

4 Implementação

Esse capítulo apresenta alguns aspectos da implementação do SACS, relativo ao funcionamento do protocolo Moratus. A primeira seção deste capítulo descreve como foi implementado o mecanismo para obtenção das informações de contexto do dispositivo. A segunda seção descreve qual foi o serviço de comunicação de grupo confiável utilizado. As duas seções seguintes apresentam alguns casos de uso das duas etapas do Moratus.

4.1.Obtenção de Contexto

Como foi mencionado no capítulo 3, há na arquitetura do SACS uma componente denominada Monitor, que é a responsável pela obtenção das informações de contexto computacional (de sistema) de um dispositivo. Na implementação realizada neste trabalho, as informações de contexto são obtidas através do Serviço de Informação de Contexto (CIS), um serviço oferecido pelo middleware MoCA (Sacramento, 2004).

O CIS é um serviço, onde um servidor recebe periodicamente informações de contexto de sistema de dispositivos móveis. Para que as informações de contexto dos dispositivos cheguem ao CIS, há uma entidade denominada *monitor* (Sacramento, 2004) responsável pelo envio periódico de informações de contexto do dispositivo para um servidor CIS. O monitor é um *daemon* que coleta periodicamente informações de contexto de sistema do dispositivo, como por exemplo: percentual de utilização do processador do dispositivo, percentual do nível residual de energia e total de memória primária, em kilobytes (LAC, 2008). Essas informações são enviadas ao CIS, deixando-o sempre atualizado com relação ao contexto local de cada dispositivo conectado a ele.

Existem duas formas para obter as informações de contexto de um dispositivo junto ao CIS: a consulta síncrona e a consulta assíncrona. A primeira forma é do tipo requisição-resposta, onde as aplicações solicitam o contexto no qual elas estejam interessadas e aguardam a resposta com os dados por parte do

CIS. Na forma de acesso assíncrono, a aplicação faz uma subscrição no CIS, solicitando receber notificações toda vez que ocorrer uma variação em uma ou mais variáveis de contexto (Sacramento, 2004). No SACS, optou-se pelo acesso síncrono as informações de contexto, pois o CIS não permite o acesso assíncrono da variável de contexto que indica a intensidade do sinal recebido, e que no SACS é usada como indicador da qualidade do enlace sem fio do dispositivo.

4.2. Camada de Comunicação de Grupo

O Moratus requer comunicação em grupo confiável para que nenhuma mensagem relacionada a coordenação do processo adaptativo seja perdida, pois tal perda poderia introduzir inconsistências no grupo. Para tal, foi utilizado o *JGroups* (Ban, 1998) (Ban, 1998b) que é uma ferramenta que permite a comunicação em grupo confiável. Outros middlewares como: Appia (MIRANDA, 2001), Cactus (HILTUNEN, 2000) e Ensemble (HAYDEN, 1998) também oferecem a funcionalidade similar. Porém o *JGroups* é o único middleware que possui uma implementação específica (*JGroups-CDC*) para dispositivos móveis como PDAs e Smart Phones.

Através do *JGroups*, processos podem conectar-se a um grupo, enviar mensagens de forma confiável a um integrante do grupo ou ao grupo inteiro. O *JGroups* também monitora os membros do grupo e os notifica sobre novas visões do grupo (isto é, a lista de membros). Isto ocorre toda vez que um novo membro entra para o grupo, ou um membro sai do grupo, espontaneamente ou em decorrência de alguma falha.

No *JGroups*, um grupo é identificado pelo nome. Os grupos não necessitam ser criados explicitamente, pois sempre que um processo solicita a entrada em um novo grupo, aquele grupo será criado automaticamente.

A Figura 4 apresenta a arquitetura do *JGroups*. Ela consiste basicamente de três partes: a API *canal* que é acessada por aplicações que precisem utilizar a comunicação em grupo confiável, a *pilha de protocolos*, que implementa as propriedades do canal e os *blocos de construção*, que são blocos que implementam serviços de mais alto nível de abstração com base no canal, tais como por exemplo, Chamadas Remotas de Procedimentos (RPC).

