para a garantia de confiabilidade do protocolo.

A figura 2 mostra um cenário que ilustra o funcionamento do RDP. Neste exemplo, o Mh envia uma requisição a um servidor e antes que receba o resultado migra para a célula do Mss_o e logo depois do Mss_n. Quando o Mss_p(localização inicial do Mh) recebe a requisição, verifica que `pRef` é nulo e cria um novo proxy. A resposta é encaminhada ao proxy que repassa ao Mss cujo endereço encontra-se em `currentLoc`(Mss_o). Junto com o resultado segue `del-pRef=true` indicando ao Mh que a última requisição pendente foi atendida.

No entanto, antes que receba a resposta, o Mh migra para o Mss_n. A mensagem `update currentLoc` é enviada pelo Mss_n ao proxy, esse atualiza o valor de `currentLoc` e encaminha a resposta pendente para o Mss_n. A resposta é enviada pelo Mss_n ao Mh, que confirma o recebimento da resposta. Essa confirmação é repassada ao Mss_p juntamente com `del-proxy=true`. Ao receber a mensagem de confirmação, o proxy é removido.

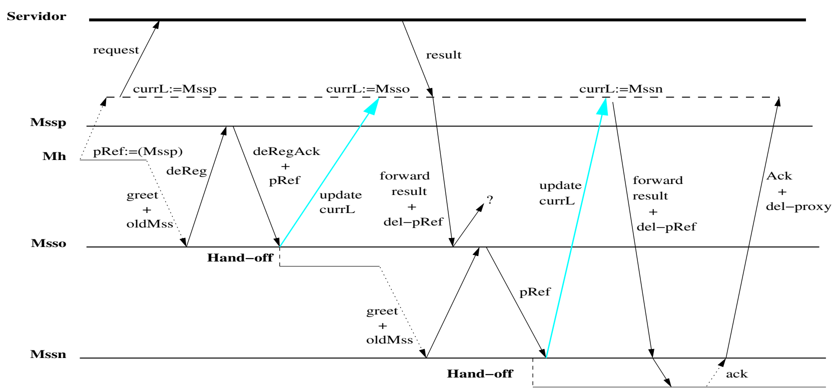

Figura 2: Cenário-Exemplo do RDP

3.2 SRDP

O SRDP(Safe Result Delivery Protocol) [Dur01] é uma extensão do protocolo RDP que busca manter as mesmas garantias de confiabilidade na entrega sem que a confiabilidade dos Mss's e da rede fixa sejam críticas para o funcionamento do protocolo.

3.2.1 Suposições

1. Cada *Mh* está associado a um único Mss;
2. Um *Mh* só deixa o sistema após confirmar o recebimento de todas mensagens;
3. O *Mh* detecta retransmissões.
4. Se um *Mh* está ativo, ele deve confirmar todas as mensagens recebidas do seu corrente *Mss*

3.2.2 Estrutura de Dados

1. Proxy - Representante do Mh na rede fixa.
2. Memento[EGV95] - Estrutura mantida pelo Mh que contém as informações mínimas necessárias para a reconstrução do proxy.

3.2.3 Especificação das Mensagens

A Tabela 2 especifica as mensagens do protocolo SRDP e seus respectivos significados.

Mensagens	Participantes	Significado
greet	Mh → Mss	Mudança para a célula controlada por Mss
request	Mh → Mss; Mss → Server	Envio de requisição a Server
result	Server → Mss	Envio de resposta a Mss
forwardResWl	Mss → Mh	Envio do resultado ao Mh
ack	Mh → Mss; Mss → Mss	Confirmação do recebimento de um resultado
forwardAck	Mss → Server	Encaminhamento de confirmação para Server

Tabela 2: Mensagens do SRDP

3.2.4 Funcionamento

Seu funcionamento é muito parecido com o do RDP diferenciando-se apenas em alguns detalhes que o torna tolerante a falhas na rede fixa. Uma característica peculiar desse protocolo é que, a cada migração, um novo proxy será criado caso uma nova requisição seja enviada pelo Mh e o proxy antigo é coletado pelo sistema. Para garantir que requisições pendentes não sejam perdidas, cada Mh mantém uma estrutura chamada memento com todas as informações necessárias para a reconstrução do proxy, dentre estas informações está a lista de requisições pendentes que são re-enviadas ao servidor logo após a criação do proxy.

