

2

A Arquitetura MoGrid

2.1.

Introdução

Na etapa de definição do tema desta tese, foi proposta uma arquitetura em camadas, denominada MoGrid, que permite a distribuição de tarefas entre os dispositivos que constituem uma grade móvel. Essa arquitetura trata aspectos relacionados à computação distribuída em redes sem fio, constituídas por grupos de dispositivos móveis, que atuam, colaborativamente, para a execução de tarefas e utilização de serviços e recursos. As duas camadas que compõem a arquitetura MoGrid especificam os mecanismos associados à descoberta de serviços e ao acesso transparente aos mesmos. Através da abordagem em camadas, os diferentes mecanismos relacionados ao compartilhamento e à utilização de serviços são separados, podendo ser utilizados como módulos independentes, acoplados, inclusive em outras arquiteturas.

Um aspecto central para a arquitetura proposta é o mecanismo responsável pela descoberta de serviços, modelado através do protocolo P2PDP. Esse protocolo coordena a distribuição de tarefas entre os dispositivos sem fio, associando a “necessidade” por recursos com a “disponibilidade” dos mesmos, em cada possível colaborador na rede – o que caracteriza o conceito de *resource matching*. Durante o processo de especificação e implementação da arquitetura MoGrid, optou-se por concentrar o desenvolvimento desta tese na camada de descoberta de serviços, que, por si só, já possui uma complexidade considerável. A camada de transparência de acesso atua sobre os resultados obtidos através do processo de descoberta, garantindo a utilização dos serviços e recursos descobertos, em meio a períodos de desconexão de provedores de serviços e seus clientes. A decisão de focar o trabalho desenvolvido nesta tese na camada de descoberta tem como intuito oferecer um mecanismo robusto e confiável de descoberta de serviços, sob o qual a camada de transparência possa atuar.

2.2. Modelo de Serviços da Arquitetura MoGrid

Pode-se afirmar que um modelo de serviços para grades móveis deve considerar uma das seguintes abordagens: (i) os dispositivos sem fio atuando como interfaces de acesso aos recursos disponibilizados por grades computacionais convencionais ou (ii) os dispositivos sem fio atuando como elementos computacionais da grade disponibilizando os seus recursos e serviços. A segunda abordagem corresponde ao modelo de serviços adotado nesta tese. Esse modelo oferece suporte à categoria de aplicações que consideram a formação de grupos de dispositivos heterogêneos e autônomos em uma rede sem fio, em um modo *ad hoc*. Nessa linha, são apresentados diversos cenários de uso no levantamento efetuado em [Lima 2005], os quais ilustram o modelo de serviços adotado nesta tese.

Alguns aspectos devem ser considerados em função do modelo de serviços definido, os quais serão brevemente discutidos nos parágrafos subseqüentes.

- **Perfil dos dispositivos.** A arquitetura MoGrid foi desenvolvida considerando-se tanto dispositivos com restrições em relação aos recursos a serem compartilhados – caracterizados por apresentar conexões de rede intermitentes, largura de banda limitada, processadores lentos, visores de dimensões reduzidas, tempo de autonomia de bateria e memória limitados, como alguns modelos de celulares, *paggers*, e PDAs –, os quais atuam na arquitetura apenas como consumidores de recursos, ou seja, desempenhando o papel de iniciador das requisições de serviço, quanto dispositivos com uma configuração de *hardware* mais potente, apresentando uma maior disponibilidade de recursos e uma conectividade mais estável – como *notebooks* e modelos mais modernos de celulares e PDAs –, os quais desempenham simultaneamente os papéis de produtores e consumidores de recursos, atuando, respectivamente, como colaboradores e iniciadores. É importante ressaltar que a arquitetura MoGrid oferece suporte às plataformas Windows e Linux;
- **Suposições sobre a rede sem fio.** A arquitetura MoGrid é independente de um endereçamento de rede explícito, a identificação dos dispositivos é feita através de um identificador único associado dinamicamente a cada dispositivo, o que possibilita que a arquitetura opere sem a necessidade de

um protocolo de roteamento, adaptando-se com poucas modificações a qualquer tipo de rede sem fio. Os dispositivos devem ser interconectados através de enlaces sem fio bidirecionais, o que possibilita o envio e a recepção de dados entre dois dispositivos quaisquer na rede. Para possibilitar a detecção de vizinhança, os dispositivos devem ser capazes de transmitir pacotes de dados os seus vizinhos imediatos, por *broadcast* local. A latência de transmissão da rede pode interferir de forma negativa no desempenho da arquitetura MoGrid. Considerações a respeito de falhas e interferências relacionadas ao enlace sem fio encontram-se na Seção 2.4;

- **Perfil de mobilidade dos dispositivos.** A arquitetura MoGrid foi especificada para prover suporte a grupos de dispositivos móveis em cenários de mobilidade moderada que reproduzam a mobilidade humana, deslocando-se com uma velocidade média de até 4 m/s. Em cenários de alta mobilidade, as taxas de *churn* são muito elevadas, o que inviabiliza a formação de grupos espontâneos por um período de tempo suficiente para contemplar o processamento envolvido nas etapas de descoberta, seleção, submissão e execução de tarefas, etapas essas que caracterizam uma grade computacional, móvel ou fixa;
- **Suposições sobre os serviços.** A arquitetura MoGrid se adequa a execução de tarefas distribuídas paralelizadas, que não apresentem relações de interdependência, caracterizadas no ambiente de grades fixas por uma abordagem mestre-escravo, também conhecida como *bag-of-tasks*. Para deixar claro os conceitos de serviço, tarefa e recurso na arquitetura MoGrid, é utilizada a ilustração de um dispositivo que precisa executar uma bateria de simulações, considerando-se um cenário ideal. Para que a simulação seja executada, é necessário que os dispositivos na grade possuam instalado o simulador no qual a simulação foi desenvolvida. Nesse caso, o serviço solicitado é o “serviço de simulação”; as tarefas correspondem a cada rodada que compõe a bateria de simulações, as quais devem ser distribuídas entre os dispositivos mais aptos, ou seja, com mais recursos disponíveis; e os recursos – que no exemplo da simulação podem ser <potência de processamento>, <memória>

e <energia> – determinam em função da sua disponibilidade se os dispositivos estão aptos ou não a executar as rodadas de simulação;

- **Requisitos de QoS.** A arquitetura MoGrid deve prover um nível aceitável de QoS na execução dos serviços disponibilizados na grade móvel que reflita a percepção de qualidade do usuário do serviço. Para tanto, é necessário que os colaboradores ofereçam garantias mínimas sobre a disponibilidade de seus recursos ao responderem a uma requisição de descoberta de serviço. Essas garantias são expressas através de requisitos de QoS. Na arquitetura MoGrid esses requisitos são especificados em função das informações de contexto do dispositivo que disponibiliza o serviço. Ao receber uma requisição, o dispositivo avalia, em função das informações de contexto priorizadas pelo cliente, se ele está ou não em condições de oferecer o serviço. Por exemplo, ao submeter suas tarefas para execução em uma grade móvel, uma aplicação de longa duração, com consumo elevado de CPU, pode especificar os recursos <poder de processamento>, <memória>, <energia> e <largura de banda> como requisitos de execução. Além disso, a aplicação pode ainda estabelecer diferentes níveis de prioridade entre esses recursos, associando um peso relativo a cada um deles, podendo definir, por exemplo, que o requisito <energia disponível> é mais importante do que o requisito <disponibilidade de largura de banda> na escolha do dispositivo que irá executar a tarefa. No tratamento de requisições concorrentes, para que garantias mínimas de QoS sejam oferecidas, implementou-se um mecanismo de adaptação simples, que reduz o indicador da disposição do dispositivo em colaborar em função do aumento do número de requisições pendentes (vide Subseção 2.3.1.2 e Seção 4.5). É importante mencionar que no instante em que o serviço for de fato utilizado, o provedor do serviço não tem meios de garantir que as mesmas condições irão se verificar. Após a fase de descoberta, podem ocorrer alterações no estado de qualquer um dos recursos considerados, comprometendo a submissão ou execução do serviço solicitado, entretanto, esse tópico não é o foco deste trabalho. O tempo de descoberta aceitável foi especificado

em 10 s, que corresponde ao período que transcorre entre o envio da requisição e o recebimento das primeiras respostas pelo nó requisitante;

- **Segurança.** A questão da segurança é especialmente relevante na integração das grades móveis com as grades fixas. Entretanto, aspectos relacionados à segurança não foram especificados na arquitetura MoGrid, os quais devem ser incorporados como trabalhos futuros.

2.3. Um *Middleware* para Grades Móveis

A arquitetura MoGrid foi especificada tendo como finalidade oferecer suporte ao desenvolvimento de aplicações de grade em redes sem fio infra-estruturadas e *ad hoc*, possibilitando aos dispositivos sem fio executar tarefas computacionalmente complexas utilizando os recursos disponíveis na grade móvel. Nesse processo foram identificados dois aspectos principais: (i) a descoberta e seleção dos dispositivos mais aptos a executar as tarefas na grade móvel, em função das necessidades por recursos definidas na aplicação, e (ii) o gerenciamento da execução dessas tarefas, garantindo a sua conclusão mesmo mediante variações na qualidade do enlace sem fio e períodos de desconexão dos dispositivos da grade móvel. A arquitetura MoGrid compreende uma camada de descoberta *peer-to-peer* e uma camada de transparência, como se pode observar na Figura 2.