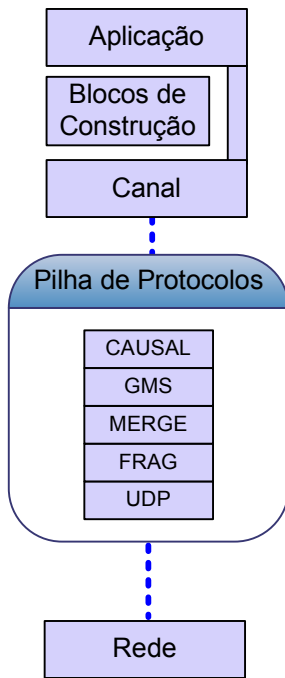


Figura 4 – Arquitetura do JGroups.

O canal é a principal entidade do JGroups. Ele permite a entrada e a saída dos processos do grupo, bem como a atualização do grupo sobre sua composição corrente. Tal composição é formada por uma lista dos endereços IP dos membros do grupo e é definida como a *visão do grupo*.

Neste trabalho, o canal é utilizado como o meio de comunicação entre os membros de um grupo para todas as trocas de mensagens do protocolo Moratus.

Em JGroups, um canal deve estar sempre associado a um nome, que é o seu identificador e ao mesmo tempo o identificador do grupo. Tal identificador atribuído ao canal é mapeado automaticamente para um endereço IP *multicast* válido. Todos os processos que devam fazer parte de um determinado grupo devem se conectar ao canal com o mesmo identificador.

Outra função do canal é manter atualizada a informação sobre a configuração corrente de um grupo (*visão do grupo*), bem como informá-la a todos os seus integrantes sempre que um processo se conecta ou se desconecta, voluntária ou involuntariamente, de um grupo. Portanto, o canal possui uma lista com todos os endereços IP dos integrantes do grupo. Sempre que houver alguma mudança na visão, seja por conexão ou desconexão, todo o grupo será notificado da nova visão do mesmo. Quanto ao envio de mensagens, quando um processo desejar se comunicar com o grupo, o canal usará a visão corrente para enviar mensagens a todos os elementos listados nela.

A pilha de protocolos (Figura 4) contém a pilha de protocolos usada pelo canal no envio e recebimento de mensagens, onde cada camada implementa uma funcionalidade requerida para a gerência e comunicação de grupo confiável. Tal entidade pode ser configurada com diferentes camadas de protocolos. No entanto no SACS optou-se por utilizar a pilha de protocolos padrão do JGroups, pois ela oferece todos os protocolos necessários para comunicação confiável. Tal pilha de protocolos é composta pelas seguintes camadas:

- *UDP*. Responsável por enviar as mensagens recebidas pelas camadas superiores, utilizando comunicação UDP não-confiável e *multicast* IP, e encaminhar para as camadas superiores as mensagens recebidas pela rede.
- *PING*. Responsável por controlar a pertinência de um membro ao grupo no momento em que o canal é criado. Nessa camada são trocadas diversas mensagens do tipo *ping* para verificar a presença de um dispositivo no grupo.
- *MERGE2*. Possibilita a descoberta de subgrupos que tenham sido criados devido ao particionamento da rede e realiza a integração destes novamente em um único grupo.
- *FAILURE_DETECTOR*. Responsável por apontar possíveis falhas ou desconexões em membros do grupo, e é baseado em um protocolo de pulso.
- *VERIFY_SUSPECT*. Responsável por verificar constantemente se um membro do grupo é suspeito de ter saído do grupo ou ter falhado.
- *NAKACK*. Responsável por implementar um protocolo para confirmação do recebimento de mensagens de um transmissor para um receptor. Nesse modelo de confirmação o receptor confirma o recebimento de cada uma das mensagens enviadas pelo transmissor individualmente. Através desta técnica garante-se maior confiabilidade da comunicação em detrimento do desempenho da comunicação.
- *UNICAST*. Responsável por implementar comunicação unicast confiável.
- *STABLE*. Computa o número de mensagens disseminadas (*broadcast*) que são estáveis, ou seja, que tenham sido recebidas por todos os membros do grupo. Sua implementação utiliza o protocolo descrito em (Guo, 1997).

- *VIEW_SYNC*. Responsável por disseminar periodicamente a visão corrente do grupo a todos os membros do grupo.
- *GMS*. Responsável por implementar o protocolo de pertinência a grupos. Ela trata a entrada, saída e falha de processos e difunde novas visões de acordo com esses três eventos.
- *FC*. Implementa um mecanismo de controle de fluxo simples, baseado em um sistema de crédito, em bytes, que um transmissor pode enviar.
- *FRAG2*. Implementa um mecanismo que permite a fragmentação de uma mensagem maior que um determinado limite, em diversos fragmentos e a composição dos fragmentos de uma mensagem no destinatário.