A figura 3 exemplifica o funcionamento do SRDP. Este cenário mostra que, a cada migração, o Mh, através da mensagem *greet + memento*, reconstrói o proxy e re-envia todas as requisições pendentes.

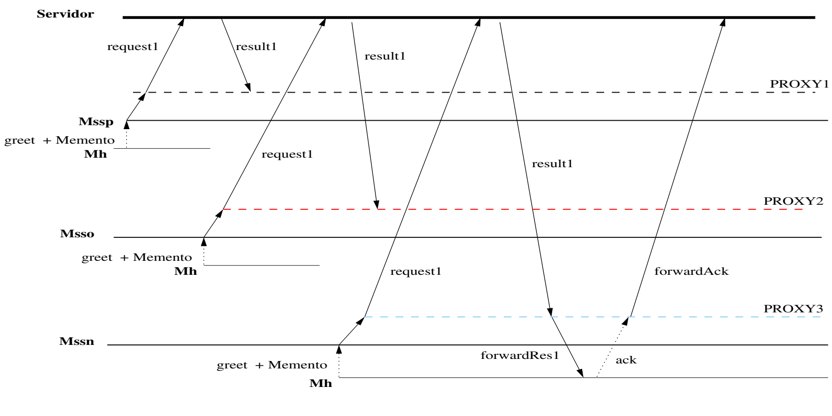

Figura 3: Cenário-Exemplo do SRDP

3.3 Mobile IP

O Mobile IP [JM96] é um protocolo desenvolvido pela IETF(Internet Engineering Task Force), o mais importante órgão de padronização de protocolos para a Internet. Este protocolo foi projetado para tornar possível o roteamento de pacotes IP's a unidades móveis utilizando-se de um endereço IP fixo.

A interconexão de redes torna-se cada vez mais uma realidade tanto no meio científico quanto no meio comercial. Prova disso é a Internet que inter-conecta milhões de redes de computadores por todo o mundo. No contexto da Internet, o protocolo IP tem essa importante função de integrar toda a estrutura.

O protocolo IP é que torna possível traçar um caminho de uma rede a outra (*roteamento*) e assim permitir a comunicação entre computadores. Isso só é possível com um esquema de endereçamento hierárquico no qual atribui-se um endereço a cada computador da rede. Através desse endereço é possível localizar um computador e a rede na qual está localizado.

A mobilidade torna esse esquema de endereçamento, da forma como foi concebido, inviável, pois uma unidade móvel pode mudar de endereço várias vezes, o que torna a mobilidade não transparente a localização.

Para buscar uma solução, um grupo de estudos do IETF criou o protocolo Mobile IP. Nesse protocolo, um computador móvel possui dois endereços: o endereço fixo que qualquer máquina possui numa rede IP, também chamado de *home address* e um endereço que muda a cada migração da unidade móvel, chamado *care-of address*. A rede fixa é composta por dois tipos de estações-base: o *home agent* que é a estação de origem de um Mh e o *foreign agent* que é a estação responsável por um Mh em um dado instante de tempo, mas que não é sua estação de origem.

Quando um pacote é direcionado a uma unidade móvel, utiliza-se o *home address* como endereço de destino. O pacote então chega na rede de destino e é recebido pelo *home agent*. Ao receber um pacote cujo destino é um Mh que não encontra-se na rede local, o *home agent*, através de sua tabela de roteamento, encaminha o pacote ao atual *foreign agent* do Mh. O *foreign agent* por sua vez, utilizará o *care-of address* do Mh para repassar-lhe o pacote.

Isso só é possível porque a cada migração do Mh, ocorre uma atualização da sua localização no *home agent* que agora deverá encaminhar qualquer pacote para o novo *foreign agent* daquele Mh.

Tomando como referência a arquitetura TCP/IP, o Mobile IP é classificado como um protocolo da camada de rede. Esta monografia propõe uma adaptação do protocolo para a camada de aplicação como forma de permitir uma comparação com os demais protocolos.

A adaptação resume-se a tratar mensagens(requisições e resultados) como pacotes. Os *home agents* e *foreign agents* são agora chamados de **origMss** e **respMss**, respectivamente. O funcionamento é semelhante, o Mh envia uma requisição a um servidor que retorna uma resposta ao origMss e este, através de sua tabela de rotas, encaminha a mensagem a atual localização do Mh(respMss). A migração do Mh implica na atualização da tabela de rotas do seu origMss.

