

Este capítulo descreve uma instanciação dos *frameworks para provisão de QoS em redes móveis sem fio* através de um cenário no qual é simulada uma rede infra-estruturada de serviços integrados funcionando sobre o IP Móvel. Para validar esta proposta, foi implementado um protocolo de reserva de recursos simplificado denominado SMQoS (Simple Mobile QoS Protocol), baseado nos protocolos MRSVP, MQoS e RSVP-A, que oferece suporte aos conceitos de reservas antecipadas e intervalos de QoS. Esses conceitos foram incorporados, respectivamente, pela avaliação da especificação de mobilidade do usuário, obtida através de uma instanciação do *framework para gerenciamento de mobilidade*, e pela representação dos parâmetros que definem a QoS através de valores limítrofes.

4.1 AMBIENTE DE IMPLEMENTAÇÃO

Por não contarmos com uma rede móvel sem fio, de porte, para pesquisa e desenvolvimento, foi utilizado um ambiente simulado criado com o auxílio do MobiCS (Rocha & Endler, 2000), um simulador de protocolos distribuídos, implementado como uma biblioteca Java. Para realizar as simulações, foi utilizada uma estação Pentium III, com 256 Mb de RAM e sistema operacional Microsoft Windows 2000.

As restrições impostas pela utilização de um ambiente simulado induziram, para efeito de aplicação dos *frameworks para provisão de QoS em redes móveis sem fio*, à implementação de um protótipo que oferecesse somente um subconjunto dos serviços definidos na arquitetura apresentada no Capítulo 3. Este protótipo, implementado em Java utilizando o MobiCS, simula um protocolo de reserva de recursos simplificado para ambientes móveis que utiliza intervalos de QoS e oferece suporte a reservas imediatas e antecipadas, com base na predição dos deslocamentos dos nós. O protótipo é constituído pelos seguintes elementos: (i) um protocolo de reservas de recursos que oferece suporte à mobilidade, estendendo a instanciação de uma versão simplificada do IP Móvel desenvolvida no MobiCS (Mele & Rolins, 2001) e (ii) um gerenciador de mobilidade centralizado, com o agente local desempenhando o papel de gerente de mobilidade.

4.2 O SIMULADOR MOBICS

O MobiCS (*Mobile Computing Simulator*) (Rocha & Endler, 2000) é uma ferramenta integrada para prototipagem, teste, validação e avaliação de protocolos distribuídos para computação móvel. O simulador facilita a programação de protótipos de protocolos, tendo como base um modelo de programação modular e orientado a eventos, baseado em abstrações de programação de alto nível e transparência total de simulação.

A versão atual do MobiCS contempla apenas as redes móveis infra-estruturadas, não oferecendo suporte às redes móveis *ad hoc*³⁵. Para oferecer suporte aos ambientes infra-estruturadas o MobiCS implementa abstrações como: nó móvel (*Mobile Host – MH*), estação de suporte à mobilidade (*Mobility Support Station – MSS*), nó fixo (*Fixed Host – FH*), célula, enlaces com fio e sem fio, componente *handoff*, entre outras (Figura 4.1).

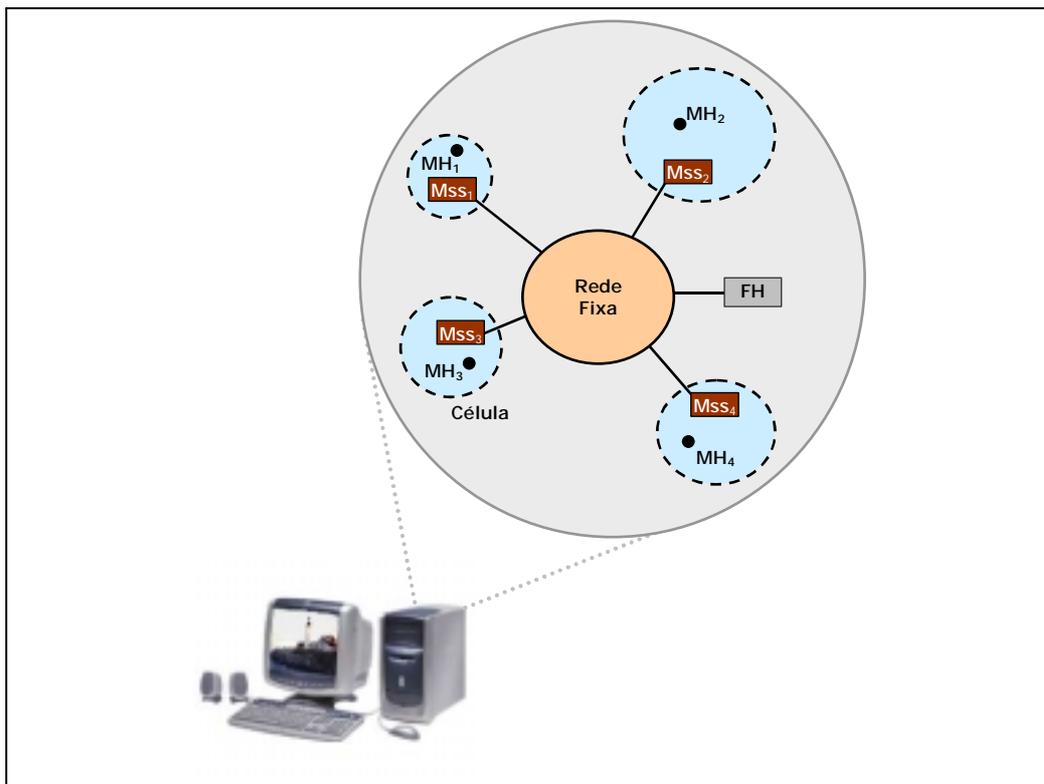


Figura 4.1 Simulando uma rede infra-estruturada com o MobiCS

³⁵ Um novo pacote MobiCS está sendo desenvolvido para permitir a prototipação e simulação de protocolos para redes móveis *ad hoc* (Mele & Endler, 2002).

Como o MobiCS foi desenvolvido como um *framework*, para criar uma simulação o usuário basicamente deve implementar as classes que definem os protocolos a serem simulados, o ambiente de rede e o modelo de simulação desejado (Figura 4.2). Essas classes estendem as classes básicas do MobiCS, que implementam as funcionalidades das várias camadas de software do simulador. A linguagem de programação Java é utilizada para o desenvolvimento dos protótipos dos protocolos e para a criação das simulações, o que facilita a portabilidade e extensão do simulador. Do ponto de vista do usuário MobiCS, há dois pacotes Java principais que são utilizados para criar uma simulação. Com o pacote `mobics.ppi` o usuário programa protocolos que podem ser simulados, fazendo uso de dois sub-pacotes: `mobics.ppi.protocol`, que declara as classes necessárias à implementação do protocolo propriamente dito; e `mobics.ppi.message`, que é utilizado para a declaração de mensagens utilizadas pelo protocolo. Com o pacote `mobics.simulation`, o usuário implementa a simulação propriamente dita, em termos de número e tipos de elementos de rede envolvidos na simulação, topologia da rede, tipos de protocolos a serem simulados, modo e modelo de simulação.

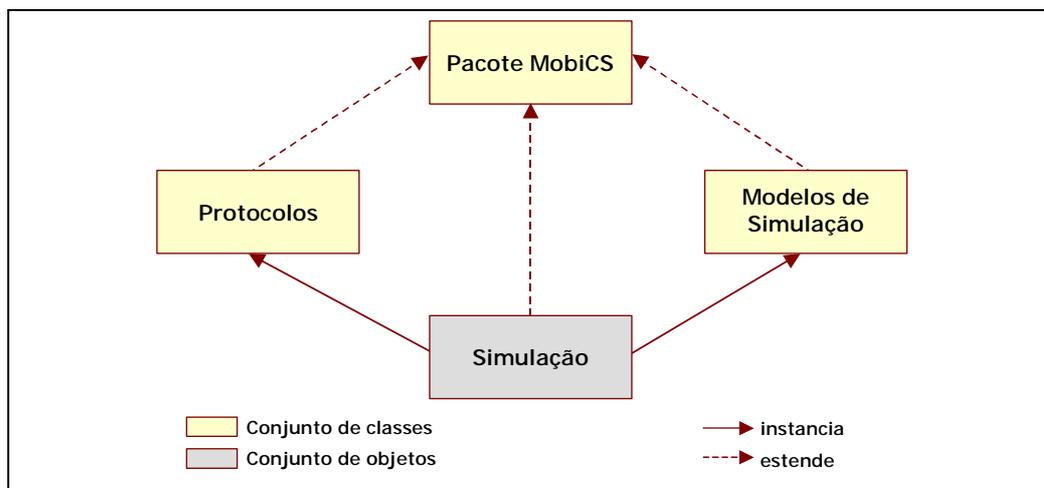


Figura 4.2 Implementação de protocolos e simulações a partir do MobiCS

No MobiCS é adotado um modelo de programação composto, no qual os protocolos são constituídos por conjuntos de módulos chamados de *micro-protocolos*, com o qual é possível descrever protocolos altamente modularizados e organizados em componentes funcionais, que interagem através de eventos. O MobiCS induz a organização dos protocolos baseada em três componentes funcionais:

- *Wired*: responsável pela troca de mensagens pela rede fixa.