Finalmente, a última entidade apresentada na Figura 4 são os blocos de construção. Como a entidade canal é bastante simples, são necessárias outras entidades, que são construídas com base no canal, para oferecer um mecanismo de comunicação de maior nível de abstração para as aplicações, como por exemplo, oferecer um mecanismo de chamada remota de procedimento (RPC). Essa entidade não será descrita em detalhe pois não foi utilizada no desenvolvimento do SACS.

4.3. Eleição do Contexto Global

Nesta seção serão descritos os casos de uso: *Convocar a Eleição do Contexto Global*, *Responder à Convocação da Eleição do Contexto Global*, *Receber a Resposta da Convocação da Eleição do Contexto Global* e *Tratar Desconexão*. Os três primeiros casos de uso foram escolhidos por representarem a forma como os membros do grupo trocam entre si seus contextos locais, para determinar futuramente o contexto global. Já o último caso de uso foi escolhido por representar o tratamento de uma exceção.

Devido à grande similaridade na implementação, os casos de uso *Responder à Convocação da Eleição do Contexto Global* e *Receber a Resposta da Convocação da Eleição do Contexto Global*, serão apresentados na mesma subseção que terá o título de *Recebimento de Contexto*.

4.3.1. Convocação da Eleição do Contexto Global

Esse caso de uso representa o momento em que a componente DVSC detecta que ocorreu uma variação significativa no valor do RSSI do dispositivo e convoca a eleição do contexto global, que representa o início de um processo adaptativo. Tal caso de uso é detalhado no diagrama de seqüência apresentado na Figura 5.

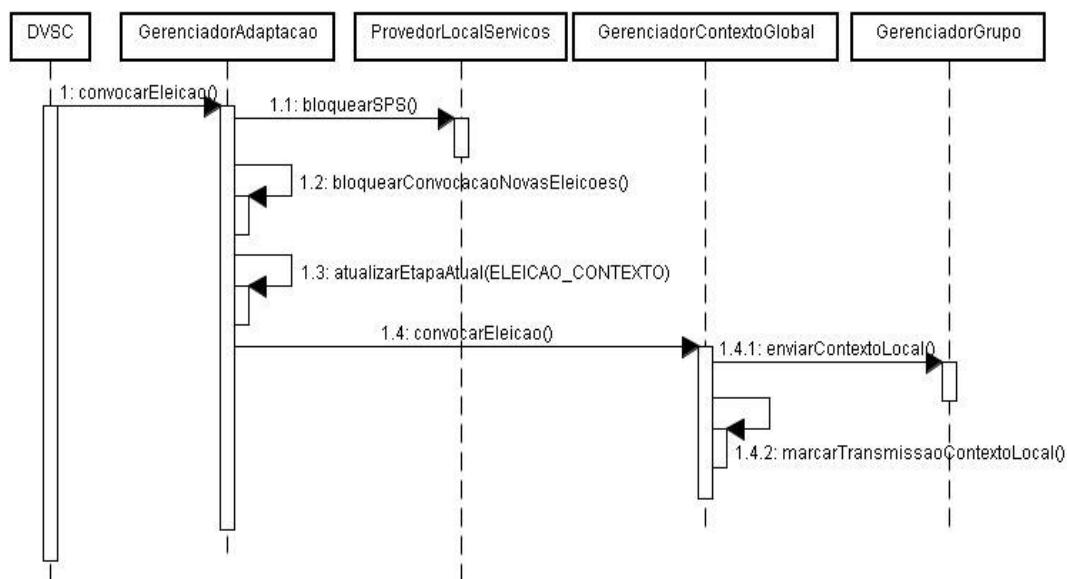


Figura 5 – Convocação da Eleição do Contexto Global.

Ao detectar uma variação significativa no contexto local, o DVSC convoca a eleição do contexto global. Antes de enviar a mensagem para os membros do grupo convocando-os para a eleição, algumas ações são executadas localmente no dispositivo.

O convocador (dispositivo que detecta a variação significativa do contexto local) de toda eleição do contexto global bloqueia o SPS (método `bloquearSPS()` da componente *Provedor Local de Serviços*) para evitar que o SPS seja consultado por novos membros enquanto a adaptação está em curso. Como mencionado anteriormente, essa é uma forma de impedir que tais membros adquiram um serviço que está prestes a ser modificado pelo processo adaptativo. O bloqueio é feito pela entidade *Provedor Local de Serviços* em um membro do grupo, que envia uma mensagem do tipo bloqueio para o SPS.