A figura 4 ilustra a mecânica do protocolo Mobile IP adaptado.

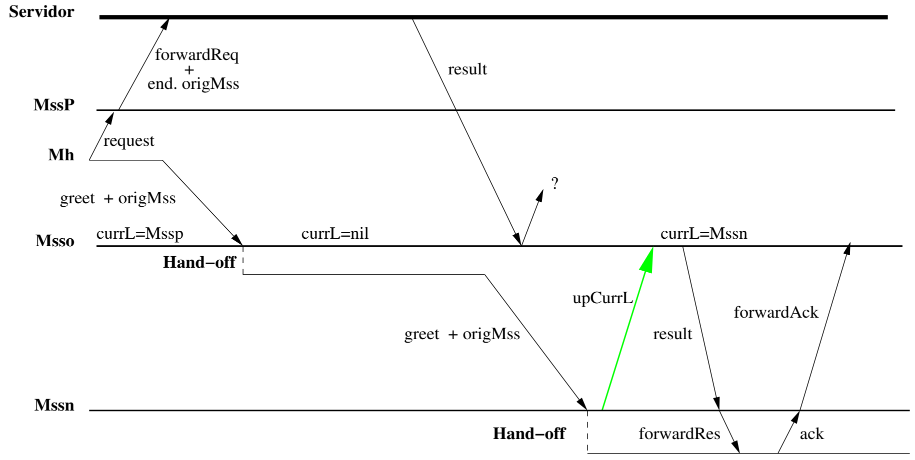

Figura 4: Cenário-Exemplo do Mobile IP

3.3.1 Especificação das Mensagens

A Tabela 3 mostra as mensagens trocadas entre duas máquinas do sistema.

Mensagens	Participantes	Significado
greet	Mh $\rightarrow$ Mss	Mudança para a célula controlada por Mss
request	Mh $\rightarrow$ Mss; Mss $\rightarrow$ Server	Envio de requisição a Server
result	Server $\rightarrow$ Mss	Envio de resposta a Mss
forwardRes	Mss $\rightarrow$ Mss	Envio do resultado ao atual respMss
forwardResWl	Mss $\rightarrow$ Mh	Envio do resultado ao Mh
updateCurrL	Mss $\rightarrow$ Mss	Atualizar a localização de um Mh no origMss
ack	Mh $\rightarrow$ Mss; Mss $\rightarrow$ Mss	Confirmação do recebimento de um resultado
forwardAck	Mss $\rightarrow$ Server	Encaminhamento de confirmação para Server

Tabela 3: Mensagens do MobileIP

3.4 Snapshot

O nome *snapshot* refere-se a um algoritmo de entrega de mensagens confiável que surgiu da adaptação da idéia de snapshots distribuídos descrita em [CL85]. O objetivo de algoritmos de snapshot(cuja a tradução literal é fotografia) é coletar o estado local dos canais e das máquinas para construir a consistência global do sistema. A técnica de entrega confiável de mensagens descrita em [MP00] utiliza as mesmas idéias. Porém, ao invés de distribuir o estado do sistema, é distribuída a mensagem a ser entregue. As seções a seguir descrevem o funcionamento do algoritmo.

3.4.1 Modelo de Sistema e Suposições

O modelo de sistema para este algoritmo possui algumas particularidades em relação ao modelo descrito na seção 2. Cada máquina fixa é chamada de nó. Os canais de comunicação entre nós são direcionados e possuem um estado associado: OPEN ou FLUSHED. Os canais em cada nó são classificados em **canais de entrada** e **canais de saída**. A migração dos Mh's, também chamados de agentes, deve atender a estrutura de canais do sistema. Assume-se que a rede é estruturada como um grafo conexo(existe um caminho entre dois quaisquer nós, mas não necessariamente um canal entre eles) e direcionado.

3.4.2 Estruturas de dados

Cada Mss possui um vetor de objetos representando todas as suas conexões(canais) com os outros Mss's do sistema. Os atributos de cada objeto são:

1. Nome ou endereço dos Mss ao qual está conectado;
2. Vetor de estados do canal com o identificador de todas as mensagens recebidas;
3. Tipo de canal
   (a) Entrada
   (b) Saída

Cada Mss possui também um buffer com as mensagens não entregues indexado pelo endereço dos Mh's destino.