- *Wireless*: responsável pela troca de mensagens pelos canais de comunicação sem fio.
- *Handoff*: responsável pela comunicação entre os elementos de rede durante o processo de *handoff*, podendo incluir o envio de mensagens pela rede fixa e pelo canal sem fio.

O MobiCS implementa dois modos de simulação: determinístico e estocástico. No modo determinístico o usuário define um *script* de simulação descrevendo um cenário particular, que é executado passo a passo, permitindo ao usuário observar o estado do protocolo ou do objeto de rede. No modo estocástico o usuário atribui padrões de comportamento probabilístico para os elementos simulados da rede. As simulações determinísticas são utilizadas para teste e avaliação da correção de protocolos, já as simulações estocásticas são utilizadas para testar a sua performance. Uma das principais contribuições do MobiCS é a implementação de uma arquitetura de software para simuladores que possibilita a troca do modo de simulação sem afetar o protocolo que está sendo prototipado, ou seja, o mesmo protocolo pode ser testado em diferentes modos de simulação sem a necessidade de alteração do seu código.

4.3 SIMPLIFICAÇÕES ADOTADAS

A instanciação dos *frameworks para provisão de QoS em redes móveis sem fio* tem como objetivo validar a modelagem proposta e analisar o comportamento do mecanismo de reservas de recursos levando em consideração a mobilidade dos nós. Foi efetuada uma série de simplificações em relação à instanciação dos *frameworks*, para viabilizar a implementação do cenário-exemplo:

- A implementação do IP Móvel que serve de base para a instanciação do SMQoSP trabalha no modo de roteamento triangular, fazendo algumas simplificações³⁶ ao modelo proposto por Perkins (1996a).

³⁶ Maiores detalhes sobre as simplificações adotadas na implementação do IP Móvel podem ser obtidos na especificação do trabalho (Endler, 2001).

- No estabelecimento de uma reserva, o nó fixo será sempre a origem (transmissor) e o nó móvel será sempre o destino (receptor).
- As estações móveis têm acesso irrestrito a todas as células definidas no cenário, ou seja, as políticas de controle não são utilizadas para validar o acesso das estações móveis nos diferentes domínios administrativos.
- O recurso considerado é a largura de banda.
- As mensagens de sinalização foram simplificadas, contendo apenas informações referentes ao endereço do transmissor e do receptor, o identificador da reserva, a quantidade de recursos (largura de banda) solicitada (intervalos de QoS) e a quantidade de recursos disponível.
- Nesse cenário, assim como no protocolo MQoS, somente são consideradas reservas nas estações base (Mss).
- O cenário não oferece suporte a *multicast*.

4.4 DESCRIÇÃO DO CENÁRIO-EXEMPLO

No MobiCS, a abstração utilizada para representar a interação entre os protocolos e as máquinas simuladas³⁷ são as mensagens. A Tabela 4.1 descreve as possíveis mensagens utilizadas pelo SMQoS durante a sua execução, indicando os elementos envolvidos e o significado de cada mensagem:

MENSAGEM	PARTICIPANTES	SIGNIFICADO
Request	Mh → Mss Mss → Server	O nó móvel (Mh) solicita um serviço com QoS para um servidor (nó fixo) através de um Mss
SMQoSRequest	Mh → Mss(HA)	O nó móvel (Mh) solicita ao seu HA (Mss) que efetue reservas antecipadas em seu benefício nas possíveis localidades para as quais possa se deslocar (MSPEC)
SMQoSResponse	Mss(HA) → Mh	O HA (Mss) informa ao nó móvel (Mh) o resultado das solicitações de reserva que ele efetuou em seu nome

³⁷ O termo máquina simulada denota qualquer elemento de rede simulado que é capaz de incorporar protocolos e processar mensagens, como estações móveis e máquinas fixas.

MENSAGEM	PARTICIPANTES	SIGNIFICADO
SMQoSPRelease	Mh → Mss(HA)	O nó móvel (Mh) informa ao HA (Mss) que as suas reservas podem ser liberadas
Path	Server → Mss(HA) Mss(HA) → Mh	Mensagem enviada pelo servidor ao nó móvel (Mh) através do HA (Mss) informando a quantidade de recursos disponíveis
PathA	Mss(HA) → Mss(FA)	Mensagem enviada pelo HA (Mss) ao FA (Mss) indicando que uma reserva antecipada deve ser efetuada, informando o identificador da reserva e a especificação do fluxo do nó móvel (Mh)
Path-Tear	Mss(HA) → Mss(FA)	Mensagem enviada pelo HA (Mss) a todos os FAs (Mss) contidos no MSPEC do nó móvel (Mh) e ao servidor (nó fixo) informando que os recursos reservados devem ser liberados
Resv	Mh → Mss Mss → Server	Mensagem enviada pelo nó móvel (Mh) ao servidor (nó fixo) através de um Mss efetuando a reserva de recursos imediata
ResvA	Mss(FA) → Mss(HA)	Mensagem enviada pelo FA (Mss) ao HA (Mss) efetuando a reserva de recursos antecipada
Resv-Tear	Mss(HA) → Server	Mensagem enviada pelo HA (Mss) ao servidor informando que os recursos reservados devem ser liberados

Tabela 4.1 Mensagens utilizadas pelo protocolo SMQoSP

Assim como no protocolo MQoSP, foram definidas mensagens especiais no SMQoSP que são trocadas entre o nó móvel e o seu agente local para que este efetue reservas antecipadas em seu benefício. As reservas antecipadas não possuem parâmetros temporais³⁸, como as mensagens do MQoSP, seguindo o modelo de reservas passivas e ativas do MRSVP. Essa limitação se deve ao fato do ambiente de simulação não oferecer mecanismos para obtenção do tempo simulado³⁹.

Ao receber uma mensagem *SMQoSPRequest* o agente de mobilidade verifica se é o agente local de mobilidade do nó móvel. Em caso afirmativo, ele aciona o mecanismo de predição do gerenciador de mobilidade obtendo a especificação de mobilidade do nó, que será utilizada na solicitação de reservas antecipadas. Em caso

³⁸ Esses parâmetros definiriam o instante de tempo em que os recursos seriam utilizados e o período durante o qual eles permaneceriam alocados.

³⁹ O MobiCS utiliza o conceito de unidades de tempo simulado no modo de simulação estocástico e obedece a uma seqüência de execução de eventos, passos de simulação, no modo de simulação determinístico.

negativo, o agente de mobilidade encaminhará a mensagem para o agente local do nó móvel.

A cada deslocamento do nó móvel, o agente local é informado durante o processo de atualização de *binding* do IP Móvel, sobre a nova localização do nó, atualizando o seu histórico de mobilidade.

O SMQoSP foi implementado seguindo o modelo de programação de protocolos do MobiCS, utilizando a abordagem de definição de micro-protocolos. O protocolo SMQoSP possui três instâncias funcionais, uma para estações base (SMQoSPMss), uma para estações móveis (SMQoSPMh) e uma para estações fixas (SMQoSPServer). Essas instâncias representam os elementos envolvidos no processo de reserva de recursos. A Figura 4.3 ilustra de forma compacta os tipos de mensagens que podem ser trocados entre esses elementos no SMQoSP. Como pode ser observado, grande parte das mensagens são tratadas por instâncias de estações base, nas quais se concentra a implementação do algoritmo do protocolo.