Como a componente Monitor continuará coletando as informações de contexto do dispositivo e solicitando ao DVSC que verifique se ocorreu alguma variação significativa no contexto, o Gerenciador da Adaptação deve se bloquear (`bloquearConvocacaoNovasEleicoes()`) para impedir que uma nova eleição do contexto global seja convocada pelo DVSC enquanto há um processo adaptativo está sendo executado.

O Gerenciador da Adaptação atualizará localmente a etapa na qual o processo adaptativo se encontra, pois a etapa atual será usada como uma

referência para que o Gerenciador da Adaptação, ao receber as mensagens do grupo, as encaminhe para a componente responsável por cada etapa.

O Gerenciador do Contexto Global enviará para os membros do grupo o contexto local do dispositivo, com o objetivo de convocar uma eleição do contexto global e irá se bloquear para impedir que o contexto local seja novamente transmitido, até que o processo adaptativo esteja finalizado.

4.3.2.Recebimento de Contexto

Nessa seção serão apresentados dois casos de uso bastante similares na prática, porém conceitualmente diferentes, que são: Responder à Convocação da Eleição do Contexto Global e Receber a Resposta da Convocação da Eleição do Contexto Global.

Todas as mensagens trocadas pelo grupo durante a etapa da eleição do contexto global são do mesmo tipo (`trocaContexto`) e transmitem o contexto local do dispositivo que as enviou. Dependendo da ordem em que a mensagem é recebida, ela pode representar a convocação de uma nova eleição ou a resposta à convocação de uma eleição. A primeira mensagem do tipo `trocaContexto` que um membro do grupo recebe representa a convocação de uma eleição. Todas as mensagens seguintes do mesmo tipo, serão consideradas respostas à convocação da eleição, enviadas pelos demais membros.

Assim que um membro do grupo receber a mensagem de convocação da eleição, ele irá respondê-la enviando seu contexto local para o grupo apenas uma única vez, já que o mecanismo de comunicação de grupo é confiável e todos os membros do grupo receberão tal mensagem.

Após responder à convocação da eleição, o membro do grupo deverá apenas coletar os contextos dos demais membros do grupo (Receber a Resposta da Convocação da Eleição do Contexto Global), até que ele adquira os contextos de todos os membros.

Os casos de uso desta seção estão descritos no diagrama de seqüência encontrado na Figura 6.

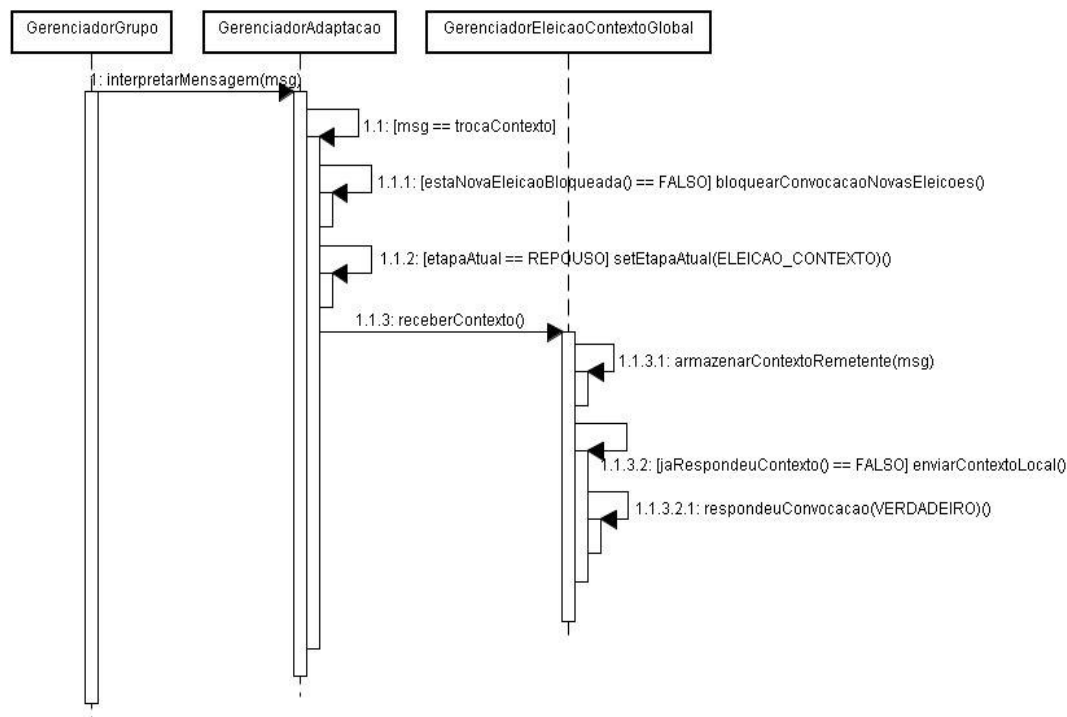


Figura 6 – Receber contexto e responder a convocação da eleição do contexto global.