3.4.3 Especificação das Mensagens

.

Mensagens	Participantes	Significado
greet	Mh → Mss	Mudança para a célula controlada por Mss
request	Mh → Mss; Mss → Server	Envio de requisição a Server
result	Servidor → Mss	Envio do resultado de uma requisição
broadResult	Mss → Mss	Envio do resultado através dos canais de saída
forwardResWl	Mss → Mh	Envio do resultado ao Mh
ack	Mh → Mss	Confirmação do recebimento de um resultado
deReg	Mss → Mss	Desregistro de um Mh

Tabela 4: Mensagens do Snapshot

3.4.4 Funcionamento

Inicialmente, todos os canais do sistema encontram-se no estado OPEN. Quando um Mss recebe uma mensagem, o canal de entrada através do qual ele recebeu essa mensagem tem seu estado mudado para FLUSHED, o que indica que qualquer Mh que siga a rota desse canal já recebeu a mensagem. Quando um Mss recebe pela primeira vez uma mensagem, ele armazena a mesma e tenta entregá-la ao Mh de destino caso este encontre-se na célula local. Num passo atômico, a mensagem é propagada por todos os canais de saída. Para todo Mh que chega por um canal no estado OPEN, o Mss tenta entregar as mensagens armazenadas caso seja o destino da mensagem. Assim que todos os canais de entrada de um Mss encontram-se no estado FLUSHED para uma determinada mensagem, essa é removida do buffer de mensagens, pois não há mais como um Mh chegar a este Mss sem ter recebido a mensagem.Desta forma garante-se que um Mh, em algum momento, receberá a mensagem, mesmo em situações de intensa migração.

Como forma de permitir uma comparação com os protocolos anteriores, a seção 3.4.3 especifica as mensagens necessárias ao funcionamento deste protocolo. Além disso, a figura 5 mostra um exemplo de funcionamento do protocolo, no qual o Mh emite uma requisição que é repassada ao servidor e esse, utilizando o protocolo de snapshot, entrega a resposta ao Mh correspondente.

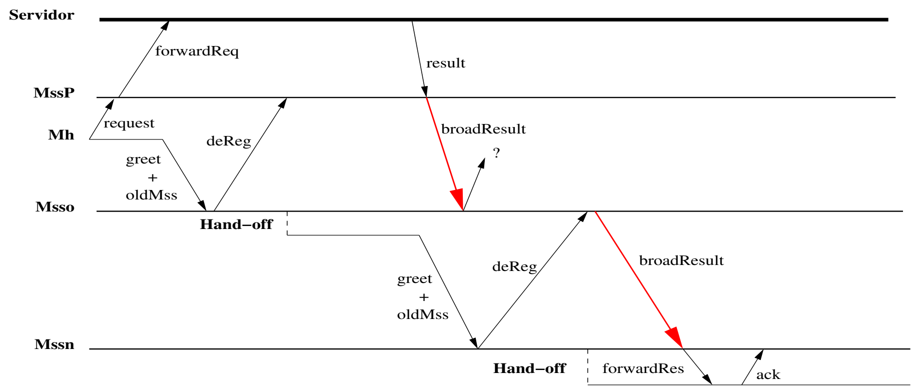

Figura 5: Cenário-Exemplo do Snapshot

4 Análise Comparativa

Esta seção analisa de maneira comparativa o funcionamento dos protocolos descritos nas seções anteriores. A comprovação dos fatos só poderá ser feita mediante simulação, já prevista para trabalhos futuros.

Todos os protocolos aqui descritos garantem, de alguma forma, a entrega de mensagens a estações móveis. Estes protocolos podem ser classificados, quanto a forma de entrega da mensagem, em duas categorias: protocolos de entrega por *encaminhamento* e entrega por *broadcast*. Os protocolos RDP, SRDP e Mobile IP encontram-se na primeira categoria, enquanto o protocolo de Snapshot na segunda. Analisando por este ponto de vista, em situações de pouca mobilidade nas quais a estação móvel permanece tempo suficiente para receber uma mensagem, os protocolos de encaminhamento mostram-se mais eficientes, pois produzem um menor tráfego levando a um menor congestionamento da rede.