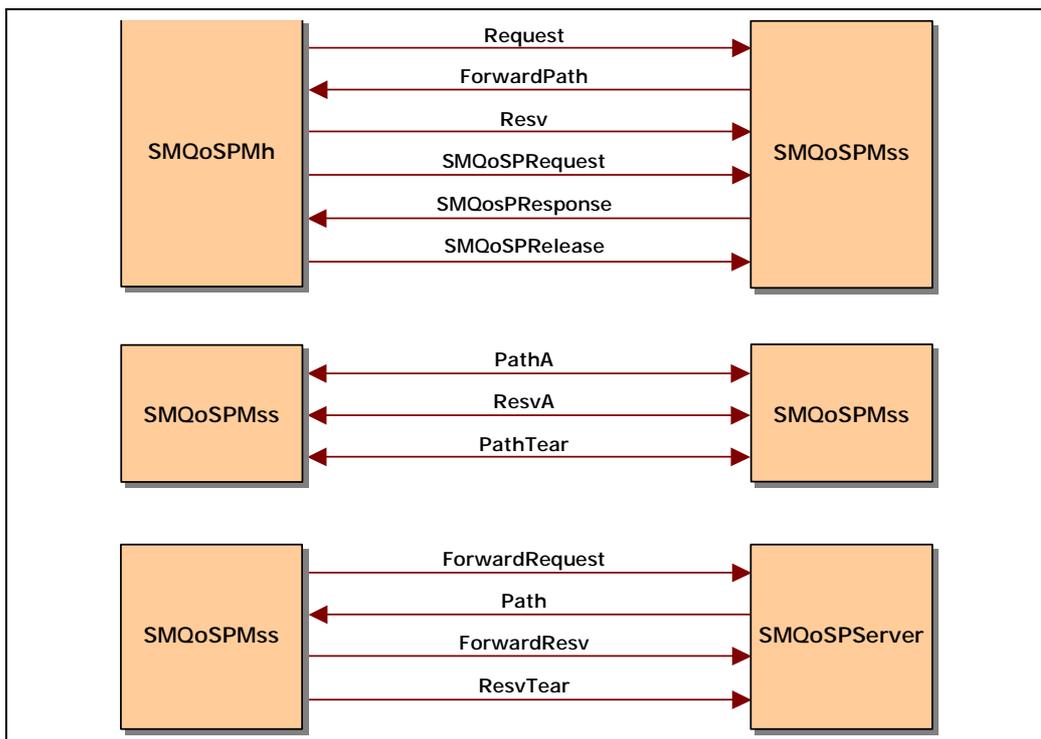


Figura 4.3 Comunicação entre os módulos do SMQoSP

A Figura 4.4 ilustra a troca de mensagens entre os micro-protocolos (*Wireless*, *Handoff* e *Wired*) que constituem as instâncias do protocolo de reserva de recursos

simplificado SMQoSP. Como se pode observar, inicialmente uma estação móvel (MH) solicita o serviço ao servidor (CN⁴⁰) (1) e esse pedido será encaminhado através do agente local (HA) da estação móvel (1'). Em resposta à solicitação de serviços do MH, o CN encaminha uma mensagem *Path* (2), que será entregue ao MH através do seu HA (2'). Ao receber uma mensagem *Path*, o MH transmite uma mensagem *Resv* (3) que será encaminhada através do HA para o CN (3'). O MH envia uma mensagem *SMQoSP-Request* ao HA solicitando que este efetue reservas antecipadas em seu benefício (4). Ao receber essa mensagem, que contém informações como a identificação da sessão, a classe de serviço e os parâmetros requisitados pelo serviço, o HA acionará o gerenciador de mobilidade para obter, através do mecanismo de predição, a especificação de mobilidade do usuário. O HA encaminha mensagens *Path-A* para os agentes estrangeiros (FAs) que constam na especificação de mobilidade do MH (5). Essa mensagem é uma adaptação da mensagem *Path* do RSVP tradicional indicando que a reserva será efetuada de forma antecipada. Os FAs respondem ao HA enviando mensagens *Resv-A* para efetivar a reserva (6). O HA envia ao MH o resultado das solicitações de reservas antecipadas de recursos através de uma mensagem *SMQoSP-Response* (7).

Como o gerenciamento do estado das reservas é *soft-state*, os recursos reservados são automaticamente liberados quando as informações sobre o estado da reserva e o seu caminho não forem atualizadas dentro de um período de tempo específico. Entretanto, como ilustrado na Figura 4.4, o MH pode finalizar voluntariamente as suas sessões solicitando ao HA a liberação de todas as suas reservas (ativas e passivas) através de uma mensagem *SMQoSP-Release* (8). O HA se encarrega, então, de liberar os recursos que foram reservados enviando mensagens *Path-Tear* para os FAs contidos na especificação de mobilidade do MH e mensagens *Resv-Tear* para o CN, que é o responsável pela provisão do serviço (8').

⁴⁰ Na terminologia do IP Móvel as estações fixas são designadas por CN (*Correspondent Node*).

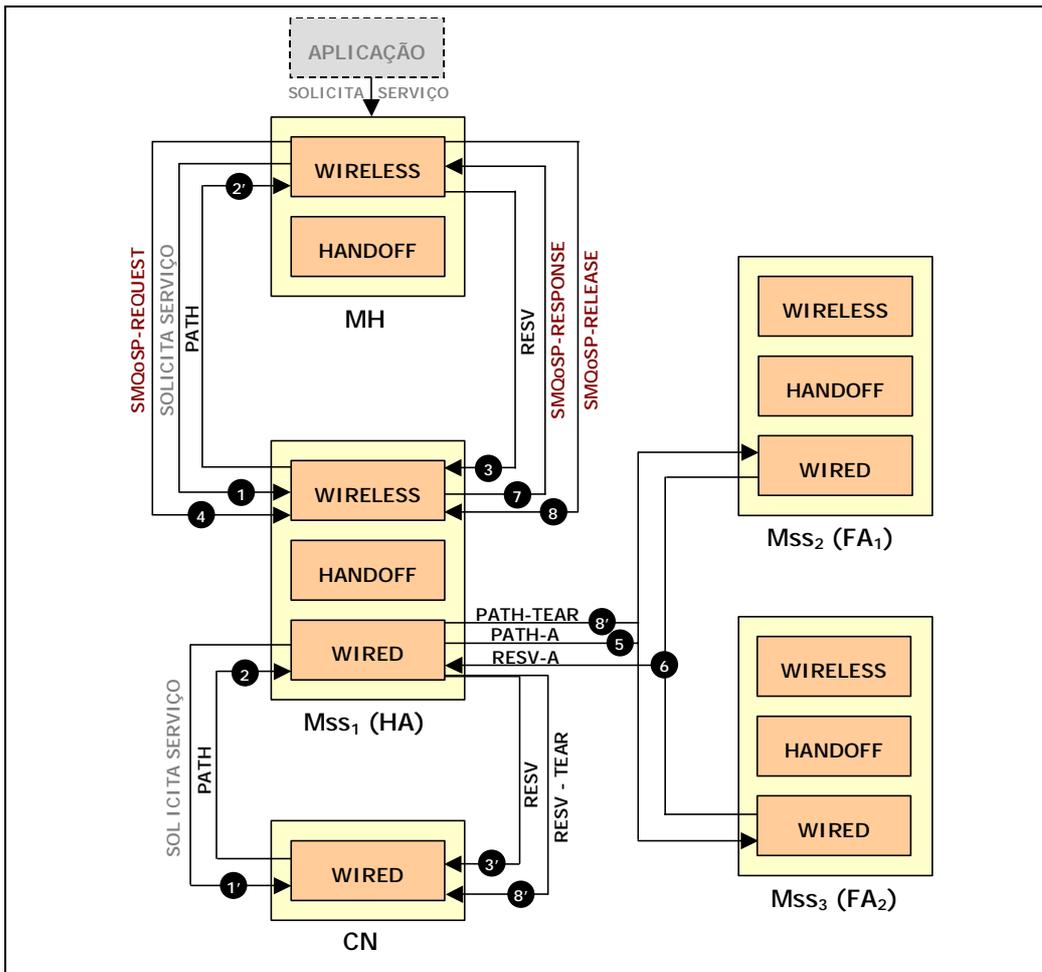


Figura 4.4 Definição das mensagens trocadas entre os micro-protocolos do SMQoS-SP

4.5 INSTANCIANDO OS FRAMEWORKS PARA PROVISÃO DE QoS EM REDES MÓVEIS SEM FIO

4.5.1 DEFININDO OS EVENTOS DE QoS

O estabelecimento de reservas de recursos em um ambiente de serviços integrados utilizando o protocolo de sinalização RSVP ou, como no cenário-exemplo apresentado, uma extensão a esse protocolo, o SRSVP-A (*Simple ReSerVation Protocol in Advance*), é precedido de um ou mais anúncios de tráfego por parte dos transmissores (mensagens *Path*), onde serão informadas aos receptores as características associadas ao tipo de fluxo que o transmissor pode gerar. Os receptores, por sua vez, respondem enviando a especificação do fluxo que desejam receber (mensagens *Resv*). Essa troca de mensagens estabelecerá uma *sessão de QoS* identificada na simulação apresentada neste capítulo, por simplificação, através de um

identificador global e unívoco, constituído pela combinação do endereço do solicitante do serviço e uma seqüência numérica gerada por ele.

Todas as mensagens trocadas entre transmissores e receptores são visíveis à aplicação através do envio de notificações de *eventos de QoS⁴¹* que estão associados a ocorrências relacionadas ao processo de negociação entre a estação e o sistema de comunicação, podendo notificar o recebimento de mensagens de anúncio de tráfego, de solicitação, de admissão ou rejeição de um serviço, a ruptura de um contrato estabelecido, entre outras. A Figura 4.5 ilustra os eventos de QoS que foram contemplados na simulação descrita neste capítulo.