Assim que o Gerenciador da Adaptação recebe uma mensagem do grupo relacionada ao protocolo Moratus, ele irá interpretá-la para identificar a qual etapa ela corresponde (no caso específico discutido nesta seção, trata-se da eleição do contexto global) para encaminhá-la para a componente responsável pela etapa (neste caso, o Gerenciador da Eleição do Contexto Global).

Ao receber uma mensagem de convocação da eleição do contexto global, o Gerenciador da Adaptação irá desativar a função de convocação da eleição do contexto global para impedir que o membro do grupo convoque uma nova eleição, já que há um processo adaptativo, do qual ele faz parte, em execução.

Em seguida, o Gerenciador da Adaptação irá atualizar localmente a etapa corrente do processo adaptativo, para que as mensagens recebidas do grupo sejam encaminhadas para a componente responsável pela etapa na qual o processo adaptativo se encontra, conforme descrito na seção anterior.

Finalmente, o Gerenciador do Contexto Global irá tratar o recebimento do contexto de um dispositivo. A primeira ação é armazenar o contexto do dispositivo que o enviou (`armazenarContexto(msg)`). Se o membro do grupo (Gerenciador do Contexto Global) ainda não respondeu com seu contexto local, é sinal de que a mensagem recebida representa a convocação da eleição do contexto

global e ele deve enviar seu contexto para o grupo. Caso contrário, o membro do grupo (Gerenciador do Contexto Global) está apenas recebendo o contexto local de um outro membro do grupo e não há necessidade de transmitir seu contexto novamente para o grupo. Assim que o Gerenciador do Contexto Global identificar que ele já recebeu os contextos de todos os membros do grupo, ele irá passar para a próxima etapa do protocolo Moratus, a adaptação de serviços de transformação de mensagens.

4.3.3. Tratamento de Desconexão

Esse caso de uso descreve como funciona o tratamento da desconexão de um membro do grupo quando o protocolo está na etapa da eleição do contexto global. A descrição desse caso de uso é especificada no diagrama de seqüência apresentado na Figura 7.

O mecanismo para detecção de falhas é implementado pelo JGroups. Ele usa uma mensagem de pulso (*heartbeat*) através da qual os membros do grupo verificam se estão mutuamente alcançáveis (conectados). Quando um dispositivo interrompe o envio dessas mensagens de pulso (por dois segundos), o JGroups assume que ele se desconectou e atualiza todos os membros do grupo com a nova visão do grupo.

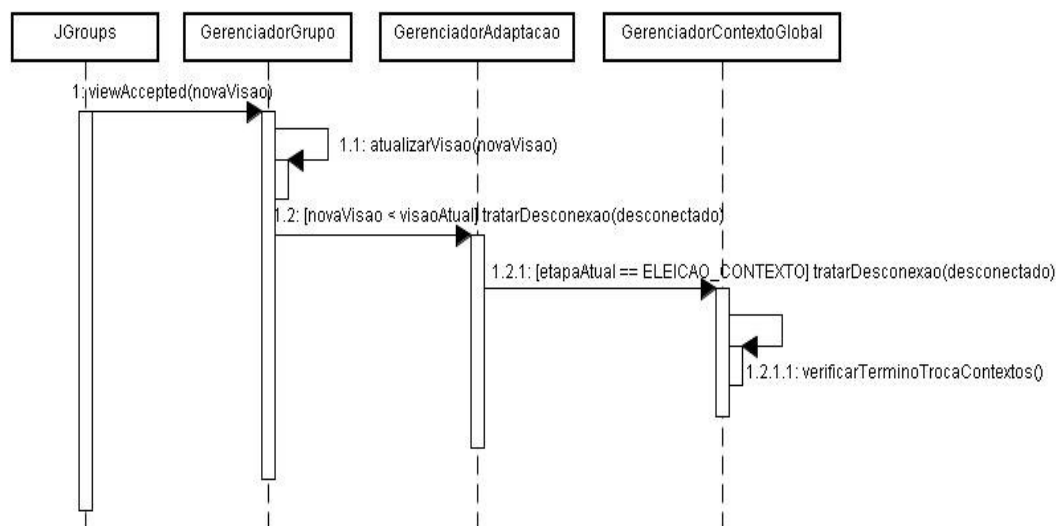


Figura 7 – Tratamento da desconexão de um dispositivo na etapa de eleição do contexto global.