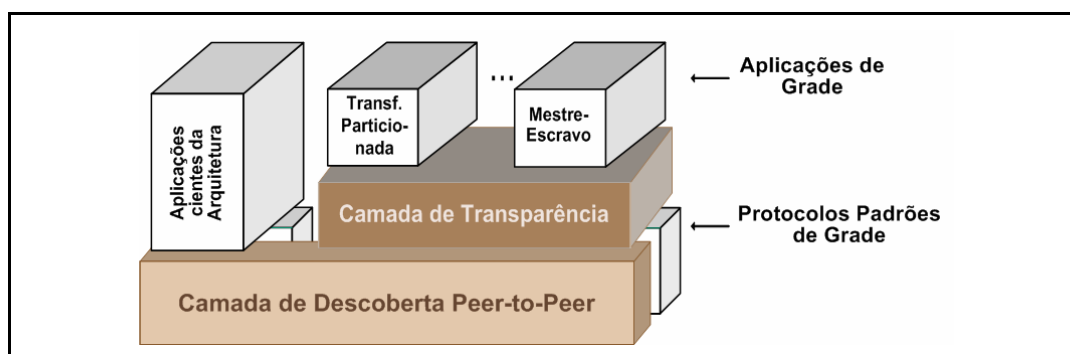


Figura 2 – Arquitetura MoGrid.

A arquitetura MoGrid possibilita a descoberta simultânea de múltiplas instâncias de um mesmo serviço, possibilitando a manutenção de um grau de redundância do mesmo. Essa abordagem é indicada para atender a demanda das grades móveis na execução de tarefas complexas, que demandam maior capacidade computacional, situação na qual, assim como nas grades

computacionais fixas, o serviço provido é o próprio poder de processamento dos dispositivos.

A arquitetura oferece suporte a dois tipos principais de aplicações: (i) aplicações de grade cientes da mobilidade dos dispositivos e (ii) aplicações convencionais de grade, como transferência particionada de arquivos e aplicações do tipo mestre-escravo, para as quais o uso da arquitetura deve ser transparente. O primeiro tipo de aplicação faz uso direto dos serviços oferecidos pela camada de descoberta e se responsabiliza, durante a submissão e execução de tarefas, pela gerência de questões relacionadas à conectividade irregular da grade móvel. No segundo tipo de aplicação, as questões são tratadas pela camada de transparência, que se responsabiliza também por acionar o serviço de descoberta.

2.3.1. Camada de Descoberta P2P

A camada de descoberta P2P é composta de três partes principais: a API de descoberta, as entidades envolvidas no processo de descoberta e o protocolo de descoberta e seleção de recursos P2PDP (*Peer-to-Peer Discovery Protocol*), o qual é apresentado em detalhes no Capítulo 4.

2.3.1.1. API de Descoberta

Aplicações executando em um dispositivo móvel devem usar as operações disponíveis na API de descoberta – diretamente ou através da camada de transparência – para usufruir dos benefícios dos mecanismos de descoberta e seleção de serviços da arquitetura. A API de descoberta provê, às aplicações, operações para registro de serviços, anúncio de serviços, especificação do perfil da requisição e descoberta de serviços. A Figura 3 mostra as principais operações desta API.

```
serviceID = register( serviceDescriptor )  
deregister( serviceID )  
advertise( serviceID, serviceDescriptor, timeToLive )  
reqProfile = createRequestProfile( ctxtInfo, numMaxReplies, maxReplyDelay )  
repList = discover( serviceQuery, reqProfile )
```

Figura 3 – API de descoberta.

Para que um serviço torne-se acessível na grade móvel, o dispositivo que provê o serviço faz o seu registro localmente, através da operação `register()`. Na arquitetura MoGrid, cada serviço é descrito por meio de um conjunto de atributos (`serviceDescriptor`), que é passado como parâmetro da operação `register()`. Esse conjunto de atributos define uma representação padronizada para todos os serviços na grade móvel através de um identificador único universal, uma descrição sucinta do serviço, um conjunto de palavras-chave e a sua localização, como ilustrado na Figura 4. O identificador único universal de um serviço é definido como um valor de 16 *bytes* gerado a partir da combinação da identificação do nó que o disponibiliza, do identificador mnemônico do serviço e do tempo corrente do sistema. Através da operação `deregister()`, é possível efetuar a remoção dos serviços registrados localmente, passando como parâmetro o seu identificador único universal (`serviceID`).

```
serviceDescriptor : < serviceID, serviceDescr, serviceKeywords, serviceLocation >
```

Figura 4 – Estrutura do parâmetro `serviceDescriptor`.

Um serviço registrado pode ser anunciado periodicamente na grade móvel (operação `advertise()`) – em uma abordagem proativa –, ou as informações sobre o novo serviço disponível podem ser obtidas sob demanda como resultado das requisições das aplicações para descoberta de serviços (operação `discover()`) – em uma abordagem reativa. Caso as duas abordagens sejam adotadas é caracterizado um mecanismo híbrido de descoberta de serviços.

No processo de descoberta, é necessária a descrição da requisição de serviço (`resourceQuery`), que é passada como parâmetro da operação `discover()`. A requisição contém uma descrição sucinta do serviço e um conjunto de palavras-chave associadas ao mesmo. Esses atributos são comparados, em cada dispositivo, com as informações sobre os serviços por ele oferecidos, informações estas registradas em uma estrutura de dados local, cujas entradas correspondem a pares do tipo `{serviceID, serviceDescriptor}`. A API permite ainda que as aplicações personalizem previamente suas requisições para descoberta de serviços (operação `createRequestProfile()`), gerando um perfil da requisição (`reqProfile`), que será informado na operação de descoberta (`discover()`). Essa personalização é útil quando são geradas várias requisições a serviços com as

mesmas características, evitando que as mesmas informações sejam passadas em cada requisição. O perfil da requisição define o número de colaboradores que o iniciador necessita envolver na execução do serviço (`numMaxReplies`; zero significa tantos colaboradores quanto os disponíveis na rede sem fio), por quanto tempo a aplicação está disposta a esperar por respostas (`maxReplyDelay`¹) e a informação de contexto de interesse (`ctxtInfo`). A informação de contexto de interesse determina quais recursos devem ser considerados em uma requisição e a importância relativa entre eles, sendo denominado de “perfil de execução”. As informações contidas no perfil de execução são utilizadas no cálculo do retardo programado associado ao envio das mensagens de resposta do protocolo de descoberta. O retardo gerado é inversamente proporcional à disponibilidade dos recursos do dispositivo que são necessários para a provisão do serviço solicitado. Esse parâmetro é utilizado no processo de seleção para medir a qualidade das respostas encaminhadas, identificando o grupo de colaboradores mais aptos, dentre aqueles que se prontificaram a oferecer o serviço.

O conceito de perfil de execução proposto na arquitetura MoGrid é similar ao conceito de “*resource matching*” [Tangmunarunkit et al. 2003; Farooq & Khalil 2006] da área de grades computacionais, o qual define a seleção do grupo de dispositivos que irá colaborar na execução de um conjunto de tarefas, com base nos recursos disponíveis em cada elemento do grupo, considerando-se os requisitos da aplicação. A informação de perfil de execução é um parâmetro importante durante o processo de seleção na arquitetura MoGrid, pois reflete os requisitos do serviço que devem ser considerados ao se medir a qualidade das mensagens de resposta a uma dada requisição. Contudo, após a fase de descoberta, podem ocorrer alterações no estado de qualquer um dos recursos considerados no perfil de execução de um dado serviço que venham a comprometer a submissão ou execução das tarefas. As aplicações que fazem uso direto dos serviços oferecidos pela camada de descoberta se responsabilizam, durante a submissão e execução de tarefas, pela gerência de questões relacionadas à conectividade irregular da grade móvel. Por outro lado, para as aplicações que não acessam

¹ Esse valor de retardo pode ser definido estaticamente ou ajustado dinamicamente através de um algoritmo adaptativo, baseado em estimativas de RTT (*Round Trip Time*).

diretamente o serviço de descoberta (vide Figura 2 da Seção 2.3), o tratamento desse tipo de condição é feito pela camada de transparência, que aciona esse serviço em benefício da aplicação. As funções da camada de transparência são descritas em detalhes na Subseção 2.3.2.

2.3.1.2. Entidades de Descoberta

A forma como as entidades envolvidas no processo de descoberta de serviços se relacionam, bem como a interação entre elas, através da troca de mensagens do protocolo de descoberta P2P (P2PDP), é ilustrada na Figura 5. Essa dinâmica é representada em dois cenários distintos: em redes sem fio *ad hoc* [Figura 5(a)] e em redes sem fio infra-estruturadas [Figura 5(b)].

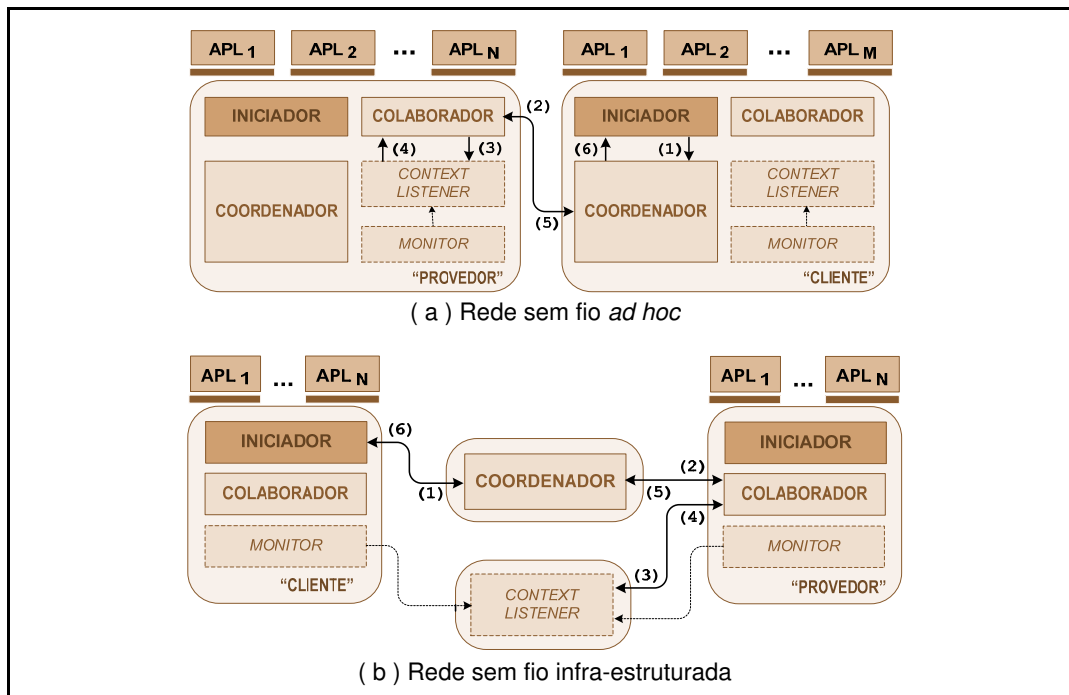


Figura 5 – Entidades envolvidas no processo de descoberta.