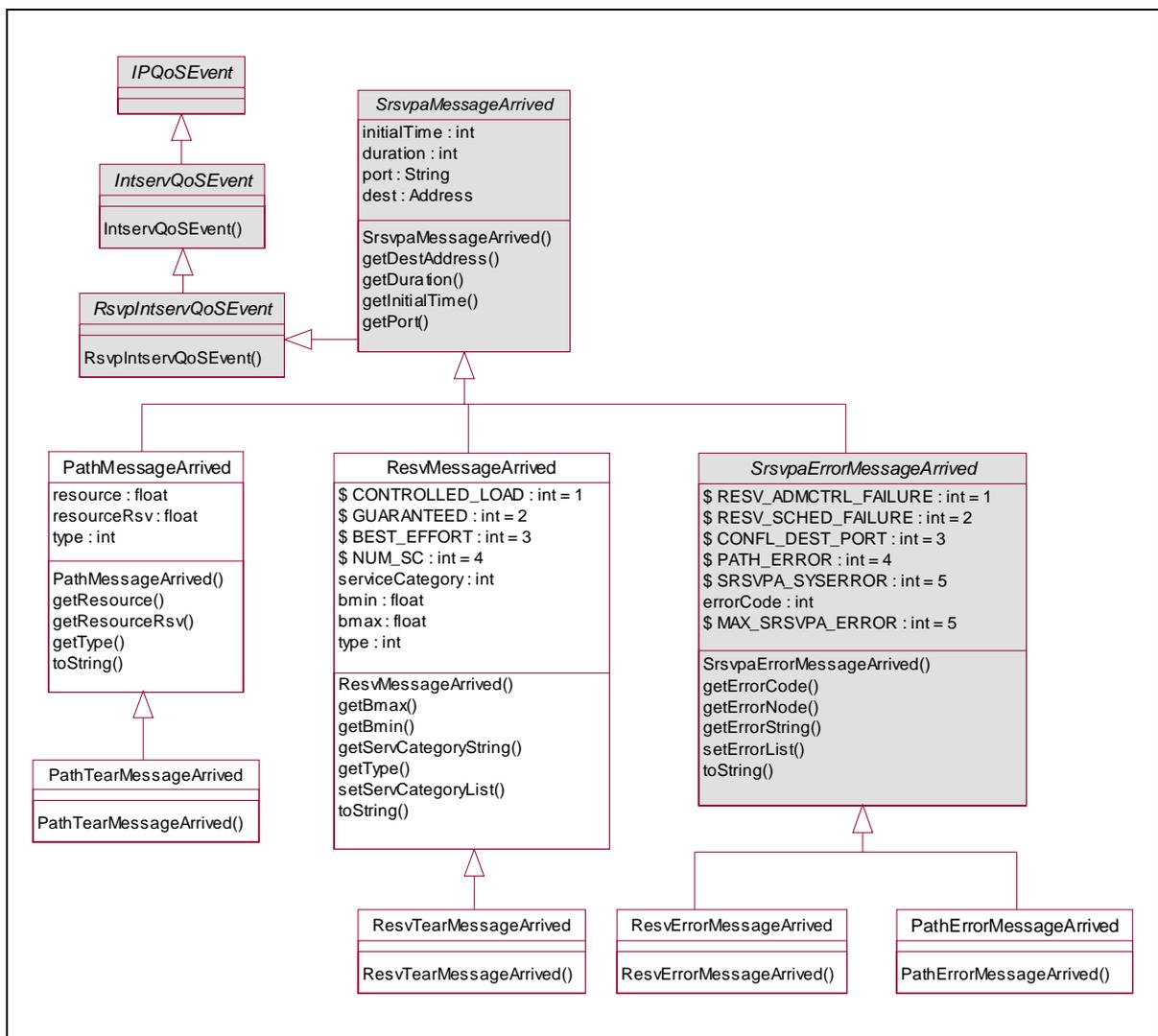


Figura 4.5 Eventos de QoS gerados pelo SRSVP-A

⁴¹ Os eventos de QoS estão associados aos tipos de mensagens trocadas pelo protocolo de sinalização.

4.5.2 INCORPORANDO O GERENCIADOR DE MOBILIDADE AO SMQOSP

O gerenciador de mobilidade foi incorporado ao protocolo de estabelecimento de reservas, sendo acionado internamente nos tratadores de eventos que correspondem aos métodos definidos na classe que representa uma instância funcional do protocolo SMQOSP para as estações base (classe *SMQoSPMss*), como apresentado na Subseção 4.5.4. Para representar as funcionalidades do gerenciador de mobilidade foram definidos os métodos `reqMspec()` e `updateHistory()`, responsáveis, respectivamente, pelos mecanismos de predição de mobilidade e atualização do histórico de mobilidade.

No MobiCS, a única forma de interação entre os protocolos é através do método `Protocol.send()`, que permite o envio de mensagens entre as máquinas simuladas. Este método é assíncrono e envia a mensagem pela rede simulada, que inclui interfaces de rede, canais de comunicação e outras máquinas, se necessário. A mensagem chega ao destinatário por uma primitiva `receive()`, com a qual ela é incluída na fila de mensagens da máquina simulada. Por esse motivo, as *classes para comunicação* definidas na Seção 3.4 foram implementadas através da utilização de chamadas a `Protocol.send()` e `receive()`.

4.5.3 DEFININDO AS MENSAGENS DO SMQOSP

A Figura 4.6 ilustra o diagrama de classes das mensagens que foram definidas para o funcionamento do protocolo SMQOSP. Cada mensagem no MobiCS está associada a um método do protocolo com o padrão de assinatura `when<classe da mensagem>`. Através dessas mensagens o SMQOSP permite que o agente local de mobilidade estabeleça novas reservas na rede fixa em benefício do nó móvel.

Além das mensagens ilustradas na Figura 4.6, existem as mensagens definidas pelo protocolo de sinalização⁴², acionadas pelo SMQOSP durante o processo de estabelecimento de reservas, ilustradas na Figura 4.7.

⁴² Na implementação, o protocolo de sinalização utilizado corresponde a uma simplificação do protocolo RSVP-A, denominado SRSVP-A (*Simple ReSerVation Protocol in Advance*).