Analisando os protocolos de encaminhamento, é possível observar que o RDP possui requisitos de confiabilidade muito fortes o que compromete a sua aplicação. No entanto, permite o balanceamento de carga do sistema, pois quando todas as requisições pendentes são atendidas, o proxy é removido e recriado na atual localização do Mh e dessa forma há uma diminuição na indireção causada muitas vezes pelo fato de o Mh encontrar-se em uma outra célula que não a do proxy.

O SRDP mostra avanços no que diz respeito aos requisitos de confiabilidade do sistema, pois garante a entrega de mensagens a unidades móveis mesmo em situações de falha da rede fixa ou do Mss. Porém, para situações em que a velocidade de migração é elevada e a migração muito intensa, o protocolo pode levar a uma sobrecarga do sistema, pois, nesse caso, as requisições pendentes seriam refeitas a cada migração do Mh, gerando muita informação repetida.

O Mobile IP comparado aos outros dois protocolos de encaminhamento possui um balanceamento de carga menor, pois a função de representante do Mh na rede fixa que era dada ao proxy, nos outros protocolos, é de responsabilidade do Mss que controla a célula de origem do Mh, chamado origMss. Toda mensagem destinada ao Mh é encaminhada ao origMss o que não permite um balanceamento de carga, pois concentra num ponto fixo do sistema todo o processo de comunicação com o Mh. Além disso, o Mobile IP possui requisitos de confiabilidade ainda mais fortes, pois em caso de falha do origMss não é mais possível estabelecer comunicação com o Mh.

Diferentemente dos outros protocolos, o Snapshot utiliza uma técnica que envolve contactar cada Mss do sistema pelo menos uma vez, gerando tráfego comparável a um broadcast. Porém para situações em que há uma latência de comunicação muito alta entre o Mh e seu representante na rede fixa(proxy ou origMss), os protocolos de encaminhamento possivelmente terão uma menor performance, além de produzirem um alto overhead em situações de alta mobilidade dos Mh's. Dessa forma, o snapshot é adequado para situações em que o Mh movimenta-se com alta frequência.

Como forma de permitir uma comparação entre os protocolos, tomou-se como referência o modelo de especificação e programação de protocolos descrito em [NTBC98]. Nesse modelo, um protocolo é composto por módulos de funcionalidade bem definida chamados *micro-protocolos*. Os micro-protocolos interagem através de eventos(mensagens ou eventos de tempo) e são compostos por um conjunto de métodos, chamados manipuladores de eventos que são executados em resposta a ocorrência de um evento.

Uma organização em micro-protocolos adequada ao ambiente de computação móvel, foi utilizada em [AB93]. Esta organização separa as funcionalidades distintas de um protocolo tanto quanto a mobilidade como no que se refere ao meio de comunicação. São três os micro-protocolos desta organização:

- **Wired:** Controla todas as mensagens trocadas entre estações fixas.
- **Wireless:** Responsável pelas mensagens trocadas pelo meio sem fio.
- **Handoff:** Responsável pelas mensagens que controlam a migração de estações móveis.

Seguindo essa organização, a tabela 5 mostra todas as mensagens dos protocolos aqui descritos. As mensagens `result`, `fowardReq`, `ack`, `request` e `forwardResWl` são comuns a todos os protocolos.

A mensagem `greet`, apesar de estar presente em todos os protocolos, possui parâmetros diferenciados para cada um. As mensagens restantes são específicas de alguns protocolos como mostrado na tabela.

Tabela 5: Classificação das mensagens quanto aos Micro-protocolos

Wired	wireless	Handoff
result	ack	greet
broadResult(Ss)	request	deReg(R,Ss)
forwardRes(R,M)	forwardResWl	deRegAck(R)
forwardAck(R,S,M)		updateCurrL(R,M)

Legenda:

R → RDP
S → SRDP
M → Mobile IP
Ss → Snapshot

5 O Arcabouço

Esta seção descreve uma proposta de Arcabouço para a construção de protocolos de entrega de mensagens como os discutidos na seção anterior. A especificação do Arcabouço toma como referência um simulador de protocolos para redes móveis chamado Mobics [dR01] que é descrito em linhas gerais a seguir.