A camada de descoberta P2P define três entidades principais – colaboradores, iniciadores e coordenadores – que correspondem aos diferentes papéis que um dispositivo pode desempenhar no processo de descoberta de serviços, como ilustrado na Figura 5. Colaboradores estão disponíveis para a execução de tarefas da grade móvel. Um dispositivo móvel está apto a atuar como colaborador após registrar seus serviços localmente (operação `register()`). Iniciadores submetem, aos colaboradores, as requisições por serviços e o perfil de

execução desses serviços, que especifica quais recursos são necessários para o processamento de tarefas da aplicação. Qualquer dispositivo móvel pode ser um iniciador em uma grade móvel. Finalmente, coordenadores atuam como intermediadores entre iniciadores e colaboradores. Coordenadores encaminham as requisições dos iniciadores para descoberta de serviços aos colaboradores e, com base nas respostas recebidas, disponibilizam, aos iniciadores, uma lista contendo os colaboradores mais apropriados. É importante ressaltar que, embora o foco principal desta tese sejam as grades móveis implantadas em redes sem fio *ad hoc* [Figura 5(a)], a arquitetura MoGrid também contempla a implementação do serviço em uma rede com um coordenador centralizado, se uma rede sem fio infra-estruturada estiver disponível. Nesse tipo de rede, um ou mais dispositivos conectados à rede fixa atuam como coordenadores da grade móvel, gerenciando as requisições de todos os dispositivos sem fio. Isso evita que os dispositivos móveis sejam sobrecarregados devido ao processamento adicional das funções associadas à coordenação de suas requisições, que é o comportamento padrão em uma rede sem fio *ad hoc*. A Figura 5(b) ilustra esse cenário alternativo.

Dois serviços desempenham funções importantes na arquitetura proposta: o monitor e o gerente de contexto (*ContextListener*). Cada dispositivo, na grade móvel, possui um monitor residente. O serviço monitor é responsável por coletar a informação de contexto dos dispositivos móveis, incluindo a qualidade do enlace sem fio, carga de CPU, carga residual da bateria, memória disponível e espaço de armazenamento. Em uma rede sem fio *ad hoc*, cada dispositivo possui também um gerente de contexto residente [Figura 5(a)]; em contraste, uma rede sem fio infra-estruturada pode ter um gerente de contexto centralizado [Figura 5(b)]. O serviço de gerente de contexto recebe, periodicamente, do serviço monitor, a informação de estado coletada, e deduz, a partir de tal informação, a disponibilidade de recursos dos dispositivos. Quando um coordenador consulta os colaboradores sobre a disponibilidade de recursos de seus dispositivos, para a execução de uma tarefa, os colaboradores interagem com o serviço de gerenciamento de contexto para verificar se eles oferecem o perfil de execução necessário para participar na execução da tarefa.

Basicamente, um colaborador adota dois critérios para decidir se está apto a prover o iniciador com o serviço solicitado. O primeiro critério representa um

parâmetro subjetivo que indica a disposição do usuário em participar ou não da provisão do serviço, de acordo com a sua “boa vontade” em colaborar (*willingness*). O segundo critério atua como um “controlador de admissão” (*AdmissionController*), verificando se o colaborador pode disponibilizar o serviço solicitado pelo iniciador e a sua “adequação” (*suitability*) para participar na execução das tarefas nas quais o serviço é particionado, considerando o seu nível de comprometimento com as demais requisições às quais tenha respondido. Fundamentalmente, a adequação de um colaborador é medida de acordo com o perfil de execução do serviço provido pelo iniciador que efetuou a requisição (`ctxtInfo` na operação `createRequestProfile()`). Retomando o exemplo de uma aplicação de longa duração, que exija um processamento intensivo, colaboradores com níveis mais elevados de energia tipicamente serão selecionados para executar uma tarefa. O mecanismo de seleção é implementado de forma distribuída no protocolo P2PDP, que é apresentado em detalhes no Capítulo 4.

O tratamento de requisições concorrentes merece ser discutido. Caso um colaborador tenha respondido a uma dada requisição, mas a execução do serviço ainda não tenha sido iniciada – por exemplo, a execução de uma rodada de simulação –, essa resposta representa uma reserva antecipada de recursos para a execução do serviço [Wolf et al. 1995; Smith et al. 2000], caracterizando o compromisso do dispositivo em atender à requisição. A validade dessa reserva é indicada quando a requisição é gerada, sendo definida através do parâmetro `maxReplyDelay`, que especifica o tempo que a aplicação está disposta a esperar por respostas. Ao receber uma nova requisição, o colaborador deve considerar o seu nível de comprometimento com as demais requisições já respondidas como um dos critérios determinantes para se definir a sua aptidão. Esse aspecto é fundamental para oferecer alguma garantia² quanto à qualidade do serviço que está sendo entregue e, em situações extremas, que o serviço seja, de fato, disponibilizado, mesmo que a qualidade na qual ele esteja sendo entregue não

² As garantias oferecidas na entrega de um serviço em ambientes computacionais móveis não são garantias absolutas, pois estão condicionadas às peculiaridades desses ambientes, como a qualidade do enlace sem fio, a disponibilidade de largura de banda e a mobilidade dos dispositivos.

corresponda integralmente as expectativas do usuário. O nível de comprometimento de um colaborador mede o quanto os recursos necessários para a entrega do serviço solicitado já estão comprometidos com a execução de outras requisições, impedindo que ele responda a mais requisições do que é capaz de atender. O monitoramento das condições do dispositivo é feito pelo serviço de gerenciamento de contexto.

Coordenadores tratam os casos de requisições não completamente atendidas, em que são recebidas menos respostas do que o esperado – devido, por exemplo, a respostas perdidas na rede. Em tais casos, a aplicação (ou a camada de transparência) que disparou a requisição do iniciador terá que lidar com essa situação através do envio de uma requisição adicional ao protocolo de descoberta ou da notificação de falha no processo de descoberta ao cliente da aplicação. Um cenário oposto ocorre quando o coordenador recebe mais respostas do que solicitado pelo iniciador. Em tais casos, o coordenador seleciona as respostas mais apropriadas, com base em critérios pré-estabelecidos, antes de encaminhá-las para o iniciador interessado. Esses critérios podem ser definidos pelo usuário, de acordo com as características dos serviços que ele usualmente solicita, ou, então, podem ser adotados os critérios-padrão definidos na arquitetura MoGrid. A abordagem mais comumente utilizada, na seleção de respostas a uma solicitação de descoberta, adota a distância, em número de saltos, entre o cliente e o provedor do serviço como critério de decisão [Lenders et al. 2005; Varshavsky et al. 2005]. Na arquitetura MoGrid, dois critérios são utilizados para selecionar as melhores respostas caso o coordenador receba mais respostas do que as que foram solicitadas: a distância, em número de saltos, do colaborador até o iniciador, e o perfil de execução contido na resposta do colaborador. Essa seleção é feita de forma automática pelo coordenador em benefício do iniciador.

2.3.2. Camada de Transparência

A camada de transparência é responsável por gerenciar as questões relacionadas à conectividade irregular da grade móvel, decorrente de períodos de desconexão, provocados pela instabilidade do enlace ou mesmo pela indisponibilidade dos provedores durante a fase de descoberta de recursos e, posteriormente, de

submissão e execução de tarefas. A camada de transparência é composta pela subcamada de acesso transparente aos recursos e pelas subcamadas de adaptação, como ilustrado na Figura 6.

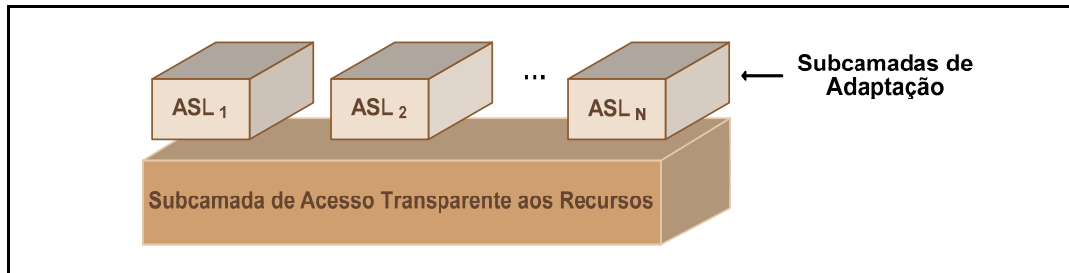


Figura 6 – Arquitetura MoGrid: a camada de transparência.

2.3.2.1.

Subcamada de Acesso Transparente aos Recursos

Considerando-se o fator mobilidade, simplificando a definição apresentada em [Peddermors et al., 2004], existem dois possíveis comportamentos que uma aplicação pode adotar, o que influenciará no nível de transparência de acesso oferecido: (i) aplicações que não têm consciência da mobilidade e, conseqüentemente, não gerenciam o problema da desconexão; e (ii) aplicações que têm consciência da mobilidade e tratam o gerenciamento de desconexão na própria aplicação. Para oferecer suporte ao gerenciamento de desconexão às aplicações que se comportam como descrito em (i), foi idealizada a subcamada de acesso transparente aos recursos (*Transparent Resource Access Sublayer – TRAS*).