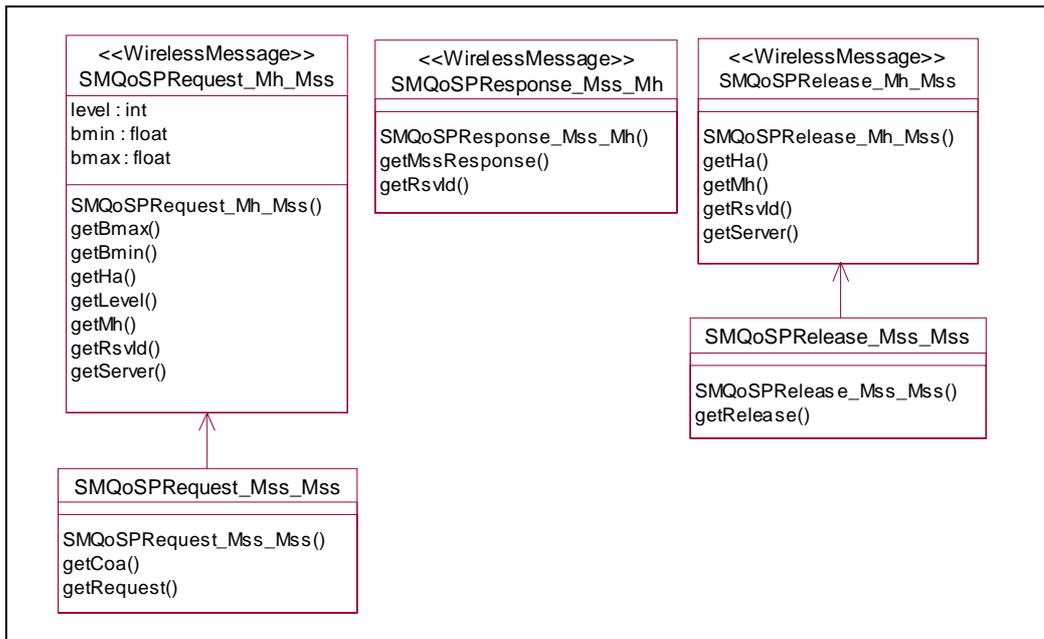


Figura 4.6 Declaração das mensagens do protocolo SMQoSP

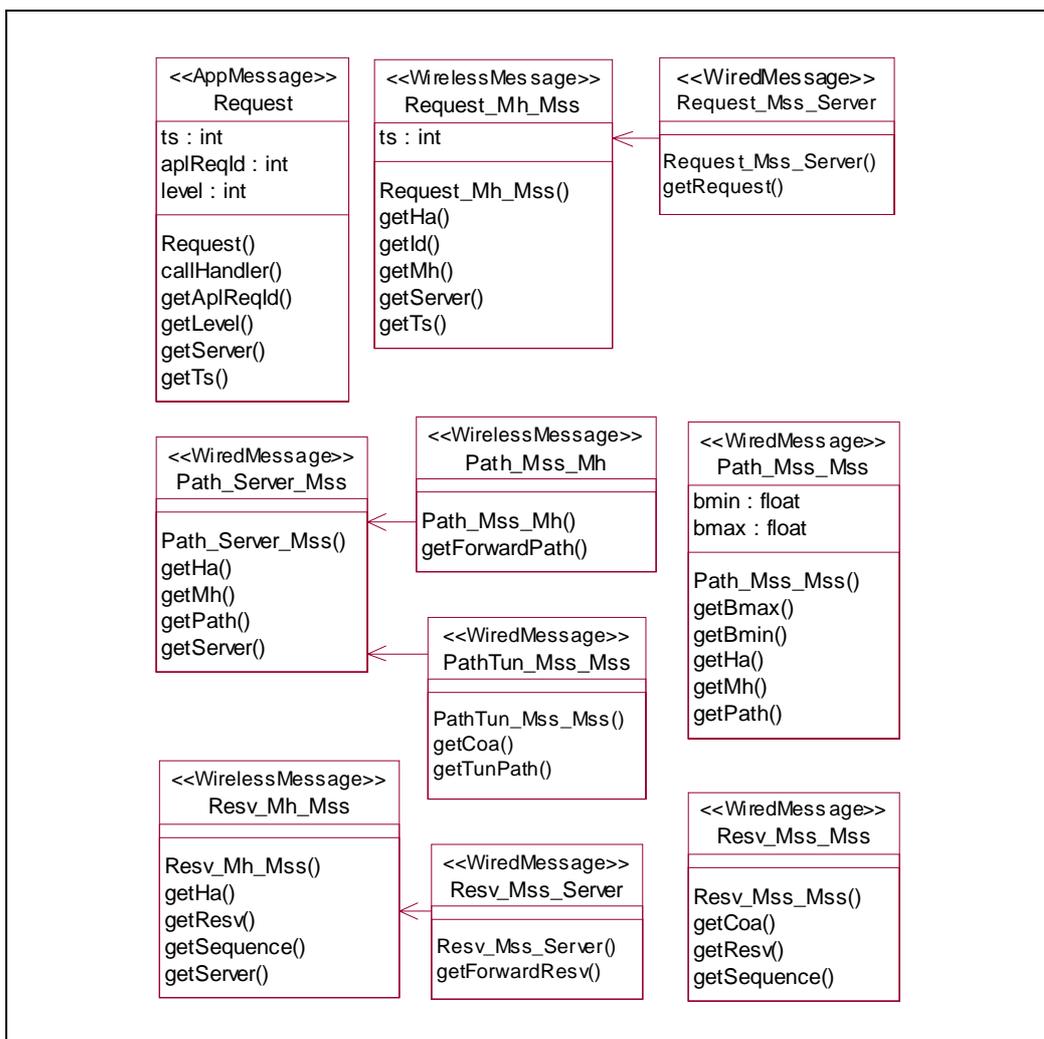


Figura 4.7 Declaração das mensagens do protocolo SRSVP-A acionadas pelo SMQoSP

4.5.4 IMPLEMENTAÇÃO DO PROTOCOLO SMQOSP

A Figura 4.8 ilustra o diagrama de classes dos micro-protocolos e das instâncias funcionais do protocolo SMQoSP. A classe *SMQoSPMss* implementa uma instância funcional do protocolo SMQoSP para as estações base, que é composta pelos micro-protocolos definidos pelas interfaces *SMQoSPMssWireless*, *SMQoSPMssWired* e *SMQoSPMssHandoff*. A classe *SMQoSPMh* implementa uma instância funcional do protocolo SMQoSP para as estações móveis, instanciando apenas o micro-protocolo definido pela interface *SMQoSPMhWireless*, responsável pelo envio de mensagens através do meio sem fio. Para realizar a simulação foi criada uma terceira instância (classe *SMQoSPServer*), que representa as estações fixas, implementando um protocolo que responde às requisições das estações móveis, efetuando as reservas de recursos.

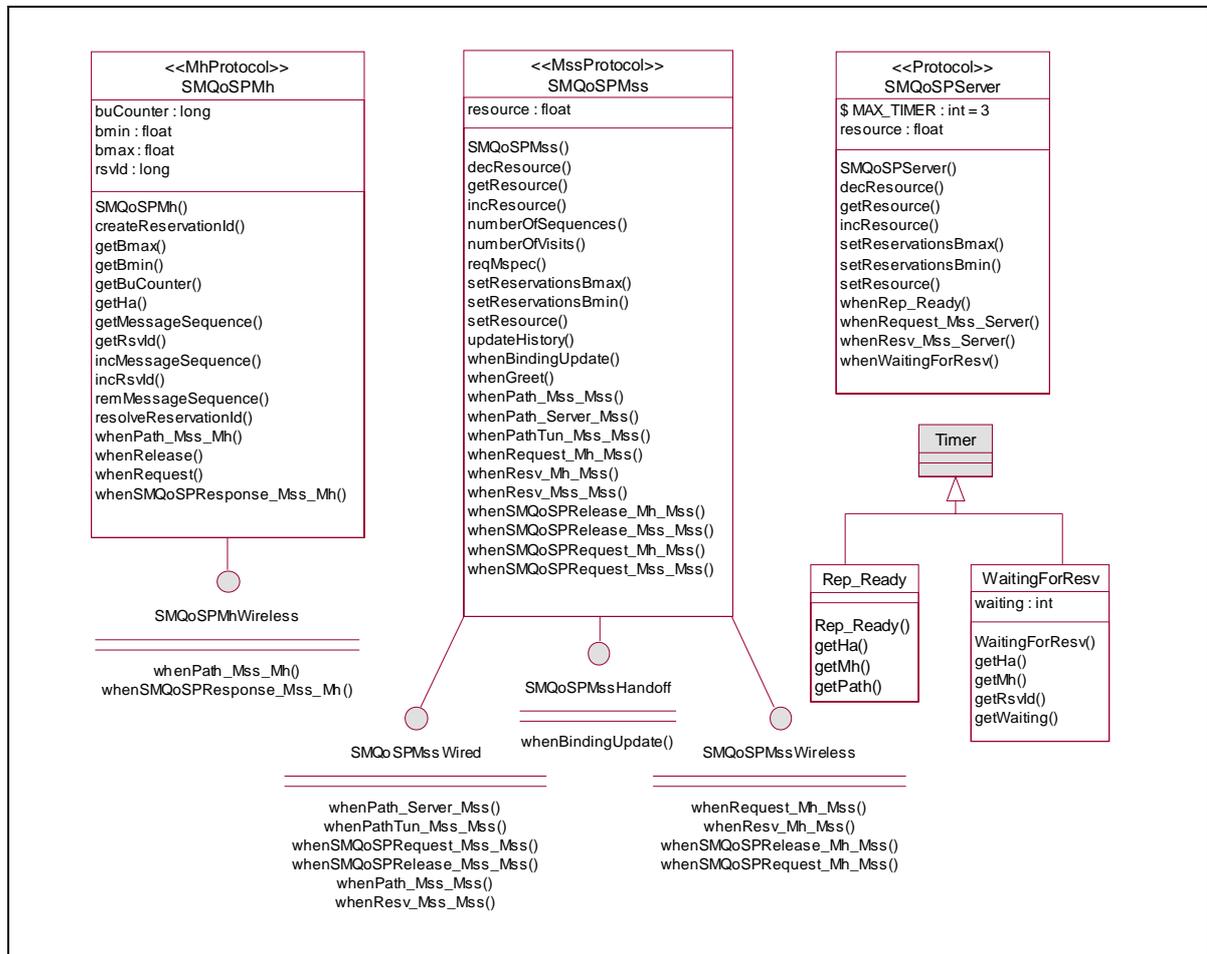


Figura 4.8 Declaração dos micro-protocolos e instâncias do protocolo SMQoSP

Como se pode observar na Figura 4.8, as classes *Rep_Ready* e *WaitingForResv* funcionam como *timeouts*, sendo instanciadas como subclasses de *Timer*, que no

MobiCS é a classe responsável pela programação de eventos de tempo. Essas mensagens são tratadas na classe *SMQoSPServer* pelos métodos `whenRep_Ready()` e `whenWaitingForResv()` e correspondem, respectivamente, à resposta a uma solicitação de serviço do nó móvel e ao *timeout* associado à recepção de mensagens para o estabelecimento de reservas.