5.1 Visão Geral

O modelo de programação de protocolos do simulador Mobics é baseado no conceito de *micro-protocolos* [NTBC98]. Os micro-protocolos são entidades que implementam uma funcionalidade bem definida. Um protocolo é uma estrutura composta por um conjunto desses micro-protocolos e que descreve uma sequência de *eventos* trocados entre esses micro-protocolos. Um evento corresponde ao envio de uma mensagem por um micro-protocolo.

Cada micro-protocolo é composto por um conjunto de métodos responsáveis por executar todas as operações necessárias ao correto tratamento da ocorrência de um evento, chamados *manipuladores de eventos*. Para cada mensagem de um protocolo deve existir um método manipulador de evento correspondente.

Baseado na análise feita no Capítulo 4 à classe de protocolos de entrega de mensagens é possível observar que muitas mensagens são comuns a maior parte desses protocolos. Dessa forma, o principal objetivo do Arcabouço é reunir todas as mensagens e métodos manipuladores de eventos comuns a essa classe de protocolos a fim de permitir o máximo reuso de código e, por conseguinte, diminuir o esforço gasto na prototipagem de protocolos.

5.2 Arquitetura do Sistema

A arquitetura do sistema propõe uma estrutura genérica para facilitar a inserção de novas implementações comuns a essa classe de protocolos o que facilitará ainda mais a simulação de protocolos.

Devido a generalidade da arquitetura, ela pode servir como referência para a construção de novos arcabouços para outras classes de protocolos tais como protocolos de multicast.

A implementação convencional de um protocolo no Mobics sem o uso do arcabouço possui a estrutura mostrada na figura 6. Um protocolo é construído instanciando ou estendendo classes do simulador e implementantando algumas interfaces do Mobics.

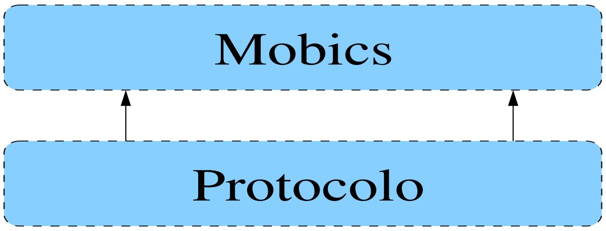

Figura 6: Interação padrão com o Mobics

Com a criação do Arcabouço, uma nova camada é inserida na estrutura, agora o código dos protocolos estende ou instancia classes do Arcabouço que por sua vez estende e instancia classes do simulador. Em algumas situações específicas de cada protocolo pode ser necessário estender ou instanciar classes diretamente do simulador. A figura 7 ilustra a nova estrutura.

Figura 7: Novo modelo de interação

A figura 8 mostra um diagrama com a estrurura de classes resultante da inserção das classes do Arcabouço no modelo de programação original do mobics. No diagrama estão destacados os três componentes que formam a nova estrutura de programação de protocolos: O Mobics, o Arcabouço e o Protocolo.