Normalmente, após receber a lista dos colaboradores mais aptos, em resposta à uma requisição de serviço, o iniciador, que efetuou a requisição, pode realizar a distribuição de tarefas, entre eles, usando um protocolo padrão, como, por exemplo, o GridFTP [Foster & Kesselman 1997] ou alguma implementação projetada para atender às necessidades da aplicação cliente. Cada colaborador, após executar completamente uma tarefa, envia os resultados obtidos para o iniciador. No entanto, durante a submissão ou execução de tarefas, o iniciador, ou qualquer dos colaboradores, pode passar por um período de desconexão, que pode ser voluntária (por exemplo, dispositivo sendo deslocado ou entrando no modo de economia de energia) ou involuntária (como a perda abrupta de conectividade).

Quando o coordenador é centralizado [Figura 5(b)], ele pode lidar com a desconexão dos iniciadores, atuando como um *proxy*, em benefício dos mesmos, armazenando os resultados enviados pelos colaboradores, até que a conexão com o iniciador possa ser restabelecida, quando, então, ele entrega os resultados obtidos. Vale ressaltar a necessidade de se definir políticas, no coordenador, que estabeleçam critérios para a manutenção do armazenamento desses resultados. Essas políticas podem ser definidas localmente, onde cada coordenador determina a sua, ou podem ser definidas de forma global, nesse caso, todos os dispositivos da rede adotam o mesmo conjunto de políticas. No caso da desconexão dos colaboradores, o coordenador atua em benefício do iniciador que efetuou a requisição, selecionando novos colaboradores, com o auxílio do protocolo P2PDP. Quando as desconexões dos colaboradores são voluntárias, independentemente do coordenador ser centralizado ou distribuído, mecanismos para migração de tarefas, entre colaboradores, podem ser empregados [Milojicic et al. 2000]. Os colaboradores podem enviar, para o coordenador, uma mensagem de notificação de desconexão, incluindo, nessa mensagem, informações sobre o estado de execução da tarefa que cada um estava executando. Ao receber essa notificação, o coordenador aciona o processo de seleção de um novo colaborador, atuando como um intermediador no processo de migração da execução da tarefa.

Para coordenadores descentralizados [Figura 5 (a)], como ocorre tipicamente no caso das redes sem fio *ad hoc*, algumas considerações adicionais devem ser feitas. O restante desta seção tem como foco os requisitos de projeto da TRAS para oferecer suporte à desconexão das entidades de descoberta, em redes sem fio *ad hoc*, e a forma como ela se relaciona com o protocolo P2PDP.

No caso de desconexão voluntária do iniciador, ele notifica antecipadamente esse evento aos seus colaboradores atuais. Quando finalizam a execução da tarefa, os colaboradores notificados armazenam os resultados correspondentes até que o iniciador anuncie que está novamente conectado, ou até que um temporizador (*waiting-for-initiator*), disparado no instante em que a tarefa é concluída, expire. Nesse último caso, os colaboradores descartam os resultados que foram armazenados. Cada colaborador define o seu temporizador de forma independente em relação aos demais colaboradores.

Quando um colaborador detecta que irá se desconectar, ele suspende as tarefas em andamento e notifica esse evento a todos os iniciadores com os quais ele está cooperando. A partir desse ponto, dependendo do tempo de conexão restante e da disponibilidade de seus recursos, o colaborador pode apresentar dois comportamentos distintos: (i) ele descarta os resultados parciais obtidos e transfere a responsabilidade de selecionar um novo colaborador, para reiniciar a execução da tarefa, ao iniciador, que, para isso, acionará o coordenador, ou (ii) ele se encarrega de fazer a migração do estado atual da sua execução para um novo colaborador, o qual deve retomar a execução da tarefa do ponto em que ela foi interrompida. Na primeira abordagem, em resposta à notificação de desconexão, o iniciador dispara o envio de mensagens P2PDP $IReq$ para a seleção de um novo colaborador. Só então, após receber o resultado indicando quem foi o colaborador selecionado, a tarefa é resubmetida, sendo executada desde o início. Na segunda abordagem, o colaborador utiliza o protocolo P2PDP para selecionar um substituto, restringindo a consulta a sua vizinhança física, ou seja, aos dispositivos que estão conectados diretamente ao colaborador. Essa estratégia é indicada para reduzir o consumo de energia adicional provocado pela inundação da rede com a difusão de uma nova mensagem de requisição do iniciador para encontrar um substituto ao colaborador que se tornou indisponível. O colaborador atual transfere a tarefa para o novo colaborador, juntamente com o seu estado de execução, gerenciando o processo de migração através da camada de transparência. Ao finalizar a execução da tarefa, o colaborador substituto entrega o resultado da execução ao iniciador. Caso a seleção de um novo colaborador não seja bem sucedida, o colaborador se comporta como descrito na primeira abordagem, transferindo a responsabilidade da seleção de um novo colaborador para o iniciador.

Para a detecção de possíveis desconexões involuntárias, tanto para o iniciador quanto para o colaborador, a camada de transparência pode monitorar os eventos, processados pelo serviço de gerenciamento de contexto, relacionados à qualidade do enlace sem fio e a sua reserva de energia; inferindo, a partir desses recursos, um estado de desconexão iminente. Ao detectar esse estado, o gerente de contexto notifica a entidade de descoberta, que executa os mesmos procedimentos acionados em uma desconexão voluntária. A implementação de um mecanismo

eficiente de detecção de desconexões involuntárias apresenta uma série de dificuldades. A detecção pode não ser imediata, impossibilitando que a entidade de descoberta seja notificada em tempo hábil para acionar os procedimentos de desconexão necessários. Em uma situação oposta, o dispositivo pode apresentar uma indisponibilidade temporária de seus recursos. Nesse quadro, o gerente de contexto pode interpretar essa indisponibilidade como um evento de desconexão involuntária e notificá-lo à entidade de descoberta, que aciona, então, os procedimentos de desconexão, antecipando uma situação que, nesse caso, não vai acontecer. É importante mencionar, que subcamadas de adaptação distintas podem aplicar tratamentos diferenciados para lidar com os períodos de desconexão involuntária das entidades de descoberta, como discutido na Subseção 2.3.2.3. Esses tratamentos são determinados pelas características específicas da família de aplicações à qual cada subcamada de adaptação provê suporte.

2.3.2.2.

Requisitos da Camada de Acesso Transparente aos Recursos

Para prover suporte ao gerenciamento das desconexões involuntárias, é necessário que se defina um outro protocolo – complementar ao P2PDP – no âmbito da camada de transparência. Nesta subseção, é apresentada uma discussão quanto aos requisitos necessários para auxiliar na especificação do protocolo de acesso transparente aos recursos (*Transparent Resource Access Protocol – TRAP*). A especificação desse protocolo tem o propósito de definir um nível aceitável de qualidade na execução dos serviços descobertos na grade móvel, definido de forma subjetiva, em função da percepção do usuário do serviço. Para tanto, é necessário que os colaboradores ofereçam garantias mínimas sobre a disponibilidade de seus recursos ao responderem a uma requisição de serviço. O projeto e a implementação do TRAP não são o foco desta tese; entretanto, os próximos parágrafos trazem uma discussão sobre os requisitos básicos que o protocolo de acesso deve atender para que os serviços descobertos na grade móvel sejam utilizados, independentemente de desconexões involuntárias.

Mecanismo de comunicação. É necessário definir como a comunicação entre o cliente e o provedor de serviço será estabelecida. Duas abordagens principais podem ser adotadas. Na primeira, o mecanismo de comunicação

utilizará o protocolo de roteamento adotado na MANET para promover o encaminhamento de mensagens de invocação do serviço e de entrega do resultado da sua execução. Na segunda abordagem, indicada para cenários de baixa mobilidade, o TRAP pode manter uma estrutura de dados, denominada tabela de execuções pendentes (TEP), com informações sobre o caminho entre a origem (colaborador) e o destino (iniciador) do serviço descoberto. Essas informações permitem que as mensagens de invocação e de resultado sejam encaminhadas, salto-a-salto, do cliente até o provedor, e vice-versa, através das informações armazenadas na tabela mantida, localmente, pelos nós intermediários. Em face a períodos de alta mobilidade, apenas a primeira abordagem seria viável, dado que a manutenção da TEP exigiria um mecanismo de atualização, acionado em função da quebras de rotas. Esse mecanismo faria uso do protocolo P2PDP para descobrir um novo caminho entre a origem e o destino. Nesse caso, o serviço requisitado seria a localização do destino.

Monitoramento do provedor do serviço. O TRAP deve fornecer garantias que o provedor do serviço selecionado desempenhe as suas funcionalidades com um nível de qualidade que não extrapole um limite mínimo de tolerância (QoS_{min}), oferecendo, com uma certa probabilidade, a garantia de que o resultado gerado atenda às necessidades do cliente. Para garantir essa propriedade, um mecanismo de monitoramento da disponibilidade dos recursos do provedor do serviço deve ser incorporado ao TRAP. Os recursos monitorados seriam aqueles definidos na requisição do serviço, em função das características da aplicação. Esse monitor notificaria ao TRAP caso o provedor atingisse um nível de disponibilidade de recursos insatisfatório para a execução do serviço. Com essa informação, o TRAP poderia, automaticamente, acionar o *mecanismo de redescoberta* e dar início à *migração de serviço* ou, simplesmente, comunicar ao cliente (iniciador) a indisponibilidade do provedor (colaborador). A ação a ser executada, em resposta ao recebimento de uma notificação de indisponibilidade do provedor, é definida na subcamada de adaptação, através do mecanismo de *callbacks*. O resultado do monitoramento dos provedores de serviço também pode ser utilizado na implementação de um mecanismo de reputação [Rheingold 2003]. O funcionamento desse mecanismo se baseia na preocupação que os usuários de sistemas computacionais – como aqueles que constituem uma grade móvel – têm

em manter a sua reputação elevada, evitando assim qualquer comportamento que reduza o seu conceito perante os demais membros do grupo. Esse fator pode ainda ser utilizado para adaptar o parâmetro que indica a disposição de um colaborador em participar na execução de um serviço, em função da reputação do dispositivo requisitando o serviço; caso a reputação do iniciador seja considerada insatisfatória, o colaborador pode se negar a atender a sua requisição de serviço. É importante levar em consideração, na implementação do mecanismo de monitoramento, que deve ser definido o que caracteriza uma indisponibilidade real de uma flutuação temporária na disponibilidade de recursos do provedor, evitando que notificações precipitadas sejam disparadas. Esse é um problema não trivial, intrínseco às redes sem fio, e para que o mecanismo de monitoramento seja implementado, é preciso que se faça uma investigação aprofundada sobre o tema.