A Figura 4.9 traz a seqüência de invocação de mensagens no protocolo SMQoSP relacionadas ao processo de solicitação de serviço pela aplicação e ao estabelecimento de reservas imediatas entre o receptor (nó móvel) e o transmissor (nó fixo), e de reservas antecipadas que serão efetuadas pelo agente local em benefício do nó móvel ao longo do caminho até as localidades por ele visitadas. Como se pode observar, em (9) o agente local aciona o mecanismo de predição para obter a especificação de mobilidade do nó móvel.

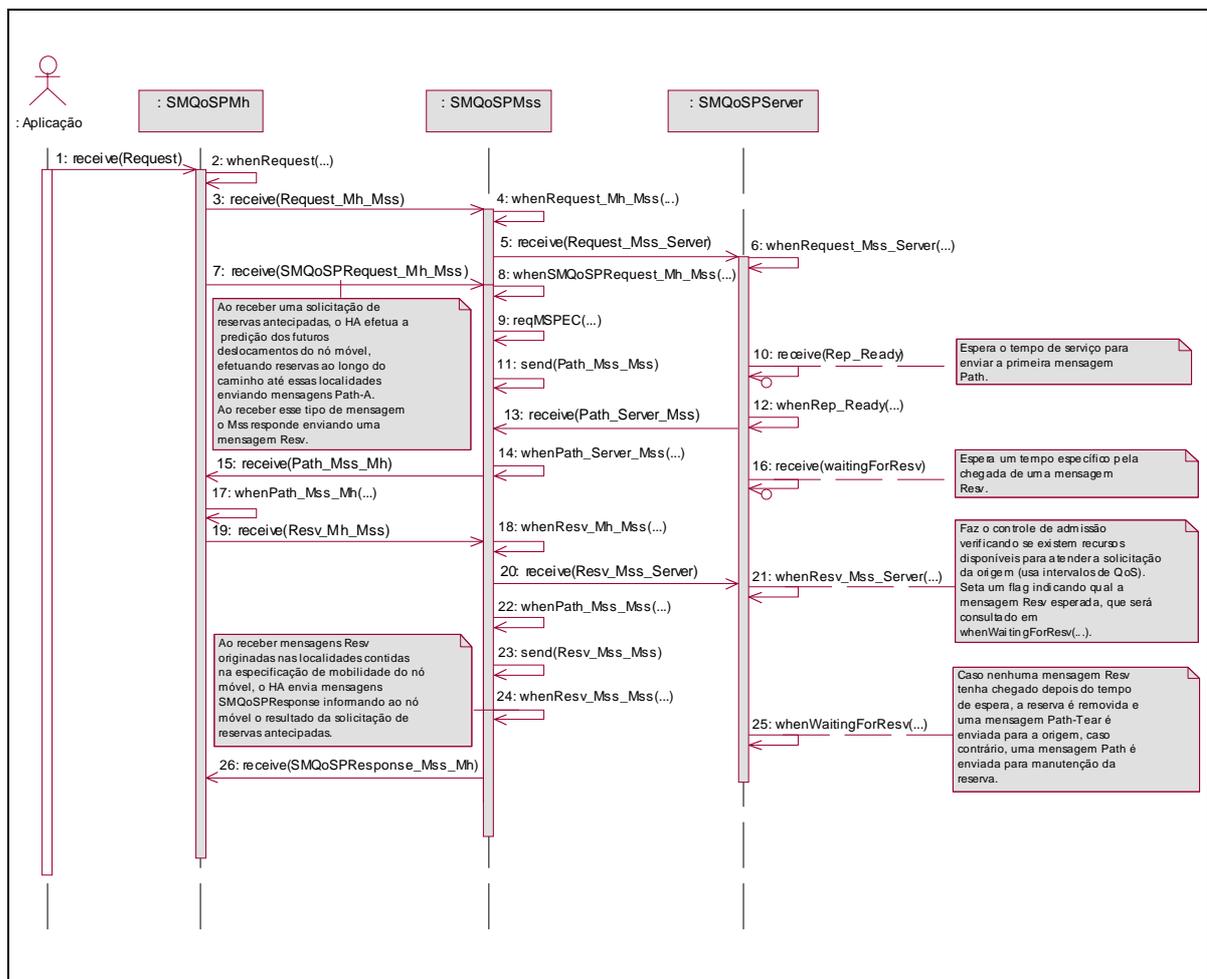


Figura 4.9 Seqüência de invocação e tratamento de mensagens após uma solicitação de serviço

A Figura 4.10 traz a seqüência de invocação de mensagens no protocolo SMQoSP relacionadas ao encerramento do serviço pela aplicação e,

conseqüentemente, à liberação das reservas (imediatas e antecipadas) que foram estabelecidas pelo nó móvel.

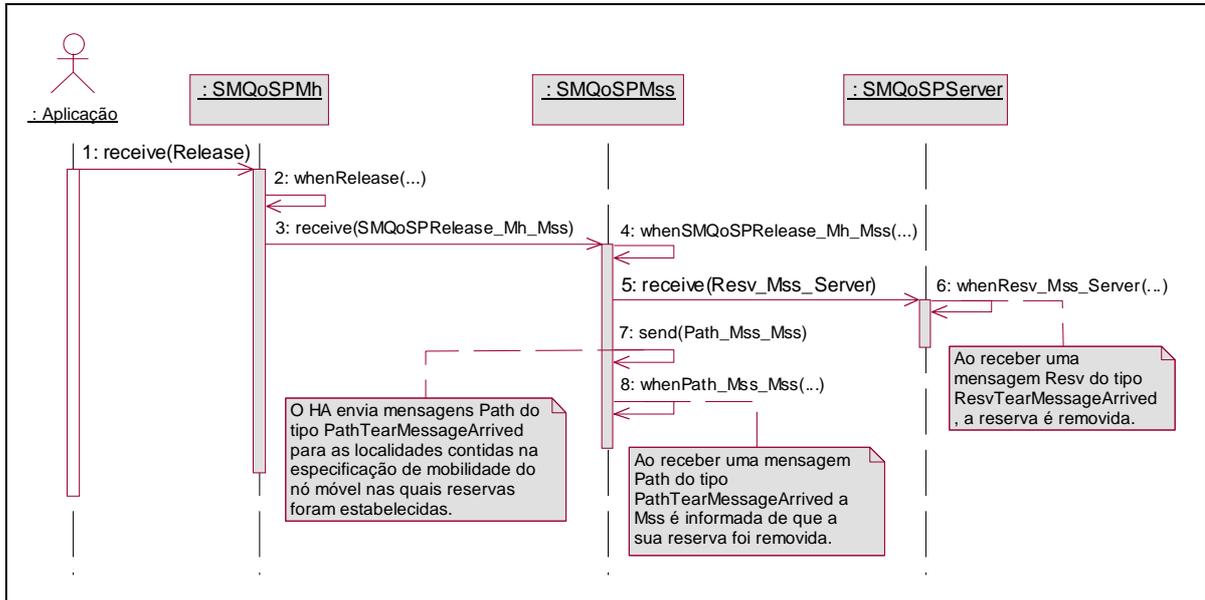


Figura 4.10 Seqüência de invocação de mensagens após a solicitação de encerramento do serviço

A Figura 4.11 traz a seqüência de invocação de mensagens no protocolo SMQoSP relacionadas à atualização do histórico de mobilidade do nó móvel quando este anuncia o seu deslocamento ao agente estrangeiro, o qual se encarrega de fazer a atualização de *binding* junto ao agente local do nó móvel.

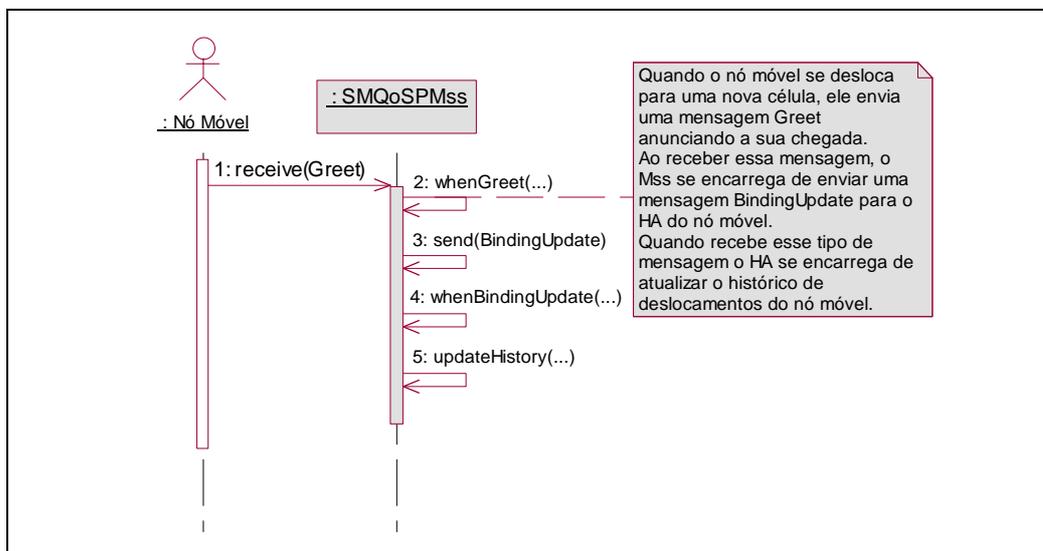


Figura 4.11 Seqüência de invocação e tratamento de mensagens durante o deslocamento do nó móvel