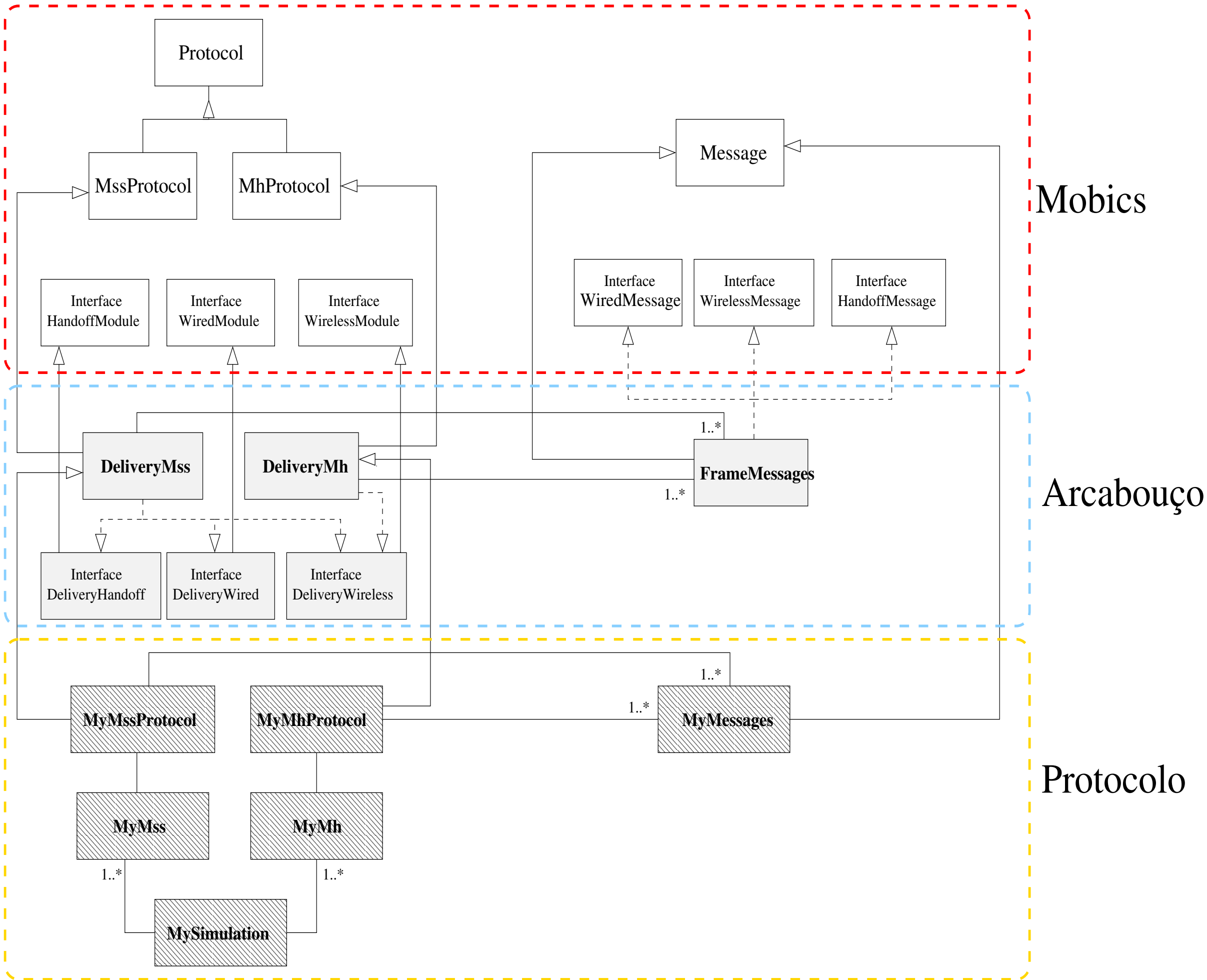

Figura 8: Diagrama de Classes do Sistema

5.2.1 Descrição do Arcabouço

Baseado no diagrama de classes do Arcabouço mostrado na figura 8, esta seção descreve de maneira detalhada os principais componentes do Arcabouço, são eles:

1. **Interfaces** - Especificam o conjunto de métodos manipuladores de eventos que devem ser implementados pelo programador de protocolos e também aqueles implementados pelo arcabouço. Parte mais importante do Arcabouço em termos de reuso de projeto, pois definem os componentes genéricos que podem ser reaplicados para implementar os mais variados protocolos de entrega de mensagens. As interfaces são:

   (a) <u>IDeliveryHandOff</u> - Especifica as mensagens trocadas entre o Mh e o Mss durante a mudança de célula.

   (b) IDeliveryWired - Mensagens trocadas entre os elementos de rede no meio com fio.
   (c) IDeliveryWireless - Mensagens trocadas entre os elementos de rede no meio sem fio.

2. **Classes** - Contém todos os métodos que realizam o tratamento de eventos. Tais métodos correspondem àquelas mensagens cujo tratamento é semelhante para todo o conjunto de protocolos. Componente fundamental no reuso de código. São elas:

   (a) DeliveryMss - Métodos manipuladores de eventos correspondente às mensagens que um Mss pode receber de um outro elemento de rede.
   (b) DeliveryMh - Métodos manipuladores de eventos correspondente às mensagens que um Mh pode receber de um outro elemento de rede.