Manutenção do estado da execução. Para que haja *migração de serviço* entre os colaboradores, é necessário que o estado da sua execução possa ser recuperado para que a nova instância do serviço retome a sua execução a partir do ponto em que ela foi interrompida. Segundo Milojevic et al. [2000], o estado de execução inclui o espaço de endereçamento do processo, o ponto de execução – conteúdo dos registradores –, o estado da comunicação – como arquivos abertos e canais de mensagens que foram estabelecidos – e estados dependentes do sistema operacional. É importante mencionar a necessidade de se definir operações para interromper e reiniciar um serviço. Para isso, é obrigatório que a implementação do serviço ofereça algum tipo de interface que possibilite as funções de exportação e importação para extrair o estado de execução do serviço do provedor de origem e recuperá-lo no provedor de destino. Essas interfaces normalmente são providas pelo sistema operacional ou por bibliotecas específicas associadas à linguagem de programação utilizada.

Migração de serviço. Esse requisito diz respeito à transferência do serviço, entre dois provedores – a origem e o destino – durante a sua execução. A transferência inclui a migração do serviço e do seu estado de execução. A camada TRAS requer um mecanismo de migração de processos para os serviços *statefull*. Existem diversos trabalhos que tratam essa questão, adotando abordagens diferenciadas. A implementação desse mecanismo exige uma investigação metódica sobre o tema. Levando em consideração o algoritmo definido por

Milojicic et al. [2000], na camada de acesso transparente, a migração de um serviço deve cumprir as seguintes etapas: (i) a migração é requisitada em função da notificação de uma indisponibilidade iminente do colaborador (provedor do serviço); (ii) é efetuada a escolha do novo colaborador, dando início a um novo processo de descoberta através do protocolo P2PDP; (iii) a execução do serviço é suspensa no colaborador de origem; (iv) os canais de comunicação utilizados pelo serviço são estabelecidos no novo colaborador; (v) o estado do serviço é extraído da instância em suspensão no colaborador de origem; (vi) uma instância do serviço é criada no novo colaborador; (vii) o estado de execução do serviço, que foi transferido, é importado pela instância em execução no novo colaborador e (viii) a nova instância do serviço pode então ter a sua execução retomada do ponto em que foi interrompida, e a instância do serviço, no colaborador de origem, pode finalmente ser removida.

Mecanismo de redescoberta. Caso seja necessário substituir um colaborador durante a execução de um dado serviço, torna-se necessário acionar o mecanismo de redescoberta de colaboradores para que a execução do serviço possa ser concluída no substituto. Para que esse requisito seja alcançado, é preciso acionar, mais uma vez, o protocolo de descoberta P2PDP. Essa etapa pode ser gerenciada pelo provedor do serviço (colaborador P2PDP) ou pelo próprio cliente (iniciador P2PDP), dependendo da ação disparada pelo protocolo TRAP quando o colaborador é notificado sobre um evento de indisponibilidade do serviço em execução. O colaborador pode acionar automaticamente o mecanismo de redescoberta ou simplesmente comunicar ao iniciador que houve uma falha na execução do serviço, transferindo para ele essa responsabilidade. Uma outra abordagem seria acionar o mecanismo de redescoberta, automaticamente, quando um provedor de serviço enviasse ao iniciador uma mensagem de notificação de indisponibilidade. Essa mensagem seria, então, encaminhada por *broadcast* limitado, na grade móvel, e encapsularia uma mensagem de requisição P2PDP (IReq). Ao receber uma notificação destinada a um outro dispositivo, os nós intermediários, no caminho entre o antigo colaborador e o iniciador, respondem ao iniciador, caso se adequem ao perfil de execução do serviço, encapsulando a mensagem de resposta (CRep) do protocolo P2PDP na notificação de indisponibilidade. Nesse caso, ao receber a notificação de indisponibilidade do

antigo colaborador, o iniciador receberá também uma lista com os novos colaboradores.

2.3.2.3. Subcamadas de Adaptação

Em contraste com as aplicações cientes da utilização da arquitetura, as aplicações-padrão de grades fixas não têm conhecimento de detalhes particulares da API de descoberta. Para oferecer suporte a essas aplicações, a camada de transparência estabelece a cooperação entre o iniciador e os seus colaboradores potenciais, utilizando o protocolo P2PDP, através das funcionalidades oferecidas pelas subcamadas de adaptação (*Adaptation SubLayers* – ASLs). Como já foi mencionado, a subcamada TRAS provê mecanismos independentes da aplicação que garantem a transparência no acesso e utilização dos recursos; no entanto, cada tipo de aplicação, em uma grade móvel, apresenta requisitos específicos. Por exemplo, aplicações baseadas no paradigma de computação paralela mestre-escravo demandam maior poder de processamento, mas são menos suscetíveis à conectividade intermitente. De um modo geral, o resultado do processamento desse tipo de aplicação gera um volume de dados pequeno que pode ser transferido em um breve período de conectividade estável, exigindo pouco do meio sem fio em termos de transmissão de dados. Entretanto, aplicações dessa mesma categoria, que gerem resultados volumosos, irão demandar, também, estabilidade da conexão e carga residual da bateria, devido à necessidade de transmissão de uma quantidade de dados considerável. Por outro lado, para aplicações que efetuam replicação de dados, requisitos como capacidade de armazenamento, estabilidade da conexão e carga residual da bateria são mais importantes, dado o grande volume de dados transferidos no meio sem fio. O processo de replicação implica na necessidade de espaço físico para armazenar os dados e uma conectividade estável para lidar com uma taxa de transmissão constante, responsável por um aumento no consumo de energia. O principal propósito das subcamadas de adaptação é implementar o tratamento dos eventos relacionados à mobilidade e à desconexão da forma que melhor se adeque a cada tipo de aplicação, implementando os procedimentos que devem ser acionados em caso de notificação de eventos de exceção.

Cada subcamada de adaptação deve implementar algumas operações de *callback* que definem o comportamento esperado dos iniciadores, coordenadores e colaboradores no caso de eventos de exceção, como requisições não atendidas ou desconexões voluntárias e, quando possível, exceções decorrentes da detecção de desconexões involuntárias. Apesar de não trivial, idealmente, a subcamada TRAS deve, ao detectar qualquer evento de exceção, identificá-lo e acionar a *callback* correspondente. Ao se implementar uma ASL específica, é fundamental que se definam os possíveis eventos de exceção e se ofereçam mecanismos de identificação dos mesmos. Além disso, deve-se realizar a associação entre esses eventos e as *callbacks* correspondentes, que devem ser implementadas na ASL. No caso de eventos determinísticos, como desconexão voluntária e requisições não atendidas, é fácil fazer essa associação. O mesmo não acontece no caso dos eventos cuja detecção está associada ao monitoramento das condições da rede e da variação da disponibilidade de recursos no próprio dispositivo, como desconexões involuntárias e redução na qualidade do serviço oferecido. Nessa situação, deve-se definir condições limites que caracterizem a exceção – como intervalos temporais nos quais as variações são permitidas, valores mínimos que cada recurso pode assumir em um dado intervalo – além de se permitir a composição dessas condições considerando-se diferentes recursos, definindo, por exemplo, para a detecção de um evento de redução de QoS, que a carga residual da bateria não pode ser inferior a 50% e a carga de processamento não pode se manter abaixo de 40% por um período de tempo superior a um minuto. Nesse cenário, quando uma entidade não consegue efetuar o mapeamento entre uma exceção específica e a sua *callback* de tratamento, a aplicação deve, então, ser notificada imediatamente. A caracterização das exceções não é trivial e requer conhecimento a respeito dos requisitos da aplicação cliente.

2.4. Os Modelos de Falhas da Arquitetura MoGrid

Falhas podem ter diversas origens, que vão desde causas físicas – como rompimento de uma trilha em um circuito eletrônico ou ruídos na rede elétrica –, até falhas de origem humana, quer sejam intencionais ou não – como erros de projeto ou mesmo erros causados pelo mau uso dos sistemas. Em um cenário de comunicação sem fio, existem ainda os problemas provocados pela perda de

pacotes de dados e desconexões constantes, problemas esses que tornam o sistema ainda mais propenso à manifestação de falhas. Independentemente da sua origem, em um sistema de tempo real distribuído, uma falha pode ser classificada de acordo com o comportamento do componente que a detecta após a sua ocorrência, como descrito em [Chow & Johnson 1997]. Os modelos de falhas considerados serão analisados nos próximos parágrafos em função dos dispositivos sem fio e dos canais de comunicação da grade móvel. Como o colaborador desempenha o papel mais crítico da arquitetura, os modelos de falhas serão ilustrados em função do comportamento deste componente, tendo como foco a camada de descoberta da arquitetura MoGrid.

Fail-Stop. Um dispositivo atuando como um colaborador da grade móvel é desativado, quer seja por intervenção humana ou por alguma falha no equipamento, deixando de responder a requisições de serviço e de executar as requisições com as quais já tenha se comprometido. Nesse caso, o dispositivo não executará operações inválidas e simplesmente deixará de funcionar, o que não interferirá no funcionamento da arquitetura como um todo, pois os demais dispositivos continuarão desempenhando as suas funções corretamente e irão detectar, com o tempo, a sua indisponibilidade. Existe ainda a possibilidade de um colaborador que faz parte do caminho de retorno de uma mensagem de resposta deixar de funcionar, provocando a quebra da rota estabelecida no processo de descoberta entre um candidato a colaborador e o requisitante do serviço. Nesse caso, se o mecanismo de reconhecimento do envio de respostas não conseguir recuperar a falha ocorrida no encaminhamento da resposta, a mesma não será entregue ao nó requisitante, podendo ser substituída por uma resposta de qualidade inferior, proveniente de um colaborador com menos recursos mas cujo caminho reverso tenha se mantido durante o período de tempo transcorrido entre as etapas de requisição e descoberta do serviço.