4.6 CENÁRIOS PARA SIMULAÇÃO

Para ilustrar o funcionamento do protocolo SMQoSP, foram criados dois cenários determinísticos, os quais estão detalhados nas Subseções 4.6.1 e 4.6.2. Devido às limitações encontradas no ambiente, não foi possível utilizar nas simulações todos os critérios de predição de mobilidade apresentados na Subseção 2.1.4. Como não existe no MobiCS uma primitiva que forneça o tempo simulado e mecanismos para obtenção do posicionamento geográfico⁴³ e da velocidade de deslocamento através, por exemplo, de GPS (*Global Positioning System*), os critérios que se baseiam na análise de informações temporais e relacionadas à direção do deslocamento foram descartados. Pelos resultados obtidos no trabalho de Chan et al. (1998), pode-se concluir que os critérios que foram adotados na simulação do SMQoSP, o *critério da localização* e o *critério do segmento*, não oferecem resultados expressivos por não considerarem a direção do deslocamento dos nós, obtendo uma média de acerto de 50% e 56%, respectivamente.

Os cenários descritos nas Subseções 4.6.1 e 4.6.2 correspondem, no MobiCS, à implementação do método `DetermSimulation.script()`, onde *DetermSimulation* é a classe que implementa a simulação determinística. Um *script* é composto de chamadas às interfaces de simulação das máquinas, pontos de sincronização, comandos de depuração e comando adicionais criados pelo usuário para facilitar a avaliação da simulação. Uma chamada à interface de simulação de uma máquina simulada escalona o respectivo evento na camada de simulação da máquina, que irá decidir quando ele será executado. Todos os eventos escalonados são executados antes do término de um passo de simulação. Como cada elemento simulado é implementado por um processo concorrente, a ordem de execução dos eventos por diferentes elementos simulados em um mesmo passo de simulação é indeterminado. Somente os eventos de um mesmo elemento simulado mantêm a ordem descrita no *script*. A saída do modo determinístico de simulação é um *log* dos eventos que ocorreram durante a simulação, com o qual o usuário pode identificar se o protocolo funcionou corretamente ou não.

⁴³ No MobiCS, atualmente, toda localização é mapeada em uma célula, portanto, o posicionamento geográfico de um nó móvel está unicamente relacionado a célula na qual ele se encontra. Entretanto, um módulo adicional poderia ser incorporado ao MobiCS permitindo a obtenção de informações relacionadas ao posicionamento geográfico do nó.

4.6.1 CENÁRIO I

O primeiro cenário testa a correção do SMQoSP quanto a três aspectos: (i) a troca de mensagens entre as instâncias do protocolo no processo de solicitação de serviços e o conseqüente estabelecimento de reservas imediatas e antecipadas, (ii) a liberação de recursos mediante a requisição de encerramento do serviço pela aplicação e (iii) a atualização das informações contidas nas estruturas de dados do protocolo.

A Figura 4.12 ilustra o primeiro cenário de reserva de recursos, trazendo um nó móvel (MH) como o receptor e um nó fixo (CN) como o transmissor, com a estação base Mss_1 atuando como o seu agente local e as estações base Mss_2 e Mss_3 como os seus agentes estrangeiros e constando na especificação de mobilidade do nó móvel. Como se pode observar pela troca de mensagens, o nó móvel efetua um pedido de reserva de recursos junto ao servidor através do seu agente local (Mss_1). Ao mesmo tempo, o nó móvel envia ao seu agente local uma solicitação para que sejam estabelecidas reservas antecipadas junto aos seus agentes estrangeiros (Mss_2 e Mss_3). Após efetuar todas as reservas e ter migrado para a célula associada à estação base Mss_3 , o nó móvel finaliza todas as suas reservas enviando uma mensagem SMQoQP-Release para o seu agente local.

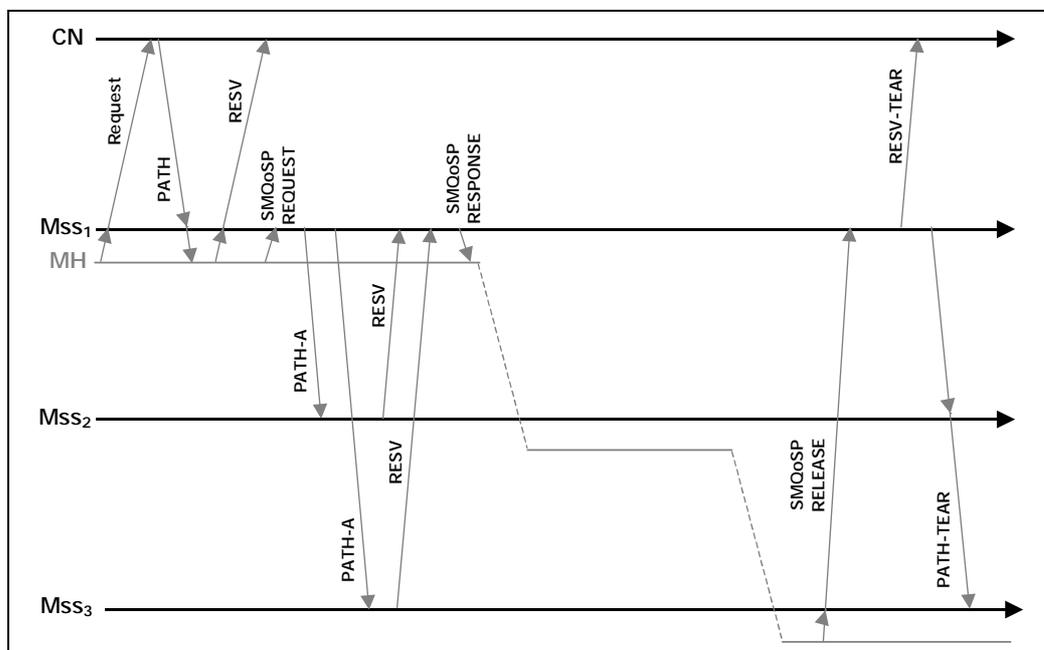


Figura 4.12 Cenário I: teste determinístico do protocolo SMQoSP

A Figura 4.13 reproduz o trecho de código referente à implementação do *script* correspondente ao cenário determinístico ilustrado pela Figura 4.12.

```
public void script() {
    acceptMessageOn(mh, true);
    acceptMessageOn(mss1, true);
    acceptMessageOn(mss2, true);
    acceptMessageOn(mss3, true);
    acceptMessageOn(cn, true);

    acceptTimeoutOn(cn, false);
    changeShowMode(cn, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mss1, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mss2, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mss3, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mh, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    next();

    mh.moveTo(cell1);
    next();

    mh.receive( new smgosp.messages.miptr.Request(cn.getAddress(), 1, 1, 3) );
    next();

    acceptTimeout(cn, Rep_Ready.class);
    next();

    acceptTimeout(cn, WaitingForResv.class);
    next();

    mh.moveTo(cell2);
    next();

    acceptTimeout(cn, Rep_Ready.class);
    next();

    acceptTimeout(cn, WaitingForResv.class);
    next();

    mh.moveTo(cell3);
    next();

    acceptTimeout(cn, Rep_Ready.class);
    next();

    acceptTimeout(cn, WaitingForResv.class);
    next();

    mh.receive( new smgosp.messages.Release(cn.getAddress(), 1) );
    next();
}
```

Figura 4.13 Implementação do *script* determinístico do Cenário I

Todos os elementos definidos no cenário apresentado na Figura 4.13, com exceção do transmissor (CN), estão no modo de aceitação de mensagens (`acceptMessageOn(elemento, true)`), o que garante que todas as mensagens geradas pelo SMQoSP serão entregues.