3. **Mensagens** - Especificação da estrutura interna das mensagens. Uma mensagem no Mobics é representada por uma classe que contém a especificação dos campos da mensagem e os métodos de acesso a esses campos. Também importante no reuso de projeto, pois especifica as mensagens comuns à classe de protocolos. São elas:

   (a) Ack_Mh - Confirmação de que o Mh recebeu o resultado de uma requisição.
   (b) Dereg - De-registro de um Mh em um Mss
   (c) ForwardRes_Mh - Resultado de uma requisição encaminhada de um Mss_i a um Mh que se encontra na celula controlada por Mss_i.
   (d) ForwardRes_Mss - Resultado de uma requisição encaminhada por um Mss_i a um Mss_j que controla a célula onde se encontra o Mh de destino do resultado.
   (e) Req_server - Requisição encaminhada do MSS ao servidor.
   (f) Req - Requisição enviada de um MH ao seu MSS responsável.
   (g) Request - Requisição feita por uma aplicação.
   (h) Result - Resultado da requisição.

6 Conclusão

A construção de protocolos é uma tarefa muito custosa e, em muitos casos, repetitiva, pois para um conjunto de protocolos com mesma finalidade, tal como os protocolos de entrega de mensagens, as operações a serem realizadas no processo de comunicação são muito parecidas. Visando diminuir o tempo gasto na implementação de protocolos de entrega de mensagens, através do reuso de código e de projeto, esta monografia propôs um Arcabouço que contém toda a funcionalidade comum a essa classe de protocolos. Além disso, propôs uma arquitetura genérica que permite a extensão do Arcabouço para outras classes de protocolos de redes móveis.

Referências

[AB93] Arup Acharya and B.R. Badrinath. Delivering multicast messages in networks with mobile hosts. In 13th International Conference on Distributed Computing Systems, pages 292–299, Pittsburgh, US, May 1993.

[BBIM93] B. R. Badrinath, A. Bakre, T. Imielinski, and R. Marantz. Handling mobile clients: A case for indirect interaction. In 4th Workshop on Workstation Operating Systems (WWOS-IV), pages 91–97, Napa, U.S., 1993.

[Bha96] Nina Trappe Bhatti. A System for Constructing Configurable High-Level Protocols. PhD thesis, University of Arizona, December 1996.

[CL85] K. M. Chandy and L. Lamport. Distributed snapshots: Determing global states of distributed systems. In Trans. on Computer Systems, volume 3(1), pages 63–75. ACM, Frebuary 1985.

[dR01] Ricardo Couto Antunes da Rocha. Um arquitetura para simulação flexível de protocolos para computação móvel. PhD thesis, Universidade de Sao Paulo, Maio 2001.

[Dur01] Alan Mitchell Durham. A conectionless protocol for mobile agents. Relatório Técnico, Janeiro 2001.

[EGV95] Ralph Johnson Erich Gamma, Richard Helm and John Vlissides. Design Patterns: Elements of Reusable Softaware. Addison-Wesley, 1995.

[ESO00] M. Endler, D. Silva, and K. Okuda. Rdp: A result delivery protocol for mobile computing. In Int. Workshop on Wireless Networks and Mobile Computing (WNMC) with 20th ICDCS, Taiwan, April 2000. R.O.C. IEEE.

[ESS+00] M. Endler, D. Silva, F. Silva, R. Rocha, and M. de Moura. Project sidam: Overview and preliminary results, May 2000.

[Gol84] Adele Goldberg. Smalltalk-80: The Interactive Programming Environment. Addison-Wesley, 1984.

[JM96] David B. Johnson and David A. Maltz. Protocols for adaptive wireless and mobile networking. In IEEE Personal Communications, 3(1), Frebuary 1996.

[Mat96] Michael Mattsson. Object-Oriented Frameworks - A survey of methodological issues. PhD thesis, University College of Karlskrona/Ronneby, 1996.

[MEFJ99] Douglas C. Schmidt Mohamed E. Fayad and Ralph E. Johnson. Building Application Frameworks. Wiley Computer Publishing, 1999.

[ML98] Geraldo Robson Mateus and Antonio Alfredo Ferreira Loureiro. Introdução à Computação Móvel. 11a Escola de Computação, COPPE/Sistemas,NCE/UFRJ, 1998.

[MP00] Amy L. Murphy and Gian P. Picco. Reliable communication for highly mobile agents. Frebuary 2000.

[NTBC98] Richard D. Schlichting Nina T. Bhatti, Matti A. Hiltunen and Wanda Chiu. Coyote: a system for constructing fine-grain configurable communication services. In Transaction on Computer Systems, volume 16(4), pages 321–366. ACM, November 1998.