Falha Bizantina. As falhas bizantinas ou arbitrárias pressupõe um comportamento incorreto dos componentes do sistema. Na arquitetura MoGrid, as entidades de descoberta poderiam gerar mensagens com erro ou mesmo maliciosas que comprometeriam o funcionamento do protocolo. Além disso, os nós poderiam gerar mensagens falsas aleatoriamente, como, por exemplo, iniciadores emitindo requisições de serviço com uma frequência elevada ou

colaboradores respondendo a requisições de serviço às quais não estejam aptos a atender. Na arquitetura MoGrid esse tipo de falha não é tratado. Os protocolos especificados para solucionar o problema do consenso apresentam uma complexidade computacional extremamente elevada e, em muitos casos, especialmente em redes instáveis como as redes sem fio *ad hoc*, apresentam problemas de convergência [Angluin et al. 2006].

Falhas de Omissão. Falhas como desconexões voluntárias e involuntárias de dispositivos – provocadas por quebras de rotas em função do particionamento da rede, atenuação do sinal no enlace sem fio, o problema do terminal oculto, entre outros – são responsáveis por fazer com que um ou mais componentes da arquitetura fiquem incomunicáveis por períodos variáveis de tempo. Em decorrência dessas falhas de comunicação, os dispositivos ficam impossibilitados de responder a novas requisições de serviço ou mesmo executar as requisições as quais tenham respondido anteriormente – quando atuando como colaboradores – e de submeter tarefas para a grade móvel e de receber os resultados da execução das mesmas – quando atuando como iniciadores. Nesse caso, as conseqüências das falhas são mais graves.

Falhas Temporais. Esse tipo de falha pode ocorrer em função da falta de sincronização de relógios entre os dispositivos da grade ou em função de variações abruptas da latência do sinal do enlace sem fio. Na arquitetura MoGrid, respostas mais adequadas, provenientes de dispositivos com mais recursos, que tenham sido atrasadas ou mesmo estejam sendo transmitidas no tempo certo podem ser suprimidas por respostas menos aptas. Essa situação se verifica caso os dispositivos que originaram essas respostas antecipem o envio das mesmas ou elas trafeguem através de enlaces com menor latência. Nesse caso, o requisitante não deixará de ser atendido, entretanto, o serviço disponibilizado não será o de melhor qualidade em função das falhas temporais que ocorreram, prejudicando o processo de seleção das respostas. Analisando as falhas temporais, tem-se que a principal demanda da camada de descoberta, em relação às camadas inferiores, diz respeito ao mecanismo de sincronização de relógios entre os dispositivos que constituem a grade móvel. Para o correto funcionamento da arquitetura, é necessário que a diferença de velocidade entre os relógios dos dispositivos seja desprezível.

O conceito básico para se implementar tolerância a falhas é redundância, de *hardware* ou *software*. Em termos gerais, a redundância pode assumir dois formatos: temporal ou espacial. A redundância temporal implica que após a detecção da falha, uma ação corretiva seja acionada como, por exemplo, a repetição do procedimento que gerou a falha. Esse tipo de redundância é em geral menos aplicável a sistemas críticos, já que o procedimento de recuperação pode violar as especificações rígidas de tempo (*hard deadlines*). No caso da redundância espacial, uma mesma ação crítica é executada por vários componentes replicados. Nesse caso, existe a possibilidade da falha ser mascarada no sentido de que a falha de um componente é compensada pela ação correta de outros componentes replicados. A redundância espacial é adequada para sistemas críticos, por não gerar atrasos extras nos processos de detecção e recuperação de erros [Macêdo et al. 2004]. Na arquitetura MoGrid, o mecanismo de tolerância a falhas da camada de descoberta pode ser implementado através de redundância espacial, onde várias instâncias do mesmo serviço são selecionadas para atender a uma requisição específica. Na camada de transparência, o mecanismo de tolerância a falhas pode ser implementado através de redundância temporal, onde o anúncio de desconexões voluntárias e a detecção de desconexões involuntárias disparam ações corretivas.

Pelo que foi exposto, é possível observar que a arquitetura MoGrid pressupõe apenas a ocorrência de falhas do tipo *fail-stop*, apresentando para esses tipos de falhas mecanismos de tolerância baseados em redundância espacial. Entretanto, os mecanismos de redundância implementados não necessariamente garantem que as respostas às requisições de descobertas obtidas na presença de falhas sejam as de melhor qualidade. Os problemas decorrentes das falhas bizantinas, de omissão e temporais podem ser tratados na arquitetura MoGrid através da camada de transparência. Na Subseção 2.3.2, é apresentada uma discussão inicial sobre as medidas de correção que podem ser acionadas em função da ocorrência de falhas de omissão. Soluções para esses tipos de falhas devem ser incorporadas com a conclusão da especificação da camada de transparência.

2.5. Trabalhos Relacionados

Como é possível constatar nos trabalhos publicados na área de grades computacionais, nos últimos anos, há um interesse crescente em viabilizar aplicações baseadas na tecnologia de grades, sejam elas móveis ou fixas, para dispositivos portáteis, o que tem apontado modelos alternativos para computação em grade. Tal interesse tem motivado o desenvolvimento de grades P2P [Bhatia 2005; Fox et al. 2003; P-Grid 2002] e grades pervasivas [Geyer et al. 2005], embora na maioria desses trabalhos seja oferecido suporte apenas para que os dispositivos móveis sejam utilizados, exclusivamente, como interfaces para o acesso sem fio às grades fixas. Nesta seção, são descritos alguns projetos específicos que oferecem algum nível de suporte para as grades móveis ou que se aproximam desse propósito, oferecendo suporte a ambientes móveis de colaboração. Inicialmente será feita uma breve apresentação desses trabalhos, os quais serão posteriormente discutidos na Subseção 2.5.1.

ISAM (Infra-estrutura de Suporte às Aplicações Móveis distribuídas) [Yamin et al. 2003]. Projeto que propõe a integração de três conceitos principais – consciência do contexto, mobilidade e grade computacional – em um ambiente de computação pervasiva. Esse ambiente oferece suporte ao desenvolvimento de aplicações móveis distribuídas que apresentam um comportamento adaptativo. Pesquisas envolvendo o conceito de grades pervasivas em ambientes P2P foram desenvolvidas no contexto do projeto ISAM, através do Grupo de Trabalho da RNP **GRADEp** [Geyer et al. 2005], no período de 2004 a 2005. O projeto **GRADEp** teve, por objetivo, a extensão da perspectiva tradicional de computação em grade, pela inclusão de aspectos de mobilidade de dispositivos, de usuário e de componentes das aplicações. O *middleware* **GRADEp** oferece suporte a operações no modo desconectado para dispositivos fixos e móveis em uma grade fixa. O projeto oferece suporte à mobilidade, considerando em sua especificação que o ponto de acesso à rede fixa pode mudar com o tempo. Os dispositivos são associados a um identificador único de modo que eles possam ser localizados na grade independente do ponto de acesso ao qual eles estão conectados.

Um ambiente de resolução de problemas baseado em grades sem fio [Kurkovsky et al. 2004]. Este projeto propõe um ambiente de resolução de

problemas baseado em uma grade computacional móvel, organizada através de uma rede sem fio infra-estruturada, na qual os dispositivos móveis são visualizados como agentes colaborativos em um sistema multi-agente. Tal ambiente trata aspectos referentes à distribuição, coordenação e remontagem de tarefas complexas de grade, lidando com aspectos como instabilidade da rede, transparência de acesso e fidedignidade. Nesse projeto os dispositivos sem fio atuam como elementos computacionais da grade propriamente dita, disponibilizando os seus recursos e serviços, sendo acionados para a execução colaborativa de tarefas.

AKOGRIMO (*Access to KnOwledge through the GRId in a MObile world*) [Akogrimo 2004]. Este é um projeto promovido pelo fundo europeu, com o propósito de desenvolver arquiteturas e protótipos para as grades de próxima geração (*Next Generation Grid – NGGs*), baseadas na arquitetura OGSA, motivado pelo crescimento de usuários de dispositivos portáteis na Europa. Atualmente, o projeto AKOGRIMO tem como foco a proposição de aplicações inovadoras para grades móveis organizadas através de redes sem fio infra-estruturadas em WLANs (*Wireless Local Area Networks*), baseadas no protocolo IPv6, de modo que se possa oferecer serviços aos cidadãos comuns, auxiliando-os em atividades do seu dia-a-dia em um contexto comercial. O domínio de tais aplicações abrange *e-health*, *e-learning*, gerenciamento de crises e catástrofes, logística, turismo, entre outros. Entretanto, nas especificações do projeto não há referências sobre o tratamento dado a dispositivos com restrições de recursos. O projeto AKOGRIMO trata as questões referentes à mobilidade em um nível mais elevado, sob a perspectiva de serviços móveis.

MASGRID (*Mobile Ad hoc Service Grid*) [Ihsan et al 2005]. Nesse trabalho, os autores propõem uma arquitetura que ofereça funcionalidades similares às disponíveis no Globus Toolkit [Foster & Kesselman 1997], para o estabelecimento e utilização de uma grade móvel organizada através de uma rede sem fio *ad hoc*, de forma simplificada, de modo que essas funções possam ser utilizadas pelos dispositivos móveis que a constituem. A arquitetura define dois mecanismos principais: o serviço de descoberta de recursos (*Resource Discovery Service – RDS*) e o serviço de acesso aos recursos (*Resource Access Service – RAS*), os quais dependem, respectivamente, dos protocolos de conectividade e de

roteamento, utilizados na MANET. Os autores apresentam um modelo conceitual, definindo as funções que cada um dos mecanismos deve desempenhar; entretanto, eles não se detêm em aspectos operacionais referentes à sua implementação. A proposta do mecanismo de descoberta, apesar de ser brevemente apresentada, recebe maior atenção em relação ao mecanismo de acesso aos recursos.