O tratamento de desconexão de um dispositivo no Moratus é baseado no mecanismo de atualização da visão do grupo do JGroups. Quando uma nova visão do grupo é detectada pelo canal, ele a envia para o Gerenciador do Grupo (`viewAccepted(novaVisao)`). Essa componente atualiza a nova visão do grupo e verifica se o número de elementos dela é menor do que o número de elementos da visão antiga (`novaVisao < visaoAtual`). Caso seja, é sinal de que ocorreu a desconexão de um dispositivo e o Gerenciador da Adaptação verifica qual é a etapa atual do protocolo, para solicitar o tratamento da desconexão para a componente responsável por ela, neste caso o Gerenciador da Eleição do Contexto Global (`tratarDesconexao(desconectado)`). Como o membro do grupo desconectado já foi retirado da visão do grupo pelo Gerenciador do Grupo, para tratar a desconexão, o Gerenciador da Eleição do Contexto Global de cada membro do grupo irá apenas verificar se as mensagens de contexto dos membros que estão na visão corrente do grupo já foram recebidas, pois o grupo poderia estar esperando apenas pelo contexto local do dispositivo desconectado. Caso todas as mensagens de contexto tenham sido recebidas, o Gerenciador da Eleição do Contexto Global determina o contexto global do grupo e finaliza a etapa de eleição do contexto global.

4.4. Adaptação de Serviços de Transformação de Mensagens

Nesta seção são descritos os casos de uso: *Instalar Serviço de Transformação de Mensagens*, *Finalizar Adaptação de Serviços* e *Tratar Desconexão*.

Os dois primeiros casos de uso foram escolhidos para serem detalhados por serem considerados os mais complexos dessa etapa, devido a comunicação com o SPS. Já o tratamento de desconexão será detalhado por distinguir o tratamento da desconexão de um membro do grupo no papel de coordenador ou no papel de participante.

4.4.1. Instalação de Serviços

Conforme discutido na seção 3.1.2, a instalação de serviços é a fase na qual um membro do grupo que está participando de um processo adaptativo irá receber do SPS o serviço de transformação de mensagens que foi escolhido de acordo com

o contexto global obtido através do consenso no grupo. O diagrama de seqüência apresentado na Figura 8 especifica esse caso de uso.