As mensagens trocadas pelo protocolo SMQoSP no processo de estabelecimento de reservas (*Path* e *Resv*) apresentam uma interdependência (Figura 4.12). Como o protocolo é *soft-state*, existe a necessidade de se enviar periodicamente essas mensagens para a manutenção da reserva efetuada. Ao enviar uma mensagem *Path* o transmissor aguarda o recebimento de uma mensagem *Resv*, de modo semelhante, ao enviar uma mensagem *Resv* o receptor aguarda o recebimento de uma mensagem *Path*. Caso isso não ocorra em um período de tempo determinado, o protocolo deverá executar ações que vão desde a retransmissão de mensagens até a remoção da reserva estabelecida. Para representar esse tipo de ação, o MobiCS oferece o conceito de *eventos de tempo (timeouts)*, representado pela classe *Timer*. Um *timeout* é um evento interno a um elemento simulado que deve ser processado após a passagem de um tempo determinado. Na Figura 4.13, inicialmente o elemento CN tem o recebimento de *timeouts* desabilitado (`acceptTimeoutOn(cn, false)`). Através de chamadas ao método `acceptTimeout(Element, Timer)` o transmissor (CN) realiza o processamento explícito de um *timeout* específico, procurando pelo evento de tempo indicado na fila de *timeouts*.

Para auxiliar na avaliação dos protocolos simulados, o MobiCS provê o método `changeShowMode()` com o qual o usuário pode habilitar ou desabilitar a exibição de mensagens durante a ocorrência de eventos na simulação, como os eventos de recebimento, envio e processamento de mensagens, migração e início de um novo passo de simulação. No exemplo ilustrado na Figura 4.13 o recebimento de mensagens (`EVENT_MESSAGE_RECEIVING`) foi desabilitado (`HIDE`), permitindo que esse evento fosse tratado internamente pelo SMQoSP.

No *script* da Figura 4.13 o método `next()` representa o encerramento de um passo de simulação determinístico e é o único comando que não é executado concorrentemente. Por sua vez, o método `moveTo(Cell)` é utilizado para efetuar o escalonamento de eventos de migração da estação móvel (MH). Já as chamadas ao método `receive()` são as responsáveis pela inclusão das mensagens do protocolo na fila de mensagens, permitindo o seu recebimento pelo destinatário.

4.6.2 CENÁRIO II

O segundo cenário amplia os aspectos avaliados no primeiro cenário (Subseção 4.6.1), testando o comportamento do mecanismo de controle de admissão ao se adotar o conceito de intervalos de QoS no estabelecimento de reservas de recursos.

A Figura 4.14 ilustra o segundo cenário de reserva de recursos, trazendo os nós móveis MH_1 e MH_2 como receptores e um nó fixo (CN) como o transmissor. As estações base Mss_1 e Mss_2 atuam, respectivamente, como os agentes local e estrangeiro de MH_1 e estrangeiro e local de MH_2 , constando nas suas especificações de mobilidade.

Como se pode observar pela troca de mensagens, os nós móveis efetuam reservas imediatas ao longo do caminho até o servidor (CN) e reservas antecipadas de acordo com as suas especificações de mobilidade. Para realizar a simulação, o recurso considerado foi a largura de banda. O servidor foi inicializado com 100% de seus recursos disponíveis e os nós móveis, MH_1 e MH_2 , com intervalos de QoS definidos com b_{min} correspondendo a 45% e b_{max} a 60% dos recursos.

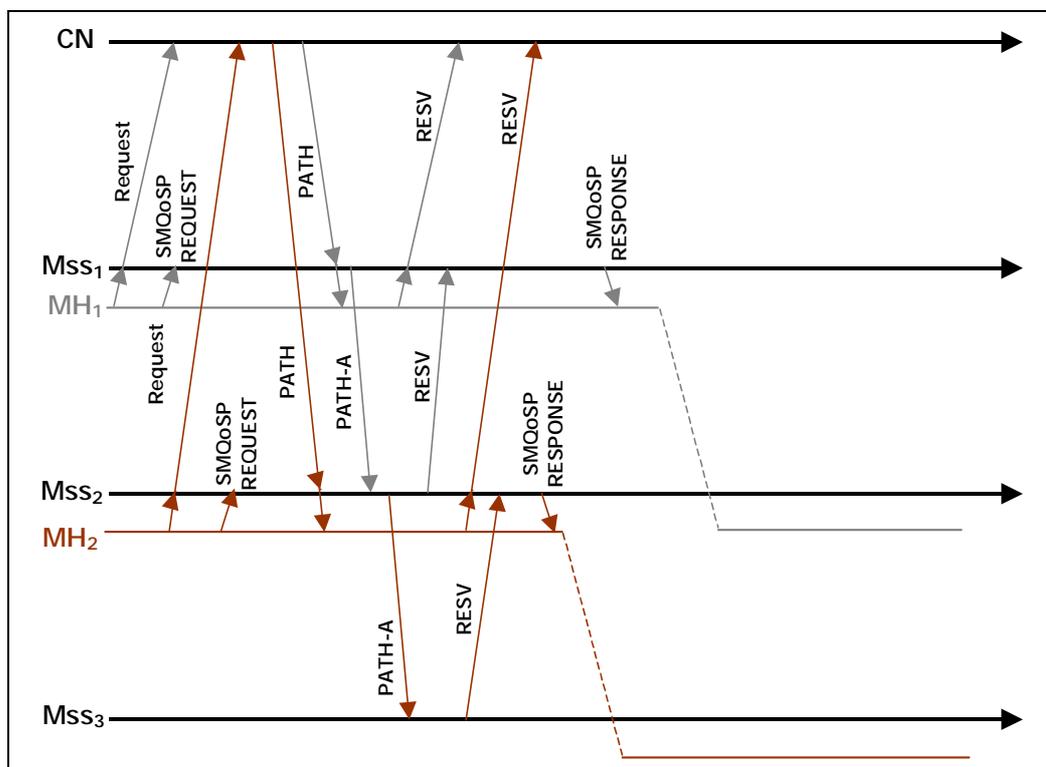


Figura 4.14 Cenário II: teste determinístico do protocolo SMQoS

A Figura 4.15 reproduz o trecho de código referente à implementação do *script* correspondente ao cenário ilustrado pela Figura 4.14.

```
public void script() {
    acceptMessageOn(mh1, true);
    acceptMessageOn(mh2, true);
    acceptMessageOn(mss1, true);
    acceptMessageOn(mss2, true);
    acceptMessageOn(mss3, true);
    acceptMessageOn(cn, true);
    acceptTimeoutOn(cn, false);

    changeShowMode(cn, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mss1, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mss2, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mss3, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mh1, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    changeShowMode(mh2, HIDE, DetermElement.EVENT_MESSAGE_RECEIVING);
    next();

    mh1.moveTo(cell1);
    next();

    mh1.receive(new smqosp.messages.miptr.Request(cn.getAddress(), 3, 1, 2));
    next();

    acceptTimeout(cn, Rep_Ready.class);
    next();

    acceptTimeout(cn, WaitingForResv.class);
    next();

    mh2.moveTo(cell2);
    next();

    mh2.receive(new smqosp.messages.miptr.Request(cn.getAddress(), 1, 2, 1));
    next();

    acceptTimeout(cn, Rep_Ready.class);
    next();

    mh1.moveTo(cell2);
    next();

    acceptTimeout(cn, Rep_Ready.class);
    next();

    acceptTimeout(cn, WaitingForResv.class);
    next();

    mh2.moveTo(cell3);
    next();
}
```

Figura 4.15 Implementação do *script* determinístico do Cenário II

Um algoritmo simples foi implementado para efetuar o controle de admissão considerando os intervalos de QoS de modo a garantir uma melhor utilização dos recursos e atender a um maior número de usuários. Esse algoritmo está reproduzido no trecho de código ilustrado na Figura 4.16.

```
if (getResource() >= resv.getBmax()) {
    rsv.setMaxResource();
    decResource(resv.getBmax()); }
else {
    if (getResource() >= resv.getBmin()) {
        rsv.setMinResource();
        decResource(resv.getBmin()); }
    else {
        setReservationsBmin();
        if (getResource() >= resv.getBmin()) {
            rsv.setMinResource();
            decResource(resv.getBmin()); }
    }
}
```

Figura 4.16 Trecho de código responsável pelo controle de admissão

4.7 SUMÁRIO

Este capítulo apresentou uma instanciação dos *frameworks para provisão de QoS em redes móveis sem fio* através da implementação de um protocolo de reservas de recursos simplificado, o SMQoSP, incorporando o conceito de intervalos de QoS e efetuando reservas antecipadas com o auxílio de um gerenciador de mobilidade centralizado. Esse protótipo foi desenvolvido utilizando o MobiCS e testado através de simulações determinísticas. Devido às características do ambiente, não foi possível incorporar parâmetros temporais ao processo de negociação das reservas antecipadas, já que o MobiCS não oferece primitivas para a obtenção do tempo simulado. Como o MobiCS não trata o posicionamento dos nós por meio de coordenadas e nem considera a velocidade com a qual as estações móveis realizam os seus trajetos, os algoritmos utilizados para realizar a predição dos deslocamentos no gerenciador de mobilidade se restringiram aos critérios da localização e do segmento, os quais apresentam uma precisão média de 50% (Chan et al., 1998).