2.5.1. Discussão sobre os Trabalhos Relacionados

O surgimento das grades móveis tornou-se viável devido à disponibilidade de dispositivos móveis com capacidade de processamento e comunicação cada vez maiores, o que é comprovado pelo número crescente de vendas de *notebooks* e PDAs em todo o mundo, chegando a casa dos milhões anualmente.³ Essa realidade proporciona um novo tipo de rede de compartilhamento de recursos que explora a larga disseminação dos dispositivos sem fio que, considerando-se a sua capacidade de interconexão, promove um ambiente de colaboração que não deve ser desprezado [McKnight et al. 2004]. Apesar das restrições de recursos apresentadas por esses dispositivos, em decorrência das suas dimensões cada vez mais reduzidas, a integração dos dispositivos móveis, interconectados através de enlaces sem fio, é defendida como uma possibilidade real de colaboração em uma grade computacional [Phan et al 2002]. O desafio de integração dos dispositivos sem fio em uma grade móvel é ampliado quando esses dispositivos estão organizados através de uma MANET de saltos múltiplos, mas não inviabilizado, como demonstram os estudos apresentados em [Red Herring 2005; Lima et al. 2005; Gomes et al. 2007].

Nos projetos anteriormente citados nesta seção, não é oferecido suporte ao desenvolvimento de aplicações de grade baseadas no compartilhamento de recursos em redes sem fio *ad hoc*, nas quais é oferecida uma visão mais focada do conceito de grades móveis – onde a infra-estrutura da rede é dinâmica, devido às constantes modificações provocadas pela mobilidade dos usuários. Para ilustrar o

³ Segundo o estudo “*Worldwide Quarterly PC Tracker*” da consultoria IDC referente ao quarto trimestre de 2006, as vendas de *notebooks* irão ultrapassar as dos computadores pessoais de mesa

que foi dito, pode-se mencionar o projeto GRADEp que, apesar dos resultados expressivos na inclusão de dispositivos móveis como elementos computacionais da grade, não se aplica às grades móveis pois foi desenvolvido para oferecer suporte à mobilidade de dispositivos e usuários – com ênfase em pervasividade – em uma grade fixa.

É imprescindível, em uma grade móvel, a adoção de uma abordagem cooperativa e completamente descentralizada para prover a descoberta e coordenação de recursos. A abordagem que mais se aproxima desse objetivo é aquela proposta em [Kurkovsky et al. 2004], que apresenta um ambiente para solução de problemas para grades móveis organizadas através de uma rede sem fio infra-estruturada. Entretanto, nessa abordagem, a responsabilidade de coordenar a distribuição e execução das tarefas da grade é delegada a um elemento central, conectado à rede fixa. Por outro lado, durante o período de vigência (2004 – 2007) do contrato europeu para implementação do projeto AKOGRIMO, vários resultados foram publicados. Esses trabalhos, entretanto, apesar de também apostarem na utilização dos recursos disponíveis na grade móvel para a provisão de serviços para os dispositivos da grade, baseiam sua arquitetura na presença de elementos centrais, com alta concentração de recursos, para coordenar os mecanismos associados ao compartilhamento de recursos e serviços.

A arquitetura MASGRID foi especificada para oferecer suporte a grades móveis *ad hoc*, entretanto, os mecanismos propostos não foram desenvolvidos, de modo que não existem resultados que comprovem a sua aplicabilidade. De modo similar, no projeto coreano K*Grid [KGrid 2002], foi apresentada uma proposta promissora com o intuito de oferecer suporte à formação de grades móveis pervasivas, organizadas como WLANs, investindo no uso de recursos ociosos dos dispositivos sem fio. Entretanto, não há, até o presente momento, nenhum registro da evolução do projeto K*Grid nessa área específica, motivo pelo qual ele não se encontra entre os trabalhos apresentados nesta seção.

Uma outra possibilidade de comparação mais específica, que no entanto merece ser mencionada, diz respeito aos requisitos da camada de transparência, descrita na Subseção 2.3.2, os quais se assemelham, em muitos aspectos, às funcionalidades oferecidas pelas arquiteturas de provisão de qualidade de serviço (*Quality of Service – QoS*) para redes sem fio. Nas grades móveis, exemplos de parâmetros de QoS relacionados à execução de serviços incluem o tempo de conclusão das tarefas e o grau de precisão dos resultados. Já em relação a capacidade dos dispositivos da grade móvel, pode-se considerar a taxa de ocupação do processador, o percentual de memória disponível e a qualidade do enlace sem fio, entre outros.

No contexto das arquiteturas de provisão de QoS, o *middleware* COLOMBA (*Context and Location-based Middleware for Binding Adaptation*) [Bellavista et al. 2003] oferece suporte à associação dinâmica de recursos através da exploração de metadados. Esses metadados são utilizados para descrever recursos, dispositivos, informações sobre os usuários, estratégias preferenciais e serviços do *middleware*. O *middleware* é estruturado em dois componentes: (i) o *binder manager*, que intermedia o acesso aos recursos de acordo com a política especificada e (ii) o *policy manager*, que oferece suporte à especificação, instalação dinâmica e alocação de políticas.

Em [Lima 2002] é proposto um conjunto de *frameworks* para provisão de QoS em redes sem fio baseados em conceitos como (i) noção de intervalos de QoS, (ii) adoção de parâmetros de QoS relacionados à mobilidade, (iii) classes de QoS, (iv) reservas antecipadas de recursos e (v) políticas de controle de acesso aos recursos. Abordagem similar é adotada pelo grupo de pesquisa TIMELY (*The Illinois Mobile Environment LaboratorY*), que propõe uma arquitetura de gerenciamento de recursos adaptativa para ambientes de computação móvel [Bharghavan et al. 1998]. A arquitetura oferece um modelo de serviços adaptativo que permite às aplicações e aos gerentes de recursos renegociar a QoS em função de alterações nas condições da rede sem fio.

Como as aplicações de grade exigem um processamento intensivo, elas têm o seu desempenho afetado pela mobilidade dos dispositivos que participam da sua execução. Se esses dispositivos apresentam um perfil de mobilidade elevada, isso pode impactar de forma adversa no tempo necessário para concluir a execução da tarefa. Fazendo um paralelo entre os mecanismos de gerenciamento de recursos

das arquiteturas de QoS mencionadas anteriormente e os requisitos da camada de transparência da arquitetura MoGrid, é possível observar que a questão da mobilidade está diretamente relacionada a questão da provisão de qualidade de serviço. O objetivo fim das aplicações colaborativas desenvolvidas para as grades móveis é oferecer aos usuários de dispositivos móveis em redes sem fio as mesmas garantias de QoS experimentadas pelos usuários de dispositivos computacionais em redes fixas. Vale ressaltar que mesmo nos ambientes fixos, as “garantias” oferecidas durante a execução de um serviço não são garantias “absolutas”, o que se agrava ainda mais em ambientes computacionais móveis, nos quais essas garantias estão condicionadas à qualidade do enlace sem fio e à mobilidade dos dispositivos. Dessa forma, as garantias são válidas apenas se os recursos do dispositivo e do enlace sem fio não sofrerem variações que degradem o seu desempenho.

Um aspecto de fundamental importância, comum tanto à camada de transparência – especialmente à subcamada de acesso transparente aos recursos – quanto às arquiteturas de provisão de QoS, diz respeito ao monitoramento da qualidade do serviço durante a sua utilização, visando a manutenção da QoS. No processo de manutenção da QoS, em alguns casos pode mesmo haver a necessidade da substituição dos provedores de serviço, já que a ocorrência freqüente de desconexões dos dispositivos pode resultar em um nível de QoS insatisfatório. De acordo com as especificações definidas na Subseção 2.3.2, na arquitetura MoGrid essas questões devem ser tratadas na subcamada de acesso transparente ao recurso, com a substituição dos colaboradores – no caso de desconexões voluntárias ou involuntárias – e a migração de tarefas; ações que “garantem” a provisão de um nível mínimo de QoS na grade móvel. Esse procedimento se assemelha ao mecanismo de sintonização,⁴ utilizado nas arquiteturas de QoS. Esse mecanismo deve ser investigado com o intuito de enriquecer as discussões quanto aos requisitos necessários para a especificação da camada de transparência, podendo ser incorporado no seu desenvolvimento, com algumas adaptações, para atender as necessidades de uma grade móvel.

2.5.2. Análise Comparativa

A seguir, são definidos, de forma breve, alguns parâmetros para mensurar o quanto as diferentes propostas, apresentadas nesta seção, adequam-se aos cenários de mobilidade das redes sem fio, destacando a forma como as características desse tipo de rede são tratadas nos projetos aqui mencionados. Os parâmetros que serão apresentados nos próximos parágrafos são utilizados, na Tabela 1, como base de comparação entre as propostas discutidas nesta seção e a arquitetura MoGrid.

Provisão de Suporte a Redes Sem Fio Infra-estruturadas e Ad hoc. Para que uma arquitetura de descoberta de recursos e serviços atenda às necessidades de uma grade móvel, de acordo com a visão defendida nesta tese – onde os dispositivos sem fio desempenham o papel de elementos computacionais da grade móvel, não atuando apenas como consumidores de recursos e serviços –, a arquitetura deve oferecer suporte tanto às redes sem fio infra-estruturadas quanto às redes sem fio *ad hoc*.

Gerenciamento de Mobilidade. Em grades móveis, a taxa de mobilidade dos dispositivos pode ser elevada e o tempo de permanência dos dispositivos na grade é variável, caracterizando altas taxas de *churn*. Portanto, é importante que se considerem aspectos relacionados à mobilidade e, conseqüentemente, aos períodos de desconexão, na especificação de uma arquitetura para esse ambiente.