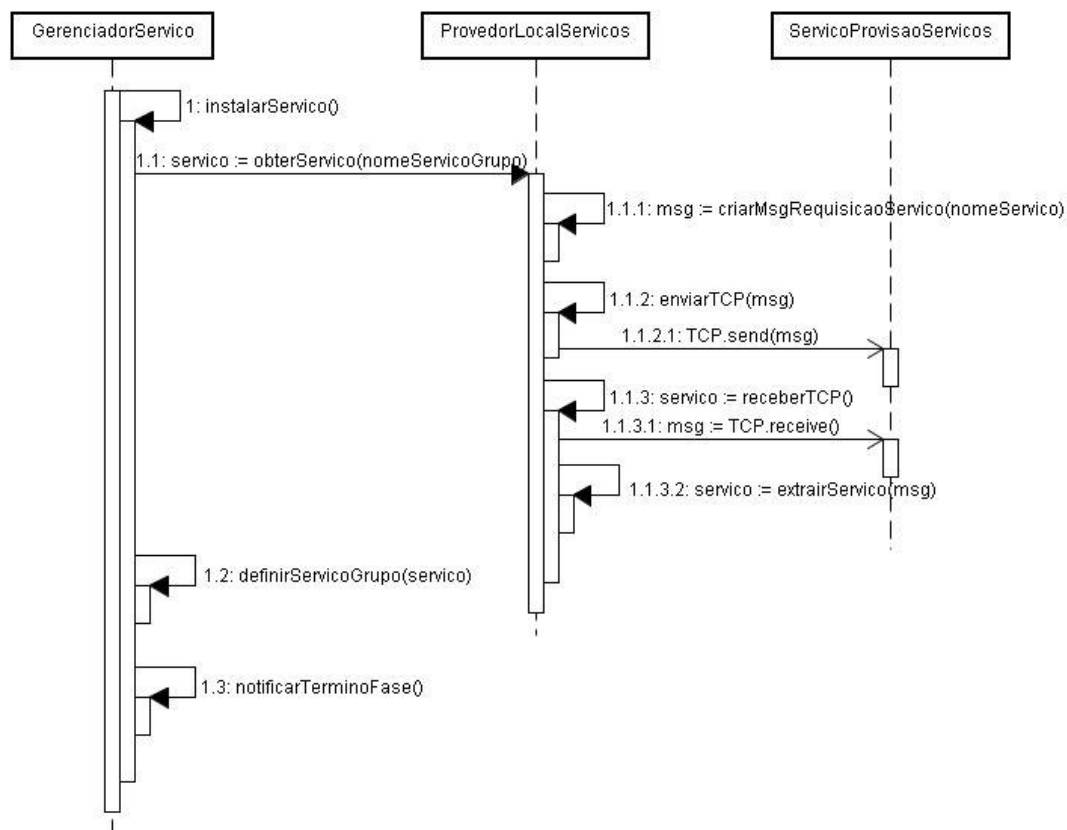


Figura 8 – Instalação de um serviço de transformação de mensagens.

Assim que os membros do grupo receberem uma mensagem do coordenador solicitando-os a instalar um novo serviço de transformação de mensagens, o Gerenciador de Serviços de cada membro do grupo solicitará ao Provedor Local de Serviços que obtenha tal serviço (`obterServico(nomeServicoGrupo)`) do SPS.

O nome do serviço que será instalado é obtido da componente DVSC a partir do contexto global acordado na etapa anterior.

O Provedor Local de Serviços irá então se comunicar com o SPS informando o nome do serviço para obter o serviço de transformação de mensagens. Tais serviços são objetos da linguagem de programação java que estão carregados no SPS e são serializados antes de serem transmitidos pelo SPS através da rede para o Provedor Local de Serviços quando este os solicita.

Considerou-se em transmitir o código-fonte dos serviços de transformação de mensagens ao invés do serviço já instanciado, porém tal opção foi descartada

pois o tempo necessário para carregar o código-fonte como objetos java iria aumentar significativamente o tempo do processo adaptativo.

Finalmente, o objeto java que representa o serviço será armazenado localmente como uma variável em cada membro do grupo, e em seguida estes enviarão uma mensagem destinada ao coordenador, informando-o que esta fase foi concluída.

O serviço de transformação de mensagens é uma classe java que encapsula um módulo codificador e outro decodificador, que também são implementados como objetos java. Portanto, a ativação ou desativação de codificadores ou decodificadores é feita através da substituição de objetos java, seguida pelo envio de mensagens indicando o término da fase.

4.4.2.Finalização da Adaptação de Serviços

Esse caso de uso representa todas as ações que são executadas pelo membro do grupo no papel de coordenador para finalizar a etapa de adaptação de serviços de transformação de mensagens e todo o processo adaptativo. O diagrama de seqüência apresentado na Figura 9 mostra todas as ações envolvidas nesse caso de uso.