Diretório de Serviços Distribuído. Uma arquitetura de descoberta de recursos e serviços é definida em função do uso de um serviço de diretório – que pode ser classificado em centralizado ou distribuído – para armazenar as informações sobre os recursos e serviços disponíveis na grade. A escolha do tipo de diretório influencia a forma como as informações sobre os serviços são recuperadas no processo de descoberta. No caso de um diretório centralizado, os clientes devem inicialmente obter a identificação de rede do diretório para então consultá-lo sobre os serviços disponíveis. Nesse caso, pode-se ainda implementar no diretório um mecanismo de notificação de informações sobre os serviços, que é

⁴ Na provisão de QoS o mecanismo de sintonização é responsável pelo redimensionamento da parcela de utilização dos recursos, a qual foi disponibilizada ao usuário do serviço durante a

ativado a partir do registro de interesse do cliente a respeito de informações sobre recursos e serviços específicos. No caso de um diretório distribuído, cada dispositivo da rede deve armazenar localmente as informações a respeito dos serviços que ele próprio disponibiliza, podendo ainda utilizar uma estratégia de *cache* para armazenar também as informações sobre os demais serviços oferecidos na grade, obtidas através de um mecanismo de anúncios periódicos de serviços. Essa abordagem é recomendada para as redes sem fio *ad hoc* nas quais a ausência de qualquer infra-estrutura pré-existente obriga que os serviços oferecidos sejam especificados de forma flexível e distribuída, sem a dependência de um ponto central de coordenação. O uso de diretórios de serviços distribuídos evita que o deslocamento ou mesmo a desconexão dos dispositivos comprometa o funcionamento do mecanismo de descoberta da grade móvel.

Transparência de Acesso. Uma arquitetura de descoberta de serviços deve oferecer mecanismos que garantam a transparência de acesso ao recurso pelo dispositivo que o solicita. Na grade móvel, o importante é a disponibilidade do recurso e não a sua origem ou localização. A arquitetura deve oferecer suporte ao gerenciamento do problema da desconexão dos dispositivos móveis, tanto as voluntárias quanto as involuntárias.

Heterogeneidade. Nas grades móveis, um fator de complexidade adicional diz respeito à heterogeneidade de *hardware*, de *software* e dos ambientes nos quais os dispositivos sem fio estão conectados, envolvendo diferentes tecnologias de rede, como Ethernet, Wi-Fi, celular, entre outros. Os novos ambientes propostos pelas grades móveis apontam para o compartilhamento de recursos e serviços através de redes com enlaces intermitentes e dispositivos finais que não necessariamente adotam políticas de segurança. Nesse novo ambiente, é importante a adoção de mecanismos para descoberta e acesso aos recursos e serviços que sejam independentes do protocolo de roteamento utilizado na MANET e não se restrinjam a uma tecnologia de rede sem fio específica. Outro fator que deve ser considerado é a diversidade de plataformas de *hardware* e sistemas operacionais. É de fundamental importância a utilização de mecanismos

negociação da QoS, na fase de estabelecimento do contrato de serviço.

para oferecer portabilidade como, por exemplo, o uso de uma linguagem de desenvolvimento multiplataforma.

Migração de Execução. Um problema específico que deve ser considerado é a migração da execução de tarefas entre colaboradores quando, por exemplo, um dispositivo infere que está prestes a perder a sua conectividade com a grade móvel que ele integra. A migração de tarefas deve ser realizada, idealmente, com a manutenção do estado de execução, o que aumenta consideravelmente a sua complexidade. Outro fator agravante nesse processo diz respeito a heterogeneidade de dispositivos, o que implica em *hardware* e *software* distintos entre os dispositivos da grade móvel; esse fator deve ser considerado ao se definir uma API para a migração de execução de tarefas.

Descrição de Recursos. A descrição de recursos deve ser implementada de forma padronizada a fim de garantir a uniformidade entre as requisições e as descrições de recursos publicadas, promovendo a interoperabilidade entre redes e equipamentos. É importante considerar mecanismos de adaptação que possibilitem a extensão do modelo adotado, sem prejuízo ao processo de descoberta.

A Tabela 1 apresenta um resumo comparativo entre os trabalhos apresentados nesta seção e a arquitetura MoGrid. Os símbolos “T”, “NT”, “PT” e “?” indicam, respectivamente, “tratado”, “não tratado”, “parcialmente tratado” e “não pôde ser avaliado”.

Tabela 1 – Resumo comparativo dos projetos relacionados à arquitetura MoGrid.

| | MoGrid | ISAM/ GRADEp | [Kurkovsky et al. 2004] | AKOGRIMO | MASGRID |
|---|--------|-----------------|----------------------------|-----------------|---------|
| <i>Rede Sem Fio: Infra-estruturada Ad hoc</i> | T T | T NT | T NT | T NT | T NT |
| <i>Gerenciamento de Mobilidade</i> | PT | NT | PT | T | T |
| <i>Diretório de Serviços Distribuído</i> | T | NT | NT | NT ⁵ | T |
| <i>Transparência de Acesso</i> | PT | T | T | ? | ? |
| <i>Heterogeneidade</i> | T | NT | NT | NT | NT |
| <i>Migração de Execução</i> | NT | PT ⁶ | NT | PT | NT |
| <i>Descrição de Recursos</i> | PT | PT | ? | T | NT |

Dentre as arquiteturas apresentadas as abordagens propostas em [Kurkovsky et al. 2004], AKOGRIMO e MASGRID são as que mais se aproximam da visão de grade móvel defendida nesta tese, ou seja, um ambiente onde os dispositivos móveis atuam como elementos computacionais da grade propriamente dita. Dessas propostas, apenas a arquitetura MASGRID se propõe a oferecer suporte a uma grade móvel *ad hoc*, as demais se limitam a uma WLAN infra-estruturada, apresentando dependências quanto às tecnologias de rede e de *software*, o que viola a condição de heterogeneidade inerente às grades móveis. Entretanto, a arquitetura MASGRID é apresentada como uma proposta preliminar, sem especificar detalhes a respeito das soluções adotadas e muito menos oferecer uma implementação que possa ser utilizada. Por sua vez, nas soluções apresentadas em [Kurkovsky et al. 2004] e AKOGRIMO a especificação da arquitetura de descoberta de recursos se baseia na presença de um elemento central, um dispositivo rico em recursos conectado tanto à rede fixa quanto à rede móvel. Essa solução é inviável para as grades móveis organizadas através de redes sem fio *ad hoc* devido a ausência de um elemento central estável que possa gerenciar os processos de descoberta de recursos e de execução de tarefas, dado que nesse tipo

⁵ Adota um serviço de diretório centralizado, baseado em uma implementação do SLP [Guttman 1999].

⁶ A migração da execução de tarefas no GRADEp é gerenciada pelo próprio usuário, de forma programada, e seu alcance se restringe ao escopo da rede local; não há gerenciamento de mobilidade física.

de rede todos os dispositivos são móveis e todos se conectam através do enlace sem fio. Além disso, em redes *ad hoc* situações de particionamento são freqüentes, o que provocaria a ausência do serviço de diretório, por um período indeterminado, em $n-1$ partições da rede, considerando n igual ao número de partições. A única alternativa que pode minimizar o efeito do particionamento das redes *ad hoc* na disponibilização das informações sobre os serviços da grade é a replicação dessas informações através da utilização de vários servidores de diretórios. Entretanto, ao se aumentar o número de servidores cria-se um novo problema, o da manutenção da consistência entre eles, cuja solução apresenta um nível de complexidade elevado em redes fixas, o qual pode crescer consideravelmente em redes sem fio *ad hoc*, inviabilizando o uso dessa alternativa [Carter et al. 2003; Farkas et al. 2004].

2.5.3. Contribuições Alcançadas

A arquitetura MoGrid foi proposta com o intuito de oferecer suporte ao desenvolvimento de aplicações para grades móveis e auxiliar a tomada de decisões referentes à descoberta e à seleção de recursos, bem como ao acesso e a utilização desses recursos. Algumas propostas foram desenvolvidas para oferecer suporte às grades móveis, entretanto, a maior parte dessas propostas tem adotado abordagens centralizadas, baseadas em redes fixas ou em redes sem fio infra-estruturadas [Yamin et al. 2003; Kurkovsky et al. 2004]. A arquitetura MoGrid é independente de elementos centralizados e provê um ambiente de execução de tarefas que oferece, ao desenvolvedor de aplicações de grade, um conjunto de funcionalidades que tratam os aspectos relacionados à colaboração entre dispositivos heterogêneos em uma rede sem fio.

Esta tese traz entre as suas principais contribuições a especificação de uma arquitetura para grades móveis que oferece suporte tanto às redes sem fio infra-estruturadas quanto às redes sem fio *ad hoc*, tendo sido concebida de modo a também ser integrável com tecnologias de grade fixa. A arquitetura MoGrid adota uma abordagem *peer-to-peer*, totalmente descentralizada, independente da tecnologia de rede utilizada e de protocolos de roteamento em uso na MANET, tendo sido projetada para atender às necessidades específicas das grades móveis.

Por ter sido especificada de forma modular, adotando uma abordagem em camadas, os diferentes mecanismos relacionados à descoberta (camada de descoberta) e à utilização de serviços (camada de transparência) são separados, podendo ser utilizados como módulos independentes, acoplados, inclusive, em outras arquiteturas. Como é o caso do protocolo de descoberta e seleção de recursos P2PDP (Capítulo 4) e do algoritmo de supressão de mensagens de resposta por vizinhança (Seção 4.6).

No próximo capítulo, os trabalhos relacionados à área-tema desta tese – descoberta de serviços para redes sem fio *ad hoc* – são apresentados. As peculiaridades de cada trabalho ilustram os diferentes critérios apontados na classificação genérica para os protocolos de descoberta de serviços introduzidos no mesmo capítulo. Na revisão bibliográfica é dada uma maior ênfase aos trabalhos que tratam, especificamente, da descoberta de serviços em redes sem fio *ad hoc* de saltos múltiplos, abordagem similar à adotada no protocolo para descoberta e seleção de recursos em grades móveis *ad hoc* proposto nesta tese, denominado P2PDP (*Peer-to-Peer Discovery Protocol*). O protocolo P2PDP é descrito em detalhes no Capítulo 4.