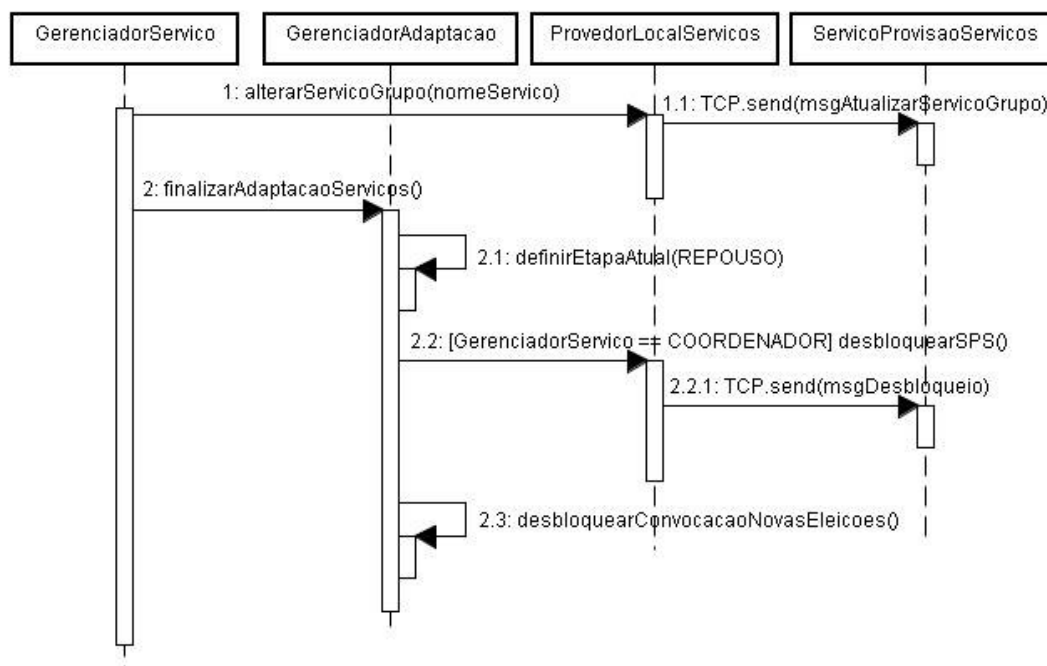


Figura 9 – Finalização da etapa de adaptação de serviços de transformação de mensagens e do processo adaptativo.

No instante em que o coordenador recebe a confirmação de todos os membros do grupo informando-o sobre o término da última fase (desativação do antigo decodificador), ele irá executar as ações para finalizar o processo adaptativo.

O coordenador irá informar ao SPS a identificação do serviço de transformação de mensagens acordado pelo grupo, para que o SPS atualize seu registro e assim passe a informar para os novos membros do grupo sobre o novo serviço instalado.

O Gerenciador da Adaptação irá atualizar a etapa na qual o processo adaptativo está (REPOUSO). A etapa denominada REPOUSO indica que o membro do grupo está novamente pronto para executar um novo processo adaptativo.

Após todas essas ações, o SPS será desbloqueado para prover novamente serviços de transformação de mensagens para novos membros do grupo que venham a se conectar ao grupo. Além disso, o Gerenciador da Adaptação irá também se desbloquear para poder voltar a emitir novas convocações para eleição do contexto global.

4.4.3. Tratamento de Desconexão

O tratamento da desconexão nessa etapa é similar ao tratamento da desconexão na etapa de eleição do contexto global. O processo inicial de detecção de desconexão é o mesmo: o JGroups detecta que algum membro do grupo se desconectou e envia a nova visão do grupo para os membros do grupo. Se a nova visão do grupo contiver menos elementos que a visão anterior e a etapa do processo adaptativo for ADAPTAÇÃO, então os membros do grupo deverão tratar a desconexão dessa etapa, como será descrito adiante.

A única diferença no tratamento da desconexão entre essa etapa e na etapa de eleição do contexto global é que aqui precisa-se diferenciar entre a desconexão de um membro no papel de coordenador e de um membro no papel de participante.

Nesta etapa, o coordenador sempre esperará todos os participantes enviarem uma mensagem indicando a conclusão de uma fase para que a próxima seja

executada. Quando um participante se desconecta, o coordenador poderia esperar a confirmação do participante indefinidamente. Porém, esta etapa usa o mesmo mecanismo para tratamento de desconexão visto anteriormente. O JGroups envia a nova visão do grupo para os membros assim que detectar que um dispositivo se desconectou. Quando o coordenador receber a visão atualizada e detectar que esperava a confirmação de um dispositivo que está desconectado, ele irá prosseguir com a execução das próximas fases da etapa.

Para o caso da desconexão do coordenador optou-se pelo seguinte procedimento simples: assim que os membros do grupo detectam, através da nova visão do grupo, que o coordenador se desconectou, eles irão iniciar novamente toda a etapa de adaptação do serviço de transformação de mensagens, escolhendo um novo coordenador para o grupo. Na atual versão do SACS, este é o primeiro membro da visão do grupo.

Com relação à escolha do novo coordenador, é possível fazer otimizações do SACS. Em vez de escolher o próximo coordenador pela sua ordem na visão do grupo, poderia-se usar heurísticas para escolher de forma eficiente um novo coordenador comparando os valores locais de uma determinada variável de contexto em todos os membros da visão do grupo. Por exemplo, o coordenador poderia ser o processo com melhor conectividade (alto valor do sinal de radiofrequência) ou maior nível de energia residual. Dessa forma, poderia-se provavelmente minimizar a probabilidade de futura falha associadas ao novo coordenador.